Understanding Sorting Algorithms Using Music and Spatial Distribution

By

Richard N Mumford

A Doctoral Thesis

Submitted in partial fulfillment of the
requirements for the award of

Doctor of Philosophy
of
Loughborough University
March 2002

Abstract.

This thesis is concerned with the communication of information using auditory techniques. In particular, a music-based interface has been used to communicate the operation of a number of sorting algorithms to users. This auditory interface has been further enhanced by the creation of an auditory scene including a sound wall, which enables the auditory interface to utilise music parameters in conjunction with 2D/3D spatial distribution to communicate the essential processes in the algorithms.

The sound wall has been constructed from a grid of measurements using a human head to create a spatial distribution. The algorithm designer can therefore communicate events using pitch, rhythm and timbre and associate these with particular positions in space. A number of experiments have been carried out to investigate the usefulness of music and the sound wall in communicating information relevant to the algorithms. Further, user understanding of the six algorithms has been tested. In all experiments the effects of previous musical experience has been allowed for.

The results show that users can utilise musical parameters in understanding algorithms and that in all cases improvements have been observed using the sound wall. Different user performance was observed with different algorithms and it is concluded that certain types of information lend themselves more readily to communication through auditory interfaces than others.

As a result of the experimental analysis, recommendations are given on how to improve the sound wall and user understanding by improved choice of the musical mappings.

Acknowledgements.

I would like to thank the following people for their help and support during the period covered by this Ph.D. project.

Prof. J.L. Alty for his enthusiasm, guidance, and ideas. He proved to be not only an invaluable source of information and good advice, but also an understanding and generous human being.

The staff and students at Loughborough University who acted as guinea pigs for the preliminary testing of the system and who provided much invaluable feedback. My close friends without whom there would have been no experimental data and also to Ally for her loving support in the final stages.

Finally, my warmest thanks go to my parents Lee and Chris. They have helped me to achieve my goals in so many directions in life. Without them I would have spent much of my time unnecessarily worrying and being lost in the wilderness of uncertainty.

Contents. **Page.**

List of figures

Chapter 1
Introduction

The aim of this document is to provide the reader with a background into spatial and non-spatial auditory displays. It focuses upon the specific use of non-speech interfaces and further concentrates on the use of music and spatial enhancement at the human-computer interface.

Chapter 1 gives a brief introduction to the topic covered by this thesis.

Chapter 2 introduces the general topic of the auditory medium. Further to this it raises some issues of concern and discusses them in an effort to resolve some of the more common problems with using non- speech audio at the human- computer interface. More specifically to this thesis it presents the topic of algorithm auralisation.

Chapter 3 introduces the basics of acoustics and explores the properties associated with sound source localisation. It then highlights and evaluates several approaches and implementations for spatial audio and 3D audio synthesis.

Chapter 4 analyses some common sorting algorithms and makes selections for some of these algorithms to be explained in greater detail. The algorithms are then assessed for their appropriateness for musical auralisation.

Chapter 5 reports preliminary experimental work carried out in this thesis pertaining to basic pitch and shape perception.

Chapter 6 introduces SIMBAA, a musical auralisation toolbox and early experimentation.

Chapter 7 introduces SIMBAA 3D a spatially enhanced musical auralisation toolbox and its associated design considerations and implementation.

Chapter 8 reports on experimental work carried out using SIMBAA 3D.

Chapter 9 draws conclusions about spatially enhanced algorithm auralisation and makes recommendations for future work and further enhancements.

The most exploited medium for computer - human communication to date is the visual medium. Simple still pictures, full motion pictures and Virtual Reality are all examples of this medium. However, there are many other media which can be exploited in Computer - Human communication and one obvious possibility is the auditory channel. Until very recently this medium has conveyed only simple single or multi-tone notes, usually indicating some form of error on the users part. Recent research, however, has seen the emergence of new information presentation formats, many utilising the properties of Auditory Displays. Such auditory displays can be utilised either as a complementary channel to a visual one, or on their own as an autonomous communication medium.

A Graphical capability has been available for many years, but inexpensive audio facilities are much more recent additions to the personal computer. This is surprising since early computer users often used sound. A common anecdote is that of early programmers who tuned an AM radio to pick up the radio interference emitted by the computer. By listening to the patterns of sounds in the interference they learnt to monitor CPU behaviour and even to identify errant program behaviour.

Although the lack of standards for sound equipment has acted as a deterrent, the emergence of the Musical Instrument Digital Interface (MIDI) specification [145] has provided a common language, although it is oriented towards the communication of musical data and not that of sound generally. By the time affordable sound generating equipment became available to the average computer user, graphical facilities were well advanced. So, for largely technological reasons, the human-computer interface has, from the start, been almost entirely visual in its construction. This may have helped to foster the belief that computer users tend to employ mental imagery in a visual form.

With advances in display technology came an inertia that led to an increasing bias towards visual interfaces. This is reflected in the natural language of those cultures that rely on the written word for communication, which, by using words like 'imagery' to describe mental processes, shows an inclination towards visual metaphors for the explanation of ideas. The very act of thinking is defined using the visually oriented word 'imagine'. Given our tendency to represent ideas as images in our minds and the historical development of visual display hardware, it is not unreasonable that the emphasis in software development and human computer interface design has focused predominantly on the visual medium.

Recent suggestions concerning how to exploit the auditory channel have focused upon the use of music within multimedia, since there is limited reported work on the subject. Music is the most sophisticated of the auditory media, allowing the conveyance of large amounts of information in parallel. Although music is a rich medium containing numerous structures introduced by musicians over many years of human evolution and multimedia systems are fully capable of producing musical sounds relatively easily and effortlessly, the use of music in interfaces is currently at a relatively low level. Music and, in particular, the auditory channel as a whole, has been neglected in the development of user-interfaces possibly because there is very little known about how humans understand and process music.

It is not intuitively obvious how to use musical structures in interface design. However, there has been some research in the field of audio in interface design that includes Gavers' SonicFinder [92], a system that uses natural sound to indicate the state of the natural environment, Earcons [18] from Blattner et al which maps audio onto visual representations of tasks, sonically enhanced graphical buttons by Brewster et al [35] and Gaver et al's ARKola simulation [96].

There are several valid reasons why the auditory channel and music in particular should be further investigated:

- The auditory channel has been somewhat neglected in the area of user interface design. This is despite the fact that auditory interaction is one of the primary forms of human interaction.
- Music has a number of powerful properties such as pitch, rhythm and melody that ought to be able to convey rich messages from software components to the user.
- Music, as well as other forms of auditory output, is of a particular value when the user cannot be disturbed visually.
- The visual channels are becoming very cluttered. For instance, current monitors are often very overcrowded yet designers still try to present more information visually.
- When output is directed to users who do not have constant visual contact with the VDU (Video Display Unit) screen, an alert or interrupt is required.
- This over-emphasis on visual communication presents serious interface difficulties for visually impaired users.

In the current information age, more and more people with diverse backgrounds and experience use computers as part of their daily work both in their work and home environments. Music is also an integral part of most people's daily lives. Research in the area of using auditory-musical stimuli in HCI may therefore benefit a large proportion of computer users.

The use of music as a communication metaphor could therefore assist in a number of interface situations including the following:

- Reducing the complexity of visually crowded user screens by presenting some information using music.

- Presenting information of a graphical nature to blind users who usually interact with computers using speech.

- Auralisation[1] of the internal execution of algorithms, in particular, sorting algorithms. This has particular implications in understanding sorting algorithms and debugging programs through auditory means.

Recent research into the use of music to communicate algorithm state and execution and program execution and debugging has seen the development of ZEUS by Brown and Hershberger [46] and CAITLIN by Vickers and Alty [181]. These systems have shown that music can be used effectively to communicate information to users. CAITLIN was primarily concerned with using metaphorical musical cues to aid novice programmers with debugging. ZEUS communicated algorithms using auditory means supplemental to visual representations, but no formal or empirical evaluation was carried to determine the effectiveness of the mappings or the degree of algorithm state and understanding attainable through algorithm auralisation. Auralisation systems such as ZEUS and CAITLIN are discussed in more detail in the following chapter.

One additional parameter in auditory presentation is stereophony. Existing attempts at algorithm and program auralisation have confined themselves to using a common stereophonic presentation format. However, more complex presentation formats now exist that will allow the exploitation of spatial location of sound sources within a three-

---

[1] *Auralisation*, a term suggested by Brown and Hershberger [45], typically refers to the mapping of program data to sound and is based on the execution of the program or algorithm.

dimensional environment. These might be employed to provide spatial enhancement to algorithm and/or program auralisations to further disambiguate the presented information through the use of spatial location and movement as extra auditory cues. There are currently several methods of enhancing auditory presentations with 3D sound ranging from simple stereophonic field extension (which is not really true 3D but more of a commercial exploitation of the 3D logo) through to more complex and thorough 3D sound modelling and synthesis systems that employ complex filtering to spatialise sounds. Within these extremes of 3D audio technologies there are several realistic and cost-effective techniques for producing spatially enhanced audio that could be readily applied to algorithm and/or program auralisation.

The main purpose of this thesis is to examine how relatively inexpensive 3-D sound techniques can be used to improve disambiguation of musically auralised sorting algorithms. This thesis is also concerned with the effect that musical training has on understanding such sorting algorithm auralisations. The emphasis on sorting algorithms is due their diverse range of events, sorting natures and data. Many other information sources exist that could be well suited to auralisation. However, this thesis is not concerned with defining which types of information sources are best suited to auralisation. It is more concerned with using sorting algorithms as a vehicle for preliminary experimentation of communicating information via spatially enhanced music. A detailed investigation is proposed in order to determine which types of information within sorting algorithms are more amenable to auralisation.

Chapter 2

Investigating the auditory medium

2.1. A multimedia approach.

Although most interface design is predominantly visual in nature, this thesis is specifically concerned with the use of the *audio medium* in multimedia interfaces as an output medium. Even though their use has been neglected, audio interfaces have a number of advantages over visual interfaces. For example, the user can work on a visually oriented task whilst listening to instructions, thus employing two media with minimal confusion. Audio interfaces are also useful when the recipient is moving around and the hands are busy. Clearly, visually handicapped people can also benefit from use of the audio channel.

In human-human communication the audio channel has long been established as a medium for communicating rich meaning. However, with the huge growth in graphical user interface design the auditory channel has been somewhat neglected. This has put visually impaired users at a great disadvantage. Such users were originally able to use computers via the use of applications such as screen readers. Unfortunately most modern interfaces are designed with the assumption that the target users have full visual abilities. Although this thesis is not concerned with providing an auditory presentation for the visually impaired, it is an issue that is often considered central to auditory interface design.

In order to encompass a greater variety of potential users, it is necessary to create an interface that could be interacted with via the auditory medium alone, the visual medium alone, or a combination of both. The user could then be presented with a choice of display format that may be employed at the user's discretion. Alty has called this "an equal opportunities interface" [4]. In a combined mode, the auditory medium could be used to assist the visual representation of information and reduce the visual 'clutter' often encountered by many computer users. For computer users with considerable visual impairments the auditory channel offers itself as an obvious communications medium and some examples of useful explorations are works carried out by Edwards [85], Rigas [159, 5] and Kennel [116]. These systems use music via the auditory medium to represent visual entities to visually impaired users. Since most operating systems use fully integrated GUIs (graphical user interfaces), such approaches could offer a bridge to conventional systems.

The types of auditory media available can be conveniently divided into three sub-areas – speech, sound and music.

2.2. Speech.

Text can be conveyed in two ways, either written or spoken (visual or audio). When spoken the speaker may convey emotion through the tone of voice used, in addition to the raw verbal information. In contrast, written information can convey emotion or emphasis through the use of special fonts, italics, emboldenment or capitalisation. A user need not be literate to understand oral communication, and spoken information is clearly of considerable use to the visually impaired. The Speech medium has traditionally been used to assist visually challenged people, for example through the use of screen readers, but the recent emphasis on GUIs has made this difficult. For the purposes of representing information through sound, speech offers very little by way of a solution.

2.3. Sound.

2.3.1. Sound and Human-Computer Interaction

The audio channel remains little used in interface applications although there now exists a reasonable amount of work in the field. In most cases, where the sound medium has been utilised, its usage is often trivial, for example, Microsoft's arbitrary association of sound files with system events in its operating systems. Most of these individual applications of sound do not usually enhance the computer-user interaction experience.

Moreover, they often serve as novelties rather than conveying any real meaning. However, there have been some useful applications of sound in Microsoft's systems, such as notification of task completion or mail notification.

Due to the obvious high information carrying capacity of the visual medium many people tend not to think of sound as an alternative computer-human communication medium. Auditory signals however, can be used to build complex mental images. As humans we naturally associate sounds with real world objects. By simply listening to the engine of an aeroplane we can determine the size of the craft and in some cases gain a feel for its altitude, speed and direction. This is all possible without the need to see the object. We simply construct the images from prior experience. This association might be of use for algorithm auralisation, since metaphorical mappings could be learnt by the listeners.

When the current usage of the sound medium in computer interfaces is compared with the sophisticated use of visual display techniques, the use of sound at the human-computer interface has been limited. This is surprising since sound is a most important communication channel for human beings, and should have much to offer in assisting human-computer interaction, in particular it might be useful in understanding complex structures and states like those present in algorithm executions.

Bly's research [22] on the use of sound in interfaces was among the first investigations into this area and since her initial work, the body of research has grown slowly. A majority of auditory display work has, until recently, concentrated on supporting existing visual interfaces (graphical user interface). However, work on computer-based icons has now been extended to the audio medium via the use of auditory icons [92, 93, 2, 94, 95], which are essentially symbolic sound effects.

Non-speech audio is a well-proven communication medium, and is extensively used in the film industry to supplement motion pictures. Although it can be effective on its own (for example radio productions) it can be highly complementary and supportive for the visual medium. It can add value in the following ways:

- To represent unseen entities - Sound can enable us to picture things in our minds eye that are not visible on a display, it can indicate specific situations and extend the visual display beyond what is actually visible.

- As a feedback mechanism - Sound can be used to acknowledge actions or can be used to signify process status. It can be used as a cue or as in Gavers' SonicFinder [92] a feedback for actions in a Graphical User Interface. Predictive sound can be used to signify impending events, much like ominous music warns of up-and-coming danger in a film [202].

- To improve perceptions of quality - The impression of quality of a multimedia work is influenced more by the quality of the audio rather than the quality of the visual media. This has been shown in tests by researchers at MIT's Media Lab [7].

- To support visual interfaces - Sound can considerably enhance a visual interface. The audio effects can punctuate and emphasize a visual action. The impact of many films would be considerably reduced if their soundtracks were removed or even just degraded. Sound is also valuable for communicating additional layers of information as users can listen to sounds without having to compromise their attentions from the visual information.

- To grab attention - It is easy to miss visual information. However, a user is less likely to miss an audio message due to the intrusive nature of the auditory medium.

- Cohen [53] offers the following reasons for adopting sound to notify users of events:

  - Audio does not take up screen space.
  - Audio fades into the background but users are alerted when it changes.
  - People can process audio information while simultaneously engaged in an unrelated task.

- The cocktail party effect [8] (the ability to selectively attend to one conversation in the midst of others in a crowded room) allows users to monitor multiple background processes via the audio channel so long as the sounds attributed to each process can be distinguished.
- Most direct manipulation tasks are visual, leaving the audio channel free.

Audio clearly has many beneficial properties and it might be useful to investigate the types of information it could convey and if any special training would be required in order to use it effectively. Brown, Newsome and Glinert [44] showed that complex auditory cues could be used to replace visual cues. The prima-facie case behind their research was to try and ascertain if sound could be used to reduce visual workload. With the increased complexity of visual user interfaces, screens have become larger and often involve multiple VDUs. This permits the conveyance of greater amounts of information. However, in some applications, the information transfer between computer and human is near saturation and the user may not be able to effectively process all the information being presented. Such difficulties suggest that sound might be a useful addition in the presentation of information, either autonomously or supplementary to the visual medium.

Brown, Newsome and Glinert [44] undertook a study to find out if information that is typically presented visually could be communicated effectively using sound. The experiment was primarily concerned with a subject's ability to locate a target character string on a computer screen using both auditory and visual cues. The results showed that subjects were equally successful in understanding the auditory cues as they were with the visual cues. This is encouraging for the field of representing complex information via sound since it suggests that the audio channel could be used to convey information typically presented visually. They also found that the human brain could extract multiple messages from a sound very quickly and then act on the information given.

Walker and Scott [185] carried out work that involved the experimental testing of perceived lights, tones and gaps. They found that humans judged one light as having a shorter duration than an identical one when a tone was also played. The durations of the lights in both cases were the same, but the durations of the tones were different. This indicates that what we perceive in the visual channel is easily influenced by information presented simultaneously in the auditory channel. They stated "auditory dominance occurred under the preceding conditions, that is auditory - visual conflicts in perceived durations were resolved in favour of the auditory modality[1]." This supports the idea of using the auditory medium for conveying information and highlights the power of its influence on our perception.

Walker and Scott [185] suggest that the auditory medium should be used for conveying temporal information and the visual medium should be used to convey spatial information. However, Perrot et al [153] found that the auditory medium could also convey spatial information effectively, since it can speed up location and identification of objects within the spatial domain. O'Leary and Rhodes [148] found that ambiguity of information in one mode can be resolved through information from another mode. This supports the findings of Walker and Scott. This suggests that the auditory channel could be used to assist a visual representation. However, this thesis is concerned with investigating the use of the auditory channel autonomously. Conversely, Wagenaar et al [184] showed that combining modalities does not necessarily have a beneficial effect. Experiments carried out by Paivio [149, 150, 151] showed that recall and recognition can be improved by presenting information in both visual and verbal form. Paivio's Dual Coding Theory [149, 150, 151] assumes that there are two separate cognitive sub-systems for processing both verbal and non-verbal representations.

The task of choosing appropriate modalities for communicating information is an important one. The choices for mappings must also be a careful one as misconception of the events or tasks can occur. Familiarity can be exploited. We often have a clue as to the meaning of a sound by the context in which it is presented, and from our previous experience. This highlights the importance providing a context when conveying complex information via the auditory channel.

---

[1] Modality is defined as a prescribed method of procedure.

Gaver suggested that sound could be used to provide a 'sonic landscape' [93], which can help us to navigate through complex information spaces. He proposed that issues such as ambient audio and peripheral awareness would be critical to future of interfaces and applications.

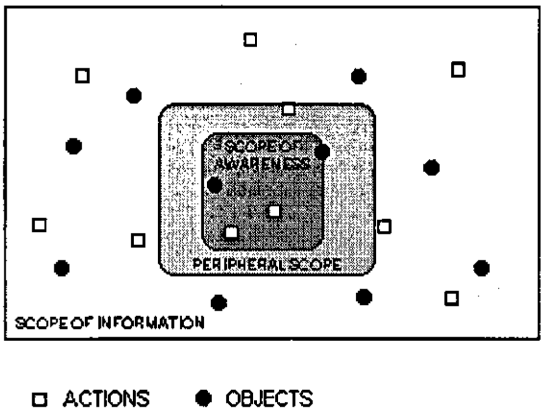


Figure 2.1 - Scope of Visual medium.

Figure 2.1 represents the conceptual scope of visual peripheral awareness to objects and events. It indicates that there exists an area of focus (area of concentration) and shows that, in most cases, the scope of information reaches far outside of this. This is why graphical interfaces can become cluttered; the information that is not in focus is often heaped into the scope of awareness. The boundaries of the peripheral scope can be seen as the edges of a VDU, the scope of awareness is then the area upon which we focus our eyes. Anything that occurs peripheral to our focus demands that we move this window of attention to the source. This highlights the limitations of the visual medium for presenting large amounts of information. The problem has initiated investigations into the use of such techniques as the 'fish eye' lens [90] which represents the visual information in a dominant magnified area in the central focal field, while other peripheral (and contextual) information is represented in a less magnified area surrounding the central focal field.

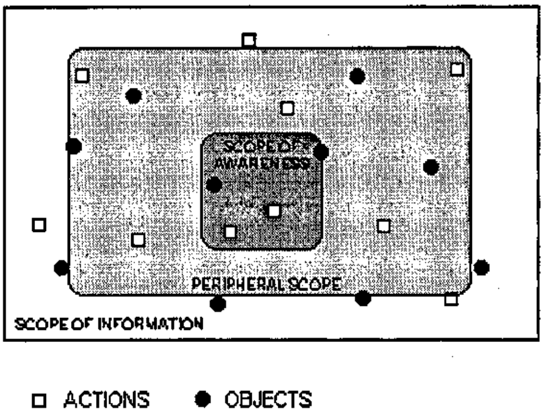


Figure 2.2 - Scope of Auditory medium.

Figure 2.2 shows the conceptual awareness and peripheral scope in the auditory medium. As can be seen the difference is that the peripheral scope is much wider, indicating that shifting our attention from one event to another is faster and the awareness of peripheral events is greater. We are therefore not limited by the focusing constraints of vision. Sound does not depend so much on the direction that the user is facing. An event that takes place behind can often not be seen, but it can usually be heard. This supports the use of spatial audio in auditory displays since it can be exploited to convey information that would otherwise add to the clutter of a visual display..

2.3.2. Problems with the auditory medium

Since the use of sound to convey complex information is the main area of concern to this thesis, it is important to highlight some of the possible problems when using audio in information display. Kramer [119, 120, 121] has highlighted the low resolution of the auditory medium in relation to the high resolution of the visual medium as a problem. He noted that it was difficult to convey fine quantitative information through the use of audio's main features such as pitch, volume and placement etc. This has important implications for the auralisation of sorting algorithms. The limited resolution of the auditory medium will limit the depth of information about the auralisations to be conveyed. It may not be possible to convey the exact state of algorithms but rather the general state. Combining features such as pitch, volume and placement might permit higher resolution. Extending the placement feature into a higher resolution form of display by employing 3D audio techniques may also resolve the resolution problem further.

Placement in the central visual field gives an angular resolution of about 2 seconds of arc difference. In the auditory field this difference falls to approximately 1 degree which again falls further to about 15 degrees resolution to the sides [190]. If pitch were used in the auditory domain, then in order for a user to understand absolute data he/she would need to have perfect pitch, which is a rare skill in human beings. This does depend upon the method of coding used to represent the data. Bregman [31] identifies a number of factors that contribute to perceiving, recognising and interpreting auditory stimuli. These factors are both perceptual and physical. Similarity and dissimilarity, proximity and good continuation are some perceptual factors. However, sound location, frequency, rhythm, scales, and keys are also examples of the physical contributing factors and are features that should be exploited if sound is to be used to represent complex information as in the context of this thesis.

In practice even most visually displayed data is presented with reference to something else, and high accuracy is not usually required. Data is usually constrained within certain limits, of which the user is also aware. Information must also have a corresponding range and context for it to be of use. So, data does not always have to be absolute to be useful. This is encouraging as some complex information that might not be presented well audibly due the limited resolution of the auditory medium might be conveyed at a higher level of abstraction. For example, if lists of numbers were represented using sound then the absolute values might not need to be understood in order to perceive the ordering of the list. However, the meaning of the data is usually grasped more quickly when presented visually especially if it is complex. This is because we can see the boundaries instantly and gauge roughly where the information of interest falls relative to this. It is interesting to note that in most visual tasks users do not recall object placement very accurately, and that real life experience rarely demands it.

Auditory communication is usually communicated serially, and this means that the instant appreciation of data that can be attained visually cannot easily be achieved audibly. It takes a much longer time for quantitative information to be conveyed via the auditory medium and may take several repetitions before the user can fully understand this data. The use of auditory space and parallel multi- timbral structures might speed up this data exchange, the auditory space being used to provide the range or context for the information. This is a prime area of concern in this thesis. Sending the data in a more parallel fashion with the use of more than one timbre could also reduce this serial limitation.

Similarly, speech interfaces suffer from the same problems as text displays [160] in that they both suffer from slow data transfer due to their inherent serial nature. In order for a user to understand a concept, the text must read or heard completely. Graphical displays certainly speed up certain interactions, and the comment "a picture is worth a thousand words" arose out if this property. This clearly discounts the use of speech to convey very complex information, as it would be limited to slow information exchange rates.

In some cases auditory stimuli can invoke meaning for a listener as effectively as pictures can invoke meaning for a viewer. Such meanings are often relative to the listener/viewer's personal experiences. Meaning is increased when pictures are recognised. The same is also true of sounds, that when heard often enough their meaning is increased. This highlights the ability of humans being able to learn mappings. This gives some flexibility when designing displays that use the auditory medium.

Continuous background sounds can be used to represent varying background information [33]. In this case, the auditory interface is not usually an intrusive medium as claimed by Berglund [17], who has suggested that sound at the computer interface is annoying to users. When played in the background, non-speech sounds can be listened to, or ignored at the listener's discretion. Although this thesis is not concerned with presenting background information, it is worth noting that some contextual information could be played in the background while more vital information is presented in the foreground.

It seems intuitive that the solution to the intrusive problems of audio at the interface is to use earphones, but it may be argued that this is unsatisfactory since users may also need to communicate aurally with each other. The alternative way to combat this problem is to adjust the volume level to just above that of ambient sound. In other words, just enough for the user to hear but not too loud as to annoy surrounding users. Another way of dealing with this problem is to use sounds which naturally occur in the working environment [49]. This thesis is more concerned with how useful the auditory medium might be for conveying information, but it is important to indicate such potential problems as intrusion.

2.3.3. Auditory displays

In information visualisation, mappings are made between information attributes and visual representations such as graphs, spectra etc. These kinds of mappings provide a framework within which users are able to construct mental images of the states and structures of the attributes of the information. Such visualisation works particularly well in cases where the information attributes naturally map into a spatial domain, such as hierarchical charts or sequenced events. Once the mapping has been made between the information and the visual representation, the user learns a framework within which he/she is able to visualise future information. This can be looked upon as a form of learning. When asked to draw the visual representation of new data they are automatically able to do so without necessarily having to revise the mappings.

In the same way, when data is presented in an auditory format it is usually necessary for the subject to first learn the mappings between the auditory cues and the information that is being represented. In some cases the mapping is obvious (for example the sound of a police siren). For other audio interfaces the mapping has to be learned. Once learned the subject will be able to model what further similar sounds represent and these may be quite abstract. It is therefore essential that users understand fully the framework within which the audio information is to be represented.

This can often be an intuitive process when real world information is to be used. However, the attributes of information in the computer science and data structure domains tend not to be real world objects and hence have few real world auditory signatures. Thus, non-intuitive mappings will need to be 'learnt' by the user in order to construct the necessary framework within which the information may be understood. Frameworks have been developed to aid this process [4].

Broadly speaking, auditory displays can be divided into two sub-sections, auditory interfaces[2] and auditory software visualisation[3] systems.

- Data sonification/audification – sonifying data input or data processed by the software. This involves mapping properties of data or events to sounds in an attempt to represent the data or events in the audio channel. *Audification* is similar but rather than using mappings, data are played back directly, e.g., scaling seismic data up until their values lie in the audible frequency range [166].

- Algorithm/program auralisation – mapping audio events to events during execution of the software itself. This is essentially 'sonifying' program progression and state.

Vickers [182] indicates that the terms "auralisation" and "sonification" are often used interchangeably. Vickers [182] states that sonification is concerned with the auditory display of generic data, whilst auralisation is more properly about the visualisation of programs and algorithms, which may involve the auditory display of data associated with or created by with a program, i.e. the data is concerned more usually with the internals of the program.

There has been much research in the field of Auditory Icons by many researchers, particularly Gaver [92, 93, 94, 95, 96, 97]. Auditory Icons are real world sounds that are used to represent an event or object. Where no real world sound exists a metaphorical

---

[2] Auditory interface, an interface that represent actions and objects using sound.

mapping is made and the listener learns this mapping. Gavers' work was based on theories by Vandeerveer [180] and Warren et al. [187]. Gaver has suggested that audio interfaces should focus upon tasks that require the user to monitor his/her environment [62]. Examples of this approach include monitoring complex systems, supporting computer access for the visually impaired [147], reading maps [19], sound-enhanced word processors for the blind [85] and debugging parallel programs [110, 87, 88]. Mountford et al. [146] stated, "*Sound can provide information about many different things within the environment.*" Many environmental monitoring systems have been developed, including a system for monitoring background file-sharing tasks entitled ShareMon by Cohen [53] and a collaborative working environment monitoring system called RAVE [97].

Other auditory interface applications include Mynatt's [139] *Mercator System,* a sound-enhanced graphical user interface for blind users. Colquhoun [55] developed a system that added simple sounds to a visual sonar monitoring system. Brown et al [44] performed similar work in that they carried out visual search experiments using auditory or visual target cues. Their implementation was performed in accordance with the multiple resource theory as described by Wickens et al [195]. Perrot et al [153] also found that giving auditory clues can help in locating visual targets on a display. They found that "*The presence of spatial information from the auditory channel can reduce the time required to locate and identify the location of a visual target...*" It is clear from their findings that auditory clues can help fix a region in a cluttered visual workspace. This is encouraging for the use of spatial audio in algorithm auralisations.

2.4. Music

2.4.1. Music as an interface medium

The following sub-sections detail some of the aspects of music and how they might relate to musical auditory display design. There has been relatively little work investigating the

---

[3] Auditory software visualisation, using sound to represent the execution or structure of software.

use of music at the human-computer interface. Initial research carried out by Alty [3] suggested there might be potential in using music to communicate information. He concluded that: "*Music involves highly sophisticated use of the auditory channel and there are some obvious reasons why the use of music should be considered for possible use in human-computer interface design.... we now need some good experimentation to determine what is possible and practicable*". Hotchkiss and Wampler [106] suggested that music lends itself well to experiencing data and events subjectively and would therefore give us a greater sense of participation than is possible when using more objective numerical representations.

Music provides a powerful medium which ought to be capable of delivering large amounts of data in parallel. Many musical styles employ such techniques as polyphony and counterpoint. The purpose of these techniques is to convey distinct ideas in parallel without confusing the listener.

Alty [3] has pointed out that: "*The information contained in a large scale musical work (say a symphony) is very large (a typical audio CD contains many hundreds of megabytes). The information is highly organised into complex structures and sub-structures. The potential therefore exists for using music to successfully transmit complex information to a user.*"

Schenker [167] proposed that perceived musical structure is represented internally in the form of hierarchies. Given that users hear the musical structures in a hierarchical manner it supports the concept of representing other hierarchical entities via music. Many data structures and information sources can be viewed as hierarchical and Brewster has employed hierarchical Earcons (discussed later in this chapter) to represent hierarchical menus. Dibben [73] represented music by abstractions which listeners were able to match with the original. This hierarchical structuring is similar to that used by many composers for coping with the short term limitations of the human memory [3]. Such techniques need to be adopted due to the temporal characteristics of music.

Alty [3] claims that: *"Music is all-pervasive in life and forms a large part of people's everyday lives. It is very memorable and durable. Most people are reasonably familiar with the language of music in their own culture. Once learned, tunes are difficult to forget."*

Music is prevalent in the everyday lives of most people. This suggests that benefits would be obtained from exploiting its properties. Most people can hum a tune after they have heard it once on the radio or television. Minsky [144] stated *"We like tunes because they have certain structural features"*. Memorised tunes can be exceedingly durable [182], and listeners often retain simple melodies long after they were first learned and committed to long-term memory. Modern dance music is particularly accessible because of the high repetition aspects of the melodies and their reliance on repeated cadences. However, listeners are more adept at recognising tunes that they experience in their everyday lives, and this raises an issue concerning the cultural differences between listeners. A subject from a modern western culture would be more comfortable when trying to understand a piece of modern western music as opposed to a subject from another culture. Differences such as scales and rhythm can often be quite diverse from culture to culture. However, some commonalities do exist that transcend many cultures such as timbre and pitch. When attempting to create a universal interface using music, HCI designers will need to take such differences into account.

Just as natural sounds play a common part in all of our lives so does music. A car horn sounding in the street is an environmental sound that most of us are used to, and the same is true of the sound of a musical instrument. More importantly, musical events do not have intuitive mappings to real world actions or objects, so it could become a complex task to convey a concept musically. It is here that metaphorical mappings are particularly useful.

Metaphors might be created by utilising the properties of music such as:

- Timbre,
- Rhythm,
- Pitch,
- Volume, and Stereo placement etc.

For example, the rising of pitch can be a metaphor for rising numbers. Different timbres could signify multiple channels active at the same time.

Without metaphor serious problems can arise. For example, what does a simple musical structure such as a triad (C-E-G) mean? Because there is no context in which this fits then it becomes very difficult to understand just exactly what it means. A framework is needed into which the mapping can fit. This provides the user with the necessary familiarity and allows him/her to place what they hear within a context, thus yielding the underlying message within the information. Kaye et al [115] suggested involving musicians and composers to help resolve some of these associated problems.

Alty [3] has commented that: "*Music involves the simultaneous transmission of a set of complex ideas related over time, within an established semantic framework. The job of a composer is to use musical resources and techniques to enable a listener to successfully disambiguate such information. There is therefore a strong parallel between the design requirements of the interface designer and those of a music composer.*"

Music composers often employ techniques such as polyphony and counterpoint to allow the simultaneous presentation of multiple melodies and musical ideas. The multiple melodies and musical ideas can be seen as different data streams. The same techniques might therefore be used when presenting real data via music i.e. it must be conveyed within a set semantic framework.

Brewster et al [38] combined Earcons (discussed later in this chapter) to deliver parallel information, each Earcon being related to an interface event. This showed how information can be conveyed simultaneously using complementary components. The prime difference between the visual interface and the musical medium is that the GUI is a spatial medium. Placement in visual space dictates how information is presented at the interface. The visual medium permits the user the peruse information at a lower pace and revisit ambiguous representations to further disambiguate the information. The musical medium however, is of a temporal nature and the disambiguation of information is dependent upon the time ordering. Information represented through this medium may only be revisited if the user has memorised past events. Stereo placement, however, can be used within the musical medium to allow spatial aspects to be presented.

Because of this, Walker and Scott [185] suggest that the visual modality is better suited to spatially oriented information whilst sound is more appropriate for processing temporal information. Wenzel [191] suggests that audio is very well suited to monitoring state changes over time. Thimbleby [179] reports that people working with early computers could, by placing an AM radio on top of the machine, tell from the changes in the radio interference when a particular batch run had finished or when the computer was in a loop.

In the 1980s the Musical Instrument Digital Interface (MIDI) was developed by Moog [145, 163, 164] and provided interface designers with a simple means for controlling peripheral electronic musical devices from a computer. It has now become very simple for programmers to incorporate MIDI commands into their code and exploit the use of musical sequences and notes to enhance the functionality of their programs. Further details of the MIDI standard can also be found [6, 186, 51, 136, 62, 27].

Music offers the following advantages. It can:

- Convey large amounts of information in parallel.
- Convey emotion on its own or supplemental to a visual display.
- Be used as a feedback mechanism for informing the user of success or failure of an action.
- Alert the user dramatically of critical situations or gently of trivial but relevant situations.
- Smooth over inconsistencies in the presentation of information.

Further benefits on the employment of music as a communications medium are:

- Most people are familiar with music of their culture. As Alty states, tunes can be hard to forget and most people are often readily exposed to some form of music throughout their daily lives [3].
- Alty indicates that there exists a strong parallel between the needs and requirements of an interface designer and those of a composer [3]. Both are trying to create representations using rules within semantic frameworks and also exploit the characteristics of human perception.
- MIDI makes incorporating music into computer interfaces easier [145].
- It provides a more important communications channel for the visually impaired [132].

However, the use of music in multimedia interfaces has some potential disadvantages.

- Music is more closely associated with conveying emotion rather than information and could therefore be considered to be inappropriate for interface design – There is some validity in this statement. However, it is the level at which music is used that is important. For example, visual interface designers work at a much lower level than that of painters and poets, and they do not create emotional interfaces, this is with exception of the games domain. The same may be possible in the case of music when not intended for conveying emotion but rather for conveying information at a lower level. Picard and Marrin's [133] study of emotional expression as it relates to musical performance showed that several forms of expressive communication can be measured and detected in physiological signals. These include the use of handedness to emphasize musical changes, the signaling of upcoming events with sudden changes in effort, the difference between information-bearing and non-information-bearing gestures, the indication of intensity and loudness with changes in muscular force, and the use of breathing to express phrasing in the music. In this case it has been shown that emotional expressiveness can be used to convey information pertaining to the musical performance.

- Music is culturally dependent which might limit its use on a wider basis – However, many musical scales are global and therefore possess universal appeal. Frequency ratios between notes are common amongst many cultures.

- Music cannot convey quantitative information - Most people can tell if a note increases or decreases, but only those with perfect pitch can determine actual pitch levels. Listeners with such capabilities are rare.

- It is an intrusive medium, it demands attention - Some users may find the information confusing if the level of stimulation is too high, particularly if audible from another user's interface. However, earphones do provide an alternative.

- Music is a language not understood by everyone – However, listeners do not have to be fully trained musicians in order for music to be a viable communication medium as shown by Brewster [33]. Most people can recollect and hum tunes with no real difficulty, thus supporting its appeal to a majority. Some visual tasks require brief training, so even if some basic grounding is required in music then it should be supported.

2.4.2. Perception and understanding of music

In order to create effective musical auralisations it is important to understand some of the perceptual factors associated with music. As with language, music has its own set of rules and structures and strong parallels can be drawn between the two. The relationship between music and language has been analysed by many scholars [16, 124, 198] and several similarities have been suggested.

- Language is capable of creating many complex combinations to convey many meanings. Music also possesses this ability.

- Music and language both possess distinctive structures which develop over time [198].

- Although music and language both have cultural dependencies, they possess universal features that can traverse many cultures making them flexible communication mechanisms.

- Human beings are able to comprehend both music and language.

- They both exploit the auditory channel by using sound patterns to convey meaning, Both are therefore forms of communication within distinct semantic frameworks.

Although these parallels highlight the similarities between music and language, some researchers such as Lerdahl and Jackendoff [126] and Kivy [118] have argued that musical meaning has little resemblance to natural language meaning, and contended that meaning in a musical sense cannot be translated into meaning in linguistic sense.

In the area of timbre[4] perception Grey [99] has performed experiments with trained musicians to determine the similarities of musical sounds produced by different instruments. The subjects were asked to rate the similarities on a scale ranging from 1 (very dissimilar) to 30 (very similar). Grey reported that the families identified with some of their sub-families were:

i. Family one. E-flat clarinet, soprano saxophone, bass clarinet , and English horn.
ii. Family two. Oboe and mute trombone.
iii. Family three. Bassoon, French horn, cello, trumpet, and flute.

However, the choice of instruments here is very limited. Rigas and Alty [158] carried out experiments to find which timbres and timbre classes work well as discriminating factors. They stated: "Our experiments suggest that one instrument from each of the following families is likely to be recalled by the listener with no prior training." [158]. Their study identified the following families of timbre classes:

Piano	Piano, harp, guitar, celesta, xylophone
Organ	Organ, harmonica
Wind	Trumpet, French horn, tuba, trombone, saxophone
Woodwind	Clarinet, English horn, pan-pipes, piccolo, oboe, bassoon, flute
Strings	Violin, cello, double bass
Drums	Drums

Figure 2.2a – Timbre classes from Rigas and Alty [158].

The consequence of this is that there are really only six unique timbres. One limitation of Rigas and Alty's findings is that the timbres were generated by a low-quality synthesiser (Roland MT-32). It is quite possible that a synthesiser with more faithful reproductions of musical instruments would yield a larger set of useful timbres.

Further experimentation by Alty and Rigas [159] showed that the most readily recognised groups of instruments were piano, woodwind and brass. They further recommend that "*...instruments such as Piano and Organ can easily be distinguished by non-musicians, but that designers should avoid the expectation that such users can distinguish timbre within musical families (e.g. a Cello and a Violin).*"

At this point it is worth noting that the perception of pitch is of interest as it forms the most basic component of tonal sequences. Other features such as intensity, placement and timbre etc. are not considered to be as important when perceiving music [29, 30, 67, 122, 47].

The perception of music is a complex issue. The individual notes are not listened to in a solitary manner but are listened to in relation to each other. Listeners group the notes. This grouping is applied to both the pitch and location of the music. However, the location of the source is less important than pitch [29, 30, 67, 122]. The reason for this is that localisation is significantly more ambiguous than pure pitch. It is subject to interference from echoes and certain sounds translate better than others. The human auditory system can also be tricked into hearing false sounds. One instance of this is when two identical sounds are played at mirrored locations either side of the head and the listener hears a single sound directly from the centre.

Butler [47] demonstrated that the pitch-grouping phenomenon is very robust even when real instrumental sounds are used with spatially separated loudspeakers. It was also reported that most of the listeners performed the grouping by pitch even when notes from

---

[4] Timbre, the distinctive character of a musical sound or voice apart from its pitch and intensity.

one speaker had a distinctive timbre. This suggests that pitch is more influential in grouping than both placement and timbre.

This grouping phenomenon has so far centred on single notes. In order to understand how we interpret music it is necessary to move up a level and look at how we perceive short combinations of musical notes, chiefly melodies. Dowling [79] describes a melody as a sequence of single pitches organised as an aesthetic musical whole. Deutsch [67, 68] states that contour, timbre, rhythm, intensity and tempo influence the perception of a melody. Dowling [78] performed experiments with real melodies constructed in such a way that two melodies were interleaved, note 1 of a tune A was played followed by note 1 of a tune B and so on alternately. The results of these experiments showed that it was practically impossible to recognise overlapping melodies because the melodies seemed to merge into one single unrecognisable sequence of notes. However, when the melodies did not overlap they were easily recognised. He concluded that overlapping melodies were only recognisable if the listener knew what to listen for. This suggests that there are instances where the listener is required to actively concentrate and scan for a particular melody when the overlapping causes some confusion as opposed to passively listening to the melodies in a non-overlapping instance. Even when users had learned an unfamiliar pattern of notes, it was found that they were not recognised when interleaved. This suggests that prior musical knowledge may not be of assistance to a user when interleaved pattern streaming is employed.

Investigation into melodic contour has been carried out to determine whether or not it assists in recognising tonal patterns [79]. Dowling and Fujitani hypothesised that if the interval size was altered but the contour remained the same then listeners would still recognise melodies. Users listened to two successive musical sequences. They were then asked to rate on a scale of 1 to 4 how close the tonal patterns were in relation to each other. The results indicated that users were making their decisions based upon whether or not the second melody was a transposed version of the first melody. In cases where the second melody was not a transposed version of the first melody, the subjects that had musical training showed to have no noticeable advantage over those with no or little musical training. This suggests that those with musical training are more able to recognise a transposed melody than those with little or no musical training.

Experiments by Davis et al [59, 60] showed that even though musical and non-musical subjects were not always able to estimate the interval sizes of well-known melodies, they were able to remember tunes. Davis and Yelland [61] found that remembering a set of tunes improves when participants are trained to work with certain melodies [61]. Bartlett and Dowling [13] found that when recognising tunes, the contour was of great significance.

Watkins [188] has shown that key signature plays a significant role in music. He conducted experiments that showed that participants from the western musical background were more capable of recognising melodies using pitch margins within the diatonic scale. Balzano and Liesch [12] argued that intervals in melodies are heard as positions within the scale and not as pure intervals. Dowling [80] also suggests that from a psychological viewpoint, listeners perceive a set of pitches as opposed to a set of intervals. This is further supported by investigations carried out by Dowling [78, 81] that indicated that listeners were able to recognise melodies even when their intervals were widened into different octaves whilst still maintaining the same pitches [78, 81]. This suggests that the representation is constructed in terms of pitches and not in intervals by the users.

Wolpert [200] argues that untrained musicians do not interpret musical stimuli in the same way that trained musicians interpret musical stimuli. He found that musicians and non-musicians follow different sets of rules when interpreting music. When matching excerpts, musicians used melody and correct harmonic accompaniment as the major criteria. Subjects that were termed non-musical did not use these same rules. Experiments carried out by Brewster [33] in which pitch recognition tasks were performed using earcons showed that musicians performed better than non-musicians. However, no differences were reported when earcons were played from instruments with different rhythms [33].

Deutsch [69, 70, 71, 72] conducted a series of experiments to determine the memorability of individual notes. In these experiments, listeners heard two notes separated by an interval. The pitch of the notes was the same for half of the trials and differed by a semitone for the other half. The subjects were asked to judge whether or not the notes had the same pitch. Deutsch reported that most of the listeners judged the pitch of the notes as 100 percent accurate. In another set of experiments, spoken numbers filled the interval. Deutsch found that this did not alter the accuracy of the subjects' judgement of pitch. However, when the interval was filled with a number of random notes it was found that the subjects' accuracy fell to 68 percent. Therefore, Deutsch argued that the intervening notes have a disruptive effect in recalling the pitch of an earlier note. This disruptive effect is even greater when the intervening notes are closer in pitch to the pitch of the earlier note.

However, Sloboda [169] remarks the following about Deutsch's experimental results:

"At first sight, Deutsch's results suggest a very gloomy conclusion about musical memory. Memory for individual pitches seems incredibly poor, if it cannot survive a few succeeding notes. How is it possible to remember notes across structures of symphonic proportions, containing tens of thousands of notes? The general answer to this problem would seem to lie in the opportunities, which most music affords for listeners to classify and organise what they hear. Deutsch's sequences were atypical in two respects. They did not confine themselves to the intervals of a common scale (using fractions of a semitone in some instances), and their notes were randomly chosen so that they were not designed to form common musical patterns within the scale framework"

Sloboda, here, suggests that Deutsch's experimental work does not closely follow the typical rules of music enough to be significantly related to the perception of music. The results reported by Deutsch relate more to standalone experiments on the perception of pitch outside of the musical framework.

In the tests carried out by Alty [3], the reaction to tasks that required interpretation of musical output was evaluated. The tasks were divided into two sub-tasks. In the first task, 12 male and 3 female subjects were asked to estimate the numerical difference between two notes. The first note was always 'Middle C' and the second note was always taken from the major scale above 'Middle C' (one octave). Sampling was limited to one octave because it is known that subjects have exhibited difficulties with intervals of greater than one octave [141]. The results for this first series of experiments showed that subjects exhibited 62% accuracy when estimating the difference in semitones between two musical pitches. The second test used the same subject group but listeners were presented with different musical shapes, each consisting of six notes from the major scale. The subjects were asked to sketch the shape they heard. The results showed that subjects were generally able to draw the perceived shape of these short musical sequences and that the ability to draw was also an important parameter in this context.

When a listener hears an extensive passage of music he/she segments the passage into smaller passages of musical sequences. These sequences are then memorised. This segmentation can be performed based upon properties such as timbre, rhythm, pitch or placement. Tan et al. [177] experimented with note sequences that contained two melodic phrases, each of which ended with a melodic cadence[5]. The melodies were played to subjects who were then asked to judge whether or not any particular two note probes were present in the melodies. There were three forms of the note probes. A pair of notes ending the first phrase, a pair of notes beginning the second phrase and a pair of notes `straddling' the phrase boundary. According to their findings, subjects recognised more of the first two types of probes as opposed to the probes 'straddling' the phrase boundary. This would suggest that humans are more adept at perceiving such probes when punctuated at the beginning or the end of a musical phrase.

Memorising the segmented passages is determined by the capacity of the human memory. Miller [143] suggested that this human short-term memory has a size of 7±2 chunks of information. Composers are very aware of these limitations of the human memory and

---

[5] Cadence, the close of a musical phrase. A fall in the pitch of the voice.

employ such techniques as patterning, repetition and structuring in order to reinforce musical sequences in the mind of the listener. Work carried out by Delis [65] provides evidence that people remember musical extracts best if they are labelled with concrete representation titles as opposed to abstract conceptual ones. These titles enable the construction of some sort of story in the human memory that in turn is associated with particular segments of music.

In psychoacoustics, experiments are performed using the scientific method of measuring the Dependent Variable (DV) and changing the Independent Variable (IV) of the musical stimuli. However, the musical stimuli may not satisfy the aesthetic qualities of music. Clarke best describes this concern [52]:

> *"There are certain obvious advantages in this very controlled kind of approach, and it has proved extremely powerful and productive for advancing our understanding of tonal and metric hierarchies. However, it has left untouched a range of issues concerned with listeners' understanding of more extended and elaborate structures in which a considerable degree of interaction between different parameters can be expected."*

Researchers have investigated how measurable variables that are apparent during exposure to continuous music are processed [169, 52, 48, 63]. Pollard-Gott [155] examined the possibility of participants focusing on particular musical themes when exposed to repetitions of the musical stimuli. Musical and non-musical listeners were asked to rate the similarity between two short musical passages. Results showed that the musically trained subjects perceived the similarities more quickly than the non-musically trained subjects. Other experiments with musically trained subjects have been performed showing that they can accurately judge excerpts from musical pieces [52]. This suggests that musically trained listeners have a distinct advantage over non-musical listeners.

2.4.3. Musical auditory display using Earcons.

This thesis proposes a musical auralisation of algorithm state and execution. Information available from algorithms can be in the form of both data and events. When representing events, it is possible to use structured metaphors as motifs much like earcons. An earcon is a brief succession of musical pitches structured to transmit specific items of information to the computer user. The term was first suggested by Buxton, Baecker and Arnott in 1985 [49] in terms of alarms and warning messages and was more formally defined by Blattner et al [18]. Brewster [36, 37, 38, 39, 40, 41] has since specified formal Earcon design principles and has carried out studies into their usefulness.

Earcons use tones in structured combinations to create auditory messages. Blattner et al define Earcons as "*non-verbal audio messages that are used in the computer/user interface to provide information to the user about some computer object, operation or interaction.*" The sounds and their respective mappings are learnt by the user. Unlike Gavers' auditory icons there is no intuitive link between the sound and what it represents. They are a much more musical approach in that the sounds are structured and formed in such a way as to produce a suggestion, much like a musical composition.

If it is possible for a user to learn the mappings between suggestive structured sounds and objects or actions then it must also be clearly feasible to use musical structures to achieve the same goal.

Earcons are basically constructed from motifs [174, 175, 176]:

> "*A motif is a brief succession of pitches arranged in such a way as to produce a tonal pattern sufficiently distinct to allow it to function as a single recognisable entity*".

Motives are simple building blocks of short, rhythmic sequences of pitches that can be combined in different ways. They are variable and may be used in conjunction with other motives, or repeated, to form larger patterns of a more complex nature.

The main features of motives are [174][33]:

- Rhythm - Blattner et al. suggest that this is the most prominent characteristic of a motive, it can be one of the most important characteristics of sound [66].
- Pitch - there are many different pitches in the western musical system. It is recommended that combinations be taken from one octave to produce different motives.
- Timbre - this is useful when differentiating between motives.
- Register - this is the position of the motive within the musical scale, duplicate motives in different registers (pitches) can be segregated and thus convey different meanings.
- Dynamics - the volume of the motive can be increased or decreased during playback of the routine.

In Earcons, the rhythm and pitch are fixed, whereas timbre, register and dynamics are variable. Blattner et al [18] describe two Earcon structures made up from motives:

- Compound Earcons - these represent actions and objects that comprise an interface. They are then combined in different ways to give information about any interactions within the interface.
- Family or Hierarchical Earcons - in this case each Earcon is a node on a tree and inherits the properties of the Earcons above it. There are only a maximum of five levels within the hierarchy due there being only five parameters of a motive.

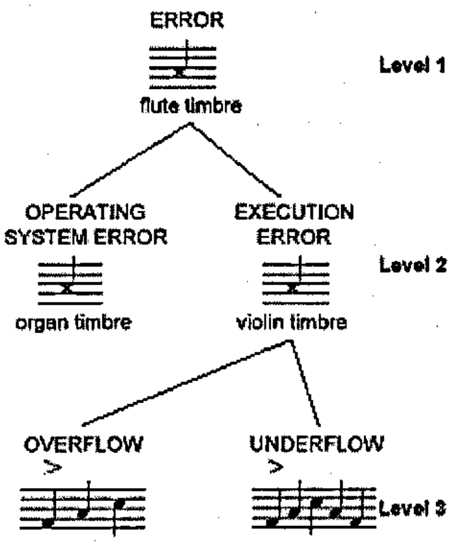


Figure 2.3 – Example Earcon hierarchy from Brewster [35].

An example of an Earcon hierarchy designed by Blattner [18] is shown in the Figure 2.3. This is the family tree of earcons representing various errors. The root motif is a structure comprising a single pitch (middle A) of indeterminate length. The two subclasses of error, (operating system and execution) inherit this structure but modify the timbre used in order to distinguish themselves from each other. Instances of these subclasses (e.g. overflow and underflow) inherit the timbre from their parent and are distinguished by melodic and rhythmic differences. Using design principles such as these, Earcons have been found to be effective in communicating hierarchical information down to four levels (for example, in telephone-based interfaces [37]).

Blattner also added these Earcons to two-dimensional maps [19]. Hierarchical Earcons were mapped onto the attributes associated with a building layout, in this case the Lawrence Livermore National Laboratory. They mapped sound to such information as the amount and type of computer equipment in each building, the security clearance required for each building and the jobs of the employees housed within each building. By selecting the buildings on the screen it was possible to hear these attributes. The technique allowed much more data to be presented than would have been possible graphically. No experimental testing was done with this system but it does indicate some of the possibilities within this field. Earcons have also been experimentally tested to see if users could extract algebraic information from sounds and identify their expressions [172]. The results indicated that the use of sound in this case proved to be of benefit as the number of correct interpretations of the information was better than chance. This again reinforces the advantages of using auditory techniques at the human-computer interface.

Earcons have also been employed in a menu hierarchy to aid navigation through its complex levels. This work undertaken by Barfield et al [11] did not fully exploit all of the features present within Earcons. The only feature employed was the use of pitch to indicate the current level of depth within the menu; the aim was that the user could link certain pitches with corresponding items within the menu structure. Barfield et al described the mapping as

> "*The tones were played with a harpsichord sound and started in the Fifth octave of E corresponding to the main or top level of the menu and descended through to B in the Fourth octave.*" [11]

The employment of Earcons in this system did reportedly improve the user performance in the task but no further suggestions were given. Due to the lack of detail in the reports that Barfield et al made, it is difficult to ascertain just how useful Earcons were in this case. Further improved results might have been gained if the other features of Earcons had been employed.

Brewster et al [42] similarly employed earcons in order to provide navigational cues in a menu hierarchy. The Earcons were based upon the guidelines earlier set out by Brewster et al [35]. Figure 2.4 shows the hierarchy of nodes used in Brewster's experimentation [35].

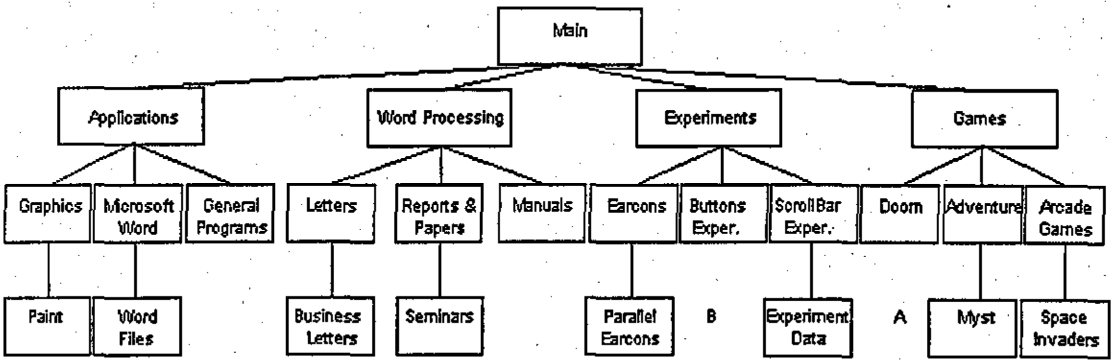


Figure 2.4 – Hierarchy of nodes from Brewster [35].

The menu hierarchy employed by Brewster et al has four distinct levels. In level one (main menu level), a constant note of 'D' in the third octave was played in the centre of the stereophonic field. In level two (application level), a second note was added to the existing note at level one. This second note was implemented with a different register and different stereophonic position. The submenus in the third level such as applications, word processing, experiments and games were assigned C4, C3, C2 and C1 using an electric synthesised organ, violin, drum and trumpet. These timbres were placed in the stereophonic positions of far left, centre left, centre right and far right, respectively. When the user descends through the levels, the timbre changes in accordance to the level whilst still maintaining the preceding level's note. Results from Brewster et al's experimentation indicated an accuracy of 81.5% in enabling listeners to identify their position within the hierarchical menu. This highlights the effectiveness of using multiple features such as timbre, pitch and position to convey information at the human-computer interface.

Brewster further showed that reductions in the quality of sound that occur with telephone systems can be offset by improvements in the design of earcons, thus making earcons a good method for providing navigation cues in telephone based intterfaces. Results showed that training techniques affected the recall rates of earcons and that there was no difference in the recall of earcons a week after their first presentation. Brewster states that the results obtained indicated that an online tutorial plus a short period of free call time can enable users to reach high recall rates without much training cost.

A further experiment by Brewster showed that by using compound earcons rather than hierarchical earcons to represent the hierarchy recall rates could be significantly increased, with 97% recalled correctly with only a small amount of training. This showed that compound earcons could represent a hierarchical structure well.

Earcons have been used in the interface to auralise data associated with turbulence [20]. Studies have shown that Earcons can be effective in communicating such information to users [41, 127], and Lucas [127] has shown that the accuracy of recognising the auditory cues increased when users were informed of the design principles of the Earcons employed This highlights the usefulness of context and the need for its presence when performing effective auralisations. Since its introduction, Brewster and others have used the Earcon to make GUI components (such as buttons, menus and scrollbars) more useable [43, 56, 125] and to reduce the length of audio messages by using *parallel Earcons* [38]. Experiments showed that the time taken to successfully operate such interface components was significantly reduced when the tasks were enhanced by the addition of Earcons. When Earcons were applied to drag and drop activities [43] a significant reduction in time taken and mental workload was similarly observed.

Brewster et al [34] reported that adding sound to a graphical interface could reduce task completion time and recovery time from errors. As with Barfield et al [11], the exploitation of the other features of Earcons could be employed to improve the effectiveness of the system. However, making the system more musical by using the rhythmic and multi-timbral features of Earcons could possibly have yielded more favourable results.

2.4.4. Musical data sonification

Several systems have been developed to allow the general sonification of data. Madhyastha and Reed's Personify system [130, 131] was capable of sonifying data sets. The system was used to explore multivariate data related to North-American cities. Variables such as population, climate, and housing cost were mapped to different sounds.

The resultant sonifications were used to compare cities. Madhyastha and Reed offered no formal or empirical evaluation of the system. Scaletti's Kyma [165] system uses a visual sound-specification language that also permits the sonification of data sets. The system was applied to models of the human arterial system and city air pollution. Scaletti and Craig claim the system was successful [166] but again no formal or empirical evaluation was offered to support their claim. Hayward [104] employed audification techniques to allow seismic data to be heard. The data from seismic recorders were collected and then scaled up so that the values lay in the audible frequency range. The data was played through an amplification system and it was possible to discern one seismic event from another without having to look at a seismogram plot. Again, Hayward failed to provide any formal or empirical evaluation of the resultant audification. Further work has been carried out by Dombois [76] in using audification in planetary seismology. No empirical evaluation was performed but Dombois did find that the signals were easy to recognise even in noisy environments and that same 'quakes' were heard differently when placed at different locations.

A DNA analysis program called *PC/*Gene was developed following the suggestion that the one-dimensional structure of DNA could be mapped onto musical sequences. This system utilises the Hayashi and Manakata algorithm [102] which facilitates the mapping of music to DNA triplets by assigning tones to DNA bases. *PC/Gene* can scan the sequence and identify such features as potential signature sequences, motifs, post-translational modification sites and membrane spanning regions in the protein. Hayashi and Manakaya's work on further gene sequences [103] required four octaves to map the necessary information. However, King and Angus [117], pointed out that this led to music with large intervals of pitch that were distracting and discordant. They further noted that the mapping itself was one-dimensional and therefore, led to a monodic[6] musical structure with no accompaniment. King and Angus developed their own system called PM (Protein Music) [117], which permitted the mapping of an amino-acid sequence onto a two-part harmonic musical structure. The mapping consisted of two parts, the melody line of the sonification which was mapped to the DNA nucleotide

---

[6] Monodic, having a single vocal part.

sequence, and the bass part which was mapped to the properties of the sequence's amino acids. King and Angus argued that to achieve an equivalent visual mapping (in terms of information presented) would "*be cumbersome as each position would have to be mapped into seven colours. This ability to display multivariate residue information represents an advance in this work*". No testing was carried to determine the effectiveness of the audio representations compared to visual representations but suggestions were made based upon the apparent complexity of mapping the complex information into the visual domain. This is a case where musical sonification can be seen to be more useful at presenting complex information over a visual representation.

Chaotic attractors have been used as a source of information in using music to sonify data. This has been carried out by Mayer-Kress et al. [138] who utilised key features of chaotic systems such as intermittency and self-similarity. By mapping the low-level sequence of system states onto auditory parameters, and high-level attributes (such as intermittency and self-similarity) onto polyphonic auditory constructs, they were able to utilise the data generated by a chaotic system by representing it musically. They reported that the music generated by the chaotic systems and their mappings was pleasing to listen to for two reasons. Firstly, the sonification possessed aesthetic qualities that recommended it as a piece of music one could actually listen to for enjoyment. Secondly, the development of the music over time yielded the underlying structure within the chaotic system. The property of self-similarity was represented by the form of musical phrases which repeated themselves but of which no two were ever exactly the same.

2.4.5. Musical software/algorithm auralisation

Alty [3] has shown through his experimentation that information about the run-time behaviour of simple sorting algorithms can be successfully communicated via musical mappings. Conveying precise quantitative information would be difficult in this manner unless the listeners are musically trained to a high enough degree that semitone changes could be easily identified. Conveying general shape pertaining to the state of the list to be sorted by the algorithm to non-musically trained users is however, a plausible scenario.

Alty showed from his early experimentation that musical experience can be useful in understanding algorithm auralisation but is not essential. His results further showed that an understanding of pitch is not essential for understanding auralisations.

Alty has shown that sorting algorithms can provide useful information sources to which musical features and structures can be successfully mapped. Given that many different sorting algorithms exist, which operate in varying manners and possess different sorting characteristics, there exists a rich source of qualitative and quantitative information to which music can be mapped. Such musical algorithm auralisation is the main focal point of this thesis, exploiting the diversity in sorting algorithms' characteristics to investigate the use of music in communicating different types of qualitative and quantitative information at the human-computer interface.

Vickers et al [181] developed a system called CAITLIN. This was a musical program auralisation tool used to assist novice programmers with debugging. CAITLIN is described as "*a pre-processor for Turbo Pascal programs that musically auralises programs with a view to assisting novice programmers with locating errors in their code*". In this case the aim of software visualisation is simply to improve the understanding of software [77]. Their auralisations were deliberately based upon musical techniques. This was presumably to exploit the features of music as much as possible.

The original source code is labelled with POI's (Points Of Interest) that in turn generate corresponding sounds. For example the POI's of an IF construct would be [181]:

- Entry to the IF construct.
- Evaluation of the conditional expression.
- Execution of the selected statement.
- Exit from the IF construct.

Instrumental sound sequences are mapped to each of these POI's and then played in real-time as the program code is executed. CAITLIN is a non-invasive system, in that it leaves the source code unchanged. Auralisations are affected by adding library routine calls to a copy of the program. This is then compiled to produce an auralised executable image.

Experiments were undertaken to evaluate the effectiveness of CAITLIN where subjects were permitted time to familiarise themselves with the auralisations of constructs via CAITLIN. They were then asked to identify the structures of nine program auralisations. The results proved favourable.

Program auralisation is beginning to attract much attention within the field of auditory displays. The case for using music to aid debugging (like CAITLIN) was also supported by Jackson et al [111], but they felt that this needed to be supplemented by a visual representation in order to provide a framework or context for the audio. This was because they found it easier to provide the boundaries and context visually and the relative information audibly as opposed to channelling all of the information into the same medium. It is an effective way of sourcing the control information through one modality and the informative data through another.

In early experimentation with the CAITLIN system Vickers et al played ten example auralisations to eight test subjects. Following this familiarisation exercise the test listeners were then presented with nine further auralisations. For each auralisation the subjects were asked to describe the structure of the program. The results from this early experimentation showed that the subjects were generally able to visualise the program structure using only the auralisation. It was found that most subjects specified exactly the program structure. Vickers et al also found that instrument selection played a very important role in successful program auralisation. Subjects commented that it was easy to deconstruct auralisations in the mind when the timbres used for the various constructs were markedly different. It was also found that as the complexity of the program constructs increased (in particular, when using nested constructs) the identification accuracy decreased. The same complexity issue was also seen when intricate signature tunes were used to identify the constructs. Another finding that is worthy of attention is that one of test subjects who scored the lowest claimed to have no familiarity with western music. This highlights the importance of cultural background in program auralisation. Vickers did not report on how the auralisation of program structure compared to the visualisation of the same program structure.

Vickers et al carried out further experimentation on twenty-two novice programmers to determine if CAITLIN could aid in bug location. They concluded that the case for general program auralisation remained unproven, but for programs of a relatively high complexity significant evidence was found for the auralisations having a beneficial effect. It was also found that programs with typical novice programming errors and programs that had high cyclical complexity[7] measures benefited from having the technique applied to them. It was also concluded from this series of experiments that musical knowledge had no effect on subjects' ability to make use of the auralisations. Additionally it was found that no evidence existed to suggest that lack of musical experience led to poorer performance.

Program structure auralisation auralises the execution state of a program. In a similar manner, algorithm auralisation auralises the execution state of an algorithm. Brown & Hershberger used music to enhance and complement an animation of several different algorithms [45]. They suggested that sound would be a *"powerful technique for communicating information about algorithms*", though some potential difficulties with using sound were highlighted:

- Sound is difficult to use effectively because of its complex cognitive-perceptual aspects. The perception and cognition of sound is not yet fully understood, it is therefore difficult to create effective auditory representations. Increased knowledge about the cognitive-perceptual aspects of sound would facilitate greater exploitation of the features of sound.

---

[7] Cyclic complexity, the level of complexity associated with the iterative properties of the structure of the program.

- To find the best mapping of data to sound characteristics is hard. Data can be mapped to frequency, amplitude, duration, timbre, stereo panorama, reverberation, attack and decay rates etc.

- When more than one computer is using audio techniques in the same room, isolation of the sounds is difficult, whereas graphical isolation is not.

Brown and Hershberger [45] animated six algorithms using newly developed techniques that employed the features of colour and sound. The information was therefore presented in a bi-modal format involving both video and audio simultaneously. It must be noted that the visual channel played the dominant role in these auralisations and the auditory medium was used merely as a supplement. A large content of their work focused upon the issues surrounding what kind of information to present and what was the best way to present it. As mentioned earlier there is no 'one-fit-all' solution to this problem as each case has independent contributory factors. Brown and Hershberger noted that previous work had been done in the area of algorithm visualisation (graphically only), but that no prior research had been carried out in the use of music to auralise algorithms (supplementary to the visual channel or autonomously). They also outline the fact that dynamic algorithm animation is still a very obscure art.

They developed a system called ZEUS [46] for algorithm animation and stated that *"it may be easy to animate a program, but it is not quite so simple to make informative animations."* This is certainly true. The actual task of mapping data to music is a relatively trivial one, but it can also result in a task to develop a system that produces meaningless, unstructured music. The mappings and timbres etc. must be carefully chosen in order to produce structured and informative musical representations of the data. Other animation systems include Stasko's Tango [171], Kahn and Saraswats' Pictoral Janus [114] and Heath and Etheridges' ParaGraph [105].

The ZEUS algorithm animation system [46] was developed from a previous algorithm animation system called BALSA [90]. The problems cited by Brown and Hershberger, which they hoped to solve using colour and sound were:

- Small screens - small real estate for presenting information.
- Lower resolution - less detail can be conveyed.
- Dynamic nature - flexible animation.
- Multiple views - views must work together to give an overall synergy.
- Multiple data sets - animation must handle a range of data values.

The principles used by Brown and Hershberger in the design of ZEUS were:

- Reinforce visual views - associate sounds with relevant program events corresponding to the visual representations. This technique uses the auditory presentation medium as a supplement to the visual one, its role is merely supportive and it is heavily dominated by the graphical display.

- Convey patterns - this technique picks out the temporal structures and paths within the data, to give the user a greater feel of the overall direction of the algorithm.

- Replace visual events - useful when a visual mapping becomes difficult, suggestive auditory representations of the data are employed in place of the visual ones.

- Signal exceptional conditions - an everyday and simple use of sound.

The system was used to animated:

- QuickSort algorithm.
- Multi- level adaptive hashing algorithm.

- Algorithm of Boolean formulas for simple polygons.
- Topological- sweepline algorithm.
- Spin / Block algorithm.
- Compliant - motion - planning algorithm.

They also refer to but do not fully document animation of the following algorithms:

- Insertion Sort algorithm.
- Bubble Sort algorithm.
- Selection Sort algorithm.

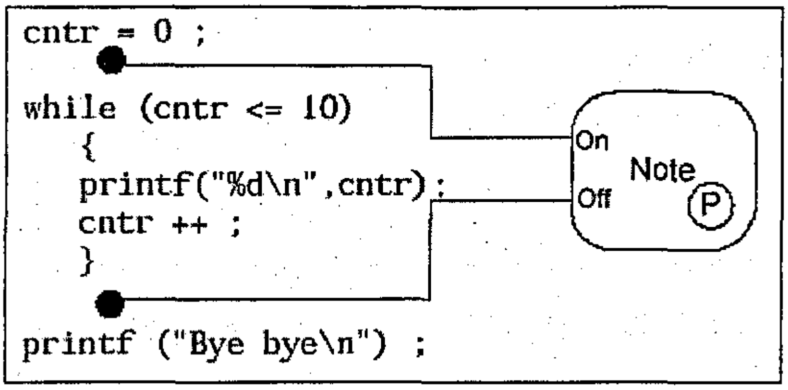
The work performed by Brown and Hershberger [45] yielded some guidelines for the implementation of algorithm animation. They concluded that the technique was a complex one and required further investigation. The animations that they performed were bi-modal. The auditory medium was used merely as a supplement to the visual animations, which highlights the need for research to be done in the area of pure autonomous auralisation of algorithms. The key point about Brown and Hershbergers' work is that no evaluation was carried out. The techniques that they proposed for algorithm animation were implemented but no evaluated results were attained. This further supports the need for thorough research and evaluated experiments in this field.

Bock has developed a specification language known as the Auditory Domain Specification Language (ADSL) [25, 26]. ADSL does not require sounds to be associated with specific lines of program code or specific variables. Users define '*tracks*' using the ADSL meta-language to associate audio cues with data and program constructs. A pre-processor interprets these user-defined tracks. The original code has the auralisations added to it during compilation time allowing the program to be heard upon execution. This approach makes program auralisation less complex. When adding lines of auralisation code within an editing window the original code can become obscure and difficult to follow. By specifying tracks, the original code is seen as untouched by the code editor and therefore remains easy to read. Another advantage of this approach is that it is possible to define a general-purpose auralisation. By specifying types of program construct to be auralised there is no requirement to tag individual lines of code with auralisation specifications.

Bock's ADSL used a mixture of MIDI messages, digitised recordings and synthesised speech. The desired output for each of the mappings is user specified and there exists the ability for the user to auralise specific data items as well as the general tracks through the program.

Bock's thesis [26] describes an experiment in which thirty programming-literate post-graduate engineering students were required to locate a variety of program bugs in three programs using only a pseudo-code representation of the program and the ADSL auditory output. Bock noted that 68% of the test subjects were able to locate the bugs. What Bock failed to provide was a measure of how successful bug detection was when no ADSL auditory output was present. Therefore, although the results were favourable, it is impossible to ascertain how useful the ADSL auditory output was on its own.

Jameson's Sonnet system [112, 113], like CAITLIN and ADSL, is specifically aimed at using auralisation agents to aid the debugging process. The code to be debugged is tagged with these auralisation agents that define how specific sections of code will sound. Figure 2.5 shows a simple *while* loop with the addition of a component that allows a note to be turned on and off. The component allows static definition of the attributes of the note such as amplitude and pitch. The first point in the source code identifies when the note will be activated and the second point in the source identifies when the note will be deactivated. Hence, upon execution, the note will be heard for the duration of the while loop.



```
cntr = 0 ;

while (cntr <= 10)
   {
   printf("%d\n",cntr);
   cntr ++ ;
   }

printf ("Bye bye\n") ;
```

Figure 2.5 – While loop from Jameson's Sonnet system [112, 113].

Clearly, in this example, the note only identifies the execution and duration of the while loop. It may be desirable to 'hear' the progression through the loop, in this case it would be necessary to auralise the data element 'cntr'. Sonnet permits the connection of other notes to specific data elements and also permits the connection of components to identify how many iterations of a loop are to be played. Given that the programmer has added the auditory components to his/her code then they have already learned what the specific sounds signify. Upon execution the programmer can hear the progression through the code and identify from the auditory cues where the program deviates from the expected execution path. When such a deviation is heard, the point at which the deviation occurred will yield the location of the bug within the program code. Like ADSL, Sonnet interfaces directly with the executing program and it is therefore non-invasive to the original code. It has the advantage of allowing auralisation in a visual programming environment that offers greater flexibility to the programmer.

Another system that shares a similar approach to Bock's ADSL is Mathur et al's *Listen* project [137, 23, 24]. Mathur developed a meta-language entitled 'Listen Specification Language' (LSL). This language is used by a programmer to write an auralisation specification that is parsed in a pre-processor phase to amend the original source code, again leaving the original source code unchanged. This entire process is carried out before the compilation phase. An auralisation specification specifies the mapping between program-domain events and auditory events.

The Listen Specification Language (LSL) is described as a true meta-language because it can be used to define auralisations for programs written in any language. Due to the complexity of the syntax of LSL, writing auralisation specifications requires a degree of programming understanding and ability. This limits its potential application to those with programming capabilities. Additionally, some musical knowledge is required in order to know how to specify which pitches are used in the auralisations. Again, the Listen project is a prime example of an auralisation system that never underwent any formal or empirical experimentation or evaluation.

DiGiano and Baecker performed work which involved sound enhancements to the programming environment [74]. Their system LogoMedia [75] dealt specifically with the use of sound within this programming environment. They defined program auralisation as "*the use of non-speech audio for supporting the understanding and effective use of computer programs.*" Their design suggestions for program auralisation are:

- to exploit the logarithmic nature of several sound dimensions to help to teach logarithms.
- to exploit the familiar connotations of everyday sounds.
- to use auralisation when screen sizes are too small to carry the communication visually.
- to use auralisation to reduce clutter in graphic workspaces.
- to exploit the many dimensions offered by sound (up to 20) [22].

LogoMedia allowed audio output to be associated with program events. A programmer annotates the original source code with probes to track control and data-flow. Upon execution of the program, the machine state and machine variables change over time. These states and data are mapped to auditory events (sounds) that can be listened to. The result is that the execution, progression and program state can be heard in real-time as the program runs. LogoMedia employs both sound effects and music.

The main features of LogoMedia are:

- During program execution, the relative values of variables are indicated and represented by tones and pitches.

- During program creation, opening and closing parenthesis are associated with a number of different tones.

- While viewing a program, special sounds are associated with particular segments of code.

LogoMedia is the auditory counterpart for the earlier LOGOMotion system [10], which was used to visualise program data and control. A specific language for use with LogoMedia and LOGOMotion is entitled LOGO [152]. As previously mentioned, a programmer may annotate their LOGO code with control probes and data probes. Control probes are mainly used for monitoring a program's control flow. Particular sections of LOGO software can be associated with particular program sections prior to execution. This results in the triggering of sound commands during execution. Data probes are used for monitoring data flow and can be associated with arbitrary LOGO software expressions. Changes to these expressions trigger sound commands during program execution resulting in the ability to hear the data flow of the program. As with some of the previously mentioned systems, the LogoMedia system is capable of producing both MIDI and recorded audio output. The main limitation is that the auralisations have to be defined by the programmer for each expression that is required to be monitored during execution. Upon entering an expression, the programmer is prompted for the desired mapping for that expression. This requires some strong programming ability from the user in order to successfully auralise program code.

2.4.6. Musical auditory applications – some existing applications

Rigas [4, 159, 160] developed a system called *AudioGraph,* which is a tool that allows information of a spatial nature to be communicated to visually impaired users using only music. AudioGraph permits visually impaired users to manipulate graphical objects on a screen by providing information about cursor position via musical tones. The outline shape of simple graphical objects is conveyed to users via a number of similar musical mappings. Rigas showed that users' ability to recognise a collection of graphical shapes was very high following minimal training. The recognition level further increased when hints about the objects represented by the shapes were given. This highlights the importance of context in metaphorical mappings.

AudioGraph is essentially a diagram-reader and manipulator aimed at users with visual impairments. It groups sequences of pitches into groups of ten rising notes with pauses between each group. Different instruments are used for the X and Y coordinates. Distance across the screen were interpreted by listening to the length of the rising tonal sequence, the longer the sequence the greater the distance. Subjects therefore obtained two clues about distance – the rising pitch and the note grouping. The rising pitch clue was particularly useful in the last group, which will normally be less than ten notes. This shows how quantitative information can be conveyed via tonal sequences.

Edwards [85] developed his Soundtrack system, which provides an auditory interface to aid users with visual impairments in using a word processor. The system adapted a mouse-based interface into one that employs audio techniques. Soundtrack used a combination of square waves of differing musical pitches and synthesised speech. When a menu was selected by a mouse click the interface 'spoke' the menu's name. The location of the mouse pointer relative to the menu item was conveyed by the pitch of a note, moving the mouse up and down the menu's options caused changes in the pitch of the note.

The main problem that users encountered with Soundtrack was recalling the layout of the internal structures of the windows. It was also found that most of the users did not use the pitch of the notes to determine the menu position but rather counted the number of tonal changes stressing the importance of multiple cues. However, one user who had musical training did use the pitches. Brewster [32] stated of these results "*It may have been that as there were only a few tones so counting was easier, if there had been more then counting would have become too slow and pitch perception would perhaps have been used.*"

Cohen and Ludwig [54] developed a prototype system called `audio windows' which combines a spatial sound output and gestural input in a teleconferencing system. The spatial sound system that they employed was based on projecting a sound into three-dimensional space. By manipulating the sound sources, virtual positions were achieved. Different listening sensations were achieved by user controllable parameters. The users moved their position around the conference room, and the sound sources would move relative to the users' virtual position. In this system, the user wears a DataGlove that feeds back gestural information to the system pertaining to the position, movement and shape of the user's hand. In order to capture the user's hand position and motion, Cohen and Ludwig employed posture recognition with the visual programming language (VPL) supplied Gesture-Editor and an arm interpretation component.

Cohen and Ludwig remarked, "*this prototype provides a test-bed for exploring the immediate potential of the emerging technology's application to teleconferencing and for `researching the relevant human factor issues'*".

Mansur et al [132] developed a system called SoundGraphs, which is an auditory interface primarily aimed at blind users, employing both speech and sound. For partially sighted users a visual display was provided. The system is bi-modal and not solely dedicated to the auditory medium alone. The system aims to permit the visually impaired user to create and view graphs. The shapes of the graphs can be conveyed in one of two ways, either as a whole continuing graph, or in an interactive manner for those areas of the graph which the user is most interested. In the interactive mode the user can control the output of sound by moving the cursor forwards and backwards. The system also incorporates a speech output whereby the coordinates of the graph are 'spoken' to the user. It is clear that the system uses several different forms to represent the graphical data, visually, tonally and orally. It is primarily aimed at aiding visually impaired users and is therefore not dedicated to determining whether an autonomous auditory channel can convey quantitative information.

Sonnenwald et al developed a system called InfoSound [170]. This system is an audio-interface tool kit that allows application developers to design and develop audio interfaces. It provides the facility to design musical sequences and everyday sounds. It also allows the storage of designed sounds and their associations to application events. One limitation is that the software developer is expected to compose the musical sequences. The research work of Sonnenwald et al is a continuation of previous work concerning the use of sound to represent numerical data and to provide cues about program events. It is also an extension of work carried out by Bly [22] to represent multivariate data using musical sound, Mezrich's [141] representation of multivariate time series using musical sound and Morrison and Lunney's [128, 129] representation of chemical spectra data for visually impaired users using musical chords. The InfoSound system offers a number of facilities and mechanisms for the design of musical auditory interfaces such as auditory icons and Earcons. It also facilitates the inclusion of everyday sounds which themselves can be associated with program events and be heard during program execution. The InfoSound toolkit is part of the IC* project [50] which is an environment for the design and development of sophisticated software systems such as telephone networks.

Camurri, Innocenti and Massucco [108] developed a software environment for the real-time processing of sound, music and multimedia entitled HARP (Hybrid Action Representation and Planning). HARP is a software architecture for the representation and real-time processing of sound, music and multimedia using artificial intelligence techniques. The HARP system is based upon the WinProcne system (WINdows PROlog tool Combining logic and semantic NEts) [89, 91, 109] and is capable of storing and processing music and sound as well as implementing data manipulation. The core architecture of the system comprises two levels of representation in the knowledge base of the system - the analogical and symbolic levels of representation. The symbolic level of representation has a declarative symbolic environment and a multiple inheritance semantic network, which has been based on KL-ONE [28, 201] with a number of additions such as temporal primitives and typing mechanisms. The analogical level is a low level sound representation that includes all associated data.

2.4.7. Audio - Visual mapping

Minsky [144] has stated that music facilitates the manipulation of space and time. If true, this indicates that it is possible to convey both temporal and spatial information via music. He also noted that *"We like tunes because they have certain structural features"* but do we have to like them in order to understand what they mean? Human beings, as listeners, have individual musical tastes and preferences. When listening to music that is not within these preferences, the listener may not be readily receptive. In this context, the underlying information might become obscured. Therefore if music is to be employed at the computer-human interface, it should be carefully chosen and structured so as to appeal to the user aesthetically as well as informatively. In order for a musical interface to be a truly global communications technique, it must fully exploit the features of music. In particular it must possess the flexibility to cater for many different musical tastes and cultures in order to appeal to a wide range of potential users. This can be accomplished by exploiting the features of music that transcend cultural differences such as tone intervals etc. At higher structured levels the presentations would need to be tailored to the classic structures inherent within the target culture.

When comparing the audio and visual senses Minsky [144] says that "*When we enter a room, we seem to see it all at once: we are not permitted this illusion when listening to a symphony...hearing has to thread a serial path through time, while sight embraces a space all at once.*" He also states that there are strong similarities between these two modalities by "*arguing that hearing music is like viewing scenery*" and that "*when we hear good music our minds react in very much the same way they do when we see things.*" Bearing these commonalities in mind it can be surmised that music might possess many of the qualities that pictures do. As Minsky said, we do not get an overall appreciation of the whole as we do with sight but there are certain similarities in the ways in which we react to the information.

To help outline the benefits of music as a communications medium it can be compared to the visual medium by identifying the parallels between their respective elementary building blocks. Alty [3] draws parallels between still images in the visual domain and music in the auditory domain:

- **Pitch** (frequency) - can be seen as the audio equivalent of **Colour.**
- **Timbre** (instrumental sound) - comparable to **Texture.**
- **Volume** (loudness) - analogous to visual **Brightness.**
- **Duration** ('on' period) - again comparable to visual **Brightness.**
- **Reverberation** (echo) – closest visual comparison might be **Focus..**
- **Location** (stereo placement) - seen as a visual coordinate on an **X-Y** plane.

In an undocumented informal test performed by Alty [3] that corresponded bounded numbers to bounded shades and tones, it was found that the majority of people thought that they would find it easier to place a random note than a random shade of colour (within the bounded regions). This is of course unsubstantiated as no actual testing was carried out, it was simply individuals views and thus based upon pure conjecture, but the users attitudes do in some way support the validity of an investigation of a musical communications medium.

The next stage building blocks are composed of the following [3].

- Chords - Harmony.
- Complex textures - Orchestration.

At this point music and the visual media diverge considerably [3]. Chords (harmony), rhythm (note repetition with different durations) and polyphony (playing simultaneous musical parts) are very difficult to map onto the visual medium because of the difference between their respective exploitations of time. That is to say that:

- Audio channel - copes with a number of simultaneous events in parallel and is a continuously moving sampler that cannot easily replay recent events without breeding confusion.
- Visual channel – for still images the focus is upon a small part of the display with awareness of peripheral events. The visual channel facilitates efficient scanning of recent history and allows the user to reflect upon it. With regard to moving pictures a temporal medium exists and such reflection and rescanning is not so easily facilitated.

When comparing the scanning of a painting to listening to a piece of music, a painting may be scanned time and time again which permits close examination, whereas a musical piece is listened to by the ear in real-time and does not easily facilitate this depth of examination. However, close examination is facilitated with repeated listening. This highlights a difference of temporal properties between visual stills and music. However, moving pictures possess a temporal dimension too which is often supplemented by an audio representation.

Composers employ complex techniques to reinforce a piece of music into the users memory. Pictures have been transformed into musical composition in an attempt to help better understand the pictorial data [140].

Given the lack of any extensive formal or empirical investigation into the use of music for communicating information and the strengths and attributes associated with the auditory medium and more specifically the strengths and attributes of music, there is clearly room for some detailed investigation.

2.4.8. Conclusions - Music in auditory systems

Within the field of human-computer interaction very little emphasis has been placed on the use of the auditory channel as a communication medium for computer generated events and entities. Interface designers have tended to concentrate on the development of graphical user interfaces, often using the auditory medium to convey trivial information as more of a novelty than of any real value. Although not an area of concern in this thesis, this direction of interface design has left visually impaired users at a disadvantage.

Over the last decade, visualisation researchers have found that aural representations can complement, enhance or sometimes be superior to visual representations alone. This has spawned the new research area of auditory display in which those involved are examining the different ways in which the auditory channel can be utilised in the process of human-computer interaction. Different techniques for using sound such as sonification and auralisation have been developed to exploit sound in various HCI applications. For example, audio enhanced software interfaces, sound-controlled data exploration systems or debuggers that use sound to represent program execution. Early systems tended to be hybrids employing aural and graphical visualisation methods. Many systems use sound effects.

Recent research has emerged in the field of human-computer interaction to support the potential of the auditory medium at the interface. In particular, there has been much interest into the use of music in this context. The evolution of such musical auralisation techniques as Earcons have begun to exploit the complex structured features of music. Designers are now becoming aware of the potential of music and its ability to communicate information to users. Music has been successfully used to aid visualisation of software structure and program execution as described previously in this chapter.

All the described musical auralisation systems use musical pitches and MIDI data in their approaches to auralisation. Although the term 'music' has been used to describe their auditory outputs, few of them employ any musical grammar or structured framework. Most of the systems that use music tend to do so only in the sense of representing data as musical pitches without any reference to musical forms, structures or syntax. Music is based upon a defined structure, or set of rules. Structuring auralisations according to simple syntactical rules offers the hope of music forming the basis for understanding algorithm and program execution. Recently, efforts have been made to use more formal musical frameworks in auditory display. Leplatre and Brewster [125] have begun investigations into using music to aid navigation around complex hierarchies of information. Hankinson and Edwards [100] have started to lay down a formal theoretical foundation for the use of musical grammars in audio communication applications. Furthermore, most systems require the programmer to compose the musical sequences, so the musicality of the output largely depends upon the programmer's musical ability. Such systems simply permit the mapping of program data and events to tonal outputs.

Very little formal or empirical evaluation has been carried out in this field. With the exception of Vickers's [181] research into understanding program state through musical software visualisation and Rigas's [4] research with the visually impaired, there is a lack of evidence that communicating algorithm (or program) information using music via the auditory medium is useful. Their experimentation using CAITLIN and Audiograph respectively has shown that listeners can use music well to understand and visualise entities such as program structure and graphical objects. Vickers also showed that there might exist some cultural issues that could be relevant, but not so as to invalidate the approach he took.

One of the key issues surrounding the use of music within algorithm visualisation is how to map domain entities to musical structures. Studies by Rigas have determined empirically that certain musical features can be shown to convey information well. The ability of music to convey temporal and spatial information in parallel data streams within a coherent structure and syntax indicates that it offers much potential as a means of communicating information at the human-computer interface. In order to help disambiguate auralised information, several techniques have been employed when performing musical mappings. These techniques have exploited the features of timbre, rhythm, volume, ascending pitch and spatial location of timbres. Given that previous experimentation in the field has shown that the approach of using musical auralisation to understand and visualise entities can be successful, it is feasible to extend the research by enhancing the auralisation techniques by designing a better spatial approach.

Many adequate tools exist for creating auditory interfaces. However, MIDI is limited for spatial audio. The spatial location employed by many of the systems described in this chapter has been trivial, only exploiting the use of stereophony in most cases. The limitations of the human perception accuracy of stereophonic placement have left it very much underused in the development of musical auditory systems and applications. Few auditory displays have employed three-dimensional timbre placement as a means of aiding the disambiguation of information. The exploitation of 3D spatial distribution as an extra information cue might improve users' understanding of musically auralised information, particularly program / algorithm state and execution. In the following chapter, this spatial distribution and the associated properties of the human perception of spatially located sound sources are investigated.

The main purpose of this thesis is to examine how relatively inexpensive 3-D sound techniques can be used to improve disambiguation of musically auralised sorting algorithms. This thesis is also concerned with the effect that musical training has on understanding such sorting algorithm auralisations. The emphasis on using 3D to enhance the spatial distribution of the auralisations has been born out of the limitations of stereophony and the aim of maximising the disambiguation of the presented information.

Chapter 3

Spatial Audio for HCI

Most modern computers provide some kind of audible "beep" to alert users, but today's more modern multimedia-equipped computers are capable of providing CD-quality stereo sound. Conventional stereophonic audio systems can easily place a sound at any position between the left and right loudspeakers (or earphones). However, with true 3-D sound, the source can be placed in any vectored location, at any height, distance or azimuth. In this section, the features of acoustics, sound localisation and several techniques for creating spatial audio environments will be investigated.

3.1. Basics of sound

The process of communication is one that involves pattern recognition and information processing. Living organisms perform these processes by interacting with their environment and other living organisms. Roederer [162] describes that external stimuli are processed through various stages in the sensory systems. In the case of the visual sense, Roederer points out that there are objects in space but in the auditory sense the objects are in time. A visual object has a physical presence in real space whereas due to the temporal nature of sound waves, an auditory object has a presence within time. It may be possible to use 3D audio to create a spatial aspect in a typically temporal medium, this extra feature might aid the disambiguation of information when musically auralising sorting algorithms.

Harris [101] reports that a human being is capable of detecting changes in frequency of about 3Hz for frequencies up to about 1000Hz. For frequencies between 1000Hz and 10,000Hz, the required frequency change for recognising a pitch difference can be specified as a constant. For example, at 10,000Hz, a 40Hz change is required in order for one to detect the change. This limits the amount of information that can be conveyed when using pitch as the resolution has limits based upon human perceptual factors. This might impose limits on the types of information that could be conveyed when auralising algorithm state and execution.

There are several characteristics that are key factors in the process of hearing:

- **Pitch** is the sound's perceived frequency. Low frequencies produce a low pitch and high frequencies produce a high pitch.
- **Loudness** is proportional to the amplitude of the sound.
- **Timbre** denotes the special set of characteristics associated with a particular creating instrument. For example, different instruments produce different timbres.

Intensity also has an effect on the perceived pitch of tones. Stevens [173] determined the effect of intensity on the pitch of tones for a number of frequencies between 150Hz and 12,000Hz. It was concluded that for frequencies above 3,000Hz, a constant pitch is maintained despite any increasing intensity. Below 3,000Hz the pitch of the tone is perceived subject to the intensity. In order to reduce this effect the intensity could be fixed fall all tones. In the context of this thesis, the intensity levels will only be adjusted as a matter of comfort.

When playing concurrent tones, if the two notes differ by more than ten percent then they become distinctly separate and the listener perceives them as two separate notes. The changes in loudness that result when two notes are separated by less than ten percent are known as beats. Risset [161] defines the difference between consonance and dissonance when more than one note is played at a given point in time: consonance occurs when a combination of tones produces a pleasant result and dissonance occurs when a combination of tones produces an unpleasant result. In the visual domain, a significantly strong visual stimulus prevents the perception of a weaker visual stimulus. This also translates to the auditory domain when one tone may mask another, though masking usually occurs when one tone is very intense and the other tone is very weak. To reduce complexity the proposed auralisations in this thesis could be performed sequentially. This means that no sound event would share its presence in time with any other sound event.

It is important to understand the process of hearing in order to create effective auralisations that exploit as many features of the process as possible. Wolf and Marsnik [199] describe listening as a complex process involving four elements:

- **Hearing.** This is the physiological process of receiving acoustic stimuli or signals. Hearing is fundamentally important in listening, because any form of listening requires a good hearing capability.

- **Attention.** The attention and the conscious awareness of the listener are required in order to attend to a certain message.

- **Understanding.** The interpretation and assignment of a meaning to the message or signal received and attended.

- **Remembering.** The process of storing the acoustical information received for later retrieval. It involves two types of memory, the short-term memory (STM) and the long-term memory (LTM).

Wolff and Marsnik [199] further describe that there are a number of axioms for listening:

- Listening is a mental operation.

- Listening is active. It involves several intellectual operations. A person needs to be alert when listening.

- Listening is learned. It can be learned and it improves with training.

- Listening is complex. It involves, as noted above, hearing, attention, understanding and remembering.

- Perceptive listeners must be trained. No matter how much a person wishes to listen, they can only do so to the level to which they have been trained.

- Listeners share responsibility for communication success. Listeners have to exercise their minds and sometimes do some fast mental manoeuvring to understand a message (this applies particularly to conversation).

- Listening is as vital a communication skill as reading.

- Listening is crucial to all communication. Listening is a major part of verbal communication and without it verbal communication itself cannot exist.

3.2. Spatial hearing and psychoacoustics

The ability of the auditory system to localise sound sources is just one component of our perceptual systems, it also has a high survival value. Living organisms have found many ways to extract directional information from sound and often use them in a self-preserving capacity. Although there are some unknowns concerning the perception of acoustic sources, the major cues have been known for a long time. Many psychological studies have established how accurately we can localise acoustic sources. In order to generate spatial sound for HCI and create effective algorithm auralisations it is important to understand that which influences the human auditory system.

3.2.1. Vectored Coordinate Systems

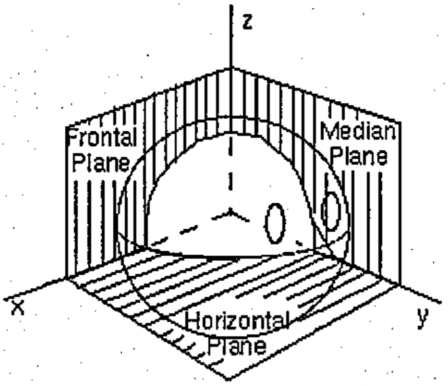


Figure 3.5 – Head-centred rectangular-coordinate system.

In order to specify the location of a sound source relative to the listener, it is necessary to utilise a coordinate system. Duda [83] noted that one natural choice is the head-centred rectangular-coordinate system shown in Figure 3.5. Here the x-axis goes (approximately) through the right ear, the y-axis points straight ahead, and the z-axis is vertical. This defines three standard planes, the x-y or horizontal plane, the x-z or frontal plane, and the y-z or median plane. The horizontal plane defines up/down separation, the frontal plane defines front/back separation, and the median plane defines right/left separation. Due to the approximate spherical shape of the human head, a spherical coordinate system is usually favoured. The standard coordinates used in a spherical coordinate system are azimuth, elevation and range. Duda [83] points out that there are two common ways to define these coordinates, the 'vertical-polar coordinate system' and the 'inter-aural-polar coordinate system'. The vertical-polar coordinate system, shown below on the left, is the most common spherical coordinate system. With this system the azimuth θ is first measured as the angle from the median plane to a frontal plane passing through both the source and the z-axis. Secondly the elevation ϕ is measured as the angle up from the horizontal plane. With this choice, surfaces of constant azimuth are planes through the z-axis, and surfaces of constant elevation are cones concentric about the z-axis.

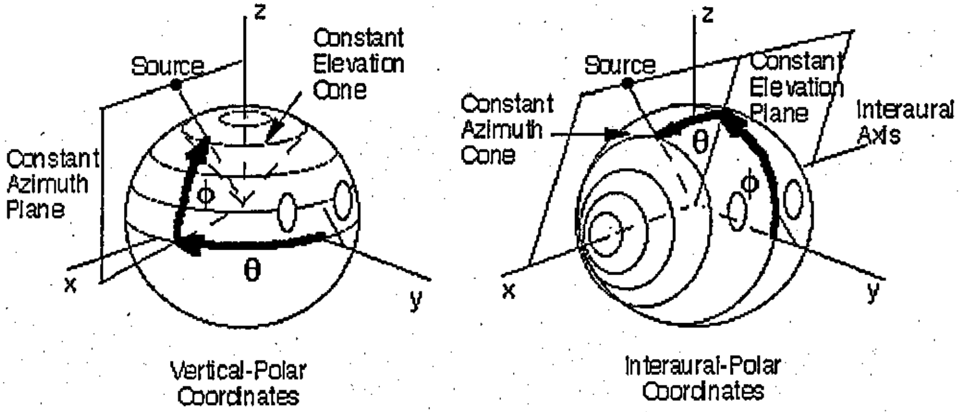


Figure 3.6 – Vector-polar and Inter-aural-polar coordinate system.

The second of the spherical coordinate systems is the inter-aural-polar coordinate system, shown on the right in Figure 3.6. With this system the elevation is first measured as the angle ϕ from the horizontal plane to a plane that passes through the source and the x-axis, which is the inter-aural axis. Secondly the azimuth is measured as the angle θ over from the median plane. With this choice, surfaces of constant elevation are planes through the inter-aural axis, and surfaces of constant azimuth are cones concentric with the inter-aural axis.

3.2.2. Azimuth cues

Lord Rayleigh (John Strutt) was one of the pioneers in spatial hearing research and approximately one hundred years ago he developed his Duplex Theory [157]. According to this theory, there are two primary cues for azimuth; Inter-aural Time Delay (ITD) and Inter-aural Intensity Difference (IID) (Figure 3.7).

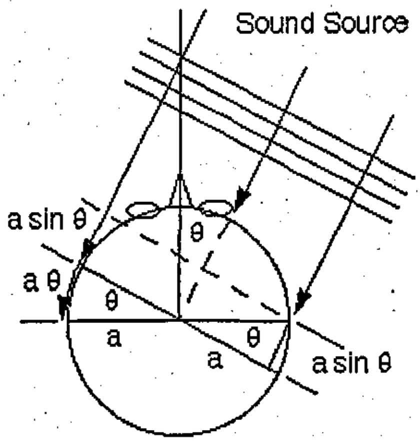


Figure 3.7 – ITD and IID diagram.

Lord Rayleigh had a simple explanation for the ITD. Sound travels at a speed of approximately 343 m/s, say ‘c’. Consider a sound wave from a distant source that strikes a spherical head of radius ‘a’ from a direction specified by the azimuth angle θ. Clearly, the sound arrives at the right ear before it arrives at the left ear, since it has to travel the extra distance $a\theta + a\sin\theta$ to reach the left ear. Dividing that by the speed of sound, it is possible to obtain the following simple formula for the inter-aural time delay:

$$ITD = \frac{a}{c}(\theta + \sin\theta) \quad , -90^\circ \leq \theta \leq +90^\circ$$

Furthermore, Lord Rayleigh also observed that the head diffracts sound waves. He solved the wave equation to show how a rigid sphere diffracts a plane wave. His solution showed that in addition to the inter-aural time delay there also existed a significant difference between the signal levels at each ear, this is now termed as the inter-aural intensity difference (IID).

IID is highly frequency dependent, at low frequencies, where the wavelength of the sound is long relative to the head diameter, there is hardly any difference in sound pressure at the two ears. However, at high frequencies, where the wavelength is short, there may well be a 20dB or greater difference. This is known as the head-shadow effect, where the far ear is in the sound shadow of the head.

Rayleigh's Duplex Theory [157] states that the IID and the ITD are complementary. At low frequencies (below 1.5 kHz), there is little IID information, but the ITD shifts the waveform a fraction of a cycle, which is easily detected. At high frequencies (above 1.5 kHz), there is ambiguity in the ITD, since there are several cycles of shift, but the IID resolves this directional ambiguity. Rayleigh's Duplex Theory says that the IID and ITD taken together provide localization information throughout the audible frequency range.

3.2.3. Elevation cues

The primary cues for azimuth are largely binaural whereas the primary cues for elevation are often considered to be monaural. The outer ear or pinna can be seen as a directionally dependent filter. It can amplify some frequencies through its resonant cavities while at the same time attenuating other frequencies due to the interference effects caused by other geometrical attributes.

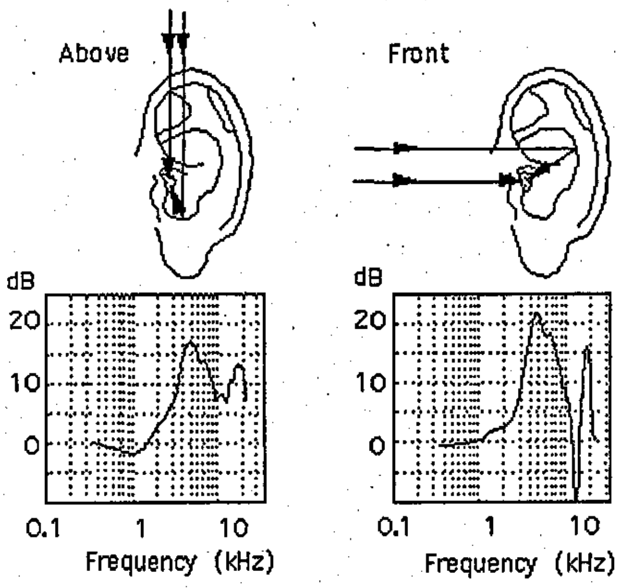


Figure 3.8 – Direction dependent frequency response from Duda [83].

Duda [83] illustrated, in Figure 3.8, measured frequency responses for two different directions of arrival. In both cases it can be seen that there are two paths from the source to the ear canal; a direct path and a longer path following a reflection from the pinna.

Duda further explains that at moderately low frequencies, the pinna essentially collects additional sound energy, and the signals from the two paths arrive in phase.

However, at high frequencies, the delayed signal is out of phase with the direct signal, and destructive interference occurs. The greatest interference occurs when the difference in path length d is a half wavelength, i.e., when f = c / 2d. In the example shown, this produces a "pinna notch" around 10 kHz. With typical values for d, the notch frequency is usually in the 6kHz to 16kHz range.

The pinna notch is noticeably larger when the sound source is in front of the listener, this is because the pinna is a less effective reflector of sounds that come from above than for sounds that come from the front. Furthermore, the length of the sound path changes with the elevation angle, this results in shifting the frequency of the pinna notch. Therefore, both the size and frequency of the notch are dependent upon the elevation.

3.2.4. Range cues

Estimating range is the most difficult element of localising sound sources in a spatial environment. As humans we are typically best at estimating azimuth, next best at estimating elevation and worst at estimating range. The cues for each of these localisation attributes are understood in the same order, we know most about azimuth cues and least about range cues. There are, however, several cues pertaining to range.

As a sound source gets closer to a human head, the inter-aural intensity difference will increase. This increase in difference is particularly noticeable for ranges under one meter. Therefore it can be used as a cue for estimating range, distant sounds have very little IID whereas close sounds have a relatively large IID.

Motion parallax refers to the fact that if a listener translates his or her head, the change in azimuth will be range dependent. For sources that are very distant, a small shift causes very little change in azimuth, whereas for a close sound source, a small shift causes a relatively large change in azimuth. This feature further assists in the human ability to estimate range based upon geometry, this feature does, of course, require the listener to actively move in order to identify the change in azimuth.

Another useful cue when estimating range is that of the ratio of direct to reverberant sound. In a normal room the surface reflected reverberant energy of a sound source does not differ much from the sound source itself when it reaches the listener. It is also known that the energy received directly from a sound source drops off inversely with the square of the range. Given these two characteristics, a comparison can be made between direct energy and reverberant energy where the variable is dictated by moving the sound source. At close ranges, the ratio is very large, while at long ranges it is quite small.

As previously mentioned the energy received directly from a sound source drops off inversely with the square of the range. Therefore, as a constant-energy source approaches a listener, the loudness will increase. There is no direct one-to-one relationship between the received energy and the energy emitted by the source, this is because the relationship is dependent upon the loudness of the signal source. When we estimate range using this cue we are more successful when we understand the context of the sound source. Changing the volume of the sound source alone does not give the impression of a change in range, therefore it is necessary to understand the source of the signal in order to create the relationship between the source and received energy.

3.2.5. Echoes and reverberation

As humans we are largely unaware of the quantity of energy that is reflected from surfaces, we are not conscious of such echoes unless they become extremely delayed and apparent. In a normal room it is obvious that sound waves are reflected from surfaces such as walls, objects, ceilings etc. Reflections also abound in the outdoor environment, they are reflections from the ground, vegetation and objects. The reflections that we are conscious of are those that exceed the echo threshold of approximately 30 to 50 ms, sounds that have a delay of less than this threshold are not easily determined as echoes but are nonetheless subconsciously used to aid localisation. Special rooms called anechoic chambers are built to absorb sound energy, so that only the directly radiated energy reaches the ears. Such chambers essentially suppress echoes by absorbing them into specially formed surfaces that line the chamber.

When creating a virtual 3D environment it is important to include such echoes in the acoustic sounds that reach the ears, they are essential in localisation and playing pure sounds with no reflections would be similar to that of recording in an anechoic chamber. We are, of course, not used to localising pure unreflected sounds and it would be alien for us to estimate sound source positions in a non-reverberant environment. Upon entering an anechoic chamber for the first time, most people are astonished by how much softer and duller everything sounds.

As previously mentioned, reflected sound is very common in ordinary acoustic environments. Such reflections do not interfere with our ability to localize sources because we quickly adapt to a new environment, and our auditory system uses only partially understood mechanisms to suppress the effects of reflections and reverberation. The fact that we localize sounds on the basis of the signals that reach our ears first is known as the 'precedence effect' or the 'Law of the First Wavefront' [186]. We are also aware of the reflections that follow, we subconsciously use them to estimate range.

In a typical room, reflections begin to arrive a few milliseconds after the initial sound. If the initial sound is low frequency (below 250 Hz) and hence has a period that is longer than that of the reflections, then the reflections arrive before the first wavefront (initial direct sound). After several cycles the sound pattern that reaches the ears becomes complicated and mixed, in this case it becomes almost impossible for the listener to localise the sound source. In summary, this means that low-frequency information is rendered useless for localising sound sources in a reverberant environment.

Clearly, the inter-aural time delay is important when localising sound sources although it is severely impaired at low frequencies. However, that does not mean that inter-aural timing differences are unimportant. Other important timing information may be utilised, in particular the inter-aural envelope difference (IED). The IED refers to the difference between the transients at the onset of the emitted acoustic signals.

If a sine wave is filtered into two channels, one low-pass filtered and the other high-pass filtered, and the two channels are played through two loudspeakers placed in different locations, then a listener would usually estimate the sound source to be emitting from the high-passed channel. This is commonly known as the Franssen Effect. Basically, the starting transient provides unambiguous localization information, while the steady-state signal (low-pass filtered signal providing most of the energy) is very difficult to localize, and in this circumstance the auditory system simply ignores the ambiguous information.

3.3. Basic spatial audio systems

Most people are aware of the possibilities of simple single plane spatial audio systems. Such single plane spatial sound systems are primarily concerned with placing sound sources at a fixed height and distance but varying the azimuth. There are two basic classes of such systems, Stereophonic systems (Two-channel systems) and Surround sound systems (Multichannel systems).

3.3.1 Stereophonic systems (Two-Channel systems)

The concept of stereophony is a simple one and was the first successful commercial attempt at spatial sound reproduction. The concept of stereophony is to utilise two loudspeakers to produce two separate streams of audio, to produce a sound in the left ear simply apply the sound to the left channel and vice versa for the right ear. If the sound is equally applied to both channels then the resulting output is that of a perceived sound source between the two loudspeakers. It is important to maintain that the two loudspeakers are in phase (pushing together and pulling together) otherwise the effect is that of cancellation. Theoretically, if both channels, both signal sources and both loudspeakers are identical and the listener is sat directly between them in the centre of a symmetrical room then the listener would hear nothing when the two channels are playing sound sources in exact anti-phase. The signal sources would effectively cancel each other out.

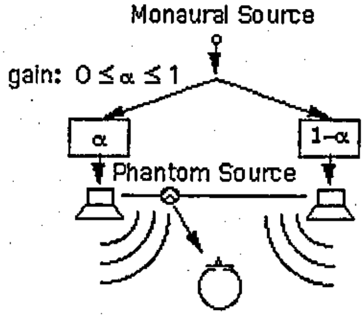


Figure 3.9 – The stereophonic 'Phantom' sound source from Duda [83].

The perceived sound source shown in Figure 3.9 is often termed a "phantom source" and will appear to originate from a point midway between the two loudspeakers when the channels are equally applied. By "crossfading" the signal from one speaker to the other, one can create the impression of the source moving continuously between the two loudspeaker positions. However, simple crossfading will never create the impression of a source outside of the line segment between the two speakers, the very physical set up and simplicity of the system will not permit this.

In a simple system such as this, another technique may be used instead of crossfading to give the perception of a sound source somewhere between the two loudspeakers. This is simply achieved by delaying the sound source to one of the speakers by a fraction of time. What this technique essentially does is to exploit the 'precedence effect' or the 'law of the first wavefront'. If the sound on the left is delayed by 10 or 15 ms relative to the sound on the right, the listener will localize the sound on the right side. This applies when the sound sources are of equal amplitude and still applies even if the sound that comes from one loudspeaker is louder than that of the other. If the delay is too excessive then the effect becomes disturbed and the listener hears the delayed signal as a mere echo. Stereophony facilitates the placement of sound sources along a line between the listener's ears. For the purposes of spatialising algorithm events it provides adequate separation. However, for spatialising more complex information like algorithm data it is limited in terms of resolution.

3.3.2. Surround sound systems (Multichannel systems)

An extension of stereophony is the obvious progressive step to utilise more than two loudspeakers, this technique essentially employs a different loudspeaker (and channel) for every desired direction (Figure 3.10). This is the same type of system that is employed within surround sound cinemas such as Dolby Pro Logic Surround Sound [82]. In a typically reverberant environment this type of system exploits the Franssen effect. Small loudspeakers are placed at many locations except for one large speaker (subwoofer) that provides the nondirectional, low-frequency content. The signals to the small speakers are then complexly filtered to place the sound in the desired location.

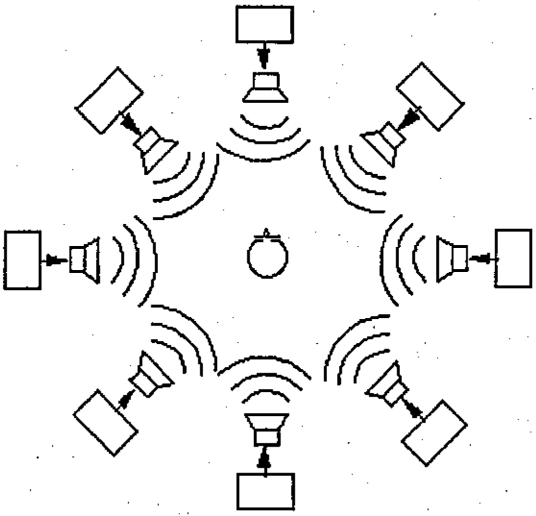


Figure 3.10 – Multi speaker system.

These types of systems clearly provide some amazing spatial sound effects but are often very complex and costly to implement. They have been adopted by many commercial cinemas and makers of home entertainment systems and have clearly made their mark in the field of spatial audio. Surround sound systems can provide effective sound spatialisation. This thesis is more concerned with using inexpensive 3D audio techniques and surround sound systems can be costly and complex.

3.4 Binaural recording

The concept behind binaural recording is a very simple one, it involves simply recreating the same sound pressure levels at each eardrum that would be present if the listener were actually in the sound field. This technique only requires the use of conventional stereophonic equipment as two channels (left ear and right ear) are all that is required. The results of binaural recording can produce vivid 3D representations of sound sources.

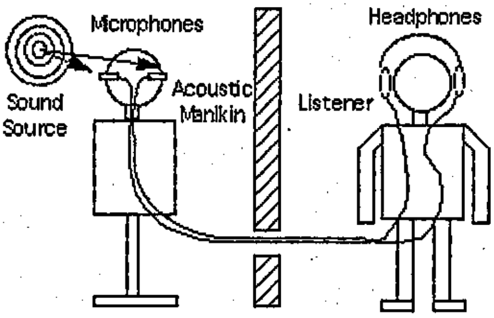


Figure 3.11 – Binaural technique.

In order to recreate the same sound pressure levels it is important to take into account the filtration effects that naturally occur when we hear sounds. As previously mentioned, sound localisation is primarily determined by the ITD, IID, reverberation and filtration effects (reflection and absorption) of the pinnae. Duda [83] explained that in order to exploit these effects a conceptually simple approach is to put two microphones in the ear canals of an acoustic manikin (or human being) and record what they pick up (as shown in Figure 3.11), the resulting recordings will have already been subjected to the effects of the environment and manikin. When the recorded left and right signals are fed to the left and right earphones respectively, the effect is that of the listener being present in the original sound field.

The immediate problem that arises is that of the geometry of the manikin. If the manikin and the listener have heads with the same size and shape, the same ITD and IID information will be present; similarly, if the manikin and the listener have pinnae with the same sizes and shapes, the same elevation cues will be present. If, however, the geometrical differences between that of the listener and the manikin are significant, the resulting perceptual 3D sound environment becomes subject to errors and localisation is difficult.

The obvious way to reproduce a more precise individual listening experience is for the listener to also be the manikin, this way the geometrical similarities of the recording head and the listening head are as close as physically possible. This does, of course, mean that each individual listener must have his/her own unique set of binaural recordings to maximise the desired effect.

The technique of binaural recording is one of obvious economy and effect, there are , however, some disadvantages to using the binaural technique:

- They require the use of headphones – due to the sounds being recorded within the ear canal it is important to reproduce at the same place, this limits this technique to earphones.

- They are not interactive, but must be pre-recorded – a pure sound source cannot be manipulated by binaural recording to produce a 3D sound, what is heard by the listener must be pre-recorded and is therefore fixed in the initial recorded location.

- If the listener moves, so do the sounds – as previously mentioned, because the sounds need to be pre-recorded and are therefore fixed in the initial recorded positions. They are not altered when the listener moves his/her head during playback. For a truly interactive immersive environment the sounds should give the same perceived fixed location even when the listener moves his/her head. This would require the addition of some head tracking equipment and further real-time filtration.

- Sources that are directly in front usually seem to be much too close – this is a known problem with binaural recording, frontal range does not seem to translate well using the binaural technique.

- Because pinna shapes differ from person to person, elevation effects are not reliable – pinnae are unique to each individual, even each of the pinnae on a single listener will always be different. Because of this individuality, binaural recordings do not translate well from user to user (particularly the pinna dependent elevation cues). The most pragmatic approach is to use a 'Mr. Average' head for recording, this way ensures maximum possible effects over a broad range of listeners. Some degradation of localisation, however, is inevitable.

A binaural recording can be improved if we overcome some of the issues mentioned above in order to make the listening experience more effective. The next step is the use of head-related transfer functions (HRTFs). Binaural recording does offer an effective and relatively inexpensive solution to the spatialisation of musically auralised sorting algorithms.

3.5 Headphones vs. Loudspeakers

It was stated earlier that the concept behind binaural recording is a very simple one. It involves simply recreating the same sound pressure levels at each eardrum that would be present if the listener were actually in the sound field. If the recording is made in the ear canal then it stands to reason that when the binaural recording is played back it should be played in exactly (or as close to as possible) the original position of the recording transducer. In order to fulfil this requirement it is necessary to employ earphones to produce the sounds at the ear canal. Earphones certainly simplify the problem of delivering one sound to one ear and another sound to another ear, however, earphones do have certain problems:

- Earphones often have filtration characteristics such as notches and peaks in their frequency responses, such characteristics often resemble pinna responses. In order to circumvent this problem it is important to use compensated earphones, if uncompensated earphones are used then elevation effects can become significantly augmented.

- Earphones are often uncomfortable and headphones can be particularly cumbersome. The better the acoustic quality of the headphones the larger and heavier they often are, this can cause them to be very uncomfortable for a listener to wear for any lengthy period of time.

- One of the most commonly noted characteristics of earphones is that when sounds are played back they often appear much closer than they were in the original recording. This can, of course, be compensated through some filtering.

Given that earphones have these disadvantages it is worthwhile looking for an alternative. Loudspeakers do not suffer from most of the problems associated with earphones and are therefore worth considering as a viable alternative. The immediate question that arises is how to successfully deliver binaural recordings over loudspeakers, the importance of reproducing the recorded sounds at the ear canals has already been highlighted.

A method of replicating the production of sound in the ear canals using loudspeakers needs to be investigated. In order to successfully implement such a system it is important to note two key issues. Firstly, the sound produced at the loudspeakers must be filtered to replicate how it would sound to the user when playback is via output devices placed within the ear canals. Secondly, some co-channel cross-talk will be present due to both channels having to traverse the same medium of air (Figure 3.12). On way of addressing the latter issue is to employ a technique known as cross-talk-cancellation. Cross-talk-cancelled stereo is also known as trans-aural stereo.

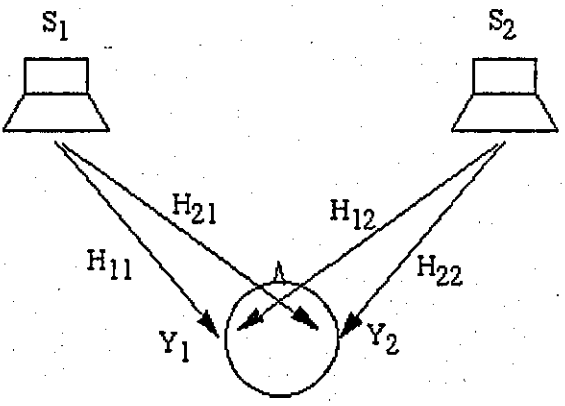


Figure 3.12 – Co-channel cross-talk from Duda [83].

Duda [83] explains that the idea is simply expressed in the frequency domain. In the configuration shown in the diagram above, signal S1 drives the left loudspeaker and signal S2 drives the right loudspeaker. The signal Y1 reaching the left ear is a mixture of S1 travelling through the H11 medium and the "cross-talk" from S2 travelling through the H12 medium. More precisely, Y1 = S1*H11 + S2*H12, where H11 is the HRTF between the left speaker and the left ear and H12 is the HRTF between the right speaker and the left ear. Similarly, Y2 = S1*H21 + S2*H22. What is required is a way of removing the cross-talk components and the effects of the Hnn mediums to yield S1 being purely present at Y1 and S2 being purely present at Y2. This diagram can be mathematically represented as follows:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$$

In order to find the desired outputs at S1 and S2 and hence cancel out the effects of the mediums and the cross-talk the equation needs to be rearranged as follows:

$$\begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}^{-1} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$$

Cross-talk-cancelled stereo can be quite effective when done carefully and can produce elevation as well as azimuth effects. The phantom source can be placed significantly outside of the line segment between the two loudspeakers. Provided the listener is centred between the loudspeakers, crosstalk cancellation is relatively insensitive to front-back motions of the listener, however, crosstalk cancellation is degraded when the listener is off-centre or not facing forward. Another way of saying this is that the sweet spot (the optimum listening position for maximum effect) is long and narrow.

Loudspeaker 3-D audio systems are effective in desktop computing environments. This is because there is usually only a single listener (the computer user) who is almost always centred between the speakers and facing forward towards the monitor. Thus, the primary user gets the full 3-D effect because the crosstalk is properly cancelled. In typical 3-D audio applications, like video gaming, other listeners may gather around to watch. In this case, the best 3-D audio effects are heard by others when they are also centred with respect to the loudspeakers. Off-centre listeners may not get the full effect, but they still hear a high quality stereo program with some spatial enhancements.

For headphone presentation, Wenzel [192] indicates that, "*Alternatively, even inexperienced listeners may be able to adapt to a particular set of HRTFs as long as they provide adequate cues for localization.*" Wenzel further notes that a reasonable approach is to use HRTFs from a subject whose measurements have been "behaviourally calibrated" and are thus correlated with known perceptual ability in both open field and headphone conditions. Wenzel reports that in a recently completed study [192], sixteen inexperienced listeners judged the spatial location of sources presented over multiple loudspeakers in the open field and over headphones. The headphone stimuli were generated digitally using HRTFs measured in the ear canals of a representative subject (a good "localiser") from Wightman & Kistler [196, 197]. For twelve of the subjects, localization performance was relatively accurate, with judgements for the non-individualized stimuli over headphones being nearly identical to those in the open field.

Wenzel further reports that this data suggests that most listeners can obtain useful directional information for an auditory display without requiring the use of individually tailored HRTFs. The results described above are based on analyses in which errors due to front/back confusions were resolved for open field versus simulated open field stimuli. Experienced listeners exhibited front/back confusion rates of about 5% versus 10% and inexperienced listeners showed average rates of about 22% versus 39%. Although the reason for such confusions is not completely understood they are probably due to the static nature of the stimulus and the ambiguity resulting from the "cone of confusion" described by Blauert [21]. For the purposes of spatialising auralisations it would be the simplest approach to used headphones. This makes the presentation as effective as possible and reduces design complexity.

3.6. Head-Related Transfer Functions

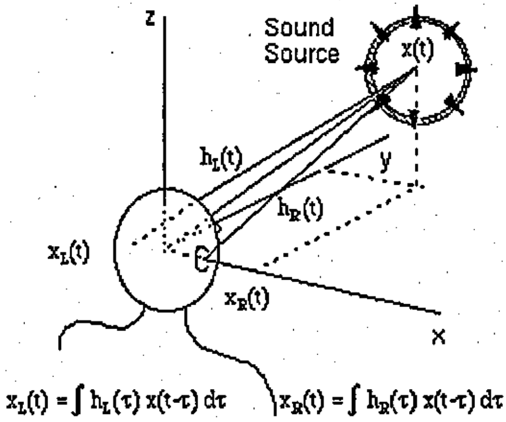


Figure 3.13 – HRTF measurement from Duda [83].

Duda [83] pointed out that in order to find the sound pressure that an arbitrary sound source x(t) produces at the eardrum, all that needs to be done is to find the impulse response h(t) from the source to the eardrum. This impulse response is called the Head-Related Impulse Response (HRIR)(Figure 3.13). In the frequency domain its Fourier transform H(f) is called the Head Related Transfer Function (HRTF). The HRTF provides all of the physical cues for source localization. Each ear requires one HRTF, so for each given fixed position in 3D space it is necessary to determine a pair of HRTFs.

When the HRTFs are applied to a monaural sound source then the resulting listening experience is that of the sound source emitting from the position in 3D space dictated by the pair of HRTFs. If a sound was required to move around the head within a cube of 5 by 5 by 5 vectored coordinates (125 vectored coordinates) then 125 pairs (250) HRTFs would need to be determined. The sound could then be placed in any of the 125 positions by applying the relevant pair of HRTFs to a monaural sound source to create a synthesised binaural sound source. For the purposes of spatialising auralisations, HRTFs offer a flexible and realistic solution. However, it is often difficult to obtain a good set of HRTFs and can be expensive to create your own.

3.6.1 Head tracking

One of the issues addressed earlier was that of binaural recordings being non-interactive. The main issue raised was that no feedback path was present to allow for user response, chiefly the motion of the listener. When a listener hears a sound his/her natural response is to shift gaze to the sound source. This often results in trying to align the head such that the sound source appears directly in the line of sight. Given that binaural recordings are usually confined to fixed positions it is necessary to adjust the perceived location in relation to the listener's head motion. If this can be achieved the listener gains the impression of being in a more realistic virtual acoustic environment. If this issue remains unaddressed then some of the spatial effects can be weakened or even destroyed. Sources that are supposed to be directly ahead or directly behind can be particularly augmented since the rate of change of binaural cues is greatest in those directions.

An obvious solution is to employ some form of head motion tracking system and use the measured parameters to filter the sound source such that it moves to the correct location in relation to the listener's head. When a monaural sound source is convoluted with the relevant pair of HRTFs to produce a perceived 3D sound it seems obvious to change the current pair of HRTFs in relation to the tracked position of the listener's head. This results in a real-time spatial audio system where the HRTFs (and hence relative location of the perceived sound source) are constantly updated. Latency is the time between when a motion is made and the corrected HRTF is used, this should typically be less than 50 ms or the lag will be perceived. If one switches between one HRTF and another, audible clicks may result. This may be overcome by implementing some "crossfading" between the two states. Head tracking is important in creating realistic audio environments. However, for simple spatialisation of algorithm auralisation it is not essential. This thesis is concerned with using low-cost 3D audio.

3.6.2. KEMAR responses

If HRTFs are to be used for the spatialisation of sorting algorithm auralisations then some understanding of their derivation is required. This derivation also shares some common features with binaural recordings.

The HRTF is a function of four variables; three space coordinates and frequency. Most HRTF measurements are made in the far field (greater that one metre), this essentially reduces the HRTF to a function of azimuth, elevation and frequency because the HRTF drops off inversely with range greater than one metre. Duda [83] made a series of HRIR measurements on an acoustic manikin that matches as closely as possible the average human head, this manikin is known as KEMAR (Knowles Electronics Manikin for Auditory Research). To gain an understanding of how KEMAR's response varies with azimuth and elevation, the following graphical representations of the HRIR and HRTF were produced by Duda [83].

Figure 3.14 is an image of KEMAR's experimentally measured head-related impulse response (HRIR). It shows the response of the right ear to an impulsive source in the horizontal plane. The strength of the response is represented by brightness. Duda [83] explains that the sound is strongest and arrives soonest when it is coming from the right side (azimuth = 90°). Similarly, it is weakest and arrives latest when it is coming from the left side (azimuth = 270°). It can also be seen that the arrival time varies with azimuth in an approximate sinusoidal fashion. The arrival time conforms well to the ITD equation. In particular, the difference between the shortest and the longest arrival times is approximately 0.7 ms as the theory predicts the delay to be from one ear to the other.

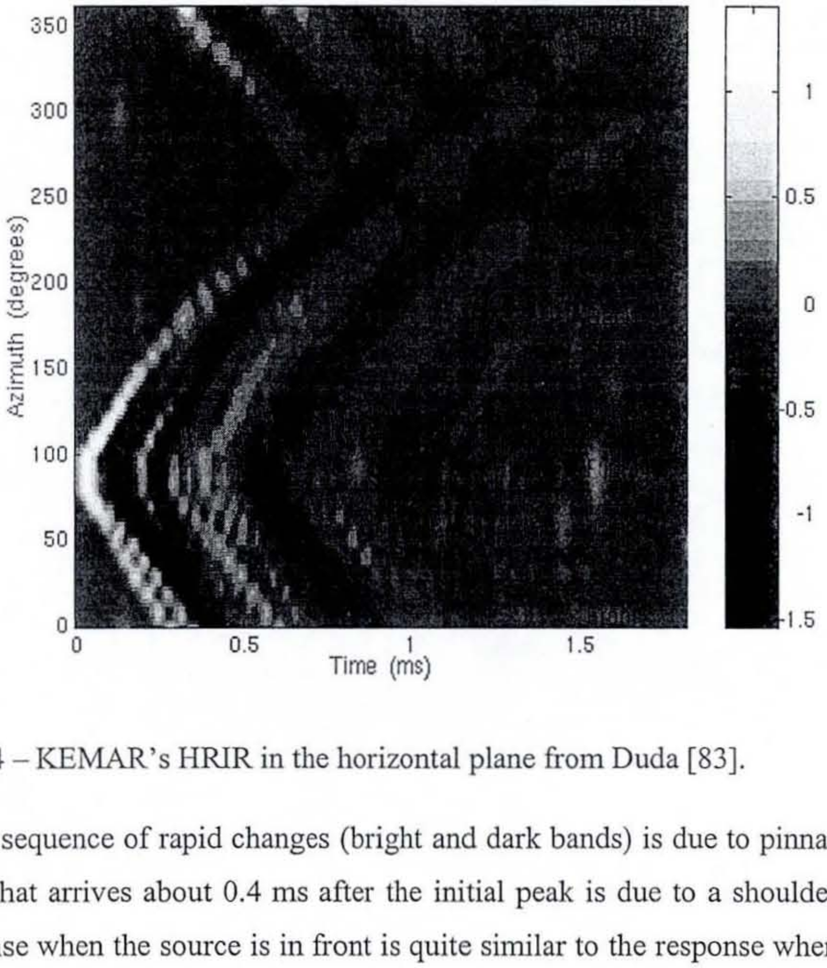


Figure 3.14 – KEMAR's HRIR in the horizontal plane from Duda [83].

The initial sequence of rapid changes (bright and dark bands) is due to pinna reflections. The peak that arrives about 0.4 ms after the initial peak is due to a shoulder reflection. The response when the source is in front is quite similar to the response when the source is at the rear. This correlates with the difficulty associated with localising sound sources in the front/back plane. This difficulty is often overcome by the user moving his/her head to further localise the sound source.

Duda [83] explains that when the source moves around the head in the median plane, the changes were much more subtle and the arrival time was approximately the same. In the horizontal plane the changes were much more dramatic, this was due to the strong asymmetry of the microphone in relation to the manikin head. In the median plane the symmetry is strong and differences do not show up quite so easily. The main changes are in the relative arrival times and strengths of the pinna reflections. This appears in the frequency domain as a notch whose frequency changes with elevation. It can be seen in Figure 3.15 that the difference between front and back shows up once again in the subtle yet clear lack of symmetry about a horizontal line at 90 degrees elevation.

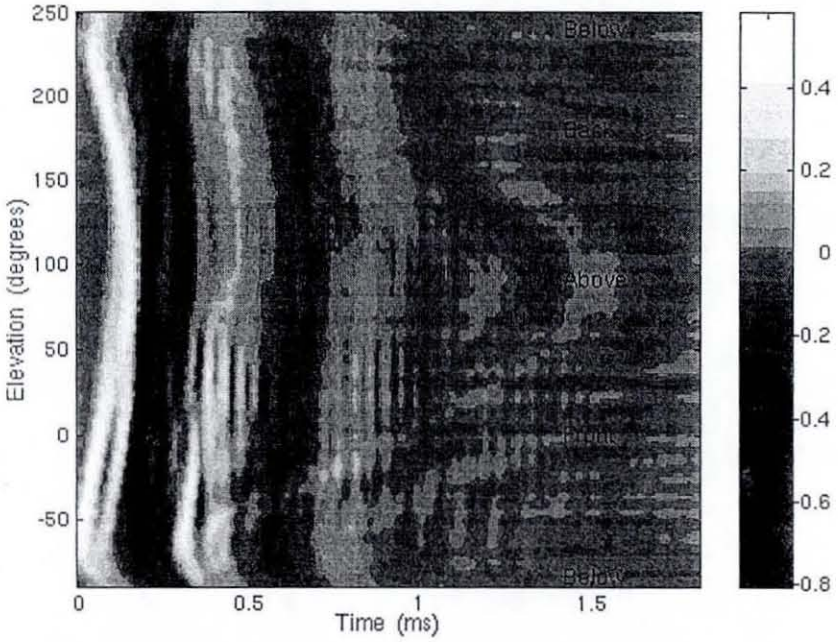


Figure 3.15 – KEMAR's HRIR in the median plane from Duda [83].

The mesh plot in Figure 3.16 shows the frequency response for KEMAR's right ear as the source moves in the horizontal plane. Duda [83] notes that although the surface is bumpy, it can be seen that at any one frequency there is an approximately sinusoidal change with azimuth. As expected, the response is greatest when the source is at 90° and directed into the right ear, and weakest when the source is at 270° on the opposite side of the head.

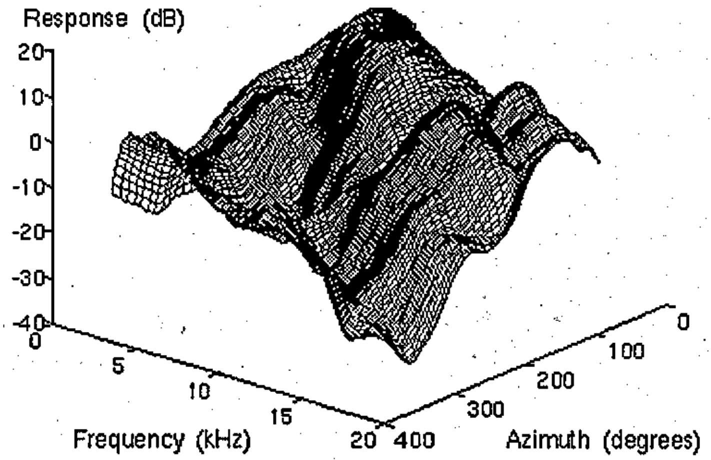


Figure 3.16 - Frequency response for KEMAR's right ear from Duda [83].

Once again, front/back (0° and 180°) responses are quite similar. The graph in Figure 3.17 shows two plots; one of the response from the front and one of the response from the back. The front response is a few dBs greater than the back response in the frequency range from around 4 to 7 kHz. This is largely due to the asymmetry of the pinna. The peak around 4 kHz is due to ear-canal resonance. The notch around 10 kHz that is also clearly visible in the surface plot above is the "pinna notch", whose frequency changes with elevation.

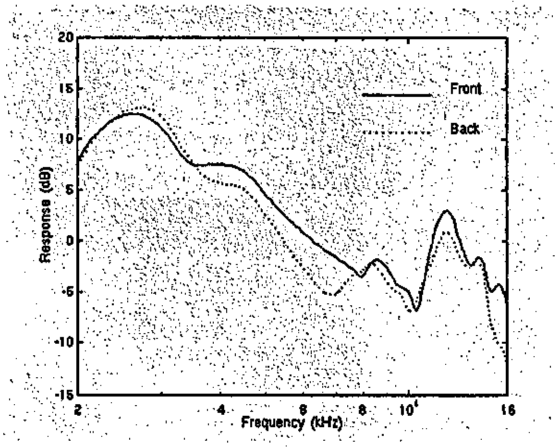


Figure 3.17 – KEMAR's front/back frequency response, horizontal plane - Duda [83].

The plot in Figure 3.18 shows how KEMAR's frequency response varies as the source moves around in the median plane. Duda [83] indicates that the broad ear-canal resonance around 4 kHz does not change and that the frequency of the pinna notch changes significantly with elevation. It goes from just below 6 kHz at low elevations up to approximately 10 kHz as the source moves overhead. When the source is directly above, the notch is hard to see, and the frequency response is fairly flat. It reappears as the source moves around the back of the head and back towards the floor.

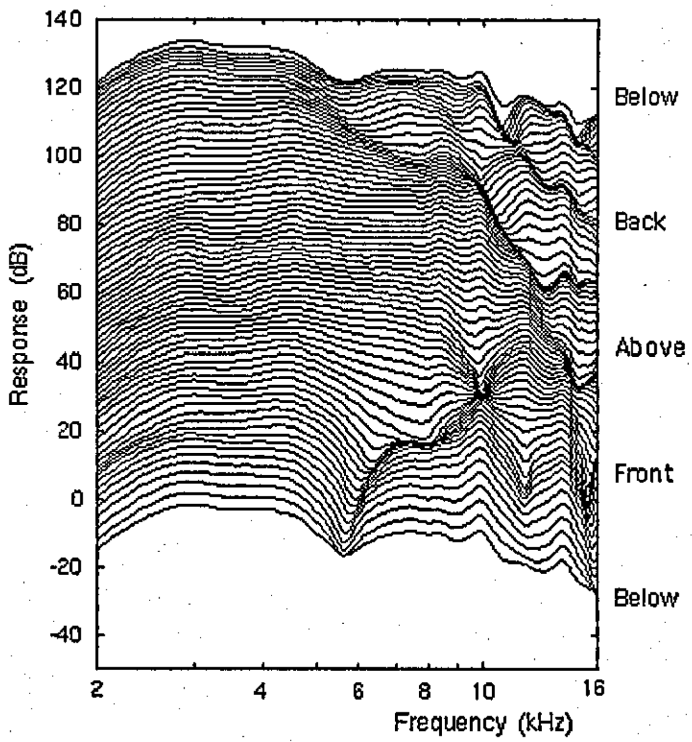


Figure 3.18 – KEMAR's frequency response in the median plane from Duda [83].

As previously mentioned, the shape of the pinnae determine the behaviour in the median plane and this differs from listener to listener. Listeners with smaller ears produce a response with the frequencies shifted higher. This difference in response once again highlights the sensitivity of localising the elevation of sound sources from person to person. KEMAR is not exact for all listeners but is certainly a good approximation for general application.

The complexity of the spectral profile presented to the ears has made it difficult to formulate a comprehensive model of human directional hearing cues for sound from any azimuth or elevation angle. Wenzel [192] highlights the issue of individual differences by stating that, "*A recent study by Wightman and Kistler [197] confirmed the perceptual adequacy of the basic technique for static sources; source azimuth was synthesized nearly perfectly for all listeners while source elevation was somewhat less well defined in the headphone conditions.*" Further to this Blauert [21] has suggested that for successful three-dimensional sound presentation over headphones it is necessary to measure each potential listener's HRTF. However, as Wenzel [192] notes, "*from an applied standpoint measurement of each potential listener's HRTF may not be practical. It may also be the case that the user of such a display will not have the opportunity of extensive training, thus a critical research issue for virtual acoustic displays is the degree to which the general population of listeners can obtain adequate localization cues from stimuli based on non-individualized transforms.*"

Preliminary data [193] suggests that using non-listener specific transforms to achieve synthesis of localized cues is at least feasible. For experienced listeners, localization performance was only slightly degraded, even for the less robust elevation cues. Furthermore, the fact that individual differences in performance could be traced to acoustical idiosyncrasies in the stimulus suggests that it may eventually be possible to create a set of "universal transforms" by appropriate averaging [98] and data reduction techniques or even enhancing the spectra of empirically derived transfer functions [84].

Martens [134] used principal components analysis (PCA) on spectral variation between HRTF's in an attempt to reduce the amount of data necessary to specify the directionally dependent spectral cues. He found that effective transfer functions could be re-synthesised from just a few principle components that captured simple distinctions such as front versus rear, and central versus lateral sound directions.

Kendall and Martens [135] created a complete sphere of simulated transfer functions using pole-zero approximations to measured HRTFs. Perceptual evaluation showed that their transfer functions could support 3D spatial imagery over loudspeakers but the cross talk cancellation filters produced timbral changes that were unacceptable for professional audio [135].

3.6.3. Modelled HRTFs vs. Measured HRTFs

HRTFs are very complex and many spatial audio systems depend upon experimental data such as Duda's KEMAR data shown earlier in this chapter. The main reason for using HRTFs is to capture elevation effects alongside azimuth effects. As previously mentioned, elevation cues are significantly sensitive to individual differences and issues arose about how best to implement a generic 3D spatial audio system. Given these key geometrical differences from person to person it is important to investigate the most effective way to implement HRTFs.

One way is to use a single standard set of HRTFs. There is, as yet, no recognised standard set of HRTFs available. As previously mentioned most data is purely experimental. The immediate problem associated with using a single set of HRTFs is that they will not necessarily translate well with all potential listeners. The best that can be hoped for it to generate a set of HRTFs taken from a model whose characteristics closely match the statistical norm. However, it is inevitable that in this case a percentage of listeners will experience poor elevation results. This 'single set' approach is certainly the most inexpensive but also the most inflexible.

Another approach would be to divide the general population into sub-groups based on their physical attributes, particularly concentrating on the pinnae and head geometry. A set of HRTFs could then be generated for each statistical norm within each of the sub-groups. Potential listeners could then be categorised into one of the sub-groups by their physical attributes and the closest relevant set of HRTFs could be implemented. The result is a greater set of HRTFs and a more improved overall effect on the general population. Although this approach will yield closer results than using a single set of HRTFs it is still not the ideal solution.

The obvious ideal solution is to tailor the HRTFs for each individual listener. This would prove costly and time consuming as each user would require measurement of HRTFs. The results, however, would be as close as is physically possible because the physical attributes of the listener would be identical to the physical attributes of the recording subject.

The most flexible approach would be to generate a model HRTF with changeable parameters. The parameters could be of or relating to the physical attributes of the target listener. In this case the HRTFs would adapt to each individual user.

3.6.4. HRTF Models

There are several possibilities for modelling head related transfer functions and the topic has been subject to some extensive research. The following sub-sections deal with Duda's explanation of modelling ITD, IID, spherical head and pinna.

The ITD model, which is shown in Figure 3.19, is one of the easiest and most effective HRTF models. The motion of the sound source within the azimuth is controlled by an azimuth-dependent time delay that differs for each ear. The ears are clearly shown to be modelled in opposition by the combined (-π/2 and +π/2) 180 degrees (π) phase difference between them.

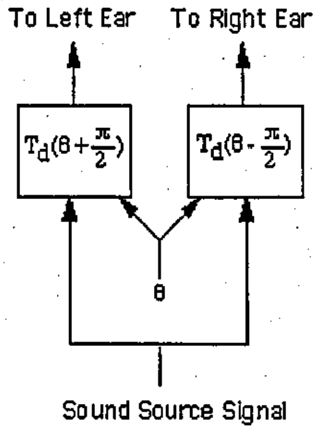


Figure 3.19 – ITD model from Duda [83].

Duda describes that using the same geometrical argument that was employed to derive the ITD:

$$\mathrm{ITD} = \frac{a}{c}(\theta + \sin\theta) \ , -90^\circ \leq \theta \leq +90^\circ$$

the time-delay function is given by:

$$T_d(\theta) = \begin{cases} \frac{a}{c}(1 - \cos\theta) & \text{if } |\theta| < \frac{\pi}{2} \\ \frac{a}{c}(|\theta| + 1 - \frac{\pi}{2}) & \text{if } \frac{\pi}{2} < |\theta| < \pi \end{cases}$$

Where 'a' is the head radius and ‘c’ is the speed of sound.

It is evident that in this model the energy emitted at each ear is the same as no attenuation or amplification components are present. It is the introduced time delay that gives the listener the impression of a phantom sound source. This model exploits the features of the ‘precedence effect’ or the ‘Law of the First Wavefront’ as a means of altering the azimuth of the sound source. The simplicity of the model does dictate that it produces no sense of externalisation and no front/back discrimination. It does, however, produce a phantom sound source that is capable of moving smoothly from the left ear through the head to the right ear as the azimuth changes from -90° to +90°.

The effects of head shadow can be modelled by filtering the high frequency component of the signal source when the head occludes the path to the receiving ear (Figure 3.20). Once again the frequency filtration will be azimuth dependent where the model boosts the high frequency component when the azimuth is zero degrees and attenuates the high frequency component when the azimuth is 180. Duda [83] describes that Lord Rayleigh’s analytical solution for the IID for a rigid sphere was in the form of an infinite series, yet its magnitude response can be well approximated by the one-pole, one-zero transfer function

$$H(s, \theta) = \frac{\alpha(\theta)\, s + \beta}{s + \beta}$$

$$\text{where} \quad \alpha(\theta) = 1 + \cos\theta \quad \text{and} \quad \beta = 2\frac{c}{a} \ .$$

By offsetting the azimuth to the ear positions (90 degrees each way), the following simple IID model is obtained:

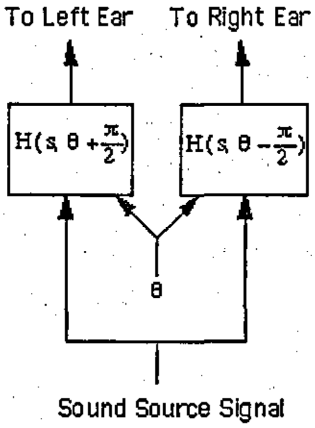


Figure 3.20 – IID model from Duda [83].

It can be seen from the simplicity of the one-pole, one-zero transfer function that the model can be implemented as an infinite impulse response filter (IIR).

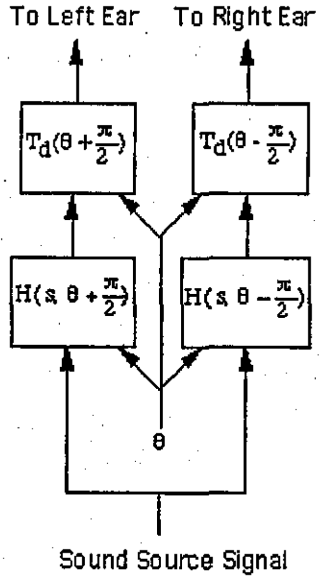


Figure 3.21 – Combined IID and ITD model from Duda [83].

It is now possible to cascade the ITD model and the IID model (Figure 3.21) in order to obtain an approximate spherical-head model. With this model, however, there is no sense of externalisation or elevation; the model simply produces an azimuth controlled phantom image somewhere along the axis between the two ears.

When no IID model is present, some wide-band signals give the impression of two sound images, one displaced and one at the centre of the head. The is due to the fact that the ITD cue is telling the brain that the source is displaced yet the energy at the two ears is the same and therefore the IID cue is telling the brain that the source is in the centre. Conversely, when no ITD model is present, a listener once again gets the impression of two sound images due to the conflicting information between the IID cue and the ITD cue. Here, the IID cue is telling the brain that the source is displaced due to the difference in energy levels received at the ears while the ITD cue is telling the brain that the source is in the centre due to no delay being present. The problem of producing split images is overcome by this combining of both the ITD model and the IID model.

One way to add the missing externalisation is to introduce some simulated room echo; this gives the impression of externalisation or 'out-of-head' sensation to the listener. The diagram shown in Figure 3.22 illustrates this method by introducing some simulated echo with variable delay and magnitude. The gain 'K' should be between zero and one, the delay 'T' should be between 10 and 30 ms. This very simple room model is not fully effective as it only produces externalisation when the azimuth is anything but zero. Also, the same echo is sent to both ears and is therefore not azimuth dependent.

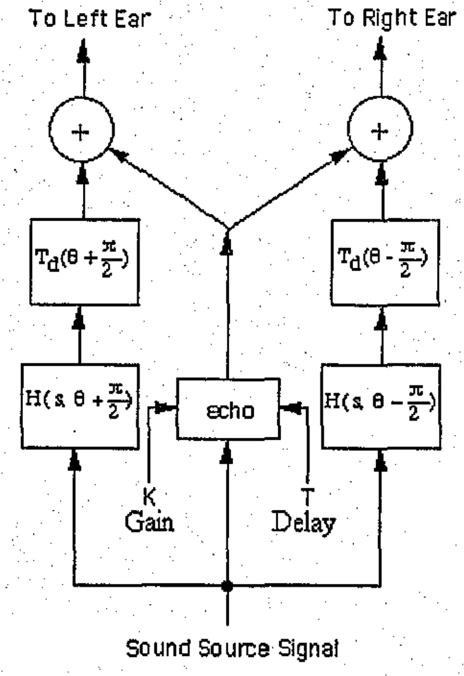


Figure 3.22 - Combined IID and ITD model with echo from Duda [83].

As previously mentioned, the outer ear or pinna can be seen as a directionally dependent filter. It can amplify some frequencies through its resonant cavities while at the same time attenuating other frequencies due to the interference effects caused by other geometrical attributes. Batteau [14], Watkins [189] and other researchers have suggested modelling the effect of the pinna in terms of one or more pinna echoes. The diagram in Figure 3.23 shows a typical model that has a multipath structure.

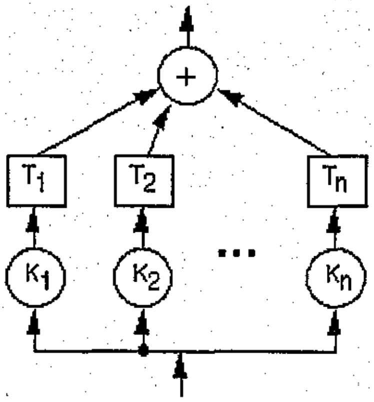


Figure 3.23 - Monaural Pinna Model from Duda [83].

Each path produces an echo that is determined by two variables of gain and time delay. The problem is to determine what the actual values of gain K and time delay T are in relation to azimuth and elevation. Some research has been carried out in this field but nothing concrete has yet emerged. A set of rules and relationships must be identified in order to estimate the parameters necessary for a given set of geometrical attributes pertaining to the physical characteristics of a listener.

In order to fully synthesise the three-dimensional listening experience it is necessary to create a much more complex model than has been shown so far. The models previously described deal with simple yet very important localisation cues. There are, however, many other features of 3D audio that need to be modelled in order to further realise a truly immersive spatial audio environment, necessary models such as shoulder reflection models, torso diffraction models, room models, object occlusion models, ear canal resonance models etc. A partially combined model is shown below in Figure 3.24 with only four components.

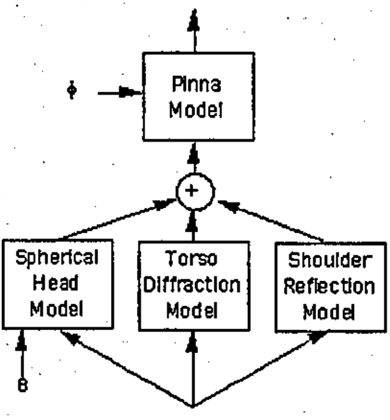


Figure 3.24 – Combined models from Duda [83].

As more and more features are modelled and added to the overall combined HRTF model so it is possible to get better approximations of the actual HRTF. It is clear that there is still room for much research in this area before a complete HRTF model can be achieved, what has been accomplished so far gives a rough approximation of the HRTF but is by no means exact or exhaustive.

3.7 Systems using HRTFs

To reiterate, simple spatial audio systems have several disadvantages. Stereophonic, multi-channel and binaural recording systems all have fixed limitations and HRTF models are not well established enough to be of any real use at present. The most commonly used technique for creating spatial audio is that of the measured HRTF based systems. They provide an acceptable level of accuracy coupled with a viable implementation complexity. HRTF based systems are capable of producing extremely accurate azimuth effects as well as reasonable elevation and range effects. Some difficulties with range and particularly elevation emerge due to geometrical differences from person to person. The following sub-sections now describe some existing HRTF based systems that have been considered for implementing spatailised sorting algorithm auralisations.

3.7.1. The Convolvotron

The Convolvotron [194] was a system manufactured by Crystal River Engineering [58] and was originally developed for the National Aeronautics and Space Administration (NASA). It provides a conceptually simple way of implementing HRTFs in order to create a spatial audio environment. The Convolvotron is a system that consists of two 'convolution engines' (Figure 3.25) each of which is used to convolve the same monaural audio input stream with a finite segment of a head-related impulse response (HRIR) retrieved from an indexed table of measured values. The outputs of the convolvers are passed through amplifiers to headphones worn by the listener. If the HRIRs for the listener are sufficiently close to the HRIRs used by the convolvers, the sound delivered to the listener's ears will contain all the correct spatial cues, and the sound image will be properly localized. The tables indexed by the desired azimuth and elevation yield the relevant pair of HRIRs for convolution.

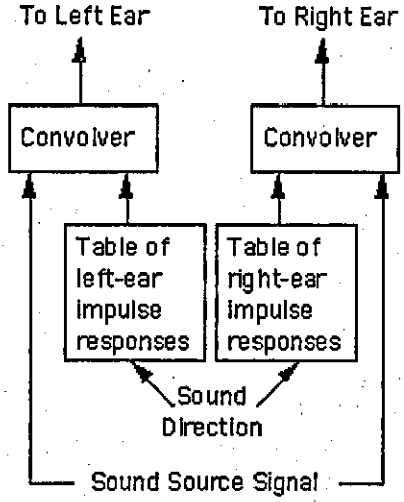


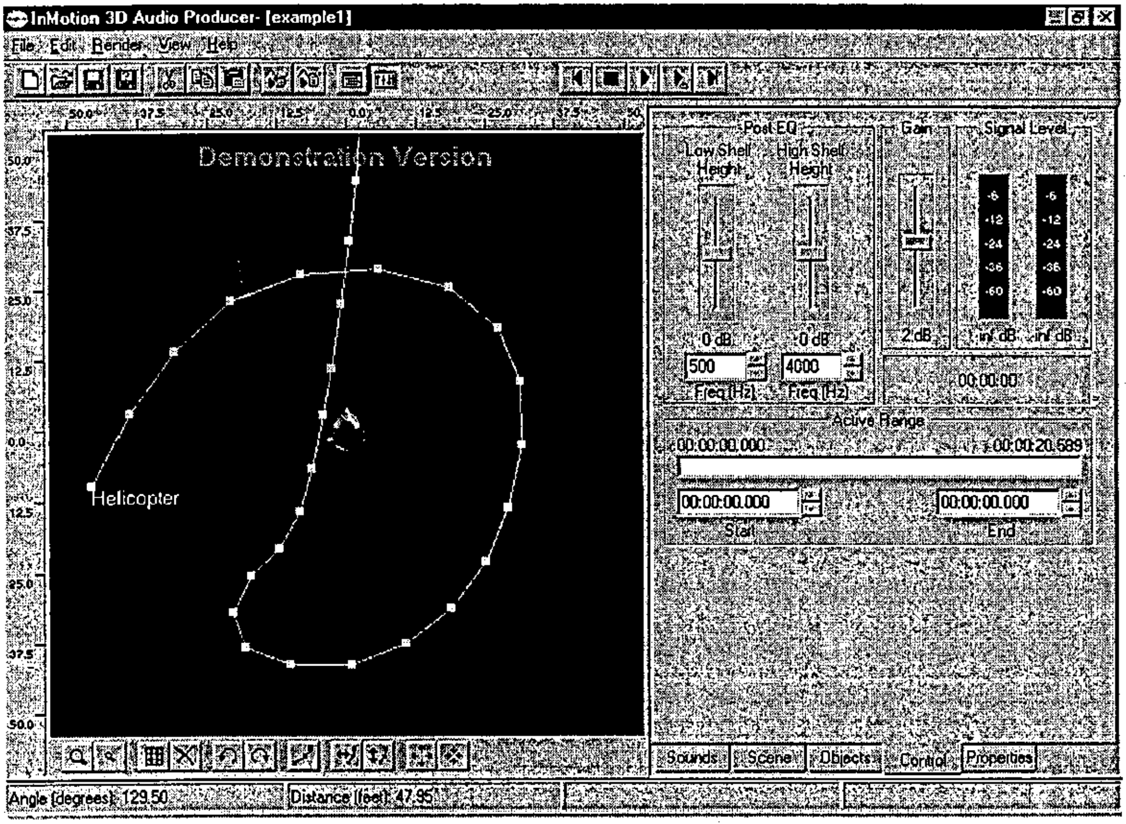
Figure 3.25 – Convolvotron system from Duda [83].

The Convolvotron system can be customized for a particular individual by measuring and using that person's HRIRs. The HRIRs are simply measured for the desired listener and stored within the HRIR tables and indexed during playback for convolution with the monaural signal source. If the stored HRIRs match those of the current listener then the spatial audio effect is as close as possible. The HRIR tables are indexed by azimuth and elevation only. Range effects are introduced by using the distance from the source to each ear. It is evident that the quantity of HRIRs required (and the size of the HRIR tables) is dictated by the parameters of the audio space and the resolution of the sound source positioning. By employing coarse spatial sampling the amount of HRIRs (and hence table sizes) may be reduced, this does introduce quantisation errors though. The coarser the spatial sampling the greater the quantisation errors and hence the more augmented the perceived sound source becomes. It is, therefore, important to maintain a minimum resolution when measuring HRIRs.

Due to the real-time nature of the Convolvotron it is possible to further enhance the spatial audio listening experience by implementing a head-tracking device. The parameters obtained by the head tracker (azimuth and elevation) can be used to index the HRIR tables within the Convolvotron system. The addition of this feedback path allows the listener to interact with the immersive spatial audio environment. When the listener moves his/her head then the perceived phantom sound source moves in accordance. The result is a more realistic listening experience. Adding extra features such as echoes and room reverberation by including a room simulation model can further enhance the basic system of convoluting tabulated HRIRs with monaural signal sources. The Convolvotron was a relatively expensive system and is therefore unsuitable for creating low-cost spatialised algorithm auralisations.

3.7.2. InMotion 3D Audio Producer

InMotion 3D Audio Producer [107], shown in Figure 3.26, was created by Human Machine Interfaces Inc and is a tool for creating realistic auditory scenes. InMotion incorporates Wave Arts 3D Audio and Acoustic Environment Modelling technology, which creates quite effective 3D audio effects over both headphones and loudspeakers. Acoustic environments are realistically simulated, including reverberation, motion effects, distance cues, object occlusion etc.



0 dB | 0 dB | 2 dB | inf dB | inf dB

Freq (Hz): 500 | Freq (Hz): 4000

00:00:00

Active Range: 00:00:00.000 – 00:00:20.589

Start: 00:00:00.000 | End: 00:00:00.000

Angle (degrees): 129.50 | Distance (feet): 47.95

Figure 3.26 – InMotion 3D Audio Producer screen shot [107].

InMotion is essentially a sophisticated sound file mixing application, creating a 3D output sound file from a set of input sound files. However, InMotion also has real-time previewing capabilities; while producing auditory scenes, any portion of the scene can be previewed in real-time. If the scene is too complicated to be rendered in real-time, InMotion will automatically render the scene to a file for later playback.

A lot of commercial audio products are described as having 3-D capability, but in fact there is great disparity between the various technologies in use. Unfortunately, many of the weakest products are marketed with the most exaggerated claims. For example, a number of stereo multimedia speakers are marketed as having "3-D" technology. These speakers incorporate a simple circuit that has the effect of widening the perceived sound field of a stereo recording. That is, the sound images that would normally extend to the locations of the left and right speakers are widened to extend beyond the speakers. These systems should more properly be called stereo enhancement or "widening" systems. They have no ability to position sounds around a listener, or to position sounds behind, above, or below the listener. The term '3-D audio' really describes a much more sophisticated system than can position sounds anywhere around a listener. Although InMotion is described as a 3D audio producer, this is not strictly true. The system omits the ability to vary the elevation of the sound source; it is only capable of synthesising sounds in a fixed elevation plane. The only variables permitted for object location are azimuth and distance. It can, therefore, be described as a pseudo-3D audio producer as it does not permit sound source placement in a truly three-dimensional capacity.

InMotion 3D Audio Producer is a system that works by mimicking the process of natural hearing, essentially reproducing the sound localization cues at the ears of the listener. This is done by using a mathematical model (HRTFs) of a human listener that can generate the proper sound cues for any desired sound direction. The model used by InMotion is chosen to be as generic as possible, so that the resulting localization cues will work for a majority of listeners. The performance of the InMotion system depends greatly on how well its generic head model happens to match the listener. The head model used by Wave Arts 3-D for the InMotion system has been specifically engineered to be optimal for a majority of listeners.

With the InMotion system the cues are applied to a sound by processing the sound through a pair of digital filters (equalizers), which create the left and right ear signals to send to the listener. Essentially, the head model used by Wave Arts consists of a large set of digital filters (HRTFs) with each pair of digital filters corresponding to a sound location. The head model used by Wave Arts 3-D consists of 710 different filter pairs to reproduce 710 different directions around the head.

The Wave Arts Acoustic Environment Modelling system combines Wave Arts 3-D with accurate simulations of the following acoustic phenomena:

- Doppler Motion Effect
- Air Absorption
- Distance Cues
- Object Occlusion.
- Reverberation

The Doppler motion effect is commonly heard in nature as a pitch change when a speeding object passes a listener. When the object is approaching the listener, the pitch is higher than the resting pitch of the object. This is because in the time it takes the object to emit one waveform the object has moved closer to the listener, and thus the emitted wavelength is shorter than normal. Similarly, when the object is retreating from the listener, the pitch is lower than the resting pitch, because the emitted wavelengths are longer than normal. InMotion allows placement and movement of sound sources within a spatial environment. It is, therefore, important for InMotion to simulate the Doppler effect as it is important for generating realistic motion effects.

When sound propagates through air, some sound energy is absorbed in the air itself. The amount of energy loss depends on the frequency of the sound and atmospheric conditions. High frequencies are more readily absorbed than low frequencies, so the high frequencies are reduced with increasing distance. For example, at 100 metres distance, 20 degrees Celsius, and 20% humidity, a 4 kHz tone will be attenuated by about 7.4 dB.

However, the attenuation would be less than 1 dB for distances less than 10 metres. InMotion simulates this effect by employing a low-pass filter whose cut-off frequency depends on the distance to the source.

The principal cue for distance is the loudness of the sound. A sound source will be louder when it is closer to the listener than when it is farther away. However, this cue is often ambiguous because the listener does not know a priori how loud the source is. Thus, a moderately loud crashing sound could be perceived as a quiet, close crash, or a distant, loud crash.

Another important cue for distance is the loudness of reverberation (see section 3.2.5). InMotion models this by decreasing the amplitude of the direct sound by a factor of one half (3dB) for every doubling of distance. The amplitude of the reverberation, however, does not decrease considerably with increasing distance. The ratio of the direct to reverberant amplitude is greater with nearby objects than with distant objects. Thus, distant objects sound more reverberant than close objects.

InMotion models the decay of reverberation over distance by attenuating the reverberated sound at a rate of half the slope of the direct sound, or 3 dB per doubling of distance (equal to 10 dB drop for a factor of 10 increase in distance). In most reverberant spaces, the reverberation does not actually drop this fast with increasing distance. However, for the purposes of creating an effective sounding scene, it is often necessary to tweak the parameters to get the desired effect. For very close distances, the reverberation is 20 dB below the direct sound, equal to a 10% reverberation mix. For increasing distances, the ratio of direct sound to reverberation decreases, and at 100 feet the reverberation is louder than the direct sound.

When a sound source is behind an occluding object, the direct path sound must diffract (bend) around the occluding object to reach the listener. Low frequencies with wavelengths larger than the size of the occluding object will not be affected much by the occluding object. High frequencies with wavelengths smaller than the size of the occluding object will be shadowed by the object, and will be greatly attenuated. InMotion

3D Audio Producer simulates the effect of an occluding object by employing a low-pass filter whose cut-off frequency depends on the size of the occluding object. Simulating object occlusion is important to achieve realism in film/video soundtracks where sound emitting objects are visibly moving behind occluding objects.

The Wave Arts Acoustic Environmental Modelling system is implemented using the signal routing shown in Figure 3.27 below. The signal routing is conceptually similar to the routing seen in multichannel mixing consoles; input signals are individually processed, mixed to a set of shared signal busses, and then the bus signals are processed and output. In Figure 3.27, the input signals shown at the top represent the individual monophonic (monaural) object sounds to be spatially processed to create the scene.

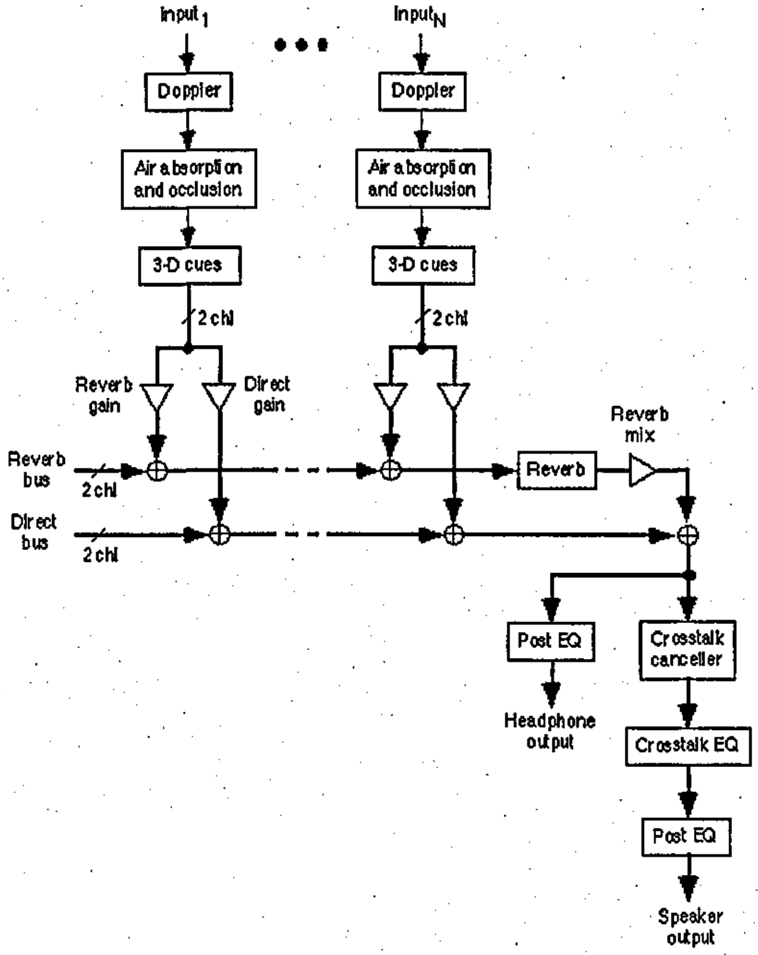


Figure 3.27 - Wave Arts Acoustic Environmental Modelling system signal routing.

Each input signal is processed through the Doppler effect, then the air absorption and occlusion effect, and then the 3-D spatial effect, labelled "3-D cues" in the figure. The Doppler effect and air absorption effect are controlled by the distance between the sound object and the listener. The occlusion effect is controlled by the position of the sound object, which determines the degree to which the sound object is occluded, and the dimensions of the occluding objects. The 3-D spatial effect is controlled by the position of the sound object relative to the listener. The 3-D spatial effect creates stereophonic (two channel) output. The figure does not show the individual left and right channels output from the 3-D spatial processor; instead the stereo signals are labelled "2 chl."

The output from the 3-D spatial processor is split into two stereo signals, which are mixed to the "reverb bus" and the "direct bus," each of which is a stereo bus. The amount of sound mixed to each bus depends on the "reverb gain" and "direct gain" mixing gains. These gains are controlled by the distance from the sound to the listener according to the current distance model. Typically, the distance model parameters are set up so that the direct to reverberant ratio increases as the sound object distance decreases.

The 'reverb bus' contains a mix of all sounds that are to be sent to the reverberator. These are processed by the reverberator and the result is mixed with the direct bus. The reverb mix gain determines the overall level of reverberation in the scene. The reverberator is controlled by the scene environment parameters, which include the reverberation time, room size, damping, etc.

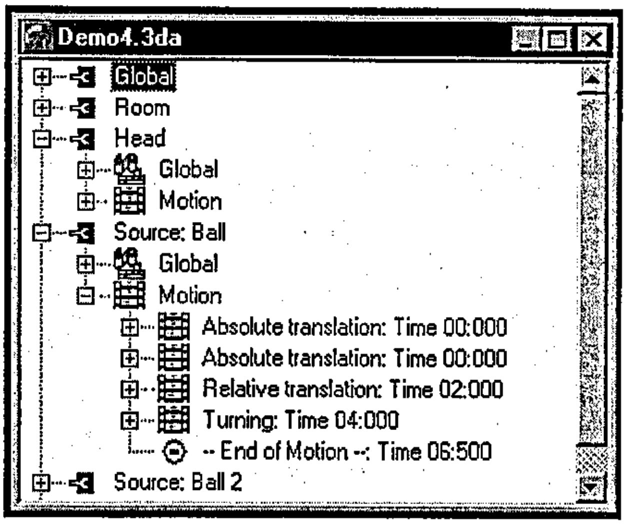
The direct bus output is suitable for listening to through headphones. The headphone output is simply the direct bus processed through a set of tone controls labelled "Post EQ".

For playback over loudspeakers, the direct bus must be further processed by the crosstalk canceller. The crosstalk canceller is controlled by the speaker angle parameter. The output of the crosstalk canceller is processed by the crosstalk equalization stage, and this signal is further processed by a set of tone controls labelled "Post EQ" and the result is output to the speakers. The InMotion system does provide quite realistic 3D audio. However, the main disadvantage is the lack of elevation which makes it unsuitable for the proposed 3D auralisations of concern in this thesis.

3.7.3. 3D Audio – Climax Software Solutions

Climax Software Solutions describe their 3D Audio system as a system that allows the creation of sound files that convey true spatial audio placement, which supports multiple sound sources and features accurate physics. The prerequisite, however, is that for full effect the rendered output is best heard over headphones. Unlike InMotion 3D Audio Producer, Climax's 3D Audio system does not incorporate cross-talk cancellation to permit playback over open loudspeakers. Although the effect is discernable over loudspeakers, it is with headphones that a direct link is made between Climax's 3D Audio system and the listener's ears.

The program focuses on the various empirical influences that determine the way the human brain localizes audio, the most essential of these being modelled during simulation. Since the real world envelops a listener with objects that emit and reflect sound waves, Climax have ensured that 3D Audio represents the physical surroundings of these sources. 3D Audio achieves this by introducing a virtual room in which, a virtual head and an arbitrary number of virtual sounds may be positioned. The reflections from virtual walls are calculated according to the order of reflections specified by the user. Both head and sound source may be animated to give the impression of sound source motion and/or head motion when listening to the rendered output. 3D Audio requires a set of parameters from the user pertaining to the head position and motion, source position and motion, monaural input file, head related transfer function, room size and reflection coefficients. Theses parameters are entered into a tree structure shown in Figure 3.28.

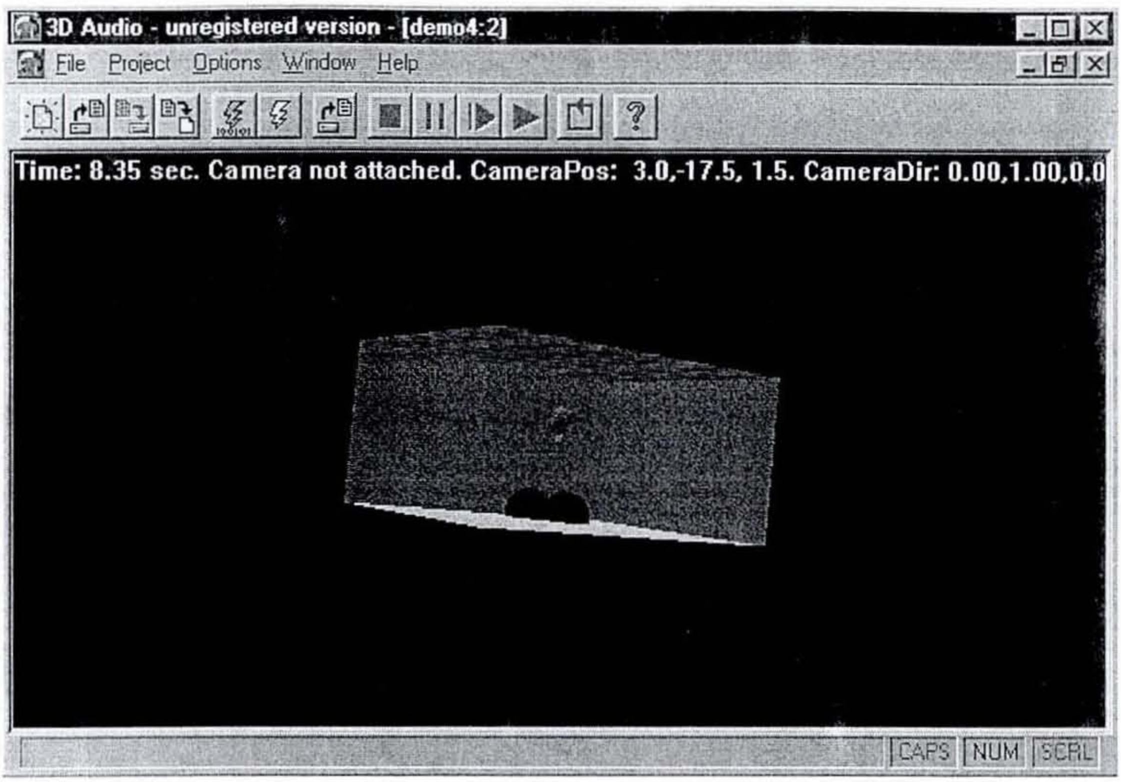


Demo4.3da

- Global
- Room
- Head
  - Global
  - Motion
- Source: Ball
  - Global
  - Motion
    - Absolute translation: Time 00:000
    - Absolute translation: Time 00:000
    - Relative translation: Time 02:000
    - Turning: Time 04:000
    - -- End of Motion --: Time 06:500
- Source: Ball 2

Figure 3.28 - 3D Audio screen shot.

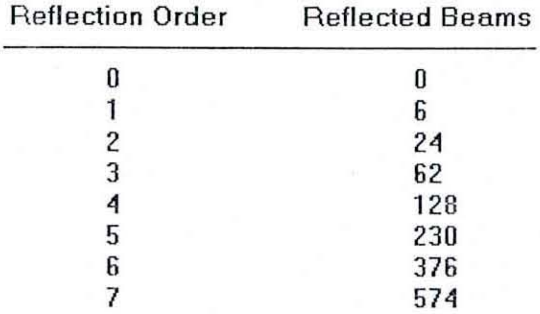
Climax supply their own set of HRTFs that were obtained by empirically measuring impulse responses from an unspecified dummy head with microphones placed within its ears. Since each individual response strongly depends on the relative direction of the sound, the included set of impulses covers the majority of possible directions. Climax state that "*Applying these data to a sound file, i.e. convoluting an impulse response corresponding to a certain direction with the sound data, yields a vague impression of spatial positioning of the sound.*" The reason for the uncertainty of this statement is due to the fact that the impulse response measurements were taken within an anechoic chamber, which is perceptually confusing. Therefore, a virtual room was introduced to allow for signal reflections from the surrounding environment. Providing that the sound intended for the left ear reaches only the left ear and likewise for the right, these reflections supply additional information for localization and, hence, a more realistic immersive spatial audio environment. Once the user has specified all of the required parameters, an animated scene preview may be undertaken, as shown in Figure 3.29, to allow the user to monitor the motion of the head and/or sound source.



Time: 8.35 sec. Camera not attached. CameraPos: 3.0,-17.5, 1.5. CameraDir: 0.00,1.00,0.0

Figure 3.29 – 3D Audio environment screen shot.

The simulation environment employed by Climax's 3D Audio system is modelled to have a regular 'boxlike' shape. To enhance realism, the user may define the interior dimensions and reflecting properties of each surface. The reflection coefficients of the walls may be given in the range from 0.0 to 1.0, where zero means total absorption and one means total reflection. Thus, a wall may be omitted by setting its coefficient to zero. Since it is computationally necessary for the rendering time to be finite, Climax has permitted the user to specify the reflection order. A value of N means convoluting the line-of-sight wave along with indirect waves arriving from a maximum of N walls. The reflection order table is given in Figure 3.30.



Reflection Order	Reflected Beams
0	0
1	6
2	24
3	62
4	128
5	230
6	376
7	574

Figure 3.30 – 3D Audio reflection order table.

Each higher order of reflection introduces additional waves from various directions; therefore, each wave must be processed using a different impulse response. Hence, rendering time dramatically increases with N. Using reflections of first order significantly increases the simulation's "naturalness". Second-order reflections perform a little better, and quadruple the rendering time. Climax state that seven is the highest order of computation that completes in finite time. With an order of the magnitude of seven, rendering takes 2000 times the project's duration to complete.

Unlike InMotion, there is no provision for such features as object occlusion, air absorption or Doppler Motion Effect. The system does permit multiple sound sources and allows the user to fully animate both the head and the sound source. Although it is not as thorough as InMotion in some areas it is more detailed in others. Climax's 3D Audio system allows the user to specify his/her own set of HRTFs which gives the system greater flexibility and permits it to be tailored to the target listener. Unlike InMotion, 3D Audio includes elevation cues that permit it to be called a truly three-dimensional spatial audio system. The main disadvantage with this system is that it facilitates predetermined 3D audio animations and would therefore be unsuitable for spatialising algorithm auralisations that are to be presented in real time during execution.

3.7.4. Ambisone - Prosoniq Products Software

Ambisone by Prosoniq [156] is a plug-in for Steinberg's Cubase VST that allows the user to mix audio tracks in full 3D stereo. Prosoniq claim that their unique Virtual Scenery Modelling approach is different to methods found elsewhere. They state that it not only allows for 3D placement when monitoring the mix through a loudspeaker set-up but also retains the three-dimensional sound image when listening through headphones. This indicates that Prosoniq have employed some cross-talk cancellation. The feature that sets Ambisone apart from other 3D audio tools is the intuitive user interface (shown in Figure 3.31); it permits the listener to adjust the different parameters while monitoring both position and elevation in real time.

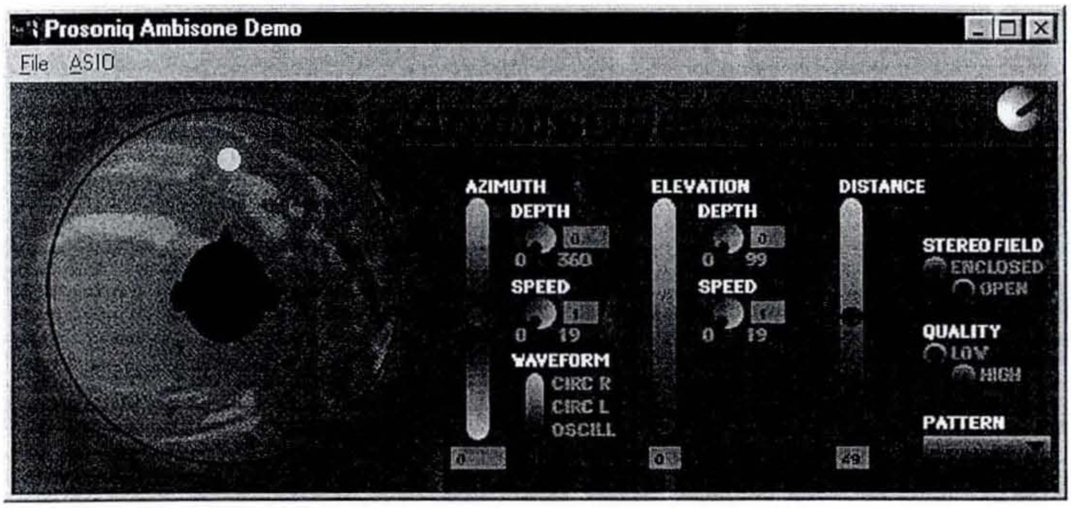


Figure 3.31 - Ambisone screen shot.

Distance and elevation parameters allow for placing the source further away from or above the listening position. As opposed to other common methods, Prosoniq's Virtual Scenery Modelling approach simulates relative sound positioning using a virtual head in a virtual listening position by rendering real objects instead of using pre-calculated or measured filters thus yielding considerable quality, continuity and convincability. The user interface allows the listener to drag the virtual sound source around the virtual head using the mouse, the response is instant and the movement of the sound source may be heard in real-time. The main disadvantage with this system is that it facilitates 3D audio animations of predefined looped audio samples and would therefore be unsuitable for spatialising algorithm auralisations that are to be presented in real time during execution.

3.7.5. Other 3D spatial audio systems.

The AKG CAP 340 system developed by Persterer [154] incorporates the use of filters to simulate head related transfer functions (HRTFs). The filter outputs are fed to headphones and no provision has been made for open field speaker systems. Persterer accommodates the importance of room reflections as they affect sound localization. He implements a simple delay and then assigns a direction by utilising a dedicated filter pair. Persterer states that, "*The simulation of HRTF's with an FIR filter requires an impulse response duration of some milliseconds. The required processing of several sound sources and their reflections calls for computing power of several hundred million operations per second.*" Hence the computational overhead of this system is relatively high. Binaural mixing software (SPATMIX) has been developed for the CAP 340. It is structured for the binaural processing of up to 32 input signals enabling sets of one direct sound and three reflections to be simulated. Special filters simulate the absorption properties of three materials. The proposed auralisations within this thesis are aimed at being low cost and independent of hardware. This system is therefore unsuitable as it requires some dedicated hardware.

The Focal Point 3-D Audio System was a Macintosh II based application that used a widely available inexpensive Macintosh II accelerator card as its signal processor. This system has been developed with the intention of being employed for the production of applications relating to virtual environments and future aircraft cockpits. The binaural technology that the system employs is based upon head-related transfer functions. This system is modular and possesses at least four 3-D audio channels that can be individually placed and moved by the use a mouse, keyboard and RS-232 port commands. The system accommodates the use of multiple sets of binaural HRTFs at the same time. The Focal Point 3-D Audio System also incorporates a head-tracking feature and has a typical

Macintosh interface. Preliminary experiments have revealed large timbre differences dependent upon the choice of pinnae sets, which illustrates the importance of the method of obtaining HRTFs. This subsequently generates differences in localization cues causing erroneous sound source localisation. The proposed auralisations within this thesis are aimed at being low cost and independent of hardware. This system is therefore unsuitable as it requires some dedicated hardware.

The Auris Corporation has developed a 3-D spatial sound processor entitled the VS-1. This sound processor is equipped for use with both headphone applications and open field loudspeaker presentation. The system assumes that the input audio is monophonic and therefore contains little or no spatial information. There are two parts to the spatial sound processor's acoustical simulation design. The first part captures the acoustics of the head pinnae and torso that are responsible for perceived direction. This provides the user with three-dimensional panning through a full range of both azimuth and elevation. The second part captures the acoustics of the user specified room or environment. This environmental simulation also includes directional acoustics and creates the illusion of the full three dimensional environment. The Auris Group states that the most important feature of their environmental modelling is that it captures the spatial-temporal distribution of sound in a natural environment. The time, intensity and direction of reflected sound changes in response to the position of a sound source and the listener in the model room. The combination of directional and environmental simulations provides the user with control of source distance and environmental shape.

With the VS1 processor design, virtually all of the signal processing algorithms employed are designed for dynamic control by the user. The direction and distance of a sound source can be smoothly varied and is intended to automatically include acoustic features such as Doppler shift and air absorption. The dynamic steering of the sound sources in three dimensions is implemented with time varying filtering based on the continuous interpolation of directional transfer functions that are stored in the processor's memory. The environmental simulator contains elements that are similar to conventional reverberation generators with the exception that the gain, delay and filtering of reflected sound is designed to be continually responsive to movement within the modelled environment. Environmental sound processing includes static filters that capture absorption and transmission loss for walls, objects and organic material. Sampling rates for the spatial sound processor are 44.1kHz or 48 kHz and the system has also been designed to be fully MIDI compatible.

For applications requiring fast animation, special hardware is used to perform a small number of HRTF convolutions to model the direct sound arrival plus a number of early reflections, and room reverberation. An example of this approach can be seen in Lake DSP's AniScape and MultiScape [123] applications. Real-time convolution with very long room responses can be achieved using very high performance processors, such as the Huron digital audio convolution workstation built by Lake DSP. The Huron is capable of computing both left and right binaural responses with a length of over 5 seconds each. The intention of the development at Lake was to produce a DSP system capable of giving a subject the illusion of a particular acoustic space, with one or more sound sources located within the space. The system was intended to fulfil the following requirements:

1. The sound sources and listener location within the space should be animated, so that any of the objects (sources or receiver) could be moved in real-time.
2. The subject should be given the illusion of the sound source(s) being localised in space with the correct direction and distance impression.
3. The direct sound source and some early reflections should be animated to give the correct impression of close reflective surfaces.
4. The absorption properties of the wall, floor and ceiling surfaces should be modelled.
5. The late reverberation should be processed to provide the correct spatial impression.

6. All configuration of the system should be possible from an external computer (such as a graphics workstation) so that the audio simulation can be linked to a graphical visualisation/simulation system.

The proposed auralisations within this thesis are aimed at being low cost and independent of hardware. This system is therefore unsuitable as it requires some dedicated hardware.

3.8. Commercial 3D audio API's

This section contains an explanation and appraisal of some commercially available APIs. They are examined with a view to facilitating real-time spatialisation of musically auralised sorting algorithms. As with 3D graphics, whenever there exists a substantial amount of processing to be executed, it is best performed directly in the hardware components of the system rather than employed in some form of software solution. Whether implemented in hardware or software there must exist some form of convention pertaining to the language that is to be used to implement the relevant functions. This is facilitated by the role of the API (Application Program Interface). There are several API's that exist within the commercial market that aim to facilitate the production of 3D spatial audio, the most famous of these is Microsoft's Direct3D API.

3.8.1. DirectSound3D – Microsoft Corporation

Initially, DirectSound was shipped as a sub-component of Microsoft's DirectX API. DirectSound supported basic WAVE file mixing alongside pitch control, volume and simple stereo placement. DirectSound also incorporated support for offloading the mixing, panning and pitch-shifting to external hardware, this is more commonly known as audio acceleration. Few commercially available soundcards support true audio acceleration due to the need for sound card designers to incorporate dedicated and expensive memory within the audio system. DirectSound3D was first shipped with DirectX 5.0 and was designed to provide a standard API for 3D sound production that developers could employ with any sound card. DirectSound3D would provide the basic

3D audio algorithm and permit acceleration of that algorithm in the same manner that standard stereo DirectSound can be accelerated. With DirectX 5.0 DirectSound supported sound cards that use third party 3D audio algorithms to accelerate DirectSound.

DirectSound3D works most effectively with headphones but employs further complex filtering to support open field stereophonic speakers, quadraphonic speakers and surround sound speakers. DirectSound3D uses virtual 3D space where the sound source is defined by x, y and z coordinates. Similarly, the listener is placed in virtual space by the same 3D vectored coordinate system. The listener also has a further parameter that defines the orientation of the head. DirectSound3D implements distance cues by attenuating or amplifying the sound source, it also employs some processing to yield a doppler effect changing the pitch of the sound source as it moves closer to the listener. DirectX can take advantage of many different types of hardware that can enhance the quality of the 3D audio production. This however, places much responsibility on the programmer. Microsoft makes no provision for such effects as environment modelling, reverberation and object occlusion but does now support Creative's [57] EAX standard. Many of these more complex features require the implementation of third party algorithms.

3.8.2. A3D – Aureal Technology

A3D was developed by Aureal Technology [9] as a third party API to be used with their own integrated circuits. Aureal developed the Vortex2 IC that accelerated the Aureal-2 standard through it's own A3D API. The specification of Aureal-2 includes all of the features of Microsoft's DirectSound3D algorithm with the addition of:

- Wall / object occlusion.
- Increased sampling rates.
- Underwater effects.
- Wavetracing technology.

The Wavetracing technology implemented by Aureal traces the various paths from the sound source to the listener as they bounce from, or are occluded by, the walls in the environment. Further to this, the definition of the wall / object material is permissible thereby defining the absorption and reflection of the surfaces.

Results from Aureal's extensive research have offered some scientific explanations of why real-time binaural audio technologies such as their own A3D are effective across a wide range of applications:

- **Binaural gain** – When an audio signal is played on top of white noise it will appear 6 to 8 dBs louder than if the signal were non-binaural. This indicates that identical audio contents can be more audible and intelligible in the binaural case as the human brain can localise and single out the signal while non-binaural signals get lost into the noise.

- **'Cocktail party effect'** – With monaural recording, the ability of the listener to focus upon one feature (or conversation) is considerably less than when recorded binaurally. This is because the audible components remain spatially separated and are subject to binaural gain.

- **Faster reaction time** – In an environment such as the cockpit of a jet fighter, where a lot of complex information is conveyed to the pilot, reaction time can be critical. Research suggests that audio information can be processed and reacted to more quickly when presented in a binaural format. The binaural signal does not only contain information about the nature of itself but also about its source.

- **Reduced listening fatigue** – Aureal suggest that listening fatigue may be reduced by employing binaural representations. Users that have to wear headphones for long periods of time are often subject to listening fatigue, this is due to the nature of monaural signals appearing to emit from within the listener's head.

- **Increased immersion and perception** – Binaural presentation offers a richer, more in-depth listening experience. Listeners often report the experience as being more immersive or of being of a higher quality.

A3D offers many features and reports to be based upon the worlds most advanced algorithms and HRTF measurement and compression techniques. However, the system does require some dedicated hardware.

3.8.3. Sensaura

Unlike Aureal, Sensaura [168] simply design audio processing techniques and then licence their technology to external integrated circuit manufacturers. The Sensaura technology consists of the four following components:

- **HRTF's** – These have been designed through Sensaura's own 'Digital Ear' technology. They have measured 1,111 HRTFs to cover one complete hemisphere and employ a cross-fading technique to switch from one filter to another thus reducing the 'glitch' noises that often appear.

- **Multidrive technology** – Both Aureal and Sensaura utilise complex cross-talk cancellation techniques to facilitate the playback of binaural signals over open-field stereophonic or quadraphonic speakers. With the four-speaker system, Sensaura have the ability to deliver HRTF based 3D audio to both front and rear speaker sets through their Multidrive technology.

- **Macro-FX technology** – Sensaura have incorporated their Macro-FX technology to allow sound sources to appear from locations that are closer to the user than other spatial sound systems would facilitate..

- **Environmental-FX technology** – This technology provides compatibility with Creative's EAX 1.0 standard and includes wall occlusions and object obstructions.

Sensaura have developed a clearly competent 3D audio technology yet it requires specific hardware for implementation. With the advent of cheap soundcards, this approach can now be a relatively inexpensive way of achieving 3D audio production on a desktop personal computer.

3.8.4. Other 3D audio API's

As previously mentioned, Sensaura have enabled compatibility with Creative's EAX 1.0 technology (Environmental Audio Technology). The main point about Creative's technology is that it does not employ the use of HRTFs. Instead, Creative have concentrated upon the 'secondary' cues that are produced by environmental effects. Creative states that any positional audio implementation is better carried out through the conventional surround-sound technique. Creative's initial Environmental Audio Technology implemented a simple feature that adjusted the reverberation of a sound source dependent upon its position within the environment. In later implementations they incorporated object obstruction and occlusion. They further added controllable variables pertaining to the early and late reverberations. Creative have clearly taken a different route to mainstream HRTF based 3D spatial audio systems and have not implemented the complex features necessary to produce realistic 3D representations.

3.9. Conclusion

It can be seen from the preceding investigative appraisal of 3D audio and its associated technologies that there are several approaches to creating 3D spatial audio systems. The trade-offs between them are based upon effectiveness, complexity and expense.

Simple stereophonic systems are limited to placing the sound source on an axis that runs through the listener's head from ear to ear. This is clearly the cheapest system to implement as stereophonic recording and playback systems are commonplace in today's technology. The main drawback however, is that it fails to produce any audio presentation that could even be remotely considered as three-dimensional.

Spatial audio systems (or enhanced or extended field systems) are the next stage of evolution from pure stereophony. This approach does not create a 3D environment but simply extends the sound field beyond the boundaries of the listener's head. The sound source still remains, as with conventional stereo, on the axis that runs between the listener's ears, the exception is that this stereo line is extended to place the sound source outside of the user's head. This is another case where implementation is relatively cost effective. The result however, is again far from a true 3D spatial audio presentation.

Surround-sound systems have been proven to be extremely effective and have been employed by many cinema theatres thus proving their commercial acceptance. They do remain, however, very expensive to implement requiring many speakers and a complex audio processing system with dedicated hardware. In terms of application at the human-computer interface, surround-sound is not really viable, the overspill of audio within a cooperative working environment would be too great to go unnoticed.

Although HRTFs are rapidly becoming the industry standard for implementing 3D audio algorithms they are not without their disadvantages. They require large overheads by demanding sets of measured HRTFs. Another, and more important problem is the geometrical differences that are present from human to human. There have been several suggestions to circumvent this issue ranging from measuring HRTFs based upon the statistical norm to measuring unique sets of HRTFs for each potential listener. In using the HRTFs based upon the geometrical measurements of the statistical norm the resulting audio experience will only translate effectively to a small percentage of the population. Creating several HRTF sets based upon geometrically categorised groups resolves this issue a little further but is still far from producing the perfect solution.

The only conceivable way of creating effective HRTF based 3D spatial audio presentations is to measure a unique set of HRTFs for each listener. The problem herein is that the storage space required becomes extremely demanding in order to accommodate such vast data. The adaptable solution that has been recently suggested is that of modelled HRTFs. Here the model contains several variables pertaining to the physical attributes of the target listener, once in place, these variables will yield the relevant set of HRTFs for subsequent processing. The main problem here is that an accurate and complete model does not yet exist, furthermore, each potential listener would require his or her physical measurements to be taken and processed.

It is clear that there are many issues still surrounding the effective use of HRTFs for 3D spatial audio presentation. To take this approach it is necessary in the first instance to obtain a set HRTFs, further to this it is necessary to obtain a system that will effectively implement these HRTFs. Some of the applications appraised in this chapter will produce acceptable 3D audio outputs but the most effective systems do require some lengthy work.

The approach that falls part way between cost-effectiveness and realism is the use of pure binaural recordings. They are, unlike HRTF based systems, inflexible in that once a sound has been binaurally recorded its position is fixed. Again the overheads can be high as creating an auditory scene may require the storage of large amounts of audio files. It is by far one of the cheapest solutions requiring a simple stereophonic digital recording device and a pair of small condenser microphones. It is possible to utilise a manikin head within which the microphones may be placed or take the much cheaper approach of placing them in the ear canals of a real human being. Again the resulting translation of the audio output will be dependent upon the physical similarities of the recording manikin/human and the listener.

For the application of 3D audio in the musical representation of algorithm state and execution in this thesis, it has been decided that the best approach to take would be the implementation of binaural recordings. The target result is not required to be exact and to translate precisely to all potential users. Furthermore, the information to be presented is to be well defined and finite, therefore only requiring a specific set of binaural recordings to be employed in creating the musical auditory scene thus minimising the storage overhead required.

Chapter 4

An analysis of sorting algorithms

The Concise Oxford Dictionary [178] defines an algorithm as:

> **algorithm** (*say* alga-ri*th*'m) *noun*
>
> *Maths:* a clearly defined sequence of operations for solving a particular mathematical problem.

4.1. Introduction

Sorting is concerned with the organisation of information into some form of sequential order to facilitate easier and faster information retrieval. It can be applied, for example, to a contact database where the information could be entered in any random order but requires to be ordered alphabetically to facilitate ordered retrieval. If the stored information were to contain names and addresses for example, the data could be reordered into alphabetical listings based upon name or address. This would permit a user to quickly retrieve the desired information as opposed to methodically searching through an unordered list of entries. There are many ways in which the reordering of the information could be implemented. Each method of reordering can be classed as an algorithm. Sorting is central to many tasks carried out on a computer, from database entries to file structures for example, to increase the efficiency of information retrieval rates. There has been a great deal of research which has yielded an interesting range of different algorithms. It is not always possible to say that one algorithm is better than another, as relative performance can vary depending on the type of data being sorted. In some situations, most of the data is in the correct order, with only a few items needing to be sorted. In other situations the data is in a random order and in others the data will tend to be in reverse order. Different algorithms will perform differently according to the organisation of the data being sorted. Some common algorithms are the *Exchange Sort*, the *Bubble Sort*, the *Selection Sort*, the *Insertion Sort* and *Quick Sort* algorithms. Some of these algorithms are easy to understand and simple to program whilst other are more complex. For a given number of data items, certain sorts always go through a set number of comparisons and exchanges, so their performance can be predicted.

4.2. Sorting algorithm summary

The more common algorithms are [183, 1]:

Exchange Sort.

In the Exchange Sort, every two numbers in the list are compared and swapped if the second number is less than the first, thus yielding an increasingly sorted ascending list.

Bubble Sort.

The idea is to make several passes through the list. On each pass, each pair of adjacent elements is compared. If they are in the wrong order, they are swapped. The sort completes after a pass with no swaps has been made.

Insertion Sort.

The Insertion Sort traverses the list, inserting each element into a second list in sorted order. Efficient implementation is achieved by quickly finding the correct position to insert the current element, and making sure that the insertion operation is inexpensive in resource usage. For instance, a binary search can be used to locate the insertion position.

Selection Sort.

A Selection Sort is very similar to an Exchange Sort. For a given list; the smallest element of the list is selected and swapped with the first element of the list. The Selection Sort is then performed on the remaining list.

Quick Sort.

The Quick Sort algorithm is a fast sort. To sort a list, it divides it into two sub-lists where the elements of the first list are all smaller than the elements of the second list. Then each of the smaller lists is recursively sorted.

Radix Exchange Sort.

This sort works by looking at the binary representation of each number in the list. It sorts based on whether a leading digit is 0 or 1, progressively going from left to right. This is similar to the Quick Sort, since essentially a pivot is chosen around which elements get sorted. For instance, if the data consisted of 4 bit numbers, the first pivot would be 1000, then the algorithm would recurse with the pivots 0100 and 1100, etc.

Heap Sort.

This is based on a data structure called a heap, which is a tree with the following properties: every leaf has height h or h-1, every leaf of height h is to the left of those of height h-1, and the value of each vertex is greater than the value of any of its descendants. This tree is represented in an array, where for each vertex at index i, its descendants are at index 2i and 2i+1. This sort works by swapping the largest element, which is the root of the tree, at index 1, with the last element of the array. This puts the largest element at the end. Then, the size of the tree is decreased by 1, leaving the largest element in its correct final position, and outside the tree. Finally, the tree is reconverted into a heap, with the largest element at the root, and this process is repeated until the heap is empty.

Shell Sort.

The list is sorted by shells of decreasing size. Say, for instance, that sizes of 8,5,3,2,1 are used. First, every list of every 8 elements is sorted, that is, those elements numbered [1,9,17,...], [2,10,18,...], ..., [8,16,24,...]. Then shells of size 5, 3, and 2 are sorted. Finally, the whole list (every 1 element) gets sorted. The elements within each shell are sorted by a Bubble Sort, although other sorts could be used, even the Shell Sort itself.

Bucket Sort. [199, 200].

In the Bucket Sort, the list is traversed, placing each element into its appropriate bucket (a container which only takes numbers in a given range, e.g. one bucket might take numbers between 10 and 20). Within each bucket, the contents can be sorted using any convenient method.

4.3. Further explanation of the more common algorithms

4.3.1. Exchange Sort

The Exchange Sort is performed by comparing every two numbers in the list and swapping them if the second number is less than the first, thus yielding a sorted ascending list. Below is the pseudo-code for the Exchange Sort algorithm:

```
Exchange Sort (Sorting the array A[size])
While not sorted
{
For index i = 1 up to i = (size-1)
Reset count=0.
    {
    While Not at end of list [(i + count) ≤ size]
        {
        Compare the element A[i]with the element A[i + count].
        If the current element is larger than the comparison element (A[i] > A[i +
        count]), swap them.
        Increment count.
        }
    }
}
```

Example:

Element	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Data	6	3	2	5	1	4	7	8	
1st pass	1	6	3	5	2	4	7	8	
2nd pass	1	2	6	5	3	4	7	8	
3rd pass	1	2	3	6	5	4	7	8	....

The current indexed element (at position 1) which is 6 is compared with the 3 (pos. 2) and swapped, the 3 then becomes our current indexed element and is compared with the 2 (pos. 3) and swapped. 2 then becomes our current indexed element, which is compared with the 5 (pos. 4), this is not swapped as the current indexed element is not larger than the comparison element. Then the 1 is considered (pos. 5) and this time a swap occurs, 1 is now the current indexed element and is compared to the 4, 7 and 8 (pos. 6, 7 and 8) where no swaps occur. The second pass now begins with the current indexed element being the element in position 2, which is compared with elements 3, 4, 5 and 6. This continues until the list is sorted or all the elements have been compared (N-1 passes).

4.3.2. Bubble Sort

Below is the pseudo-code for the Bubble Sort Algorithm:

```
Bubble Sort (Sorting the array A[size])
While not sorted
{
For index i = 2 up to i = size
    {
    Compare the element A[i]with the preceding element (A[i - 1]).
    If the element is smaller than the preceding one (A[i] < A[i - 1]), swap them.
    }
}
```

Example:

Element	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Data	4	6	1	8	7	5	3	2	
1st pass	4	1	6	7	5	3	2	8	
2nd pass	1	4	6	5	3	2	7	8	
3rd pass	1	4	5	3	2	6	7	8	....

The first two data items (4 and 6) are compared and the smaller one placed on the left-hand side and the larger one on the right hand side. The second and third items (6 and 1) are then compared and the smaller one placed on the left and so on. After all the data has been passed through once, the largest data item (8) will have "bubbled" through to the end of the list. At the end of the second pass, the second largest data item (7) will be in the second last position. For n data items, the process continues for n-1 passes, or until no exchanges are made in a single pass.

4.3.3. Insertion Sort

The basic step in this algorithm is to insert data into an ascending ordered sequence. Thus if the data 'D' is inserted in position i, then the data to the left of position i will be less than 'D', and the data to the right of position i will be greater than 'D'. Starting at the end of the list and working from right to left, all data on the right of the index is considered as the sub-list. This means that as the index decreases toward the start of the list, the sub-list increases in size. The data at the current index is 'rippled' through the sub-list via comparisons until it finds its correct position in the sub-list. Below is the pseudo-code of the algorithm.

```
Insertion Sort (Sorting the array A[size])
For index i = size down to i = 2
  { While before reach the end of cells
      {
      Compare the element A[i]with the preceding element (A[i - 1]).
      If the element is smaller than the preceding one (A[i] < A[i - 1]), swap them;
            else, go to the next element.
      }
  }
```

Example:

Element	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Data	4	6	1	8	7	5	3	2	
1st pass	4	6	1	8	7	5	2	3	
2nd pass	4	6	1	8	7	2	3	5	
3rd pass	4	6	1	8	2	3	5	7	....

The insertion sort starts with the last two elements and creates a correctly sorted sub-list, which in the example contains 2 and 3. It then looks at the next element (5) and inserts it into the sub-list in its correct position. It takes the next element (7) and does the same, continuing until the sub-list contains all the data.

4.3.4. Selection Sort.

Below is the pseudo-code for the Selection Sort:

```
Selection Sort (Sorting the Array A[Size])
For index = 1 to Size-1
    {
        small = index
        For count = index to Size
            {
                If A[count] < A[small] then small = count   {Find element number of smallest value in list}
            }
        Swap A[index] for A[small]
        {Swap current indexed value for smallest value in list}
    }
```

Example:

Element	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Data	4	6	1	8	7	5	3	2	
1st pass	1	6	4	8	7	5	3	2	
2nd pass	1	2	4	8	7	5	3	6	
3rd pass	1	2	3	8	7	5	4	6	....

The selection sort marks the first element (4). It then goes through the remaining data to find the smallest number (1). It swaps this with the first element and the smallest element is now in its correct position. It then marks the second element (6) and looks through the remaining data for the next smallest number (2). These two numbers are then swapped. This process continues until n-1 passes have been made.

4.3.5. Quick Sort

This algorithm partitions the array into two parts by moving a pivot into its correct position, so that items to the pivot's left are smaller than the pivot, and the items to the right are bigger. The algorithm is then called recursively so that it will partition the two subordinate arrays on either side of the pivot until the entire array is sorted. Below is the pseudo-code of the Quick Sort:.

```
Quick Sort (Sorting array A[size])
  While Low is less than High
  {
        Choose Pivot as the element at position A[Low]
        While A[High] is greater than Pivot, decrement High; else move A[High] to A[Low]
        While A[Low] is less than Pivot, increment Low; else move A[Low] to A[High]
  }
  Move Pivot into A[High], see Pivot position as High
  If Low is less than Pivot point, recursively call Quick Sort with Low = Low, High = Pivot point - 1
  If High is greater than Pivot point, recursively call Quick Sort with Low = Pivot point + 1, High = High.
```

Example:

Element	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Data	4	6	1	8	7	5	3	2	
1st pass	1	2	6	8	7	5	3	4	
2nd pass	1	2	3	4	7	5	8	6	
3rd pass	1	2	3	4	5	6	8	7	
4th pass	1	2	3	4	5	6	7	8	sorted.

The Quick Sort takes the last element (2) and places it such that all the numbers in the left sub-list are smaller and all the numbers in the right sub-list are bigger. It then Quick Sorts the left sub-list and then Quick Sorts the right sub-list. This is a recursive algorithm, since it is defined in terms of itself. This reduces the complexity of programming it. In this implementation the pivot is chosen at random (by picking the last element in the list, whatever it may be), but if certain patterns of sorting are required then a pivot is chosen through more selective means. For instance, if two sub-lists are required in the first pass to be of approximately equal size then a pivot would be selected that would split the list roughly in half.

4.3.6. Radix Sort

Unlike most other sorting algorithms, the Radix Sort does not involve comparison between the items being sorted. Instead, Radix Sort shuffles the items into small bins, then collects the bins and repeats the process until the array is sorted. The efficient operation of the Radix Sort lies in finding the best key to shuffle the items. For integer data, the key is each individual digit. In a group of data, there can be up to ten bins for each digit (0 - 9). Thus each individual digit of each data is isolated and placed into the corresponding bin. At the start, the least significant digit is chosen and the algorithm works its way up to the most significant digit. Below is the pseudo-code of the Radix Sort:

```
Radix Sort (Sorting array A[size])
  Create all of the bins.
  From the least significant digit to the most significant digit
  {
  For each element (from the first to the last)
      {
      Isolate the value of the significant digit.
      Store the element in the bin with the matching significant digit value.
      }
  For each bin (from the first to the last)
      {
      Retrieve all of the elements and store them back into the array.
      }
  }  Destroy all of the bins.
```

4.3.7. Heap Sort

By viewing the array as a complete binary tree, the Heap Sort transforms such a binary tree into a heap. This algorithm does not require overheads and is not recursive. The algorithm basically follows the following steps:

1. The complete binary tree (actually an array) is sorted so that it becomes a max-heap, thus the first element is always the biggest element.
2. Since exactly the opposite is required (the last element should be the biggest instead), the first element and the last element are swapped.
3. Now the array has to be re-sorted (except the last element), so that the first element is again the biggest.
4. The second step is repeated, so that first element is swapped with current last element.
5. Steps 2 and 3 are repeated so that all the elements are sorted.

Such a strategy takes advantage of a binary tree. Every time an element is moved, it is moved to its current position's child. Thus it moves a greater distance than the Insertion Sort. Below is the pseudo-code of the Heap Sort:

```
Heap Sort (Sorting array A[size])
For each parent node,
  {
      if there is any child node, we compare it with bigger child. If the parent
      is less, we walk down the parent until none of its new children nodes are greater.
  }
  While we do not reach the first cell,
  {
      swap the first cell with the last cell.
      Change the last cell index to the cell preceding the last cell.
      Walk down the first cell until none of its new children nodes are greater.
  }
```

4.3.8. Shell Sort

This sorting algorithm was conceived by D. L. Shell (hence the name), and was inspired by the Insertion Sort's ability to work quickly on an array that is almost in order. It is also called a 'diminishing increment' sort. Unlike the Insertion Sort, the Shell Sort does not sort the entire array at once. Instead, it divides the array into non-contiguous segments, which are separately sorted by using an Insertion Sort. Once all of the segments are sorted, the Shell Sort re-divides the array into fewer segments and repeats the algorithm until the number of segments equals one, then the array is sorted.

There are two advantages of the Shell Sort over the Insertion Sort. When the swap occurs in a non-contiguous segment, the swap moves the item over a greater distance within the overall array. The Insertion Sort only moves the item one position at a time. This means that in the Shell Sort, the items being swapped are more likely to be closer to their final position than with the Insertion Sort. Since the items are more likely to be closer to their final positions, the array itself becomes partially sorted. Thus when the segment number equals one, and the Shell Sort is performing basically the Insertion Sort, it will be able to work very fast, since the Insertion Sort is quick when the array is almost in order. There are variations of the Shell Sort depending on the method of arranging segments. The "2X" method determines the number of segments by dividing the number of cells by two (integer division), so that in the first round each segment will have mostly two cells. After the first round, the number of segments is decreased by dividing them by two again. This is repeated until there is one segment left. Below is the Shell Sort's pseudo-code:

```
Shell Sort (Sorting the array A[size])
Determine the number of segments by dividing the number of cells by two.
  While the number of segments are greater than zero
  {
      For each segment, we do an Insertion Sort.
      Divide the number of segments by two.
  }
```

4.3.9. Bucket Sort

This algorithm partitions the array into two parts by moving a pivot into its correct position, so that items to the pivot's left are smaller than the pivot, and the items to the right are bigger. Once the list has been sorted into two buckets, any convenient algorithm may be employed to complete the sorting of the contents of each bucket. Below is the pseudo-code for an instance of the Quick Sort employing the Bubble Sort as the secondary sorting algorithm:

```
Quick Sort (Sorting array A[size])
  Choose pivot as mid-point
  Reset Current pointer
  Reset Sub1 pointer and Sub2 pointer
   While not at end of list
   {
  If A[Current pointer]<=pivot
        {
        A_Sublist1[Sub1 pointer]=A[Current pointer]
        Increment Sub1 pointer
        }
   else
        {
        A_Sublist2[Sub2 pointer]=A[Current pointer]
        Increment Sub2 pointer
        }
  Increment Current pointer
   }
Call Bubble Sort for A_Sublist1 and A_Sublist2
```

Example:				**pivot = 4**					
Element	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Data	4	6	1	8	7	5	3	2	
1st pass	{4	1	3	2}	{6	8	7	5}	
2nd pass	{1	3	2	4}	{6	7	5	8}	
3rd pass	{1	2	3	4}	{6	5	7	8}	
4th pass	{1	2	3	4}	{5	6	7	8}	sorted.

The Bucket Sort selects the pivot as the mid-point value in the list which is 4. It the traverses the list comparing each element to the pivot. Elements with a value of less than or equal to the pivot are placed into the left hand side of the list (A_Sublist1) while elements of value greater than the pivot are placed into the right hand side of the list (A_Sublist2). Hence, on the first pass the 4 is compared to the pivot of 4 and placed into the left hand sub-list, the 6 is then compared to the pivot of 4 and placed into the right hand sub-list. This continues until the entire contents of the list have been sorted around a pivot into two sub-lists. The Bubble Sort is then applied to each of these sub-lists. In A_Sublist1, the first element (4) is compared to the following element (1) and swapped. It is then compared to the third element (3) and swapped. The last compare on this pass of this sub-list is then performed against the fourth element (2) and swapped into its final correct location. The same action is repeatedly performed on each of the sub-lists until the entire list is eventually sorted.

4.4. Algorithm selection for auralisation

As stated in Chapter 1, the main purpose of this thesis is to examine how relatively inexpensive 3-D sound techniques can be used to improve disambiguation of musically auralised sorting algorithms. To reiterate, the emphasis on sorting algorithms is due their diverse range of events, sorting natures and data. Many other information sources exist that could be well suited to auralisation. However, this thesis is not concerned with defining which types of information sources are best suited to auralisation. It is more concerned with using sorting algorithms as a vehicle for preliminary experimentation of communicating information via spatially enhanced music. Furthermore, this thesis also sets out to determine which types of information within sorting algorithms are more amenable to auralisation.

For the purpose of musical algorithm auralisation it is necessary to look at the nature of the sorting algorithms. For simplicity and the ability to draw comparisons, it will be assumed that each algorithm will aim to sort the list into ascending order. It will also be assumed that the target users will have little or no prior knowledge about the nature and function of sorting algorithms, this is in order to ensure that all users possess the same amount of prior training. Certain algorithms will posses particular characteristics that make them more appropriate for musical auralisation than others. Such characteristics will be defined by the available information present within the algorithm execution. Some algorithms will possess characteristics that will be more readily represented by musical metaphors. An example of this would be that a musical triad denoting the occurrence of a swap in a Bubble Sort would translate with greater meaning than a single note representing a pivot in a Quick Sort. This is because the musical triad 'suggests' the occurrence of a swap more than a single note 'suggests' the presence of a pivot point. The swap structure lends itself more readily to a musical metaphor than a pivot does. It is therefore important to identify the key features of each algorithm. The list below identifies the prominent characteristics of each of the algorithms:

- Exchange Sort - This algorithm gradually sorts the list starting from the left by **swapping** elements. This algorithm is **iterative** and as it repeats its sorting cycle the sorted elements of the list grow to the **right** until the entire list becomes fully sorted when the algorithm can **terminate.**

- Bubble Sort - This algorithm progressively sorts the list from the right. As large elements are 'rippled' through the list to the right on each **iteration** by a process of **swapping** neighbouring elements, the list of sorted elements gradually grows to the **left** until the entire list becomes fully sorted when the algorithm can **terminate.**

- Insertion Sort - This algorithm, like the Exchange Sort, gradually sorts the list starting from **left.** This algorithm is **iterative** and as it repeats its sorting cycle by **swapping** elements, the sorted elements of the list grow to the **right** until the entire list becomes fully sorted when the algorithm can **terminate.**

- Selection Sort - This algorithm, like the Exchange Sort and Insertion Sort, gradually sorts the list starting from the **left.** Again this algorithm is **iterative** and as it repeats its sorting cycle by swapping pairs of **elements** the sorted elements of the list grow to the **right** until the entire list becomes fully sorted and the algorithm can **terminate.**

- Quick Sort - This algorithm sorts the list into 'buckets' by deciding their **destination** about a **pivot.** The resulting list shape progression yields a **grouping** characteristic where the groups become smaller yet their number grows in size with each **iteration.** When the sub-arrays or 'bucket' sizes become single then the list is deemed to be sorted and the algorithm can **terminate.**

- Radix Exchange Sort - As this algorithm operates in the same manner as the Quick Sort, it also yields a **grouping** characteristic and shares the **placement, iteration** and **termination** characteristics.

- Heap Sort - This algorithm **iteratively** sorts the list from the right. As large elements are **sent** to the end of the list to the right, the list of sorted elements gradually grows to the **left** until the entire list becomes fully sorted and the algorithm can **terminate.**

- Shell Sort - The characteristics pertaining to the shape progression of the list are once again dependent upon the sub-algorithm that is employed.

- Bucket Sort - This algorithm does not fully sort a list and therefore requires a sub-algorithm to complete the sort. The initial characteristic that the algorithm yields is one of a **grouping** nature by **placement** of elements around a **pivot,** the subsequent characteristics are then dependent upon the sub-algorithm employed.

Features that are common to all of the above algorithms can be identified by working through the examples given earlier in this chapter. Each shares the following common steps:

1. **Check** the state of the list to see if it is sorted.
2. **Pass** through list **manipulating** the elements.
3. **Check** the state of list and **terminate** if sorted or **loop** back to step 1 if not sorted.

The characteristics from all of the above algorithms can now be identified:

- **Checking** the list state.
- **Progression** of list shape – sorting to the left, to the right, of into groups.
- **Manipulation** of elements during a **sorting pass** – **swapping** elements or **placement** of elements into sub-arrays based around **pivot** points.
- **Iteration** – denotes the amount of passes the algorithm has made through the list.
- **Termination** – denoting the successful completion of a sort.

The algorithms described above can be further categorised by their sorting **progression** characteristics, the nature of the list progression:

- **Algorithms that sort from the left side of the list-**
    - Exchange Sort.
    - Selection Sort.
    - Insertion Sort.

- **Algorithms that sort from the right side of the list-**
    - Bubble Sort.
    - Heap Sort.

- **Algorithms that sort into groups and sub-groups-**
    - Quick Sort.
    - Radix Exchange Sort.
    - Postman's Sort.
- **Algorithms that sort dependent upon implemented sub-algorithms-**
    - Bucket Sort.

  - Merge Sort.
  - Shell Sort.

- **Algorithms that have no sequential nature and might not translate well musically -**
  - Tree Sort.

It would be useful to auralise algorithms with different sorting characteristics. It would therefore be a logical step to use one algorithm from each of the previously mentioned groups. The algorithms listed below have therefore been chosen to represent each of the sorting characteristics with the exception of the non-sequential characteristic that might not translate well musically. The exception is that two algorithms with the 'sort from the left' characteristic have been chosen. This is due to the different ways in which this characteristic manifests itself. The Selection Sort causes a single swap on each pass whereas the Exchange Sort can cause multiple swaps when sorting and may therefore create confusion for the listener. Auralising both of these algorithms and comparing the resulting experimental data will show if this difference in sorting nature has any effect on users' perception and understanding of the musical auralisations.

The chosen algorithms for auralisation are:

1. **Selection Sort** – Sorts from the left hand side.
2. **Exchange Sort** – Sorts from the left hand side.
3. **Bubble Sort** – Sorts from the right hand side.
4. **Quick Sort** – Sorts into groups.
5. **Bucket Sort** – Sorts dependent upon sub-algorithms.

Given these characteristics it would also be useful to convey algorithms that sort from the middle-out and sort from the outside to middle. This can be achieved by combining the aforementioned algorithms and essentially implementing different versions of the Bucket Sort algorithm.

A Bucket Sort algorithm can be employed to achieve a sorting nature that causes the list to be gradually ordered from the centre of the list growing out towards the ends of the list. The Quick Sort algorithm can be applied for one cycle only to split the list into two sub-lists. The resulting two sub-lists can then be sorted with the Bubble Sort algorithm and either the Selection Sort or Exchange Sort respectively. Because the Bubble Sort sorts from the right hand side and the Selection / Exchange Sort sorts from the left hand side, the resulting list appears to progressively sort from the middle-out when combined. Similarly, the Bucket Sort algorithm can also be used to achieve a sorting nature that causes the list to be gradually ordered from the ends of the list towards the centre of the list. The Quick Sort algorithm can be applied for one cycle only to split the list into two sub-lists. The resulting two sub-lists can then be sorted with either the Selection Sort or Exchange Sort and Bubble Sort respectively. Again, because the Selection / Exchange Sort sorts from the left hand side and the Bubble Sort sorts from the right hand side, the resulting list appears to progressively sort from the outside to the middle when combined.

4.5. Conclusion

The final list of chosen algorithms for auralisation can now be summarised as follows:

1. **Bubble Sort** – Sorts from the right hand side.
2. **Selection Sort** – Sorts from the left hand side.
3. **Exchange Sort** – Sorts from the left hand side.
4. **Quick Sort** – Sorts into groups.
5. **Bucket 'Inside-Out' Sort** – Quick Sort + (Bubble Sort + Selection Sort). Sorts from the middle to the outside.
6. **Bucket 'Outside-In' Sort** – Quick Sort + (Selection Sort + Bubble Sort). Sorts from the outside to the middle.

The characteristics from all chosen algorithms were identified as:

- **Checking** the list state – passing through the list testing each element's placement in relation to all other elements in the list based upon a desired 'sorted' state.

- **Progression** of list shape – sorting to the left, to the right, of into groups. The progression of the state of the list of numbers due to the algorithm's sorting nature. This is concerned with the 'evolution' and presence of certain features within the list that provide information about its state. It has been identified that certain algorithm sort from the left hand side while others sort from the right hand side. Other algorithms have been shown to sort by 'segmentation' or 'grouping' while others sort from the inside-out or outside-in.

- **Manipulation** of elements during a **sorting pass** – **swapping** elements or **placement** of elements into sub-arrays based around **pivot** points. The Bubble, Exchange, Selection and Insertion Sort algorithms manipulate the data by swapping pairs of elements. As previously mentioned, a musical swapping metaphor could be employed here to denote the occurrence of this type of data manipulation. In contrast, the Quick Sort, and both Bucket Sort algorithms on the first pass, sort the data into sub-arrays by comparing the current data to a predefined pivot value. The representation of this characteristic in the musical domain is not as simple as the swap structure mentioned earlier. The visualisation of sub-arrays and pivot points lends itself more readily to mapping into a spatial domain. The use of timbre and placement could play a key role here in order to represent this characteristic. Such an approach could exploit the spatial nature of timbre placement to provide a metaphor for a central pivot, while lesser values would be sent to a left hand sub-array and greater than of equal to values would be sent to a right hand sub-array.

- **Iteration** – denotes the amount of passes the algorithm has made through the list. This feature would require the mapping of a 'control' type event to represent the iteration count.

- **Termination** – denoting the successful completion of a sort. Would require a 'control' type metaphor to denote that the algorithm has successfully sorted the list into the desired order.

This chapter has identified the key features and characteristics of sorting algorithms that might be amenable to musical auralisation. It is now important to determine how each of these features translates for each of the algorithms. The following chapters document experiments that are aimed at understanding the effect that musical training has on the perception and understanding of sorting algorithm auralisations. Preliminary experiments will be carried to test the basic building blocks of representing algorithms musically, such as pitch perception, shape perception, list state perception and list shape progression perception.

Chapter 5

Preliminary experimental work

5.1. Introduction

This chapter documents a set of initial experiments carried out in order to obtain some understanding of how an average person (i.e. non musically educated) perceives sequences of notes, shapes of musical tonal sequences with and without musical timing, musically auralised list state, musically auralised list shape progression and musically auralised data manipulation. These tests from the building blocks for sorting algorithm auralisation and are based upon the information attributes highlighted in Chapter 4. The results, when taken with existing guidelines on the use of sound in interfaces, are used as the basis for the design of experiments that use music to communicate algorithm state and execution. The empirical evaluation of these musical auralisations should provide a preliminary understanding of how useful music might be for communicating information to assist algorithm understanding through algorithm auralisation. Supporting statistical data, all raw data and all stimuli are given on the accompanying CD.

5.2 Research approach

5.2.1 Musical structure and understanding

There are many different types of musical structures that can be used to communicate information. At the basic level there are single notes (or a short series of single notes). Such single notes can be used to alert a user about a particular event but in order to communicate more complex information it is necessary to take advantage of higher structures in music that involve a number of other properties such as pitch, timbre, rhythm and harmony. Earcons [18] are examples of simple musical structures that communicate information and experimentally derived guidelines are available for using such structures effectively (see Section 2.4.3). At a higher level, music is characterised by structures such as Major and Minor scales, tunes, complex rhythms, timbre combinations and harmony. Most of these structural devices can be used both in works for large sets of instruments or for solo instruments. The more complex the music becomes, usually the more the emphasis on higher level structures is required to hold the work together.

It is currently not clear as to how much information listeners with no special musical ability can comprehend. The important research question is - to what level of complexity can we utilise musical structures in HCI without losing comprehension in the average user? To answer this question we need to determine the musical abilities of the average listener. Examples of relevant questions pertaining to the perception of musical auralisation include:

- How accurately can the average user identify musical tones?
- Can users distinguish note sequences?
- Can users visualise the shapes of tonal sequences?
- Can users comprehend patterns of tones that denote the presence of a structure?
- At what level can users comprehend rhythms and tunes?
- To what accuracy can users identify and distinguish different timbres?
- How useful is timbre placement in the stereophonic field?
- Does musical training have any effect on the each of the tests?

Some musical structures may be understood with little or no training. For example, a siren consisting of two-tone repetition to indicate an error is an intuitive representation that indicates urgent attention is required. This is, of course, culturally biased as the users requires prior knowledge of the meaning of a siren in order to recognise it as a warning sign. Rigas [160] calls these 'self explanatory messages' and are essentially auditory icons. Other messages may require learning such as those employed in Earcons. Rigas [160] calls these 'trained messages'. Interfaces may need to use both types but a preference of the 'self explanatory messages' will make the interface easier to use. To examine the questions posed above, four experimental procedures were designed and carried out. These experiments investigated some key list properties and manipulations of lists and how they might be perceived aurally.

Firstly, experiments were performed to determine if users (both trained and untrained) can perceive different pitch intervals.

Secondly, experiments were carried out using short pitch sequences in order to understand how subjects perceive the shape of tonal patterns. These experiments are important in determining whether or not subjects can comprehend the contour of the sequence. If so, it would indicate that they are capable of understanding patterns of numerical data.

Thirdly, experiments were carried out on pitch sequences that are essentially in a sequential order with only one or two notes being incorrectly placed. If users are capable of identifying these 'out of order' notes then it would indicate that they are capable of pinpointing erroneous elements in numerical data lists.

Fourthly, experiments were carried out using pitch sequences as above, but with addition of a second timbre to denote the manipulation of the incorrectly placed data elements. If users can identify the erroneous elements (as above) and can also identify the manipulation of such elements then it would indicate that they are capable of comprehending list manipulation, which is the basis of sorting algorithms.

The manipulations in the above experiments represent some of the core transformations used in sorting algorithms. The results of the experiments will enable us to verify and understand how listeners perceive and process the following:

- Ordered and non-ordered pitch ranges.
- Rhythm in combination with pitch.
- Temporal arrangements and pitch comparisons between one or two instruments.
- How far a pattern of what the algorithm does can be understood without the listener knowing its detailed processing.
- The abstract development of mental models of current list states.

The results will enhance our understanding of how users mentally react to, and process musical stimuli. They will further highlight the feasibility of conveying information about algorithm state and execution via music.

5.2.2 Tools used

The experiments in this section have been implemented on a personal computer equipped with a standard soundcard. The system can be used with both an external MIDI compatible multi-timbral synthesiser or the internal sound set of the sound card. Wave Table synthesis is normally used as the quality of the produced timbres is much closer to that of real instruments.

5.2.3 Subjects and feedback

In order to classify the musical ability of the test subjects, information related to the musical experience, interest and exposure of each listener was gathered. Edwards developed an interactive musical ability test entitled MAT (Musical Ability Test) [86] that required considerable time and effort on the part of the test subject. This thesis is concerned with understanding what effect musical training and exposure has on the above described tests. It does not concern itself with the natural or actual ability of the test subjects but relies more on history and self classification. The resulting individual scores were placed into a scale ranging from 1 to 6. This information was obtained using a questionnaire containing questions determining listeners' interest in music on a scale one to six, their ability to play a musical instrument on a scale of one to five and their singing ability on a scale of one to six. The questionnaire is based upon questionnaires used by Rigas [160], Alty [3] and Vickers [182]. The questionnaire is given in full in Appendix B.

The received information was used to determine a users' overall musical ability classification as follows:

- 1 – Absolutely no musical ability.
- 2 – Little to average musical ability.
- 3 – Average to greater than average musical ability.
- 4 – Greater than average musical ability.
- 5 – Greater than average to exceptionally high musical ability.
- 6 – Exceptionally high musical ability.

It was found that in all subsequent experiments, listeners with the ability extremes of 1 or 6 were never encountered. All of the test subjects fell into the scale of 2 to 5 and were therefore split into two classifications. Those with a score of 2 were classed as having little to average musical ability and those with a score of 3, 4 or 5 were classed as having greater than average to high musical ability. These two sub-groups will in future be referred to as 'non-musical' and 'musical' listeners respectively. User feedback was gathered before and during the experiments. Users then answered questions on a form in response to musical stimuli.

5.3. Pitch perception experiments

5.3.1. Experiment construction

In this set of experiments thirty subjects were asked to listen to pairs of musical notes and determine their position within a bounded diatonic[1] scale. The timbre employed was an acoustic grand piano, which was placed in the centre of the stereophonic field with no reverberation or chorus added. The bounded diatonic scale started at 'Middle C' and ascended by one octave (eight notes). Each of the note pairs were played within this scale. In order to create a context, the scale was first played once and each note pair was repeated three times. The time interval between the termination of the first note and the initiation of the second note was zero and the time interval between repetitions was 2 seconds. In some cases the second note in the pair was higher than the first, in other cases the note pairs were identical and in some cases the second note was lower in pitch than the first note. Subjects were told that each of the eight notes within the bounded scale were mapped to the numbers one to eight. Upon listening to each pair, the subjects were asked to write down the numerical values of the notes. There was also the option of writing down the initial note value and the difference between the two notes. This is to allow for listeners that may not perceive the two notes absolutely and distinctly but hear them relative to each other. An example of the scale and one pair of notes is shown in the Figure 5.1.

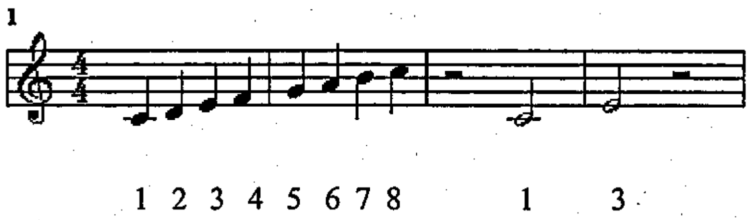



Figure 5.1 – Pitch test scale and note pair.

This diagram in Figure 5.1 was both shown and played to listeners. A further three demonstrations with answers were given, but this time no visual representation was present. Ten of these pitch pair tests were then carried on each of the thirty subjects. The full workbook is given in Appendix C.

5.3.2. Results and analysis

Figure 5.2 shows the musical ability distribution of the group of thirty test subjects. Of this test group, 17 have a musical ability score of 2 and 13 have a musical ability score of between 3 and 5. Therefore the test group consists of 17 'non-musical' listeners and 13 'musical' listeners. It must be noted here that the musical ability classification were

---

[1] Diatonic, involving only notes proper to the prevailing key without chromatic alteration.

derived such the difference between abilities 2 and 3 is greater than the difference between abilities 3 and 4. This facilitates the classification of the 'non-musical' group as abilities 1 and 2, and the 'musical' group as abilities 3, 4, 5 and 6. This group are termed 'Group 1' and consist of 17 university students from University College London and 13 non-students. Of the whole group, 9 are male and 21 are female.

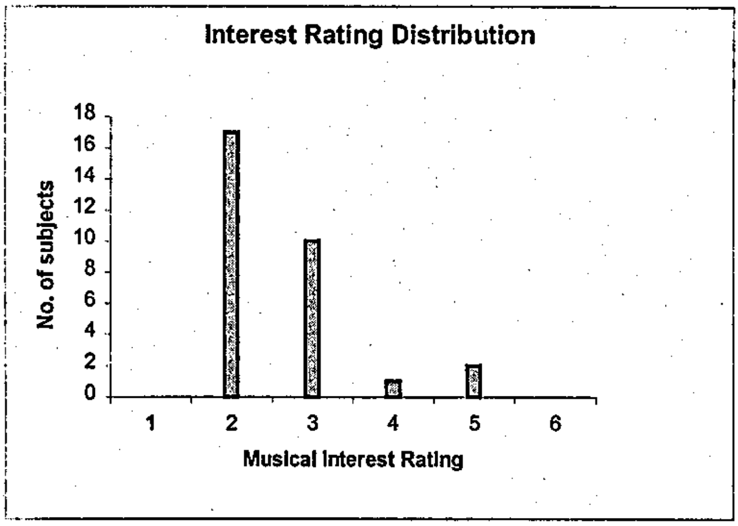


Figure 5.2 – Musical interest rating for pitch perception test subjects.

Figure 5.3 shows the users' perception of each of the tones separately. The plot indicates the accuracy of each 'absolute' tone within the bounded diatonic single octave scale. Thus the results have been analysed as twenty (ten pairs) individual notes.

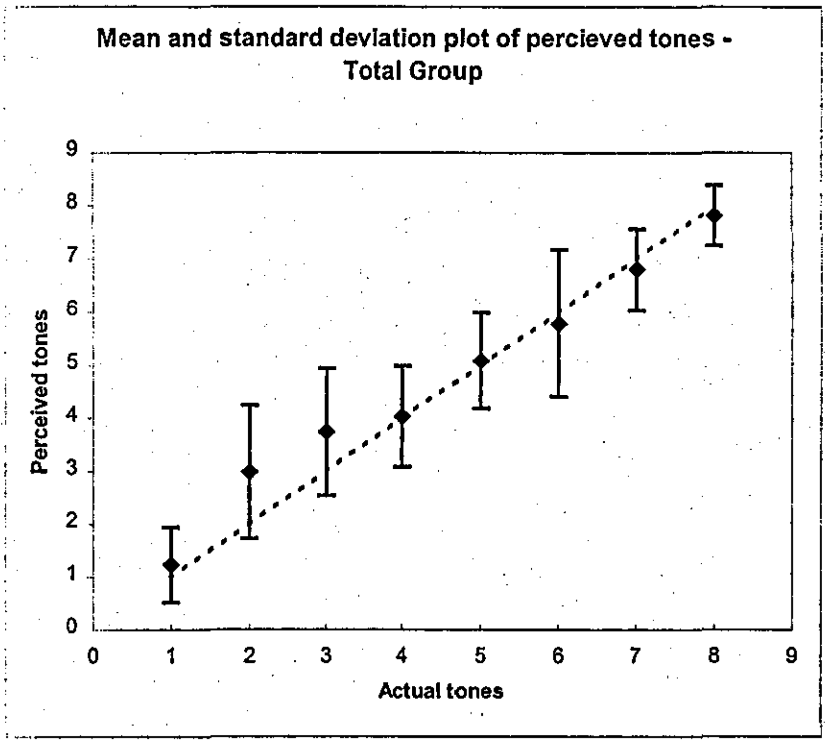


Figure 5.3 – Perceived tone accuracy for pitch perception.

Note	1	2	3	4	5	6	7	8
Mean	1.228571	2.983333	3.733333	4.033333	5.088889	5.783333	6.8	7.816667
S.D	0.708893	1.255384	1.201532	0.956098	0.907453	1.378917	0.761124	0.567231

Figure 5.4 – Table of perceived tone accuracy for pitch perception.

Figure 5.4 indicates the mean perception for each of the notes, the standard deviation and the high and low boundaries. It can be seen that notes that fall close to the boundaries of the scale are identified with greater accuracy than those that appear closer to the middle of the scale [3]. This is because the scale that provides the boundaries gives fixed points that the user can more readily recall. The middle of the scale has little boundary information and can create an area of ambiguity. Overall, the group performed well but it is important to split up the test subjects into their musical classification groups in order to understand how the 'non-musical' and 'musical' groups performed. Figure 5.5 indicates the accuracy of the 'non-musical' compared to the 'musical' group.

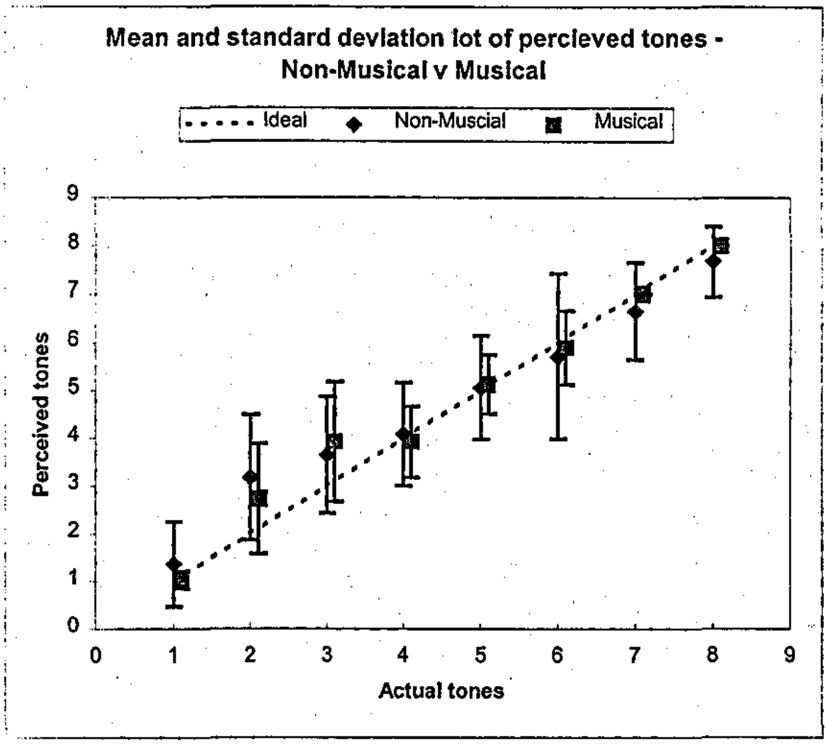


Figure 5.5 – Perceived tone accuracy for pitch perception – non-musical v. musical.

Musical	Note 1	Note 2	Note 3	Note 4	Note 5	Note 6	Note 7	Note 8
Mean	1.032967	2.730769	3.923077	3.923077	5.128205	5.884615	7	8
S.D	0.17954	1.150919	1.255756	0.744208	0.614709	0.765607	0	0

Non-Musical	Note 1	Note 2	Note 3	Note 4	Note 5	Note 6	Note 7	Note 8
Mean	1.361345	3.176471	3.647059	4.088235	5.058824	5.705882	6.647059	7.676471
S.D	0.889973	1.313579	1.221739	1.083419	1.084652	1.714986	0.996317	0.726994

Figure 5.6 – Table of perceived tone accuracy for pitch perception, musical and non-musical.

The data in Figure 5.6 suggest that the accuracy of the 'musical' listeners appears to be greater than that of the 'non-musical' group. Again the occurrence of inaccuracy appears close to the middle of the scale where perception seems to be most ambiguous.

Figure 5.6 also shows that the non-musical test subjects have the greatest influence over the performance of the entire group. Figure 5.8b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the differences of perceived notes from the true notes. This test has been employed due to fact that the assumptions required for parametric testing were not satisfied. The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving musical notes.

H_1 : The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving musical notes.

	NOTE1	NOTE2	NOTE	NOTE	NOTE	NOTE	NOTE	NOTE
Mann-Whitney U	29.500	81.000	99.000	101.500	87.500	68.500	52.000	65.000
Wilcoxon W	120.500	172.000	190.000	192.500	178.500	159.500	143.000	156.000
Z	-3.59	-1.27	-.514	-.405	-1.04	-1.887	-3.04	-2.57
Asymp.[2] Sig. (1-tailed)	.000	.102	.305	.341	.149	.029	.001	.005

Figure 5.8b – Table of test statistics for each perceived note, 'non-musical' v. 'musical'.

As previously noted the 'musical' subjects performed extremely well, whereas now it can be seen that the 'non-musical' subjects perform with greater inaccuracy particularly at the boundaries. The null hypothesis can be rejected at the 5% level for note 6, at the 1% level for note 8 and at the 0.1% level for notes 1 and 7. For the remaining notes (2, 3, 4 and 5) there is no significant difference between the 'non-musical' and 'musical' test groups and we cannot reject the null hypothesis. This data suggest that when notes are played close to the boundaries of the context scale the 'musical' test group perform significantly better than the 'non-musical' test group.

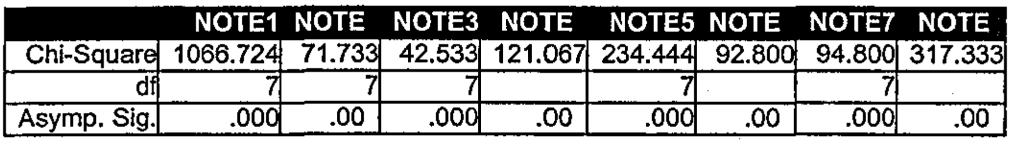
---

[2] Asypm. The significance level based on the asymptotic distribution of a test statistic.

The data given in Figure 5.8c shows the results of a Chi-Squared test applied to the results obtained for pitch identification. The hypotheses are:

H_0 : Users are not capable of identifying pitch. In particular, they are not capable of understanding musically represented numerical values.

H_1 : Users are capable of identifying pitch. In particular, they are capable of understanding musically represented numerical values.



	NOTE1	NOTE	NOTE3	NOTE	NOTE5	NOTE	NOTE7	NOTE
Chi-Square	1066.724	71.733	42.533	121.067	234.444	92.800	94.800	317.333
df	7	7	7		7		7	
Asymp. Sig.	.000	.00	.000	.00	.000	.00	.000	.00

Figure 5.8c – Table of test statistics for each perceived note.

These data show that the probability of obtaining the scores gathered from the users for the pitch tests at random are extremely low. From these data the null hypothesis can be confidently rejected concluding that users are capable of identifying pitch and in particular that they are capable of identifying musically represented numerical values. This is encouraging since it suggests that users might be capable of understanding musically auralised sorting algorithm lists.

The second feature of this experiment that warrants some investigation is how the test groups performed when the results are analysed as relative pitch tests as opposed to absolute pitch tests. Here, the data that are evaluated are the perceived differences between the two notes and not how accurately they are placed within the scale. The intervals played to the listeners ranged from 1 to 7 with the omission of intervals of 2 due to the constraints of the absolute pitches used in the previous test. Figure 5.9 shows how the entire group of test subjects performed as a whole.

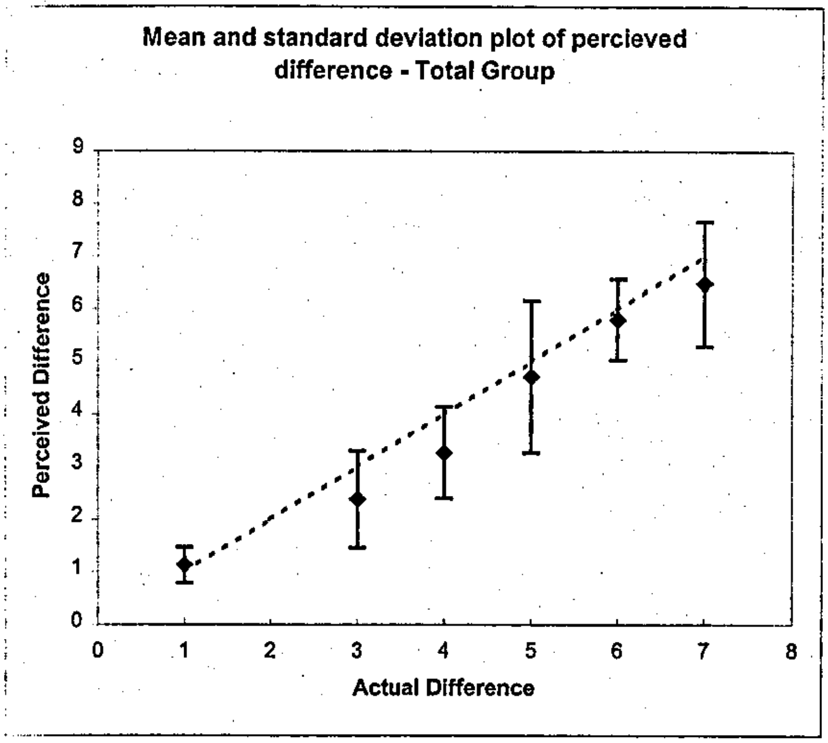


Figure 5.9 – Perceived tonal interval.

Note	1	2	3	4	5	6	7
Mean	1.133333	-	2.383333	3.266667	4.716667	5.8	6.483333
S.D	0.342803	-	0.922261	0.868345	1.450794	0.761124	1.185958

Figure 5.10 – Table of perceived tonal interval.

Figures 5.9 and 5.10 show that the greatest accuracy occurs when the actual tonal difference is least. The greatest inaccuracy occurs when the difference between the two tones is large. This indicates that it is easier to estimate small interval differences as opposed to large interval differences. Again it is important to divide the data into two groups in order to better understand how the 'musical' and 'non-musical' sub-groups perform.

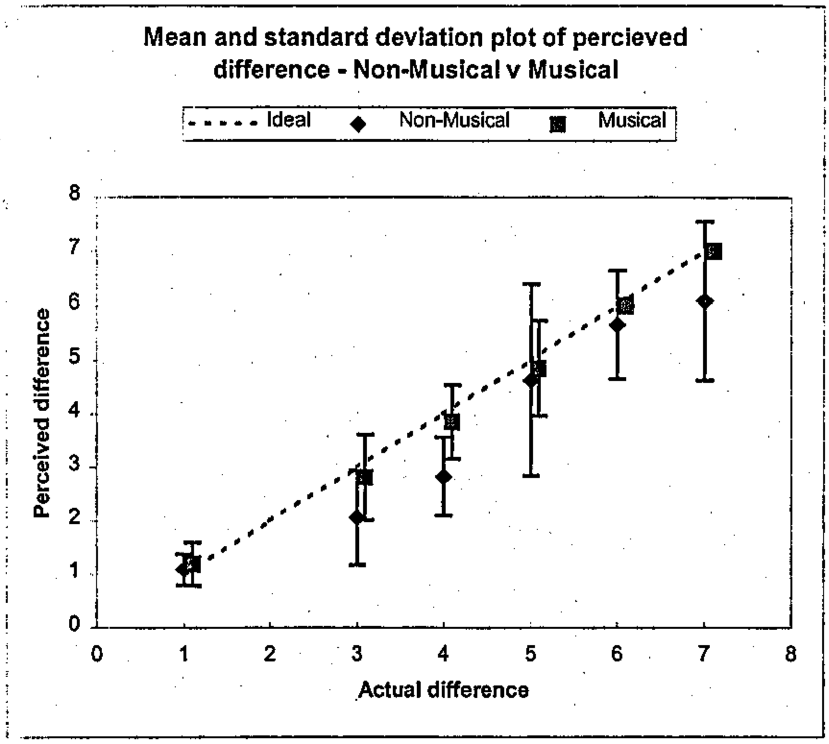


Figure 5.11 – Perceived tonal interval – non-musical v musical.

Musical	Diff 1	Diff 2	Diff 3	Diff 4	Diff 5	Diff 6	Diff 7
Mean	1.192308	-	2.807692	3.846154	4.846154	6	7
S.D	0.401918	-	0.800961	0.688737	0.880559	0	0

Non-Musical	1	2	3	4	5	6	7
Mean	1.088235	-	2.058824	2.823529	4.617647	5.647059	6.088235
S.D	0.287902	-	0.885615	0.727607	1.775502	0.996317	1.464068

Figure 5.12 – Table of perceived tonal interval – non-musical v musical.

The data given in Figures 5.11 and 5.12 suggest the 'musical' group perform with greater accuracy than the 'non-musical' group. Figure 5.14b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the differences of perceived intervals from the true intervals.

The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving musical intervals.

H_1 : The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving musical intervals.

	DIFF1	DIFF3	DIFF4	DIFF5	DIFF6	DIFF7
Mann-Whitney U	86.000	44.000	43.000	45.000	52.000	32.500
Wilcoxon W	239.000	135.000	134.000	136.000	143.000	123.500
	-1.331	-2.901	-3.02	-2.853	-3.040	-3.696
Asymp. Sig. (1 tailed)	.092	.002	.001	.002	.001	.000

Figure 5.14b – Table of test statistics for each perceived interval, 'non-musical' v. 'musical'.

From the data given in Figure 5.14b the null hypothesis cannot be rejected for single intervals suggesting that there is no significant difference between 'musical' and 'non-musical' listeners when perceiving intervals of 1. For the remaining intervals (i.e. >1) the null hypothesis can be rejected at the 1% level. This level of significance increases approximately in relation to the size of the true interval. This data suggests that there is no significant difference between 'non-musical' listeners and 'musical' listeners when perceiving small intervals. The data also suggests that as the test interval increases in size then 'musical' listeners tend to perform increasingly better than 'non-musical' listeners. Once again the data shows that the overall inaccuracy of the entire group increases as the pitch interval grows. In a similar experiment Alty[3] also showed that users of 'average musical ability' were capable of identifying pitch. This is supported by similar experiments performed by Rigas [160].

The data given in Figure 5.14c shows the results of a Chi-Squared test applied to the results obtained for pitch interval identification. The hypotheses are:

H_0 : Users are not capable of identifying pitch intervals. In particular, they are not capable of understanding musically represented numerical differences.

H_1 : Users are capable of identifying pitch intervals. In particular, they are capable of understanding musically represented numerical differences.

	DIFF1	DIFF3	DIFF4	DIFF5	DIFF6	DIFF7
Chi-Square	307.200	89.067	50.000	94.400	94.800	241.067
df	7	7	7	7	7	7
Asymp. Sig.	.000	.000	.000	.000	.000	.000

Figure 5.14c – Table of test statistics for each perceived interval.

These data show that the probability of obtaining the scores gathered from the users for the pitch interval tests at random are extremely low. From these data the null hypothesis can be confidently rejected concluding that users are capable of identifying pitch intervals and in particular that they are capable of identifying musically represented numerical differences. This is again encouraging since it suggests that users might be capable of understanding musically auralised sorting algorithm lists.

5.4. Shape perception experiments

5.4.1. Experiment construction

This experiment was designed to in order to help understand how listeners perceive the shape of short sequences of musical notes. As with the pitch tests documented earlier, the sequences were all played within one diatonic octave starting at 'Middle C'. Again, thirty subjects were asked to listen to the sequences of musical notes and determine their shape within the bounded diatonic scale. The timbre employed was again an acoustic grand piano, which was placed in the centre of the stereophonic field with no reverberation or chorus added. In order to create a context, the scale was first played once and each musical sequence was repeated three times. The time interval between repetitions was 2 seconds. The sequences had timing applied to them to make them appear more musical and hence more memorable. By this it is meant that rhythmic timing was applied to the tunes as opposed to presenting them with equal note durations and equal spaces between notes. The listeners were informed that information pertaining to the timing of the actual sequences was not important, the key feature was the overall contour of the tonal sequence and how it fitted into the bounded diatonic scale. Subjects were once again told that each of the eight notes within the bounded scale were mapped to the numbers one to eight. Upon listening to each sequence, the subjects were asked to 'draw' the shape of the tonal pattern by placing 'X' marks within a provided grid as shown in the demonstration sequences below. Each sequence contained notes that were only present in the contextual bounded diatonic octave scale. Each sequence was of varying length containing between six and eight notes. Thus each grid was eight notes high by eight note events wide.

Demo 1

Pitch	Note Sequence --->							
8								
7								
6					x			
5						x		
4		x		x			x	
3								x
2								
1	x		x					

Demo 2

Pitch	Note Sequence --->							
8		x						
7								
6								
5			x			x		
4					x			
3				x				
2								
1	x							

Demo 3

Pitch	Note Sequence --->							
8		x						
7				x				
6						x		
5							x	
4								
3					x			
2			x					
1	x							

Demo4

Pitch	Note Sequence --->							
8								
7								
6								
5					x			
4				x		x		
3			x				x	
2		x						x
1	x							

Figure 5.15 – Example answer grids, shape perception experiments.

The test subjects were played the four musical sequences and shown the four diagrams in Figure 5.15 above. The questionnaire is given in full in Appendix D. The subject group was then asked to draw the shape of a further six tonal sequences by placing 'X' marks in blank grids. Each test was performed on thirty individual test subjects. The six sequences are given in the grid diagrams shown below in Figure 5.15a.

Shape 1.

Pitch	Note Sequence -->							
8								
7								
6								
5			X			X		
4	X	X			X			
3				X				
2								
1								

Shape 2.

Pitch	Note Sequence -->							
8								
7								
6								
5					X			
4						X		
3	X	X					X	
2			X					
1				X				

Shape 3.

Pitch	Note Sequence -->							
8								
7								
6								
5					X		X	
4				X				
3			X					
2		X				X		
1	X							

Shape 4.

Pitch	Note Sequence -->							
8								
7								
6								
5	X		X					
4		X		X				
3					X			
2						X		X
1							X	

Shape 5.

Pitch	Note Sequence -->							
8	X			X				
7								
6			X					
5		X					X	
4						X		
3					X			
2								
1								

Shape 6.

Pitch	Note Sequence -->							
8								
7								
6								
5				X		X		X
4								
3	X							
2		X						
1			X		X		X	

Figure 5.15a – Shape test stimuli.

5.4.2. Evaluation mechanism

In order to analyse the results of the above experiments it is necessary to create some form of scoring mechanism. Certain algorithms exist that allow for the measurement of similarities between two graphs. Parametric bivariate correlation equations such as

Pearson's correlation coefficient and non-parametric bivariate correlation equations such as Kendall's tau-b and Spearman's rho can be used to test for linear correlation between two data sets. Such approaches can be applied to shapes represented by sequential numerical values to compare similarities. However, significant correlation is often only observed when a large amount of the data correlation is present. The similarities that are required for identification in this series of experiments are not simply relationships between numerical values. Certain general features such as 'ascending', 'descending' or 'randomness' are required to be identified and scored. Correlation algorithms as mentioned above do not possess the ability to correlate shapes in this way. In this case the desired result is an approximate gauge of the accuracy of the perceived contour of the musical sequences. Therefore a scoring mechanism has been developed and employed.

In order to score the correlation between the reference shapes and those shapes drawn by the test subjects it is important to identify the features of interest within the context of this series of experiments. The following features were identified:

- Sequential pattern progression – this feature is concerned with identifying the key points within each pattern and outlining the direction of each note with reference to the other notes within the sequence. In the initial design of the scoring mechanism, this feature was scored by comparing each element to the element that preceded it and noting the direction of 'tonal travel' (up, down or same). After preliminary experimentation with this mechanism, it was found that the comparisons were too confined to their neighbouring elements and scoring appeared inaccurate in certain cases where simple shifting occurred but where the shape was relatively accurate. Due to these inaccuracies it was decided that the position of each element should be compared with the positions of all subsequent elements. This mechanism scored the shape progression relative to all elements within the list and therefore gave greater accuracy when scoring pairs shapes that were almost identical with the exception of one or two elements missing or being misplaced.

- Amplitude – this feature is concerned with the spread of the amplitude of the perceived shape. If the reference shape was spread over one entire diatonic octave (8 notes) and the perceived shape was spread over 4 notes then the score should be low, if the spread is close to or exactly the same as the reference spread then the score should be high.

- General shape – this feature is the subjective element of the scoring mechanism. The scorer gives a point score based upon how similar the general shape of the reference pattern is to the general shape of the perceived pattern. In order to control the subjection, an informal test was performed by the author on 15 test subjects which asked them to rate the similarity between graphs based upon extracting features concerned with this thesis. Specifically, the identification of up to 3 general features classed as random or ordered in a specified direction.

The scoring mechanism was used to rate the similarity between 50 sets of graphs each with a reference graph for comparison. The same data was also presented to 15 test subjects and the results of the two tests analysed. The weighting of the components in the scoring mechanism were then corrected to reflect the scores obtained from the subjective opinions of the test subjects. It is important here to reiterate the reason for developing a scoring mechanism with a subjective component. The correlation required between shapes that is of concern in this thesis is at a higher level of abstraction than mathematical correlation schemes would permit. Several 'problematic' shapes were subjected to the scoring mechanism to identify the likely points of failure. By problematic, it is meant that they would score a low mathematical correlation but would capture the more abstract properties of interest in this thesis. The scoring mechanism was therefore derived by identifying the more abstract features both mathematically and observationally in order to capture the general attributes of a graph that would satisfy the concern of this thesis.

The derived weighting of each of the three features for the overall score is given as:

- Sequential pattern progression – maximum 7 points.
- Amplitude – maximum 1 point.
- General shape – maximum 2 points.

The scoring out of a maximum of 7 for the sequential pattern progression has been implemented with the following considerations. The total number of valid comparison points between each current element and all subsequent following elements is given by (n-1)+(n-2)+(n-3)…..(n-n) points, where 'n' represents the number of elements in the list. 1 point is given for an accurate correlation for each of the valid comparison points between the reference list and the perceived list. The total score is then divided by 7 and multiplied by the equation given above, thus yielding a final score out of 7.

The scoring for the amplitude is calculated by finding the high and low points in both the reference list and the perceived list and finding the difference between high and low points in each case. The differences are then compared between lists to yield the difference in spread between the reference pattern and the perceived pattern. If the difference in spread is zero, or one, then the lists are deemed to have similar amplitudes and a score of 1 point is awarded. If the difference in spread is two then 0.5 points are awarded and any spread difference greater than two yields a score of 0. These scores have been distributed based upon the fact that the tonal sequences are all within the same eight notes, therefore the maximum difference in spread between two sequences (one confined to the same note and one using the entire eight note octave) would be 7.

The scoring of the general shape out of a maximum of 2 is calculated by awarding the appropriate portion of the maximum based upon the amount of identifiable features. One feature that became apparent when initially scoring and analysing the results using this mechanism was that patterns that are correct in every way except that they have been shifted left or right score relatively low. To accommodate this it was decided that the scoring should be done on the given perceived list and on left and right shifted versions of the perceived list. The highest of these three scores would be the score of interest.

The scoring mechanism has been implemented on PC and has been written in the C programming language. This implemented version of the conceptual scoring mechanism has been used to score the correlation between the reference sequence shapes and those given by the test subjects in this series of experiments.

5.4.3. Results and analysis

The group of test subjects used were the previously described 'Group 1'. The diagram given in Figure 5.17 shows the accuracy of each of the thirty test subjects for all six of the shape perception questions. The graph has been ordered and colour coded in terms of musical ability. It can be seen that there is a general trend that might suggest that 'musical' test subjects tend to perform slightly better than 'non-musical' test subjects when perceiving the contour of the tonal sequences.

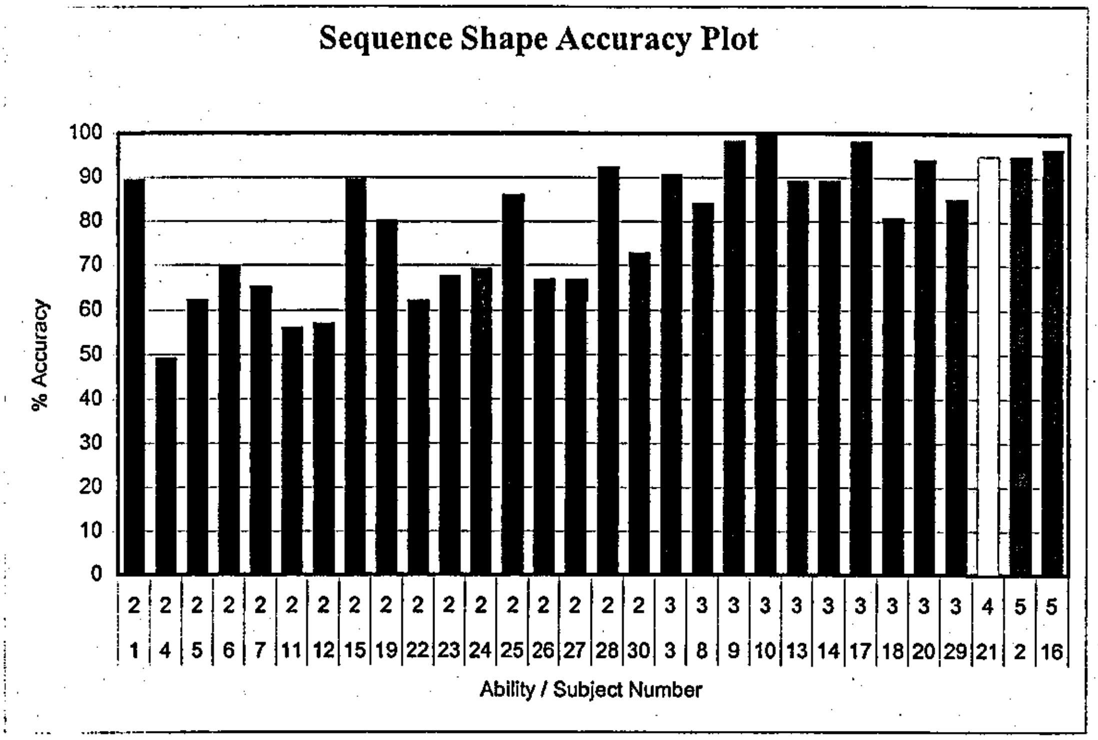


Figure 5.17 – Shape perception accuracy plot – entire test group.

The graph in Figure 5.17 shows that of the 'non-musical' test subjects with a musical ability rating of two (out of a maximum of six), five performed with a greater than 80% accuracy level when perceiving the shapes of the tonal sequences. The remainder still performed well but generally not as accurately as the 'musical' test subjects. These 'musical' listeners performed with no less than 81% accuracy. Since the data in this case have been combined to give an average score over all six shapes, it is important to ascertain whether certain shapes were perceived more accurately than others and whether the 'musical' group performed differently to the 'non-musical' group for these different information types.

Figure 5.18 shows how the group of test subjects as a whole performed on each of the six shape perception questions.

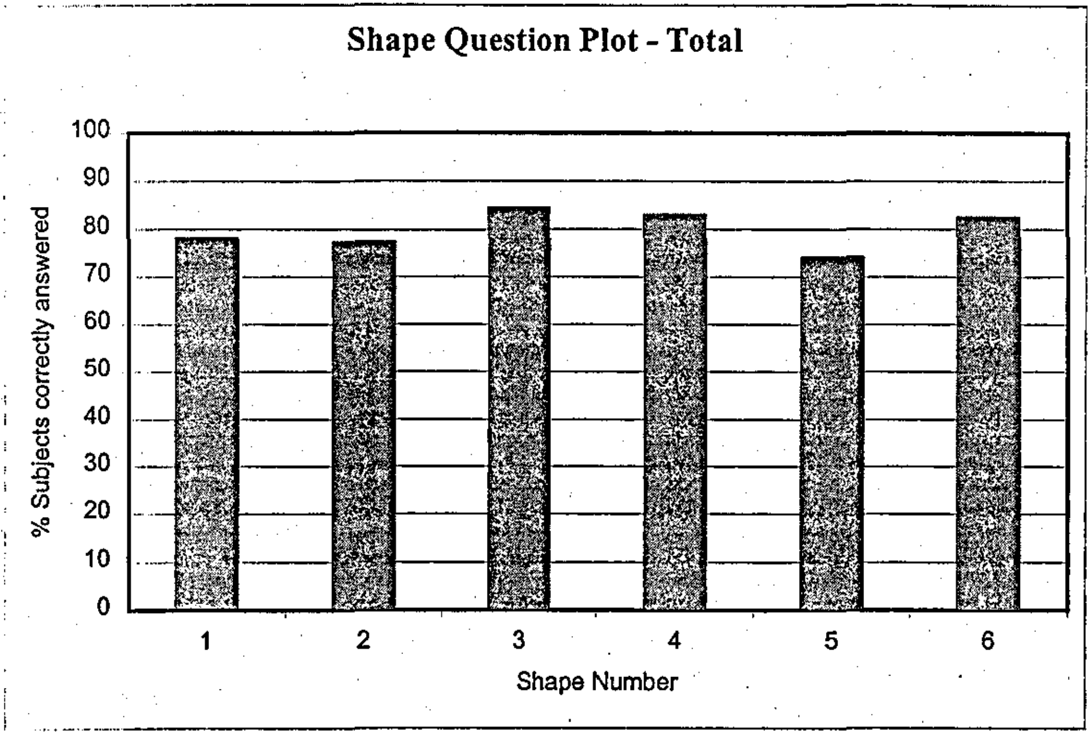


Figure 5.18 – Shape perception accuracy plot by shape – entire test group.

Figure 5.18 suggests that a small difference between the perceptions of the different shapes. The noticeable difference is the increase in accuracy for the subjects' perception of shapes 3, 4 and 6. The only observable difference in these shapes is that each possesses very prominent and obvious features.

The data given in Figure 5.18a shows the results of the Wilcoxon Signed Ranks test applied to the shapes perception results given above. The hypothese are:

H_0 : There is no difference in perception between the shapes 3, 4 and 6 and shapes 1, 2 and 5.

H_1 : There is a difference in perception between the shapes 3, 4 and 6 and shapes 1, 2 and 5.

	EASY - HARD
Z	-3.10
Asymp. Sig. (1-tailed)	.00

Figure 5.18a – Table of test statistics for the perception of shape 3, 4 and 6 compared to shapes 1, 2 and 5.

From this data the null hypothesis can be rejected at the 0.1% level of confidence concluding that the perception of shapes 3, 4 and 6 is significantly better than the group's perception of shapes 1, 2 and 5.

The features for shapes 3, 4 and 6 are:

- Shape 3 – Long ascent followed by sharp descent followed by sharp ascent.

Pitch	Note Sequence --->							
8								
7								
6								
5					X		X	
4				X				
3			X					
2		X				X		
1	X							

- Shape 4 – Sharp shallow trough followed long descent followed by small ascent at the tail.

Pitch	Note Sequence --->							
8								
7								
6								
5	X		X					
4		X		X				
3					X			
2						X		X
1							X	

- Shape 6 – Medium descent followed repeated sharp peaks.

Pitch	Note Sequence -->							
8								
7								
6								
5				X		X		X
4								
3	X							
2		X						
1			X		X		X	

The features of Shapes 1, 2 and 5 are:

- Shape 1 – Short flat section followed by sharp ascent followed by short medium descent followed by medium length ascent.

Pitch	Note Sequence -->							
8								
7								
6								
5			X			X		
4	X	X			X			
3				X				
2								
1								

- Shape 2 – Short flat section followed by medium length descent followed by short sharp ascent followed by medium length descent.

Pitch	Note Sequence -->							
8								
7								
6								
5					X			
4						X		
3	X	X					X	
2			X					
1				X				

- Shape 5 – Sharp descent followed by medium ascent followed by sharp ascent followed by very sharp descent followed by medium length ascent.

Pitch	Note Sequence -->							
8	X			X				
7								
6			X					
5		X					X	
4						X		
3					X			
2								
1								

The best perceived shapes were those that possess long and obvious ascents or descents, or repeated patterns. The shapes that did not translate quite so well each had more complex and less obvious and non-repetitive features.

Figure 5.19 shows how 'musical' listeners performed compared to 'non-musical' listeners in terms of accuracy of perception for each of the six shapes.

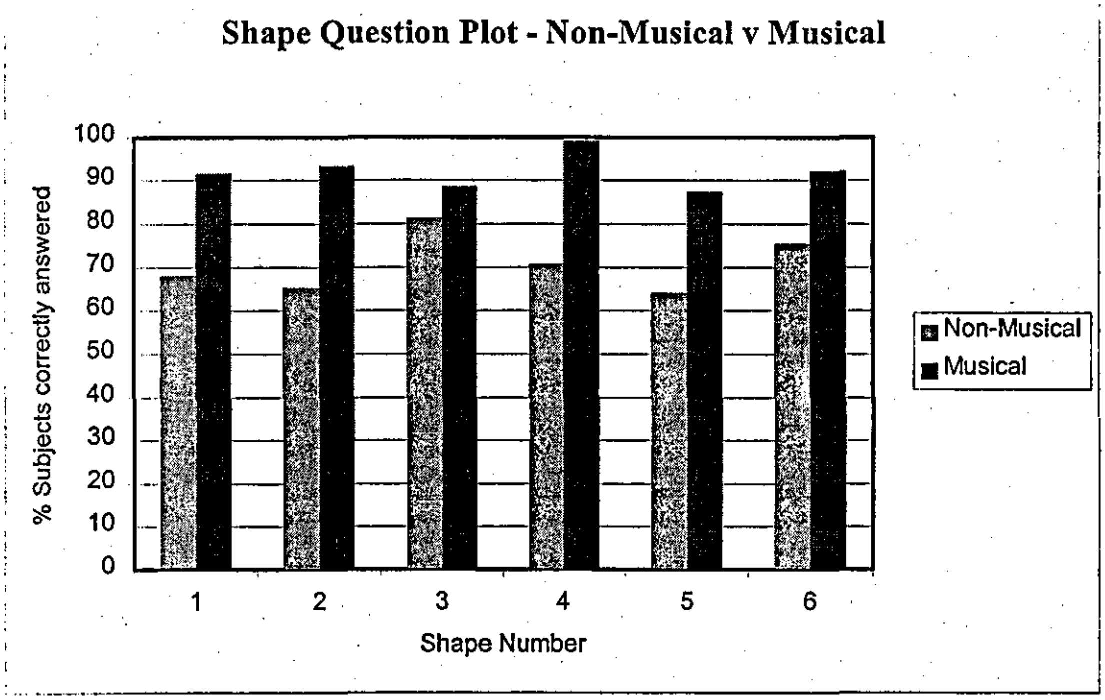


Figure 5.19 – Shape perception accuracy by shape – non-musical v musical listeners.

The same feature is observed for the musically untrained group of listeners as was observed for the group as a whole. Certain obvious or repetitive features translate better than more complex or non-repetitive features.

Figure 5.19b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the scores obtained for the perceived shapes compared to the true shapes.

The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving musical shapes.

H_1 : The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving musical shapes.

	SHAPE1	SHAPE2	SHAPE3	SHAPE4	SHAPE5	SHAPE6
Mann-Whitney U	35.500	22.000	71.000	37.500	27.500	50.500
Wilcoxon W	188.500	175.000	224.000	190.500	180.500	203.500
Z	-3.25	-3.74	-1.80	-3.32	-3.508	-2.573
Asymp. Sig. (1-tailed)	.00	.00	.033	.000	.000	.005

Figure 5.19b – Table of test statistics for each perceived shape, 'non-musical' v. 'musical'.

The null hypothesis can be rejected for all shapes suggesting that there is significant difference between 'musical' and 'non-musical' listeners when perceiving musical shapes. The null hypothesis can be rejected at the 0.1% level of significance for shapes 1, 2, 4 and 5. It can also be rejected for shape 3 at the 5% level of confidence and for shape 6 at the 1% level of confidence. This data suggests that overall 'musical' listeners perform significantly better than 'non-musical' listeners when perceiving short musical patterns. The data also suggests that there is a greater difference between 'non-musical' listeners and 'musical' listeners when perceiving complex shapes compared to simpler shapes. The overall accuracy for all test listeners is however, observably high. This data suggests that musically trained people might be more suitable for understanding sorting algorithm auralisation. However, this experiment presented the listeners with tonal sequences that had musical timing applied to them in an attempt to make them more 'musical'. It would be of interest to remove this musical timing as the proposed algorithm auralisations will not represent the list states with such 'musicality', but rather play the lists with equal note durations and equal spacing between the note. This might yield more favourable results for the musically untrained subjects.

The overall average score for the perception of musically represented shapes in the context of this experiment and subject to the marking scheme is 80.1%. By examining the marking scheme and using the binomial expansion the probability of obtaining such a score at random can be approximated.

1. Sequential pattern progression – Assume simplest shape with a list size of six, the minimum size played. Each element compared to all following elements. Therefore (6-1)! possible outcomes = 720. However, the score is given for the relationship in terms of direction between element pairs. This mechanism allows for 'alias' shapes that would not exactly correlate with the reference shape by more classical correlation methods but would score full marks here as the directional relationships between all element pairs is satisfied. For a list size of 6, the highest amount of alias shapes is 70. This value has been derived by using the mechanism on all possible shapes and references. The mechanism also left and right shifted the data to allow for shifting errors, yielding 3 comparisons, the best of which was taken as the score. This trebles the chance of success. Therefore, assuming the simplest and shortest shape with the highest amount of possible aliases and accommodating the shifting nature of the mechanism, the probability of gaining a maximum score at random is given as (3x70/720). Or **7/24.**

2. Amplitude – Full marks awarded if amplitude is the same or 1 values either side, therefore there are 3 chances at gaining full marks. There are 8 possible values of difference, so the probability of gaining a full score at random is **3/8.**

3. General shape – The features are identified as randomness, sorted ascending and sorted descending. Therefore there are three possible answers. Assuming the simplest shape with only one single feature the probability of gaining a full score at random is given as **1/3.**

Where the number of successes (r) is approximately 8 and the number of trials (n) is 10. The probability of the occurrence of **r** successes in **n** trials is given by the binomial distribution:

$$P(r\ successes) = \frac{n!}{(n-r)!\,r!}\,q^{n-r}\,p^{r}$$

Where **p** represents the probability of success and **q** represents the probability of failure.

By applying this equation to each of the three features of the marking scheme and combining the probabilities and trebling (to account for the three possible scoring positions due to the left and right shifting in the marking scheme), the probability of scoring 80% at random is calculated as being approximately of the order of $7.434e^{-8}$ which is extremely low. Although it is not entirely appropriate to use a formal statistical method on data that has been subjected to a subjective marking scheme, the value of probability gives the approximate order of magnitude and along with observation of the experimentally derived data it might suggest that the users are able to understand certain features of the shape of a list when represented musically. It is worth noting here that determining at what level and how accurately users can interpret and understand shapes of tonal sequences is not a real concern of this thesis.

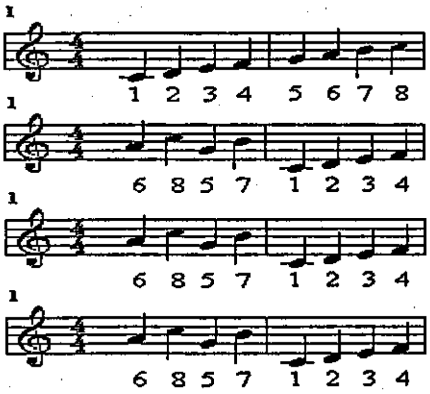
5.5. List feature extraction experiments

5.5.1. Experiment construction

In this set of experiments twenty subjects were asked to listen to sequences of eight musical notes that corresponded to a list containing the numbers 1 to 8. Once again the tonal sequences were all within a bounded diatonic octave scale. The timbre employed was an acoustic grand piano, which was placed in the centre of the stereophonic field with no reverberation or chorus added. The bounded diatonic scale started at 'Middle C' and ascended by one octave (eight notes). In order to create a context, the scale was first played once before each note sequence. Subjects were told that each of the eight notes within the bounded scale were mapped to the numbers one to eight and that the sequences therefore represented lists of eight numbers. The noticeable difference between this set of experiments and the previously documented shape perception experiments is that no musical timing was present in the sequences. Each element was separated by the same time interval and all the shapes were extracts from algorithm executions.

After listening to each sequence, the subjects were asked to interpret the shapes of the lists. This was done in three stages. The first time that the test subjects were played a sequence they were asked to identify the features of the list and explain them with a written description. The second time, they were presented with the list again and asked to describe in words the shape features of the list. On the third repetition, test subjects were asked to draw the shape of the tonal sequence by placing 'X' marks in a blank grid (much like the List Shape Perception experiments previously documented in this chapter).

In order to ensure that subjects understood the procedure, an example shape was presented three times with two written descriptions followed by a diagram of the tonal sequence. The subjects were presented with the information in Appendix I and shown Figure 5.20:



The octave scale starting with middle C.

1st repetition of musical sequence.

2nd repetition of musical seqence.

3rd repetition of musical sequence.

Figure 5.20 – List feature extraction example.

Five shapes each comprising eight notes within the bounded diatonic octave scale starting at 'Middle C' were then presented to 20 subjects.

5.5.2. Results and analysis

Subjects were given the musical questionnaire (shown in full in Appendix B) and the results are shown in Figure 5.58. Subjects were all computer studies undergraduates, comprising 1 female and 19 male. The group consisted of 8 'non-musical' listeners and 12 'musical' listeners. This group are termed 'Group 2'.

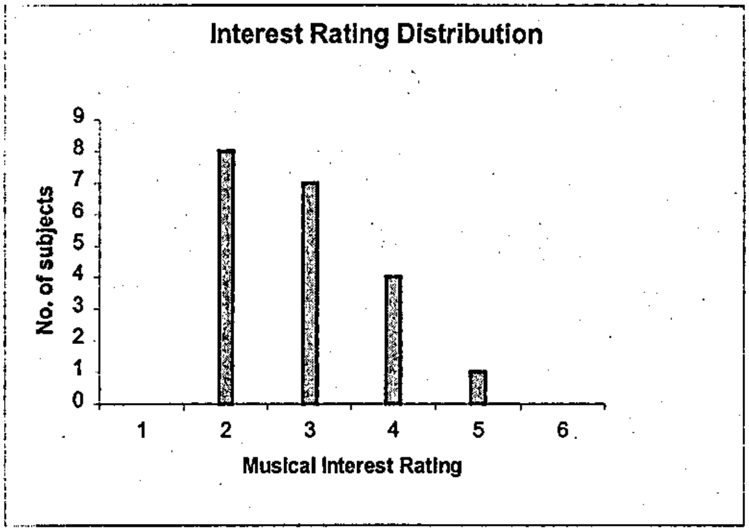


Figure 5.21 – Musical ability rating – list feature extraction experiments.

An additional test was carried out in order to check if the subjects could draw the shapes of simple tunes. For example, some subjects might fully understand the shape of the tonal sequences in their minds, but be unable to draw them, and the test would then be testing their drawing ability rather than their musical comprehension of the shapes. They were each played a well-known tune. Figure 5.22 shows the performance of the entire group.

The results were scored using the scoring mechanism described in section 5.4.2. The classification boundaries were set at:

Good	–	70% to 100%
Average	–	40% to 69%
Bad	–	0% to 39%

The data shows that 60% of the test group were able to draw simple tunes at a 'good' level, with 25% being 'average' and the remaining 15% being 'bad' at drawing. These classifications were derived from experiments where each listener was played a simple well-known tune and asked to draw it. This drawing ability (or lack of it) is distributed across both the 'non-musical' and 'musical' groups as shown in Figures 5.23 an 5.24.

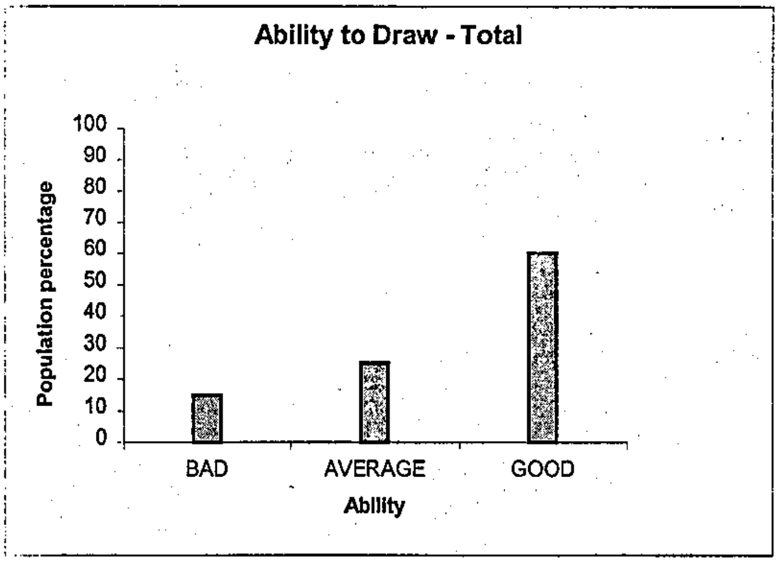


Figure 5.22 – Drawing ability rating – list feature extraction experiments.

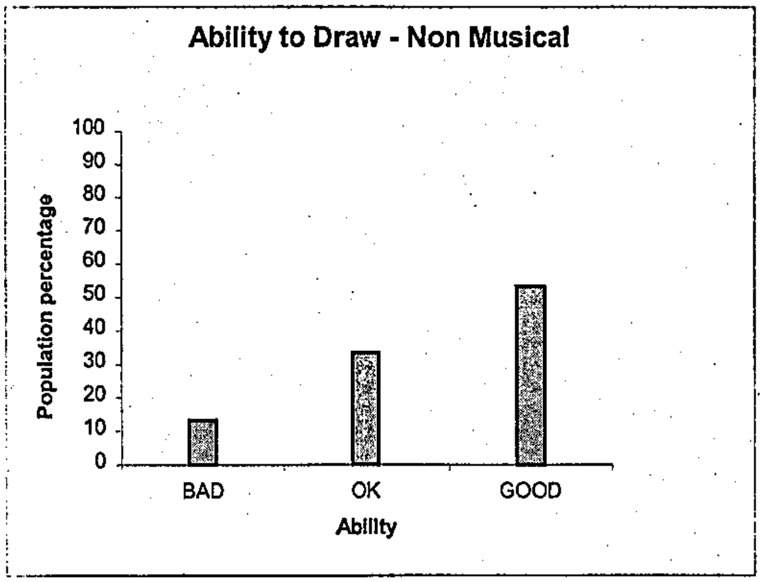


Figure 5.23 – Drawing ability rating – list feature extraction experiments – non-musical.

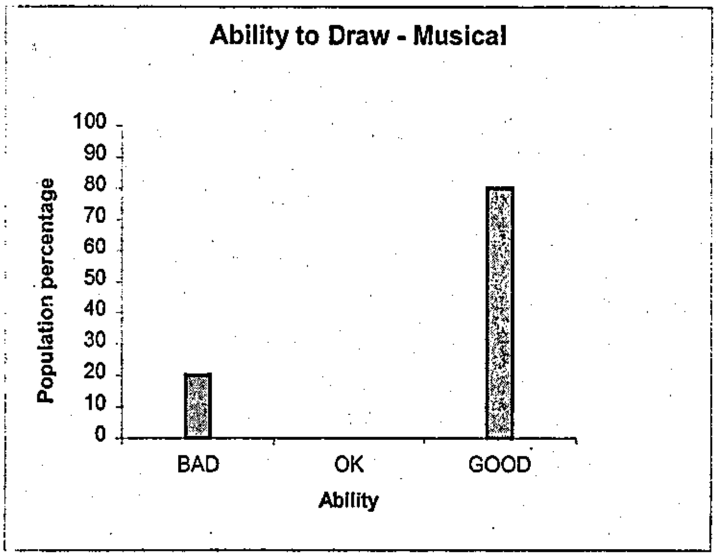


Figure 5.24 – Drawing ability rating – list feature extraction experiments – musical.

Figures 5.23 and 5.24 show that both sub-groups contain a majority of test subjects who have the ability to draw.

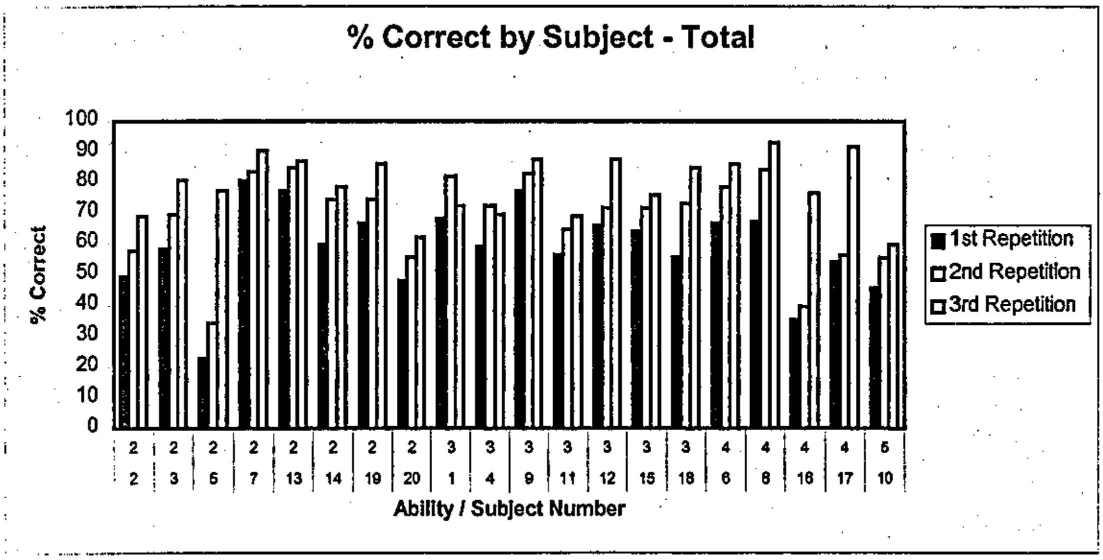


Figure 5.25 – List feature extraction accuracy 1st v 2nd v 3rd repetition – entire group.

Figure 5.25 shows how the entire group of test subjects performed for the first, second and third repetitions when describing and drawing the shape of the tonal sequences. The graph is ordered in terms of musical ability with the least musical subjects being plotted to the left and those with the greatest musical ability being plotted to the right. There is a small increase in accuracy; due to learning effects where the second repetition reinforces and further defines the first repetition and might also suggest that the shapes are easier to draw than to describe. Again it is important to divide the entire group of test subjects into their musical ability classifications in order to show if being musically trained has any effect on perceiving the shapes of these tonal sequences. Figure 5.26 shows how the 'non-musical' group of test subjects perform against the 'musical' group of test subjects when drawing the shapes during the third repetition.

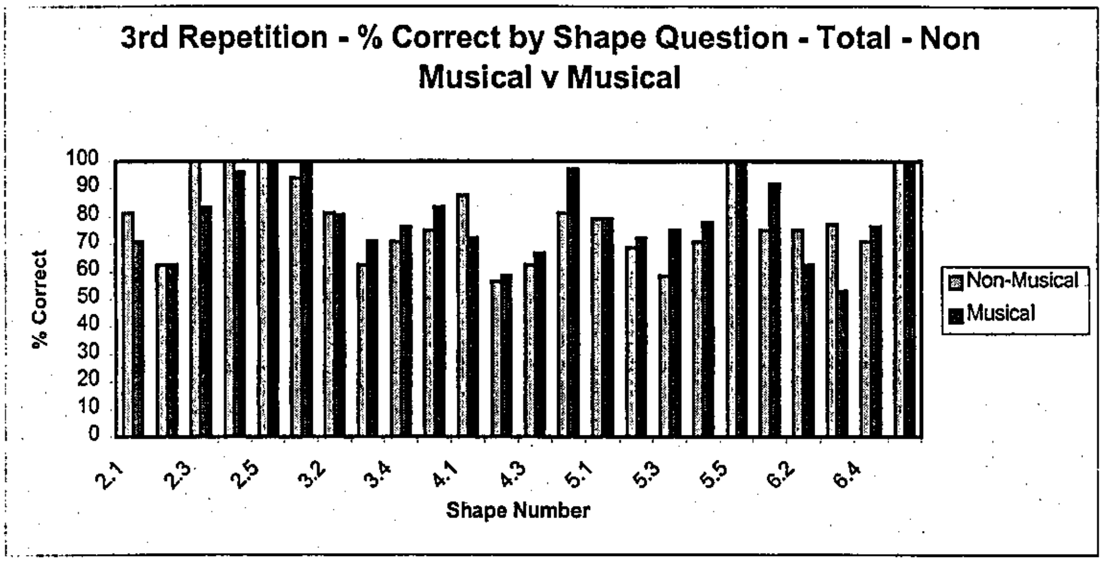


Figure 5.26 – List feature extraction accuracy 3rd repetition – musical v non-musical.

Figure 5.27b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the scores obtained for the perceived shapes of short tonal sequences compared to the true shapes of short tonal sequences with no musical timing.

The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving tonal sequences with no timing.

H_1 : The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving tonal sequences with no timing.

	TOTAL Q's
Mann-Whitney U	26707.000
Wilcoxon W	45235.000
	-.583
Asymp. Sig. (1-tailed	.281

Figure 5.27b – Table of test statistics for perceived shape, 'non-musical' v. 'musical'.

Based upon the test statistics given in Figure 5.27b the null hypothesis cannot be rejected and we conclude that there is no difference between the performance of the 'non-musical' group and the 'musical' group when perceiving short tonal sequences with no musical timing. As previously mentioned the data in this series of experiments has been analysed using the marking described in Section 5.4.2. In terms of magnitude, the results are observably comparable to those in the previous experiment. It is therefore not unreasonable to suggest that the probability of obtaining these test scores at random would also be exceptionally low. Suggesting that listeners might be able to determine the features of interest to this thesis from musically represented lists of numbers.

Another feature of this series of experiments that warrants investigation is the discrimination between the information types that have been used. It is important to ascertain whether the test subjects in this context more readily understand certain types of information. The diagram below in Figure 5.28 shows how the entire group of test subjects performed with shapes that were termed as being 'easy', that is, shapes that comprised of two or less features such as 'all ascending' or 'random then ascending'. A "hard" shape contains three or more features.

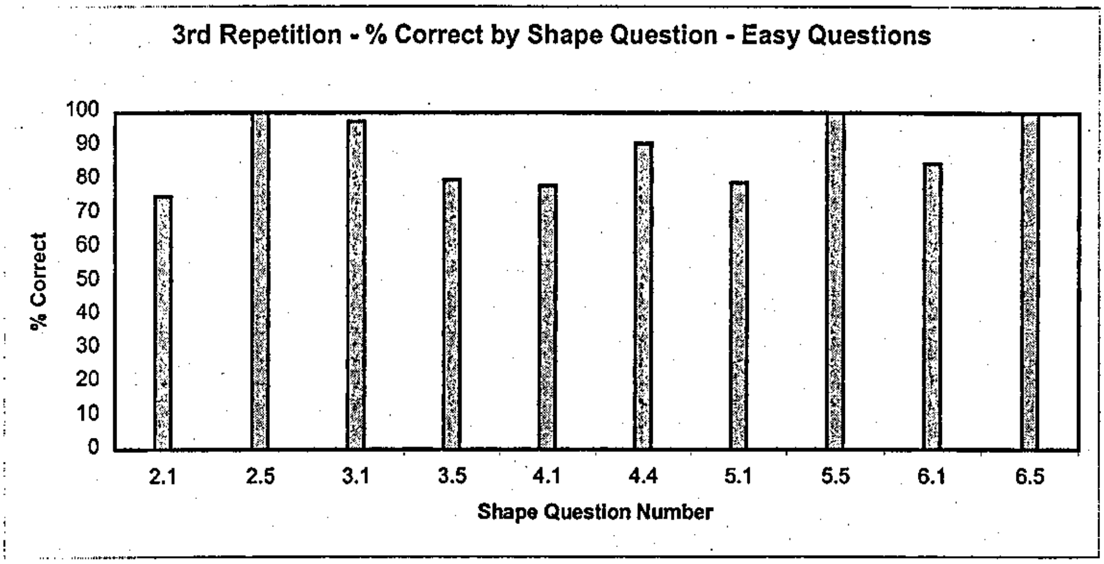


Figure 5.28 – Feature extraction accuracy 3rd repetition by easy question type – entire group.

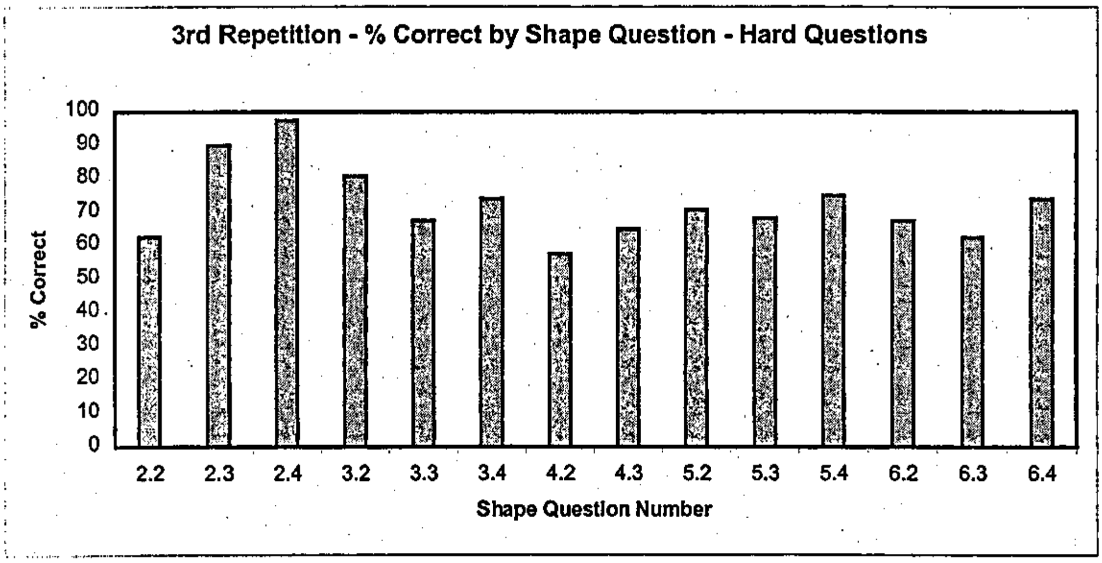


Figure 5.29 – Feature extraction accuracy 3rd repetition by hard question type – entire group.

Figures 5.28 and 5.29 suggest that tonal sequences with one or two features are more readily identified than tonal sequences with three or more features.

The data in the tables given in Figure 5.31 shows the results of the Wilcoxon Signed Ranks non-parametric test applied to the scores obtained for the perceived shapes of short tonal sequences with no musical timing compared to the true shapes of short tonal sequences with no musical timing for both groups of hard and easy questions over the entire test group of listeners.

The hypotheses are:

H_0 : There is no difference between the 'hard questions' and 'easy questions' when perceiving tonal sequences with no musical timing.

H_1 : There is a difference in accuracy between 'hard questions' and 'easy questions' when perceiving tonal sequences with no musical timing.

	HardQ – EasyQ
Z	-5.059
Asymp. Sig. (1-tailed)	.000

Figure 5.31 – Table of test statistics for perceived shape, hard questions v. easy questions

From the data given in Figure 5.31 the null hypothesis can be rejected at the 0.1% level of confidence suggesting that that there is a highly significant difference between 'hard' question types and 'easy' question types when perceiving the shapes of short tonal sequences with no musical timing. It can therefore be concluded that 'hard' question types are understood significantly less than 'easy' question types.

As these data have proven that a significant difference exists between accuracies when perceiving 'hard' and 'easy' questions it is necessary to test for any significant difference between 'non-musical' and 'musical' test groups for each of the question types. The data in the tables given in Figure 5.33 shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the scores obtained for the perceived shapes of 'easy' and 'hard' short tonal sequences with no musical timing for the 'non-musical' group compared to the scores of the shapes of 'easy' and 'hard' short tonal sequences with no musical timing for the 'musical' group.

The hypotheses are:

H$_0$: There is no difference between the 'non-musical' and 'musical' test groups when perceiving 'easy' or 'hard' tonal sequences with no timing.

H$_1$: The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving tonal sequences with no timing.

	EasyQ	HardQ
Mann-Whitney U	4611.500	9392.500
Wilcoxon W	7851.500	23588.500
Z	-.68	-.025
Asymp. Sig. (1-tailed)	.24	.488

Figure 5.33 – Table of test statistics for 'hard' and 'easy' shapes 'non-musical' v 'musical'.

From the data given in Figure 5.33 the null hypothesis cannot be rejected concluding that there is no significant difference between the 'non-musical' group and the 'musical' group when perceiving short tonal sequences with no musical timing irrespective of the level of difficulty of the shapes. These same data also suggest that there is less difference between the two groups for hard questions than there is for easy questions suggesting that the 'musical' group perform better on easy question types.

The next feature of interest in this series of experiments is the investigation of whether the users' ability to draw affects the accuracy of the answers that they provided.

The data shown below in Figure 5.34 represents the perception accuracy of shapes of tonal sequences during the third repetition for all test subjects. The data are ordered in terms of their drawing ability.

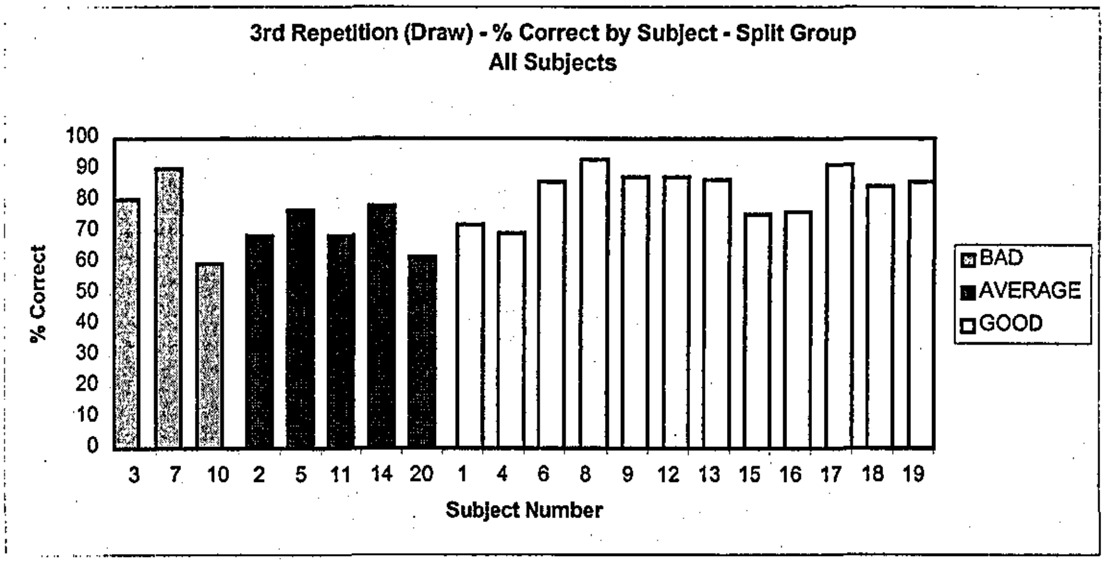


Figure 5.34 – Feature extraction accuracy 3rd repetition by drawing ability – entire group.

These data suggest that there is little appreciable difference in perception accuracy between those of 'good' drawing ability and those with less than 'good' drawing ability.

5.6. List state perception experiments

5.6.1. Experiment construction

In this set of experiments thirty subjects were asked to listen to sequences of eight musical notes that corresponded to the numbers one to eight. Once again the tonal sequences were all within a bounded diatonic octave scale. The timbre employed was an acoustic grand piano, which was placed in the centre of the stereophonic field with no reverberation or chorus added. The bounded diatonic scale started at 'Middle C' and ascended by one diatonic octave (eight notes). In order to create a context the scale was first played once and each note sequence was repeated three times. Subjects were told that each of the eight notes within the bounded scale were mapped to the numbers one to eight and that the sequences therefore represented lists of eight numbers. Upon listening to each sequence, the subjects were asked to select from a list of options comprising 'Unsorted/Random', 'Sorted Ascending' and 'Sorted Descending'. The feature that was required to be identified was the state of the list. The information presented to the group is given in Appendix E.

Group 1 were used in this set of experiments. Three examples were played to each test subject three times. Following the examples, a further five tests were played. Only five tests were chosen at this stage in order to investigate if the more simple features could be extracted. The most complex lists in this series only have two features in that they are almost sorted with the exception of one element. This test has been designed to also determine if one element being out of place would be classed as unsorted by the listener. These tests are given below.

Pitch	Note Sequence --->							
8								
7							X	
6						X		
5					X			
4				X				
3			X					
2		X						
1	X							X

Pitch	Note Sequence --->							
8					X			
7		X						
6	X							
5							X	
4				X				X
3								
2			X					
1						X		

Pitch	Note Sequence --->							
8								
7		X						
6			X					
5				X				
4					X			
3						X		
2							X	
1	X							X

Pitch	Note Sequence --->							
8	X							
7		X						
6			X					
5				X				
4					X			
3						X		
2							X	
1								X

Pitch	Note Sequence --->							
8								X
7							X	
6						X		
5					X			
4				X				
3			X					
2		X						
1	X							

Figure 5.34a – Preliminary list shape stimuli.

The results indicated that all of the test subjects successfully identified the states of all of the lists with an accuracy of 100% (refer to data on accompanying CD). These results showed that all listeners, regardless of their musical ability, were fully capable of distinguishing between musically represented sorted and unsorted lists of numbers, even when some lists were sorted with the exception of one element.

In a further set of further experiments, test subjects were again asked to listen to sequences of eight notes that represented lists of eight numbers all played within the same diatonic octave staring from 'Middle C'. The same timbre and placement were also employed. This time listeners were played lists that were sorted into ascending order but between one and three elements were incorrectly placed. Subjects were first shown and played the example diagram given below in Figure 5.41. In this example elements 4 and 5 produce a descent in pitch and are therefore incorrectly placed.

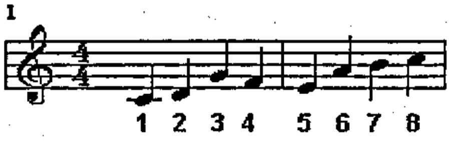



Figure 5.41 – Incorrect element placement example.

The thirty test subjects were then each asked to identify the incorrectly placed elements in each of the five tests by circling their position in diagrams.

5.6.2. Results and analysis

Group 1 were used in this set of experiments. Figures 5.43 and 5.44 show the users' perception of each of the incorrectly placed elements within the partially sorted lists. The stimuli were derived from real sorting algorithm list states and hence the occurrence of swap a swap in the fifth position has been omitted due to the limitations of the derived lists. The results show that the error distribution is fairly even across the list of numbers except for the eighth and final element. This decrease in placement accuracy is due to the fact that the test lists incorporated some sequences where both the seventh and eighth elements were successively incorrectly placed. This successive erroneous information has clearly been shown to confuse the listeners and would suggest that single out of place elements are more easily identified than multiple neighbouring out of place elements.

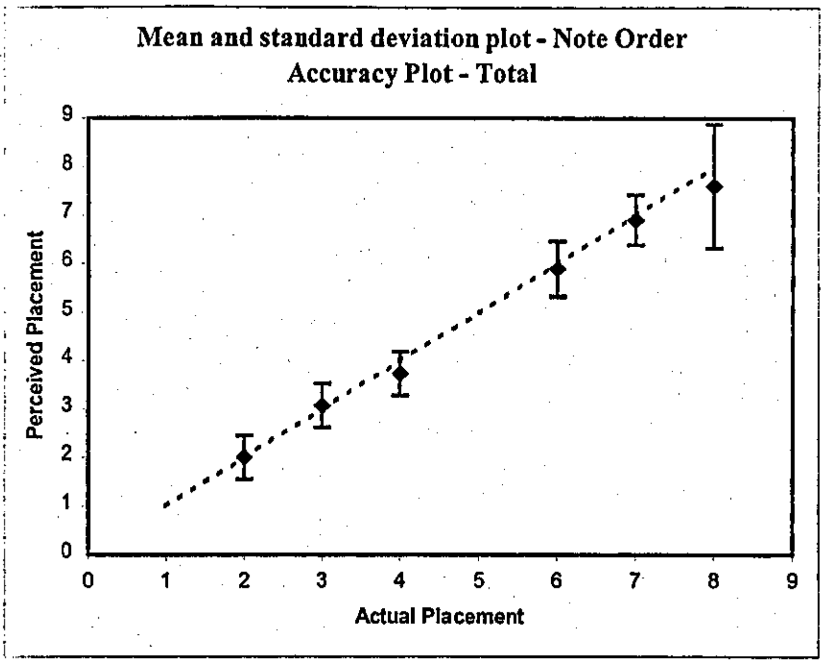


Figure 5.43 – List state note order accuracy – entire group.

Actual Placement	1	2	3	4	5	6	7	8
Perceived Placement Mean		2	3.066667	3.733333		5.892857	6.898305	7.6
S.D		0.454859	0.449776	0.449776		0.566947	0.515113	1.275769

Figure 5.44 – Table of list state note order accuracy – entire group.

Figure 5.45 shows how musically trained and untrained listeners performed in this series of experiments.

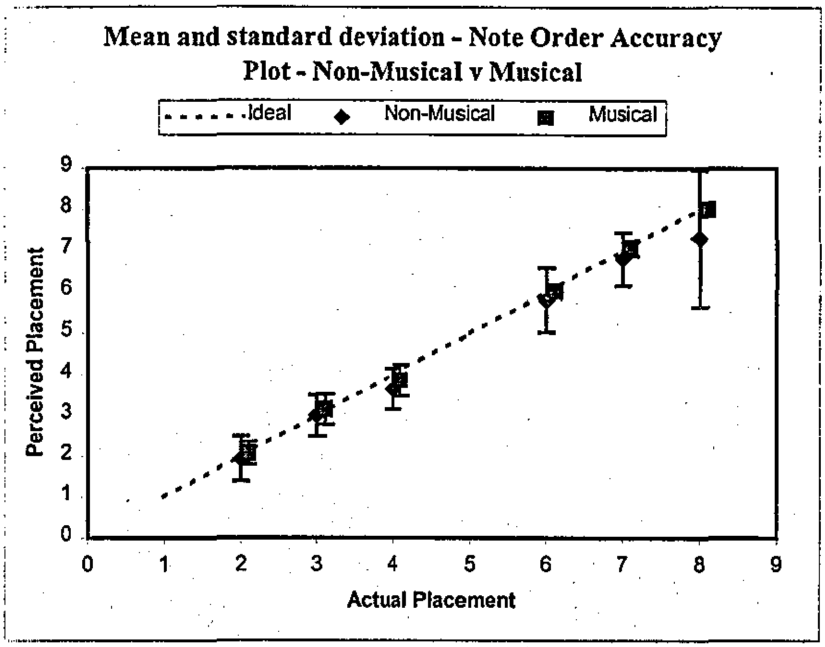


Figure 5.45 – List state note order accuracy – non-musical v. musical.

Musical	1	2	3	4	5	6	7	8
Perceived Placement Mean	-	2.076923	3.153846	3.846154	-	6	7.038462	8
S.D	-	0.27735	0.375534	0.375534	-	0	0.196116	0

Non-Musical	1	2	3	4	5	6	7	8
Perceived Placement Mean	-	1.941176	3	3.647059	-	5.8	6.787879	7.294118
S.D	-	0.555719	0.5	0.492592	-	0.774597	0.649883	1.649421

Figure 5.46 – Table of list state note order accuracy – non-musical v. musical.

The data for the 'musical' test subjects given in Figure 5.46 suggest that they might perform with greater accuracy than the group as a whole. The data obtained from the 'non-musical' test subjects shows reduced levels of accuracy when compared to the 'musical' group for placing incorrectly ordered elements. Once again the greatest error occurs when successive erroneous elements are played. The general accuracy of the test subjects within this sub-group is relatively high with most of the element placements being perceived with little error.

Figure 5.48b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the scores obtained for the difference between perceived erroneously placed elements compared to the true erroneously placed elements. The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving erroneously placed elements.

H_1 : The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving erroneously placed elements.

	PLACE2	PLACE3	PLACE4	PLACE6	PLACE7	PLACE8
Mann-Whitney U	86.500	101.500	88.500	78.000	65.500	84.500
Wilcoxon W	177.500	192.500	179.500	169.000	156.500	175.500
	-1.44	-.543	-1.201	-2.096	-2.337	-1.842
Asymp. Sig. (1 tailed)	.07	.295	.115	.018	.010	.033

Figure 5.48b – Table of test statistics for each perceived placement, 'non-musical' v. 'musical'.

The null hypothesis can be accepted for erroneously placed elements in all but the last three positions suggesting that there is no significant difference between 'musical' and 'non-musical' listeners when perceiving erroneously placed elements in positions 2, 3 and 4 of the list. In contrast, the null hypothesis can be rejected at the 5% level of significance for erroneously placed elements in positions after position 4 suggesting that there is significant difference between 'musical' and 'non-musical' listeners when perceiving erroneously placed elements towards the end of the list. This difference in significance is due to the increase in complexity as the positions of erroneously placed elements become further away from the start of the list. This suggests that 'musical' listeners are more adept at perceiving locations further into the scale. It is important to determine how the group perform in general in this experiment.

The data given in Figure 5.48c shows the results of a Chi-Squared test applied to the results obtained for the perceived erroneously placed elements compared to the true erroneously placed elements. The hypotheses are:

H_0 : Users are not capable of identifying a descent in pitch that denotes an out of place numerical value. In particular, they are not capable of understanding musically represented out of place numerical values within data lists.

H_1 : Users are capable of identifying a descent in pitch that denotes an out of place numerical value. In particular, they are capable of understanding musically represented out of place numerical values within data lists.

	PLC2	PLC3	PLC4	PLC	PLC	PLC8
Chi-Square	112.179	116.105	94.545	151.867	130.568	126.676
df	7	7	7	7	7	7
Asymp. Sig.	.000	.000	.000	.00	.000	.000

Figure 5.48c – Table of test statistics for each perceived out of place element.

These data show that the probability of obtaining the scores gathered from the users for the identification of out of place elements at random are extremely low. From these data the null hypothesis can be confidently rejected concluding that users are capable of identifying descents in pitch and in particular that they are capable of identifying musically represented out of place numerical values in data lists. This is encouraging since it suggests that users might be capable of understanding musically auralised sorting algorithm lists.

5.7. List manipulation experiments

5.7.1. Experiment construction.

In the previous experiment, listeners' perception of the state of lists of numbers was tested by measuring how accurately the test subjects identified incorrectly placed individual elements. The next step towards testing algorithm execution and state is to introduce some manipulation of the numerical data lists. The manipulation employed in this series of experiments is the swapping of incorrectly placed neighbouring elements, the same sorting mechanism as that utilised by the Bubble Sort algorithm.

Thirty subjects were asked to listen to sequences of musical notes within a bounded diatonic octave scale beginning at 'Middle C'. Each test comprised two components, a checking phase as with the previous experiment followed by a sorting phase. Different timbres were chosen for each of these components. The timbres were chosen from distinct families as suggested by Rigas and Alty [158] in order maximise disambiguation of the two components. The timbre employed for the checking phase was a flute that was placed to the left of the stereophonic field with no reverberation or chorus added. The timbre for the sorting phase was an acoustic grand piano placed in the centre of the stereophonic field with no added chorus or echo. The auralisation of the sorting phase has been employed to distinguish between the actions of testing the list and sorting the list. A descent in pitch would indicate out of place elements. Also present in the sorting phase was a trumpet to indicate the swapping action of the incorrectly placed elements. This provides a second cue for identifying out of place elements and a cue for the manipulation of the data. The Subjects were told that each of the eight notes within the bounded scale was mapped to the numbers one to eight. Upon listening to each test, the subjects would first hear the flute checking through the list. This would be followed by the progression of the piano through the list where a swap would be denoted by a trumpet triad. All test subjects were shown and played the example in Figure 5.49 that represented the swapping of two elements after a descent in pitch indicated that element 4 should be placed before element 3.

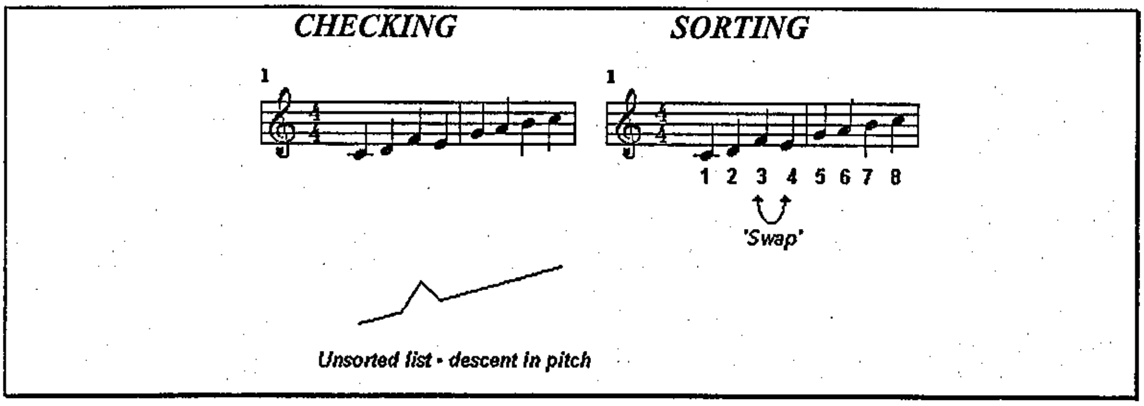


Figure 5.49 – Checking and swapping example – list manipulation experiments.

Following the test example, listeners were played five further instances of checking and swapping where they were asked to identify which elements had been swapped. In comparison to the previous experiment, where the only cue that denoted erroneous placement was a descent in pitch, this experiment provided two cues. The first cue was the descent in pitch during the checking phase, the second cue incorporated a descent in pitch in the sorting phase directly followed by the trumpet triad denoting the occurrence of a swapping of elements. Test subjects were asked to identify the elements that were swapped by circling an element pair within a list. The swaps occurred between positions 1 and 7 with the omission of position 4 as no swap occurred in the algorithm derived examples. Test comprised experiments that contained single swaps and multiple successive swaps.

5.7.2. Results and analysis

Group 1 were used in this series of experiments. Figure 5.51 shows the users' perception of each of the swapped element pairs within the partially sorted lists. The results show that the error distribution is fairly even across the range of positions. As with the previous experiment, multiple erroneous elements were placed (and in this case swapped) in the final portion of the list. In this case, however, there is no noticeable decrease in the users accuracy of identifying the swapping of these latter elements. This may, in part, be due to the addition of a second and more distinct cue that highlighted the swapping of the incorrectly placed elements and hence yielded a second cue as to the positional location within the scale. This suggests that the listeners are able to identify the trumpet triad with data manipulation and have also used it to help localise the position of the out of place elements within the list. The triad employed has been used to identify the occurrence of a swap, this has been re-enforced by the use of a distinct timbre.

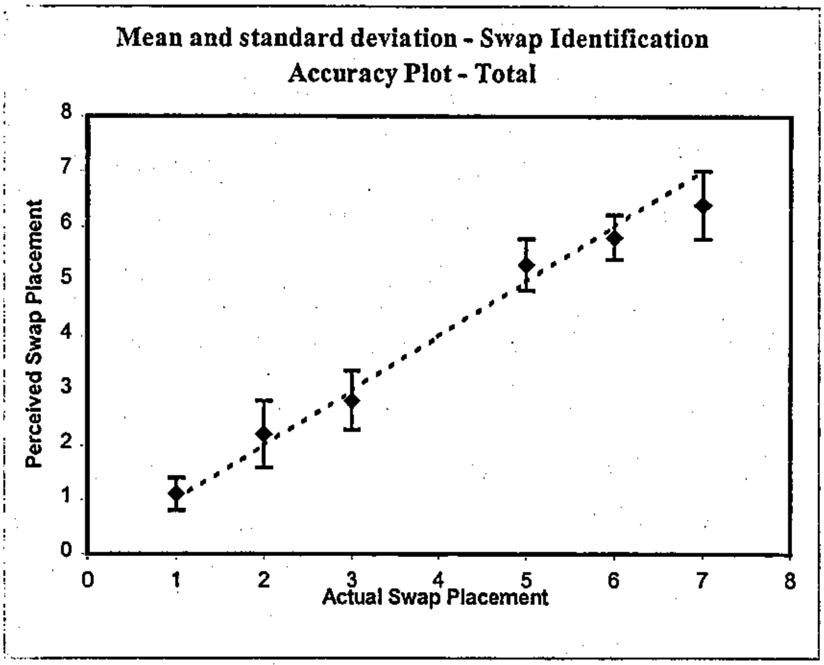


Figure 5.51 – Swapping identification accuracy – entire group.

Actual Swap Element	1	2	3	4	5	6	7
Perceived Swap Mean	1.1	2.2	2.824561		5.3	5.8	6.386364
S.D	0.305129	0.610257	0.53861		0.466092	0.403376	0.618171

Figure 5.52 – Table of swapping identification accuracy – entire group.

Figure 5.53 shows how musically trained and untrained listeners performed in this series of experiments. The results for the 'musical' test subjects given in Figure 5.54 suggests that the 'musical' group might be capable of identifying the swapped elements with greater accuracy than the group as a whole. The data clearly shows that the musically trained listeners identified a large majority of the incorrectly placed elements including some of the successive multiple erroneous elements. These successive multiple erroneous elements have, in this instance, shown to cause a small amount of confusion with the musically trained test subjects.

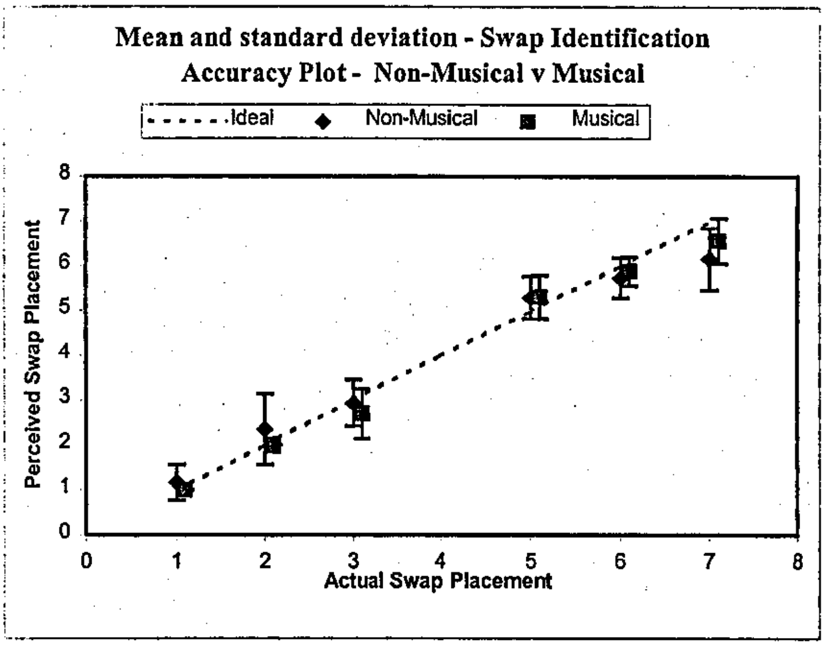


Figure 5.53 – Swapping identification accuracy – non-musical v. musical.

Musical	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5	Pos 6	Pos 7
Perceived Swap Mean	1	2	2.692308	-	5.307692	5.884615	6.56
S.D	0	0	0.549125	-	0.480384	0.325813	0.50637

Non-Musical	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5	Pos 6	Pos 7
Perceived Swap Mean	1.176471	2.352941	2.935484	-	5.294118	5.735294	6.157895
S.D	0.392953	0.785905	0.512216	-	0.469668	0.447811	0.688247

Figure 5.54 – Table of swapping identification accuracy – non-musical v. musical.

The data given in Figure 5.54 also suggests that the 'non-musical' test subjects perform with a reduced accuracy compared to the 'musical' listeners when placing incorrectly ordered and swapped elements. It can be seen from these data that no noticeable decrease in accuracy occurs when successive multiple element swaps take place. The general accuracy of the test subjects within this sub-group is relatively high with most of the element placements being perceived as with little error.

Figure 5.56b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the scores obtained for the difference between perceived erroneously placed and swapped elements compared to the true erroneously placed and swapped elements. The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving erroneously placed and swapped elements.

H_1 : The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving erroneously placed and swapped elements.

	SWAP1	SWAP2	SWAP3	SWAP5	SWAP6	SWAP7
Mann-Whitney U	91.000	91.000	98.500	109.000	92.500	39.500
Wilcoxon W	182.000	182.000	189.500	262.000	183.500	130.500
Z	-1.570	-1.570	-.562	-.079	-.935	-3.052
Asymp. Sig. (1-tailed)	.058	.058	.288	.468	.175	.001

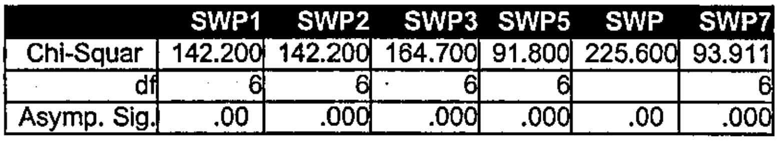
Figure 5.56b – Table of test statistics for perceived placement/swap, 'non-musical' v. 'musical'.

It can be seen from the data given in Figure 5.56b that the null hypothesis cannot be rejected for erroneously placed and swapped elements in the first six positions suggesting that there is no significant difference between 'musical' and 'non-musical' listeners when perceiving erroneously placed and swapped elements in the first six positions of the list. In contrast, the null hypothesis can be rejected at the 0.1% level of significance for erroneously placed and swapped elements in the final position in the list (where the successive swaps occur) suggesting that there is a significant difference between 'musical' and 'non-musical' listeners when perceiving successive erroneously placed and swapped elements towards the end of the list. This difference in significance is due to the increase in complexity as successive swaps occur towards the end of the list. This suggests that 'musical' listeners are more adept at perceiving successively swapped elements than 'non-musical' listeners. It is important to see how the group perform in general in this experiment.

The data given in Figure 5.56c shows the results of a Chi-Squared test applied to the results obtained for the perceived erroneously placed and swapped elements compared to the true erroneously placed and swapped elements. The hypotheses are:

H_0 : Users are not capable of identifying descents in pitch and metaphors that denote swapping. In particular, they are not capable of understanding musically represented out of place and manipulated numerical values within data lists.

H_1 : Users are capable of identifying descents in pitch and metaphors that denote swapping. In particular, they are capable of understanding musically represented out of place and manipulated numerical values within data lists.



	SWP1	SWP2	SWP3	SWP5	SWP	SWP7
Chi-Squar	142.200	142.200	164.700	91.800	225.600	93.911
df	6	6	6	6		6
Asymp. Sig.	.00	.000	.000	.000	.00	.000

Figure 5.56c – Table of test statistics for each perceived out of place and swapped element.

These data show that the probability of obtaining the scores gathered from the users for the identification of out of place and swapped elements at random are extremely low.

From these data the null hypothesis can be confidently rejected concluding that users are capable of identifying descents in pitch and metaphors that denote swapping. In particular, they are capable of identifying musically represented out of place and swapped numerical values in data lists. This is encouraging since it suggests that users might be capable of understanding musically auralised sorting algorithm lists.

5.8. List shape progression experiments

5.8.1. Experiment construction

In this set of experiments twenty subjects were asked to listen to sequences of musical notes of varying length within a bounded diatonic octave scale that corresponded to the numbers one to eight. The lists corresponded to sequences of notes that might be produced during the execution of Bubble Sort, Exchange Sort, Quick Sort, Inside-Out and Outside-In Sorting Algorithms. List produced by the Selection Sort were not used in this experiment as the Exchange Sort already provides these types of algorithm-derived lists. Five sequences were played – comprising 8, 5, 10, 8, and 8 notes. The objective of the experiment was to determine if listeners could identify the progressive changes in shape that occur during algorithm execution. The timbre employed was an acoustic grand piano, which was placed in the centre of the stereophonic field with no reverberation or chorus added. The bounded diatonic scale started at 'Middle C' and ascended by one diatonic octave (eight notes). In order to create a context the scale was first played once before each note sequence. Subjects were told that each of the eight notes within the bounded scale were mapped to the numbers one to eight and that the sequences therefore represented lists of eight numbers.

Upon listening to each set of sequences, the subjects were asked to interpret the shapes of each of the lists within each sequence set. After having described the shape of each list, test subjects were then asked to describe what had progressively happened to the shape of the list in the given set of sequences.

In order to better explain the concept to the listeners, a simple example of five sequences was played three times with a description of each list shape followed by a description of what has progressively happened to the shape of the list. Figure 5.57 shows the information that was presented to subjects during the experiments. Each test listener was also provided with the information given in Appendix K.

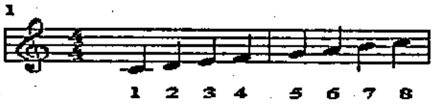



Figure 5.57 – Example scale.

Following an example, each of the twenty test subjects was played five of these tests each of which comprised five tonal sequences within the bounded diatonic octave scale starting at 'Middle C'.

5.8.2. Results and analysis

Group 2 were used in this series of experiments.

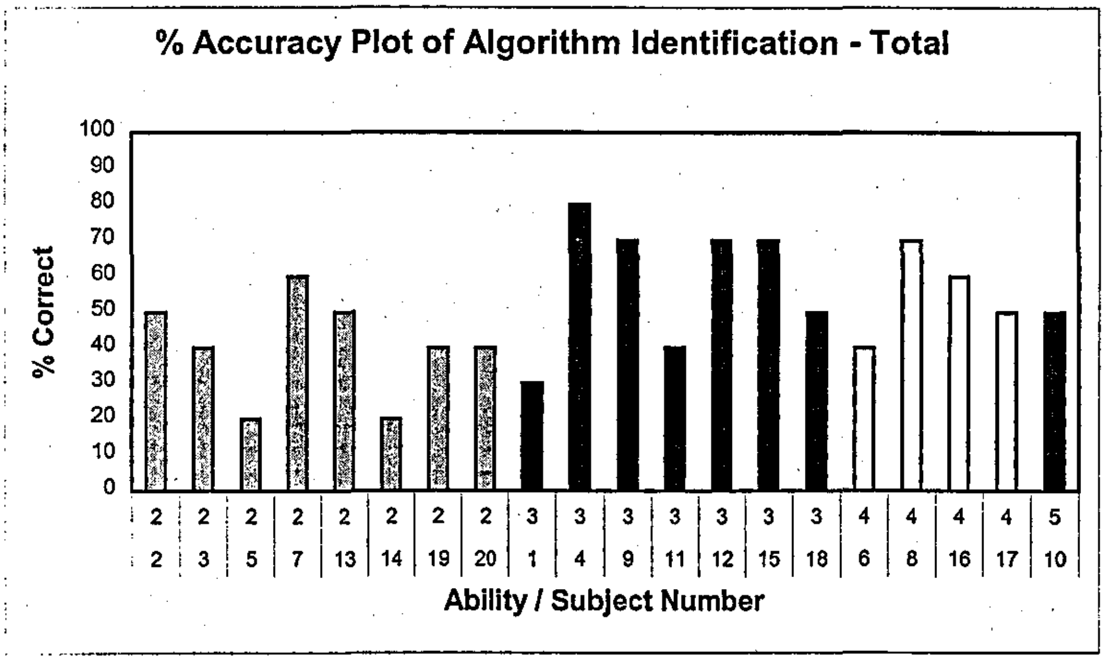


Figure 5.62 – Algorithm identification accuracy – entire group.

The diagram in Figure 5.62 above shows how the entire group of test subjects performed when describing what had progressively happened to each of the five sets of sequences. The graph is ordered and colour coded in terms of musical ability with the least musical subjects being plotted to the left and those with the greatest musical ability being plotted to the right (abilities from 2 to 5). The graph suggests little difference between musically trained and musically untrained listeners. Figure 5.62b shows the results of the Mann-Whitney non-parametric test applied to the scores obtained for the perception of shape progression from 'musical' listeners compared to 'non-musical' listeners. The hypotheses are:

H_0 : There is no significant difference between 'musical' and 'non-musical' listeners when perceiving the progression of musically represented algorithm states.

H_1 : There is a significant difference between 'musical' and 'non-musical' listeners when perceiving the progression of musically represented algorithm states.

	SCORE
Mann-Whitney U	21.500
Wilcoxon W	57.500
Z	-2.086
Asymp. Sig. (1-tailed)	.018

Figure 5.62b – Table of test statistics for algorithm progression identification, 'musical' v. 'non-musical'.

From the data given, the null hypothesis can be rejected at the 5% level of confidence concluding that there is significant difference between 'musical' and 'non-musical' listeners when perceiving the progression of musically represented algorithm states. In particular, 'musical' listeners performed better than 'non-musical' listeners. This might suggest that musically untrained listeners may not be capable of understanding sorting algorithm auralistaions. However, it is important to indicate that although this data shows that musical training has a significant effect, it do not discount musically untrained listeners from being able to understand the sorting natures. This data simply shows that musically trained listeners are better at the task. It does mean that musically untrained listeners are incapable of performing the task. It is therefore important to determine how the group perform in general.

The overall average score for the perception of musically represented algorithm sorting natures is 50%. By using the binomial expansion the probability of obtaining such a score at random can be calculated. Where the number of successes (r) is 5 and the number of trials (n) is 10. The probability of the occurrence of **r** successes in **n** trials is given by the binomial distribution:

$$P(r\ successes) = \frac{n!}{(n-r)!\,r!} q^{n-r} p^{r}$$

Where **p** represents the probability of success and **q** represents the probability of failure.

This yields a probability of 0.026, which strongly suggests at a level of 97.4% that the group of listeners are capable of identifying the algorithm sorting natures when represented musically.

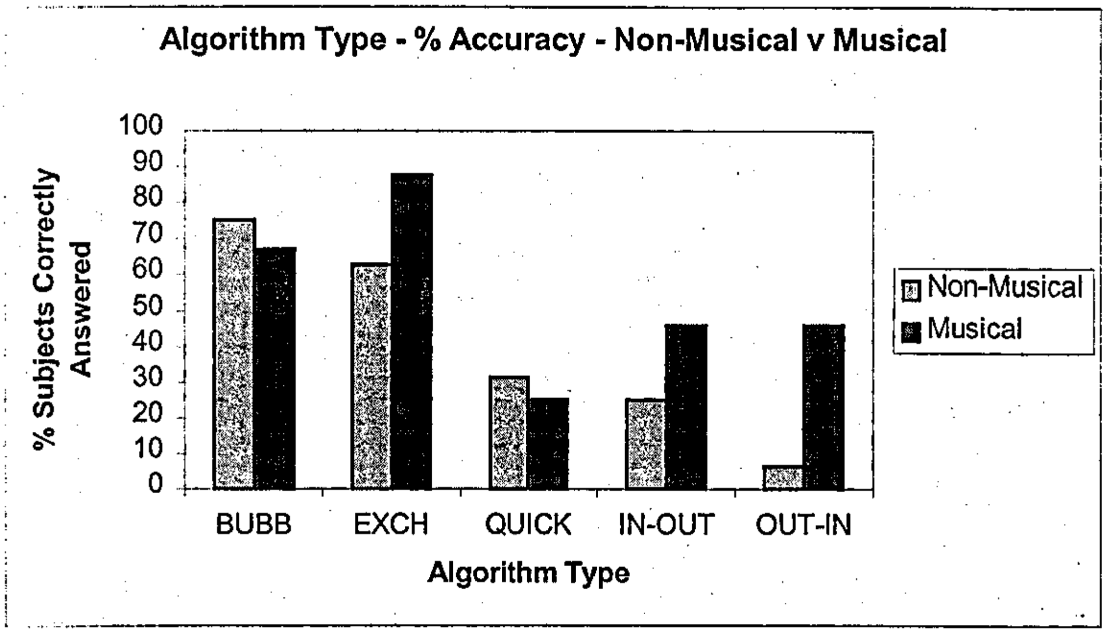


Figure 5.62c - Algorithm identification accuracy by algorithm type – 'musical' v. 'non-musical'.

This difference can also be seen in Figure 5.62c, which represents how musically trained and musically untrained listeners performed in this series of experiments on each of the information source types. The data shows that 'musical' test subjects show improved identification and understanding of the nature of the Exchange Sort, Bucket Sort (In-Out) and Bucket Sort (Out-In) algorithms. The results for Bubble Sort and Quick Sort algorithms show little difference between 'musical' and 'non-musical' listeners. However, it is interesting to note that in the case of the Bubble Sort auralisation musically untrained listeners appear to perform better than musically trained listeners.

In terms of information type, it is important to highlight whether certain types of algorithms are more easily understood by the group of test subjects. Figure 5.63 shows how the entire group performs as whole on each type of list shape progression.

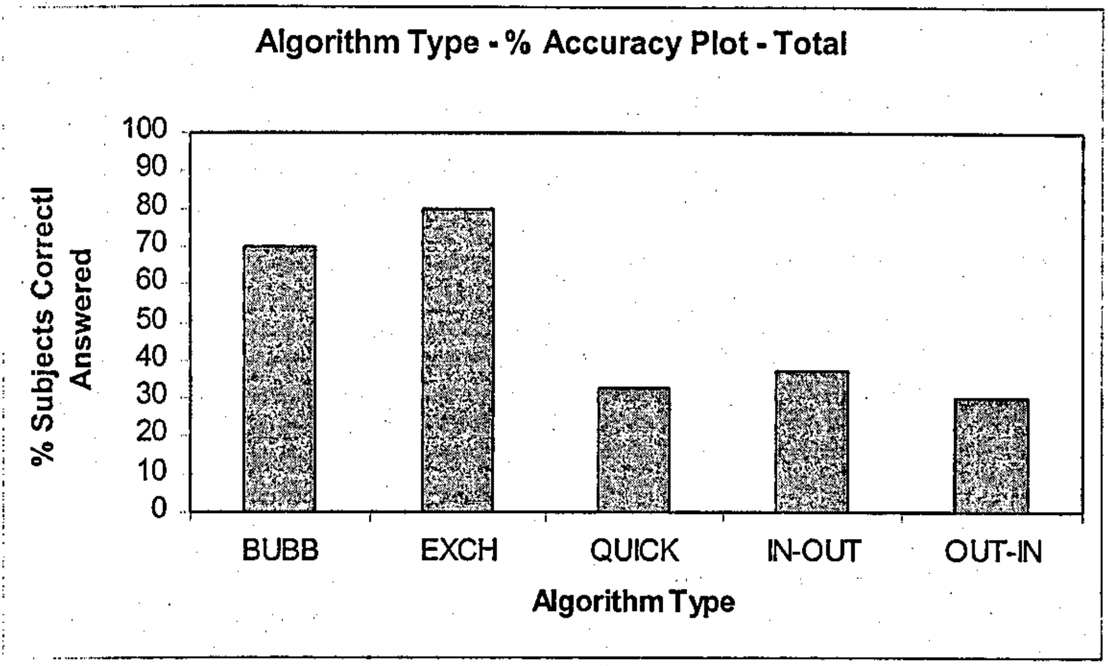


Figure 5.63 – Algorithm identification accuracy by algorithm type – entire group.

These data suggest that the less complex algorithms are more easily understood. Both the Bubble Sort and the Exchange Sort share the common feature that the sorted list 'grows' from one end of the list boundary, which can be seen as a single easily identifiable 'anchor' point from which the list grows. Given that complex algorithms are less understood by the listeners, some prior training might be of benefit. This suggests that the simpler algorithms are learnt more quickly and with greater ease than complex algorithms. It may also be that the difficulty associated with identifying the nature of the least understood algorithms is not attributed to the complexity of the functionality of the algorithm itself, but rather the mechanism by which it has been auralised. The mapping of pivot points and sub-buckets is harder to achieve than sequential swapping. This suggests that at this level it is not the complexity of the algorithm that is problematic. Instead, it suggests that certain algorithms do not translate well musically as their features and natures are not easily represented by musical metaphors. There is a clear case for learning the mappings in this instance. However, this is not of real concern to this thesis.

In previous experiments in this thesis it was found that tones, out of place elements and swapped elements were more easily identified when they were near to the lower and upper boundaries of the diatonic octave scale starting at 'Middle C'. In contrast, tones, out of place elements and swapped elements near the middle of the scale were not so easily identified as these positions are in the area of greatest ambiguity. This is mirrored in the Quick sort algorithm where the list is progressively sorted into increasingly smaller groups. In this instance, there is more than one 'anchor' point. As the list becomes increasingly sorted, the number of 'anchor' points increases and their positions are always changing. Due to the varying nature of these 'anchor' points, understanding the Quick sort algorithm is more difficult than understanding the Bubble Sort and Exchange Sort algorithms (which in contrast only have a single constant fixed 'anchor' point throughout the entire auralisation). The auralisation for the Quick Sort algorithm utilised the available information as effectively as would allow. The nature of the varying positions of the pivot points and the direction of the placement of elements into sub-buckets imposed limitations on the auralisation, as this information did not translate well sequentially. This information might be understood with greater clarity if extra cues were used to further disambiguate the information. Since the Quick Sort algorithm segments the entire list into increasing groups of smaller sub-lists the spatial placement of the pivots and the buckets in a 3D audio environment might help to clarify the information presented. Essentially, the Quick Sort algorithm operates in a sequential manner. This would suggest its suitability for mapping into the temporal domain. However, the visualisation of sorting elements into sub-buckets, to the left or right of a pivot, suggests that it might be more suitable for mapping into the spatial domain. These concerns suggest a spatially enhanced version of this auralisation, utilising both the spatial and temporal domains, might better represent the sorting nature of the algorithm.

The same can be seen for the Inside-Out and Outside-In sort algorithms where the Quick Sort is used for the first pass. In this instance, there are two 'anchor' points in the first pass followed by one 'anchor' point for each of the sub-sorting algorithms in subsequent sorting passes. With the Inside-Out sort, the 'anchor' point is a single point in the centre of the list, but this is in the area of ambiguity previously identified when compared to 'anchor' points that are placed at the low and high boundaries of the context scale. In contrast, for the Outside-In sort, both 'anchor' points are at the high and low boundaries. This suggests that the increase of the number of 'anchor' points has impaired the understanding of the sorting nature of the algorithm. Test subjects have shown to be able to identify each of the sub-sorting algorithms when presented autonomously. In contrast, when these algorithms are combined in a more complex manner the understanding is significantly reduced. This may be attributed to the use of the Quick Sort algorithm even though this has only been used in the first pass in order to split the list into two sub-lists. The majority of the sorting is achieved by the dominant Bubble Sort and Exchange Sort algorithms but are not as clearly identified by the test subjects.

As with the previous experiments that involved the perception of shapes of tonal sequences, no appreciable difference in perception accuracy between those of 'good' drawing ability and those with 'less than good' drawing ability was observed. In general, although musical training was found to have a beneficial effect, the data clearly shows that both 'musical' and 'non-musical' listeners are capable of perceiving and understanding the shape progression of algorithm generated lists.

5.9. Conclusion

For the pitch test experiments, the results have shown that there is a significant difference between the 'musical' and 'non-musical' groups when perceiving tones that are close to the boundaries of the context scale. These data further showed that there is no significant difference between the groups when perceiving tones that fall into the area of greatest ambiguity in the middle of the context scale. The use of an extra cue such as spatial location may improve the perception results in this case. If this additional cue further aids disambiguation then the 'non-musical' group might show an improvement in accuracy when perceiving tones close to the boundaries of the context scale decreasing the difference between the two test groups. The data obtained for the pitch interval test experiments showed that for small intervals (less than 2) there is no significant difference between 'musical' and 'non-musical' listeners. This difference becomes significant and increases in relation to the increase in interval size. Again, the use of spatial location cues might further disambiguate the information and increase the accuracy when perceiving pitch intervals.

For the shape perception experiments using short musical sequences with musical timing the results showed significant difference between the 'musical' and 'non-musical' listeners for all shapes. In contrast, the series of experiments using short tonal sequences with no musical timing showed that there is no significant difference between the two groups when perceiving the shapes. This difference in significance suggests that when tonal sequences have musical timing applied to them, making more '*musical*', the 'musical ' group of test subjects tend to perform with greater accuracy than the 'non-musical' group, subsequently suggesting that 'musical' listeners are more capable of exploiting musical timing. When this musical timing is removed, as is employed for algorithm state auralisation within this thesis, the data shows that there is no significant difference between the two groups when perceiving tonal shapes (or musically auralised algorithm list states).

The data obtained for the series of experiments concerned with identifying out of place elements in an otherwise ascending list of numeric elements showed there was no significant difference between the two groups of 'musical' and 'non-musical' listeners for elements identified in approximately the first half of the list. For the remainder of the list, the difference between the two groups becomes significant due to the increasing complexity as the positions of erroneously placed elements move further up the context scale. Again the use of an extra cue such as spatial location may increase location accuracy results, since the information required from the listeners is positional.

Similar results were observed for the identification of erroneously placed and swapped elements in an otherwise ascending list of numerical elements. For the majority of positions (all except the last) no significant difference was observed between the 'musical' and 'non-musical' test groups. The only significant difference between the two groups was observed when successive multiple erroneously placed and swapped elements occurred in the final position. These data suggest that multiple successive swaps increases misunderstanding of swap occurrence and location. In comparison to the results obtained in the previously described out of place elements experiments, the same data suggests that the addition of the extra cue (the sound of the elements swapping) aids localisation and reduces the observable difference between 'musical' and 'non-musical' listeners.

The results of the experiments investigating the perception and understanding of algorithm derived list shape progression showed that 'musical' listeners tend to perform significantly better than 'non-musical' listeners. The data also showed that algorithms that progress with easily identifiable and constant anchor points from which the sorted list grows are more easily understood than the more complex algorithms that produce anchor points which are constantly moving and changing in quantity.

In general it has been shown that musical training does have some affect on the perception of musical sequences and pitch. However, the results have shown that both musically trained and untrained listeners are quite capable of discerning pitch and understanding shape and musically represented numerical data and that the difference between the groups depends upon the complexity of the musical structure. These results are encouraging, since the content of the tests form the basic building blocks of algorithm auralisation.

Chapter 6

SIMBAA – A musical auralisation tool

In order to support the suggested investigations into algorithm auralisation using music, it was decided to develop a system to allow experimenters to easily auralise the execution of a number of algorithms. The main design objective of this system was to permit the mapping of key events and objects in algorithms to musical structures and timbres. Like Vickers' CAITLIN system, the program code is "marked" in order to auralise key points, and this allows musical auralisation of its progression in real-time. After processing a program through the system, the actions of the algorithm and the state of the variables will be musically audible. It is important to clarify that this tool has been developed purely to facilitate the auralisations required for the experimentation required in this thesis. It does not aim to be commercial auralisation tool.

6.1. SIMBAA design and considerations

The system, entitled SIMBAA (System for the Implementation of Music Based Algorithm Auralisation), is based upon a combination of Vickers' CAITLIN [181] and Brown and Hershberger's ZEUS [45, 46]. The key objectives of SIMBAA are to:

- Musically auralise any algorithm in real-time.
- Allow musical attributes to be allocated to events / objects.
- Provide a toolbox for working on existing algorithm program code.
- Exploit the features of MIDI. (Timbre, stereo placement, echo, volume, chorus etc.)
- Allow the user to play the auralisation at his/her own speed, permitting the user to alter the information presentation rate to match their own processing ability. This is also useful for facilitating different levels of abstraction during playback.
- Permit the identification of key events / objects by selectively masking the output.
- Allow on-line adjustment of tempo to give different levels of abstraction during auralisation.

Like CAITLIN, SIMBAA is a pre- processor for program code but it processes C/C++ rather than PASCAL. SIMBAA needs more user input than CAITLIN, since it requires the user to include a header file that contains the necessary library routines for auralisation. The original code also needs to be 'marked - up' with the necessary auralisation calls (adding one-line function calls at the desired steps of the routine). Auralisation is achieved in real-time during algorithm execution.

The SIMBAA system provides the following features:

- Library routine calls to -
    - Play tones from the chromatic scale for given octaves and notes.
    - Play chords following a root note and octave, standard, first inversion and second inversion triads (21 chord possibilities for one root note).
    - Play a cadence for given root notes.

- Pre- execution controls include -
    - Instrument-to-channel assignment to allow the user to choose timbres that are aesthetically pleasing to their own preferences.
    - Channel pan - allowing instruments to be placed in auditory space to help disambiguate between information members.
    - Channel volume - allowing the user to highlight or suppress certain aspects of the information, this helps to focus on key events or actions.
    - Channel echo level - this feature can give a feeling of space to the instrument ensemble and support the panning feature.
    - Channel chorus depth - permitting specific actions or events to be emphasised over other instruments.
    - Global adjustment of the octave offset - this shifts the entire ensemble up or down in octave steps, much like the register feature used in Sumikawas' motifs for Earcons [174, 33].

- Global online controls to -
    - Toggle muting of each channel in real- time to permit key events or actions to be identified and understood in solitude, or to permit verbose instruments to be excluded. Supporting the information masking feature of the system.
    - Key shift each channel in semitone steps to allow the system to be 'tuned' to the users' own preferences, also to help align the 'orchestra' to give a better overall feel to the auralisation.
    - Adjust a global tempo to give different levels of abstraction during routine execution.

- General features -
    - Choice of internal OPL3 FM synthesizer or,
    - External General MIDI.
    - Up to 8 instrument polyphony with full control.
    - Environment saving allowing the system to retain a users preferences of the variables.

SIMBAA makes no prior judgement about the musical ability of the user. However, being musical would facilitate the production of more 'musical' presentations. The output of SIMBAA can be fed into any multi-timbral device. The experiments in this thesis employed the use of external General MIDI driving a 16 channel multi- timbral Roland Boss DS 330 synthesizer.

6.2. Early experimentation on Algorithms using SIMBAA

It was decided first to auralise the simple Bubble Sort algorithm. This kind of algorithm provides a useful testing ground for the use of music to convey events through time. By auralising it with the SIMBAA system it will be possible to test the usefulness of the system and the viability of using music in algorithm understanding. As with any interface design, success depends upon the choice of mapping of the algorithm to the music [3]. The first step is to identify the key information from the Bubble Sort algorithm that is to be auralised. These features were identified in Chapter 4. This preliminary auralisation does not include the auralisation of the iteration count. This is because this study is aimed at bringing together the components examined in Chapter 5. More detailed experiments follow that auralise all of the features identified in Chapter 4. By working through the algorithm in Chapter 4 the features for this preliminary auralisation can be summarised:

1. The current state of the list.
2. Progression of the algorithm through the list of elements.
3. The swapping of elements.
4. Successful termination.

As with previous attempts to auralise the Bubble Sort [3], the list of numbers was converted to a sequence of notes in the diatonic scale. As the goal of the algorithm is to sort the elements into an ascending order then successful termination will be achieved when the current state of the list is heard as an ascending succession of notes. The diatonic scale was chosen over notes in the chromatic or pentatonic scales due to the early experimentation carried out by Alty [3], which highlighted listeners' preferences to the presentations within the diatonic scale. All subsequent experiments and auralisations in this thesis are performed with the same diatonic scale.

The auralisation code is termed 'ghost code', this is due its transparent nature. This 'ghost code' is unseen by the algorithm and is only of use to the SIMBAA auralisation tool. The nature of the integration technique adopted by the SIMBAA system enables the desired code to be auralised without any adverse effects on its original functionality provided that the original program code is not critically time dependent.

The mapping of music to the four points of interest has been chosen to be as communicative as possible. Three different timbres were employed, these are not changeable by the listener during experimentation. The three instruments were an acoustic grand piano, a flute and a brass ensemble (each instrument being from a distinct family to further aid disambiguation). In order to help disambiguate the information, the use of auditory placement was also used. The flute was placed centrally in the auditory field, the piano to the left and the brass ensemble to the right.

Disambiguation of information in the algorithm was further achieved by the use of -

- Contrasting timbre using a piano, a flute and a brass ensemble.
- Harmony using a major triad.
- The use of auditory space by employing instrument placement via stereophony.

The auditory mappings were:

1. The current state of the list 'Play Entire List' - this auralisation was achieved by mapping element values to pitch (a metaphor). The chosen instrument here was a flute.

2. Progression of the algorithm through the list 'Play Current Element' - the chosen mapping here was an acoustic grand piano, again the element values were mapped to pitch.

3. The swapping of elements 'Play Swap Structure' - this is heard in parallel with the ascending acoustic grand piano, the structure is a brass ensemble playing a major triad. The first note is an element to pitch mapping of the higher value in the current pair, the second note is an element to pitch mapping of the lower note in the current pair and finally the third note is a repetition of the first note.

4. Successful termination 'Play Successful Termination' - this auralisation was achieved by again using the brass ensemble, but this time it was used to produce a simple yet suggestive 'Ta - Da' sequence.

6.2.1. Preliminary findings

Preliminary experiments were performed with the SIMBAA system using the Bubble Sort algorithm on ten subjects, five male and five female. Each listener answered a series of questions about their musical interest, exposure and training. No attention was paid to their computing skill but all subjects had no prior knowledge of algorithms. From this information it was determined that each subject had an average musical ability, by this it is meant that the subjects were not trained musicians.

The subjects were told about the nature of the Bubble Sort algorithm:

> **algorithm** (*say* alga-ri*th*'m) *noun*
>
> *Maths:* a clearly-defined sequence of operations for solving a particular mathematical problem.

"The basis of the Bubble Sort algorithm is to repeatedly iterate through a list of elements comparing every adjacent pair of elements and swapping them if they are not in the correct relation. When an iteration takes places without any pairs of elements being swapped then the list is known to be sorted into numerical order and the algorithm can successfully terminate."

Via informal verbal feedback, all ten subjects gave descriptions about the nature of the Bubble Sort algorithm suggesting that they understood the audible process of the algorithm after being presented with the auralisation for the fourth time. The first time that the algorithm was played 8 of the subjects requested that it be played slower the next time around. On the fourth pass, 6 of those 8 were comfortable with reverting back to the normal tempo. The listeners were asked a series of question about the auralisation and answers were entered into blank workbooks. The subjects were able to extract the following quantitative and qualitative information given in Figure 6.3 from the auralisation procedure:

Attribute identified	**No. of subjects**
• The number of elements in the list.	10
• The number of passes before successful termination.	8
• The act of swapping elements. (Descriptive answers accepted).	9
• The amount of 'swap' occurrences within each pass / iteration.	4
• The 'Test and Sort' nature of the algorithm.	9
• The successful termination. (Descriptive answers accepted).	10

Figure 6.3 - Bubble Sort algorithm auralisation information extraction accuracy.

Subjects had no direct interaction with the SIMBAA system controls. The results were encouraging so a more detailed set of experiments was carried out.

6.3. Further experimentation with the Bubble Sort algorithm

A further series of experiments were carried out that employed auralisation of the Bubble Sort algorithm in the same manner as was used in the previously documented study. The differences between the two sets of experiments were:

- Increased test subject group size to thirty listeners.
- The addition of a fourth timbre in the form a wooden block, which is played at the beginning of each sorting pass to indicate the iteration count. This has been added to complete the set of features identified in Chapter 4.
- The addition of algorithms that contained errors.

The SIMBAA system was utilised to facilitate the auralisations after the necessary changes were made to incorporate the fourth timbre.

The points of interest that were auralised in this implementation of the bubble sort were:

1. The current state of the algorithm.
2. The iteration count.
3. Progression of the algorithm through the list of elements.
4. The swapping of elements.
5. Successful termination.

The subjects were once again told about the nature of the Bubble Sort algorithm with the following information given in the workbook in Appendix L.

6.3.1. Results

Group 2 were used for this experiment. Accuracy was tested through a set of questions posed to the test subjects concerning the state and execution of the algorithm. The questions are designed to reflect the knowledge required to understand the Bubble Sort algorithm. These test the understanding of list manipulation through swapping neighbouring elements, list checking, iteration and termination. The questions are:

1. How many numbers (elements) are there in the list?....................................
2. How many swaps are there in the first pass?..
3. How many swaps are there in the second pass?...
4. How many swaps are there in the third pass?..
5. How many swaps are there in the fourth pass?..
6. How do you know when elements are out of order?...................................
7. How do you know when the recipe swaps elements?...................................
8. How do you know when the list is sorted?...
9. How many times does the recipe pass through the list?..............................
10. What order is the list sorted into?...

Questions 6 and 7 produced some interesting responses. Some listeners answered both questions with the same answer, that of the trumpet triad. In these cases, users clearly failed to identify the descent in pitch in the list with out of place elements. Moreover they failed to distinguish between the out of place elements cue and the swapping cue. Instead, the stronger cue of the trumpet was used for identification of both. The results are shown in the graph below in Figure 6.5, which has been plotted horizontally in terms of musical ability from least to greatest.

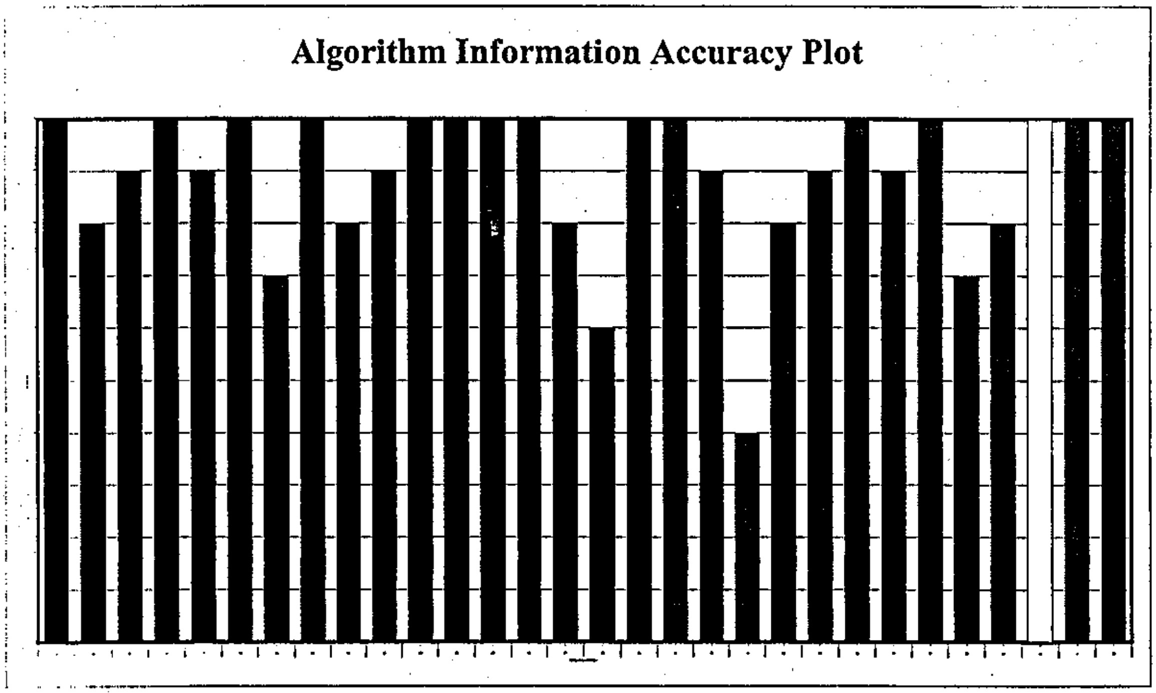


Figure 6.5 – Bubble Sort auralisation information accuracy by musical ability.

This data shows that all of the test subjects with a greater than average musical ability were able to identify all of the state and execution information requested with an accuracy of 100%. By observation of the data, a large proportion of the test subjects with average musical ability were also capable of understanding the information that was questioned.

A check was then made on which types of information were more easily identified and understood. Figure 6.6 shows how successfully the subject group answered each of these questions.

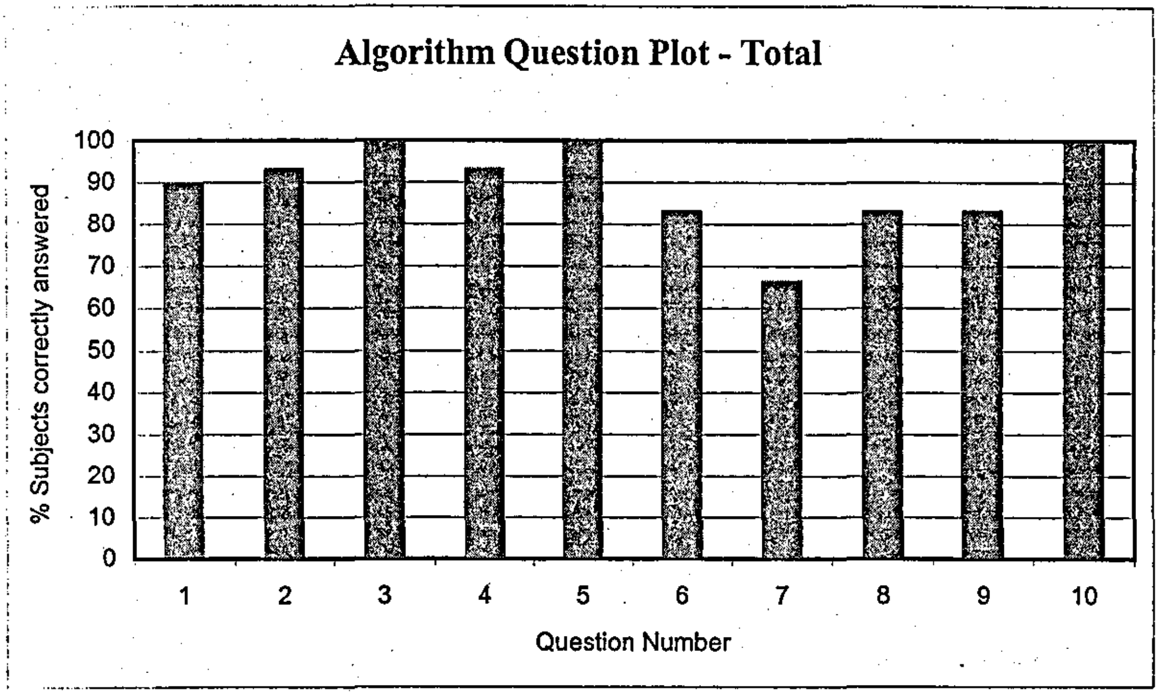


Figure 6.6 – Bubble Sort auralisation information accuracy by question type.

Questions 1 to 5 were requesting information of a quantitative nature and the accuracy level of listeners' identification and understanding was in the range 90% to 100%. In contrast to this, questions 6, 7 and 8 were requesting information of a qualitative nature and the test subjects scored within the range of 67.5% to 82.5%. Question 9 asked for the number of iterations required for the bubble sort algorithm to completely sort the list, and the identification accuracy about this information was 82.5%. The tenth question requested information about the final state of the list, the data shows that all listeners successfully identified this list state to an accuracy of 100%. The final state of the list in this experiment was ascending order.

The data gathered from this series of experiments suggest that information of a quantitative nature was grasped better than information of a qualitative nature. The exception is that the quantitative information did not include absolute data corresponding to the value of each of the elements within the list.

Figure 6.6b shows the results of the Wilcoxon Signed Ranks non-parametric test applied to the scores obtained for the qualitative information extracted from this Bubble Sort auralisation compared to the quantitative information extracted. The hypotheses are:

H_0 : There is no difference in identification accuracy between qualitative information and quantitative information in this auralisation.

H_1 : There is a significant difference in identification accuracy between qualitative information and quantitative information in this auralisation.

	QL - QN
Z	-2.98
Asymp. Sig. (1-tailed)	.00

a Based on positive ranks.
b Wilcoxon Signed Ranks Test

Figure 6.6b – Table of test statistics for algorithm information extraction, qualitative v. quantitative.

From the data given in the above figures, the null hypothesis can be rejected at the 0.1% level of confidence concluding that there is a very significant difference between the extraction and understanding of qualitative information and the extraction and understanding of quantitative information in this instance of the Bubble Sort auralisation. This data together with the scores represented in Figure 6.6 suggest that quantitative information translates better than qualitative information through musical auralisations in this implementation.

The graph shown below in Figure 6.7 shows how 'musical' and 'non-musical' test subjects perform comparatively.

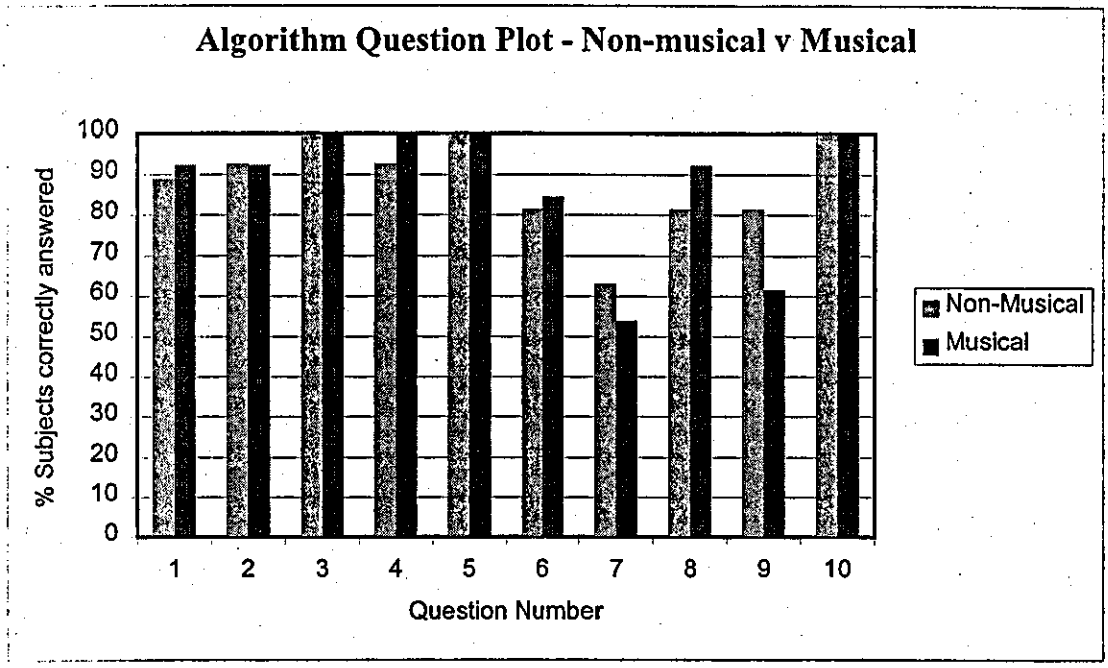


Figure 6.7 – Bubble Sort auralisation information accuracy – 'non-musical' v 'musical' listeners.

These data qualitatively suggest little difference between the performance of 'non-musical' listeners and 'musical' listeners. Overall, the entire group of test subjects identify the information requested about the state and execution of the Bubble Sort algorithm to an accuracy of 86.3%. This suggests that the majority of the requested information was successfully translated.

By using the same binomial expansion used in the previous chapter the probability of obtaining such a score at random can be calculated. Where the number of successes (r) is 8.63 and the number of trials (n) is 10.

The probability of obtaining the correct answers at random for each of the questions is given:

1. How many numbers (elements) are there in the list? List sizes are always presented between 6 and 8 elements. Therefore 5 possibilities gives the probability of giving the correct answer at random is **1/5**.
2. How many swaps are there in the first pass? Played list size is 8 elements so the maximum possible number of passes is 7. Therefore, probability of answering correctly at random is **1/7**.
3. How many swaps are there in the second pass? As question 2. **1/7.**
4. How many swaps are there in the third pass? As question 2. **1/7.**
5. How many swaps are there in the fourth pass? As question 2. **1/7.**
6. How do you know when elements are out of order? Five cues present, iteration count, checking phase, sorting phase, swapping and success. Therefore **1/5.**
7. How do you know when the recipe swaps elements? As question 6. **1/5.**
8. How do you know when the list is sorted? As question 6. **1/5.**
9. How many times does the recipe pass through the list? Maximum iterations in this experiment is 5. Therefore random probability of giving the correct answer is **1/5.**
10. What order is the list sorted into? Possible answers are sorted ascending, sorted descending or remain unsorted. Therefore probability of answering this question correctly at random is **1/3.**

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
p	1/5	1/7	1/7	1/7	1/7	1/5	1/5	1/5	1/5	1/3
q	4/5	6/7	6/7	6/7	6/7	4/5	4/5	4/5	4/5	2/3
n	10	10	10	10	10	10	10	10	10	10
r	8.6333	8.6333	8.6333	8.6333	8.6333	8.6333	8.6333	8.6333	8.6333	8.6333
p(r)	6.129e^{-05}	3.688e^{-06}	3.688e^{-06}	3.688e^{-06}	3.688e^{-06}	6.129e^{-05}	6.129e^{-05}	6.129e^{-05}	6.129e^{-05}	3.930e^{-03}

Figure 6.7a – Table of statistics for each question for Bubble Sort algorithm auralisation.

This yields a total probability of answering the questions at random with a success rate of 86.3% of $6.283e^{-46}$, which strongly suggests that the group of listeners are capable of understanding Bubble Sort algorithm when represented musically.

Figure 6.7b shows the results of the Mann-Whitney non-parametric test applied to the scores obtained by 'musical' listeners for the Bubble Sort auralisation compared to the scores obtained by 'non-musical' listeners for the Bubble Sort auralisation.

The hypotheses are:

H_0 : There is no difference in identification accuracy between 'non-musical' listeners and 'musical' listeners when understanding the Bubble Sort algorithm auralisation.

H_1 : There is a significant difference in identification accuracy between 'non-musical' listeners and 'musical' listeners when understanding the Bubble Sort algorithm auralisation.

	SCOREALG
Mann-Whitney U	103.000
Wilcoxon W	194.000
Z	-.338
Asymp. Sig. (1-tailed)	.367

Figure 6.7b – Table of test statistics for algorithm information extraction, 'non-musical' v. 'musical'.

The data give confirms that the null hypothesis cannot be rejected concluding that there is no significant difference between 'non-musical' listeners and 'musical' listeners when extracting and understanding information in this instance of the Bubble Sort auralisation. The second part of this series of experiments tested the conveyance of erroneous algorithm auralisation.

The same thirty test subjects (Group 1) were asked to listen to the auralisation of a Bubble Sort algorithm that contained the following five errors:

1. False 'success' trumpet fanfare after the first checking pass.
2. Reversal of list during the second checking pass.
3. Third checking pass ignores changes made in the second sorting pass.
4. Iteration count indicates '1' before the 3rd sorting pass, it should indicate '3'.
5. Incorrect swap of elements '1' and '2' in the third sorting pass.

The test subjects were told that the algorithm contained five errors but the nature of the errors was unknown. Figure 6.9 indicates the accuracy of error identification for each of the listeners, the graph has been plotted in terms of the subjects' musical ability starting with least and progressing through to greatest.

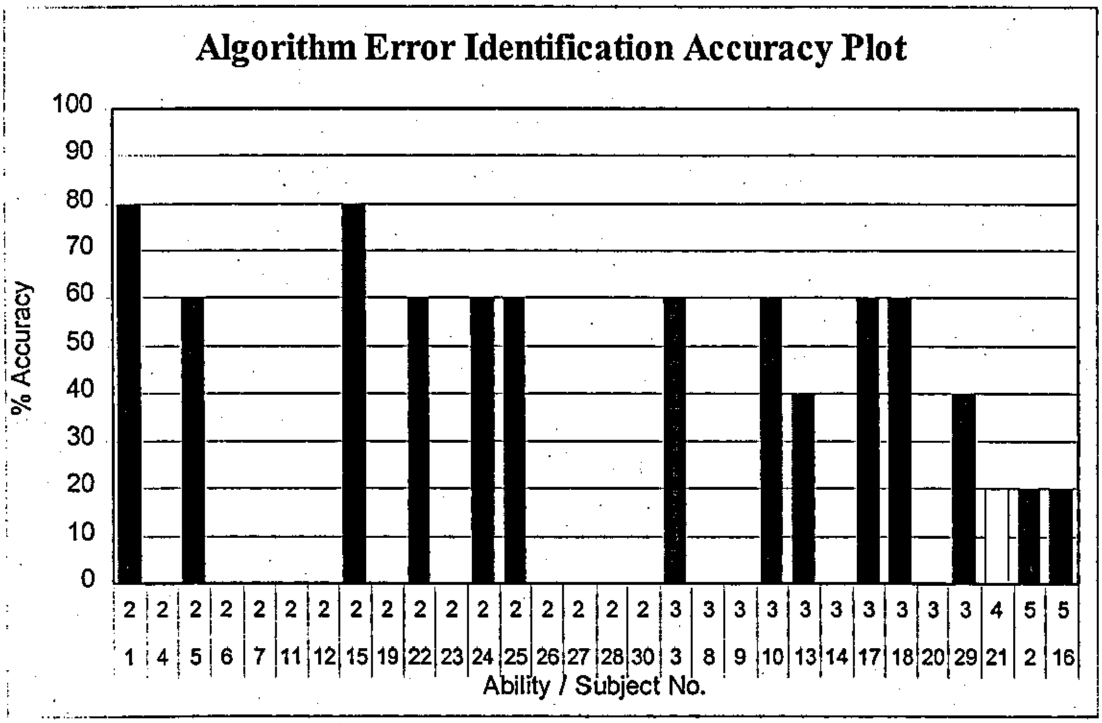


Figure 6.9 – Bubble Sort auralisation error accuracy.

It can be seen that algorithm error was not recognised well at all. This may be due to the listeners' limited exposure to algorithms (possibly because the test subjects had little prior knowledge about the nature and mechanics of the Bubble Sort algorithm). Such inexperience may lead to a limited understanding of the nature of this particular algorithm. Almost half of the test subjects failed to identify any of the errors. Of the remaining participants, ten identified approximately half of the errors. The data also suggests significant difference between the two groups of listeners.

Figure 6.9b shows the results of the Mann-Whitney non-parametric test applied to the error identification scores obtained by 'non-musical' listeners for the Bubble Sort auralisation compared to the error identification scores obtained by 'musical' listeners for the Bubble Sort auralisation. The hypotheses are:

H_0 : There is no difference in error identification accuracy between 'non-musical' listeners and 'musical' listeners when understanding the Bubble Sort algorithm auralisation.

H_1 : There is a significant difference in error identification accuracy between 'non-musical' listeners and 'musical' listeners when understanding the Bubble Sort algorithm auralisation.

	Error Identification Score
Mann-Whitney U	92.000
Wilcoxon W	245.000
	-.837
Asymp. Sig. (1-tailed)	.201

Figure 6.9b – Table of test statistics for erroneous algorithm information extraction, 'non-musical' v. 'musical'.

From the data given in the above figures, the null hypothesis cannot be rejected concluding that there is no significant difference between 'non-musical' listeners and 'musical' listeners when extracting and understanding error information in this instance of the Bubble Sort auralisation.

Once again it is important to look at the data from an information type viewpoint. The graph below in Figure 6.10 shows how successfully the group as a whole identified each of the algorithm errors.

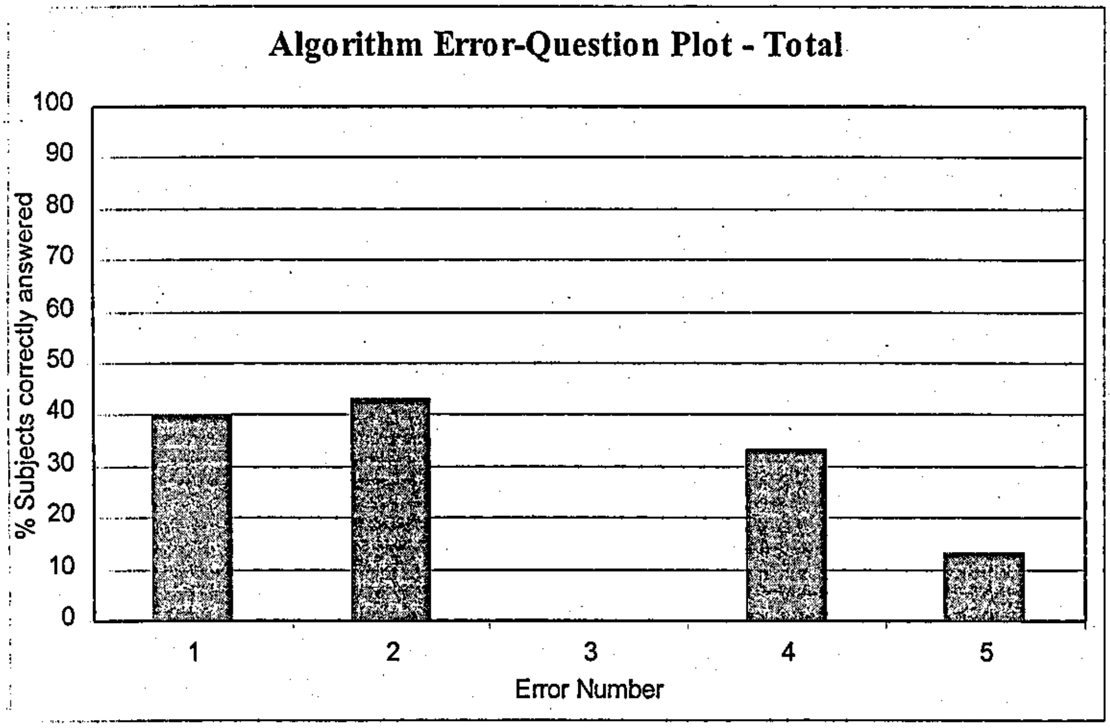


Figure 6.10 – Bubble Sort auralisation error accuracy by error type.

The more obvious errors were identified with greater success than the more subtle errors. The third error, which ignored all swaps in the third checking phase, went entirely unnoticed by all of the test subjects. The second least identified error was the incorrect swapping of already ordered elements in the third sorting pass. The remaining three errors were less subtle.

Although the general information identification is poor, between 33% and 43% of test subjects identified the more obvious errors. This data suggests that the task of identifying errors within unfamiliar algorithms is not a simple one and is certainly not as easy as identifying characteristics within an error free algorithm. To be able to identify such subtle information the listener would require some in-depth knowledge of the nature of the algorithm, so that greater familiarisation of this Bubble Sort algorithm could increase error identification accuracy in this context. It is also possible that a better auralisation might yield higher detection rates. The SIMBAA tool has been shown to successfully convey information about the Bubble Sort algorithm in the previous experiment. The possible available features of the algorithm were fully exploited. However, some training of the mappings used might increase the identification accuracy of the algorithm errors. It is important to state here that this thesis is not concerned with using auralisation to aid bug location in algorithms. It is more concerned with the ability of musical auralisation to convey information about the nature of the algorithms.

6.4. Multiple algorithm auralisation

Although in preliminary experiments the SIMBAA system was well received, it was decided to auralise an array of different algorithms much like Brown and Hershberger's selection [45]. This might highlight musical information structures that have a more successful information transfer rates than others in algorithm auralisation.

The algorithms previously examined in Chapter 4 have been auralised in a similar manner to that of the Bubble Sort algorithm. The following sections explain the chosen mappings and their implementation using the SIMBAA toolbox.

6.4.1. Selection Sort auralisation

The information defined in Chapter 4 to be auralised so that the Selection Sort algorithm can be understood musically is:

1. The current state of the list.
2. The iteration count.
3. Progression of the algorithm through the list of elements.
4. The swapping of elements.
5. Successful termination.

As the goal of the algorithm is to sort the elements into an ascending order then successful termination will be achieved when the current state of the list is heard as an ascending succession of notes in the diatonic scale. The mapping of music to the five points of interest has been chosen to be as understandable as possible. The mappings are:

1. The current state of the list 'Play Entire List' - this auralisation was achieved by mapping element values to pitch (a metaphor). The chosen instrument here was a flute, chosen from a distinct family as suggested by Alty and Rigas [158].

2. Iteration count – this auralisation was achieved by mapping the iteration counter to a wooden block. The sound of the wooden block is repeated for each iteration.

3. Progression of the algorithm through the list 'Play Current Element' - the element values were mapped to pitch using a piano.

4. The swapping of elements 'Play Swap Structure' - the structure uses a brass ensemble playing a major triad. The first note is an element to pitch mapping of the higher value in the current pair, the second note is an element to pitch mapping of the lower note in the current pair and finally the third note is a repetition of the first note.

5. Successful termination 'Play Successful Termination' - this auralisation also uses a brass ensemble, but this time it was used to produce a simple yet suggestive 'Ta - Da' sequence.

Four timbres were necessary in order to achieve auralisation of the Selection Sort algorithm, an acoustic grand piano, a flute, a brass ensemble and a wooden block. In order to help disambiguate the information, panning was employed. The flute was centrally located, the piano on the left, the brass ensemble on the right, and the wooden block also on the right within the stereophonic field.

6.4.2. Exchange Sort auralisation

The information which needs to be auralised is as follows:

1. The current state of the list.
2. The iteration count.
3. Progression of the algorithm through the list of elements.
4. The swapping of elements.
5. Successful termination.

As the goal of the algorithm is to sort the elements into an ascending order then successful termination will be achieved when then current state of the list is heard as an ascending succession of notes in the diatonic scale. The mapping of music to the five points of interest was identical to the previous algorithm.

6.4.3. Quick Sort auralisation

The information from the Quick Sort algorithm that needs to be auralised is as follows:

1. The current state of the list.
2. The iteration count.
3. The value of the current pivot.
4. The value of the current element.
5. The placing of the current element into the left or right sub-lists.
6. Successful termination.

As the goal of the algorithm is to sort the elements into an ascending order then successful termination will be achieved when then current state of the list is heard as an ascending succession of notes in the diatonic scale. The mapping of music to the six points of interest has been chosen to be as understandable as possible. The mappings are:

1. The current state of the list 'Play Entire List' - this auralisation was achieved by mapping element values to pitch (a metaphor). Again the chosen instrument here was a flute.

2. Iteration count – this auralisation was achieved by mapping the counter that is used to control number of iterations to a wooden block. The sound of the wooden block is repeated for each iteration.

3. Value of the current pivot – this is heard in the centre of the stereophonic field. The chosen timbre is the trumpet with no addition of chorus or echo. The duration of this note is twice as long as the other notes to highlight it as a decision point.

4. Playing the current element that is to be sorted based upon the current chosen pivot - the chosen mapping here was a simple acoustic grand piano placed in the centre of the stereophonic field. Again the element values were mapped to pitch.

5. The placement of elements – this is heard to the left of the stereophonic field if the current element is smaller than or equal to the pivot and heard to the right if it is greater. The chosen timbre is again the acoustic grand piano with no additional effects.

6. Successful termination 'Play Successful Termination' - this auralisation was achieved by again using the brass ensemble, but this time it was used to produce a simple yet suggestive 'Ta - Da' sequence.

The four instruments used were an acoustic grand piano, a flute, a brass ensemble and a wooden block. All mappings pertaining to the values within the list during the sorting passes employed the same timbre of the acoustic grand piano. The decision-making pivot utilised the trumpet much like the previously described algorithms used the trumpet to denote swapping.

Auditory space was used to assist disambiguation. The flute was central, the piano on the left, the brass ensemble on the right and the wooden block also on the right within the stereophonic field. The placement of elements into sub-lists also exploited the used of auditory space by placing all elements that fall into the left sub-list to the left of the stereophonic field and vice versa for the right sub-list.

6.4.4. Bucket Sort (Inside-Out) auralisation

The chosen information from the Inside-Out Sort algorithm that is to be auralised is that of a single pass in the Quick Sort auralisation followed by the auralisations of the Bubble Sort and Selection Sort algorithms on the left and right hand sub-lists respectively. As the goal of the algorithm is to sort the elements into an ascending order then successful termination will be achieved when then current state of the list is heard as an ascending succession of notes in the diatonic scale.

The same instruments and placements within the stereophonic field have been employed as those previously described for the Quick, Bubble and Selection Sort algorithms.

6.4.5. Bucket Sort (Outside-In) auralisation

The chosen information from the Outside-In Sort algorithm that is to be auralised is that of a single pass in the Quick Sort auralisation followed by the auralisations of the Selection Sort and Bubble Sort algorithms on the left and right hand sub-lists respectively. As the goal of the algorithm is to sort the elements into an ascending order then successful termination will be achieved when then current state of the list is heard as an ascending succession of notes in the diatonic scale. The same instruments and placements within the stereophonic field have been employed as those previously described for the Quick, Selection and Bubble Sort algorithms.

6.5. Multiple algorithm auralisation information extraction

6.5.1. Experiment construction

In this series of experiments the Bubble Sort, Exchange Sort, Selection Sort, Quick Sort, Bucket In-Out Sort and Bucket Out-In Sort algorithms were auralised and played to thirty test subjects. The SIMBAA system created the algorithm auralisations. The points of interest that were auralised in for the various algorithms can be summarised as follows:

1. The current state of the algorithm.
2. The iteration count.
3. Progression of the algorithm through the list of elements.
4. The swapping or placement of elements around a pivot.
5. Successful termination.

The subjects were told about the nature of each of the algorithms. Information and played example auralisations pertaining to each of the algorithms were presented to the test subjects. The questionnaire presented to all test subjects pertaining to this series of experiments is given in full in Appendix L sections 9.1 to 9.6 and can be summarised as:

9.1 – Bubble Sort example and auralisation test.
9.2 – Selection Sort example and auralisation test.
9.3 – Quick Sort example and auralisation test.
9.4 – Bucket In-Out Sort example and auralisation test.
9.5 – Bucket Out-In Sort example and auralisation test.
9.6 – Exchange Sort example and auralisation test.

The information requested from the each of the algorithm auralisations differs between algorithms. The question set shown below apply to the Bubble Sort, Exchange Sort and Selection Sort algorithm auralisations. The questions shown in bold print are those that are of a quantitative nature. The remaining questions shown in italic print are those of a qualitative nature.

1 -How many numbers (elements) are there in the list?
2 -How many swaps are there in the first pass?
3 -How many swaps are there in the second pass?
4 -How many swaps are there in the third pass?
5 -*How do you know when elements are out of order?*
6 -*How do you know when the recipe swaps elements?*
7 -*How do you know when the list is sorted?*
8 -How many times does the recipe pass through the list?
9 -*What order is the list sorted into? (Perceived general shape of the list).*
10 - *How does the shape of the list progress?*

The questions that were applied to the Quick Sort algorithm auralisation are given as:

1 -How many numbers (elements) are there in the list?

2 –What value is the pivot in the first pass?

3 –What are the sub-list sizes on the first pass?

4 - How many pivots are there on the final pass?

5 –*What identifies element placement into sub-lists?*

6 – *How is the pivot musically represented?*

7 -*How do you know when the list is sorted?*

8 -How many times does the recipe pass through the list?

9 -*What order is the list sorted into?*

10 - *How does the shape of the list progress?*

The questions applied to the Bucket In-Out Sort and Bucket Out-In Sort are given as:

1 -How many numbers (elements) are there in the list?

2 –What value is the pivot in the first pass?

3 –What are the sub-list sizes on the first pass?

4 - How many swaps are there in the 2nd pass?

5 –*After 1st pass, what denotes swapping?*

6 – *How is the pivot musically represented?*

7 -*How do you know when the list is sorted?*

8 -How many times does the recipe pass through the list?

9 -*What order is the list sorted into?*

10 - *How does the shape of the list progress?*

6.5.2. Results and analysis

The diagram below in Figure 6.30 shows the musical ability distribution of the group of thirty test subjects. The group consists of 14 'non-musical' listeners and 16 'musical' listeners. This group of test subjects are referred to as Group 4.

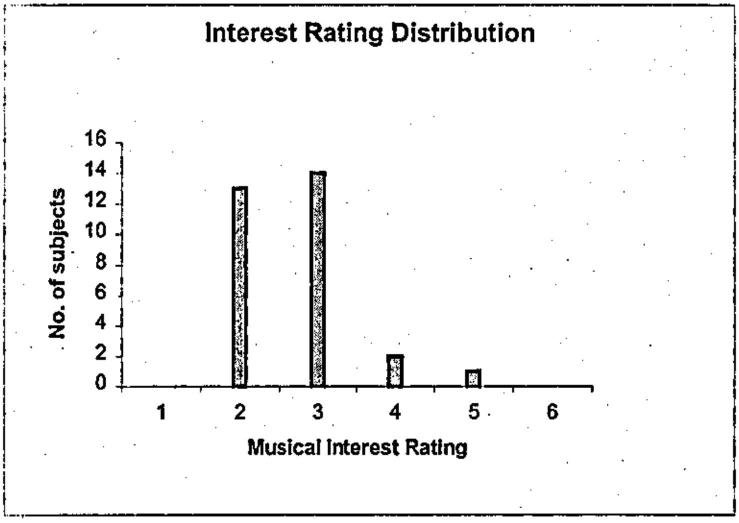


Figure 6.30 – Musical ability rating – Multiple algorithm auralisation experiments.

With this series of experiments a further preliminary test was carried out in order to understand the users' ability to draw the shapes of simple tunes. Given that some musically trained test subjects might fully understand the shape of the tonal sequences it may also be possible that they do not have the ability to draw. The test listeners that were considered incapable of drawing regardless of their musical ability were omitted from the experimental data obtained through this series of experiments. All test subjects were chosen on the basis that they had very little or no prior knowledge of sorting algorithms.

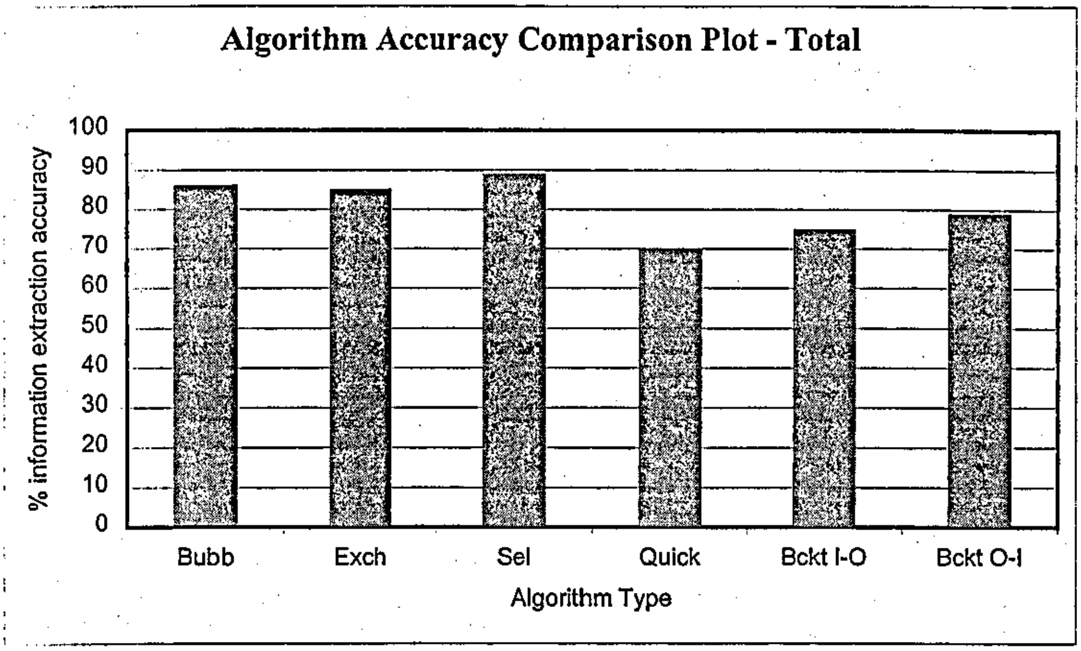


Figure 6.30a – Algorithm information extraction accuracy for each algorithm.

The graph given in Figure 6.30a shows how each of the algorithm auralisations compare. These data represent the average information extraction for each of the algorithms for the entire group of test listeners. These data suggest that the algorithms with the previously described (Chapter 5) anchor points near to the boundaries of the context scale tend to be more easily understood than the algorithms that employ the Quick Sort algorithm where the anchor points are either moving between passes or becoming larger in number. It is necessary to split this data into sub-groups defined by musical ability to investigate if musical training has any effect on understanding the information. It is also necessary to investigate whether certain information types, quantitative or qualitative, translate better during algorithm auralisation.

The graph in Figure 6.31 shows how Group 4 performed when answering questions pertaining to information extraction from the Bubble Sort auralisation. The data is displayed along the x-axis in order of musical ability.

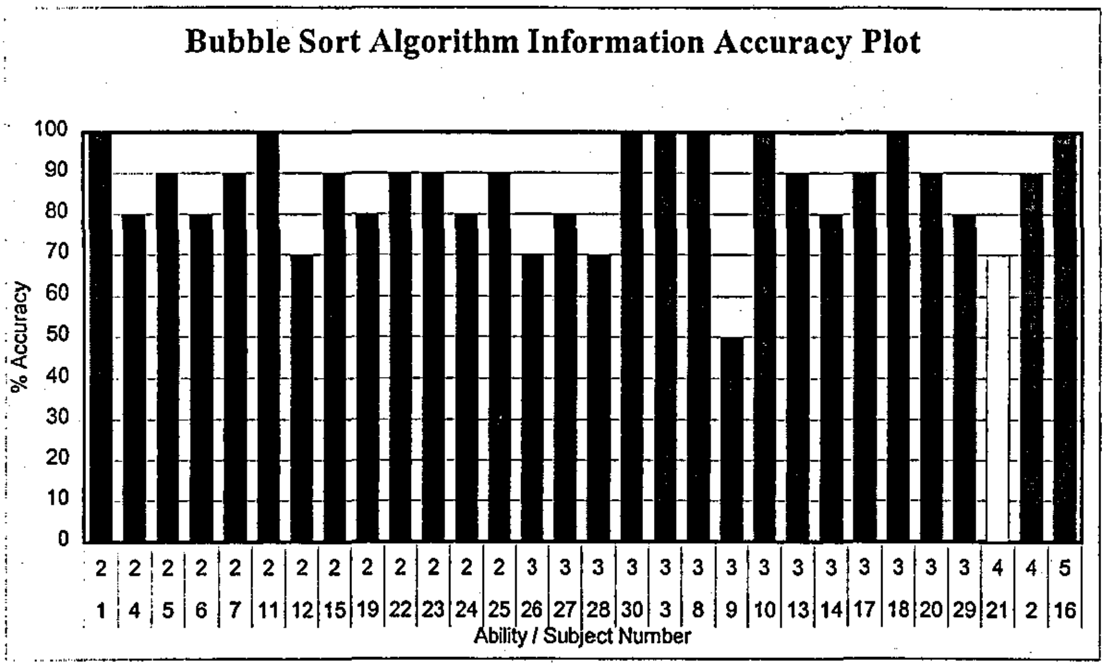


Figure 6.31 – Bubble Sort information extraction accuracy.

The data for the remaining algorithm auralistaions exhibit similar results. The graphs for these results are given in Figures M.1 to M.5 in Appendix M.

The data suggests that there is little difference between 'musical' and 'non-musical' subjects and that the overall performance of the test group looks encouraging. The average scores and the probabilities of the occurrence of these scores at random for each algorithm auralisation calculated using the same Binomial expansion previously shown in this chapter are:

	Bubble	Exchange	Selection	Quick	BIO	BOI
Score% or r	86.333	84.667	89	70	75	78.667
p(r)	$6.283e^{-46}$	$7.393e^{-45}$	$1.217e^{-47}$	$3.467e^{-34}$	$1.257e^{-32}$	$5.544e^{-35}$

Figure 6.32 – Table of statistics for each algorithm auralisation.

These data strongly suggest that the group of listeners are capable of understanding each of the six algorithms when represented musically.

The data in the table given in Figure 6.36b shows the results of the Mann-Whitney non-parametric test applied to the information extraction scores obtained from each of the six algorithm auralisations for the 'musical' listeners compared to 'non-musical' listeners. The hypotheses are:

H$_0$: There is no significant difference between 'musical' and 'non-musical' listeners when extracting information from the specified algorithm auralisation.

H$_1$: There is a significant difference between 'musical' and 'non-musical' listeners when extracting information from the specified algorithm auralisation.

	Bubbl	Exchang	Selection	Quick	Bucket In-Out	Bucket Out-In
Mann-Whitney U	106.500	101.500	91.000	109.000	93.50	103.000
Wilcoxon W	197.500	192.500	244.000	262.000	184.500	194.000
Z	-.17	-.40	-.866	-.065	-.73	-.325
Asymp. Sig. (1-tailed)	.43	.34	.192	.472	.23	.371

Figure 6.36b – Table of test statistics, algorithm information extraction, 'musical' v. 'non-musical'.

From the data given in the above figures, the null hypothesis cannot be rejected for each algorithm auralisation concluding that there is no significant difference between 'musical' and 'non-musical' listeners when understanding and extracting information from the each of the algorithm auralisations.

Figures M.6, M.7, M.8, M.9, M.10 and M.11 in Appendix M show the performance of the group for each of the questions on the Bubble Sort, Exchange Sort, Selection Sort, Quick Sort, Bucket In-Out Sort and Bucket Out-In Sort auralisations respectively.

Quantitative question are shown as solid bars and qualitative questions are shown as clear bars.

The data suggests that there is some difference between quantitative and qualitative information perception. It also suggests that overall performance of the test group is generally high for each of the questions. Figure 6.42b shows the results of the Wilcoxon Signed Ranks non-parametric test applied to the information extraction scores obtained from each of the six algorithm auralisations for qualitative questions compared to quantitative questions. The hypotheses are:

H_0 : There is no significant difference between quantitative and qualitative information perception and understanding for the specified algorithm auralisation.

H_1 : There is a significant difference between quantitative and qualitative information perception and understanding for the specified algorithm auralisation.

	BQN2 - BQL2	EQN2 - EQL2	SQN2 - SQL2	QQN2 QQL2	BIOQN2 - BIOQL2	BOIQN2 - BOIQL2
Z	-1.67	-2.20	-1.851	-1.869	-1.94	-1.854
Asymp. Sig. (1-tailed)	.047	.01	.032	.031	.02	.032

Figure 6.42b – Table of test statistics for algorithms' information extraction, qualitative v. quantitative.

This analysis has been performed as a matter of completeness because cases might exist where only qualitative or only quantitative information might need to be presented. However, in the context of this thesis, both information types are used for algorithm understanding. From the data given in the above figures, the null hypothesis can be rejected at the 5% level of confidence for each algorithm auralisations concluding that there is a significant difference between the perception and understanding of qualitative and quantitative information types for the each of the algorithm auralisations.

Furthermore the data shows that quantitative information translates better than qualitative information. This suggests that the use of sonification of the data in the list (quantitative information) translated with greater success than the more abstract metaphorical mappings (qualitative information). This might suggest that the use of different metaphors could lead to increased understanding of qualitative information types.

Given that no significant difference between 'musical' test subjects and 'non-musical' test subjects when understanding musically auralised algorithm execution and state has been proven, it is important to examine if this also holds true for each of the information types. The graphs given in Figures M.12, M.13, M.14, M.15, M.16 and M.17 in Appendix M show how the two sub-groups perform on each question for the Bubble Sort, Exchange Sort, Selection Sort, Quick Sort, Bucket In-Out Sort and Bucket Out-In Sort algorithms respectively.

The data given in the above figures suggests that there is little difference between 'musical' listeners and 'non-musical' listeners when understanding either quantitative information types or qualitative information types.

Figure 6.48b shows the results of the Mann-Whitney non-parametric test applied to the information extraction scores obtained from each of the six algorithm auralisations for 'musical' listeners compared to 'non-musical' listeners for **qualitative** question types. The hypotheses are:

H_0 : There is no significant difference between 'musical' listeners and 'non-musical' listeners when understanding **qualitative** information for the specified algorithm auralisation.

H_1 : There is a significant difference between 'musical' listeners and 'non-musical' listeners when understanding **qualitative** information for the specified algorithm auralisation.

	BUBB	EXCH	SEL	QUICK	BCKT I-O	BCKT O-I
Mann-Whitney U	102.500	107.000	86.500	93.500	110.000	105.500
Wilcoxon W	255.500	260.000	239.500	246.500	263.000	196.500
Z	-.357	-.15	-1.08	-.746	-.023	-.225
Asymp. Sig. (1-tailed)	.361	.43	.138	.228	.491	.411

Figure 6.48b – Table of test statistics - algorithms' qualitative information extraction, mus v. non-mus

From the data given in the above figures, the null hypothesis cannot be rejected for each of the algorithm auralisations concluding that there is no significant difference between 'musical' listeners and 'non-musical' listeners for the perception and understanding of **qualitative** information types for the each of the algorithm auralisations.

Figure 6.49b shows the results of the Mann-Whitney non-parametric test applied to the information extraction scores obtained from each of the six algorithm auralisations for 'musical' listeners compared to 'non-musical' listeners for **quantitative** question types. The hypotheses are:

H_0 : There is no significant difference between 'musical' listeners and 'non-musical' listeners when understanding **quantitative** information for the specified algorithm auralisation.

H_1 : There is a significant difference between 'musical' listeners and 'non-musical' listeners when understanding **quantitative** information for the specified algorithm auralisation.

	BUBB	EXCH	SEL	QUICK	BCKT I-O	BCKT O-I
Mann-Whitney U	101.500	96.000	103.000	96.00	79.000	104.500
Wilcoxon W	192.500	187.000	256.000	187.00	170.000	195.500
Z	-.452	-.68	-.405	-.64	-1.416	-.268
Asymp. Sig. (1-tailed)	.326	.24	.343	.25	.078	.395

Figure 6.49b – Table of test statistics - algorithms' quantitative information extraction, mus v. non-mus

From the data given in the above figures, the null hypothesis cannot be rejected for each of the algorithm auralisations concluding that there is no significant difference between 'musical' listeners and 'non-musical' listeners for the perception and understanding of **quantitative** information types for the each of the algorithm auralisations.

Given that no significant difference has been shown between 'musical' listeners and 'non-musical' listeners when understanding either quantitative information types or qualitative information types, it is also important to analyse the variance between the information types.

Figure 6.50b shows the results of the Wilcoxon signed rank non-parametric test applied to the information extraction scores obtained from each of the six algorithm auralisations for '**musical**' listeners for quantitative question types compared to qualitative question types. The hypotheses are:

H_0 : There is no significant difference between qualitative and quantitative question types for '**musical**' listeners when understanding information for the specified algorithm auralisation.

H_1 : There is a significant difference between qualitative and quantitative question types for '**musical**' listeners when understanding information for the specified algorithm auralisation.

	Bubble	Exchang	Selection	Quic	Bucket in-Out	Bucket Out-In
Z	-.690	-1.31	-.749	-.52	-.690	-1.115
Asymp. Sig. (1-tailed)	.245	.095	.227	.30	.245	.133

Figure 6.50b – Table of test statistics, algorithms' 'musical' information extraction, qualitative v. quantitative

From the data given in the above figures the null hypothesis cannot be rejected for each algorithm auralisations concluding that there is no significant difference between qualitative and quantitative question types for **'musical'** test listeners when perceiving and understanding information for the each of the algorithm auralisations. Again, both information types are required for understanding sorting algorithms within the context of this thesis. The testing of information types here is not a primary concern of the thesis but it does provide some useful information for auralistaion of qualitative only or quantitative only presentations.

Figure 6.51b shows the results of the Wilcoxon signed rank non-parametric test applied to the information extraction scores obtained from each of the six algorithm auralisations for **'non-musical'** listeners for quantitative question types compared to qualitative question types. The hypotheses are:

H_0 : There is no significant difference between qualitative and quantitative question types for **'non-musical'** listeners when understanding information for the specified algorithm auralisation.

H_1 : There is a significant difference between qualitative and quantitative question types for **'non-musical'** listeners when understanding information for the specified algorithm auralisation.

	Bubble	Exchang	Selection	Quick	Bucket In-Out	Bucket Out-In
Z	-1.64	-1.80	-1.814	-2.077	-1.99	-1.539
Asymp. Sig. (1-tailed)	.050	.03	.035	.019	.02	.062

Figure 6.51b – Table of test statistics, algorithms' 'non-musical' information extraction, qualitative v. quantitative

From the data given in the above figures, the null hypothesis can be rejected at the 5% level of confidence for almost all algorithm auralisations concluding that there is significant difference between quantitative and qualitative question types for **'non-musical'** listeners when perceiving and understanding information for the each of the algorithm auralisations. This data suggests that although no significant difference has been shown between 'musical' and 'non-musical' listeners when perceiving and understanding qualitative and quantitative information types, the spread in accuracy between the information types is greater for 'non-musical' listeners than for 'musical' listeners. It can be further concluded that 'musical' listeners are more reliable at perceiving and understanding both information types given that the spread in accuracy is smaller. This increased reliability is present but not statistically significant, though increasing the sample size in this case may have shown some significance.

6.6. Conclusions

Initial experimentation with the Bubble Sort algorithm showed that information pertaining to state and execution can be successfully represented musically with test subjects scoring generally high and identifying a large majority of the requested information.

Further experimentation using the Bubble Sort algorithm with the addition of the fourth timbre to denote iteration count, with increased test group sizes to allow for reliable statistical analysis and the addition of deliberate algorithm error showed, that once again the information exchange was significantly high with the majority of information being identified. Statistical analysis of the results showed that there is no significant difference between 'musical' and 'non-musical' test subjects when identifying algorithm state and execution information and erroneous algorithm state and execution information. The analysis also showed that quantitative information translates significantly better than qualitative information in the context of the experimental auralisations. Although the identification of erroneous algorithm information was low, the data suggested that obvious errors were more easily identified than subtle errors. The identification of bugs in sorting algorithms through musical auralisation was not of concern in this thesis and SIMBAA was not designed to be a tool for aiding such bug location. However, it does provide some interesting information about the difficulties associated with untrained listeners attempting to identify the more subtle and intricate information present in sorting algorithm. Further experimentation here might yield better error identification rates if alternative musical metaphors were used. This requires significant investigation and is beyond the scope of this thesis.

The results for the experimentation using the six different algorithms showed that once again the algorithms with fixed and constant anchor points are more readily understood (Bubble Sort, Exchange Sort and Selection Sort algorithms). The data also showed that overall there is no significant difference between 'musical' listeners and 'non-musical' listeners when perceiving and understanding musically represented information pertaining to state and execution for each of the algorithms. It was also shown that quantitative information types are significantly more easily understood and identified than qualitative information types. Within each of these information types it was further shown that there is no significant difference between 'musical' and 'non-musical' listeners. The results again showed that for the 'non-musical' sub-group alone quantitative information types were more easily understood than qualitative information types. In contrast, no significant difference was observed between the two information types for the 'musical' sub-group alone. This suggested that the spread between information types for each sub-group was different but not different enough to be significant when comparing the two sub-groups.

Once again the addition of extra cues might further disambiguate the information presented in each of the algorithm auralisations. Given that the addition of spatial location cues could aid perception of pitch and shape, it could also be used to aid understanding of algorithm auralisations. In order to incorporate the extra spatial cues, three-dimensional sound source placement could be employed. The following chapter documents the design and implementation of the SIMBAA system incorporating spatial enhancement to facilitate the subsequently documented algorithm auralisation experimentation using three-dimensional sound.

Chapter 7

SIMBAA 3D – A spatially enhanced musical auralisation tool

7.1. Spatially enhancing SIMBAA

In the SIMBAA experiments in Chapter 6, disambiguation of the musically presented information was achieved by the use of the following:

- Rhythm of ascension.
- Contrasting timbre using a piano, a flute and a brass ensemble.
- Harmony using a major triad.
- The use of auditory space by employing instrument placement via stereophony.

The auralisations implemented so far have exploited the features of the sorting algorithms identified in Chapter 4. It might be argued that better auralisations could have produced improved perception. This thesis is not concerned with developing optimum auralisations but it is concerned with investigating the effect of the addition of spatialisation on the existing auralisations. However, the use of auditory space in this instance was under exploited. Although placement of the different timbres in the left, centre and right locations within the stereophonic field aided disambiguation between events and actions it was limited to the line between the listener's ears. Better disambiguation might be achieved by extending the two-dimensional stereophonic field into a three-dimensional auditory environment. In particular the spatialisation of the data might provide positional cues about the values of the data and their positions within the list. This might further aid understanding of the execution and sorting natures of the algorithms.

It was therefore decided to develop SIMBAA into a 3-D auditory environment (called SIMBAA 3D). The enhanced system will have all the features of the existing SIMBAA system together with a greatly enhanced 3-D environment. The method chosen has already been extensively discussed and justified in Chapter 3, namely, the use of stored binaural recordings created using microphones placed inside the ears of subjects. The main new aspect of SIMBAA 3D is the creation of what we call a "SoundWall" – a two-dimensional projected wall onto which sounds can traverse aurally in two directions. In addition, control sounds can be played behind the listener (left or right).

The diagram below in Figure 7.1 shows the conceptual 3D spatial auditory environment with an example eight element list spatially distributed on the wall.

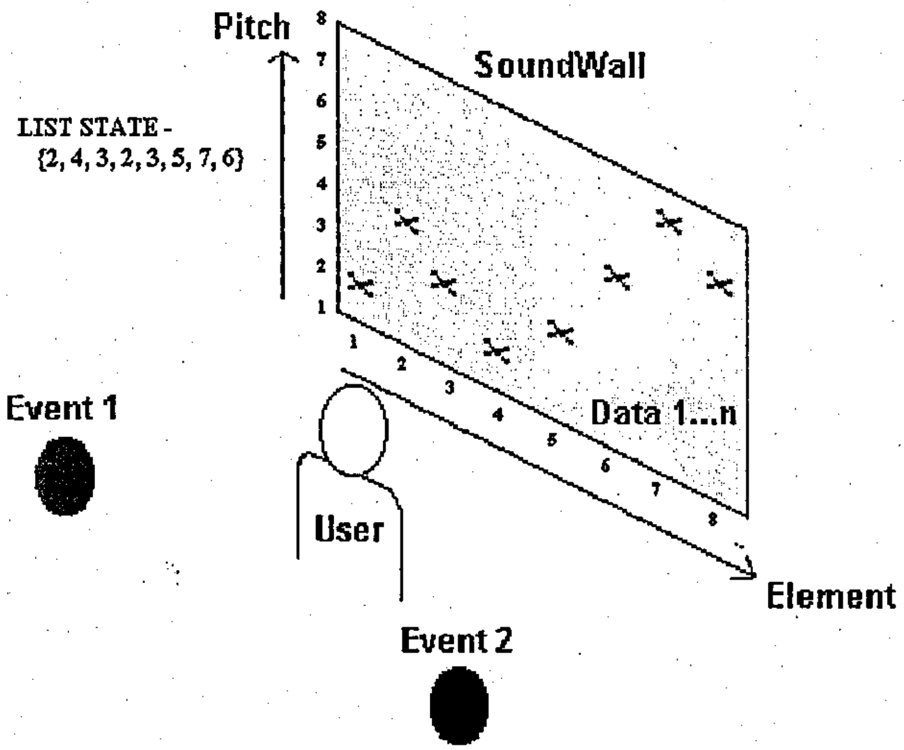


Figure 7.1 – SIMBAA 3D conceptual auditory scene.

Control Events such as successful termination and iteration are located behind the listener's ears and identified as 'Eventl' and 'Event2'. The remaining mappings pertaining to the data in the list are projected onto the 3D 'SoundWall'. The different processes that can be applied to the data in the list (checking the data, sorting the data and moving the data) could be distinguished through the use of different timbres.

7.2. Algorithm experimentation and the SIMBAA 3D environment

All of the algorithm auralisation experiments performed in Chapter 6 utilise lists that contain numbers between 1 and 8, this is to allow for a single diatonic octave to be used. The reason for limiting the representations to one octave is based upon Sumikawa et al's [175, 176] findings that perception is increased when mappings are confined to one octave as opposed to spanning octaves. Given that the varying pitch (8 notes in this case) is to be mapped to the elevation cue of the 'SoundWall', it can be defined that the wall must facilitate eight elevation positions.

To limit the complexity of the algorithm auralisations, the maximum number of elements within the lists used in the experiments in Chapter 6 was never less than six and never greater than ten. It is important to clarify for design purposes that this limitation has not been based on the constraints of the human memory reported by Miller [142] which stated that the human short term memory can only hold about 7±2 chunks of information at any one time. The experiments involving algorithm auralisation are not concerned with listeners being able to remember exact elements but rather the general shapes of the lists of numbers, in particular which portions of the list are random and which are sorted into order. Given this, Miller's observations on the constraints of human short term memory are not of any significance as the features of the shapes of the lists are never likely to exceed three (e.g. random-smooth-random) due to the natures of chosen algorithms. Having determined that the maximum list size is never greater than ten for this series of experiments, it can be defined that the number of azimuth locations required on the 'SoundWall' is to be ten.

In order to create an auditory scene capable of accommodating the necessary information represented by algorithm auralisations as employed in Chapter 6 of this thesis it is necessary to define the information that is to be represented musically for experimentation. This information has been identified in Chapter 4 and 6 but needs to be reiterated here, the information common to the selected algorithms for auralisation are:

- State of the list – tones mapped to numbers within the list. The relative pitch of the tones alone gives the impression of the shape of the list, this shape could be further clarified and disambiguated through the placement of the elements in 3D auditory space. The shape of the list could be projected onto the 'SoundWall'. As the sequence progressed, the instrument would move along the wall from left to right. Similarly, as the pitch of the tones increased, the elevation position of the projected sound source would also increase. This would provide an additional cue, one of a positional nature, as to the data values and their positions within the list.

- Sorting the list – tones mapped to numbers within the list. As with the previous mapping this phase of the algorithm's execution requires the musical representation of current elements within the list that are presently being sorted. Again this could be projected onto the 'SoundWall'. The difference between this mapping and the previous mapping is the discrimination of timbre as employed in the algorithm auralisations described in Chapter 6. The mapping for representing the state of the list uses a flute, this mapping employs an acoustic grand piano. This would also provide an additional cue, one of a positional nature, as to the data values and their positions within the list.

- Swapping/placement of elements – tones mapped to numbers within the list. As with the previous mappings, this requires the numbers within the list to be represented musically. Again this can be projected onto the 'SoundWall' with the employment of different timbre, in this case the choice of timbre is a trumpet as described for experimentation in Chapter 6. This would provide an additional cue, one of a positional nature, as to the swapping of the data values and their swapped positions within the list.

- Iteration count – wooden block indicating the iteration count. This mapping requires the pass count to be represented musically. This does not require to be projected onto the 'SoundWall', rather, to discern this mapping from the previous pure data element mappings it could be placed 180 degrees (for maximum segregation) away from the 'SoundWall', locating it behind the listener. As with the earlier experiments with algorithm auralisation described in this thesis the choice of timbre here is the wooden block.

- Successful termination – 'Ta-Da' success on algorithm sorting completion. The choice of timbre here for previous auralisations within this thesis has been a brass ensemble. Again this timbre shall be employed and to discern this mapping from the pure data element mappings it could again be placed behind the listener.

To summarise, for application to algorithm auralisation within the scope of this thesis the 'SoundWall' needs to be constructed with the following parameters:

- Three instruments – flute, acoustic grand piano and trumpet.
- Eight elevation positions – mapping to pitch within one diatonic octave starting at 'Middle C'.
- Ten azimuth positions – mapping to the position of the current elements in the list, being between 6 and 10.

Therefore it is necessary to binaurally record 240 (3 instruments x 8 pitch locations x 10 positional locations) real audio samples with the addition of the two control events (iteration count and successful termination).

7.3. Creating the auditory scene

In order to binaurally record the necessary real audio sounds a pair of binaural microphones is required. It has been decided that for cost effectiveness and simplicity a real human being is to be used instead of a manikin for the object listener. One of the concerns with using a real human being is the unpredictable motion of the listener. When recording a series of audio samples it is important to retain the same relative position of the object listener's ears, head and torso. To accomplish this, a small laser pointer has been employed. This laser was mounted on a pair of glasses that are rigidly attached to the listener's head, this is shown in the picture given below in Figure 7.2. It should be noted here that the head mounted equipment might have an effect on the binaural recordings, in that they might introduce extra reflections. To minimise this, all leads and connectors were tied back behind the listener's head. The reflections from the laser mount, casing and glasses could not be effectively reduced without extensive design. It is important to reiterate here that this thesis is concerned with using a low cost spatialisation technique.

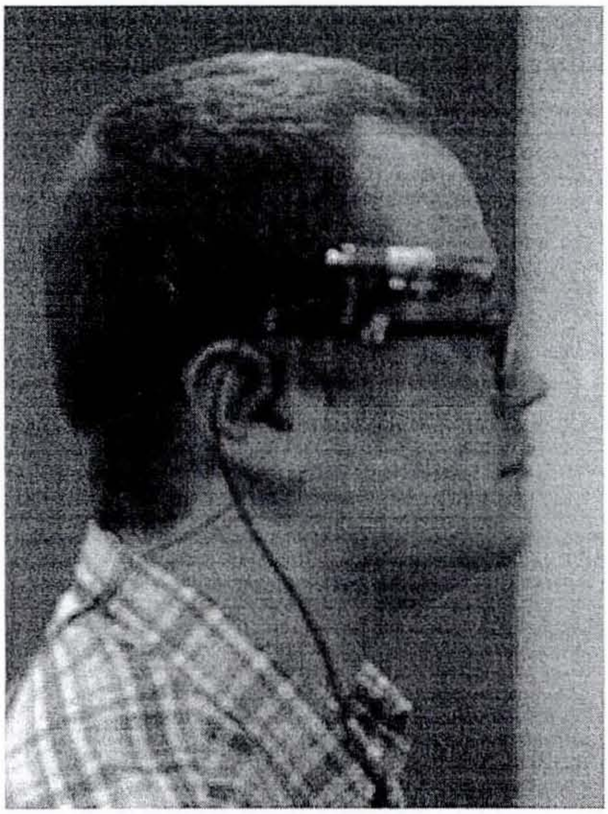


Figure 7.2 – Laser head alignment.

The centre point of the 'SoundWall' was located and a marker was affixed to retain the focal point. Upon recording the samples, the object listener aligned the laser pointer to this focal point (shown below) thereby maintaining approximately the same relative pinna, head and torso positions.

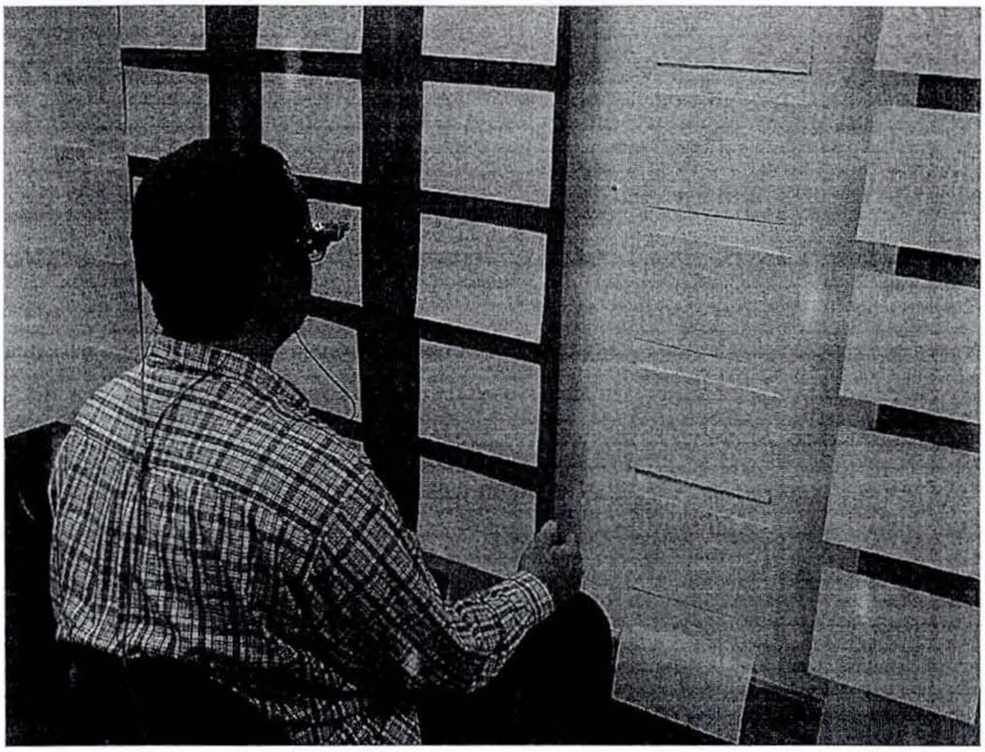


Figure 7.3 – Laser head alignment in semi-anechoic recording studio.

In order to reduce the localisation confusion introduced by excessive environmental reverberation the sound samples were recorded in a professional recording studio with semi-anechoic walls. The sound absorbent surfaces of the walls are shown in the picture above in Figure 7.3 as the large blue cloth covered panels. The dimensions of the wall were made as large as the recording studio would facilitate. This was done in order to maximise the possible space between the three-dimensional coordinates of each sound source to aid instrument location separation. The dimensions of the wall were 411.5 cm (162 inches) wide by 186.7 cm (73.5 inches) high. The sound sources were therefore placed with 45.7 cm (18 inches) of separation in the horizontal plane and 26.77 cm (10.5 inches) of separation in the vertical plane. The object listener was placed in the centre of the 'SoundWall' 205.7 cm (81 inches) from the left most boundary and 93.4 cm (36.75) inches from the lower most boundary.

One concern that became apparent during the initial recording phase was the position of the listener relative to the 'SoundWall' and the event objects. Two 'trial' auditory scenes were therefore created with the listener being positioned six feet from the 'SoundWall' and three feet from the 'SoundWall'. The four corners and an ascending pattern were played to several test listeners each of whom was asked to choose which representation gave the most realistic impression of three dimensional movement of sound on a projected virtual 'SoundWall'. It was found that all test listeners much preferred the 'SoundWall' that was recorded at a distance of three feet. It was therefore decided that this parameter should be used when recording the final 'SoundWall'. As mentioned in Chapter 3, the closer a source gets closer to a human head, the greater the inter-aural intensity difference. This increase in difference is particularly noticeable for ranges under one meter. This may be a factor in the listeners' preference for the 'SoundWall' at 3ft.

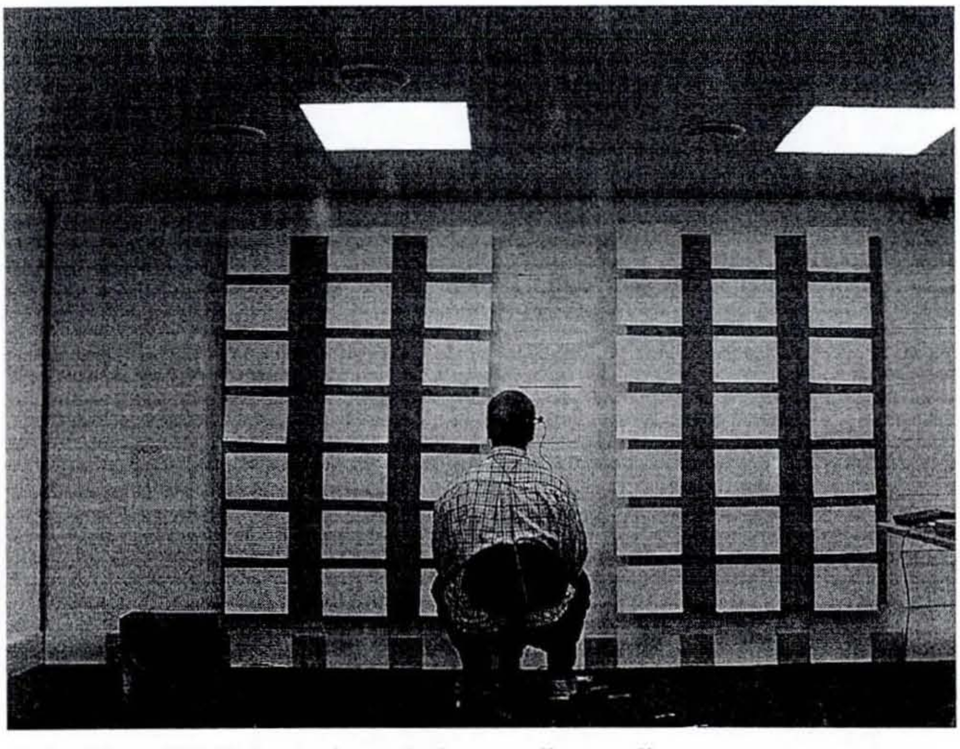


Figure 7.4 – 'SoundWall' in semi-anechoic recording studio.

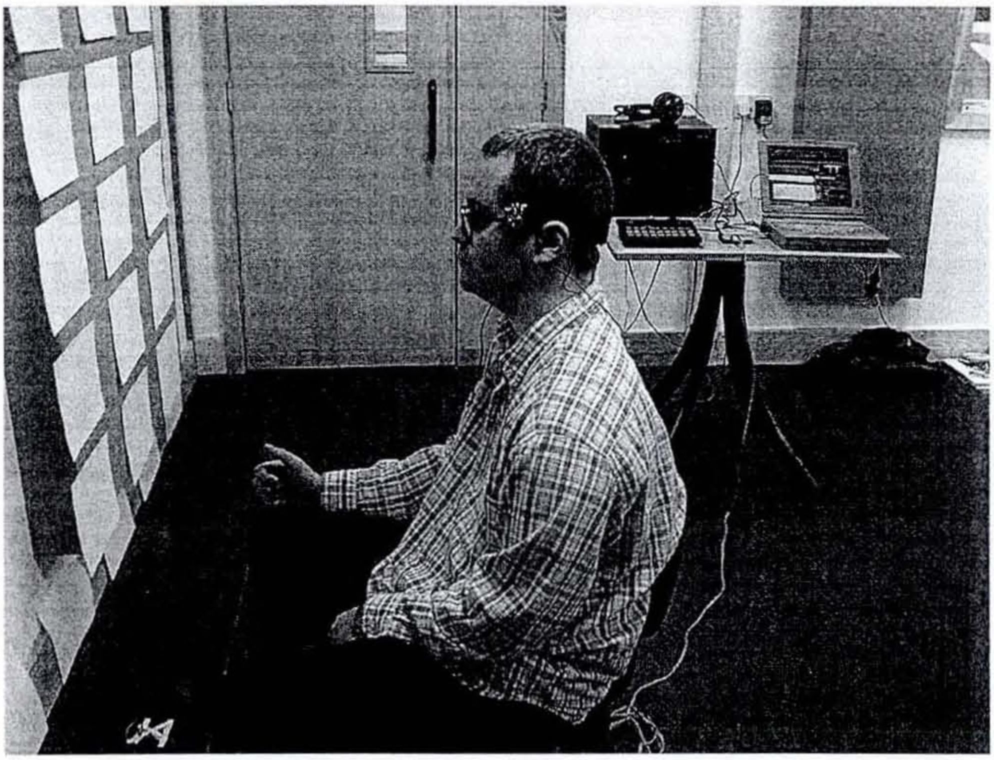


Figure 7.5 – 'SoundWall' and object listener – left perspective.

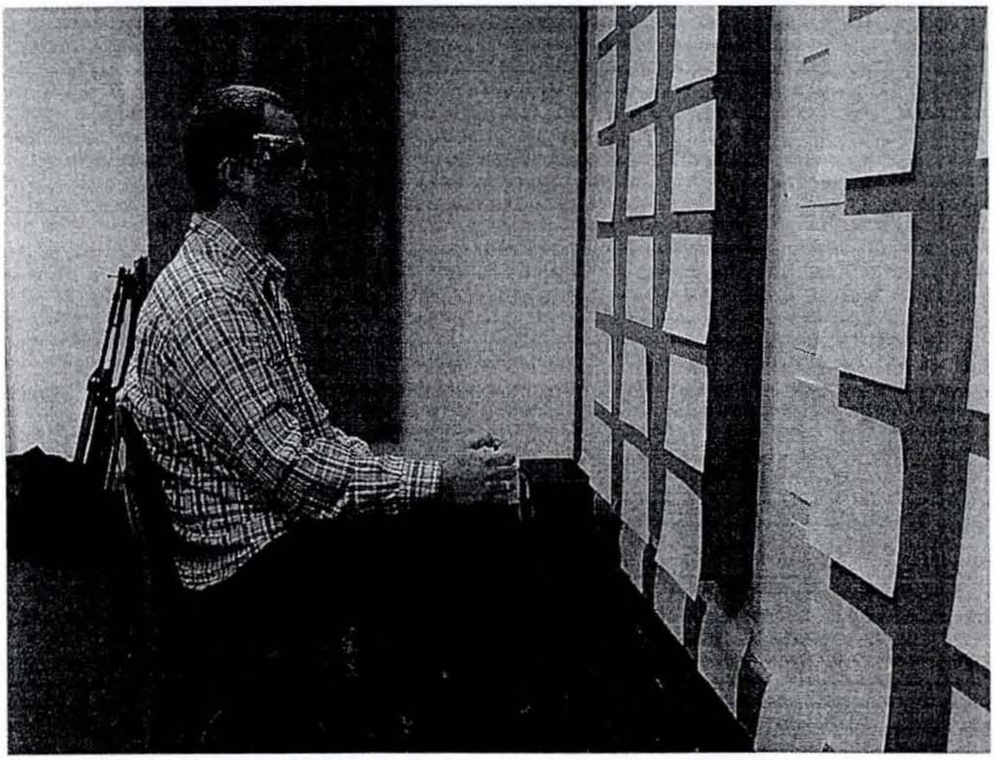


Figure 7.6 – 'SoundWall' and object listener – right perspective.

The entire 'SoundWall' and object listener are shown in the picture above in Figure 7.4. The paper squares on the studio wall represent the indexed positions of the eighty sound source locations. The pictures above in Figures 7.5 and 7.6 show the 'SoundWall' and the object listener from both the left and right hand sides.

The hardware configuration used to construct the three dimensional 'SoundWall' and spatially located control events was:

- Sound source generator – a multi-timbral Roland Boss DS 330 synthesiser with 'real' sampled instruments driven by a PC running Steinberg's CUBASE VST24.
- Sound source radiator –a high quality acoustic loudspeaker with a frequency range of 50Hz to 11kHz (enough to cover the audible range).
- Binaural microphone pair – pair of ELECTRET condenser microphones placed in the ear canals of the object listener.
- An ELECTRET condenser microphone interface – required to drive the microphone pair.
- A two-channel digital recording device – a PC running Steinberg's WaveLab.

This configuration is diagrammatically represented in Figure 7.7.

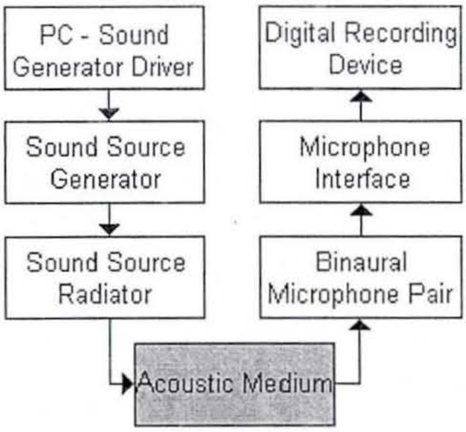


Figure 7.7 – Auditory scene generation system.

The hardware consisted therefore of a straightforward sound source recording configuration with the addition of some filtration supplied by the Acoustic Medium. These filtration effects of this acoustic medium are dependent upon the location of the sound source relative to the binaural microphones and the echoes and properties of the object listener and recording environment. The hardware configuration employed during the recording phase corresponding to this diagrammatical representation is shown in the photograph in Figure 7.8.

Figure 7.8 – Auditory scene generation system hardware set-up.

As previously mentioned a pair of binaural microphones were required in order to facilitate the binaural recording of the auditory scene. These microphones each require secure placement within each ear canal of the listener. In order to accomplish this, 'hearing aid' like devices were fabricated. The diagram in Figure 7.9 shows the initial design of the hearing aid which consisted of the ELECTRET condenser microphone, lead wire and a pinna clasp. The pinna clasp is made from rigid 2.5mm diameter wire and coated with a rubber sheath to aid comfort when being worn by the object listener. The pinna clasps were then customised to the dimensions of the test listener.

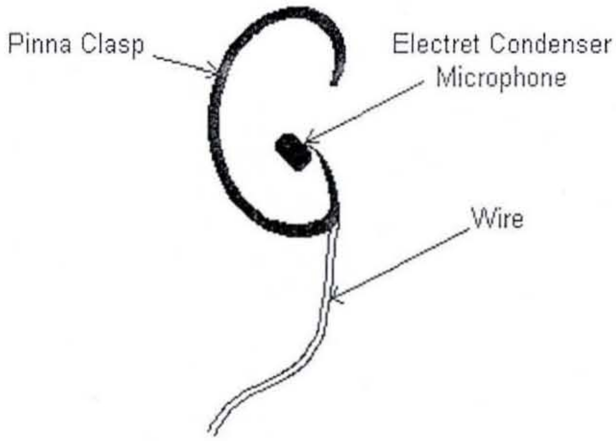


Figure 7.9 –Binaural microphone and pinna clasp – conceptual.

The picture given below in Figure 7.10 shows the object listener's pinna with the custom formed pinna clasp and microphone assembly attached in position.

Figure 7.10 –Binaural microphone and pinna clasp.

ELECTRET condenser microphones are not passive transducer devices and therefore require some simple interface circuitry. Each microphone was interfaced with the circuit shown in the diagram given in Figure 7.11 below.

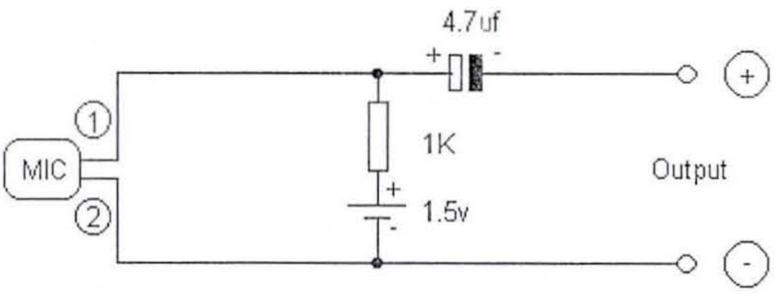


Figure 7.11 –Binaural electret condenser microphone interface circuit diagram.

After initial construction, the author tested the realism of the resulting binaural recordings created with the microphone pair. This initial testing took the form of recording two real world sounds onto a portable digital recording device. The author recorded the auditory environment typically encountered when crossing a busy road, stopping in the centre island to capture the sounds of motor vehicles passing in both the front and rear planes. The second sample consisted of a simple helicopter sample played and moved to give the impression of the helicopter flying toward the listener object, circularly around the head of the listener object and flying away off to the left field. The results of these preliminary tests yielded most favourable results and the effects were quite dramatic, even between different listeners. In order to make the binaural recording more precise, further tuning of the matching of the characteristics of the two ELECTRET microphones was carried out to minimise localisation error in the horizontal plane.

7.4. Hardware considerations

It is necessary to ensure that both left and right binaural microphones are as closely matched as possible in order to create faithful 3-D recordings. Gross mismatching causes unbalanced recordings that lead to inaccurate sound source placement. Preliminary experimentation which compared two randomly selected ELECTRET microphones showed a noticeable difference in the recorded amplitudes. Indeed the highest measured difference in this case was up to 6dBs.

Four ELECTRET microphones were therefore selected, at random, and the amplitude and frequency responses of each were analysed. Only four were chosen to reduce complexity and to eliminate only 'rogue' devices with significantly different characteristics. The two most closely matched microphones were then selected for use in our binaural recordings. The frequency range investigated covered the range 20Hz through to 20kHz, since this more than covers the audible range and facilitates CD quality recordings with a sample frequency of up to 44kHz. Nyquist's criterion states that to have effective sampling, the sampling frequency [Sample Frequency 40kHz] must be twice that of the highest frequency component in the sample [20kHz]. Since the frequency response of the microphones used was 50Hz to 8kHz, this more than covered the desired range of frequencies. The frequency response of each microphone was investigated using a 3D Frequency-Time plot, and both waveforms and peak amplitudes were examined.

Each frequency and amplitude plot had common features - peaks and troughs that did not appear in the reference plot (Figure 7.12). These features were due to the frequency response characteristics of the hardware used in this experiment. Thus, common elements that differ from the source could be attributed to signal degradation through the hardware. The important features to be compared were those that differ between microphones and not those that differ between microphone and source. All speakers and microphones were obtained from Maplin electronic supplies. The speaker was chosen for its frequency response, which accommodated those of the microphones. The characteristics of speaker and microphones are shown in Figures 7.11a and 7.11b.

6.5 inch High Fidelity Loudspeaker.	
Frequency Response	50Hz to 11kHz
Impedance	8 Ohms
Coil Diameter	20mm
Chassis Diameter	166mm
Free Air Resonance	55Hz +/- 8Hz
Acoustic Output	89dB

7.11a. Loudspeaker specification.

Omni directional Electret Condenser Microphone.	
Frequency Response	50Hz to 8kHz
Impedance	1 kOhm max
Power Supply	1.5v to 10v
Sound Pressure Level	120dB max
Sensitivity	-62dB +/- 3dB

7.11b. Electret condenser microphone specification.

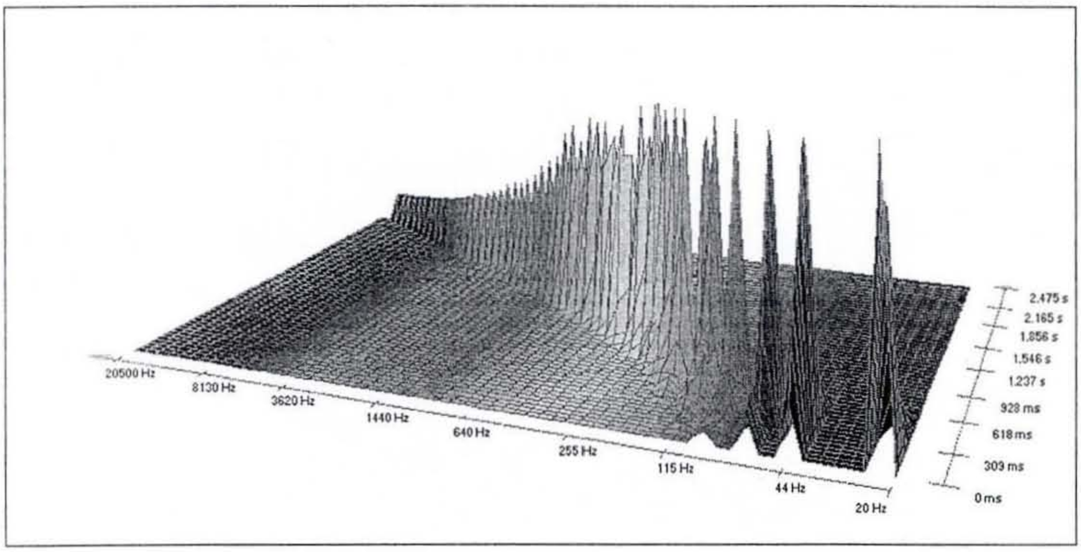


Figure 7.12 – Generated source sweep

Figure 7.12 shows the frequency sweep in the frequency domain and Figure 7.13 shows the same frequency sweep in the temporal domain starting at 20Hz and sweeping through to 20kHz, it also shows that the waveform is at maximum amplitude (-0dBs) throughout the desired frequency range.

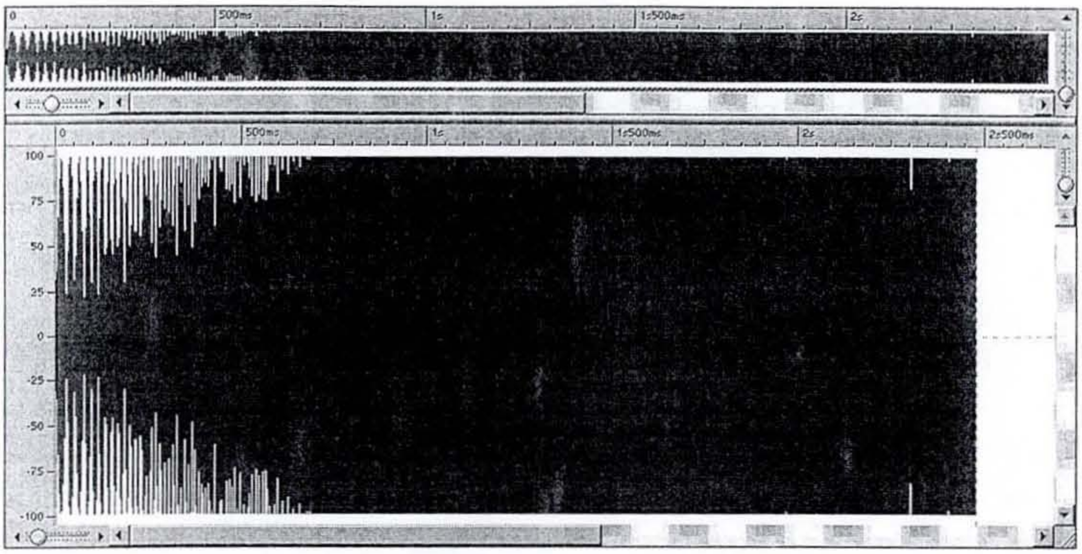


Figure 7.13 – Generated Source Waveform.

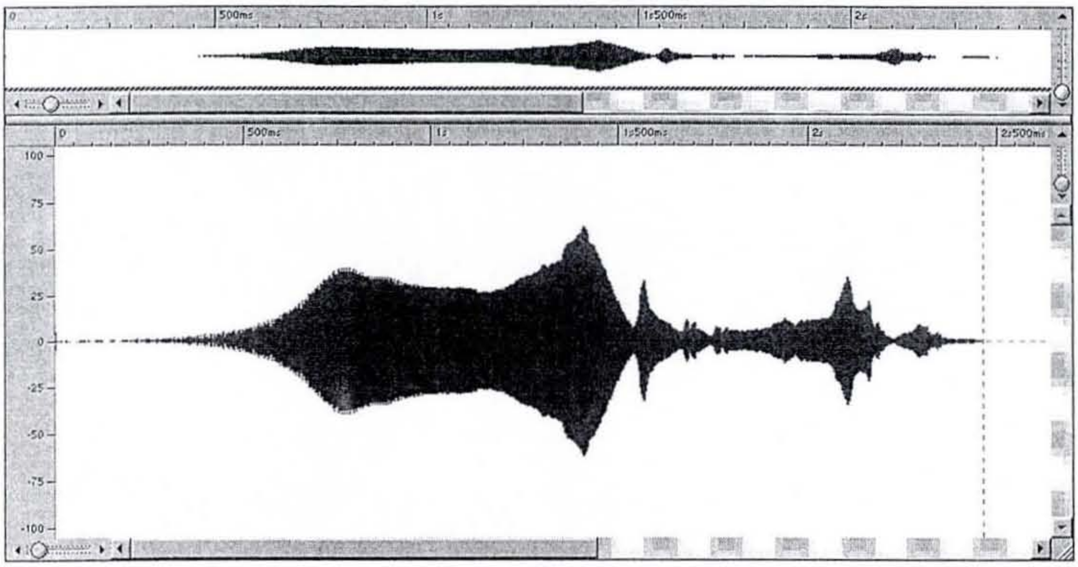


Figure 7.14 –Microphone 1 Waveform.

In the waveform taken from recording through microphone 1, shown in Figure 7.14, it can be seen how the response of both the speaker and the microphone contributed to signal degradation, this becomes even more evident towards the higher frequencies where the amplitude is significantly attenuated. In comparison to the source signal, a peak amplitude was measured using a peak level meter. This measurement indicated that the recorded signal was **-4.1dBs** down from the source at its highest peak.

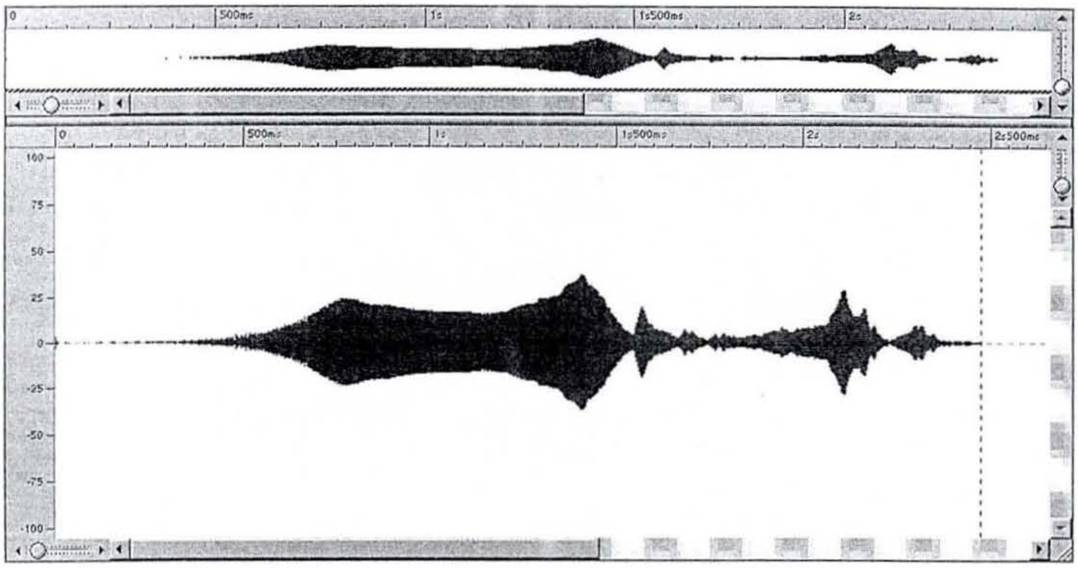


Figure 7.15 –Microphone 2 Waveform.

In the waveform taken from recording through microphone 2 shown in Figure 7.15 it can again be seen how the speaker and microphone both contributed to common signal degradation. The envelope is much the same as that of the waveform obtained using microphone 1, the main difference however is the overall amplitude - it can be seen that this envelope is much smaller.

The measurement taken using the peak level indicator showed that the waveform obtained recording through microphone 2 was **–8.6dBs** down from the reference signal source at its highest peak. This was an extra **-4.5dBs** down compared to that of microphone 1.

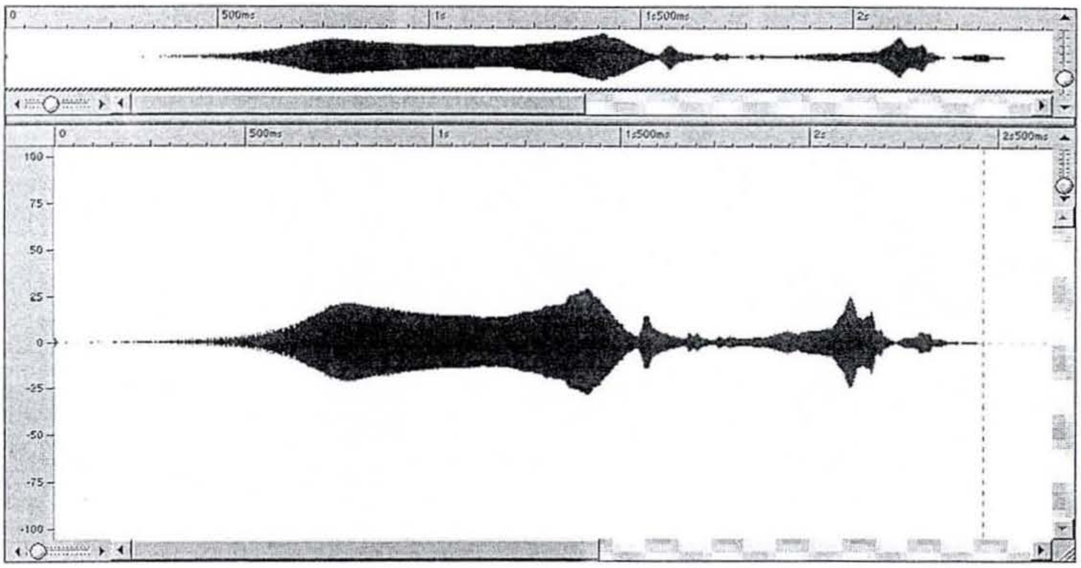


Figure 7.16 – Microphone 3 Waveform.

In the waveform taken from recording through microphone 3 shown in Figure 7.16 it can yet again be seen how the speaker and microphone both contributed to common signal degradation. The shape of the envelope is much the same as that of the waveforms obtained using microphones 1 and 2, again the main difference is the overall amplitude - it can be seen that this envelope is small as in the waveform for microphone 2.

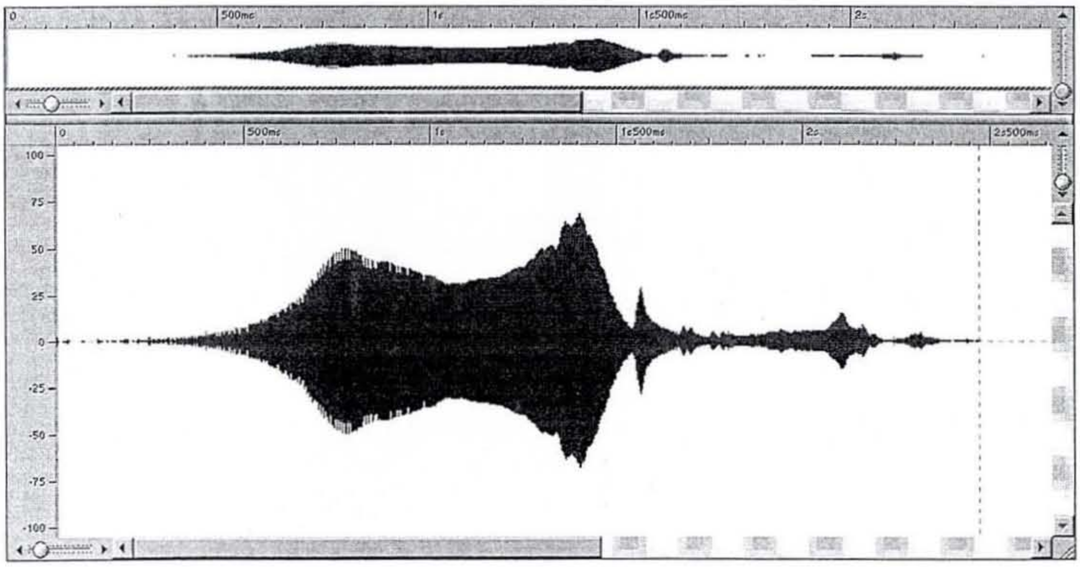


Figure 7.17 – Microphone 4 Waveform.

The measurement taken using the peak level indicator showed that this waveform from microphone 3 was **–10.7dBs** down from the reference signal source. This shows an extra **-6.6dBs** down compared to that of microphone 1 and an extra **–2.1dBs** down compared with microphone 2.

The waveform taken from recording through microphone 4 shown in Figure 7.17 again shows how the speaker and microphone both contributed to common signal degradation. The envelope shape is much the same as that of the waveform obtained using microphones 1, 2 and 3, the main difference again however is the overall amplitude. It can be seen that this envelope is much larger than those of microphones 2 and 3. It is closest in terms of amplitude to the waveform obtained using microphone 1. The measurement taken using the peak level indicator showed that this waveform was **–3.2dBs** down from the reference signal source at its highest peak. This is comparable to the characteristics associated with microphone 1 which is an extra **-0.9dBs** (being at **–4.1dBs**).

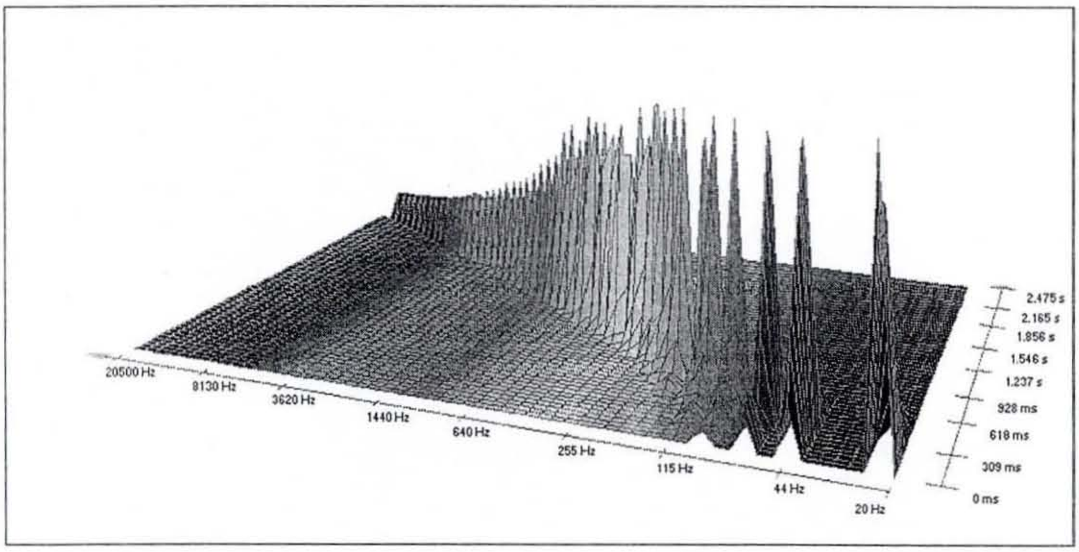


Figure 7.18 – Generated Source Frequency Response.

Figure 7.18 shows the frequency sweep starting at 20Hz and sweeping through to 20kHz over a period of time T=2.5s (approx.)

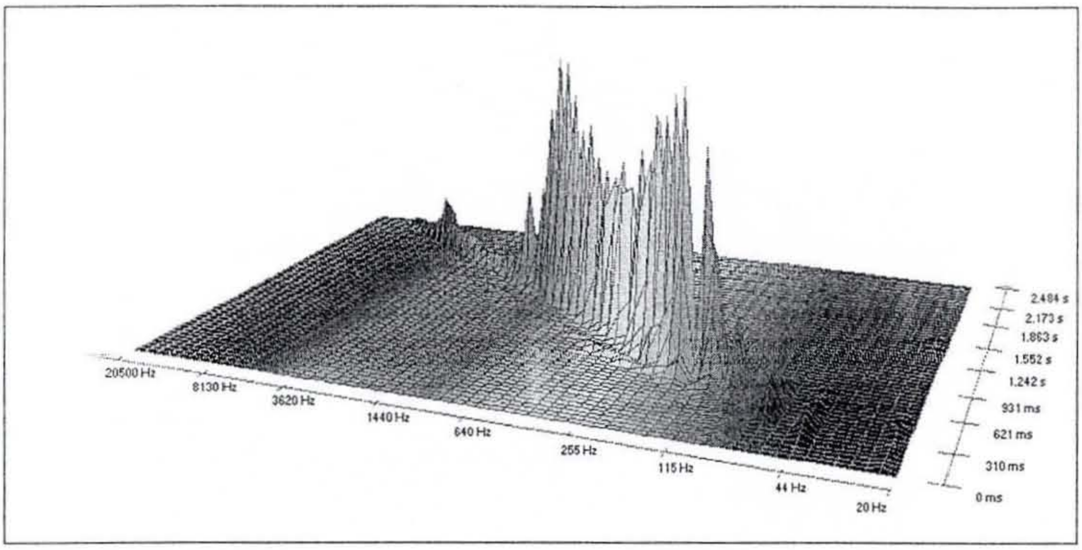


Figure 7.19 – Microphone 1 Frequency Response.

It can be seen from the frequency response plot shown in Figure 7.19 above for microphone 1 that there was significant attenuation of frequencies outside of the range 115Hz to 1400Hz, this attenuation is largely due to the characteristics of both the speaker and the microphone. Very high and very low frequencies did not transfer very well with the given acoustic hardware. As previously mentioned, of greater interest are the differences between the microphones and not the differences between the microphone and source, although this plot does yield some useful information.

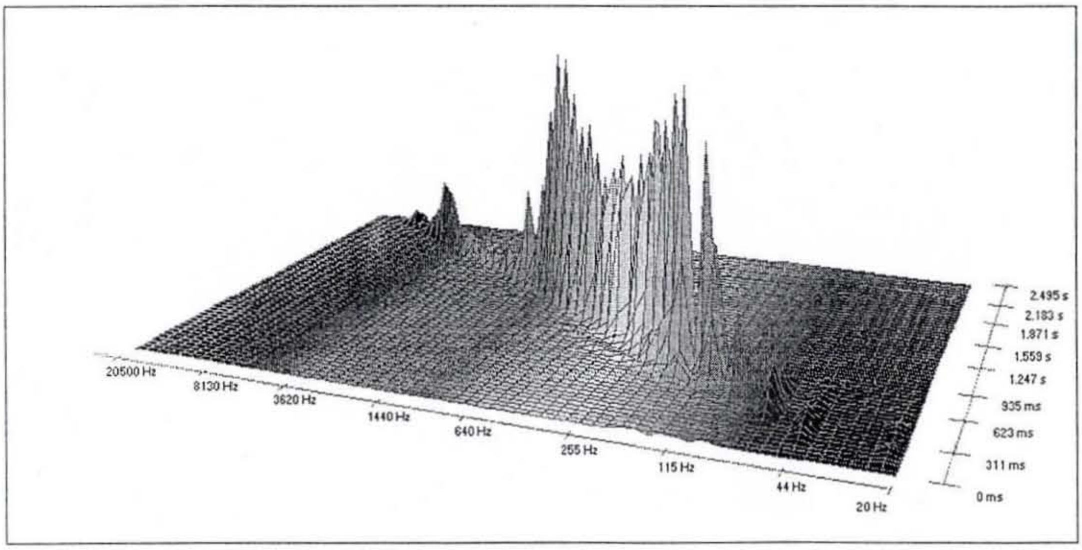


Figure 7.20 – Microphone 2 Frequency Response.

It can be seen from the frequency response plot shown in Figure 7.20 for microphone 2 that again there was significant attenuation of frequencies outside of the range 115Hz to 1400Hz, again this attenuation is largely due to the characteristics of both the speaker and the microphone. It can be seen that there are no significant differences between microphones 1 and 2 when looking at their respective frequency responses.

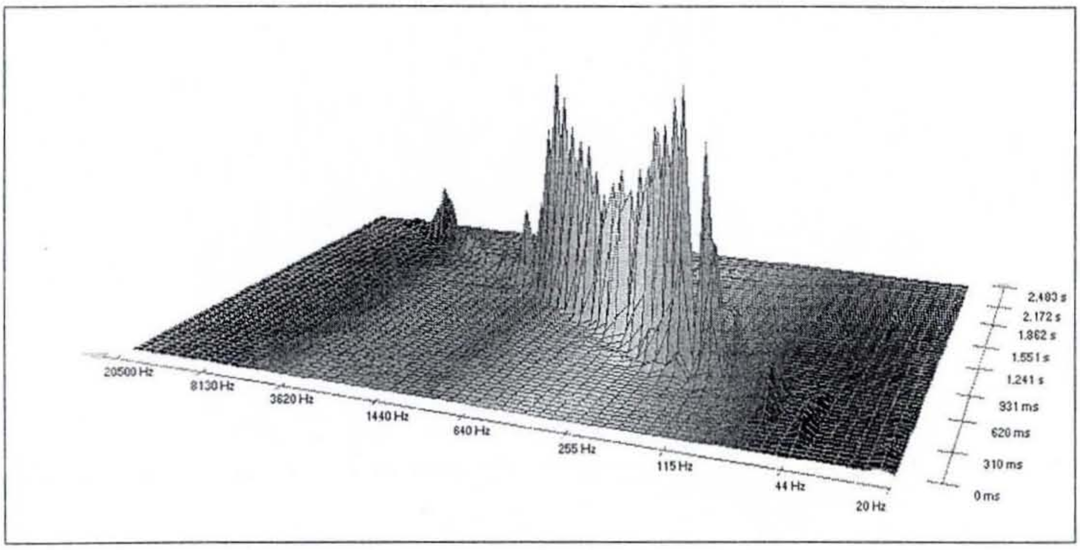


Figure 7.21 – Microphone 3 Frequency Response.

It can be seen from the frequency response plot shown in Figure 7.21 for microphone 3 that again there was significant attenuation of frequencies outside of the range 115Hz to 1400Hz, again this attenuation is largely due to the characteristics of both the speaker and the microphone. It can be seen that there are no significant differences between microphones 1, 2 and 3 when looking at their respective frequency responses.

It can be seen from the frequency response plot shown in Figure 7.22 for microphone 4 that again there was significant attenuation of frequencies outside of the range 115Hz to 1400Hz, again this attenuation is largely due to the characteristics of both the speaker and the microphone. It can be seen that there are no significant differences between microphones 1, 2, 3 and 4 when looking at their respective frequency responses. However microphone 4 did attenuate the frequencies around 8kHz more than any of the other microphones did.

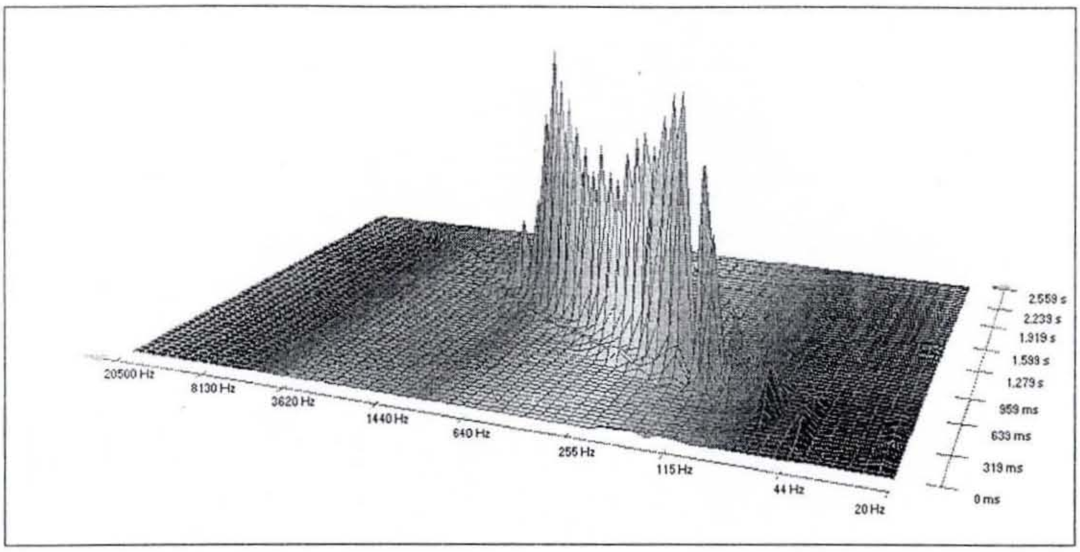


Figure 7.22 – Microphone 4 Frequency Response.

When comparing the frequency response of all four microphones there is negligible difference with the exception of microphone 4 attenuating the frequencies around 8kHz more than the other microphones. The technical specifications of the microphones state that their frequency response is 50Hz to 8kHz, this can be seen in all four frequency plots where the attenuation increases to almost fully suppress any frequencies outside of this response range. This response is also aided by the characteristics of the speaker having a frequency response of 50Hz to 11kHz as stated in the technical specifications. When comparing the waveform amplitudes for each microphone there are significant differences. To show this we have compared all microphones with each other in Figure 7.23 to highlight the differences.

	MIC 1 -4.1dB	**MIC 2** -8.6dB	**MIC 3** -10.7dB	**MIC 4** -3.2dB
MIC 1 -4.1dB	* *	**+4.5dB**	**+6.6dB**	**-0.9dB**
MIC 2 -8.6dB	**-4.5dB**	* *	**+2.1dB**	**-5.4dB**
MIC 3 -10.7dB	**-6.6dB**	**-2.1dB**	* *	**-7.5dB**
MIC 4 -3.2dB	**+0.9dB**	**+5.4dB**	**+7.5dB**	* *

Figure 7.23 – Microphone amplitude comparisons.

Comparing all of the microphones with each other shows us that the most closely matched pair of microphones are microphones 1 and 4 with a difference of **0.9dBs** between them. Microphones 2 and 3 are also a close match with a difference of **2.1dBs**. The largest difference exists between microphones 3 and 4 with a difference of **7.5dBs**.

The best matched pair of microphones were therefore decided to be microphones 1 and 4. The first reason for choosing this pair is due to them having the smallest difference in peak level of 0.9dBs. The second reason is that they are more sensitive than microphones 2 and 3, this is shown in the individual peak levels. Microphones 1 and 4 were only **–4.1dBs** and **–3.2dBs** down from the source compared with microphones 2 and 3 (which were **–8.6dBs** and **–10.7dBs** down from the source). If the matching of the equipment required greater precision then the difference in peak levels could be matched with greater accuracy by using a potentiometer.

The other hardware used in this experiment may also have contributed considerably to the frequency responses obtained for each microphone, since the responses are limited by the characteristics of the source speaker. However, the points of interest are the differences between pairs of microphones as opposed to microphones and the source. Although more detailed characteristics of these types of microphone might have been obtained if a speaker with an overall broader frequency response had been used, the important differences between the microphones have been determined, the main differences being the recorded peak wave amplitude of the microphones and the general frequency response being of negligible difference.

Chapter 8
Experimentation with SIMBAA 3D

8.1. Introduction

In Chapter 6 it was shown how sorting algorithms can be musically auralised so that they can be understood by the average user. In this chapter, a set of experiments is described which were designed to determine if the understanding of algorithms presented aurally to an average person (i.e., non musically educated) would be improved if spatial distribution of the sounds was used in support. In other words - by how far would use of the 'SoundWall' in SIMBAA 3-D enhance user understanding compared with presenting the algorithms aurally alone?

8.2 Research approach

8.2.1 Musical structure and understanding

We have already shown that users with no special musical ability can comprehend sorting algorithms aurally, provided that appropriate mappings are chosen. Spatial distribution provides an additional dimension and might add new capabilities to shape understanding since the shapes can be represented in space as well as in sound. The 'SoundWall' provided by SIMBAA 3-D enables much more accurate aural placement to be realised, and the research question is: can users take advantage of this additional information thereby allowing more complex ideas to be auralised?

The first step is therefore to investigate how much spatial information users can understand in the SIMBAA 3-D environment. Questions that need to be answered include:

- How accurately can users discern spatially distributed musical tones?
- Can users distinguish between different note sequences on the 'SoundWall'?
- Can users visualise the shape of spatially distributed tonal sequences?

- Can users comprehend patterns of spatially distributed tones that denote the presence of a structure or control?
- At what level can users comprehend spatially distributed rhythms and tunes?
- To what accuracy can users identify and distinguish different spatially distributed Timbres?
- How useful is Timbre placement in the 3D spatial audio field?

Firstly, experiments have been performed, using basic structures involving pitch, in order to determine how well users can perceive tones when presented using spatially distributed music.

Secondly, experiments have been carried out that use short spatially distributed musical sequences in order to understand how listeners perceive the shape of tonal patterns when represented in a 3D spatial environment. If the spatial distribution in the 'SoundWall' can assist users in understanding the contour of the sequence then this could open up additional possibilities for aural interfaces.

Thirdly, experiments have been carried out which contain only one or two notes incorrectly placed in otherwise ordered lists. These experiments have been performed in order to investigate if comprehension is improved through the use of spatial distribution to further disambiguate the musically represented information.

Fourthly, experiments have been carried out that use spatially distributed musical sequences, with the addition of a second spatially located timbre to denote the manipulation of the incorrectly placed data elements.

The mappings employed in auralising the sorting algorithms use continuous changes of musical structure to communicate the state of the list and the rearrangements that occur.

They are similar to those used in the earlier documented 2D experiments:

- Spatially located ordered and non-ordered pitch ranges.
- Rhythm in combination with spatially distributed pitch.
- Temporal arrangements and pitch comparisons between one or two spatially located instruments.
- The development of a pattern of what the algorithm does without the listener knowing its detailed processing.
- The abstract development of mental models of current list states.

The results from each of the experiments will then be compared to the earlier 2D representations (in Chapters 5 and 6) in order to determine how far spatial distribution has improved users' perception and understanding.

8.2.2 Tools used

The experiments with spatially distributed music in this section have been implemented on an IBM compatible computer equipped with a soundcard capable of supporting the CD quality format of 44.1kHz / 16-bit. The sound is provided by the 'SoundWall' , which was described in the previous chapter.

Thirty subjects were classified according to their musical ability (the test has been described earlier and is given in Appendix B). The group of test subjects are identified as Group 3.

8.3. Pitch perception experiments

8.3.1. Experiment construction

In this set of experiments, like those using the 2D stereophonic MIDI output in Chapter 5 (Section 5.3), thirty subjects were asked to listen to pairs of musical notes, and determine their position within a bounded diatonic scale. The experiment construction was the same as the one given in Section 5.3 with the exception of the added spatialisation. The timbre employed was an acoustic grand piano, whose position on the Sound Wall was dependent upon pitch. The linear mapping between pitch and vertical placement and position and horizontal placement produced a line on the 'SoundWall' that corresponded to the scale starting from bottom-left on the 'SoundWall' up to top-right. Each of the test pitches was placed along this virtual scale-line. Instrument play-back used binaurally recorded sound with no reverberation or chorus added. The instrument was kept in its pure form to maintain the binaural cues obtained through the recordings.

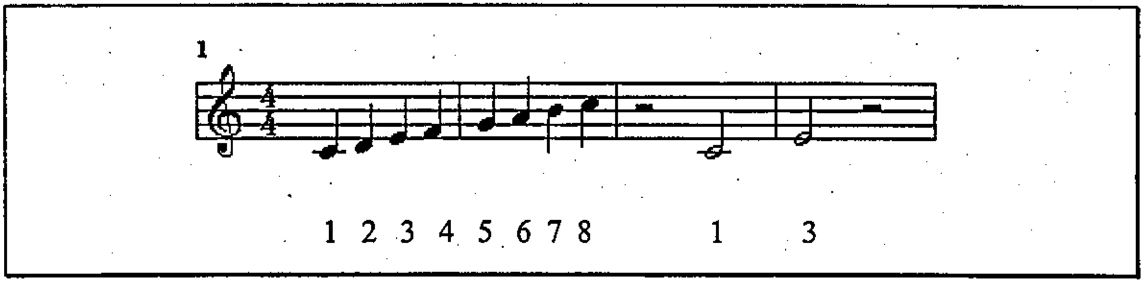



Figure 8.1 – Spatially distributed pitch perception example.

Figure 8.1 was both shown and played to listeners. The scale and test pitches were all mapped to linearly related horizontal and vertical locations on the 'SoundWall', elements with a value of 1 appearing to the bottom-left of the wall and elements with a value of 8 appearing to the top-right of the 'SoundWall'. This is demonstrated in the diagram given below in Figure 8.2 that shows the distribution for Figure 8.1.

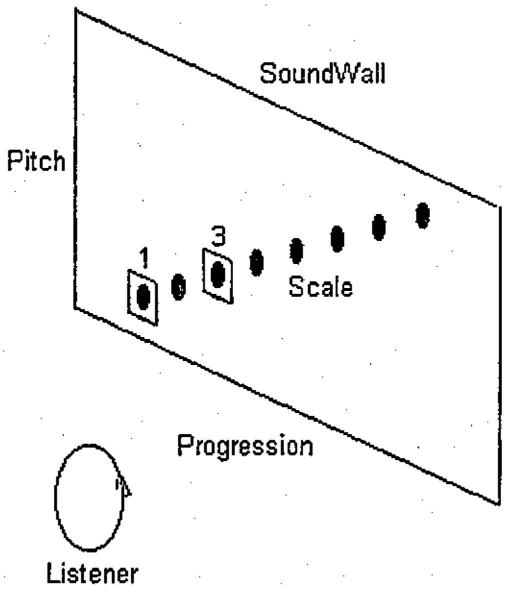


Figure 8.2 – Spatial location and movement of the pitch perception example.

8.3.2. Results and analysis

Figure 8.3 shows the musical ability distribution of the group of thirty test subjects known as Group 3. Of this test group, 14 have a musical ability score of 2 and 16 have a musical ability score of between 3 and 5. Therefore the test group consists of 14 'non-musical' listeners and 16 'musical' listeners.

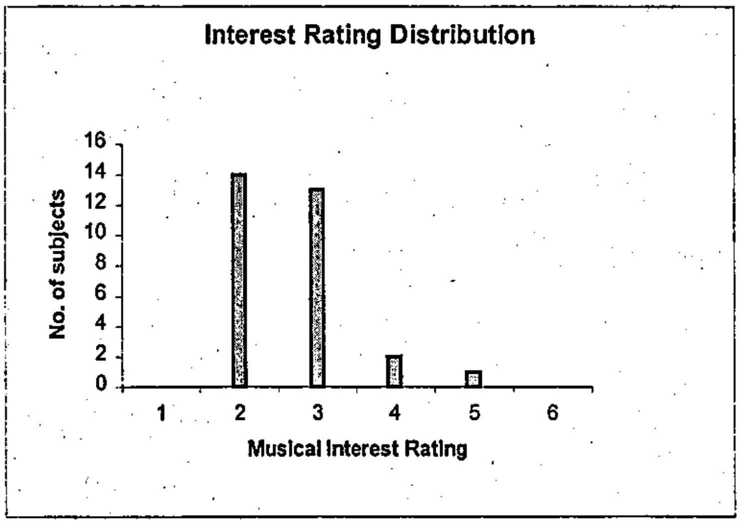


Figure 8.3 – Musical interest rating for pitch perception test subjects.

Figure 8.4 shows the users' perception of each of the tones separately. The plot indicates the accuracy of each 'absolute' tone within the bounded diatonic single octave scale. Thus the results have been analysed as twenty individual notes (even though they were presented as pairs).

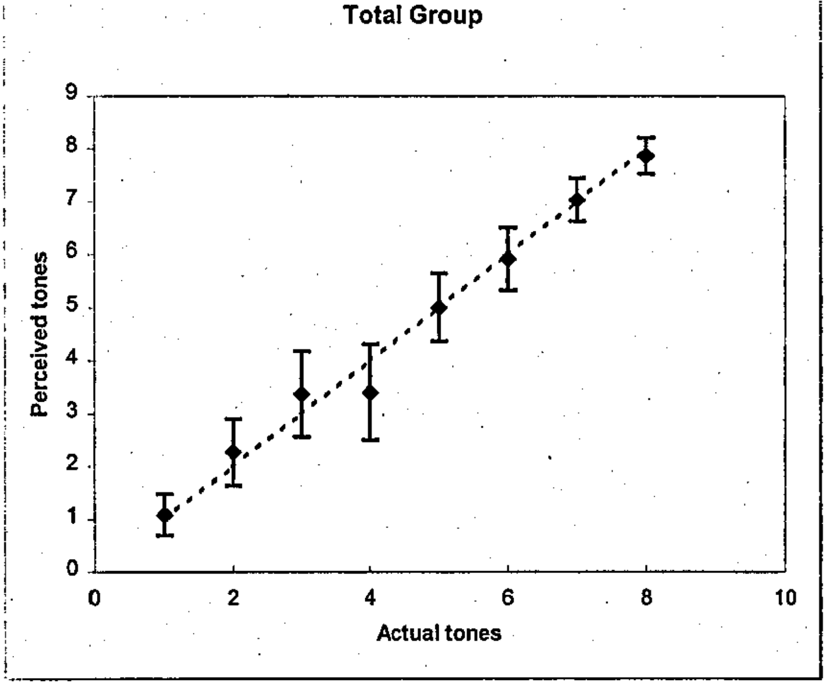


Figure 8.4 – Perceived tone accuracy for individual pitch perception.

Note	1	2	3	4	5	6	7	8
Mean	1.085714	2.266667	3.366667	3.4	5	5.916667	7.033333	7.866667
S.D	0.381743	0.63424	0.808717	0.905726	0.635999	0.590652	0.413841	0.342803
Hi	1.467458	2.900906	4.175384	4.305726	5.635999	6.507319	7.447174	8.20947
Lo	0.703971	1.632427	2.55795	2.494274	4.364001	5.326014	6.619492	7.523863

Figure 8.5 – Table of perceived tone accuracy for pitch perception.

Figure 8.5 indicates the mean perception for each of the notes, the standard deviation and the high and low boundaries. It can be seen that notes that fall close to the boundaries of the scale are identified with greater accuracy than those that appear in the middle of the scale (as was documented in Chapter 5). This is because the scale that provides the boundaries gives fixed points that the user can more readily recall. The middle of the scale has little information and can create an area of ambiguity. Overall, the group performed well, however it is interesting to see how subjects performed when separated into their musical classification groups. Figure 8.6 compares the accuracy of the 'musical' and 'non-musical' groups.

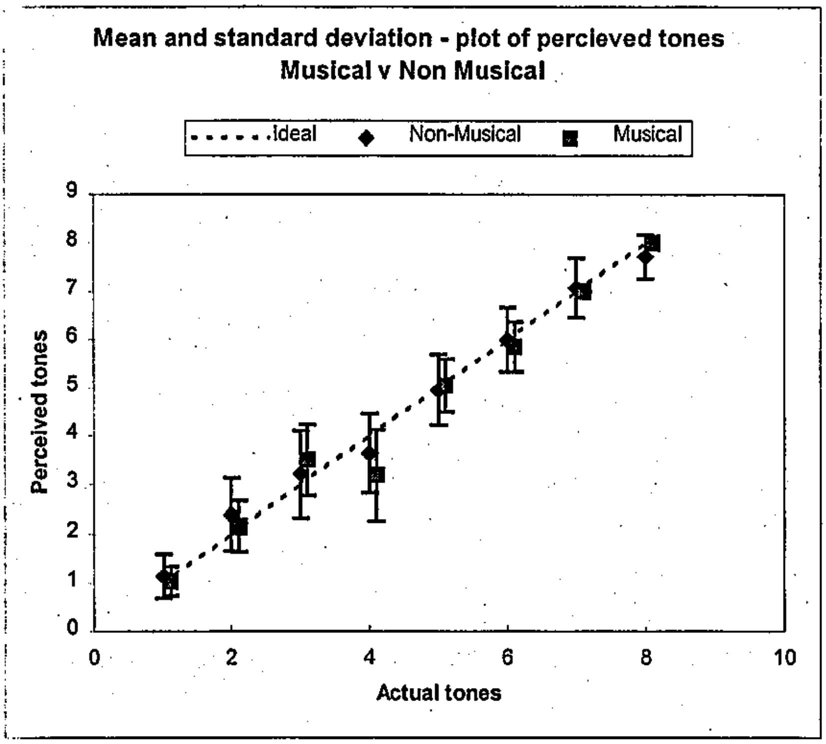


Figure 8.6 – Perceived tone accuracy for pitch perception of musical and non-musical listeners.

Musical	Note 1	Note 2	Note 3	Note 4	Note 5	Note 6	Note 7	Note 8
Mean	1.035714	2.15625	3.5	3.1875	5.041667	5.84375	7	8
S.D	0.297998	0.514899	0.730297	0.931094	0.54415	0.514899	0	0

Non-Musical	Note 1	Note 2	Note 3	Note 4	Note 5	Note 6	Note 7	Note 8
Mean	1.142857	2.392857	3.214286	3.642857	4.952381	6	7.071429	7.714286
S.D	0.454077	0.737327	0.892582	0.82616	0.730933	0.666667	0.615728	0.460044

Figure 8.7 – Table comparing perceived tone accuracy for pitch perception of musical and non-musical listeners.

It can be seen in Figure 8.7 that the accuracy of the 'musical' listeners is greater than that of the 'non-musical' group. The inaccuracy appears close to the middle of the scale where perception seems to be most ambiguous.

Figure 8.8b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the differences of perceived notes from the true notes for the 'non-musical' group compared to the 'musical' group. The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving spatially distributed musical notes.

H_1 : The 'non-musical' listeners perform with differing accuracy than the 'musical' listeners when perceiving spatially distributed musical notes.

	ERRNOTE1	ERRNOTE2	ERRNOTE3	ERRNOTE4	ERRNOTEE5	ERRNOTE6	ERRNOTE7	ERRNOTE8
Mann-Whitney U	70.000	87.000	95.000	80.000	103.000	96.500	72.000	48.000
Wilcoxon W	206.000	223.000	231.000	185.000	239.000	232.500	208.000	184.00
	-2.158	-1.188	-.791	-1.420	-.465	-.723	-2.575	-3.472
Asymp. Sig. (1-tailed)	.016	.118	.215	.078	.321	.235	.005	.001

Figure 8.8b – Table of test statistics for each perceived note, 'non-musical' v. 'musical'.

The 'non-musical' subjects perform with greater inaccuracy. The null hypothesis can be rejected at the 5% level for note 1 and at the 1% level for notes 7 and 8. For the remaining notes (2 to 6) there is no significant difference between the 'non-musical' and 'musical' test groups and the null hypothesis cannot be rejected. These data suggest that when notes are played close to the boundaries of the context scale the 'musical' test group perform significantly better than the 'non-musical' test group, whilst there is no significant difference between the two groups for notes that fall into the middle of the context scale where the greatest level of ambiguity and error can be observed.

Figure 8.9b shows the overall impact of the addition of spatial distribution and compares the data for this series of experiments with the data obtained for the non-3D pitch tests carried out in Chapter 5 using the Wilcoxon Signed Ranks non-parametric test.

The hypotheses are:

H_0 : The addition of spatial distribution has had no significant impact on the whole group's perception of absolute pitch.

H_1 : The addition of spatial distribution has significantly increased the accuracy of the whole group's perception of absolute pitch.

	ErrNote1 - ERRNOTE 1	ErrNote2 - ERRNOTE 2	ErrNote3 - ERRNOTE 3	ErrNote4 - ERRNOTE 4	ErrNote5 - ERRNOTE 5	ErrNote6 - ERRNOTE 6	ErrNote7 - ERRNOTE 7	ErrNote8 - ERRNOTE 8
Z	-2.71	-3.466	-2.76	-2.740	-3.162	-2.538	-2.333	-.540
Asymp. Sig. (1-tailed)	.00	.001	.003	.003	.001	.006	.010	.295

Figure 8.9b – Table of test statistics for each perceived note, 3D v. non-3D.

From this data the null hypothesis can be rejected at the 1% level for all notes except note 8 for which the null hypothesis cannot be rejected. This suggests that the addition of spatial distribution has significantly affected the entire group's perception of absolute pitch. In particular, it has significantly improved the perception of pitch. The fact that no significant difference is observed for note 8 may be due the fact that it is the last note heard in the context scale before the test note pairs are heard. It is therefore the most recently held form of reference in the listener's memory and as such is already subject to high levels of identification accuracy prior to the addition of spatial location, hence the lack of any significant increase in perception accuracy.

The results also need to be analysed in terms of relative pitch tests as opposed to absolute pitch tests. Here, the data that is evaluated is the perceived difference between the two notes and not how accurately they are placed within the scale. The intervals ranged from 1 to 7 with the omission of 2 due to the constraints of the previous absolute pitch experiments. The diagram in Figure 8.10 shows how the entire group of test subjects performed as a whole.

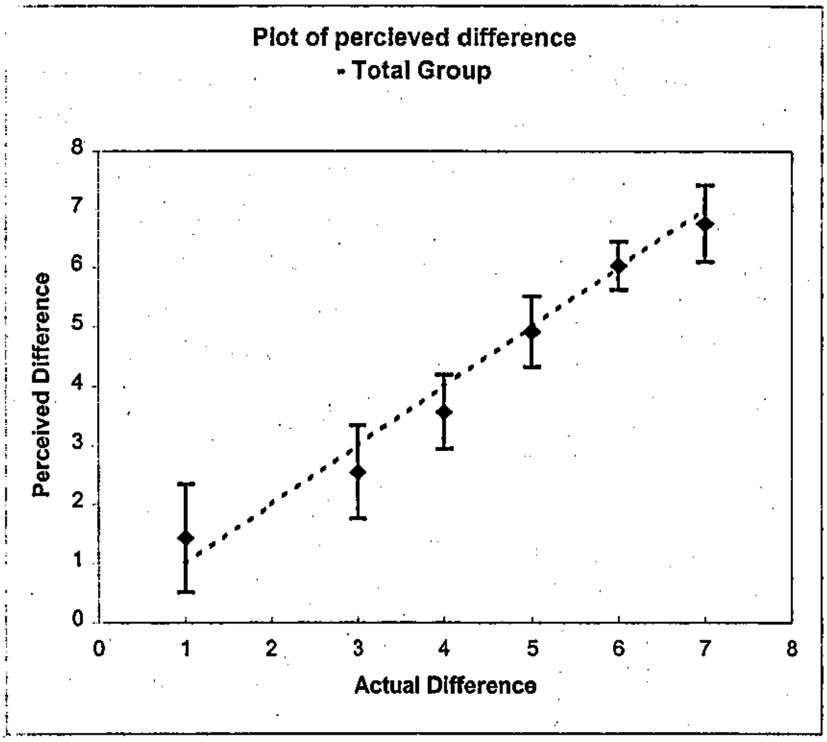


Figure 8.10 – Perceived tonal interval.

Note	1	2	3	4	5	6	7
Mean	1.433333		2.55	3.566667	4.916667	6.033333	6.75
S.D	0.908839		0.790301	0.626062	0.590652	0.413841	0.654191

Figure 8.11 – Table of perceived tonal interval.

Figures 8.10 and 8.11 show that the accuracy of the entire group is again high when perceiving the interval difference between two notes. Again it is important to divide the data into two groups in order to better understand how the 'musical' sub-group performs.

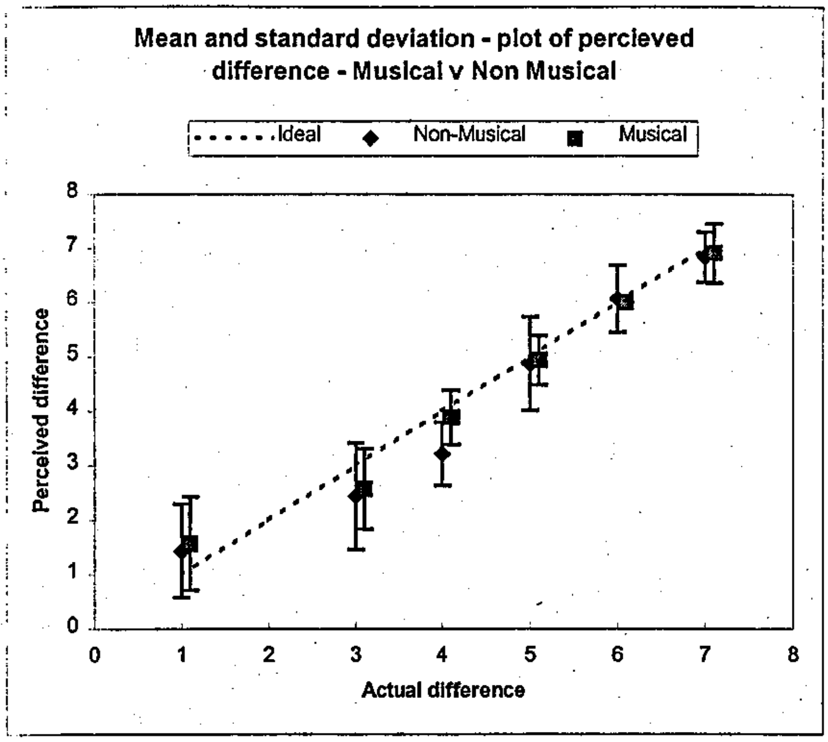


Figure 8.12a – Perceived tonal interval – musical and non-musical listeners.

Musical	Diff 1	Diff 2	Diff 3	Diff 4	Diff 5	Diff 6	Diff 7
Mean	1.566667		2.566667	3.875	4.933333	6	6.9
S.D	0.85836		0.727932	0.5	0.449776	0	0.547723

Non-Musical	Diff 1	Diff 2	Diff 3	Diff 4	Diff 5	Diff 6	Diff 7
Mean	1.433333		2.433333	3.214286	4.866667	6.071429	6.833333
S.D	0.85836		0.971431	0.578934	0.860366	0.615728	0.461133

Figure 8.12b – Table of perceived tonal interval – musical and non-musical listeners.

Figures 8.12a and 8.12b again suggests the 'musical' group perform with greater accuracy than the 'non-musical' group.

Figure 8.13b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the differences of perceived intervals from the true intervals for the 'non-musical' group compared to the 'musical' group. The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving spatially distributed musical intervals.

H_1 : The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving spatially distributed musical intervals.

	ERRDIFF1	ERRDIFF3	ERRDIFF4	ERRDIFF5	ERRDIFF6	ERRDIFF7
Mann-Whitney U	89.000	72.000	58.000	50.000	72.000	89.000
Wilcoxon W	194.000	208.000	194.000	186.000	208.000	225.000
Z	-1.00	-1.78	-2.56	-2.74	-2.575	-1.474
Asymp. Sig. (1-tailed)	.15	.03	.00	.003	.005	.070

Figure 8.13b – Table of test statistics for each perceived interval, 'non-musical' v. 'musical'.

It can be seen from the data given in Figure 8.13b that the null hypothesis can be rejected at the 5% level for interval 3 and at the 1% level for intervals of 4, 5 and 6. For intervals of 1 and 7 the null hypothesis cannot be rejected. This suggests that the 'musical' group perform better than the 'non-musical' group when the intervals are extreme (1 or 7) but that there is no significant difference when the intervals are not extreme. When comparing this data to the data obtained for the non-3D interval pitch tests documented in Chapter 5 it can be seen that the only difference is that large intervals of 7 yield no significant difference between the two groups. This suggests that the addition of spatial distribution has narrowed the margin between the groups for such large intervals. Figure 8.14b shows the overall impact of the spatial distribution by comparing the data for this series of experiments with the data obtained for the non-3D interval pitch tests carried out in Chapter 5 using the Wilcoxon Signed Ranks non-parametric test.

The hypotheses are:

H_0 : The addition of spatial distribution has had no significant impact on the whole group's perception of pitch intervals.

H_1 : The addition of spatial distribution has significantly increased the accuracy of the whole group's perception of pitch intervals.

	ErrDiff1 ERRDIFF1	ErrDiff3 ERRDIFF3	ErrDiff4 - ERRDIFF4	ErrDiff5 ERRDIFF5	ErrDiff6 ERRDIFF6	ErrDiff7 - ERRDIFF7
Z	-3.61	-1.55	-3.000	-3.711	-2.33	-2.469
Asymp. Sig. (1-tailed)	.000	.06	.002	.000	.01	.007

Figure 8.14b – Table of test statistics for each perceived note, non-3D v. 3D.

From this data the null hypothesis can be rejected at the 1% level for all differences except the interval of 3 for which the null hypothesis cannot be rejected. The addition of spatial distribution here has significantly affected the entire group's perception of pitch intervals. The fact that no significant difference has been observed for the interval of 3 may be due to the fact that this interval is between the extremes. This suggests that the addition of spatial distribution has effected the perception of intervals that are nearest to the extremes. The interval that falls between the extremes and shows the least effect of the addition of 3D placement is considered to be in the area of highest ambiguity. The results obtained from the experiment documented in Section 5.3 showed that listeners could successfully identify pitch and intervals, the results obtained in this set of experiments have shown that the addition of spatial location has significantly improved the group's perception of pitch and intervals. This might suggest that the addition of spatial location in sorting algorithm auralisations might aid understanding of the operation and execution of algorithm, pitches and intervals form the basic building blocks for musical algorithm auralisation within the context of this thesis.

8.4. Shape perception experiments

8.4.1. Experiment construction

This experiment was designed in order to help understand how listeners perceive the shape of short sequences of spatially distributed musical notes. This set of experiments are the same as shape tests documented earlier in Chapter 5 (Section 5.4) with the exception of the addition of spatialisation. Again, thirty subjects were asked to listen to the sequences of musical notes and determine their shape within the bounded diatonic scale. The timbre employed was again an acoustic grand piano whose position on the 'SoundWall' was mapped directly to the shape of the tune. More specifically, its position was dependent upon the pitch for the vertical placement and the sequence order progression for the horizontal placement. The sound sources employed were those obtained by the binaural recording of the 'SoundWall'. The stimuli were the same as defined in Section 5.4. The workbook is given in full in Appendix D. The subject group were then asked to draw the shape of a further six tonal sequences by placing 'X' marks in blank grids. Each of these tests was performed on thirty individual test subjects. Figure 8.16 shows how Demo 1 (Figure 5.15, Section 5.4) was mapped into 3D audio space. The stimuli are the same as used in the experiment documented in Section 5.4.

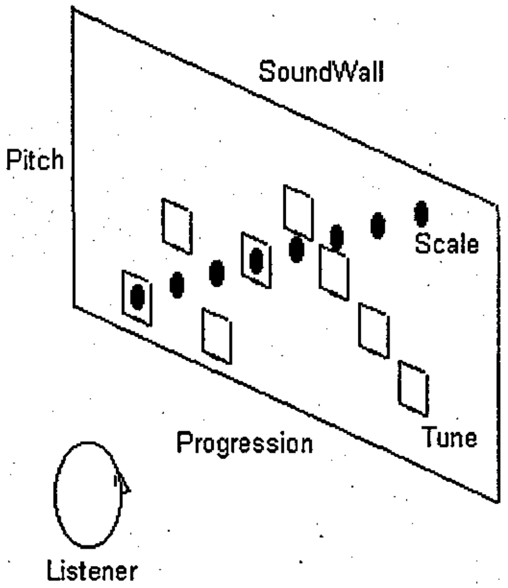


Figure 8.16 – Spatial location and movement of the shape perception Demo1 example.

8.4.2. Results and analysis

The evaluation mechanism was the same as described for the 2D stereophonic (non 3D) shape perception experiments documented in Chapter 5. The test group was the same as was used for the previous experiment, Group 3.

Figure 8.17 shows the accuracy of each of the thirty test subjects for all six of the shape perception questions. The graph has been ordered and colour coded in terms of musical ability. It can be seen that there is a general trend that might suggest that 'musical' test subjects tend to perform slightly better than 'non-musical' test subjects when perceiving the contour of the tonal sequences.

Figure 8.17 shows that of the 'non-musical' test subjects with a musical ability rating of two, five of the listeners performed above 80% accuracy level when perceiving the shapes of the tonal sequences. The remainder still perform well but generally not as accurately as the 'musical' test subjects. These 'musical' listeners perform with no less than 70% accuracy. Since the data, in this case have been combined to give an average score over all six shapes, it is important to ascertain whether certain shapes are perceived more accurately than others and whether the 'musical' group performed differently to the 'non-musical' group for these different information types.

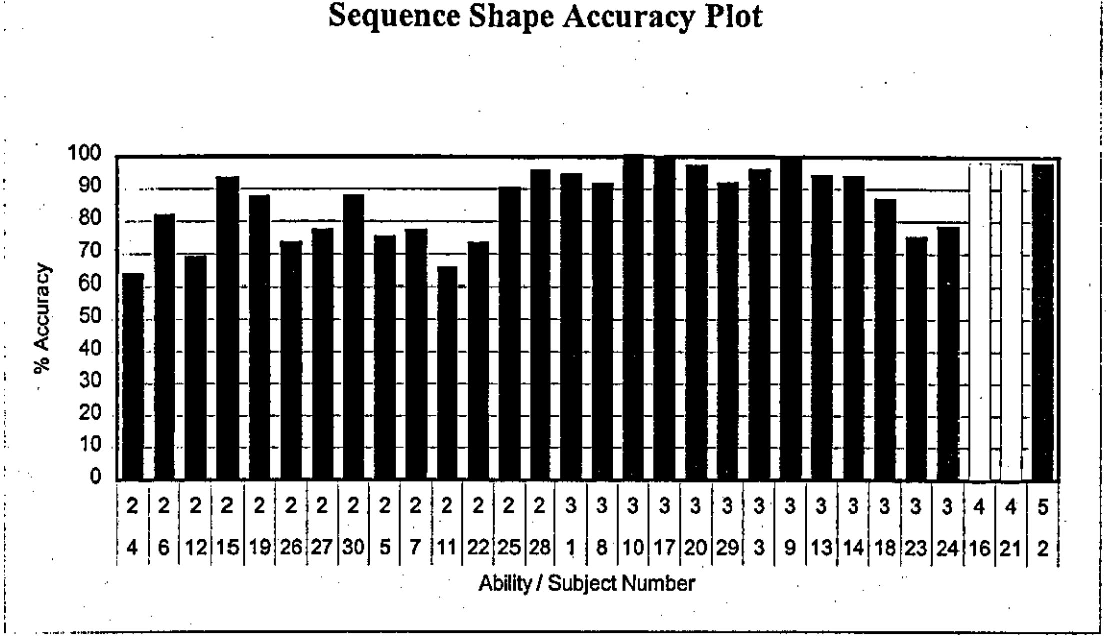


Figure 8.17 – Spatially distributed shape perception accuracy plot – entire test group.

Figure 8.18 shows how the group of test subjects as a whole perform on each of the six shape perception questions.

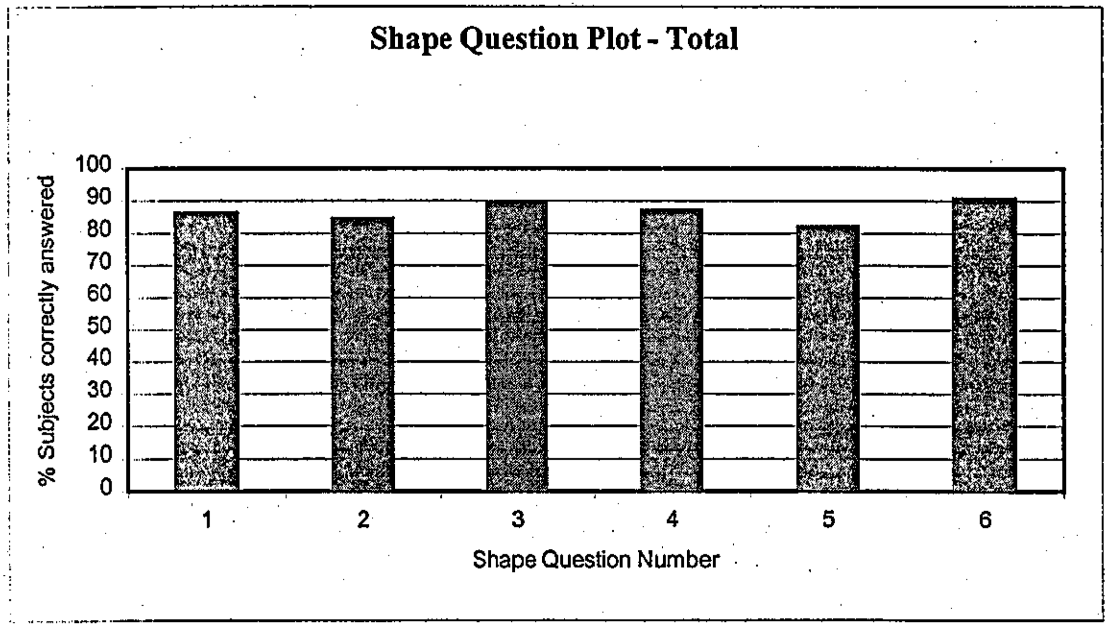


Figure 8.18 – Shape perception accuracy plot by question – entire test group.

Figure 8.18 shows that there is little difference between the perception of the different shapes. The only noticeable difference is the very slight increase in accuracy for the subjects' perception of shapes 3, 4 and 6. The only observable difference in these shapes is that each possesses very prominent and obvious features (Section 5.4).

The best perceived shapes were those that possess long and obvious ascents or descents, or repeated patterns. The shapes that did not translate quite so well each had more complex and less obvious and non-repetitive features. This supports findings by Alty [3] who reported that as shape complexity increases and the number of changes in direction of pitch increases then the understanding and perception accuracy of listeners decreases.

Figure 8.19 shows how 'musical' listeners compared to 'non-musical' listeners in terms of accuracy of perception for each of the six shapes.

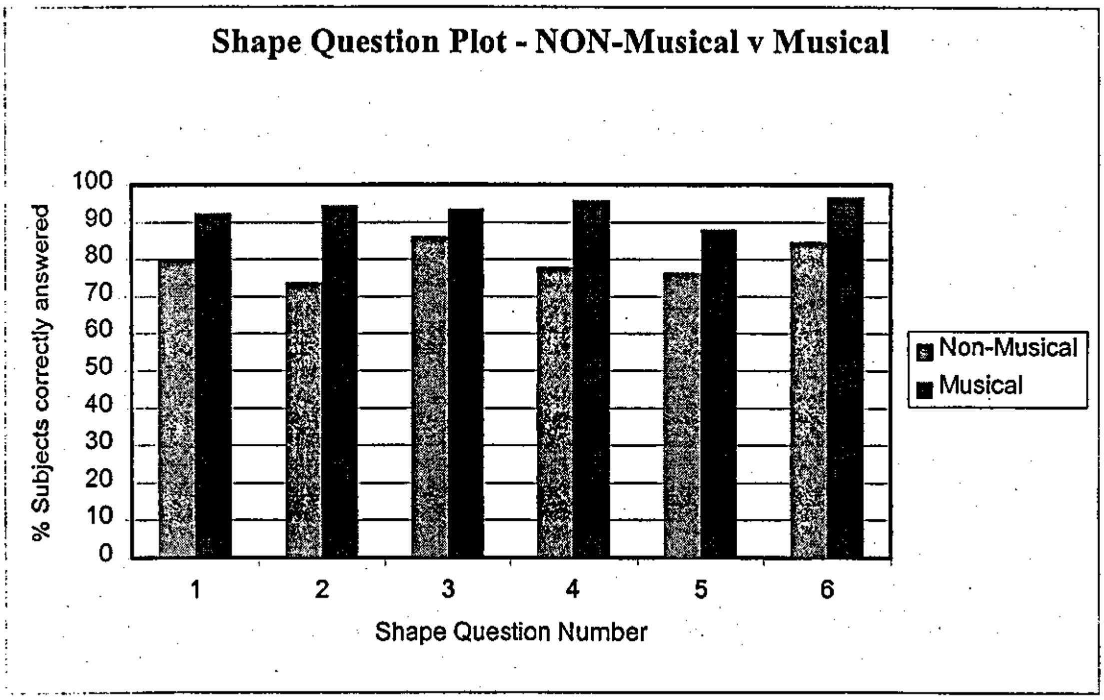


Figure 8.19 – Shape perception accuracy by shape – non-musical v musical listeners.

The same feature is observed for the musically untrained group of listeners as was observed for the group as a whole. Certain obvious or repetitive features translate better than more complex or non-repetitive features.

Figure 8.20b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the scores obtained for the perceived shapes compared to the true shapes for the 'musical' group compared to the 'non-musical' group. The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving spatially distributed musical shapes.

H_1 : The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving spatially distributed musical shapes.

	Q13D	Q23D	Q33D	Q43	Q53D	Q63D
Mann-Whitney U	46.500	24.000	87.000	46.50	55.500	47.500
Wilcoxon W	151.500	129.000	192.000	151.50	160.500	152.500
Z	-2.849	-3.715	-1.06	-3.07	-2.410	-2.78
Asymp. Sig. (1-tailed)	.002	.000	.14	.00	.008	.00

Figure 8.20b – Table of test statistics for each perceived shape, 'non-musical' v. 'musical'.

The null hypothesis can be rejected for all shapes except shape 3 suggesting that there is generally a significant difference between 'musical' and 'non-musical' listeners when perceiving spatially distributed musical shapes. The null hypothesis can be rejected at the 0.1% level of significance for shapes 2 and 4. It can also be rejected for shapes 1, 5 and 6 at the 1% level of confidence. The overall accuracy for all test listeners is observably high. It is interesting to note that overall, shape 3 is the easiest shape to understand, so both groups do particularly well in recognising this shape, hence the lack of significance in the difference between them. This suggests that as the complexity of the shapes increases, the effects of musical training become significantly beneficial. However, it was shown in Chapter 5 that if the musical timing is removed then no difference is observed. In the algorithm auralisations employed in this thesis the representation of lists is performed without any musical timing, where the effect of musical training has been of no benefit.

The data given in the table in Figure 8.21b show the overall impact of the spatial distribution by comparing the data for this series of experiments with the data obtained for the non-3D shape tests carried out in Chapter 5 using the Wilcoxon Signed Ranks non-parametric test.

The hypotheses are:

H_0 : The addition of spatial distribution has had no significant impact on the whole group's perception of the shapes of short tonal sequences.

H_1 : The addition of spatial distribution has significantly increased the accuracy of the whole group's perception of shapes of short tonal sequences.

	Q13D - Q12D	Q23D - Q22D	Q33D - Q32D	Q43D - Q42D	Q53D - Q52D	Q63D - Q62D
Z	-1.81	-1.90	-2.748	-1.007	-2.24	-2.160
Asymp. Sig. (1-tailed)	.035	.02	.003	.157	.01	.016

Figure 8.21b – Table of test statistics for each perceived shape, non-3D v. 3D.

From this data the null hypothesis can be rejected at the 3.5% level for all shapes except shape 4 for which the null hypothesis cannot be rejected. This suggests that the addition of spatial distribution has significantly effected the entire group's perception of short tonal sequences. Although the data suggests a difference in accuracy for shape 4, increased sample size may have resulted in the possibility of rejecting the null hypothesis. The results obtained from the experiment documented in Section 5.4 suggested that listeners could identify patterns and shapes of musically auralised lists of numbers. This set of experiments has shown that the addition of spatialisation significantly improves listeners' perception of the musical shapes and patterns.

8.5. List state perception experiments

8.5.1. Experiment construction

This set of experiments was performed using the same stimuli and same construction as documented in Section 5.5 with the exception of the addition of spatialistaion. Thirty subjects were asked to listen to sequences of eight musical notes that corresponded to the numbers one to eight. Once again the tonal sequences were all within a bounded diatonic octave scale. The timbre employed was an acoustic grand piano whose position on the 'SoundWall' was again mapped directly to the shape of the tonal sequence. More specifically, its position was dependent upon the pitch for the vertical placement and the sequence order progression for the horizontal placement. Again the sound sources employed were those obtained by the binaural recording of the 'SoundWall' and no reverberation or chorus were added. The results indicated that again all of the test subjects successfully identified the states of all of the spatially distributed lists with an accuracy of 100%. Some of the lists were totally random and others were almost ordered with the exception on one element. These results showed that all listeners, regardless of their musical ability and regardless of the addition of 3D spatial location, were fully capable of distinguishing between musically represented sorted and unsorted lists of numbers.

In order to determine the level of understanding a more detailed investigation was required. In this set of further experiments, test subjects were once again asked to listen to sequences of eight notes that represented lists of eight numbers. The questionnaire is given in full in Appendix F. Again these sequences were all played within the same diatonic octave staring from 'Middle C'. The same timbre and positional relationships were also employed. This time listeners were played lists that were sorted into ascending order with the exception of between one and three incorrectly placed elements. Subjects were shown and played the example diagram below in Figure 8.22 that showed how elements 4 and 5 caused a descent in pitch and were therefore incorrectly placed.

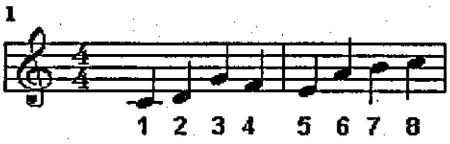



Figure 8.22 – Spatially distributed list state perception example.

The thirty test subjects were then each asked to identify the incorrectly placed elements by circling diagrams for five tests. The diagram shown below in Figure 8.23 corresponds to the spatial location of the instrument for the example given above.

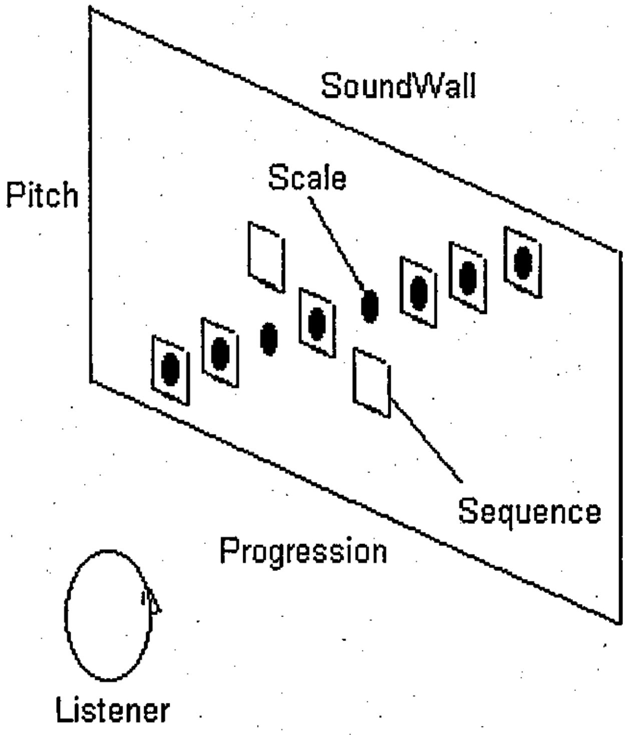


Figure 8.23 – Spatial location and movement of list state perception example.

8.5.2. Results and analysis

The test group used for this set of experiments was Group 3. Figures 8.24 and 8.25 show the users' perception of each of the incorrectly placed elements within the partially sorted lists. The results show that the error distribution is fairly even across the list of numbers. The greatest inaccuracy can be seen at the final position. This is due to the fact that the test lists again incorporated some sequences where both the seventh and eighth elements were successively incorrectly placed. This successive erroneous information has clearly been shown to confuse the listeners and would indicate that single out of place elements are more easily identified than multiple neighbouring out of place elements.

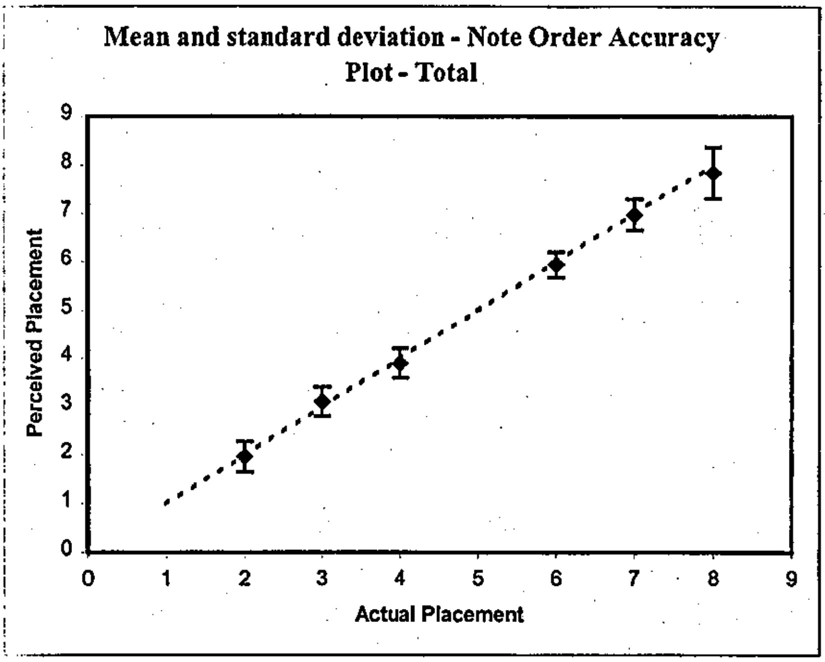


Figure 8.24 – List state note order accuracy – entire group.

Actual Placement	1	2	3	4	5	6	7	8
Perceived Placement Mean		1.966667	3.1	3.9		5.928571	6.966102	7.833333
S.D		0.319842	0.305129	0.305129		0.262265	0.319811	0.530669

Figure 8.25 – Table of list state note order accuracy – entire group.

Figure 8.26 shows how musically trained and untrained listeners performed in this series of experiments.

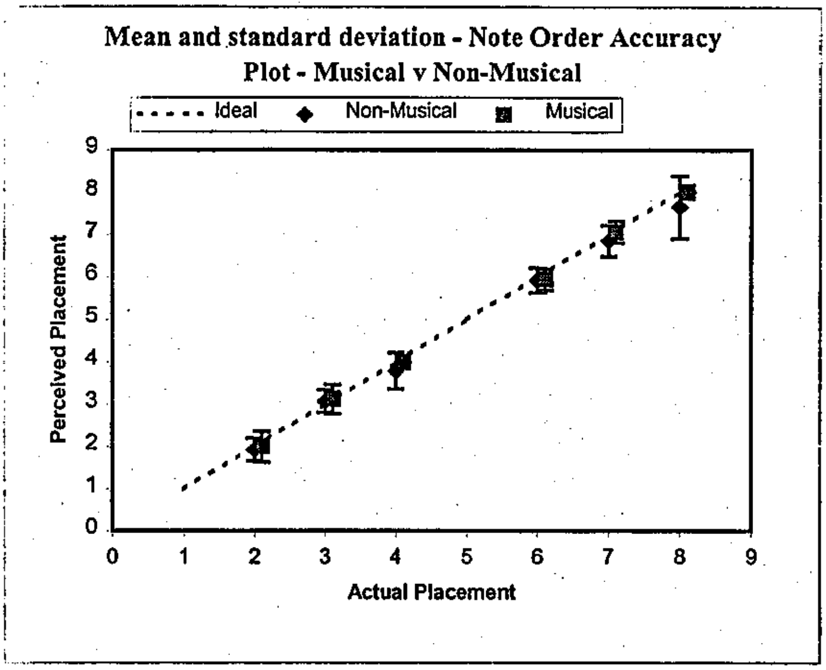


Figure 8.26 – List state note order accuracy – musical and untrained listeners.

Musical - Actual Placement	1	2	3	4	5	6	7	8
Perceived Placement Mean		2	3.125	4		5.9375	7.0625	8
S.D		0.365148	0.341565	0		0.25	0.25	0

Non-Musical - Actual Placement	1	2	3	4	5	6	7	8
Perceived Placement Mean		1.928571	3.071429	3.785714		5.916667	6.857143	7.642857
S.D		0.267261	0.267261	0.425815		0.288675	0.363137	0.744946

Figure 8.27 – Table of list state note order accuracy – musical and untrained listeners.

The data for the 'musical' test subjects given in Figure 8.27 again suggests that the musically trained group perform with greater accuracy than the results of the group as a whole.

Figure 8.28b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the scores obtained for the difference between perceived erroneous placed elements compared to the true erroneous placed elements for 'non-musical' listeners compared to 'musical' listeners. The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving spatially distributed erroneously placed elements.

H_1 : The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving spatially distributed erroneously placed elements.

	PLC	PLC	PLC	PLC	PLC7	PLC
Mann-Whitney U	106.00	106.000	88.000	94.000	78.000	88.000
Wilcoxon W	211.00	211.000	224.000	230.000	214.000	224.000
Z	-.48	-.48	-1.91	-1.267	-2.027	-1.91
Asymp. Sig. (1-tailed)	.31	.31	.02	.103	.022	.02

Figure 8.28b – Table of test statistics for each perceived placement, 'non-musical' v. 'musical'.

The null hypothesis cannot be rejected for erroneously placed elements in positions 2, 3 and 6 suggesting that there is no significant difference between 'musical' and 'non-musical' listeners when perceiving erroneously placed elements in these three positions. In contrast, the null hypothesis can be rejected at the 5% level of confidence for erroneously placed elements in positions 4, 7 and 8 suggesting that there is significant difference between 'musical' and 'non-musical' listeners when perceiving erroneously placed elements towards the end of the list. This difference in significance is again due to the increase in complexity as the positions of erroneously placed elements appear further away from the start of the list. This suggests that 'musical' listeners are more adept at perceiving locations further into the scale and successive erroneously placed elements.

The data given in the table in Figure 8.29b show the overall impact of the spatial distribution by comparing the data for this series of experiments with the data obtained for the non-3D element placement tests carried out in Chapter 5 using the Wilcoxon Signed Ranks non-parametric test. The hypotheses are:

H_0: The addition of spatial distribution has had no significant impact on the whole group's perception of element placement.

H_1: The addition of spatial distribution has significantly increased the accuracy of the whole group's perception of element placement.

	3d2 - 2d	3d3 - 2d	3d4 - 2d	3d6 - 2d	3d7 - 2d	3d8 - 2d8
Z	-1.73	-1.73	-2.23	-1.73	-2.12	-1.633
Asymp. Sig. (1-tailed)	.04	.042	.01	.04	.01	.051

Figure 8.29b – Table of test statistics for each perceived placement, non-3D v. 3D.

From this data the null hypothesis can be rejected at the 5% level for all locations except the very last location for which the null hypothesis cannot be rejected. This suggests that the addition of spatial distribution has significantly increased the entire group's perception accuracy of element placement but has not increased the accuracy when understanding and identifying successive erroneously placed elements. It was shown in the experiment documented in Section 5.6 that listeners could successfully identify out of place elements in musically represented lists of numbers. It has been shown in this experiment that the addition of spatialisation has improved listeners' identification of the out of place elements.

8.6. List manipulation experiments

8.6.1. Experiment construction

This experiment uses the same construction and stimuli as the experiment documented in Section 5.7 with exception of the addition of spatialisation. In the previous experiment, users' perception of the state of lists of numbers was tested by measuring how accurately the users identified incorrectly placed individual elements. The next step towards testing algorithm execution and state is to introduce some manipulation of the numerical data lists. The manipulation employed in this series of experiments is the swapping of incorrectly placed neighbouring elements, the same sorting mechanism as that utilised by the Bubble Sort algorithm.

In this set of experiments, thirty subjects were asked to listen to sequences of musical notes within a bounded diatonic octave scale beginning at 'Middle C'. Each test comprised two components, a checking phase as with the previous experiment followed by a sorting phase. The timbre employed for the checking phase was a flute whose position on the 'SoundWall' was again mapped directly to the shape of the tonal sequence. More specifically, its position was dependent upon the pitch for the vertical placement and the sequence order progression for the horizontal placement. The sound sources employed were those obtained by the binaural recording of the 'SoundWall' with no reverberation or chorus added. The timbre employed for the sorting phase was an acoustic grand piano whose position on the 'SoundWall' was mapped directly to the shape of the tonal sequence. Also present in the sorting phase was a trumpet to indicate the swapping action of the incorrectly placed elements. The position of this trumpet on the 'SoundWall' was dependent upon the position within the list for the horizontal and the pitch of the currently mapped element for the vertical with no chorus or reverberation added. Subjects were told that each of the eight notes within the bounded scale were mapped to the numbers one to eight. Upon listening to each test, the subjects first would hear the flute check through the list. This would be followed by the progression of the piano through the list where a swap would be denoted by a trumpet triad. All test subjects were shown and played the example shown below in Figure 8.30 that represented the swapping of two elements after a descent in pitch indicated that element 4 should be placed before element 3.

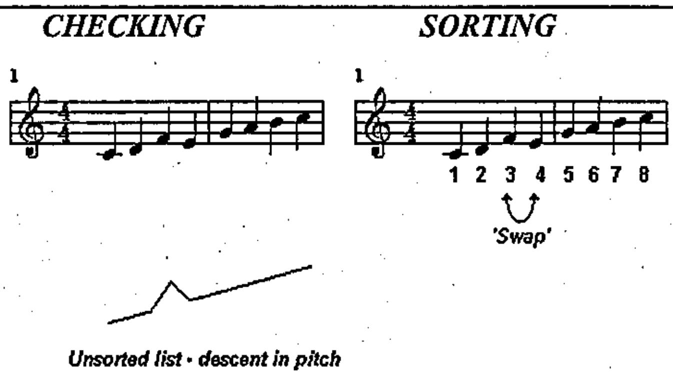


Figure 8.30 – Spatially distributed list manipulation perception example.

The workbook is given in full in Appendix G. Following the test example, listeners were played five further instances of checking and swapping where they were asked to identify which elements had been swapped.

In comparison to the previous experiment, where the only cues that denoted erroneous placement were a descent in pitch and a descent in the placement of the instrument, this experiment provided four cues. The first and second cues were the descent in pitch and instrument location during the checking phase, the third and fourth cues incorporated a descent in pitch and instrument location in the sorting phase directly followed by the trumpet triad denoting the occurrence of a swapping of elements. Test subjects were asked to identify the elements that were swapped by circling an element pair within a list. The diagrams given below represent the spatial locations of the instruments corresponding to the checking and sorting phases of the example shown above.

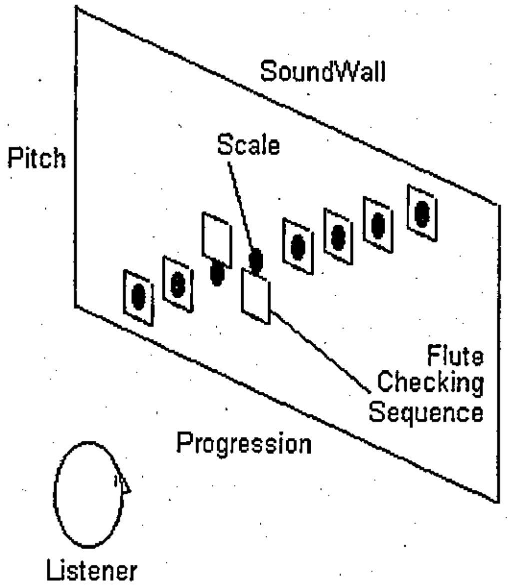


Figure 8.31 – Spatial location and movement of list checking example.

The diagram given above in Figure 8.31 shows the context scale as the solid circles placed upon the 'SoundWall' and the flute as the square symbols creating the shape of the sequence on the projected 'SoundWall'.

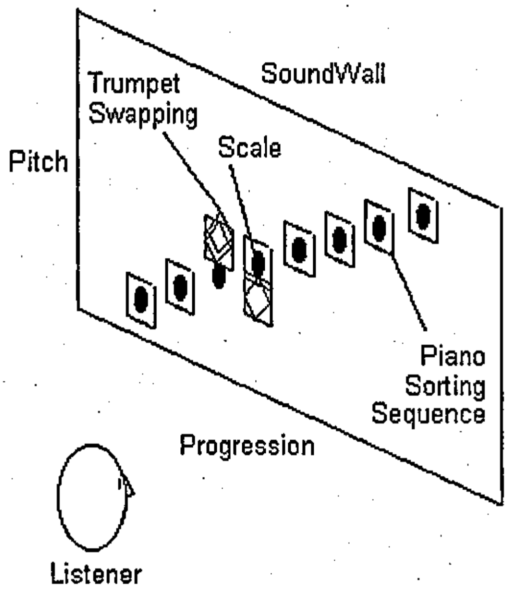


Figure 8.32 – Spatial location and movement of list element swapping example.

The diagram given above in Figure 8.32 shows the more complex representation of the sorting/swapping phase. Again the solid circles represent the context scale. This time the square symbols represent the progression of the piano through the list in the sorting phase. The diamond symbols represent the swapping action mapped to the trumpet.

It can be seen that at the third element there is a double diamond whereas at the fourth element there is a single diamond. This is due to the 'hi-lo-hi' sequence of the trumpet triad where the element with the higher value is played twice during the triad.

Another noticeable feature of the diagram is that the elements of values 3 and 4 both appear at the fourth position. This is also seen as the element of value 4 appearing at both the third and fourth locations. This replication is due to the sorting nature of the algorithm. The piano plays the list up to and including the fourth element where a descent in pitch is heard denoting an out of place element at the fourth position. The third and fourth elements are then swapped and heard by the 'hi-lo-hi' trumpet triad. The piano then continues its sorting progression through the list continuing from the fourth location containing its newly swapped element. Hence the representation plays the contents of location four twice (representing the old and new contents before and after swapping). Within the tests the swaps occurred between positions 1 and 7 with the omission of position 4 as no swap occurred in the algorithm derived examples.

8.6.2. Results and analysis

The test group used for this experiment were Group 3. Figure 8.33 and 8.34 show the users' perception of each of the swapped element pairs within the partially sorted lists. The results show that the error distribution is fairly even across the list of numbers. As with the previous experiment, multiple erroneous elements were placed (and in this case swapped) in the final portion of the list. In this case, however, there is no noticeable decrease in the users accuracy when identifying the swapping of these latter elements. This may, in part, be due to the addition of a second and more distinct cue that highlighted the swapping of the incorrectly placed elements and hence yielded a second cue as to the positional location within the scale.

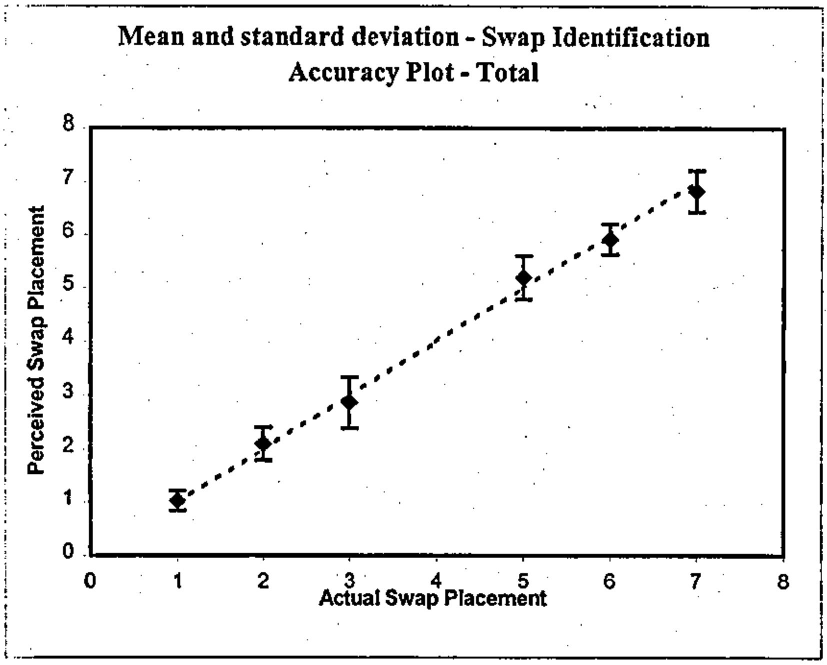


Figure 8.33 – Swapping identification accuracy – entire group.

Actual Swap Element	1	2	3	4	5	6	7
Perceived Swap Mean	1.033333	2.1	2.859649		5.2	5.916667	6.818182
S.D	0.182574	0.305129	0.479531		0.406838	0.278718	0.390154

Figure 8.34 – Table of swapping identification accuracy – entire group.

Figure 8.35 shows how musically trained and untrained listeners performed in this series of experiments.

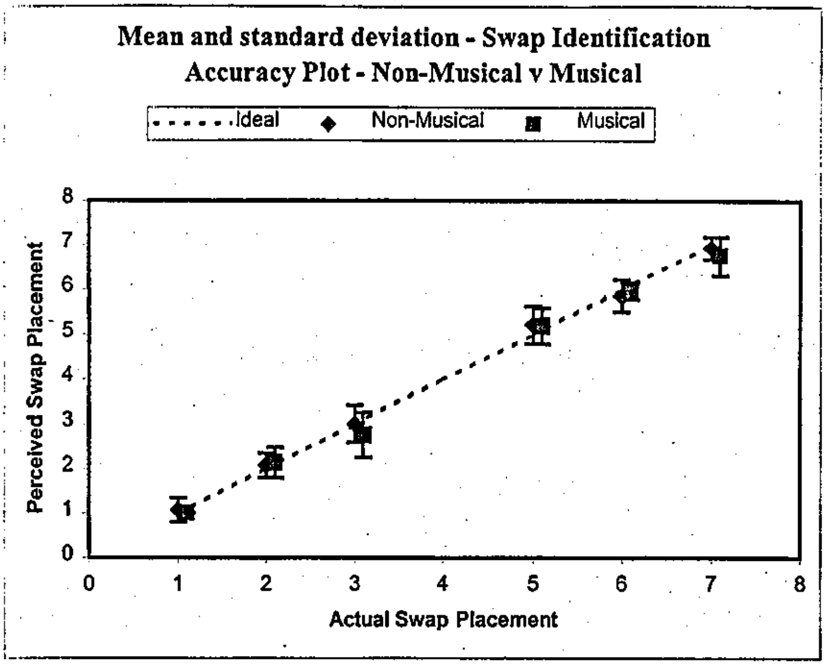


Figure 8.35 – Swapping identification accuracy – musical and non-musical listeners.

'Musical' Actual Swap Element	1	2	3	4	5	6	7
Perceived Swap Mean	1	2.125	2.75		5.1875	5.96875	6.75
S.D	0	0.341565	0.508001		0.403113	0.176777	0.440959

'Non-Musical' Actual Swap Element	1	2	3	4	5	6	7
Perceived Swap Mean	1.071429	2.071429	3		5.214286	5.857143	6.9375
S.D	0.267261	0.267261	0.408248		0.425815	0.356348	0.25

Figure 8.36 – Table of swapping identification accuracy all listeners.

The results for both 'musical' and "non-musical" subjects given in Figure 8.35 and 8.36 suggest little difference between the 'musical' group and the group as a whole.

Figure 8.37b shows the results of the Mann-Whitney (Wilcoxon independent samples) non-parametric test applied to the scores obtained for the difference between perceived erroneous placed and swapped elements compared to the true erroneous placed and swapped elements for 'musical' listeners compared to 'non-musical' listeners. The hypotheses are:

H_0 : There is no difference between the 'non-musical' and 'musical' test groups when perceiving spatially distributed erroneously placed and swapped elements.

H_1 : The 'musical' listeners perform with differing accuracy than the 'non-musical' listeners when perceiving spatially distribution erroneously placed and swapped elements.

	SWP	SWP	SWP3	SWP5	SWP6	SWP
Mann-Whitney U	104.00	106.000	106.500	109.000	87.000	88.000
Wilcoxon W	240.00	211.000	242.50	245.000	223.000	224.000
Z	-1.06	-.48	-.26	-.180	-1.609	-1.05
Asymp. Sig. (1-tailed)	.14	.13	.39	.429	.054	.14

Figure 8.37b – Table of test statistics for perceived placement/swap, 'non-musical' v. 'musical'.

It can be seen from the data given in Figure 8.37b that the null hypothesis cannot be rejected for erroneously placed and swapped elements at any position in the list suggesting that there is no significant difference between 'musical' and 'non-musical' listeners when perceiving spatially distributed erroneously placed and swapped elements. This data, in contrast to the results gathered for the non-3D version of this test documented in Chapter 5, shows that the previously observed margin between the two sub-groups for successive swaps towards the end of the list cannot be seen in this spatially distributed version. This suggests that the 'non-musical' group have benefited more from the addition of 3D placement. More specifically, the 'non-musical' group have shown increased understanding of successively swapped elements due to addition of spatial distribution. The overall accuracy for all test listeners is still observably high.

The data given in the table in Figure 8.38b show the overall impact of the spatial distribution by comparing the data for this series of experiments with the data obtained for the non-3D element swapping tests carried out in Chapter 5 using the Wilcoxon Signed Ranks non-parametric test. The hypotheses are:

H_0 : The addition of spatial distribution has had no significant impact on the whole group's perception of element swapping/placement.

H_1 : The addition of spatial distribution has significantly increased the accuracy of the whole group's perception of element swapping/placement.

	3d1 - 2d	3d2 - 2d	3d3 - 2d3	3d5 - 2d	3d6 - 2d	3d7 - 2d7
Z	-1.41	-.966	-2.00	-1.73	-1.80	-3.491
Asymp. Sig. (1-tailed)	.07	.167	.02	.04	.03	.000

Figure 8.38b – Table of test statistics for each perceived swap, non-3D v. 3D.

From this data the null hypothesis can be rejected at the 5% level for the final four locations. For the first two locations the null hypothesis cannot be rejected. This suggests that the addition of spatial distribution has significantly increased the entire group's perception accuracy of element swapping placement for the majority of locations in the list, particularly in the latter half. In the experiment documented in Section 5.7 it was shown that listeners could successfully identify erroneously placed and swapped elements within numerical lists when represented musically. It has been show in this experiment that the addition of spatial location has improved listeners' identification accuracy.

8.7. Spatially enhanced multiple algorithm auralisation

Having carried out preliminary work which indicates that spatial distribution can improve the understanding for both musical and non-musical listeners when listening to elements of auralised sorting routines, this section describes experiments using the complete algorithms and spatial distribution.

The algorithms identified previously in this thesis have now been auralised in a similar manner to those of the 2D stereophonic auralisations but have been extended into 3D auditory space using the SIMBAA 3D toolbox.

8.7.1. Spatially enhancing the Bubble Sort auralisation

In order to remain in a constant semantic framework, the musical mappings used in this auralisation are the same as were used in the experiments documented in Chapter 6 with the exception of the spatialisation of the data onto a 'SoundWall' and the spatial location of control events behind the listener. The basis of this algorithm is to repeatedly iterate through the list comparing every adjacent pair of elements and swapping them if they are not in the correct relation. When an iteration takes place without any pairs of elements being swapped then the list is known to be sorted and the algorithm can terminate.

The features identified in Chapter 4 and auralised in the spatially distributed version of the algorithm auralisation are:

1. **The current state of the list** - this auralisation was achieved by mapping element values to pitch (a metaphor). The chosen instrument here was a flute whose position is projected onto the virtual 'SoundWall'. The horizontal position of the flute upon the 'SoundWall' is mapped directly to the position within the list of the element that is currently being played. The vertical position of the flute is similarly mapped directly to the numerical value (which is also mapped to pitch) of the element that is currently being played. The sequence is played on the flute and moves across the wall from left to right and the vertical position reflects its

pitch. The contour of the moving flute positions yields the shape of the numerical list. It was expected that these extra spatial cues would allow the listener to better visualise the shape of the tonal sequence.

2. **Iteration count** – this auralisation was achieved by mapping the counter that is used to control the number of iterations to a wooden block. The sound of the wooden block is repeated as many times as the iteration count indicates. This mapping is to one of the binaurally recorded control events and is placed behind the listener.

3. **Progression of the algorithm through the list** - the chosen mapping here is a simple acoustic grand piano whose position is projected onto the virtual 'SoundWall'. The horizontal position of the piano upon the 'SoundWall' is mapped directly to the position within the list of the element that is currently being played. The vertical position of the piano is similarly mapped directly to the numerical value (which is also mapped to pitch) of the element that is currently being played. As the sequence is played, the piano moves across the wall from left to right. The contour of the moving piano positions yields the shape of the numerical list.

4. **The swapping of elements** - this is heard in parallel with the ascending acoustic grand piano. The structure is a brass ensemble playing a major triad. The first note is an element to pitch mapping of the higher value in the current pair, the second note is an element to pitch mapping of the lower note in the current pair and finally the third note is a repetition of the first note. As with the two previous mappings the position of the brass ensemble is dependent upon the positions of the two swapping elements within the list and their individual values (also mapped to pitch). The positions of the triad notes are in neighbouring positions due to the sorting nature of the Bubble Sort algorithm.

5. **Successful termination** - this auralisation is achieved by again using the brass ensemble, but this time it was used to produce a simple yet suggestive 'Ta - Da' sequence. This mapping is in the form of the other binaurally recorded control event and is placed behind the listener's head.

8.7.2. Spatially enhancing the Selection Sort auralisation

The basis of this algorithm is to repeatedly iterate through the list searching for the smallest element and then placing it in its correct location. When all target elements have been filled then the list is known to be sorted and the algorithm can terminate.

The features identified in Chapter 4 and the auralisation of these features are:

1. **The current state of the list** – same as Bubble Sort

2. **Iteration count** – same as Bubble Sort

3. **Progression of the algorithm through the list** – same as Bubble Sort

4. **The swapping of elements** – again the structure employed here is a brass ensemble playing a major triad. The values (and pitches) and positions within the list of the two elements being swapped again yield the coordinates of the position of the brass ensemble upon the 'SoundWall'. Unlike the Bubble Sort algorithm auralisation however, the positions of notes within the triad are not always neighbouring. This is due to the difference in sorting natures between the two algorithms

5. **Successful termination** – same as Bubble Sort

8.7.3. Spatially enhancing the Exchange Sort auralisation

The basis of this algorithm is to repeatedly iterate through the list comparing the current element to all subsequent elements and swapping them to place the smallest element in the current location. When all target elements have been filled then the list is known to be sorted and the algorithm can terminate.

The features identified in Chapter 4 and the auralisation of these features are:

1. **The current state of the list** – implemented in exactly the same manner as the Bubble Sort and the Selection Sort..
2. **Iteration count** – implemented in exactly the same manner as the Bubble Sort and the Selection Sort.
3. **Progression of the algorithm through the list** – implemented in exactly the same manner as the Bubble Sort and the Selection Sort.
4. **The swapping of elements** – implemented in a similar manner to the Bubble Sort and the Selection Sort. Again the positions of the notes within the triad will not always be neighbouring and can theoretically be anywhere within the list.
5. **Successful termination** – implemented in exactly the same manner as the Bubble Sort and the Selection Sort.

8.7.4. Spatially enhancing the Quick Sort auralisation

The basis of this algorithm is to divide the list into two sub-lists where the elements of the first list are all smaller than the elements of the second list, this is decided about a pivot that is chosen to be the mid point of the data in the list. Each sub-list is then recursively sorted until all of the elements have been correctly placed.

The features identified in Chapter 4 and the auralisation of these features are:

1. **The current state of the list** – implemented in exactly the same manner as the Bubble Sort, Selection Sort and Exchange sort algorithms.

2. **Iteration count** – implemented in exactly the same manner as the Bubble Sort, Selection Sort and Exchange sort algorithms.

3. **Value of the current pivot** –The chosen timbre is the trumpet whose position upon the 'SoundWall' is solely dependent upon the numerical value (pitch) of the pivot. Pivots will therefore always be heard along the line of a scale as projected onto the 'SoundWall'. The duration of this note is held for twice the period of all others to highlight it as a decision point.

4. **Playing the current element that is to be sorted based upon the current chosen pivot** – implemented in exactly the same manner as the list elements for the Bubble Sort, Selection Sort and Exchange Sort algorithms. The position is dependent upon numerical value (pitch) and current position of the element within the list.

5. **The placement of elements** – the chosen timbre here is again the acoustic grand piano whose position is dependent upon the target location. The location is given by the next empty slot within the sub-list that is to receive the current element. Placement is mapped to the position of these current sub-lists.

6. **Successful termination** – implemented in exactly the same manner as the Bubble Sort, Selection Sort and Exchange sort algorithms.

8.7.5. Spatially enhancing the Inside-Out Sort auralisation

The basis of this algorithm is to divide the list into two sub-lists where the elements of the first list are all smaller than the elements of the second list, this is decided about a pivot that is chosen to be the mid point of the data in the list. The left sub-list is then auralised and sorted with the Bubble Sort algorithm and the right sub-list is auralised and sorted using the Selection Sort algorithm.

The auralisation is based upon a combination of the Quick Sort algorithm and the Bubble and Selection Sort algorithms. The spatial distributions for this algorithm are therefore given by the spatial distributions described previously.

8.7.6. Spatially enhancing the Outside-In Sort auralisation

The basis of this algorithm is to divide the list into two sub-lists where the elements of the first list are all smaller than the elements of the second list, this is decided about a pivot that is chosen to be the mid point of the data in the list. The left sub-list is then auralised and sorted with the Selection Sort algorithm and the right sub-list is auralised and sorted using the Bubble Sort algorithm.

The auralisation is based upon a combination of the Quick Sort algorithm and the Bubble and Selection Sort algorithms. The spatial distributions for this algorithm are therefore given by the spatial distributions described previously.

8.8. Multiple algorithm auralisation information extraction

8.8.1. Experiment construction

In this series of experiments all six of the previously described algorithms were auralised and played to thirty test subjects. The auralisations were the same as used in the experiments documented in Chapter 6 with the exception of the addition of spatialisation of the information. The SIMBAA 3D system was utilised to provide the spatially distributed auralisations. The points of interest that were auralised in these implementations of the various algorithms can be summarised as follows:

1. The current state of the list.
2. The iteration count.
3. Progression of the algorithm through the list of elements.
4. The swapping or placement of elements.
5. Successful termination.

The subjects were told about the nature of each of the algorithms. The same information and played examples pertaining to the algorithms were used as described in Chapter 6. The workbook presented to all test subjects for this series of experiments is given in full in Appendix L. The questions asked of the test subjects are also the same as documented in Chapter 6.

8.8.2. Results and analysis

The test group used for this series of experiments was Group 4. This group were also used in the algorithm auralisation experiments documented in Chapter 6. The experiment was a counter balanced within groups construction to compensate for the learning effect. With this series of experiments a further preliminary test was carried out in order to understand the users' ability to draw the shapes of simple tunes. Given that some musically trained test subjects might fully understand the shape of the tonal sequences it may also be possible that they do not have the ability to draw. All subjects passed this test.

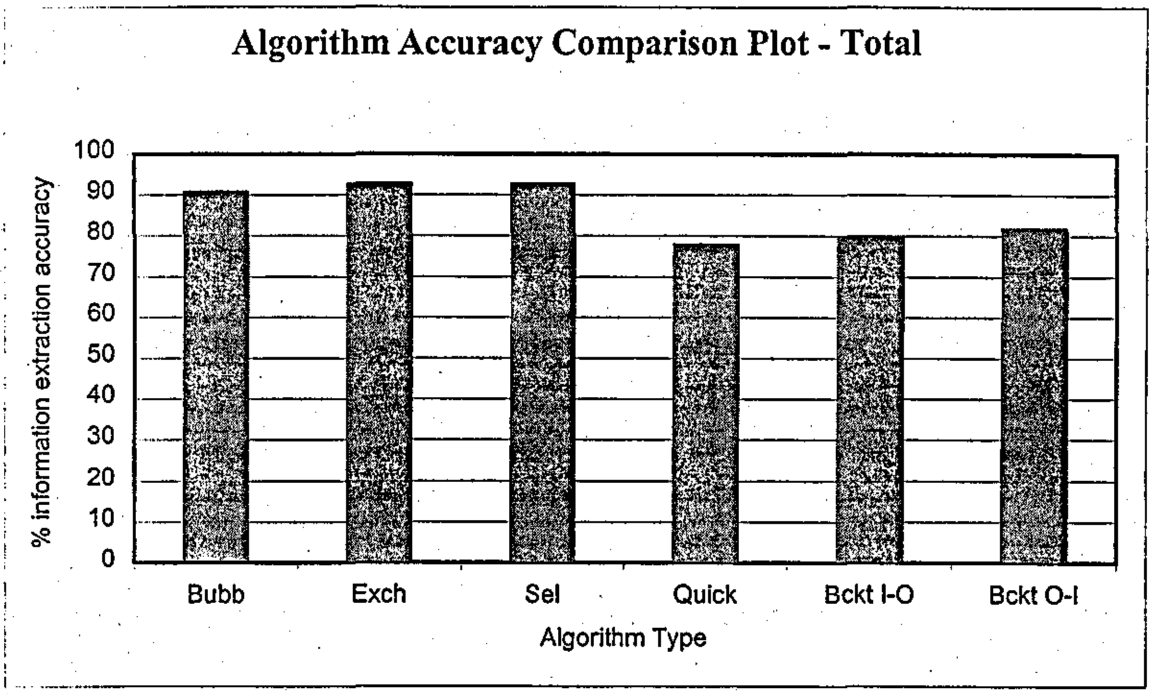


Figure 8.40 – Algorithm information extraction accuracy for each algorithm.

Figure 8.40 shows how each of the algorithm auralisations compared. The data represents the average information extraction for each of the algorithms for the entire group of test listeners. The data again suggests that the algorithms with the previously described anchor points near to the boundaries of the context scale tend to be more easily understood than the algorithms that employ the Quick Sort algorithm where the anchor points are either moving between passes or becoming larger in number. It is necessary to split this data into sub-groups defined by musical ability to investigate if musical training has any effect on understanding the information.

Figure 8.41 shows how the group of test subjects performed when answering questions on information extraction from the spatially distributed Bubble Sort auralisation. The data is displayed along the x-axis in order of musical ability.

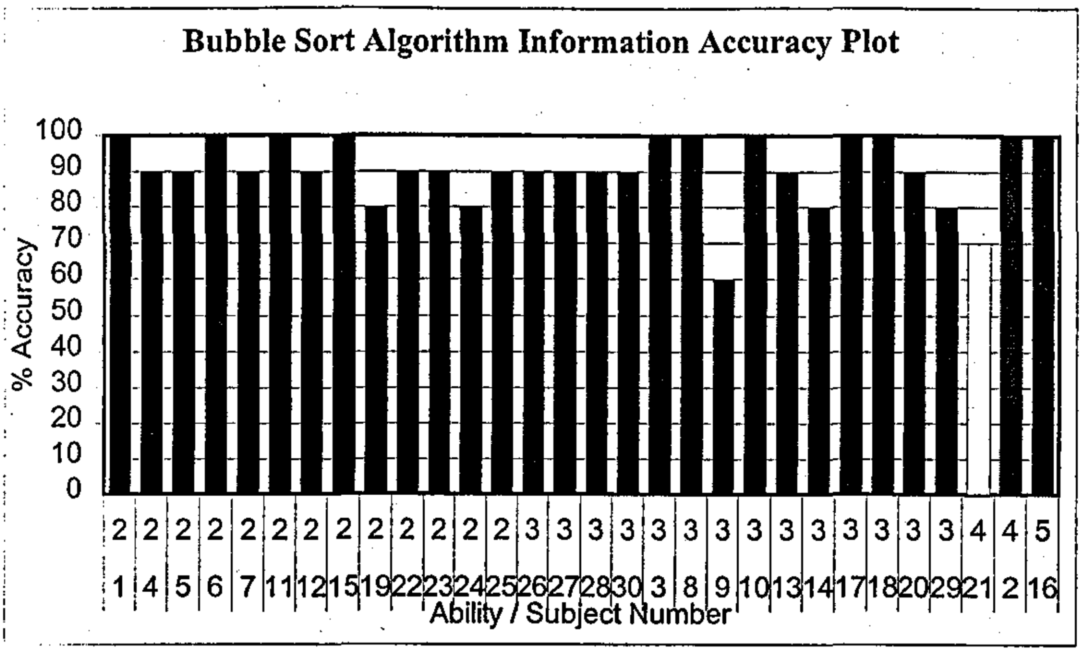


Figure 8.41 – Bubble Sort information extraction accuracy.

The data for the remaining algorithm auralistaions exhibit similar results. The graphs for these results are given in Figures M.18 to M.22 in Appendix M. The data suggests that there is little difference between 'musical' and 'non-musical' subjects and that the overall performance of the test group is generally high.

The data in the table given in Figure 8.47b show the results of the Mann-Whitney non-parametric test applied to the information extraction scores obtained from each of the six algorithm auralisations for the 'musical' listeners compared to 'non-musical' listeners. The hypotheses are:

H_0 : There is no significant difference between 'musical' and 'non-musical' listeners when extracting information from the specified spatially distributed algorithm auralisation.

H_1 : There is a significant difference between 'musical' and 'non-musical' listeners when extracting information from the specified spatially distributed algorithm auralisation.

	B	E	S	Q3	BIO3	BOI
Mann-Whitney U	109.00	94.500	108.500	104.500	98.000	104.500
Wilcoxon W	200.00	247.500	199.500	257.500	189.000	195.500
Z	-.06	-.74	-.09	-.258	-.548	-.25
Asymp. Sig. (1-tailed)	.47	.23	.46	.349	.292	.39

Figure 8.47b – Table of test statistics, alg info extraction, 'musical' v. 'non-musical'.

From the data given in the above figures, the null hypothesis cannot be rejected for each algorithm auralisation concluding that there is no significant difference between 'musical' and 'non-musical' listeners when understanding and extracting information from the each of the spatially distributed algorithm auralisations.

The graphs in Figures M.23 to M.28 given in Appendix M show the performance of the group for each of the questions on the spatially distributed Bubble Sort, Exchange Sort, Selection Sort, Quick Sort, Bucket In-Out Sort and Bucket Out-In Sort auralisations respectively. Quantitative question are shown as solid bars and qualitative questions are shown as clear bars.

The data suggest that there is some difference between quantitative and qualitative information perception. It also suggests that overall performance of the test group is generally high for each of the questions.

Figure 8.54b shows the results of the Wilcoxon Signed Ranks non-parametric test applied to the information extraction scores obtained from each of the six spatially distributed algorithm auralisations for qualitative questions compared to quantitative questions.

The hypotheses are:

H$_0$: There is no significant difference between quantitative and qualitative information perception and understanding for the specified spatially distributed algorithm auralisation.

H$_1$: There is a significant difference between quantitative and qualitative information perception and understanding for the specified spatially distributed algorithm auralisation.

	BQN3 - BQL3	EQN3 - EQL3	SQN3 - SQL3	QQN3 QQL3	BIOQN3 - BIOQL3	BOIQN3 - BOIQL3
Z	-2.07	-2.55	-2.399	-2.095	-2.44	-2.200
Asymp. Sig. (1-tailed)	.019	.00	.008	.018	.00	.014

Figure 8.54b – Table of test statistics for algorithms' info extraction, qualitative v. quantitative.

From the data given in the above figures, the null hypothesis can be rejected at the 5% level of confidence for each algorithm auralisations concluding that there is a significant difference between the perception and understanding of qualitative and quantitative information types for the each of the algorithm auralisations. Furthermore the data shows that quantitative information translates better than qualitative information.

Given that there is no significant difference between 'musical' test subjects and 'non-musical' test subjects when understanding musically auralised algorithm execution and state, does this also hold true for each of the information types?

The graphs given in Figures M.29 to M.34 in Appendix M show how the two sub-groups perform on each question for the spatially distributed Bubble Sort, Exchange Sort, Selection Sort, Quick Sort, Bucket In-Out Sort and Bucket Out-In Sort algorithms respectively.

The data given in the above figures suggests that there is little difference between 'musical' listeners and 'non-musical' listeners when understanding either quantitative information types or qualitative information types.

The data in the table given in Figure 8.61b show the results of the Mann-Whitney non-parametric test applied to the information extraction scores obtained from each of the six spatially distributed algorithm auralisations for 'musical' listeners compared to 'non-musical' listeners for **qualitative** question types. The hypotheses are:

H$_0$: There is no significant difference between 'musical' listeners and 'non-musical' listeners when understanding **qualitative** information for the specified spatially distributed algorithm auralisation.

H$_1$: There is a significant difference between 'musical' listeners and 'non-musical' listeners when understanding **qualitative** information for the specified spatially distributed algorithm auralisation.

	BQL3	EQL3	SQL3	QQL	BIOQL3	BOIQL3
Mann-Whitney U	101.000	103.500	96.500	90.00	109.000	106.000
Wilcoxon W	254.000	256.500	249.500	243.00	200.000	197.000
	-.439	-.331	-.66	-.93	-.067	-.199
Asymp. Sig. (1-tailed)	.331	.371	.26	.17	.473	.241

Figure 8.61b – Table of test statistics - algorithms' qualitative info extraction, musical v. non-musical

From the data given in the above figures, the null hypothesis cannot be rejected for each algorithm auralisations concluding that there is no significant difference between 'musical' listeners and 'non-musical' listeners for the perception and understanding of **qualitative** information types for the each of the algorithm auralisations.

Figure 8.62b shows the results of the Mann-Whitney non-parametric test applied to the information extraction scores obtained from each of the six spatially distributed algorithm auralisations for 'musical' listeners compared to 'non-musical' listeners for **quantitative** question types. The hypotheses are:

H₀ : There is no significant difference between 'musical' listeners and 'non-musical' listeners when understanding **quantitative** information for the specified spatially distributed algorithm auralisation.

H₁ : There is a significant difference between 'musical' listeners and 'non-musical' listeners when understanding **quantitative** information for the specified spatially distributed algorithm auralisation.

	BQN3	EQN3	SQN3	QQN	BIOQN3	BOIQN3
Mann-Whitney U	98.000	99.500	98.000	100.500	74.000	102.500
Wilcoxon W	189.000	252.500	189.000	191.500	165.000	255.500
Z	-.71	-.781	-.810	-.44	-1.671	-.370
Asymp. Sig. (1-tailed)	.23	.218	.209	.32	.048	.356

Figure 8.62b – Table of test statistics - algorithms' quantitative info extraction, musical v. non-musical.

From the data given in the above figures, the null hypothesis again cannot be rejected for all but one of the spatially distributed algorithm auralisations. However, the algorithm that does show a significant difference between the two groups is only just considered to be significant. This suggests that in general there is no significant difference between 'musical' listeners and 'non-musical' listeners for the perception and understanding of **quantitative** information types for the each of the algorithm auralisations. Given that no significant difference has been shown between 'musical' listeners and 'non-musical' listeners when understanding either quantitative information types or qualitative information types, it is also important to analyse the variance between the information types.

The data in the table given in Figure 8.63b show the results of the Wilcoxon signed rank non-parametric test applied to the information extraction scores obtained from each of the six spatially distributed algorithm auralisations for **'non-musical'** listeners for quantitative question types compared to qualitative question types. The hypotheses are:

H_0 : There is no significant difference between qualitative and quantitative question types for **'non-musical'** listeners when understanding information for the specified spatially distributed algorithm auralisation.

H_1 : There is a significant difference between qualitative and quantitative question types for **'non-musical'** listeners when understanding information for the specified spatially distributed algorithm auralisation.

	MBQN3 - MBQL3	MEQN3 - MEQL	MSQN3 - MSQL3	MQQN3 - MQQL3	MBIOQN3 MBIOQL3	MBOIQN3 - MBOIQL3
Z	-1.89	-1.81	-2.530	-2.021	-2.12	-1.615
Asymp. Sig. (1-tailed)	.029	.03	.006	.022	.01	.053

Figure 8.63b – Table of test statistics, algorithms' 'non-musical' info extraction, qualitative v. quantitative

From the data given in the above figures, the null hypothesis can be rejected at the 5% level of confidence for almost all algorithm auralisations concluding that there is significant difference between quantitative and qualitative question types for **'non-musical'** listeners when perceiving and understanding information for the each of the spatially distributed algorithm auralisations.

The data in the table given in Figure 8.64b show the results of the Wilcoxon signed rank non-parametric test applied to the information extraction scores obtained from each of the six spatially distributed algorithm auralisations for **'musical'** listeners for quantitative question types compared to qualitative question types.

The hypotheses are:

H_0 : There is no significant difference between qualitative and quantitative question types for '**musical**' listeners when understanding information for the specified spatially distributed algorithm auralisation.

H_1 : There is a significant difference between qualitative and quantitative question types for '**musical**' listeners when understanding information for the specified spatially distributed algorithm auralisation.

	NBQN3 - NBQL3	NEQN3 - NEQL	NSQN3 - NSQL3	NQQN3 NQQL	NBIOQN3 NBIOQL3	NBOIQN3 - NBOIQL3
Z	-.905	-1.89	-.905	-.811	-1.15	-1.461
Asymp. Sig. (1-tailed)	.183	.03	.183	.209	.12	.072

Figure 8.64b – Table of test statistics, algorithms' 'musical' info extraction, qualitative v. quantitative

From the data given in the above figures the null hypothesis cannot be rejected for all but one of the spatially distributed algorithm auralisations suggesting that there is no significant difference between qualitative and quantitative question types for '**musical**' test listeners when perceiving and understanding information for the each of the algorithm auralisations.

This data suggests that although no significant difference has been shown between 'musical' and 'non-musical' listeners when perceiving and understanding qualitative and quantitative information types, the spread in accuracy between the information types is greater for 'musical' listeners than for 'non-musical' listeners.

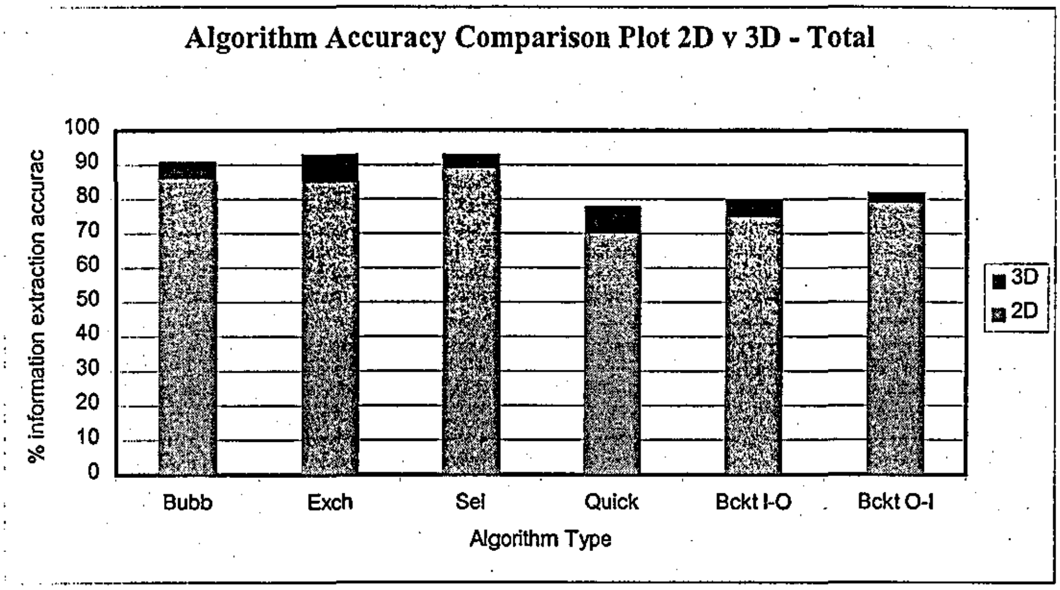


Figure 8.65a – Information extraction accuracy for each algorithm (2D v 3D).

Figure 8.65a shows the impact of the addition of spatialisation for each of the algorithm auralisations. These data might suggest that the addition of spatialisation has had a beneficial effect. Figure 8.65b shows the overall impact of the spatial distribution by comparing the data for this series of experiments with the data obtained for the non-3D algorithm auralisation tests carried out in Chapter 5 using the Wilcoxon Signed Ranks non-parametric test. The hypotheses are:

H_0 : The addition of spatial distribution has had no significant impact on the whole group's identification and understanding of musically auralised algorithm state and execution.

H_1 : The addition of spatial distribution has significantly increased the accuracy of the whole group's identification and understanding of musically auralised algorithm state and execution.

	B3 - B2	E3 - E2	S3 - S2	Q3 - Q	BIO3 - BIO2	BOI3 - BOI2
	-2.667	-3.318	-2.261	-2.79	-2.287	-1.727
Asymp. Sig. (1-tailed)	.004	.001	.012	.003	.011	.042

Figure 8.65b – Table of test statistics for each algorithm auralisation, non-3D v. 3D.

From this data the null hypothesis can be rejected at the 5% level for the Bucket Sort Out-In auralisation and at the 1% level for the remaining algorithm auralisations. This strongly suggests that the addition of spatial distribution has significantly increased the entire group's identification and understanding of musically auralised algorithm state and execution

8.9. Conclusion

For the pitch test experiments, the results have shown that there is a significant difference between the 'musical' and 'non-musical' group when perceiving tones that are close to the boundaries of the context scale supporting the findings in Chapter 5 for the non-3D pitch tests. The data obtained for the pitch interval test experiments showed that for small intervals (less than 2) and large intervals (greater than 6) there is no significant difference between 'musical' and 'non-musical' listeners. This difference becomes significant as the interval size moves furthest away from the extremes. The addition of spatial location in this context has narrowed the difference between the 'musical' and 'non-musical' groups for large intervals. The addition of spatial distribution has also shown an observable increase in the perception and accuracy of absolute pitch and pitch intervals. Thus overall, spatial distribution does act as an additional cue and listeners can use it particularly if they are not musically trained.

For the shape perception experiments using short musical sequences with musical timing the results showed significant difference between the 'musical' and 'non-musical' listeners for all shapes, supporting the findings reported in Chapter 5 for the non-3D version of this test. The addition of spatial distribution has also shown an observable increase in the perception and accuracy of the shapes of short tonal sequences with no timing. In other words the performance of both groups has improved significantly as a result of the addition of spatial distribution, but the improvement does not favour one group over the other. The only shape which did not show improvement was a very variable one.

The data obtained for the series of experiments concerned with identifying out of place elements in an otherwise ascending list of numeric elements showed there is no significant difference between the two groups of 'musical' and 'non-musical' listeners for elements identified in approximately the first half of the list. For the remainder of the list, the difference between the two groups becomes significant due to the increasing complexity as the positions of erroneously placed elements move further up the context scale. This again supports the findings documented in Chapter 5 for the non-3D version of this test. However, the addition of spatial location has also shown a significant observable increase in the perception and accuracy of out of place elements.

Similar results were observed for the identification of erroneously placed and swapped elements in an otherwise ascending list of numerical elements. For approximately the first half of the list no significant difference was observed between the 'musical' and 'non-musical' test groups. The significant difference between the two groups is observed where successive multiple erroneously placed and swapped elements occur in the latter half of the list. Again the addition of spatial location has also shown an observable increase in the perception and accuracy of the location of swapped elements.

The results for the experimentation using the six different spatially distributed algorithms showed that once again the algorithms with fixed and constant anchor points are more readily understood (Bubble Sort, Exchange Sort and Selection Sort algorithms). The data also showed that overall there is no significant difference between 'musical' listeners and 'non-musical' listeners when perceiving and understanding musically represented information about the state and execution for each of the algorithms.

However quantitative information types are significantly more easily understood and identified than qualitative information types for both groups. Though for both information types there is no significant difference between 'musical' and 'non-musical' listeners. The results again showed that for the 'non-musical' sub-group alone quantitative information types were more easily understood than qualitative information types. In contrast, no significant difference was observed between the two information types for the 'musical' sub-group alone. This again suggested that the spread between information types for each sub-group was different but not different enough to be significant when comparing the two sub-groups. These findings generally support the findings documented in Chapter 6 for the non-3D version of this series of experiments. Again the addition of spatial distribution has also shown an observable increase in the perception and accuracy of information about each algorithm's state and execution. In Chapter 6 it was shown how the identification of errors in algorithm auralisations was difficult. It has not been subject to spatialisation in this chapter because the SIMBAA tool is not aimed at being used to aid bug location in sorting algorithms. Furthermore, the accuracy of the identification of bugs in musically auralised sorting algorithms is not an area of concern within this thesis.

In general it has been shown that musical training does have some affect on the perception of musical sequences and pitch but the effect is not a strong one. However, the results have shown that both musically trained and untrained listeners are quite capable of discerning pitch and understanding shape and musically represented numerical data with a promising degree of accuracy. The algorithm auralisation experiments have shown that no significant difference exists between the 'musical' and 'non-musical' groups with or without spatial distribution. Throughout each of the experiments the addition of spatial location cues has been proven to increase listeners' perception, identification and understanding. Spatial distribution of the type used here, is therefore really useful and could enhance aural based presentations.

Chapter 9
Discussion and conclusions

9.1. Introduction

This final chapter summarises the research described in this thesis and the results obtained. It also analyses these results and draws conclusions about the usefulness of the auditory approach taken. It then discusses the limitations of the work and how these might be overcome. It assesses the contribution of the thesis to the field of musical auditory display and suggests areas for future investigation of the use of music and spatial enhancement in algorithm auralisation.

9.2. The results obtained

The listening tests (Chapter 5) and the algorithm auralisation tests (chapters 6 and 8) form an extension of earlier work carried out by Alty [3], Vickers [181], Rigas [160], and Brown and Hershberger [45] etc. The difference is in the evaluation of the listening tests and algorithm auralisations. All the results were analysed for statistical significance.

9.2.1. The effect of musical training

The listening tests (described in Chapters 5 and 8) assessed how humans perceive pitch, shape, list state and list manipulation.

The results obtained for the pitch tests showed that there was a significant difference between the 'musical' and 'non-musical' group when perceiving tones that appear close to the boundaries of the context scale. This data further showed that there is no significant difference between the groups when perceiving tones that fall into the area of greatest ambiguity in the middle of the context scale. The experiment further showed that for small intervals (less than 2) there was no significant difference between 'musical' and 'non-musical' listeners. This difference became significant for interval sizes greater than

1 and increased in relation to the increase in interval size. These results suggested that musical training could be beneficial when estimating large pitch intervals or pitches that are close to the limits of their context scales. For small intervals and the majority of pitches (which are not close to the boundaries of the context scale) there was no evidence to support the notion of musical training being beneficial to pitch perception. For the majority of cases little or no noticeable difference between the two groups was observed suggesting that musically trained and untrained listeners are both viable groups as target users for musical auditory interfaces.

For the shape perception experiments using short musical sequences with musical timing the results showed a significant difference between the 'musical' and 'non-musical' listeners for all shapes. In contrast, the series of experiments using short tonal sequences with no musical timing showed that there was no significant difference between the two groups when perceiving the shapes. This difference in significance suggests that when tonal sequences have musical timing applied to them, making them more '*musical*', the 'musical ' group of test subjects tend to perform with greater accuracy than the 'non-musical' group. When this musical timing was removed, as was employed for algorithm state auralisation within this thesis, the data showed that there was no significant difference between the two groups when perceiving tonal shapes (or musically auralised algorithm list states). Musically trained listeners were therefore able to use the properties of the timing applied to the musical shapes. This exploitation of timing highlights the importance of maintaining the musicality of the auralisations, since both musical and non-musical listeners are familiar with scales and rhythm. Although these results suggest that musicians can benefit more from these features, they also suggest that the musical context should be maintained for both musically trained and untrained listeners. Making the auralisations as musical as possible can be achieved by paying attention to the properties of such features of music as scales and rhythm and using realistic timbres. Following these musical guidelines allows us to exploit listeners' experience of musical presentations making the auralisations more effective.

The previously described results suggested that musical training has no beneficial effect when understanding algorithm-derived shapes with no musical timing. The results from experiments investigating the perception and understanding of algorithm derived list shape progression showed that 'musical' listeners performed significantly better than 'non-musical' listeners. This further suggests that musically trained listeners are more adept at understanding the progressive evolution of a musical shape.

The results obtained for listeners when identifying out of place elements in an otherwise ascending list of numeric elements showed that musical training was not significantly beneficial when identifying out of place elements in approximately the first half of the list. For the remainder of the list, the difference between the two groups became significant due to the increasing complexity as the positions of erroneously placed elements move further up the context scale. These results suggest that musical training is only beneficial when identifying elements further in the list.

Similar results were observed for the identification of erroneously placed and swapped elements in an otherwise ascending list of numerical elements. For the majority of positions (all except the last) no significant difference was observed between the 'musical' and 'non-musical' test groups. The only significant difference between the two groups was observed where successive multiple erroneously placed and swapped elements occurred in the final position. This data suggests that multiple successive swaps increase misunderstanding of swap occurrence and location. In comparison to the results obtained in the previously described out of place elements experiments, the same results suggest that the addition of the extra cue (the sound of the elements swapping) aids localisation and reduces the observable difference between 'musical' and 'non-musical' listeners.

The results obtained for the multiple algorithm auralisation information extraction tests showed that overall there was no significant difference between 'musical' listeners and 'non-musical' listeners when perceiving and understanding musically represented information pertaining to state and execution for each of the algorithms. Within each of the information types presented (qualitative and quantitative) it was further shown that there was no significant difference between 'musical' and 'non-musical' listeners. For the 'non-musical' sub-group alone, quantitative information types were more easily understood than qualitative information types. In contrast, no significant difference was observed between the two information types for the 'musical' sub-group alone. This suggested that the spread between information types for each sub-group was different but not different enough to be significant when comparing the two sub-groups. This further suggests that musically trained listeners are generally a more reliable group when identifying either information type.

In general it has been shown that musical training does have some effect on the perception of musical sequences and pitch but the effect is not large. However, the results have shown that both musically trained and untrained listeners are quite capable of discerning pitch and understanding shape and musically represented numerical data with an acceptable degree of accuracy. It has also been shown that the difference between the groups depends upon the complexity of the musical structure and that musical training has no significant effect when understanding algorithm auralisations. These results are encouraging as one of the principal motivations behind this research was to demonstrate that music could be used as a communication medium regardless of musical skill.

9.2.2. Algorithm understanding

In the algorithm information extraction experiments (Chapters 6 and 8) subjects were asked to identify qualitative and quantitative features about the state and execution progress of six different algorithms. The auralisations were designed using the same criteria as for the listening tests. The algorithm information extraction experiments aimed to test the general hypothesis: *The musical program auralisations generated by SIMBAA 3D can successfully convey information about algorithm state and execution.*

From this general statement three specific hypotheses were identified:

1. Users can successfully understand musically auralised algorithm state and execution.
2. Certain information types are more readily identified and understood than others.
3. Certain algorithms convey information more readily than others.

To reiterate the findings concerned with the effect of musical training on understanding auralised algorithm state and execution it was found that no significant difference was present. Figure 9.1 shows that listeners are capable of understanding and identifying algorithm information with between 78% and 93% accuracy. These accuracy scores are high enough to suggest that music has been highly successful in conveying algorithm state and execution. It can also be seen that the listeners more readily understand certain algorithm types than they do others.

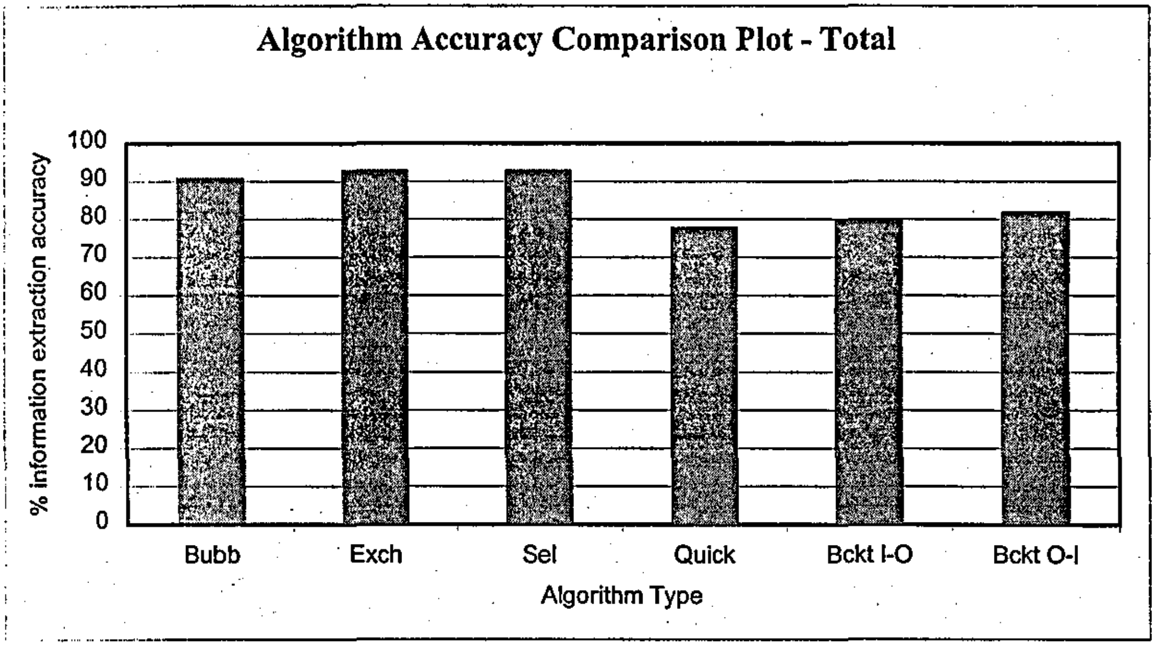


Figure 9.1 – Information extraction accuracy for each algorithm (3D auralisation).

The experiments showed that quantitative information types were significantly more easily understood and identified than qualitative information types for both groups of musically trained and untrained listeners. Though for both information types no significant difference between 'musical' and 'non-musical' listeners was observed, the results showed that for the 'non-musical' sub-group alone quantitative information types were more easily understood than qualitative information types. In contrast, no significant difference was observed between the two information types for the 'musical' sub-group alone. This suggested that the spread between information types for each sub-group was different but not different enough to be significant when comparing the two sub-groups. In general these results suggest that musically trained listeners are more reliable when understanding both information types and that quantitative information types are more readily understood through musical auralisation than qualitative information types.

It has been shown that listeners can successfully identify and understand musically auralised algorithm information with a high degree of accuracy. Before the experimental auralisations were performed, the six sorting algorithms were chosen based upon their relative sorting characteristics and natures. Each was chosen to create as diverse a selection of musical sorting algorithm auralisations as possible.

Figure 9.1 shows that certain algorithms are more readily understood than others. The results showed that the algorithms with fixed and constant anchor points were more readily understood (Bubble Sort, Exchange Sort and Selection Sort algorithms) than the other more complex algorithms (Quick Sort, Bucket Sort In-Out and Bucket Sort Out-In algorithms). The data also showed that overall there is no significant difference between 'musical' listeners and 'non-musical' listeners when perceiving and understanding musically represented information about the state and execution for each of the algorithms.

9.2.3. The usefulness of spatial enhancement

The addition of spatial distribution had shown an observable increase in the perception and accuracy of pitch and pitch intervals. This suggests that spatial distribution does act as an additional cue and listeners can use it effectively. The addition of spatial distribution had also shown an observable increase in the perception and accuracy of the shapes of short tonal sequences with no timing. In other words the performance of both groups had improved significantly as a result of the addition of spatial distribution, but the improvement does not favour one group over the other. The only shape that did not show improvement was a very variable one. The addition of spatial location had also shown a significant observable increase in the perception and accuracy of out of place elements and swapped out of place elements. In the final experimentation, the addition of spatial distribution had also shown an observable increase in the perception and accuracy of information about each algorithm's state and execution ranging from a 3% to 8% accuracy increase (Figure 8.65a in Chapter 9). Throughout each of the experiments the addition of spatial location cues has been proven to increase listeners' perception, identification and understanding. Spatial distribution of the type used here, is therefore really useful and could enhance aural based presentations.

9.3. Limitations of research

One of the limitations in this thesis is lack of investigation into the use of musical auralisation to aid bug location in sorting algorithms. A cursory study was undertaken and documented in Chapter 6 but this was by no means exhaustive as it was beyond the scope of this thesis. It might also have been useful to investigate whether the addition of spatialisation aids bug location. This thesis set out to determine how useful music might be when used to convey information about the state and execution of sorting algorithms. It was also concerned with the effects of musical training and which types of information translated best. It was not concerned with determining how useful music might be for assisting algorithm designers in bug location.

The main limitation of this research has been concerned with the Binaural Recording technique, more specifically, the problem of the geometrical differences that are present from human to human. There have been several suggestions to circumvent this issue ranging from measuring HRTFs based upon the statistical norm to measuring unique sets of HRTFs for each potential listener. In using the HRTFs based upon the geometrical measurements of the statistical norm the resulting audio experience will only translate effectively to a small percentage of the population. Creating several HRTF sets based upon geometrically categorised groups resolves this issue a little further but is still far from producing the perfect solution.

Manikins exist that are modelled on the statistical norm. If the manikin and the listener have heads with the same size and shape, the same ITD and IID information will be present; similarly, if the manikin and the listener have pinnae with the same sizes and shapes, the same elevation cues will be present. If, however, the geometrical differences between that of the listener and the manikin are significant, the resulting perceptual 3D sound environment becomes augmented and localisation is difficult.

The obvious way to reproduce a more precise individual listening experience is for the listener to also be the manikin, this way the geometrical similarities of the recording head and the listening head are as close as physically possible. This does, of course, mean that each individual listener must have his/her own unique set of binaural recordings to maximise the desired effect.

In the binaural recordings used in this research work the manikin was actually the head of a real human being selected at random and having no obvious prominent features that differentiated him from the norm. The desired spatially distributed auralisations were not designed to be as 'truly' 3D as possible but more cost-effective and aimed at giving a general idea as to the usefulness of the technique. The obvious limitation here is the geometrical differences between the live manikin and any potential listeners. More specifically, the limitation is the reduced effect such differences might have on the perception of spatially distributed musically represented information.

Another limitation of the experimentation carried out in this research is concerned with the perceived distance between the listener and the 'SoundWall'. The SoundWall was initially tested at 6 feet but it was found that the separation between positions was too narrow and this caused ambiguity when locating positions on the 'SoundWall'. The 'SoundWall' was finally constructed at a distance of 3 feet from the listener. This distance was obtained through a process of trial and error and subjective judgement. The immediate limitation here is that no optimum distance was calculated. More accurate spatial location identification might have been achieved if more attention had been paid to the relative physical positioning of the 'SoundWall' to the listener.

9.4. Future work

It has been shown that music (with or without spatial enhancement) can communicate information about algorithm execution and state to a wide range of subjects with differing musical abilities. The SIMBAA 3D system was intended as a tool used to facilitate the spatially enhanced musical auralisation of several algorithms. Ultimately it might be desirable to construct an algorithm auralisation and visualisation environment in which the user had full control over the application of visualisation and auralisation techniques. Alty and Rigas [4] described this as an '*equal opportunities interface*'. This is specifically an interface that makes no prior judgement about the capabilities of the user population regarding the use of different input/output modalities. Such an interface would offer "*...a variety of communication media, from which the user can select an appropriate mix to match their capabilities and limitations*" [4]. Because this research was investigating the role of music as a communication medium and because there is little prior research into this field, it was beyond the scope of this thesis to investigate the multi-modal representation of algorithm state and execution. Further work could investigate ways in which music, other non-speech audio, and visual display techniques could be combined.

A complementary modality has been provided that goes some way to fulfilling the aim of an equal opportunities interface [4]. It is not proposed that auralisation necessarily be used exclusively, but that it offers an additional tool. The results of this research could allow musical frameworks to be developed for general interface tasks and so future projects could use these frameworks to assist in the construction of a true equal opportunities interface as envisaged by Alty and Rigas [4].

A feature not investigated in the research described in this thesis was the rate at which the musical representations were played. SIMBAA 3D was designed to execute at variable rates of tempo in order to provide different levels of abstraction. In the experiments, each musical output was played at the same tempo. Further work should be undertaken to assess the effect of different tempos on the auralisations. This is important, as some algorithm auralisations may take a long time to complete and might become tedious and tiresome for the listener. It is also possible that different speeds of presentation will create different levels of abstraction for the listener.

Further work could be carried out as an extended study of SIMBAA 3D on an extended set of subjects. Such a study could involve training the subjects in the use of the SIMBAA 3D system alongside basic instruction of algorithms. Assuming relevant control samples were used, such a study would be able to show what effects training and familiarity with the technique have on subjects' ability to identify and understand algorithm state and execution. Similarly, studies could be carried out to determine the benefit of extensively training the listeners on the 'SoundWall' before hearing the musical algorithm auralisations.

It has been shown that listeners can successfully identify and understand musically auralised algorithm information with a high degree of accuracy. The results from this research suggest that certain classes of algorithm are more amenable to auralisation than others (Chapters 6 and 8). For example, the algorithms with fixed and constant anchor points (Bubble Sort, Exchange Sort and Selection Sort algorithms) were more readily understood than the other more complex algorithms (Quick Sort, Bucket Sort In-Out and Bucket Sort Out-In algorithms). Before the experimental auralisations were performed, the six sorting algorithms were chosen based upon the properties of their sorting characteristics and natures. Each was chosen to create as diverse a selection of musical sorting algorithm auralisations as possible. Extended studies could be undertaken to test whether algorithms of a higher complexity or algorithms that produce output that contains little information benefit from auralisation. Algorithms of a highly complex nature might benefit from auralisation more than less complex sorting algorithms. This complex information might be further disambiguated and more easily understood with the aid of auralisation techniques. In contrast, less complex algorithms might not benefit as much from auralisation as little information is available for musically auralised presentation. Similarly, information from sources other than sorting algorithms might show how information types and structures of differing complexity benefit from the auralisation technique.

Although this research did not address the needs of blind and visually impaired users, the results suggest that a musical auralisation system could be of use to the visually impaired. Given that existing sighted users have shown that auralisations can communicate algorithm information, it is not unreasonable that the system could be adapted and extended for use by the visually impaired. Visually impaired users have limited spatial perception. The use of 3D audio in the algorithm auralisations within this thesis has added a spatial element to a typically temporal medium. Such a technique may be adapted to provide spatial visualisation/location for non-sighted users.

The 3D audio technology used for the spatial enhancement of the experimental musical algorithm auralisations in this work was simple and inexpensive. A study into the acceptability of other 3D audio technologies might yield some guidelines for the cost effective spatial enhancement of similar auralisation systems. Of particular interest would be the emerging field of modelled head related transfer functions (HRTFs). Such modelling techniques are aimed at producing flexible parameter driven 3D audio syntheses that can be tailored to each user's specifications to produce the most effective and realistic spatial audio presentations. The binaural recording technique used in the experimentation described in this thesis has been subject to the limitations explained in section 9.4. These limitations could be overcome by the use of more precise tailored 3D audio presentations. The use of surround sound systems could also be investigated as a means of providing 3D program / algorithm auralisations in open-field presentations. Many spatial audio techniques exist offering differing degrees of realism. The 'reality' of these 3D audio techniques should be tested against the accuracy increases in information understanding to identify which approaches are most effective, most ineffective and most cost-effective.

The spatial enhancement technique used in this research could be applied to other existing auditory systems such as Vickers's CAITLIN [181] and Rigas's AudioGraph [4, 159, 160]. The addition of spatial location to CAITLIN could further disambiguate the musical representation of program execution. Elevation cues could be used to help visualise the top-down procedural nature of the program execution and the cyclic movement in iterations. Azimuth cues could also be used to visualise the decision making process of conditional statements. Similarly, AudioGraph might benefit from the addition of spatial location. Visually impaired users have limited spatial perception. The use of 3D audio in the algorithm auralisations within this thesis has added a spatial element to a typically temporal medium. This 3D enhancement could provide visually impaired listeners with spatial visualisation/location of graphical objects within AudioGraph. 3D spatial audio could also be applied to other works like Blattner's research using Earcons in turbulent fluid flow [20]. Again this enhancement could provide a spatial element to the presentations helping listeners to better visualise and understand the information.

9.5. Conclusions and contribution of this thesis

Prior to this research there was little evidence to support or discount algorithm auralisation as a useful tool. Previous auralisation systems had been published without empirical evidence to prove their efficacy. Brown and Hershberger [45] performed simple algorithm auralisations for supporting visualisation. Alty [3] performed some early research that documented the usefulness of simple autonomous algorithm auralisations. However, no formal or empirical evaluation of the usefulness of several different types of algorithm auralisations had been performed. In particular, no attempt had been made to employ spatial audio technologies to assist in the musical conveyance of such information. This research offers the first attempt and empirical evaluation of spatially enhanced algorithm auralisation. Some recommendations for the incorporation of spatially enhanced musical sounds have been suggested but without attention to music-cognitive and music-theoretic principles.

The development of spatially enhanced musical algorithm auralisations is a measured contribution to the field. Prior to this research, any form of algorithm auralisation had not been evaluated. Following this work, further and more detailed investigations of how far the technique can be taken could be undertaken. The major strength of this work over prior algorithm auralisation is that it is based on empirical results.

This thesis set out to address the questions of whether music can be used as a communication medium to convey information about algorithm state and execution and whether spatial enhancement could be of benefit to listeners in this context. The experiments have shown that the musical mappings can be comprehended and that, within certain constraints, they can be used to help understand algorithm progression. No musical experience is necessary, indeed, those with musical training generally performed no better on the experimental musical algorithm auralisation tasks. The experiments also showed that the addition of spatial enhancement was generally significantly beneficial in all auralisations.

In summary, it has been shown that music can convey information about algorithm state and execution. Secondly, it has been suggested that it can play a complementary role in the process of algorithm visualisation. Finally, it has been shown that the addition of spatial location cues can aid understanding and help further disambiguate information about algorithm state and execution.

A. References.

1. Aho A., Hopcropft J., and Ullman J. Data Structures and Algorithms. Addison-Wesley, 1983.
2. Albers, M. C. The Varèse System, Hybrid Auditory Interfaces, and Satellite-Ground Control: Using Auditory Icons and Sonification in a Complex, Supervisory Control System. http://www.isye.gatech.edu/chmsr/.../papers/ICAD94/ICAD_94_paper.html. 1994.
3. Alty, James. L. Can We Use Music in Computer-Human Communication? *People and Computers X:* Proc HCI'95, Cambridge Academic Press, 1995, 409-424.
4. Alty, J. L. and Rigas, D. I., Communicating Graphical Information to Blind Users Using Music: The Role of Context, in *Proc. CHI98 Conference on Human Factors in Computing Systems* (Los Angeles, CA, April 18-23, 1998), ACM Press, 574-581.
5. Alty, J. L., Vickers, P. and Rigas, D., Using Music as a Communication Medium, in *Proc. Refereed Demonstrations, CHI97 Conference on Human Factors in Computing Systems*, ACM Press, 1997, 30-31.
6. Anderton C. MIDI for Musicians. Amsco Publications, New York, 1986.
7. Apple Computer. Demystifying Multimedia: A Guide for Developers from Apple Computer, Inc. San Francisco, Vivid, 1993.
8. Arons, B., A Review of the Cocktail Party Effect, *Journal of the AmericanVoice I/O Society*, 1992, 35-50.
9. Aureal Semiconductor Inc., 4245 Technology Drive, Fremont, CA 94538, (510) 252-4245 http://www.aureal.com
10. Baecker, R. M. and Buchanan, J., A Programmer's Interface: A Visually Enhanced and Animated Programming Environment, in *Proc. Twenty-Third Annual Hawaii International Conference on Systems Sciences*, 1990, 531-540.
11. Barfield, W., Rosenberg, C. & Levasseur, G. The use of icons, earcons and commands in the design of an online hierarchical menu. *IEEE Transactions on Professional Communication,* 34(2), 1991, 101-108.

12. Balzano G. J. and Liesch B. W. The role of chroma and scalestep in the recognition of musical intervals in and out of context. Psychomusicology, 1982, 3-31.
13. Bartlett J. C. and Dowling W. J. Recognition of transposed melodies: A key distance effect in developmental perspective. Experimental Psychology, Human Perception and Performance, 1980, 501-515.
14. Batteau, D. W., The Role of the Pinna in Human Localization, Proc. Royal Society London, Vol. 168 (series B), 1967, 158-180.
15. Begault, D. & Wenzel, E.M. Techniques and Applications for Binaural Sound Manipulation in Man-Machine Interfaces, NASA-Ames Technical memorandum TM 102279. 1990.
16. Bernstein L. The Unanswered Question. Harvard University Press, Cambridge, Massachussetts, 1976.
17. Berglund, B., Harder, K. & Preis, A. Annoyance Perception of Sound and Information Extraction. *Journal of the Acoustic Society of America,* March 1994, 1501-1509.
18. Blattner, M. M., Sumikawa, D. A. and Greenberg, R. M. Earcons and Icons: Their Structure and Common Design Principles. *Human-Computer Interaction,* 1989, 11-44.
19. Blattner, M., Papp, A. L. & Gilnert, E. P. Sonic Enhancement of Two-Dimensional Graphic Arrays. *Auditory Display: Sonification, Audification and Auditory Interfaces.* Reading, MA: Addison-Wesley Publishing Company, 1994, 447-470.
20. Blattner, M. M., Greenberg, R. M. and Kamegai, M., Listening to Turbulence: An Example of Scientific Audiolization, in *Multimedia Interface Design*, M. M. Blattner and R. M. Dannenberg, Eds. Wokingham: ACM Press, 1992, 87-102
21. Blauert, J. Spatial Hearing, The MIT Press, Cambridge, MA. 1983.
22. Bly, S. A., Presenting Information in Sound, in *Proc. ACM CHI'89 Conference on Human Factors in Computing Systems*, *The Proceedings of ACM-SIGCHI* New York: ACM Press/Addison-Wesley, 1982, 371-375.

23. Boardman, D. B., Greene, G., Khandelwal, V. and Mathur, A. P., LISTEN: A Tool to Investigate the Use of Sound for the Analysis of Program Behaviour, in *Proc. 19th International Computer Software and Applications Conference,* 1995 47-65.
24. Boardman, D. B. and Mathur, A. P., *Preliminary Report on Design Rationale, Syntax, and Semantics of LSL: A Specification Language for Program Auralization*, W. Lafayette, IN: Dept. of Computer Sciences, Purdue University Sept. 21, 1993.
25. Bock, D. S., ADSL: An Auditory Domain Specification Language for Program Auralization, in *Proc. Second International Conference on Auditory Display, ICAD '94,* 1994, 303.
26. Bock, D. S., Auditory Software Fault Diagnosis Using a Sound Domain Specification Language, Ph.D. thesis, Syracuse University, Syracuse, 1995.
27. Boom M. Music Through MIDI. Microsoft Press, Redmond WA, 1987. 242.
28. Brachman R. J. and Schmolze J. G. An overview of the KL-ONE knowledge representation system. Cognitive Science, 1985, 171-216.
29. Bregman A. S. Asking the `what if' question in auditory perception. In M. Kubovy and J. R. Pomerantz, editors, Perceptual Organisation, Hillsdale, New Jersey, 1981.
30. Bregman A. S. and Campbell J. Primary auditory stream segregation and perceptual order in rapid sequences of tones. Experimental Psychology, 1971, 89-244.
31. Bregman A. S. Auditory Scene Analysis. Cambridge, Mass., MIT Press, 1990.
32. Brewster, S. A., Providing a model for the use of sound in user interfaces. *University of York, Department of Computer Science, Technical Report YCS169,* 1991.
33. Brewster, S. A. Providing a structured method for integrating non-speech audio into human-computer interfaces. *PhD Thesis,* University of York, UK, 1994.
34. Brewster, S. A., Wright, P. C. & Edwards, A. D. N. The design and evaluation of an auditory- enhanced scrollbar. *In Proceedings of CHI 94',* Boston, MA: ACM Press, Addison-Wesley, 1994, 173-179.

35. Brewster, S. A., Wright, P. C., Dix, A. J. & Edwards, A. D. N. The sonic enhancement of graphical buttons. *In Proceedings of Interact 95'*, 1995, 43-48.
36. Brewster, S., The Design of Sonically-Enhanced Widgets, *Interacting with Computers*, 1998, 211-235.
37. Brewster, S., Using Non-Speech Sounds to Provide Navigation Cues, *ACM Transactions on Computer-Human Interaction*, 1998, 224-259.
38. Brewster, S., Wright, P. C. and Edwards, A. D. N., Parallel Earcons: Reducing the Length of Audio Messages, *International Journal of Human-Computer Studies*, 1995, 153-175.
39. Brewster, S. A., A Sonically-Enhanced Interface Toolkit, in *Proc. Third International Conference on Auditory Display*, 1996, 47-50.
40. Brewster, S. A., Raty, V. and Kortekangas, A., Enhancing Scanning Input with Non-Speech Sounds, in *Proc. ASSETS 96*, 1996, ACM, 10-14
41. Brewster, S. A., Wright, P. C. and Edwards, A. D. N., A Detailed Investigation into the Effectiveness of Earcons, in *Auditory Display*, vol. XVIII, *Santa Fe Institute, Studies in the Sciences of Complexity Proceedings*, G. Kramer, Ed. Reading, MA: Addison-Wesley, 1994, 471-498
42. Brewster S. A, Raty V, and Kortekangas A. Earcons as a method of providing navigational cues in a menu hierarchy. In Proceedings of HCI'96, Imperial College, London, British Computer Society, 1996, 167-183.
43. Brewster, S., Sonically Enhanced Drag and Drop, in *Proc. ICAD 98 International Conference on Auditory Display*, 1998, 88-99.
44. Brown, M. L., Newsome, S. L. & Gilnert, E. P. An experiment into the use of auditory cues to reduce visual workload. *In Proceedings CHI 89'*, ACM Press, Addison-Wesley Publishing Company, 1989, 339-346.
45. Brown, M. H. & Hershberger, J. Color and Sound in Algorithm Animation. *Computer*, 25(12), 1992, 52-63.
46. Brown, M. H. & Hershberger, J. Zeus: A System for Algorithm Animation. *In Proceedings IEEE 91' Workshop on Visual Languages*, IEEE CS Press, 1991, 4-9.

47. Butler D. A further study of melodic channelling. Perception and Psychophysics, 1979, 264.
48. Butler D. The musician's guide to perception and cognition. Schirmer Books, New York, 1992.
49. Buxton, W., Baecker, R. M. and Arnott, J., A Holistic Approach to User Interface Design, *Unpublished manuscript*, 1985.
50. Cameron E. J, Cohen D. M, Gopinath B, Keese W. M, Ness L, Uppaluru P, and Vollaro J. R. The ic* model and environment. IEEE Transactions on Software Engineering., 1988, 317-326.
51. Casabona H. and Frederick D. Using MIDI. GPI Publications, Cupertino, CA, 1987.
52. Clarke E. Perceiving musical time. Music Perception, 1990, 213-251.
53. Cohen, J. Monitoring Background Activities. *Auditory Display: Sonification, Audification and Auditory Interfaces*. Reading, MA: Addison-Wesley Publishing Company, 1994, 499-531.
54. Cohen M, and Ludwig L. F. Multidimensional audio window management. Man Machine Studies, 1991, 319-336.
55. Colquhoun, W. P. Evaluation of auditory, visual and dual mode displays for prolonged sonar monitoring tasks. *Human Factors,* 17, 1975, 425-437.
56. Crease, M. G. and Brewster, S. A., Making Progress With Sounds - The Design and Evaluation Of An Audio Progress Bar, in *Proc. ICAD'98 International Conference on Auditory Display*, 1998, 23-34.
57. Creative Labs, Inc. (408) 546-6773 http://www.creative.com
58. Crystal River Engineering12350 Wards Ferry, RoadGroveland, CA 95321. (209) 962-6382.
59. Davis J. B. and Jennings J. The reproduction of familiar melodies and the perception of tonal sequences. Acoustical Society of America, 1977, 534-541.
60. Davis J. B. and Barclay G. Consonance and dissonance and the fusion of non-simultaneous tones. Psychology of music, 1977.
61. Davis J. B. and Yelland A. Effects of training on the production of melodic contour in memory of tonal sequences. Psychology of music, 1977.

62. De Furia S. and Scacciaferro J. MIDI Programmers Handbook. M and T Publishing, Redwood City, CA, 1989.
63. Deliege I. A perceptual approach to contemporary musical forms. Contemporary Music Review, 1989, 21-230.
64. Deliege I. and El Ahmadi A. Mechanisms of cue extraction in musical groupings. Psychology of Music, 1990. 18-44.
65. Delis D, Fleer J, and Kerr P. Memory for music. Perception and Psychophysics, 1978, 215-218.
66. Deutsch, D. The processing of structured and unstructured tonal sequences. *Perception and Psychophysics*, 1980, 381-389.
67. Deutsch D. Grouping mechanisms in music. In The Psychology of Music, New York, 1982.
68. Deutsch D. The processing of pitch combinations. In D. Deutsch, editor, Psychology of Music. Academic Press, Orlando, Florida, 1982, 239.
69. Deutsch D. Tones and numbers: Specificity of interference in short-term memory. Science, 1970, 5.
70. Deutsch D. Mapping of interactions in the pitch memory store. Science, 1972, 22.
71. Deutsch D. Interference in memory between tones adjacent in the musical scale. Experimental Psychology, 1973, 228-231.
72. Deutsch D. and Feroe J. Disinhibition in pitch memory. Perception and Psychophysics, 1975, 320-324.
73. Dibben, N., The Cognitive Reality of Hierarchic Structure in Tonal and Atonal Music, *Music Perception*, 1994, 1-26.
74. DiGiano, C. J. & Baecker, R. M. Program Auralisation: Sound Enhancements to the Programming Environment. *In Proceedings of Graphics Interface 92'*, 1992, 44-52.
75. DiGiano, C. J., Baecker, R. M. & Owen, R. N. LogoMedia: A sound- enhanced programming environment for monitoring program behaviour. *In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel & T. White (Ed.), INTERCHI 93', Addison-Wesley*, 1993, 301-302.

76. Dombois, F. Using Audification in Planetary Seismology. In Proc. ICAD2001. http://www.acoustics.hut.fi/icad2001/proceedings/navig/toc.htm
77. Domingue, J., Price, B. A. & Eisenstadt, M. A Framework for Describing and Implementing Software Visualization Systems. *In Proceedings of Graphics Interface,* 1992, 53.
78. Dowling W. J. and Fujitani D. S. Contour, interval and pitch recognition in memory for melodies. Acoustical Society of America, 1971, 524-531.
79. Dowling W. J. The perception of interleaved melodies. Cognitive Psychology, 1973, 322-337.
80. Dowling W. J. Pitch structure. In P. Howell, R. West, and I. Cross, editors, Representing Musical Structure, Academic Press, London. 1991, 33-57.
81. Dowling W. J. Musical experience and tonal sequences in the recognition of octave scrambled melodies. Psychomusiocology, 1984, 13-32.
82. Dressler R., Dolby Pro Logic Surround Decoder Principles of Operation, Dolby Laboratories 1988. http://www.dolby.com/tech/whtppr.html
83. Duda R. O. 3-D Audio for HCI. Course notes. Department of Electrical Engineering, San Jose State Unniversity. 1997.
84. Durlach, N.I. and Pang, X.D. Interaural Magnification, J. Acnl~.stical Societv of America, 1986, 1849-1850.
85. Edwards, A. D. N. Soundtrack: An auditory interface for blind users. *Human-Computer Interaction,* 4(1), 1989, 45-66.
86. Edwards A.D.N, Challis B.P, Hankinson J.C.K and Pirie F.L. Development of a Standard Test of Musical Ability for Participant in Auditory Interface Testing. *International Conference on Auditory Display (ICAD)*, 2000.
87. Francioni, J. M., Albright, L. and Jackson, J. A., Debugging Parallel Programs Using Sound, *SIGPLAN Notice*s, 1991, 68-75.
88. Francioni, J. M., Jackson, J. A. and Albright, L., The Sounds of Parallel Programs, in *Proc. 6th Distributed Memory Computing Conference*, 1991, 570-577.
89. Frixione M., Ardizzone E., Camurri A. and Zaccaria R.. A hybrid scheme for action representation. International Journal of Intelligent Systems, John Wiley and Sons, 1992.

90. Furnas, G. W., Generalized fisheye views. In *Proc. ACM SIGCHI '86 Conf. On Human Factors in Computing Systems*, 1986, 16-23.
91. Gaglio S., Costa M., Frixione M. and Palladino D. Procne: a prolog tool combining logic and semantic nets. Technical report, Computer Music Laboratory, University of Genoa, 1988.
92. Gaver, William. The SonicFinder: An Interface that Uses Auditory Icons. *Human-Computer Interaction,* 1989, 71-84.
93. Gaver, W. W. Auditory Icons: Using sound in computer interfaces. *Human-Computer Interaction,* 2, 1986, 167-177.
94. Gaver, W. W., Using and Creating Auditory Icons, in *Auditory Display*, vol. XVIII, *Santa Fe Institute, Studies in the Sciences of Complexity Proceedings*, G. Kramer, Ed. Reading, MA: Addison-Wesley, 1994, 417-446
95. Gaver, W. W. and Smith, R. B., Auditory Icons in Large-Scale Collaborative Environments, in *Proc. Human-Computer Interaction: Interact '90* Cambridge, UK, 1990, 735-740
96. Gaver, W. W., Smith, R. B. and O'Shea, T., Effective Sounds in Complex Systems: The ARKola Simulation, in *Proc. CHI '91,* ACM Press/Addison-Wesley, 1991, 85-90.
97. Gaver, W., Moran, T., MacLean, A., Lövstrand, L., Dourish, P., Carter, K. and Buxton, W., Realizing a Video Environment: EuroPARC's RAVE System, in *Proc. CHI '92,* ACM Press/Addison-Wesley, 1992, 27-35
98. Genuit, K. A Description of the Human Outer Ear Transfer Functions By Elements of Communication Theory, Proceeding 12th I Q, Toronto, 1986.
99. Grey J. M. Multidimensional perceptual scaling of musical timbres. Journal of the Acoustical Society of America, 1982, 1270-1277.
100. Hankinson, J. and Edwards, A. D. N., *Musical Grammars as the Basis of Non-Speech Audio Communication*, poster presentation at ICAD '98 International Conference on Auditory Display, Glasgow, 1998.
101. Harris J. The decline of pitch discriminators with time. Journal of Experimental Psychology, 1952, 96-99.
102. Hayashi, K. and Manakata, N., Basically Musical, *Nature* (310), 1984, 96.

103. Hayashi, K. and Manakata, N., in *Mazes for the Mind: Computers and the Unexpected*, C. A. Pickover, Ed. New York: St Martins Press, 1992.

104. Hayward, C., Listening to the Earth Sing, in *Auditory Display*, vol. XVIII, *Santa Fe Institute, Studies in the Sciences of Complexity Proceedings*, G. Kramer, Ed. Reading, MA: Addison-Wesley, 1994, 369-404

105. Heath, M. T. & Etheridge, J. A. Visualizing the Performance of Parallel Programs. *IEEE Software,* 8(5), 1991, 29-39.

106. Hotchkiss, R. S. and Wampler, C. L., The Auditorialization of Scientific In-formation, in *Proc. Supercomputing*, ACM Press, 1991, 453-461

107. Human Machine Interfaces, Inc. Nick Skrepetos. 541/687-6509 http://www.humanmachine.com

108. Innocenti C., Camurri A. and Massucco C. A multi-paradigm software environment for the real-time processing of sound, music and multimedia. Technical report, Computer Music Laboratory, University of Genoa, Via Opera Pia 11A, I-16145 Genoa, Italy, January 1993.

109. Innocenti C., Camurri A., Frixione M. and Zaccaria R. Sinets and analogical models for reasoning in dynamic domains. Technical report, Computer Music Laboratory, University of Genoa, Italy, 1993.

110. Jackson, J. & Francioni, J. Synchronisation of Visual and Aural Parallel Program Performance Data. *Auditory Display: Sonification, Audification and Auditory Interfaces.* Reading, MA: Addison-Wesley Publishing Company, 1994, 291-306.

111. Jackson, J. A. & Francioni, J. M. Aural Signatures of Parallel Programs. *In Proceeding of 25th Hawaii International Conference on System Sciences,* 1992, 218-229.

112. Jameson, D. H., The Run-Time Components of Sonnet, in *Proc. Second International Conference on Auditory Display ICAD '94*, 1994, 241-250.

113. Jameson, D. H., Sonnet: Audio-Enhanced Monitoring and Debugging, in *Auditory Display*, vol. XVIII, *Santa Fe Institute, Studies in the Sciences of Complexity Proceedings*, G. Kramer, Ed. Reading, MA: Addison-Wesley, 1994, 253-265.

114. Kahn, K. M. & Saraswat, V. A. Complete Visualizations of Concurrent Programs and Their Execution. *In Proceedings 1990 Workshop on Visual Languages,* IEEE CS Press, Los Alamitos, CA, 1990, 7-15.

115. Kaye, D. & LeBrecht, J. Sound and Music for the Theatre: The Art and Technique of Design. *Backstage Books*, 1992.

116. Kennel, A. R., AudioGraf: A Diagram-Reader for the Blind, in *Second Annual ACM Conference on Assistive Technologies*, *Vision Impairments*, 1996, 51-56

117. King, R. D. and Angus, C., PM—Protein Music, *CABIOS*, 12 (3), 1996, 251-252.

118. Kivy P. Music Alone. Cornell University Press, Ithaca, 1990. 153.

119. Kramer, G. Some Organising Principles for Representing Data with Sound. *In G. Kramer (Ed.), Auditory Display: Sonification, Audification and Auditory Interfaces.* Reading, MA: Addison-Wesley Publishing Company, 1994.

120. Kramer, G. Auditory display, sonification, audification and auditory interfaces. *The Proceedings of the First International Conference on Auditory Display.* Reading, MA: Santa Fe Institute, Addison-Wesley, 1992, 32-39.

121. Kramer, G., Preface in *Auditory Display,* Vol. 18, *Addison-Wesley,* 1994.

122. Kubovy M. Concurrent-pitch segregation and the theory of indispensable attributes. In Kubovy M. and Pomerantz J. R., editors, Perceptual Organisation, Hillsdale, New Jersey, 1981.

123. Lake Technologies (formerly Lake DSP). http://www.laketechnologies.com

124. Langer S. Philosophy in a new key (3rd ed.). Harvard University Press, Cambridge, MA, 1956.

125. Leplatre, G. and Brewster, S. A., An Investigation of Using Music to Provide Navigation Cues., in *Proc. ICAD'98 International Conference on Auditory Displa,* 1998, 35-52.

126. Lerdahl F. and Jackendo R. A Generative Theory of Tonal Music. MIT Press, Cambridge, Massachussetts, 1983. 238.

127. Lucas, P. A., An Evaluation of the communicative Ability of Auditory Icons and Earcons, in *Proc. Second International Conference on Auditory Display ICAD '94* 1994, 12-23.

128. Lunney, E. and Morrison, R. C., High Technology Laboratory Aids for Visually Handicapped Chemistry Students, *Journal of Chemical Educatio*n, 58 (3), 1981, 228-231.

129. Lunney, D. and Morrison, R. C., Auditory Presentation of Experimental Data, in *Extracting Meaning from Complex Data: Processing, Display and Interactio*n, vol. 1259, E. J. Farrell, Ed., 1990, 140-146.

130. Madhyastha, T. M. and Reed, D. A., A Framework for Sonification Design, in *Auditory Displa*y, vol. XVIII, *Santa Fe Institute, Studies in the Sciences of Complexity Proceeding*s, G. Kramer, Ed. Reading, MA: Addison-Wesley, 1994, 267-290.

131. Madhyastha, T. M. and Reed, D. A., Data Sonification: Do You See What I Hear? *IEEE Software* (March), 1995, 45-56.

132. Mansur, D., Blattner, M. and Joy. K. Soundgraphs: A Numerical Data Analysis method for the Blind. *Proc. 18th Hawaii Conference on System Sciences*, 1985, 163-174.

133. Marrin T. and Picard R., Analysis of Affective Musical Expression with the Conductor's Jacket, Proceedings of the XII Colloquium on Musical Informatics, Gorizia, Italy, September 1998.

134. Martens, W.L. Principal Components Analysis and Resynthesis of spectral Cues to Perceive Direction. Proceedings of the International Computer Music Conference, 1987.

135. Martens, W.L., Kendall, G.S. Freed, D.J. , Ludwig, M.D., & Karstens, R. The evaluation of Digital Filters for Controlling Perceived Direction in Stereo Loudspeaker Reproduction. Paper presented at the Audio Engineering Society 81st Convention, Los Angeles, November, 1986

136. Massey H. MIDI Home Studio. Music Sales Corporation, New York, 1988.

137. Mathur, A. P., Boardman, D. B. and Khandelwal, V., LSL: A Specification Language for Program Auralization, in *Proc. Second International Conference on Auditory Display, ICAD '94,* 1994, 257-264.

138. Mayer-Kress, G., Bargar, R. and Choi, I., Musical Structures in Data from Chaotic Attractors, in *Auditory Display*, vol. XVIII, *Santa Fe Institute, Studies in the Sciences of Complexity Proceedings*, G. Kramer, Ed. Reading, MA: Addison-Wesley, 1994, 341-368.

139. Mynatt, E. Auditory Presentation of Graphical User Interfaces. *Auditory Display: Sonification, Audification and Auditory Interfaces.* Reading, MA: Addison-Wesley Publishing Company, 1994, 533-555.

140. Mestel, R. Hearing Pictures, Seeing Sounds. *New Scientist*, 1928, 20-23.

141. Mezrich, J. J., Frysinger, S. and Slivjanovski, R., Dynamic Representation of Multivariate Time Series Data, *Journal of the American Statistical Association*, 1984, 34-40.

142. Miller, G. A. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review 63*, 1956, 81-97.

143. Miller G. A. The magical number seven plus or minus two: Some limitations on our capacity for processing information. Psychology Review, 1956, 63-97.

144. Minsky, M. Music, Mind and Meaning. *The Music Machine: Selected Readings from Computer Music Journal,* Curtis Roads (Ed.), 1989, 639-655.

145. Moog, R. MIDI: Musical Instrument Digital Interface. *Journal of the Audio Engineering Society*, 1984, 394-404.

146. Mountford, S. J. & Gaver, W. W. Talking and listening to computers. *In B. Laurel. (Ed.), The Art of Human-Computer Interface Design,* Reading, Massachusetts: Addison-Wesley Publishing Company, 1990, 319-334.

147. Mynatt, E. & Weber, G. Providing Access to Graphical User Interfaces: Contrasting Two Approaches. *In Proceedings CHI 94',* Boston, MA, 1994, 24-28.

148. O'Leary, A. & Rhodes, G. Cross-modal effects on visual and auditory object perception. *Perception and Psychophysics,* 35(6), 1984, 565-569.

149. Paivio A., Mental Representations. *New York : Oxford University Press*, 1986.

150. Paivio A., Imagery and Verbal Processes. *New York : Holt, Rinehart and Winston*, 1971.

151. Paivio A., Dual Coding Theory : Retrospect and Current Analysis. *Canadian Journal of Psychology, 45 (3)*, 1991, 255-287.

152. Papert, Resnick, Mitchel, Stephen Ocko. LEGO, Logo, and Design. *Children's Environments Quarterly*, Vol. 5, No. 4. 1988.

153. Perrot, D., Sadralobadi, T., Saberi, K. & Strybel, T. Aurally aided visual search in the central visual field: Effects of visual load and visual enhancement of the target. *Human Factors,* 33(4), 1991, 389-400.

154. Persterer, A. A Very High Performance Digital Audio Signal Processing System, ASSP Workshop on applications of Signal Processinq to Audio and Acoustics, New Paltz N.Y. 1998.

155. Pollard-Gott L. Emergence of thematic concepts in repeated listening to music. Cognitive Psychology, 1983, 66-94.

156. Prosoniq Products Software GmbH, Audio Research & Development Blumenstrasse ,2376133 Karlsruhe, Germany. http://www.prosoniq.com

157. Rayleigh, Lord. (J. W. Strutt, 3rd Baron of Rayleigh), On our perception of sound direction, Philos. Mag. 13, 1907, 214-232.

158. Rigas, D. I. and Alty, J. L., How Can Multimedia Designers Utilize Timbre? *In People and Computers XIII*, H. Johnson, L. Nigay and C. Roast, Eds. Springer-Verlag, 1998, 273-286.

159. Rigas, D. I. and Alty, J. L., The Use of Music in a Graphical Interface for the Visually Impaired, in *Proc. Interact '97,* 1997, 83-99.

160. Rigas, D. I., Guidelines for Auditory Interface Design: An Empirical Investigation, Ph.D. thesis, Loughborough University, Leicestershire, UK, 1996.

161. Risset J. C. Musical acoustics. Handbook of Perception, 4, 1978.

162. Roederer J. G. The physics and psychophysics of music: An introduction. Springer-Verlag, Berlin, 3rd edition, 1994.

163. Rothstein J. MIDI: A Comprehensive Introduction. A-R Editions, Madison, WI, 1991.

164. Rumsey F. MIDI Systems and Control. Focal Press, Boston, MA, 1990.

165. Scaletti, C., Sound Synthesis Algorithms for Auditory Data Representation, in *Auditory Display*, vol. XVIII, *Santa Fe Institute, Studies in the Sciences of Complexity Proceedings*, G. Kramer, Ed. Reading, MA: Addison-Wesley, 1994, 223-252

166. Scaletti, C. and Craig, A. B., Using Sound to Extract Meaning from Complex Data, in *Extracting Meaning from Complex Data: Processing, Display, Inter-action*, vol. 1259, E. J. Farrel, Ed. San Jose, California: SPIE, 1990, 207-219

167. Schenker, H., *Free Composition*, vol. III. New York London: Longman, 1979.

168. Sensaura Ltd, Scipher Building, Dawley Road, Hayes, Middlesex, UB3 1HH. http://www.sensaura.com

169. Sloboda J. A. The Musical Mind, The Cognitive Psychology of Music. Clarendon Press, Oxford, England, 1985.

170. Sonnenwald, D. H., Gopinath, B., Haberman, G. O., Keese, W. M., III and Myers, J. S., InfoSound: An Audio Aid to Program Comprehension, in *Proc. Twenty-Third Hawaii International Conference on System Sciences*, IEEE Computer Society Press, 1990, 541-546

171. Stasko, J. T. Tango: A Framework and System for Algorithm Animation. *Computer*, 23(9), 1990, 27-39.

172. Stevens, R. D., Brewster S. A., Wright, P. C. & Edwards, A. D. N. Providing an audio glance at algebra for blind readers. *In G. Kramer (Ed.) Proceedings of ICAD 94'*, 1994.

173. Stevens S. S. The relation of pitch to intensity. Journal of the Acoustical Society of America, 1935, 150-154.

174. Sumikawa, D., Blattner, M. & Greenberg, R. Earcons: Structured Audio Messages. *Unpublished paper*, 1986.

175. Sumikawa, D., Blattner, M. & Greenberg, R. Guidelines for the syntactic design of audio cues in computer interfaces. *Technical Report No. UCRL 92925,* Lawrence Livermore National Laboratory, 1986.

176. Sumikawa, D. Guidelines for the integration of audio cues into computer user interfaces. *Technical Report No. UCRL 53656*, Lawrence Livermore National Laboratory, 1986.

177. Tan N, Aiello R, and Bever T. G. Harmonic structure as a determinant of melodic organisation. Memory and Cognition, 1981, 533-539.

178. The Concise Oxford Dictionary. Clarendon Press, Oxford.

179. Thimbleby, H. User Interface Design. New York: ACM Press, Addison-Wesley. 1990.

180. Vanderveer, N. J. Ecological acoustics: Human perception of environmental sounds. *Thesis No. 40/09B, 4543),* Dissertation Abstracts International, 1979.

181. Vickers, P. & Alty, J. L. CAITLIN: A Musical Program Auralisation Tool to Assist Novice Programmers with Debugging. In *Proc. Third International Conference on Auditory Display* (Palo Alto, Nov 4-6, 1996), Xerox PARC, Palo Alto, CA 94304, 17-24

182. Vickers, P, CAITLIN: Implementation of a Musical Program Auralisation System to Study the Effects on Debugging Tasks as Performed by Novice Pascal Programmers, *PhD Thesis,* Loughborough University, UK, 1999.

183. Weiss, M. Data Structures and Algorithm Analysis. Benjamin/Cummings, second edition, 1995.

184. Wagenaar, W. A., Varey, C. A. & Hudson, P. T. Do audiovisuals aid? A study of bisensory presentation on the recall of information. *In H. Bouma & D. G. Bouwhuis (Eds.), Attention and Performance,* 7(6), 1984, 379-391.

185. Walker, J. T. & Scott, K. J. Auditory-visual conflicts in the perceived duration of lights, tones and gaps. *Journal of Experimental Psychology: Human Perception and Performance,* 7(6), 1981, 1327-1339.

186. Walker D. How MIDI works? Alexander Publishing, Newbury Park, CA, 1989.

187. Warren, W. H. & Verbugge, R. R. Auditory perception of breaking and bouncing events: A case study in ecological acoustics. *Journal of Experimental Psychology: Human Perception and Performance,* 10, 1984, 704-712.

188. Watkins J. A. Scale, key and contour in the discriminations of tuned and mistuned approximations to melody. Perception and Psychophysics, 1985, 275-285.

189. Watkins, A. J., Psychoacoustical Aspects of Synthesized Vertical Locale Cues, J. Acoust. Sot. Am., Vol. 63, 1978, 1152-1165.

190. Wenzel, E. M. Three- dimensional virtual acoustic displays. *Presence: teleoperators and virtual environments,* 1, 1992, 80-107.

191. Wenzel, E. M., Spatial Sound and Sonification, in *Auditory Display*, vol. XVIII, *Santa Fe Institute, Studies in the Sciences of Complexity Proceedings*, G. Kramer, Ed. Reading, MA: Addison-Wesley, 1994, 127-150.

192. Wenzel, E.M. Virtual Acoustic Displays, Conference on Human Machine Interfaces for Teleoperators and Virtual Environments, Santa Barbara, CA. 1990.

193. Wenzel, E.M., Wightman, F.L., Kistler D.J. and Foster, S.H. Acoustic Origins of Individual Differences in Sound Localization Behavior, J. Acoust. Soc.Am. Volume 84, 1988.

194. Wenzel, E. M., F. L. Wightman, D. Kistler, and S. H. Foster (1988). The Convolvotron: Realtime synthesis of out-of-head localization, Second Joint Meeting Acoustic Societies of America and Japan, Honolulu, HI.

195. Wickens, C. D., Mountford, S. J. & Schreiner, W. Multiple resources, task-hemispheric integrity and individual differences in time sharing. *Human Factors,* 23(2), 1981, 211-229.

196. Wightman, F.L. and Kistler, D.J. (1989a) Headphone Simulation of Freefield Listening I: Stimulus Synthesis. J. of Acoust. Soc. Am, 1989, 858-867.

197. Wightman, F.L., and Kistler, D.J. Headphone Simulation of Freefield Listening II: Psychophysical Validation. J. of Acoust. .Soc. Am. 1998, 868-878.

198. Winn J. A. Unsuspected eloquence: A history of the relations between poetry and music. Yale University Press, New Heaven, CT, 1981.

199. Wolf F. I. and Marsnik N. C. Perceptive Listening. Harcourt Brace Jovanovich, Orlando, Florida, 1992.

200. Wolpert R. Recognition of melody, harmonic accompaniment, and instrumentation: musicians vs. non-musicians. Music Perception, 1990, 95-105.

201. Woods W. A. and Schmolze J. G. The kl-one family. Computers Math. Applic., Pergamon Press, 1992, 133-177.

202. Zettl, Herbert. Sight, Sound, Motion: Applied Media Aesthetics. *Belmont, Ca: Wadsworth*, 1990.

B. Musical ability questionnaire.

IMPACT Research Group - Musical Ability Questionnaire.

Welcome, thank you for agreeing to take part in this test which will last approximately 40 Minutes. We are trying to determine how the AVERAGE person processes musical information. Some of you may be professional musicians and others may describe themselves as having no musical ability at all, but we need all of these types in our study so do not worry if you consider yourself not to be musical.

You can neither pass nor fail this test, we are simply looking for results in order for us to determine an average viewpoint. This series of tests is designed at understanding how each of us as individuals perceives music, hence there are no right or wrong answers.

Your participation in this test is entirely voluntary and you may decline to take part at any point, but please leave quietly so as not to disturb the other participants.

Please ensure that you have a pen or pencil ready for the test.

1.0 - Your Age 10-19 / 20-29 / 30-39 / 40-49 / 50-59 / 60-69 / 70+
1.1 - Your Sex Male / Female
1.2 - Education None / O Level / A Level / Degree / Higher Degree / Higher+
1.3 - Ethnic Origin (This is only to determine the culture that you may have experienced)

White African Afro-Caribbean Chinese Indian Other

2.0 - How would you classify your interest in music ?

I have no interest in music at all

I enjoy music as background or to dance to

I am very interested in music as a listener

I enjoy performing music to myself or friends

I play music to others (not just close friends)

I am a professional musician

2.1 - If you play an instrument/s, please indicate below :

Level	Instrument
1.	Not at all (never)
2.	Can play a simple tune
3.	Reasonably competent
4.	Very competent - can play before others
5.	Professional

Which instrument -........................ At what level - 1, 2, 3, 4, 5

.... 1, 2, 3, 4, 5

.... 1, 2, 3, 4, 5

2.2 - Do you sing ?

Not at all - no ability

Sing to myself, but embarrassed in public

Will sing along with others

Sing in a choir

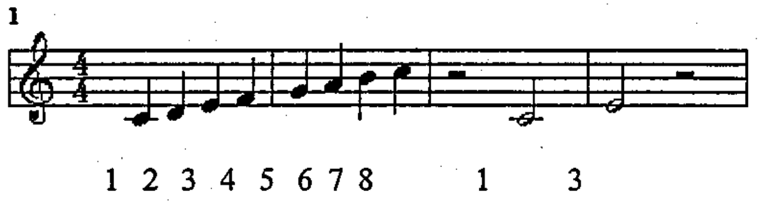
Will confidently sing solo in public

Professional singer

C. Pitch perception workbook.

IMPACT Research Group – Pitch Perception Test.

3.0 - Pitch.

In this section we are going to play two notes and ask you to determine the numerical difference between them. To enable you to do this we select notes from the normal scale of 8 notes and we want you to think of them as having values from 1 through to 8. The bottom note having a numerical value of 1 and the top note of the scale having a numerical value of 8. To help you we will play a demonstration, you will hear the scale first then you will hear the two notes, this will be repeated three times over.

The answer to this demonstration test is :

Note 1 ......1...... Note 2 ......3...... Difference ......2......

Now we will carry out ten of these tests. After you have heard the last of each three repetitions, please write your answers down in the spaces provided.

3.1 - Note 1 .... Note 2 .... Difference ....
3.2 - Note 1 .... Note 2 .... Difference ....
3.3 - Note 1 .... Note 2 .... Difference ....
3.4 - Note 1 .... Note 2 .... Difference ....
3.5 - Note 1 .... Note 2 .... Difference ....
3.6 - Note 1 .... Note 2 .... Difference ....
3.7 - Note 1 .... Note 2 .... Difference ....
3.8 - Note 1 .... Note 2 .... Difference ....
3.9 - Note 1 .... Note 2 .... Difference ....
3.10 - Note 1 .... Note 2 .... Difference ....

D. Shape perception workbook.

IMPACT Research Group – Shape Perception Test.

4.0 - Musical Shape Perception.

You will now hear 6 short musical sequences, in this test unlike the previous one the scale will be played only once at the beginning of each test, then each sequence will be repeated 3 times after which you must draw in the boxes the shape of the music by placing X's. The boxes are 8 high, the musical sequences are all within this scale of 8 notes (1 being the lowest note and 8 being the highest note). Don't worry about the timing.

4.1 -

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

4.2 -

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

4.3 -

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

4.4 -

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

4.5 -

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

4.6 -

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

E. List state perception workbook.

IMPACT Research Group – List Shape Perception Test.

5.0 - Examining Lists of Numbers.

Lists are everywhere in our daily lives, they come in many forms from names in an address book to postal codes in a mail sorting office. These lists often need to be sorted into some order, such as surnames into alphabetical order or postal codes into alphanumeric order. There are many different recipes available for sorting such lists, this is because there are many different ways in which they can be sorted. Each entry into a list is known as an element, such as one persons contact details in an address book. The concepts introduced in this section of the test involve the use of lists of numbers, these numbers (elements) have been mapped to music - the higher the number, the higher the note.

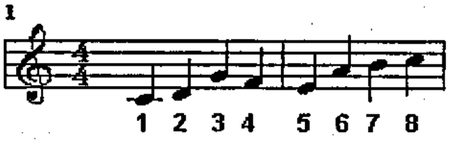
The following example is an unsorted list of 8 numbers :	4, 2, 8, 5, 1, 7, 3, 6
Here is the same list of numbers sorted into descending order :	8, 7, 6, 5, 4, 3, 2, 1
and here is the same list of numbers sorted into ascending order :	1, 2, 3, 4, 5, 6, 7, 8

You will now hear 5 different lists, each list will be repeated 3 times in a row. For each of these lists we want you to identify which are ***unsorted***, ***sorted into ascending*** order or ***sorted into descending*** order.

5.1 -	Unsorted ☐	Sorted - Ascending ☐	Sorted - Descending ☐
5.2 -	Unsorted ☐	Sorted - Ascending ☐	Sorted - Descending ☐
5.3 -	Unsorted ☐	Sorted - Ascending ☐	Sorted - Descending ☐
5.4 -	Unsorted ☐	Sorted - Ascending ☐	Sorted - Descending ☐
5.5 -	Unsorted ☐	Sorted - Ascending ☐	Sorted - Descending ☐

F. Out of place element perception workbook.

IMPACT Research Group – Out Of Place Element Perception Test.

Think about how the patterns of the notes help you to determine if a list is sorted or not. If we require a list that is sorted into ascending order, then we can identify an incorrectly placed element because it would cause a descent in pitch in the list and would deem it unsorted. This can be seen in the following example where elements 4 and 5 cause a descent in pitch and are therefore incorrectly placed.

You will now hear another 5 lists of numbers, each repeated three times in a row. We want these lists to be sorted into ascending order, see if you can identify which numbers (elements) are out of place in each list. (Ring the elements that you think are out of place).

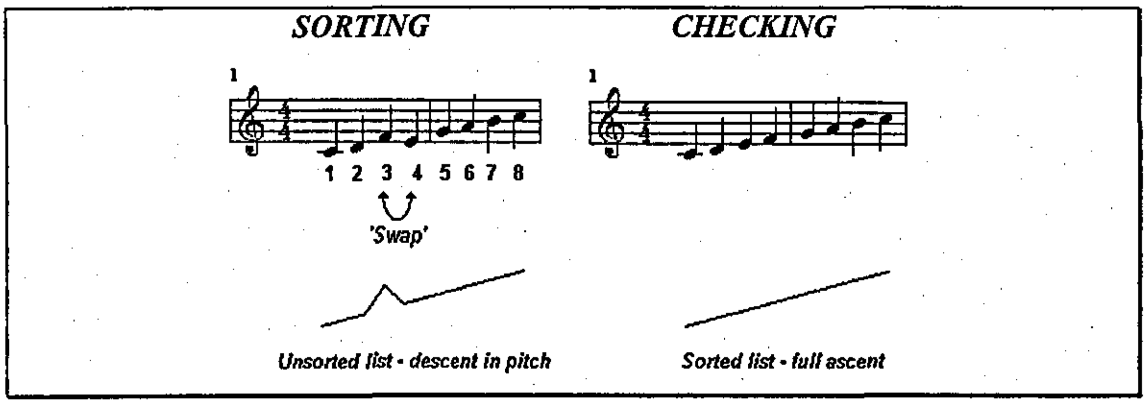
5.6 - A B C D E F G H
5.7 - A B C D E F G H
5.8 - A B C D E F G H
5.9 - A B C D E F G H
5.10 - A B C D E F G H

G. Element swapping perception workbook.

IMPACT Research Group – Element Swapping Perception Test.

6.0 - Manipulating Lists of Numbers.

So far we have only checked the state of a list by listening to the order of its elements, we are now going to introduce the concept of element swapping. This is done by simply swapping two neighbouring elements for each other, this 'neighbour swapping' action forms the basis of one of our sorting recipes known as the Bubble Sort. The following example shows the swapping of two elements after a descent in pitch indicated that element 4 should be placed before element 3. You will first hear the list being sorted, this will then be followed by a checking of the list. Listen for the musical sequences that denote the occurrence of a swap during the sorting phase and the occurrence of success after the checking phase.



You can now see that not only have we mapped music to the elements in the lists, but also to the action of swapping. You will now hear another five lists, each will be repeated three times in a row. See if you can identify which elements are being swapped. (Ring the neighbouring element pairs that are swapped)

6.1 - **A B C D E F G H**
6.2 - **A B C D E F G H**
6.3 - **A B C D E F G H**
6.4 - **A B C D E F G H**
6.5 - **A B C D E F G H**

H. Bubble Sort algorithm auralisation workbook.

IMPACT Research Group – Bubble Sort Algorithm Auralisation Test.

7.0 - Recipes for Sorting.

So far we have seen that we can examine and manipulate lists of numbers which are represented by musical notes. In the previous section we saw that by passing through the list we could swap incorrectly placed neighbouring elements. If we repeat this action of passing through the list several times then the list would eventually become sorted.

You will now hear this repetition of sorting to achieve a fully sorted list. As before, we will examine and manipulate our list. As well as these two techniques, listen for the addition of a wooden block that counts how many times the recipe is passing through the list to sort it. The entire recipe, which is shown below, will be played three times in a row.

Step 1 - Examine the list. 1 2 3 4 6 7 5 8 10 9
(Identify out of place elements) ^ ^ ^ ^

Step 2 - Pass through the list manipulating as we go.
On this step we swap the 7 for the 5, and the 10 for the 9.

Step 3 - Examine the new list 1 2 3 4 6 5 7 8 9 10
(Identify out of place elements) ^ ^

Step 4 - Pass through the list manipulating as we go.
On this step we swap the 6 for the 5.

Step 5 - Examine the new list 1 2 3 4 5 6 7 8 9 10
(Identify out of place elements) *None*

Step 6 - Successfully terminate and indicate that the list is sorted.

This above recipe is the recipe for the Bubble Sort that you just heard. You will now hear this recipe being used on another list, see how many of the following questions you can answer. As before the entire recipe from start to finish will be repeated three times in a row.

7.1 How many numbers (elements) are there in the list?.......................................
7.2 How many swaps are there in the first pass?..
7.3 How many swaps are there in the second pass?...
7.4 How many swaps are there in the third pass?...
7.5 How many swaps are there in the fourth pass?...
7.6 How do you know when elements are out of order?.......................................
7.7 How do you know when the recipe swaps elements?......................................

7.8 How do you know when the list is sorted?…………………………………………
7.9 How many times does the recipe pass through the list?…………………………
7.10 What order is the list sorted into?………………………………………………

You will now hear the same Bubble Sort recipe used on yet another list, but this time we have included some errors in the sorting procedure that we would like you to find. Again the entire recipe from start to finish will be repeated three times in a row.

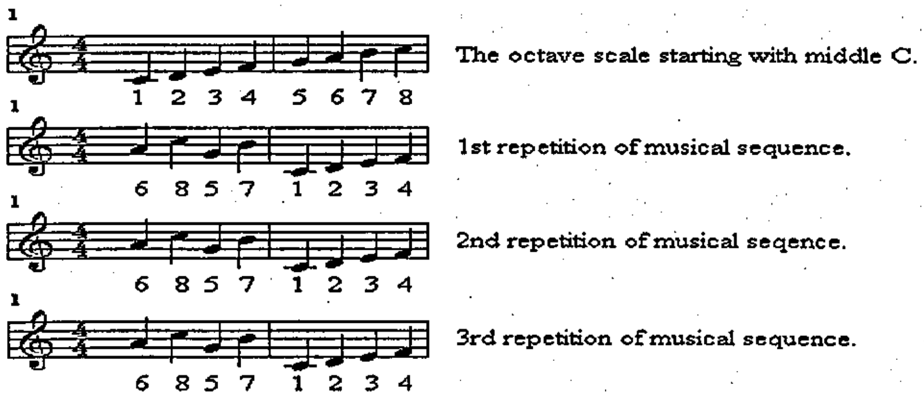
Comments on errors:

********You have now reached the end of the test, thank you for your time.*******

I. Algorithm generated shape perception workbook.

IMPACT Research Group – Algorithm Generated Shape Perception Test.

You are about to hear a series of short musical sequences. Each sequence will contain between 6 to 10 notes, these notes that you hear will all be within the same octave starting from 'Middle C'. For each test you will first hear this octave scale followed by three repetitions of the musical sequence. Musical scores showing an example comprising of 8 notes are shown below.



On the following sheets each question has three entry fields in which you should place your answers. These fields shown in the example below as 1st **Rep, 2nd Rep** and 3rd **Rep** correspond to the three sequence repetitions you will hear within each question. After the first repetition you should describe (in the 1st **Rep** field) the shape of the musical sequence. After the second repetition you should again describe the shape adding or changing anything that differs from your original description (this second description should be entered in the 2nd **Rep** field). After the third repetition you should draw the shape of the sequence in the grid provided, once you have drawn this please do not change your previous descriptions. It is also important that during the 1st and 2nd **Rep** stages you do not sketch the shape to aid your descriptions.

Some of the features that you should listen for and describe are – ***jaggedness or randomness, ascending or descending order, note repetitions, ordered except for one note, random then ordered, ordered then random*** or any combination of these features. You will now hear the example shown above, this corresponds to the answer below so listen carefully. (Remember – Scale, 1st rep, 2nd rep and 3rd rep).
Answer to Example. (Typical descriptions that would be correct).

1st Rep. – "Sort of random at the start and then smooth at the end."
2nd Rep. – "Jagged at the start then a smooth ascending pattern at the end."
3rd Rep. –

Pitch	Note Sequence -->							
8		X						
7				X				
6	X							
5			X					
4								X
3							X	
2						X		
1					X			

You will now hear the tests, listen and enter you answers on the following pages.
The following 5 tests are each comprised of 8 notes.

2.1 -
1st Rep............................
....
2nd Rep............................
....
3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

(Draw what you hear)

2.2 -
1st Rep............................
....
2nd Rep............................
....
3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

2.3-
1st Rep............................
....
2nd Rep............................
....
3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

2.4 -
1st Rep............................
....
2nd Rep............................
....
3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

2.5-

1st Rep…………………………

…………………………………

2nd Rep………………………...

…………………………………

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

The following 5 tests are each comprised of 6 notes.

3.1 -

1st Rep…………………………

…………………………………

2nd Rep………………………...

…………………………………

3rd Rep.

Pitch	Note Sequence -->					
8						
7						
6						
5						
4						
3						
2						
1						

(Draw what you hear)

3.2 -

1st Rep…………………………

…………………………………

2nd Rep………………………...

…………………………………

3rd Rep.

Pitch	Note Sequence -->					
8						
7						
6						
5						
4						
3						
2						
1						

3.3-

1st Rep…………………………

…………………………………

2nd Rep………………………...

…………………………………

3rd Rep.

Pitch	Note Sequence -->					
8						
7						
6						
5						
4						
3						
2						
1						

3.4 -

1st Rep…………………………

…………………………………

2nd Rep………………………...

…………………………………

3rd Rep.

Pitch	Note Sequence -->					
8						
7						
6						
5						
4						
3						
2						
1						

3.5-

1st Rep.............................

....

2nd Rep.............................

....

3rd Rep.

Pitch	Note Sequence -->					
8						
7						
6						
5						
4						
3						
2						
1						

The following 4 tests are each comprised of 10 notes.

4.1 -

1st Rep.............................

....

2nd Rep.............................

....

3rd Rep.

Pitch	Note Sequence --> -->									
8										
7										
6										
5										
4										
3										
2										
1										

(Draw what you hear)

4.2 -

1st Rep.............................

....

2nd Rep.............................

....

3rd Rep.

Pitch	Note Sequence --> -->									
8										
7										
6										
5										
4										
3										
2										
1										

4.3-

1st Rep.............................

....

2nd Rep.............................

....

3rd Rep.

Pitch	Note Sequence --> -->									
8										
7										
6										
5										
4										
3										
2										
1										

4.4 -

1st Rep.............................

....

2nd Rep.............................

....

3rd Rep.

Pitch	Note Sequence --> -->									
8										
7										
6										
5										
4										
3										
2										
1										

The following 5 tests are each comprised of 8 notes.

5.1 -

1st Rep............................

....

2nd Rep............................

....

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

(Draw what you hear)

5.2 -

1st Rep............................

....

2nd Rep............................

....

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

5.3-

1st Rep............................

....

2nd Rep............................

....

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

5.4 -

1st Rep............................

....

2nd Rep............................

....

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

5.5-

1st Rep............................

....

2nd Rep............................

....

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

The following 5 tests are each comprised of 8 notes.

6.1 -

1st Rep...............................

....

2nd Rep..............................

....

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

(Draw what you hear)

6.2 -

1st Rep...............................

....

2nd Rep..............................

....

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

6.3-

1st Rep...............................

....

2nd Rep..............................

....

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

6.4 -

1st Rep...............................

....

2nd Rep..............................

....

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

6.5-

1st Rep...............................

....

2nd Rep..............................

....

3rd Rep.

Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

J. Drawing ability workbook.

IMPACT Research Group – Drawing Ability Test.

In this final section we would like you to listen to a well-known tune which is played within our familiar octave scale starting from 'Middle C'. What we would then like you to do is to either sing it or hum it to the instructor, once you have done this we would like you to draw the shape of this tune in the grid provided below.

You will first hear the octave scale, this will then be followed by three repetitions of the tune. Listen carefully.

7.1 – Do you recognise this tune ? Yes / No

7.1 - Can you sing or hum this tune ? Yes / No

7.2 - Now draw the shape of the tune.

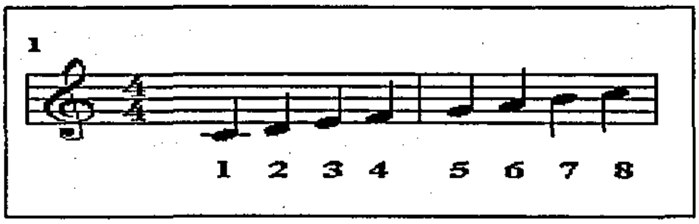
Pitch	Note Sequence -->							
8								
7								
6								
5								
4								
3								
2								
1								

K. Sorting characteristics workbook.

IMPACT Research Group – Sorting Characteristics Identification Test.

The following section is divided into 5 parts, each of these parts contains 4 or 5 musical sequences. Each musical sequence is composed of between 6 to 10 notes which are all within the same octave starting from 'Middle C' (shown in the diagram below). At the start of each of the 5 parts you will hear this octave scale, this will then be followed by the musical sequences within that part.

We have selected notes from the normal scale of 8 notes (shown below) and we want you to think of them as having values from 1 through to 8. The bottom note having a numerical value of 1 and the top note of the scale having a numerical value of 8. To help you we will play the scale shown below which corresponds to the numbers 1,2,3,4,5,6,7 and 8.



As previously mentioned, all of the sequences that you will hear will be within the octave scale shown above, therefore when a sequence contains 10 notes then it must mean that some notes in our octave scale are repeated. Similarly when a sequence contains 6 notes then it must mean that some notes in our octave scale have been omitted. Given that each note represents a number, then the sequences that you will hear represent lists of numbers.

For each of the five parts we would like you to listen carefully to the sequences and try to visualise the shapes of the lists from their musical representations. Starting with the first sequence and proceeding through to the last we would like you to try and explain what has progressively happened to the shape of the list. (E.g.- Starts random, gradually gets ordered beginning from the left, ends in ascending order etc.) Write your observations in the spaces provided for each part

You will now hear a set of example sequences, they correspond to the descriptions given below and are followed by an explanation of what has progressively happened to this example list. Remember that first you will hear the scale, this will then be followed by the five example sequences (each comprising 8 notes).

1. Scale.
2. 1st sequence – **All disordered random notes.**
3. 2nd sequence – **3 ordered** ascending notes followed by **jagged randomness.**
4. 3rd sequence – **4 ordered** ascending notes followed by **jagged randomness.**

5. 4th sequence – **6 ordered** ascending notes followed by **2 disordered notes.**
6. 5th sequence – **8 ordered** ascending notes.

What has progressively happened to the list?

The list **began in random order,** gradually **became more ordered starting from the left** and **ended up in sorted ascending order.**
You will now hear the tests, listen and enter you answers in the following sections.

PART 1 – 5 sequences each containing 8 notes.
Remember that you will first hear the scale and then you will hear the sequences. Write down in the space below what you think has progressively happened to the list.

PART 2 – 5 sequences each containing 5 notes.
Again, remember that you will first hear the scale then you will hear the sequences. Write down in the space what you think has progressively happened to the list.

PART 3 – 4 sequences each containing 10 notes.
Again, remember that you will first hear the scale then you will hear the sequences. Write down in the space what you think has progressively happened to the list.

PART 4 – 5 sequences each containing 8 notes.
Again, remember that you will first hear the scale then you will hear the sequences. Write down in the space what you think has progressively happened to the list.

PART 5 – 5 sequences each containing 8 notes.
Again, remember that you will first hear the scale then you will hear the sequences. Write down in the space what you think has progressively happened to the list.

L. Multiple algorithm auralisation workbook.

IMPACT Research Group - Musical Ability Questionnaire & Multi-Sort Information Auralisation Test.

So far we have seen that we can examine and manipulate lists of numbers which are represented by musical notes. In the previous section we saw that by passing through the list we could swap incorrectly placed neighbouring elements. If we repeat this action of passing through the list several times then the list would eventually become sorted, this is one method of implementing a sorting algorithm. In this section you will listen to several different sorting algorithms and will be asked questions about what the algorithm is doing.

9.1 - Algorithm 1.

You will now hear this repetition of sorting to achieve a fully sorted list. As before, we will examine and manipulate our list. As well as these two techniques, listen for the addition of a wooden block that counts how many times the recipe is passing through the list to sort it. The entire recipe, which is shown below, will be played three times in a row.

Example of Algorithm 1. **(B)**

Step 1 - Examine the list. (Identify out of place elements)	1 1 3 2 4 6 7 5 8 8 ^ ^ ^ ^
Step 2 - Pass through the list manipulating as we go. On this step we swap the 3 for the 2, and the 7 for the 5.	
Step 3 - Examine the new list (Identify out of place elements)	1 1 2 3 4 6 5 7 8 8 ^ ^
Step 4 - Pass through the list manipulating as we go. On this step we swap the 6 for the 5.	
Step 5 - Examine the new list (Identify out of place elements)	1 1 2 3 4 5 6 7 8 8 *None*
Step 6 - Successfully terminate and indicate that the list is sorted.	

You will now hear this recipe being used on another list, see how many of the following questions you can answer. As before the entire recipe from start to finish will be repeated three times in a row.

9.1.1 -How many numbers (elements) are there in the list?..............................
9.1.2 -How many swaps are there in the first pass?..
9.1.3 -How many swaps are there in the second pass?.......................................
9.1.4 -How many swaps are there in the third pass?..
9.1.5 -How do you know when elements are out of order?...............................
9.1.6 -How do you know when the recipe swaps elements?................................
9.1.7 -How do you know when the list is sorted?...
9.1.8 -How many times does the recipe pass through the list?.............................
9.1.9 -What order is the list sorted into?...
9.1.10 - How does the shape of the list progress?...

9.2 - Algorithm 2.

You will now hear a different algorithm that also sorts a list of numbers but in a different manner. This recipe searches all elements in the list until it finds the one that belongs in the current location, when it finds this element its swaps it into the current location. As before, we will examine and manipulate our list. As well as these two techniques, listen for the addition of a wooden block that counts how many times the recipe is passing through the list to sort it. The entire recipe, which is shown below, will be played three times in a row.

Example of Algorithm 2. **(S)**

Step 1 - Examine the list.	4 3 1 2
(Looking for correct element for current location)	^
Step 2 - Pass through the list manipulating as we go. Identify correct element for current location. On this step we swap the 4 for the 1,	4 3 1 2
	^ ^
Step 3 - Examine the new list	1 3 4 2
(Looking for correct element for current location)	^
Step 4 - Pass through the list manipulating as we go. Identify correct element for current location. On this step we swap the 3 for the 2,	1 3 4 2
	^ ^
Step 5 - Examine the new list	1 2 4 3
(Looking for correct element for current location)	^
Step 6 – Pass through the list manipulating as we go. Identify correct element for current location. On this step we swap the 4 for the 3,	1 2 4 3
	^ ^
Step 7 – Examine the new list.	1 2 3 4
Step 8 - Successfully terminate and indicate that the list is sorted.	

You will now hear this recipe being used on another list, see how many of the following questions you can answer. As before the entire recipe from start to finish will be repeated three times in a row.

9.1.1 -How many numbers (elements) are there in the list?................................
9.1.2 -How many swaps are there in the first pass?..
9.1.3 -How many swaps are there in the second pass?......................................
9.1.4 -How many swaps are there in the third pass?..
9.1.5 -How do you know when elements are out of order?.................................
9.1.6 -How do you know when the recipe swaps elements?................................
9.1.7 -How do you know when the list is sorted?...
9.1.8 -How many times does the recipe pass through the list?............................
9.1.9 -What order is the list sorted into?..
9.1.10 - How does the shape of the list progress?...

9.3 - Algorithm 3.

You will now hear a different algorithm that also sorts a list of numbers but in a different manner. This algorithm splits the list into two parts by moving a pivot into its correct position, so that items to the pivot's left are smaller than the pivot, and the items to the right are bigger. The algorithm is then called recursively on each of the sub-lists until the list is eventually fully sorted. As before, we will examine and manipulate our list. As well as these two techniques, listen for the addition of a wooden block that counts how many times the recipe is passing through the list to sort it. The entire recipe, which is shown below, will be played three times in a row.

Example of Algorithm 3. **(Q)**

Step 1 - Examine the list.	7 1 8 3 6 4 2 5 3 7
(Pivot is mid point of list, pivot is 5)	
Step 2 - Pass through the list manipulating as we go.	
Sort list into sub-list around the pivot.	[{1 3 4 2 3 }{ 7 8 6 5 7}]
(LHS takes < pivot, RHS takes >= pivot)	^
	pivot (p)=5
Step 3 - Examine the new list	1 3 4 2 3 7 8 6 5 7
(Sub-list pivots are 3 for Left list and 7 for Right list)	
Step 4 - Pass through the list manipulating as we go.	
Sort into smaller sub-lists around pivots.	[{1 2}{3 4 3}] [{6 5}{7 8 7}]
Left list pivot around 3, Right list pivot around 7	^ ^
	p=3 p=7
Step 5 - Examine the new list	1 2 3 4 3 6 5 7 8 7
(Sub list pivots are now 2, 4, 6 and 8)	
Step 6 – Pass through the list manipulating.	[{1}{2}] [{3 3}{4}] [{5}{6}] [{7 7}{8}]
Sort into smaller sub-lists around pivots	^ ^ ^ ^
Sub-list pivots are 2, 4, 6 and 8	p=2 p=4 p=6 p=8
Step 5 - Examine the new list	1 2 3 3 4 5 6 7 7 8
Step 6 - Successfully terminate and indicate that the list is sorted.	

You will now hear this recipe being used on another list, see how many of the following questions you can answer. As before the entire recipe from start to finish will be repeated three times in a row.

9.1.1 -How many numbers (elements) are there in the list?................................

9.1.2 –What value is the pivot in the first pass?...

9.1.3 –What are the sub-list sizes on the first pass?..

9.1.4 - How many pivots are there on the final pass?...............................

9.1.5 –What identifies element placement into sub-lists?...............................

9.1.6 – How is the pivot musically represented?...

9.1.7 -How do you know when the list is sorted?...

9.1.8 -How many times does the recipe pass through the list?............................

9.1.9 -What order is the list sorted into?...

9.1.10 - How does the shape of the list progress?..

9.4 - Algorithm 4.

You will now hear a different algorithm that also sorts a list of numbers but in a different manner. This recipe uses **Algorithm 3** on the first pass to split the list into two sub lists, it then uses **Algorithm 1** on the left-hand sub list and **Algorithm 2** on the right-hand sub list. As before, we will examine and manipulate our list. As well as these two techniques, listen for the addition of a wooden block that counts how many times the recipe is passing through the list to sort it. The entire recipe, which is shown below, will be played three times in a row.

Example of Algorithm 4. **(BIO)**

Step 1 - Examine the list.	8 4 7 3 5 2 6 1
(Pivot is mid point of list, pivot is 5)	^
Step 2 - Pass through the list manipulating as we go.	
Sort list into sub-list around the pivot.	[{4 3 2 1}{8 7 5 6}]
(LHS takes < pivot, RHS takes >= pivot)	^
	pivot (p)=5
Step 3 - Examine the new list	4 3 2 1 8 7 5 6
Step 4 - Pass through the list manipulating as we go.	{4 3 2 1 } {8 7 5 6}
Alg 1 swaps the 4 for the 3, then the 2 and then 1.	^ ^ ^ ^ ^ ^
Alg 2 swaps the 8 for the 5.	
Step 5 - Examine the new list	3 2 1 4 5 7 8 6
Step 6 – Pass through the list manipulating as we go.	{3 2 1 4} {5 7 8 6}
Alg 1 swaps the 3 for the 2, then the 1.	^ ^ ^ ^ ^
Alg 2 swaps the 7 for the 6.	
Step 7 – Examine the new list	2 1 3 4 5 6 8 7
Step 8 – Pass through the list manipulating as we go.	{2 1 3 4} {5 6 8 7}
Alg 1 swaps the 2 for the 1.	^ ^ ^ ^
Alg 2 swaps the 8 for the 7.	
Step 9 – Examine the new list	1 2 3 4 5 6 7 8
Step 6 - Successfully terminate and indicate that the list is sorted.	

You will now hear this recipe being used on another list, see how many of the following questions you can answer. As before the entire recipe from start to finish will be repeated three times in a row.

9.1.1 -How many numbers (elements) are there in the list?...............................
9.1.2 –What value is the pivot in the first pass?...
9.1.3 –What are the sub-list sizes on the first pass?.......................................
9.1.4 - How many swaps are there in the 2nd pass?.......................................
9.1.5 –After 1st pass, what denotes swapping?...
9.1.6 – How is the pivot musically represented?..
9.1.7 -How do you know when the list is sorted?..
9.1.8 -How many times does the recipe pass through the list?...............................
9.1.9 -What order is the list sorted into?..
9.1.10 - How does the shape of the list progress?...

9.5 - Algorithm 5.

You will now hear a different algorithm that also sorts a list of numbers but in a different manner. This recipe uses **Algorithm 3** on the first pass to split the list into two sub lists, it then uses **Algorithm 2** on the left-hand sub list and **Algorithm 1** on the right-hand sub list. As before, we will examine and manipulate our list. As well as these two techniques, listen for the addition of a wooden block that counts how many times the recipe is passing through the list to sort it. The entire recipe, which is shown below, will be played three times in a row.

Example of Algorithm 5. **(BOI)**

Step 1 - Examine the list.	8 4 7 3 6 1 5 2
(Pivot is mid point of list, pivot is 5)	^
Step 2 - Pass through the list manipulating as we go.	
Sort list into sub-list around the pivot.	[{4 3 1 2}{8 7 6 5}]
(LHS takes < pivot, RHS takes >= pivot)	^
	pivot (p)=5
Step 3 - Examine the new list	4 3 1 2 8 7 6 5
Step 4 - Pass through the list manipulating as we go.	{4 3 1 2 } {8 7 6 5}
Alg 2 swaps the 4 for the 1.	^ ^ ^ ^ ^ ^
Alg 1 swaps the 8 for the 7, then 6, then 5.	
Step 5 - Examine the new list	1 3 4 2 7 6 5 8
Step 6 – Pass through the list manipulating as we go.	{1 3 4 2} {7 6 5 8}
Alg 2 swaps the 3 for the 2.	^ ^ ^ ^ ^
Alg 1 swaps the 7 for the 6, then the 5.	
Step 7 – Examine the new list	1 2 4 3 6 5 7 8
Step 8 – Pass through the list manipulating as we go.	{1 2 4 3} {6 5 7 8}
Alg 2 swaps the 4 for the 3.	^ ^ ^ ^
Alg 1 swaps the 6 for the 5.	
Step 9 – Examine the new list	1 2 3 4 5 6 7 8
Step 10 - Successfully terminate and indicate that the list is sorted.	

You will now hear this recipe being used on another list, see how many of the following questions you can answer. As before the entire recipe from start to finish will be repeated three times in a row.

9.1.1 -How many numbers (elements) are there in the list?.................................

9.1.2 –What value is the pivot in the first pass?...

9.1.3 –What are the sub-list sizes on the first pass?.......................................

9.1.4 - How many swaps are there in the 2nd pass?......................................

9.1.5 –After 1st pass, what denotes swapping?...

9.1.6 – How is the pivot musically represented?...

9.1.7 -How do you know when the list is sorted?..

9.1.8 -How many times does the recipe pass through the list?...........................

9.1.9 -What order is the list sorted into?..

9.1.10 - How does the shape of the list progress?...

9.6 - Algorithm 6.

You will now hear a different algorithm that also sorts a list of numbers but in a different manner. This recipe compares each location in the list with all other following elements, if out of place it swaps them until the correct element is placed at the current location. As before, we will examine and manipulate our list. As well as these two techniques, listen for the addition of a wooden block that counts how many times the recipe is passing through the list to sort it. The entire recipe, which is shown below, will be played three times in a row.

Example of Algorithm 6. (E)

```
Step 1 - Examine the list.                                          4 2 1 3
         (Compare current element to all following elements)       ^
Step 2 - Pass through the list manipulating as we go.
         On this step we swap the 4 for the 2,                      4 2 1 3
                                                                    ^ ^

         then we swap the 2 for the 1.                              2 4 1 3
                                                                    ^   ^

          we do not swap the current element for the 3 as we already have the smallest
         element in the right place                                 1 4 2 3

Step 3 - Examine the new list                                       1 4 2 3
          (Compare next current element to all following elements)    ^
Step 4 - Pass through the list manipulating as we go.
         On this step we swap the 4 for the 2,                      1 4 2 3
                                                                      ^ ^

          we do not swap the newly placed 2 for the 3 as 2 is in the correct place.
                                                                    1 2 4 3

Step 5 - Examine the new list                                       1 2 4 3
         (Compare next current element to all following elements)       ^
Step 6 – Pass through the list manipulating as we go.
        On this step we swap the 4 for the 3,                       1 2 4 3
        3 is now correctly placed, so is 4 by default.                  ^ ^
Step 7 – Examine the new list.                                      1 2 3 4
Step 8 - Successfully terminate and indicate that the list is sorted.
```

You will now hear this recipe being used on another list, see how many of the following questions you can answer. As before the entire recipe from start to finish will be repeated three times in a row.

9.1.1 -How many numbers (elements) are there in the list?................................

9.1.2 -How many swaps are there in the first pass?...

9.1.3 -How many swaps are there in the second pass?.....................................

9.1.4 -How many swaps are there in the third pass?...

9.1.5 -How do you know when elements are out of order?...................................

9.1.6 -How do you know when the recipe swaps elements?................................

9.1.7 -How do you know when the list is sorted?..

9.1.8 -How many times does the recipe pass through the list?..............................

9.1.9 -What order is the list sorted into?..

9.1.10 - How does the shape of the list progress?...

M. Experimental data.

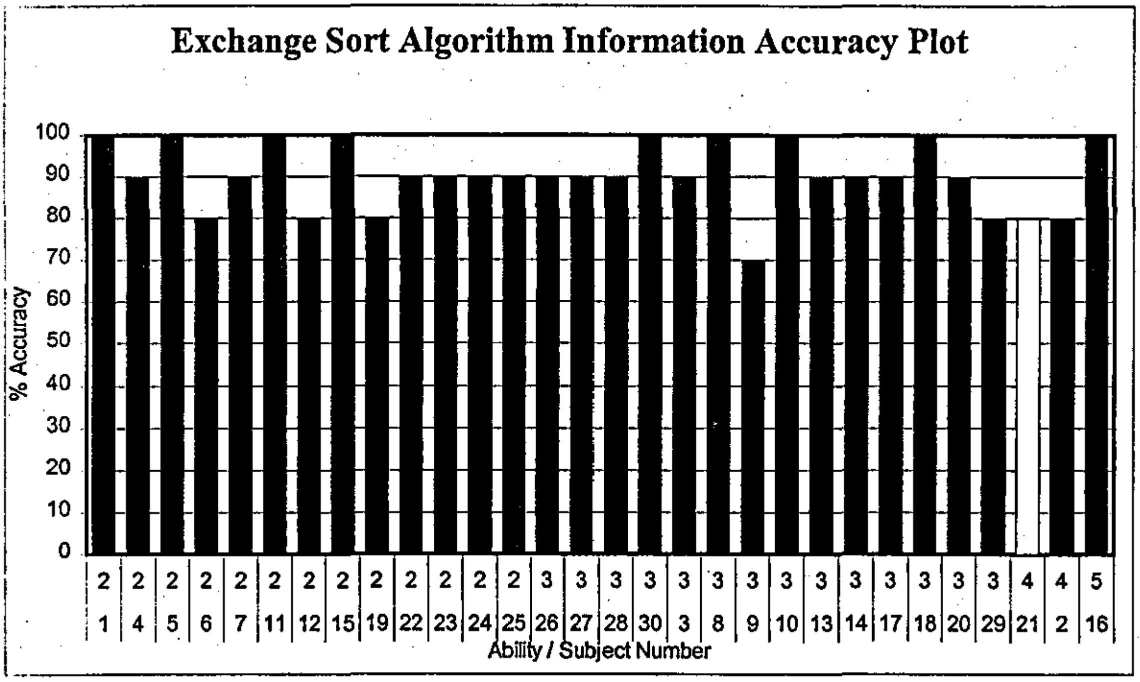


Figure M.1 – Exchange Sort information extraction accuracy.

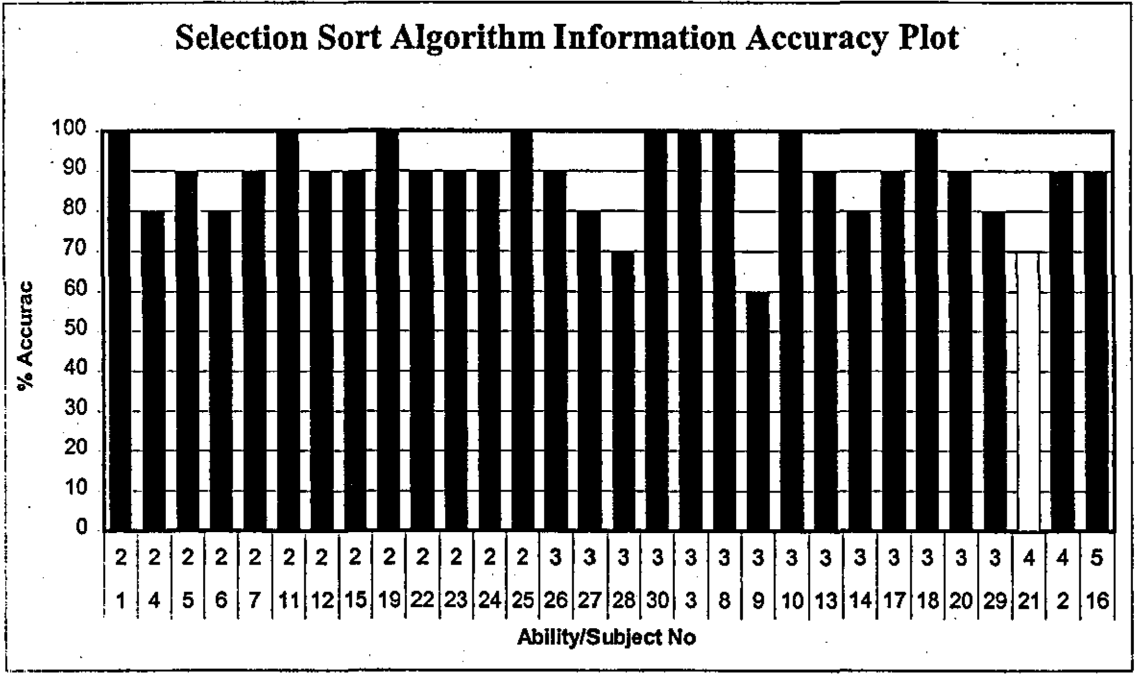


Figure M.2 - Selection Sort information extraction accuracy.

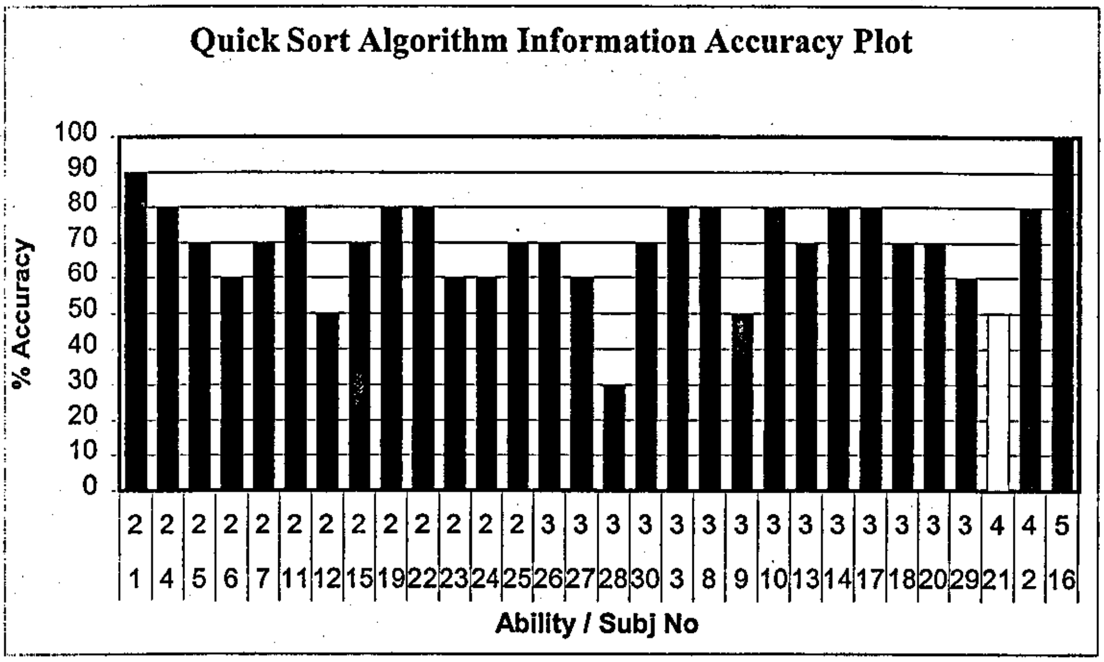


Figure M.3 - Quick Sort information extraction accuracy.

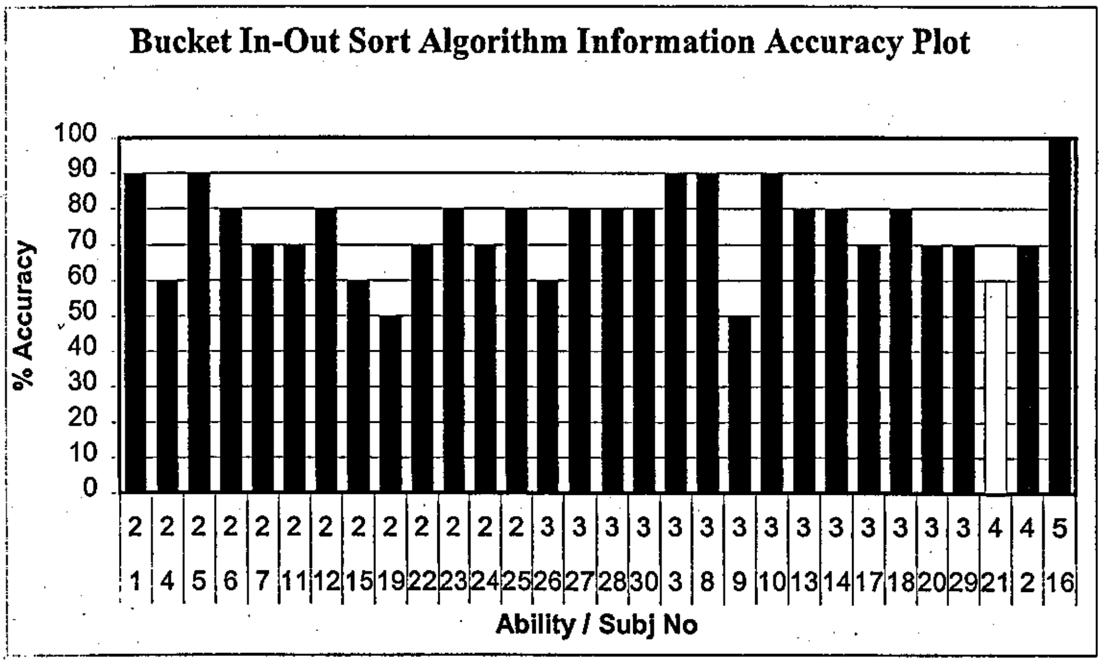


Figure M.4 – Bucket In-Out Sort information extraction accuracy.

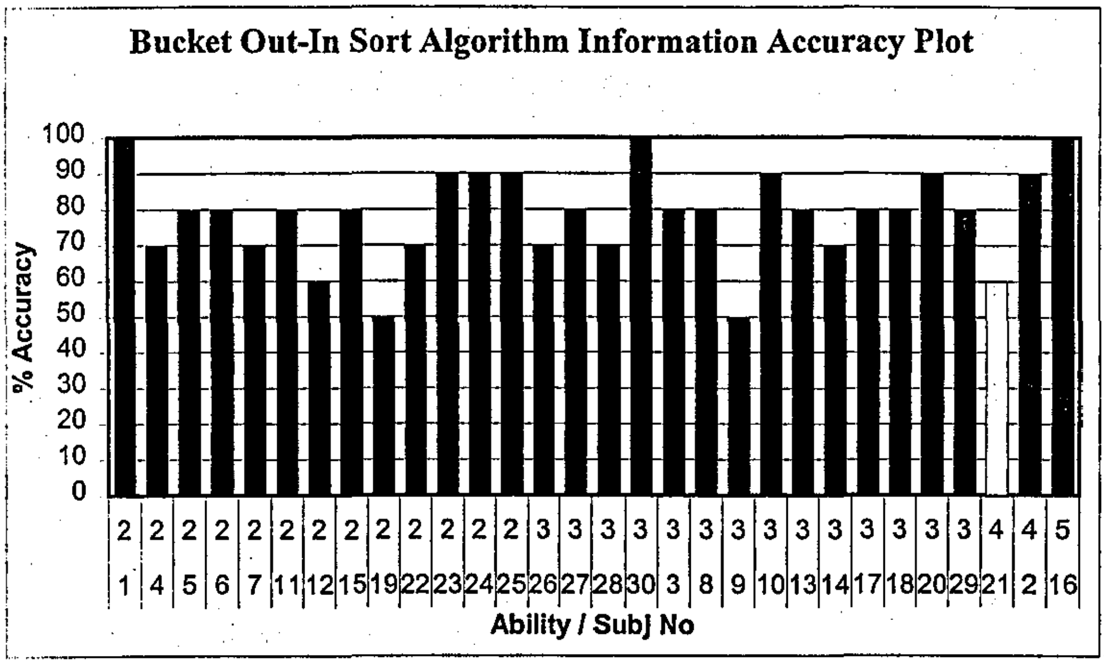


Figure M.5 – Bucket Out-In Sort information extraction accuracy.

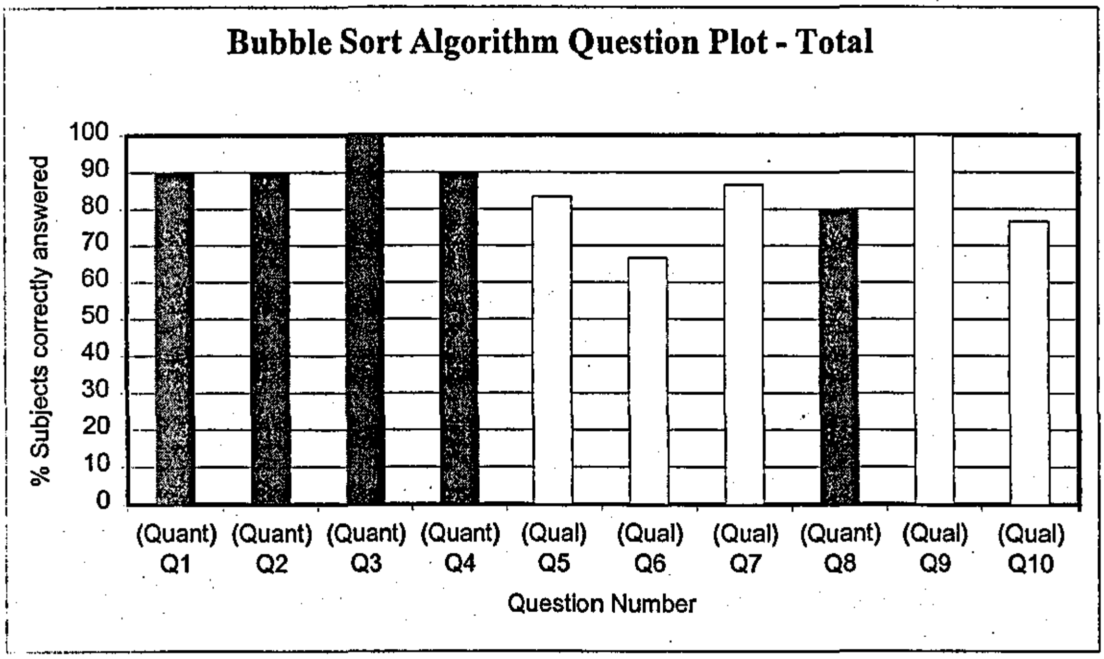


Figure M.6 – Bubble Sort information extraction accuracy by question type.

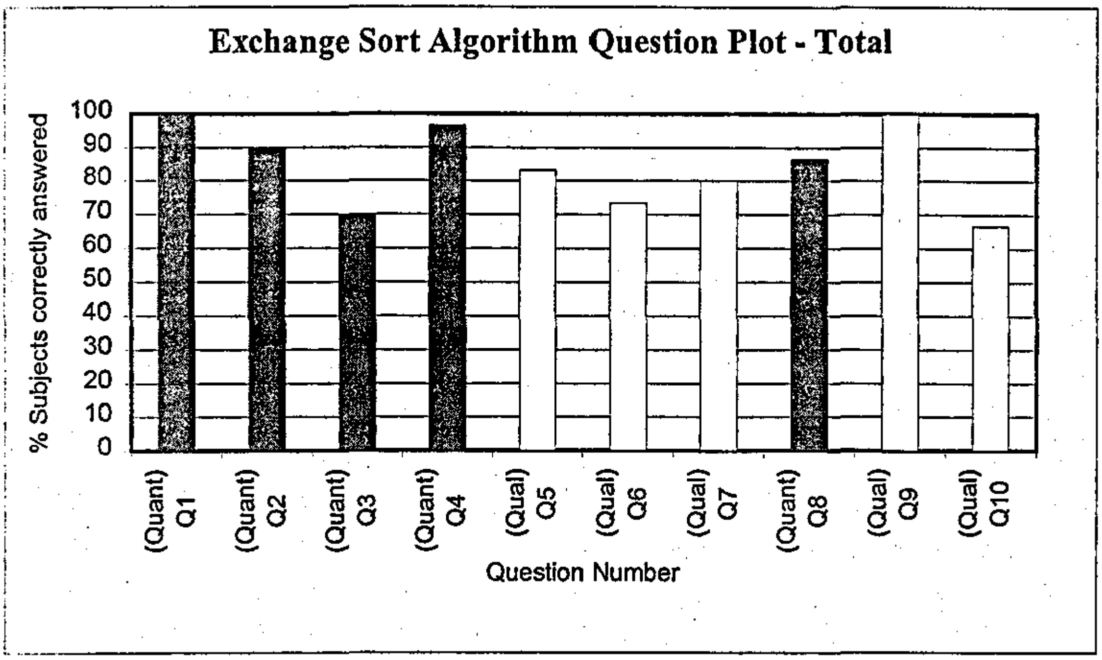


Figure M.7 – Exchange Sort information extraction accuracy by question type.

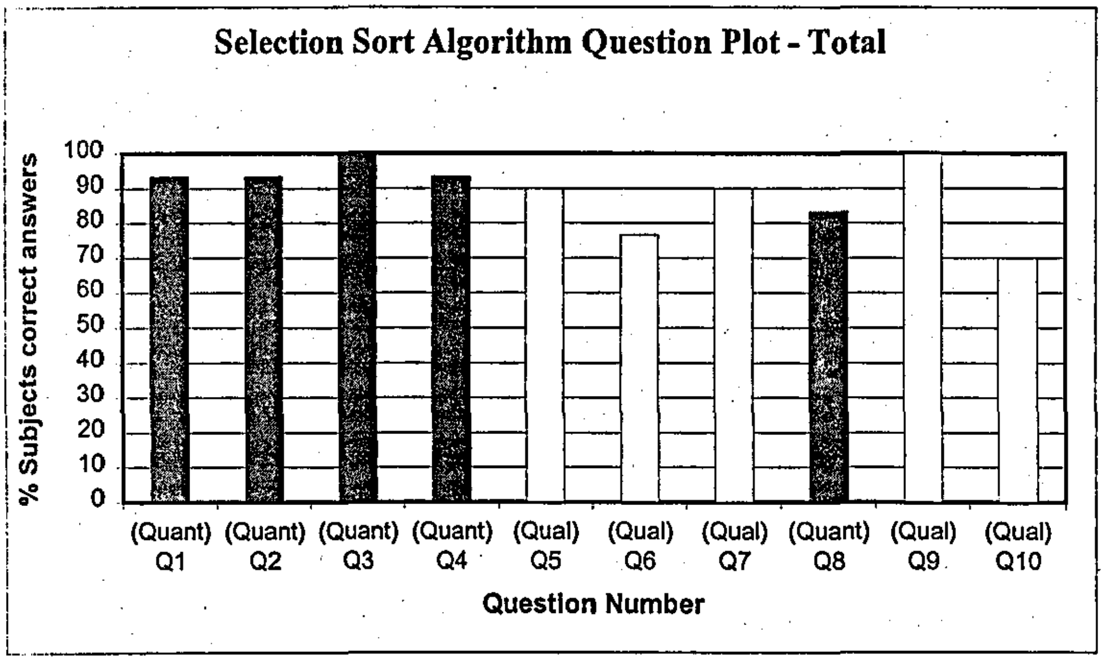


Figure M.8 – Selection Sort information extraction accuracy by question type.

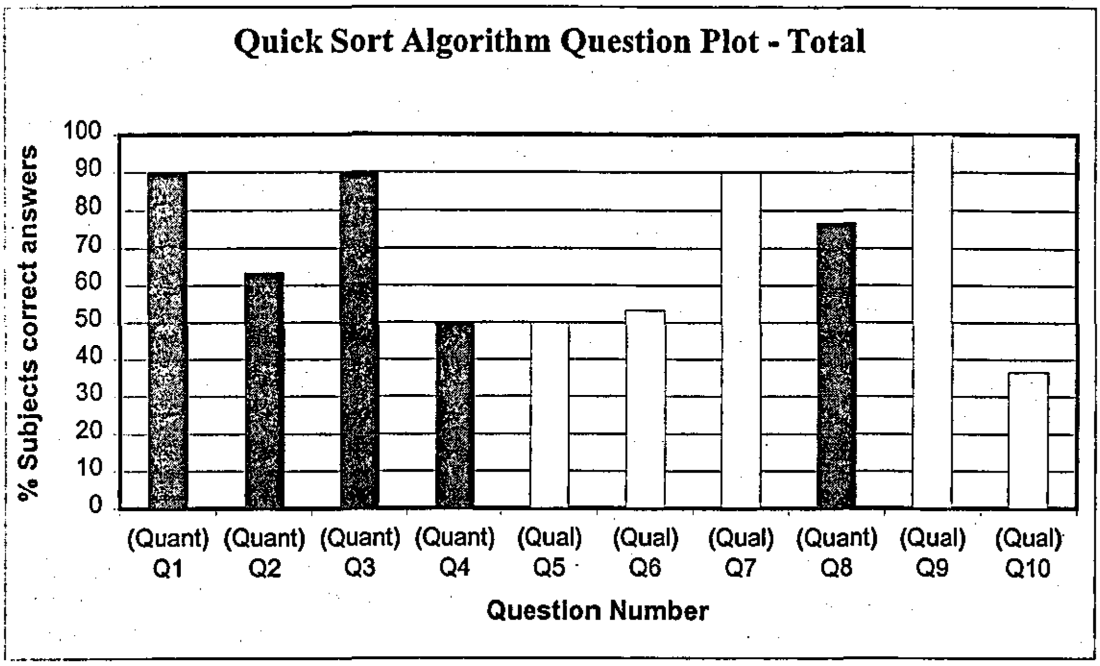


Figure M.9 – Quick Sort information extraction accuracy by question type.

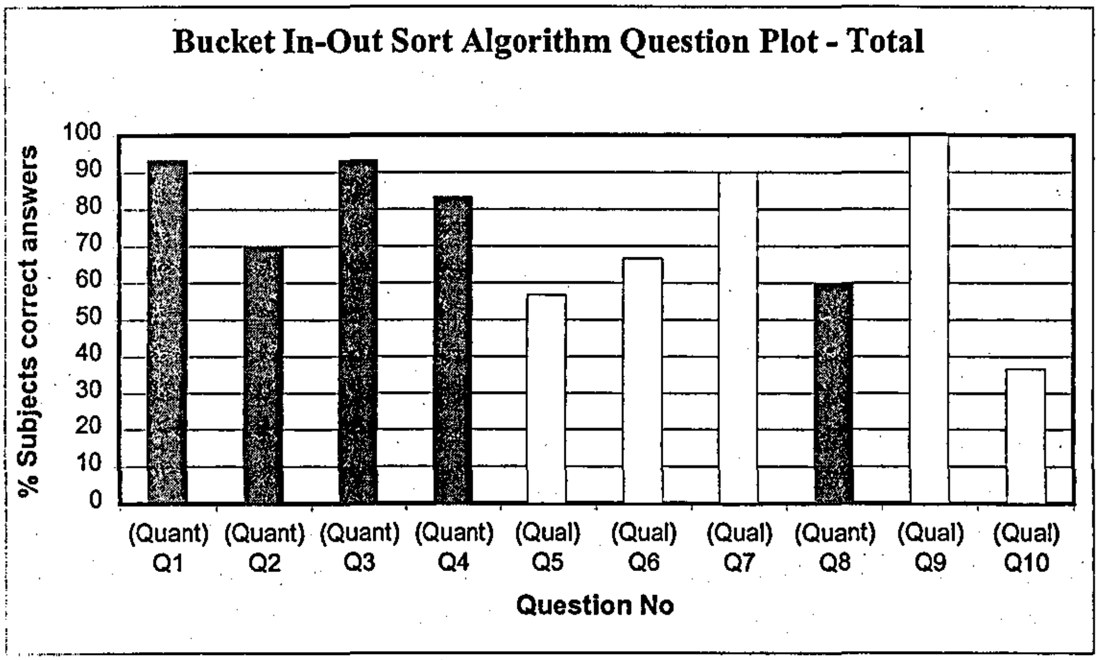


Figure M.10 – Bucket In-Out Sort information extraction accuracy by question type.

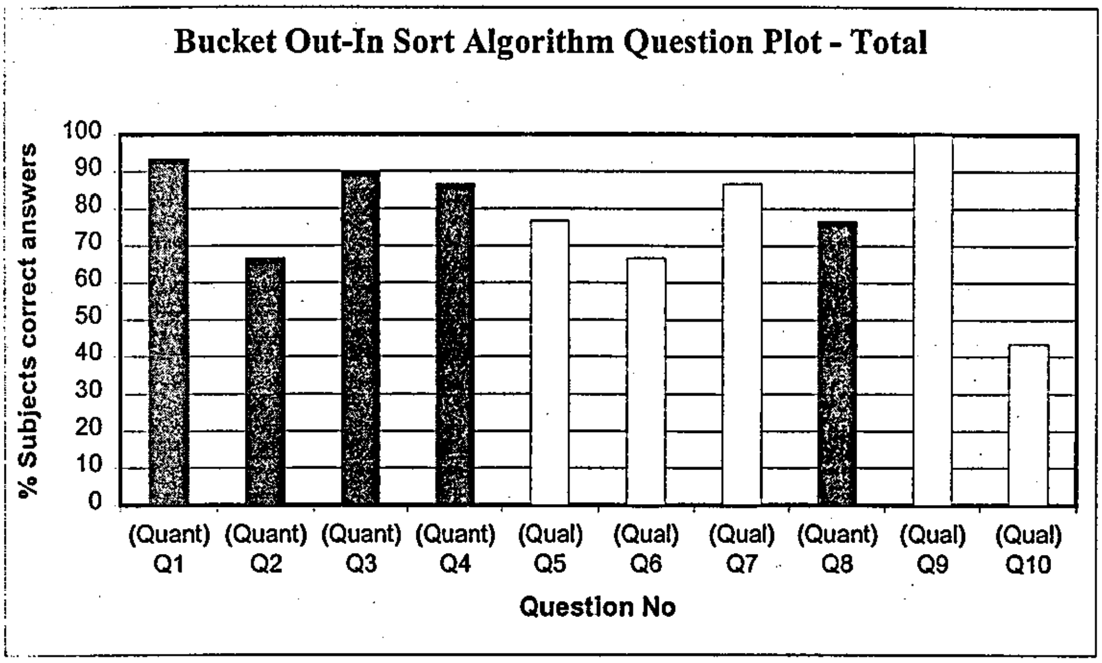


Figure M.11 – Bucket Out-In Sort information extraction accuracy by question type.

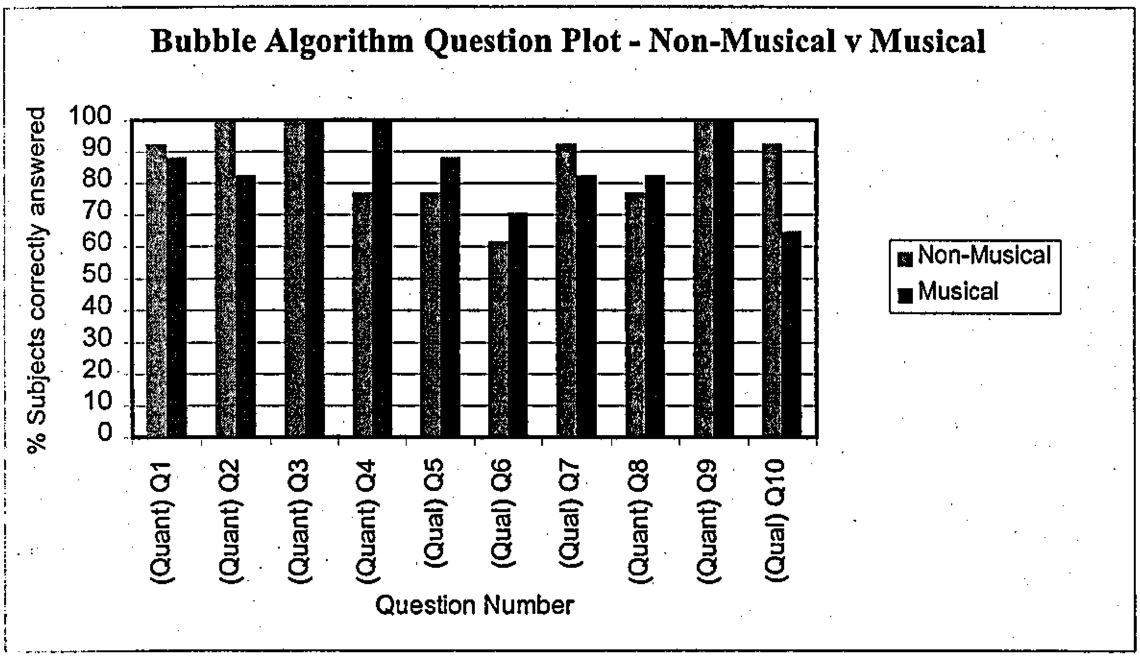


Figure M.12 – Bubble Sort information accuracy by question type, 'musical' v. 'non-musical'.

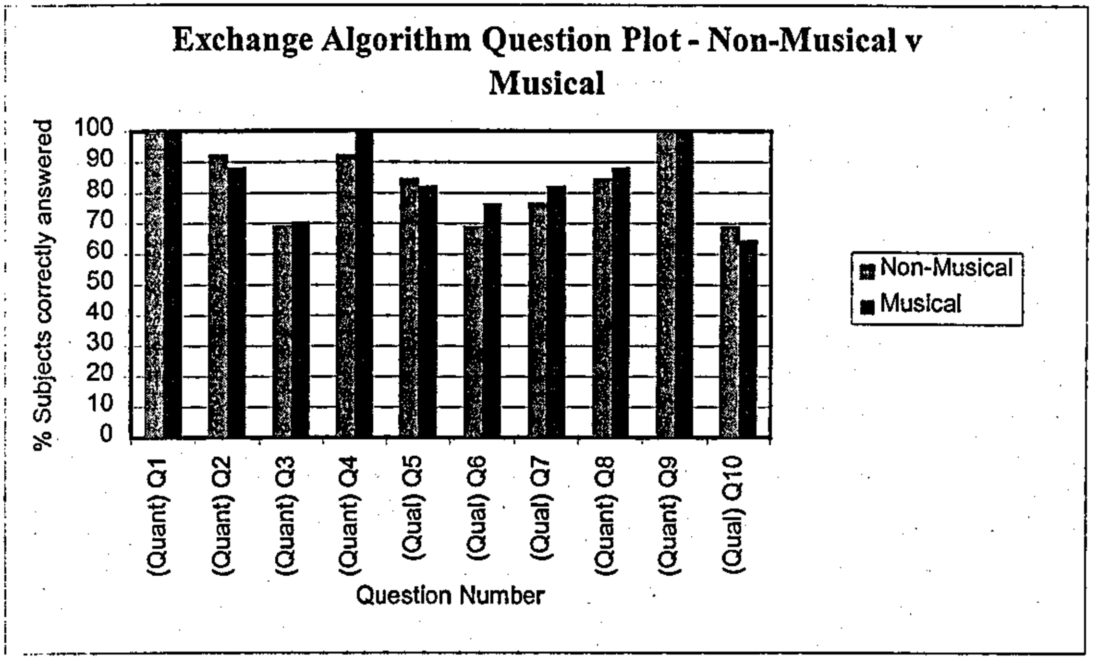


Figure M.13 – Exchange Sort information accuracy by question type, 'musical' v. 'non-musical'.

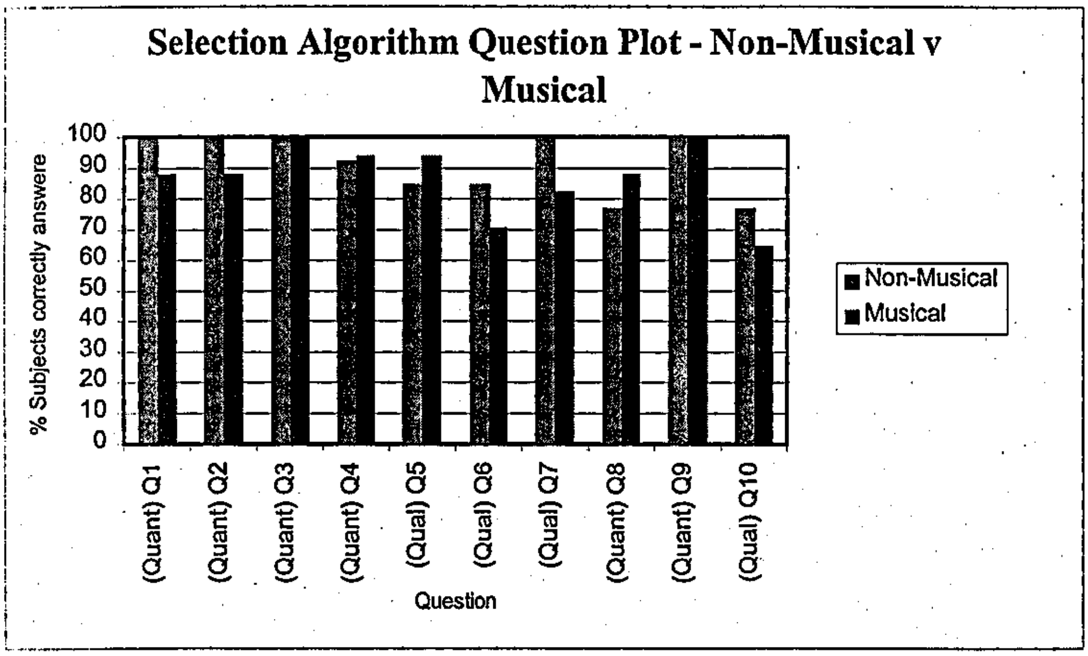


Figure M..14 – Selection Sort information accuracy by question type, 'musical' v. 'non-musical'.

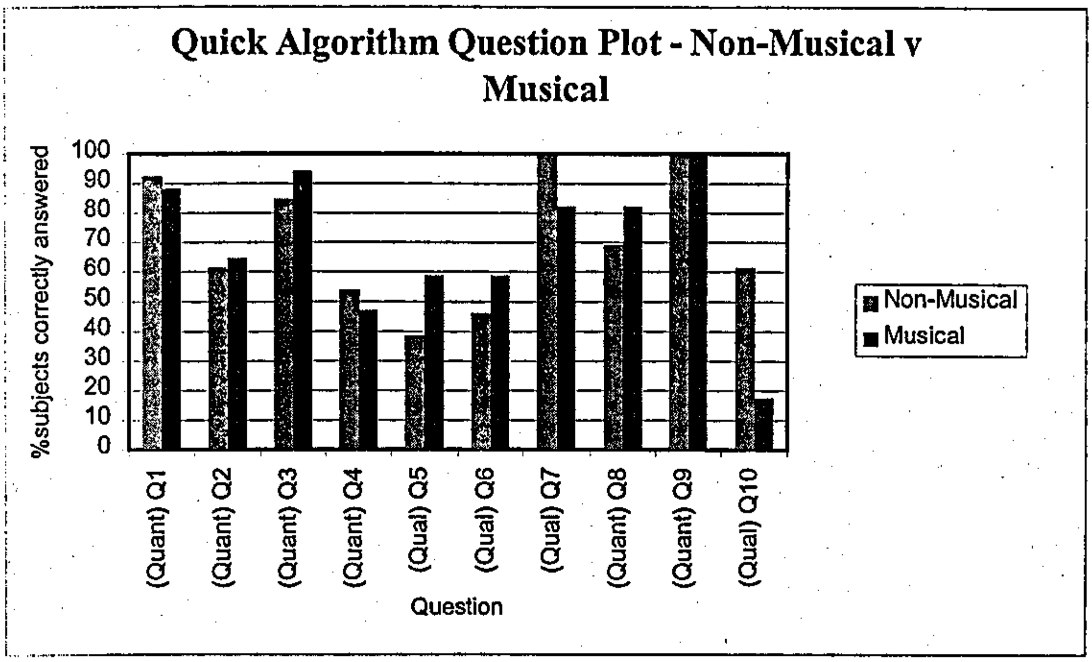


Figure M.15 – Quick Sort information accuracy by question type, 'musical' v. 'non-musical'.

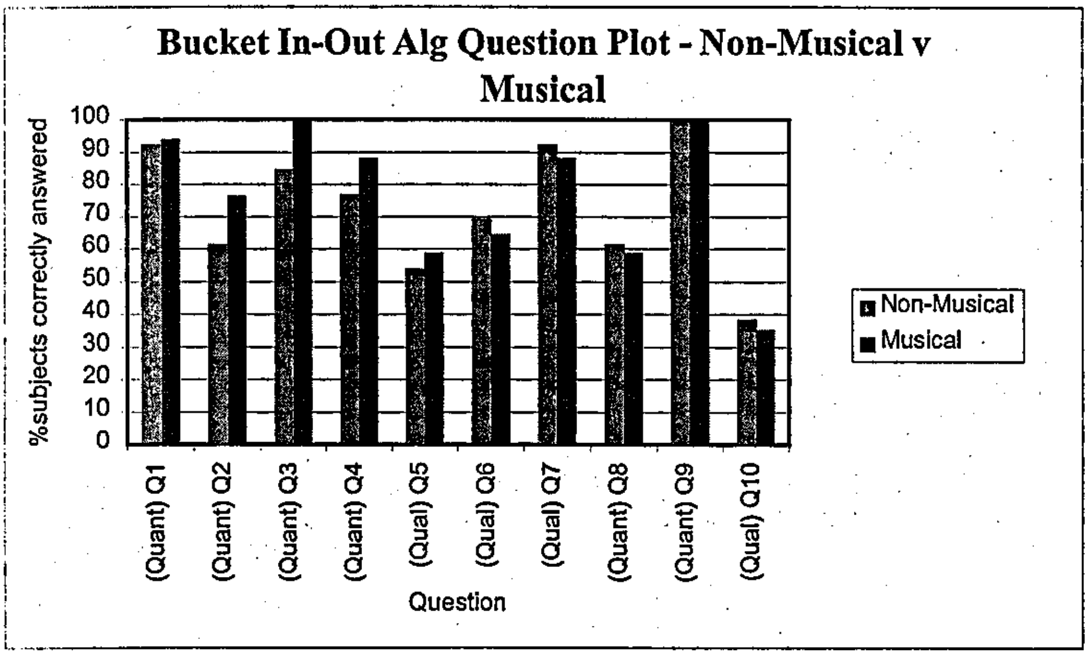


Figure M.16 – Bucket In-Out Sort information accuracy by question type, 'musical' v. 'non-musical'.

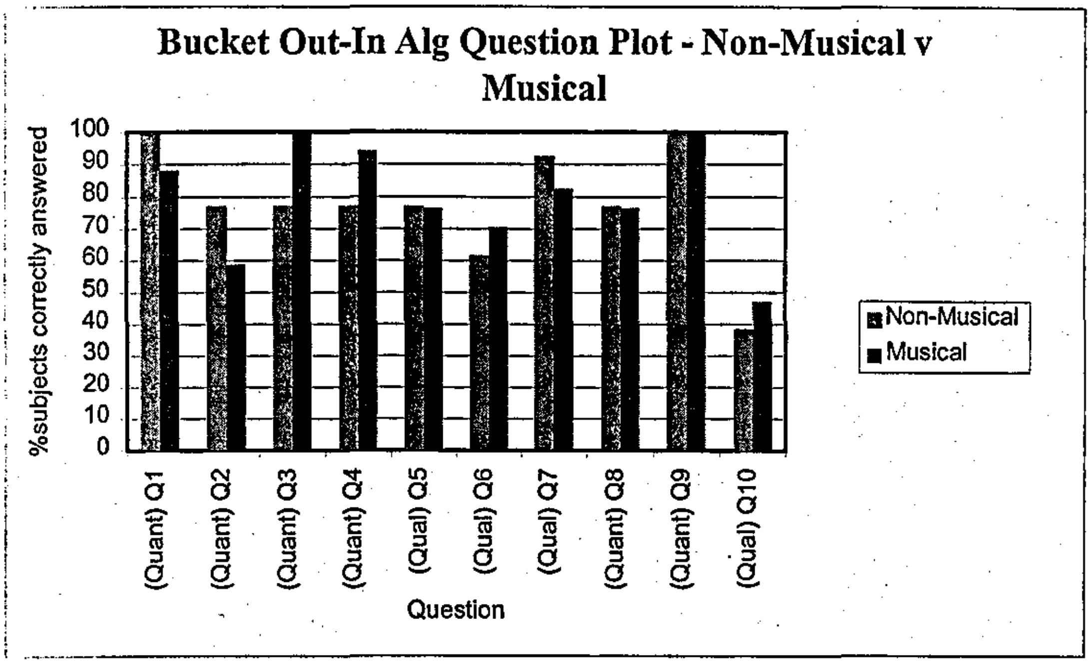


Figure M.17 – Bucket Out-In Sort information accuracy by question type, 'musical' v. 'non-musical'.

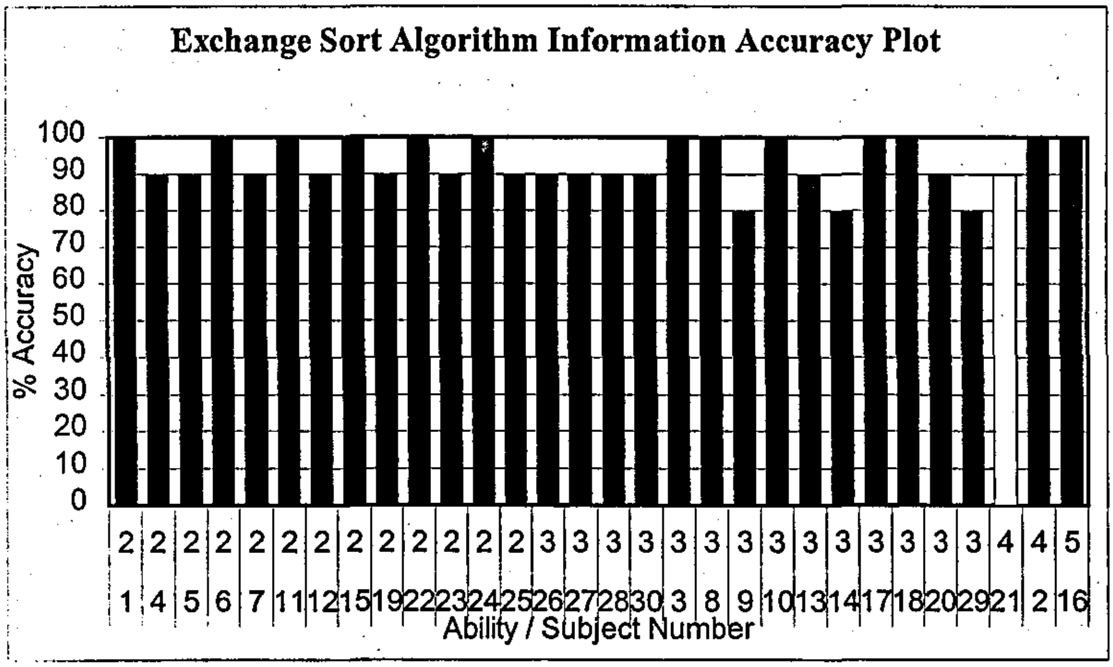


Figure M.18 – Exchange Sort information extraction accuracy.

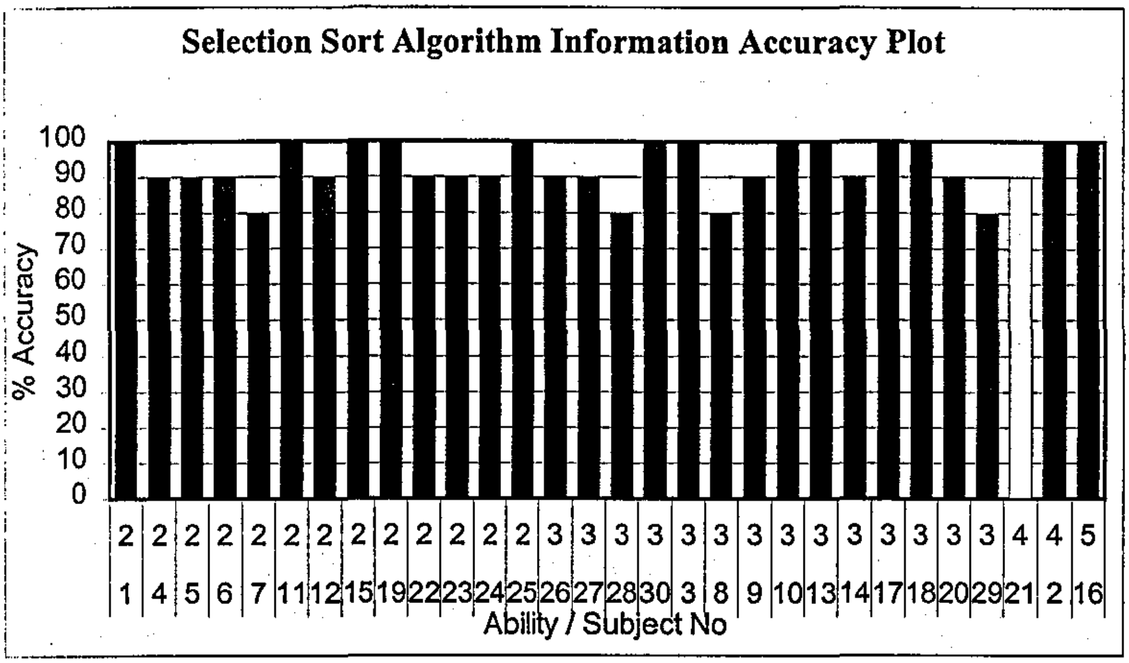


Figure M.19 - Selection Sort information extraction accuracy.

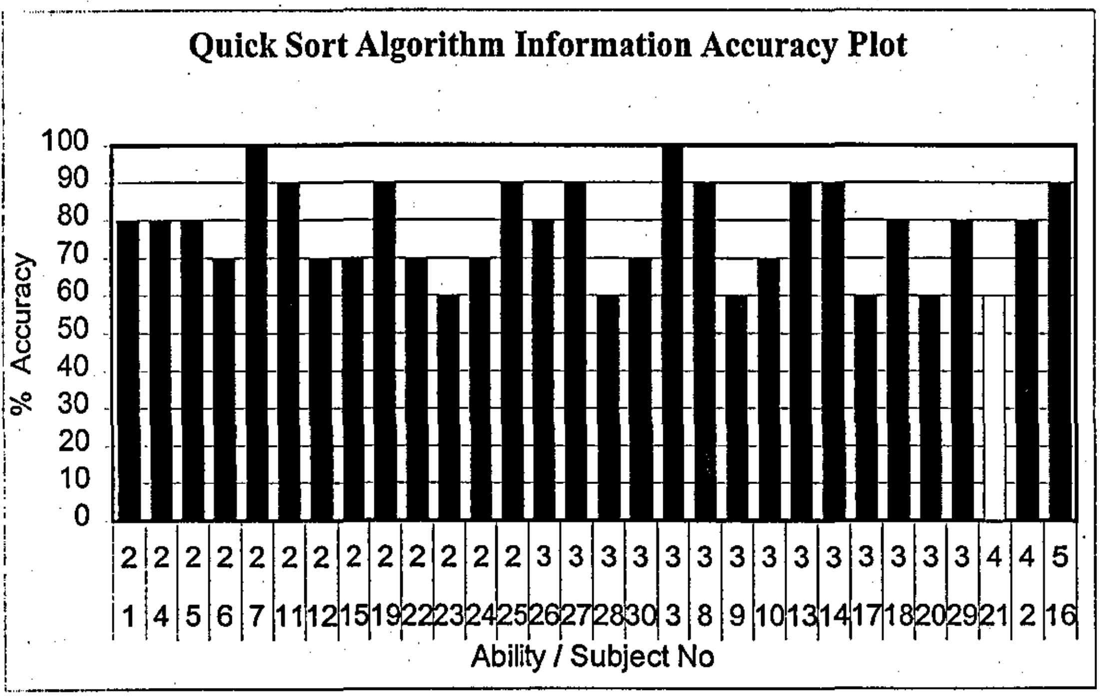


Figure M.20 - Quick Sort information extraction accuracy.

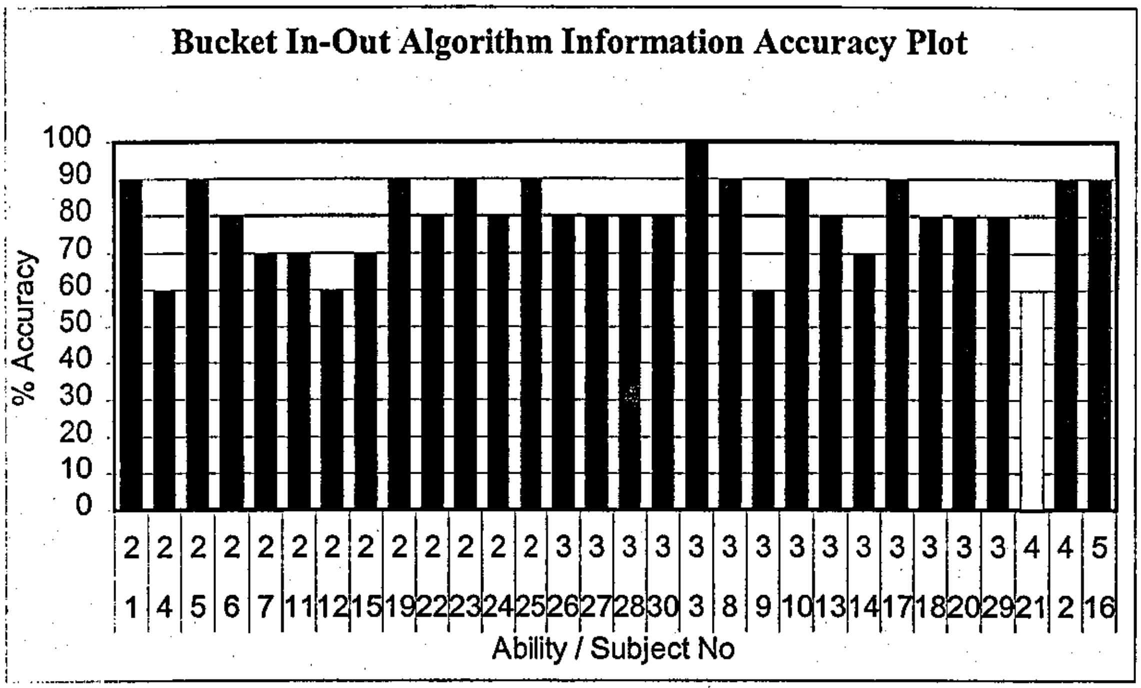


Figure M.21 – Bucket In-Out Sort information extraction accuracy.

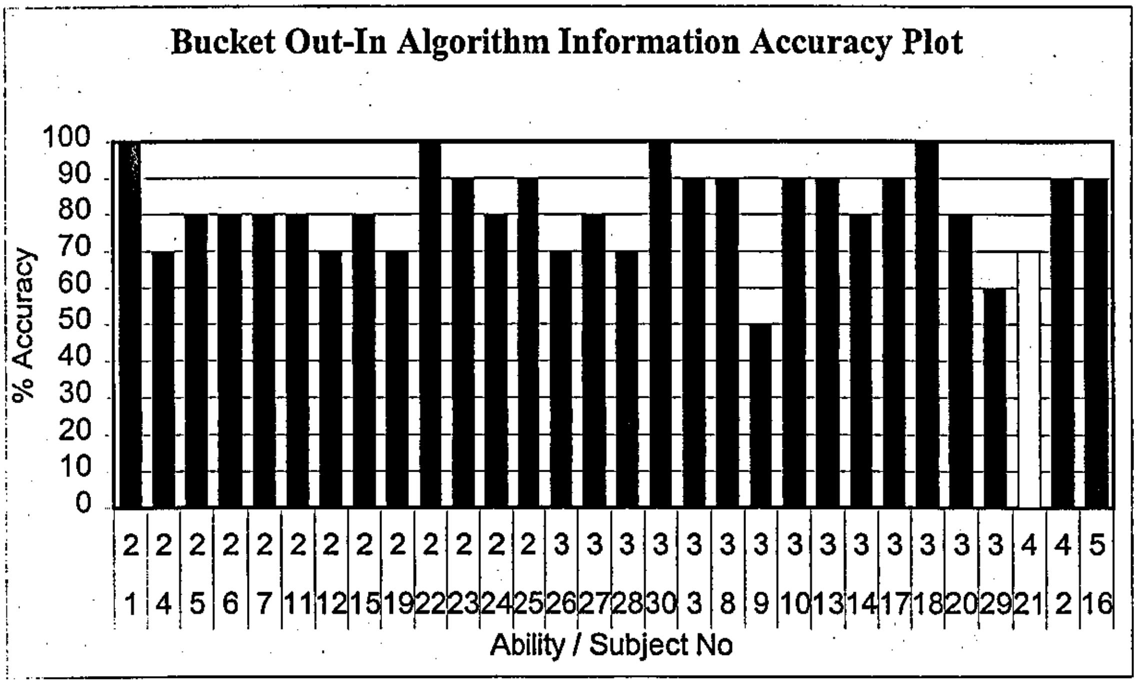


Figure M.22 – Bucket Out-In Sort information extraction accuracy.

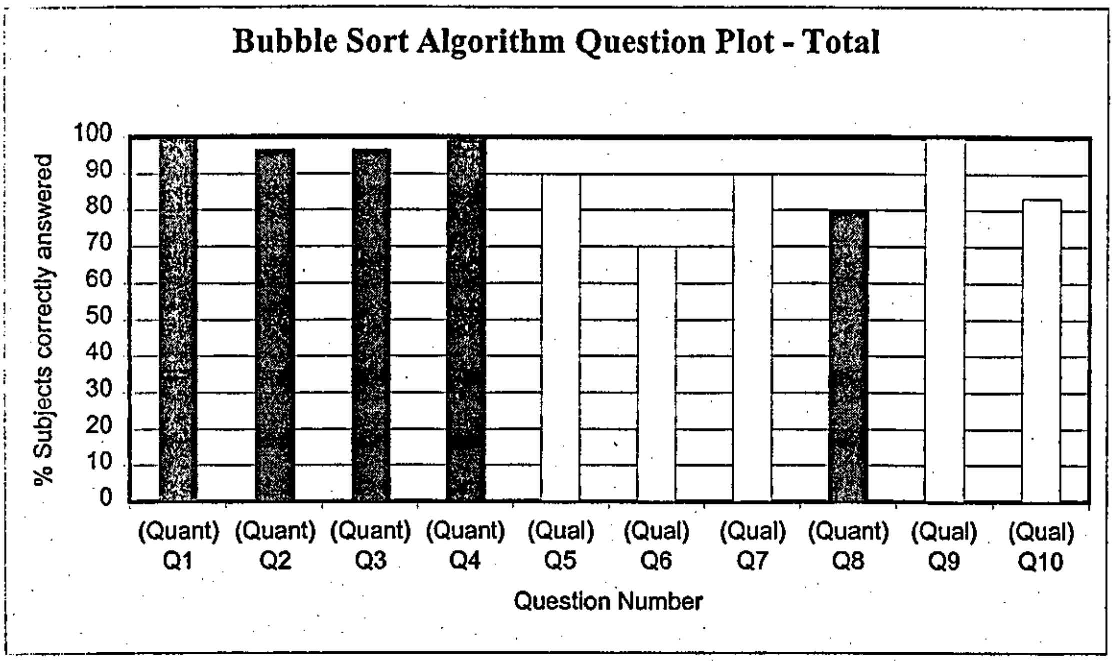


Figure M.23 – Bubble Sort information extraction accuracy by question type.

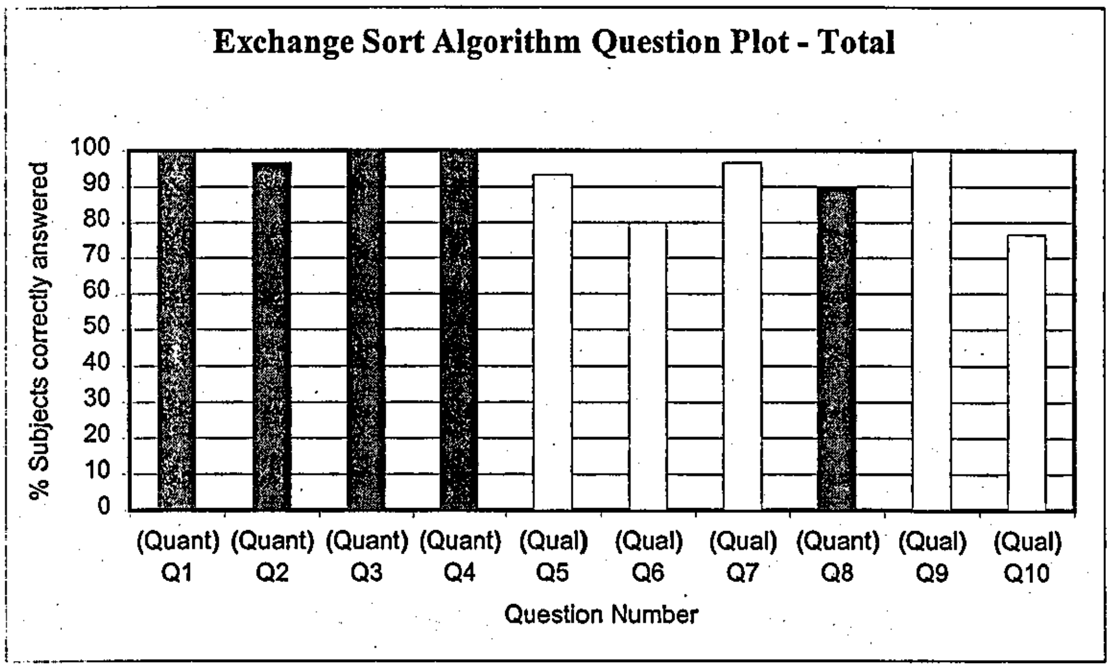


Figure M.24 – Exchange Sort information extraction accuracy by question type.

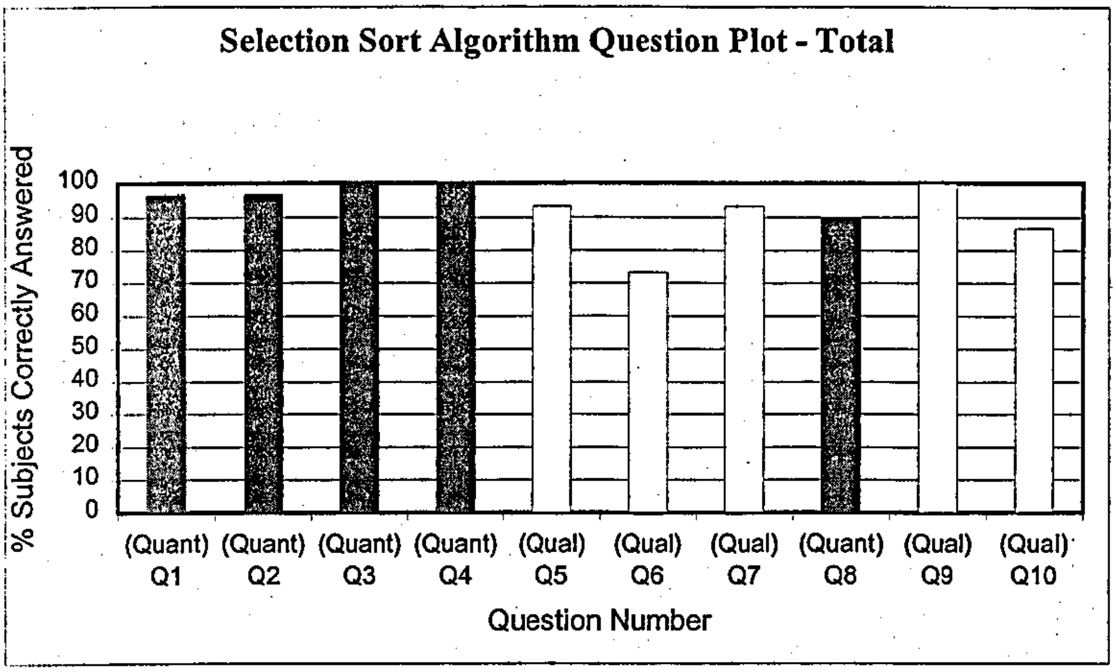


Figure M.25 – Selection Sort information extraction accuracy by question type.

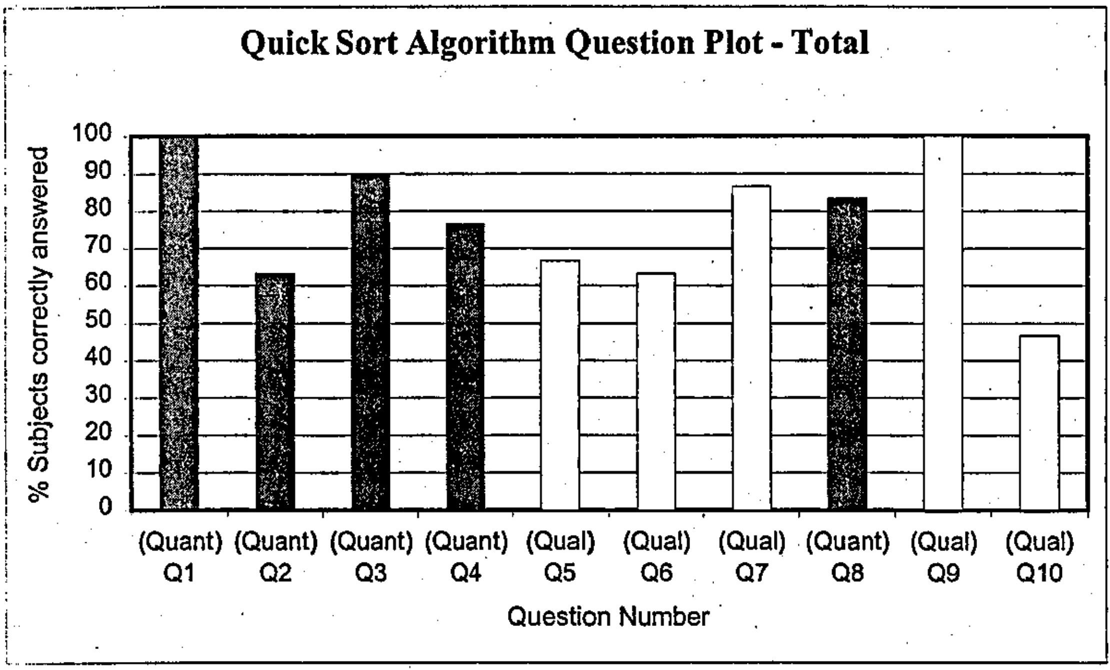


Figure M.26 – Quick Sort information extraction accuracy by question type.

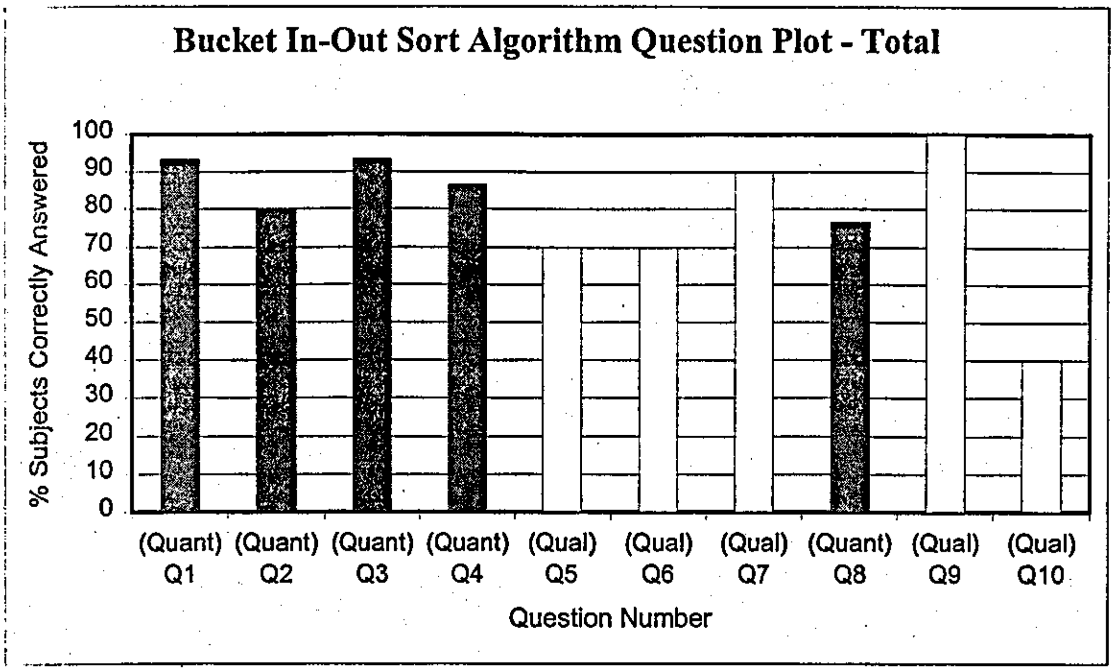


Figure M.27 – Bucket In-Out Sort information extraction accuracy by question type.

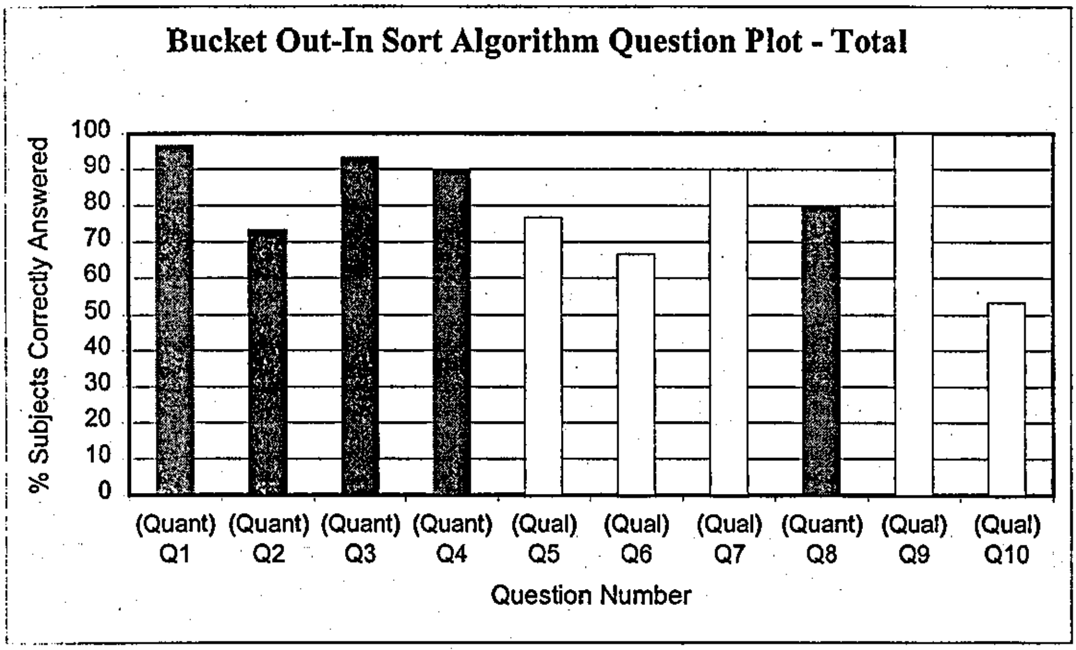


Figure M.28 – Bucket Out-In Sort information extraction accuracy by question type.

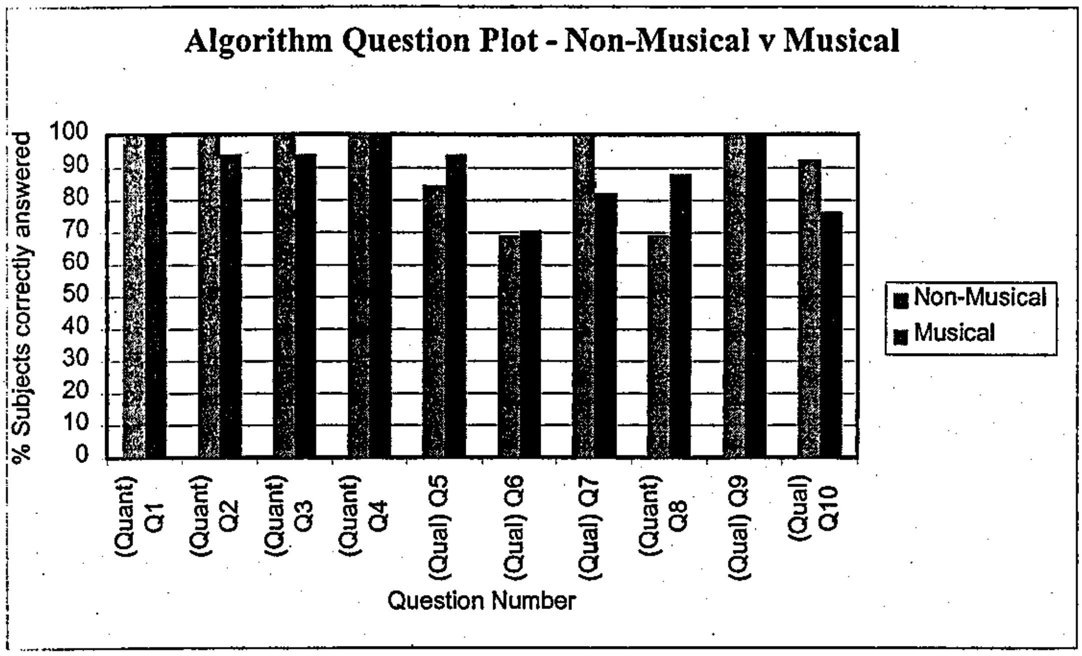


Figure M.29 – Bubble Sort info accuracy by question type, 'musical' v. 'non-musical'.

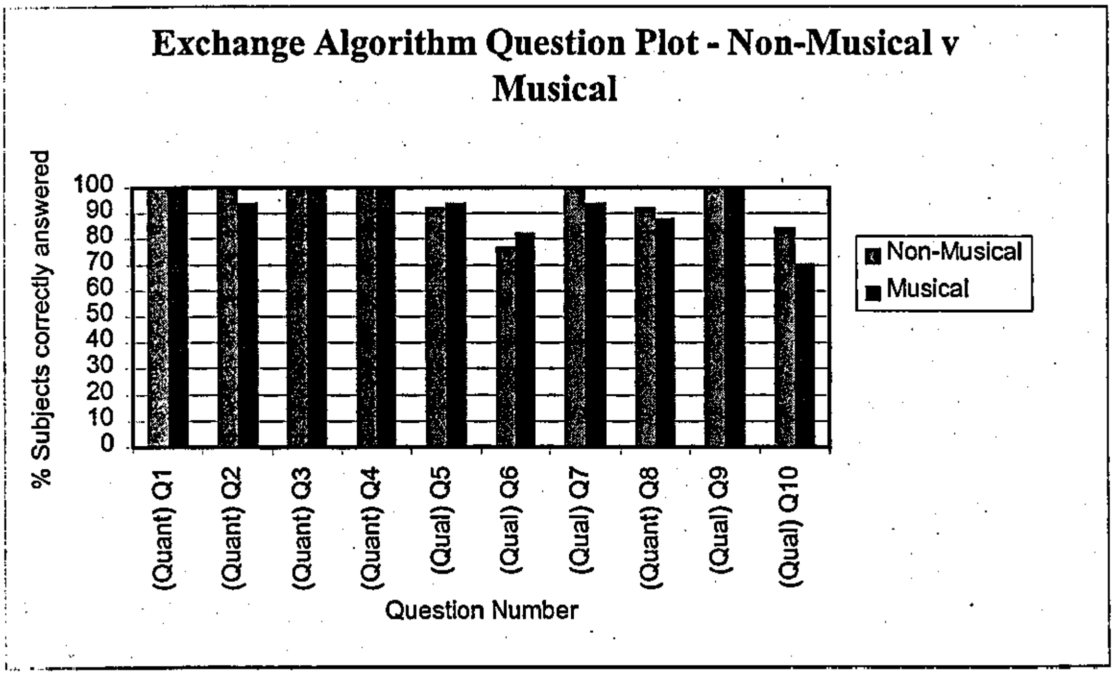


Figure M.30 – Exchange Sort info accuracy by question type, 'musical' v. 'non-musical'.

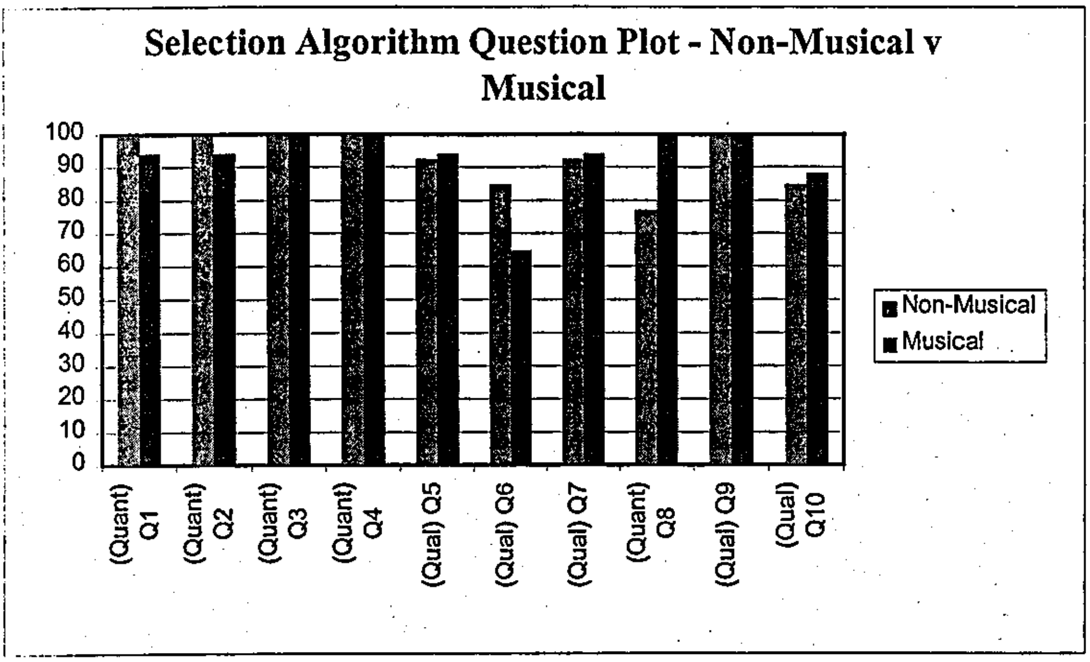


Figure M.31 – Selection Sort info accuracy by question type, 'musical' v. 'non-musical'.

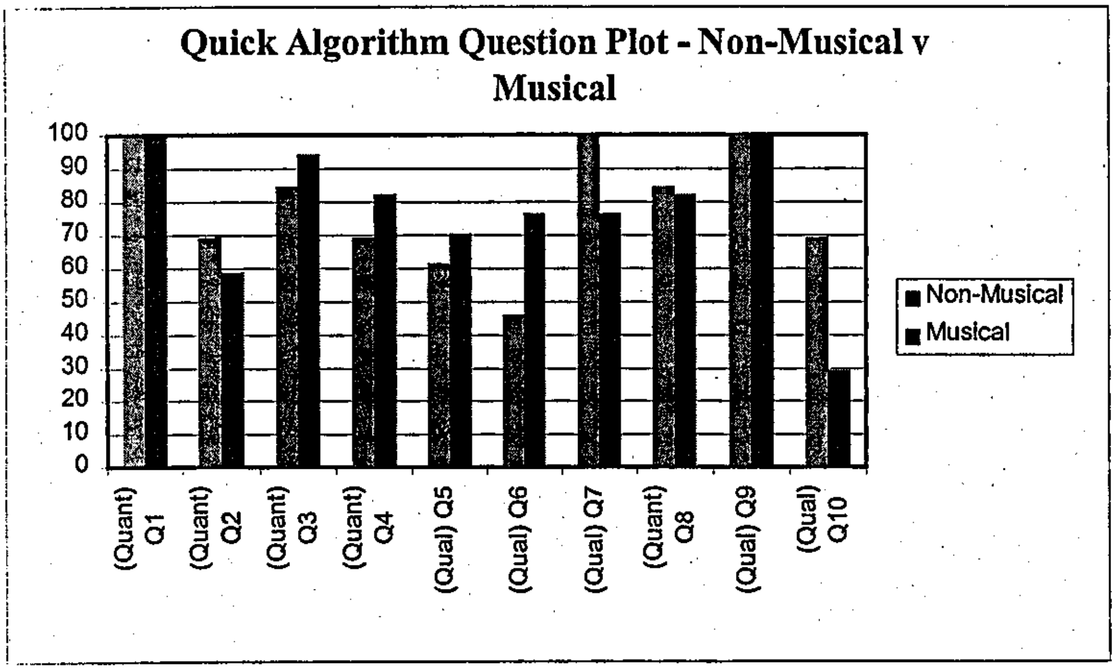


Figure M.32 – Quick Sort info accuracy by question type, 'musical' v. 'non-musical'.

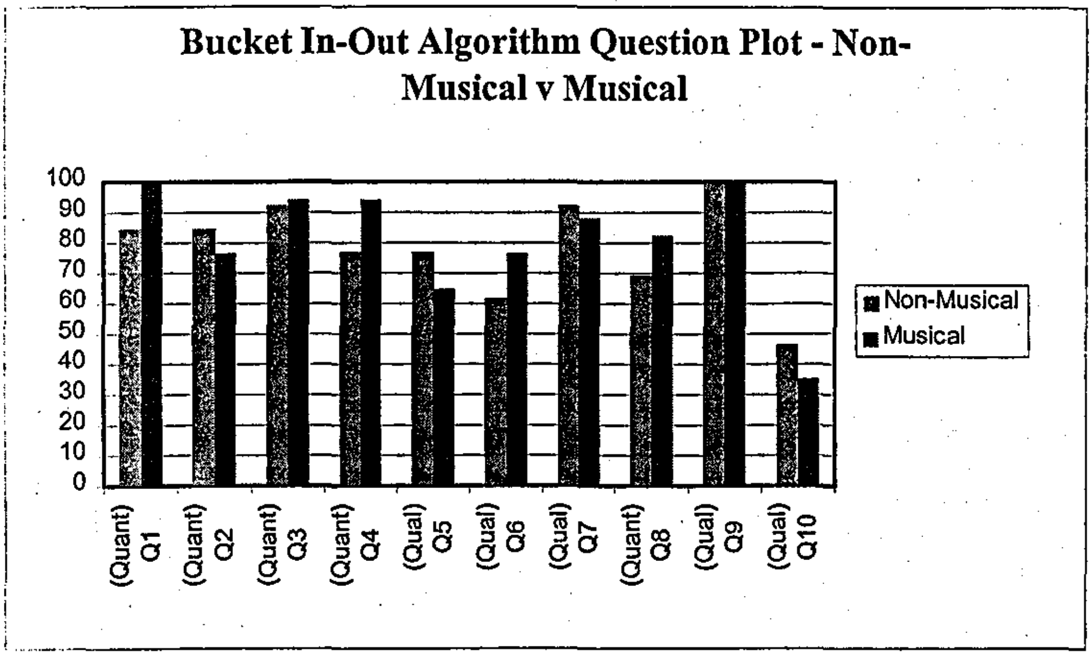


Figure M.33 – Bucket In-Out Sort info accuracy by question type, 'musical' v. 'non-musical'.

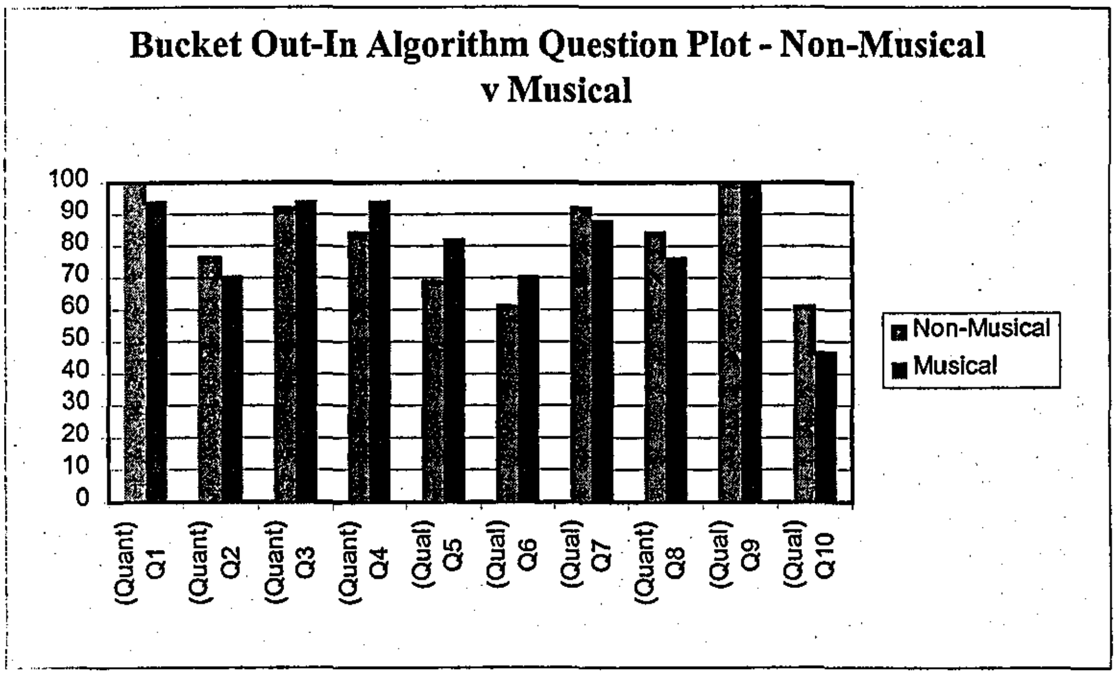


Figure M.34 – Bucket Out-In Sort info accuracy by question type, 'musical' v. 'non-musical'.