Sensory Augmentation for Navigation in Difficult Urban Environments by People With Visual Impairment

Thesis

UNRESTRICTED

Sensory Augmentation for Navigation in Difficult Urban Environments by People with Visual Impairment

Anthony Denis Johnston BEng (Hons)

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

of the Open University

December 2012

Department of Design, Development, Environment and Materials

Faculty of Mathematics, Computing and Technology

The Open University

Walton Hall

Milton Keynes

UK

DATE OF SUBMISSION: 21 DECEMBER 2012

DATE OF AWARD: 5 JUNE 2013

Abstract

Independent mobility in completing such tasks as walking through a town centre is taken for granted by well-bodied individuals. However, for those with a disability such as impairment of vision, mobility and navigation can become challenging tasks not easily undertaken. The barriers to access for blind and partially sighted individuals are increased when familiar navigational cues are removed in difficult urban environments such as Shared Space. The research consisted of investigating methods of navigation employed by people with visual impairment and designing a device to restore confidence to this group so as to lower the barriers of access to such environments.

Investigation was carried out through the deployment of a questionnaire; discussions with groups representing blind and partially sighted people; and a site visit to Shared Space environments. Statistical analysis was carried out on the results of the questionnaire to ascertain the navigational habits of blind and partially sighted individuals in different environments. From the analysis and the results of the discussions and site visit it was established that it would be socially acceptable to design a secondary aid to navigation that would complement the primary aids of long cane or guide dog. A concept experiment was carried out to test the idea that knowledge about changes in surface colour could help with navigation.

A prototype device that could be used by individuals with visual impairment to increase their confidence when navigating a difficult environment was designed, built and tested. Different programming methods were researched and trialled to effectively use machine vision to provide a solution to analyse video feed from a passive camera and return useful information to a blind or partially sighted user.

The device was tested indoors and outdoors and found to be effective at detecting changes in surface colour. Further work is needed to run the software on a more compact platform such as a mobile phone, but initial results show that the concept is viable and that the barriers that present to blind and partially sighted people navigating difficult urban environments can be much reduced through the use of this technology.

Acknowledgements

I have been privileged to have the help of many people in the research underlying the thesis and in the writing up of the final document. I want to thank them for their encouragement, help and support.

Particular thanks are due to

My supervisors Professor Jeff Johnson and Dr. Anthony Lucas-Smith who have given lots of encouragement and helped me formulate ideas as well as providing hospitality which is much appreciated. My 3rd party monitor, Dr Joe Rooney, who was there as a helpful support throughout.

Andrew Murdoch from Guide Dogs who listened to my ideas and made helpful suggestions. Indeed the initial idea of looking at Shared Space and the difficulty of crossing a road was his. He also organised the concept experiment with Eddie Chittick to whom my thanks are due as well.

Julie Polzerova from the Royal National Institute for the Blind who helped with the questionnaire distribution. Thanks to the members of both the RNIB and Guide Dogs who completed the questionnaire and some who also spoke to me on the phone.

Several researchers who have entered into correspondence with me about various aspects and given me helpful information including Peter Meijer, James Coughlan, Ioannis Katramados, Rose Johnson, Kurt Kaczmarek, Professor Paul Garthwaite who made suggestions regarding the statistics and others.

All the programs in this thesis were written by me but I am grateful to Jeremy Cowles as I adapted and extended part of a program he posted on Internet.

Finally, I am especially grateful to my wife and children who have let me "get on with it" and gave me encouragement to complete the project.

Contents

List of Figures

List of Tables

Chapter 1 Introduction: motivation and research aims

1.1 Overview

Independent mobility in completing such tasks as walking through a town centre is taken for granted by well-bodied individuals. However, for those with a disability such as impairment of vision, mobility and navigation can become challenging tasks not easily undertaken. An impetus to the research was the development of more difficult urban environments in the United Kingdom and Europe (and indeed further afield) which present additional problems to the blind and partially sighted pedestrian, over and above the ones he or she already has to deal with. Such areas present new challenges to navigation which traditional aids, such as the long cane or the assistance of a guide dog, are not able to provide solutions to. The research considered the case of a person with visual impairment who has to employ a range of skills not familiar to others, and who is reliant on cues in the environment which are less essential for those with good vision.

The overall aim of the research was to design and build a prototype device that used machine vision and computer technology to provide useful navigational information in difficult urban environments to a person with visual impairment. After a brief overview, this chapter provides an outline of the concepts involved and sets out the structure of the thesis.

The solution to the navigation problem considered in the research was the use of sensory substitution, also known as sensory augmentation. Sensory substitution is the use of one sensory modality, such as touch, to replace or augment another modality, such as sight, in an individual who has a deficiency or impairment in that sense (Bach-y-Rita and Kercel 2003). Sensory substitution devices have been designed using the increasing power of computers to help people with various sensory deficits to gain information about the environment around them, and so to complete tasks.

1.2 The problem of navigation in a difficult urban environment for people with visual impairment

Pedestrians make use of the many visual cues available for them as they navigate their way through an urban environment. People with visual impairment also employ cues, but they use non-visual ones such as the numerical position of a junction in relation to a starting point; traffic noise and the noise of people; change in gradient of the terrain they are walking on; the location of pedestrian crossings and other street furniture such as traffic lights. Nevertheless, barriers to navigation exist for such users simply because the urban environment is designed for visual users (Heylighen and Devlieger, 2007).

Recently urban designers and architects have been designing junctions and throughways in which many of the ordinary cues have been removed and vehicular traffic has been less segregated from pedestrians and other street users such as cyclists. Examples of such built environments incorporate 'Shared Space' and 'Home Zones' where familiar landmarks, such as pedestrian crossings and traffic lights, have been removed as part of a planned urban redesign. Other, temporary difficulties can arise, such as the partial blocking of pavements by objects not normally there, including rubbish bins outside a shop, or tables and chairs outside a cafe. The consequence of these changes and difficulties are that the barriers to the navigation of urban environments become so high that the person with visual impairment is unable to enter such a zone. They have lost confidence in their ability to reach a given destination within, or on the other side, of the difficult area. Anecdotal evidence and research carried out by 'Guide Dogs' (formerly the Guide Dog Association) indicates that blind people are avoiding difficult urban areas where vehicular and pedestrian traffic share the same surface, and traditional street furniture has been removed (Guide Dogs 2006a).

1.3 Requirements of a solution to the navigation problem

An essential part of helping people with visual impairment to navigate difficult urban environments is the restoration of a measure of confidence in their ability to find their way. This will enable them to be motivated to overcome the barriers to navigation. Restoration of confidence relies in turn on a system which has been proven to work in the environment.

The system should be able to provide a level of feedback to the user that can be interpreted in order to give him or her useful representation of the urban area. The system should not be difficult to operate, and the feedback should be such that it does not interfere with the user's ability to use his or her existing senses and abilities.

1.4 Sensory Augmentation

Historically various aids have been designed and built to help people with visual impairment, and more recently the use of machine vision has been explored with the aim of either providing information about the environment to the user, or of enabling an alternative pathway through the brain for information so that the visual cortex can be activated (Kupers and Ptito, 2011). The latter relies on brain plasticity, and in the seminal work by Bach-y-Rita a television camera was interfaced to a series of vibrating rods which were in direct contact with a subject's skin (Bach-y-Rita *et al.*, 1969). This device was able to instil in the subject a sensation akin to vision itself, with the result that the subject reported being able to perceive the object in the focus of the television camera as an object in a visual field, and to describe its visual attributes such as parallax and perspective.

1.5 The approach taken in the research

The research started by investigating the problem of navigation through difficult urban environments by people with visual impairment. This was followed by the design and building of a device which used machine vision to read a video feed of the environment from a simple camera. The video feed was processed by a computer and useful information extracted from it which was in turn fed back to a user.

One method of investigating the navigation problem further was through the use of a questionnaire which was distributed among blind and partially sighted volunteers, and which asked about the participants' experience of navigating urban environments with particular reference to more problematic situations. The questionnaire analysed the methods used by the participants to navigate, so that any aid or device designed and built as a result of the research could act in a complementary way to such methods. Alongside this a thought experiment was undertaken in which a person who was congenitally blind was led through a street, and information relayed to him to find out what data might be helpful in the task of navigating the street.

As a result of discussions with representatives of Guide Dogs (Northern Ireland) the design of a prototype device, which provides feedback on the nature of the surface over which the user is travelling, formed a large part of the research. Two different methods of analysing the surface using machine vision and associated software were investigated. The first involved the use of neural networks to classify surface types with the aim of indicating to the pedestrian when the surface type over which he or she was walking was about to change. The second compared the surface the user was standing on with one that was about to be encountered in the path just ahead. This second method was influenced by work on autonomous vehicles, and used the concept of colour spaces to extract useful information from the video feed.

During the design and building of the prototype, the system was tested to find its efficacy at analysing surface types, and a feedback mechanism involving haptics was designed, built and tested.

1.6 The research questions

A number of research questions were investigated during the course of the research. These included questions to do with the existing aids that a person with visual impairment might use, as well as specific questions about how a system could help with aspects of urban navigation, and more general questions about the efficacy of such a system. The questions are enumerated below.

1. Do existing assistive technologies form the basis of an urban environment navigational system that can tackle difficult urban environments such as Shared Space?
2. How can any other information about the urban environment be gathered easily using suitable technology and fed to a blind or partially sighted person in a helpful way?
3. Can a device be built which will be under the control of the user and which, with some training, will provide sufficient feedback to a blind or partially sighted user so as to increase their confidence when travelling through a difficult urban environment?
4. Can a neural network program classify surfaces, if appropriate pre-processing of images can lead to identifiable inputs, and will that classification be carried out at a sufficient speed and provide enough detail to be of use to a blind or partially sighted user?
5. Can intensity invariant, colour space components be used to compare the current surface on which the pedestrian is standing with the path ahead so as to give a warning of potential change in surface type or upcoming obstacle?
6. Would a person with a long cane or other assistive technology benefit from having information about the surface they are travelling along available as either tactile or audible feedback?
7. How can a device be constructed that will provide such feedback to the user?

8 Can the overall performance of a person with visual impairment making his or her way across the difficult urban environment be improved through knowing what surface they are on?

9 Can such a device indicate the angle at which the user was moving from one surface to another so that he or she could cross perpendicularly to the edge?

10 What constitutes a successful urban environment navigational system?

Questions 1-7 arose from study of the literature, and their derivation is explained in chapters 2 and 3. Question 9 arose out of discussion of the problem with a representative of Guide Dogs, formerly The Guide Dog Association, as explained in section 4.2. Questions 8 and 10 are more general questions that arose from consideration of the problem of navigation and conjecture about how to assess the success of the prototype device.

1.7 Outline of the thesis

The chapters that follow describe the research undertaken, the tackling of the research questions, and the results and conclusions reached.

Chapter 2 explores in more detail the problem of difficult urban environments, and looks at the history of Shared Space along with Home Zones and other urban architectural concepts. Concerns expressed by groups representing people with disability are explained, and a recent report on Shared Space by the Department for Transport is analysed.

Chapter 3 is a more extensive literature review reporting on literature pertaining to the research. The topics of vision substitution history and brain plasticity; neural networks; autonomous vehicle obstacle avoidance; human computer interfaces; and mobility, orientation and aids used by partially sighted and blind individuals are examined.

Chapter 4 discusses the methodology of the research including a section on the questionnaire; the testing of the concept; and the design of the prototype.

The analysis of the questionnaire, including a statistical investigation of the results, is described in chapter 5 along with a description of the concept experiment carried out.

The vision subsystem and the Human Computer Interface (HCI) of the prototype are discussed in some detail in chapters 6 and 7. These chapters include sections of software code and an explanation of how the code was designed and tested.

Chapter 8 provides a report on the experiments carried out using the prototype device, both inside the George Rzevski Complexity Laboratory on the Open University campus, and outside in the town of Holywood, Co Down.

Chapter 9 brings the thesis to a conclusion by looking at the extent to which the research questions have been answered; investigating further work that might be done on the basis of this research; and exploring the further questions that have been produced.

Chapter 2 Difficult Urban Environments

2.1 Introduction

As explained in chapter 1, this thesis explains the development of a prototype device that uses machine vision and computer technology to provide useful navigational information in difficult urban environments for blind and partially sighted users. This chapter explores the problem of difficult urban environments in some depth, looking at recent developments in urban planning and concerns expressed by groups representing those with disability.

The idea of developing an aid to help with navigation is not novel; as is discussed later, primary and secondary aids exist which help blind and partially sighted individuals to navigate the environment (section 3.7). The purpose of any aid is to increase the confidence of the user so as to decrease the barriers to access. Work has been done for other groups of disabled people to produce aids to help them. For example, Matthews *et al.* (2002) and Beale *et al.* (2006) have explored how accessibility maps can be created for wheelchair users using a Geographical Information Systems network model. The needs of deaf people in urban environments have been explored at Gallaudet University in a project known as the DeafSpace project (Byrd, 2007).

People with visual impairment make use of items in the urban landscape to help them navigate their way through the built environment. Such items include pedestrian crossings with traffic lights where a blind person familiar with a certain street may feel confident that he or she can cross at a certain known point. If the crossing is removed along with some of the other items, the urban landscape becomes more difficult for the person with visual impairment to find their way around.

This chapter looks briefly at normal navigational cues and aids, such as the pedestrian crossing mentioned above, and this is followed by an exploration of some urban landscapes where these navigational aids are lacking. To finish with, there is a comparison of two recent reports which principally deal with recommendations for the design of Shared Space streets. One report is written by Guide Dogs (formerly The Guide Dogs for the Blind Association) and the other produced on behalf of the Department for Transport.

2.2 Navigational Cues and Aids in Urban Environments

Finding one's way around a busy town centre or through an unknown village on foot is an ability that is mostly taken for granted. Navigation is carried out through the use of familiar signs, signals and artefacts while pavements, zebra-crossings and traffic lights incorporating pedestrian signals all help the individual to find his or her way around, mostly subconsciously. Many of these navigational cues and aids are visual, but for a person with visual-impairment there are other aids which fulfil the same function. These include tactile pavements; audio warning or turning rods associated with traffic lights; the edge of pavements; and the noise of traffic, other pedestrians and echoes from building surfaces, to name a few. As will be seen later, there are, in fact, a wide variety of non-visual cues that visually impaired people use to navigate around streets and outdoor spaces (table 5.1 lists many non-visual cues asked about in the questionnaire).

In areas where familiar objects and navigational cues are not present, it is much more difficult to find one's way around. This is highlighted by the need for training for people undertaking navigation at sea, or in other unfamiliar environments such as mountainous terrains. Without such training individuals typically feel vulnerable and at risk, and lack confidence unless accompanied by a guide. In a similar way if normal navigational cues, such as traffic lights, are removed from an urban environment, or if sounds are distorted and unclear, then it becomes more difficult for people with visual impairment to find their way, and they feel an increased sense of vulnerability. Areas where such navigational aids are often lacking include pedestrian precincts in towns and cities; large indoor shopping centres, where sounds are distorted; and new urban development environments, for example 'Home Zones' and 'Shared Spaces'.

2.3 Shopping Centres

Typical shopping centres consist of wide floor surfaces made of a sound-reflective material and a high ceiling or roof. People walk in both directions across the surface, but there is no clear flow of people and no consistent speed, so some people may cross the concourse to see something of interest, while others will drift from one side to the other, or simply stop to look at something. Frequently, there are shopping trolleys, push-chairs, and 'buggies', together with groups of people who have stopped to talk, often in the middle of the concourse. Shops line either side of the concourse, much as in a street, but sounds are distorted through echoes from the ground and roof surfaces. Furthermore, to make use of the available space, some kiosks or small shops are located in the middle of the concourse. With a lack of clear order, it is very difficult for a person with visual impairment to find their way, or to successfully navigate from one end of the shopping complex to the other.

A questionnaire distributed to members of the RNIB and Guide Dogs Northern Ireland (formerly The Guide Dog Association, Northern Ireland) asked if participants had comments on the experience of moving through an area with fewer navigational cues or aids. One respondent stated that she found "large open parks and indoor shopping centres difficult" and cited the fact that her GPS system would not work indoors. Another problem she had was that inside shopping centres sounds are masked and more difficult to distinguish (questionnaire respondent 28, appendix E).

The issue of using sound to help with navigation is an important one when considering a vision substitution or sensory augmentation system. Since users may well be relying heavily on the sense of hearing to gain orientation in such areas, it is important that this sensory gateway should be kept as clear as possible.

Nevertheless visiting a shopping centre is popular among blind and partially sighted people – research shows it to be the second most popular activity outside the home (Crosier 2009). In fact the RNIB reports that shopping is the main reason for 7 out of 10 blind and partially sighted people to leave their home. However, they also report that 50% of such people suffer huge anxiety at the thought of going out and may rarely do so independently (RNIB, 2009a).

2.4 Shared Spaces

Figure 2.1 Photograph of Shared Space in Drachten, The Netherlands

In a new type of urban development, named 'Shared Space', many of the sensory aids familiar to people with visual impairment have been removed (Interreg IIIB, 2003). This is because the aim of Shared Space is to create an area where vehicles, bicycles and pedestrians interact more closely with each other, obliging the road users to take more care and be more aware of each other. As a direct consequence of slowing down and being more aware of their surroundings and others in the vicinity, road users are paradoxically safer (Hamilton-Baillie, 2005). However, this environment presents a problem for people with visual impairment who find they have much fewer navigational cues or aids to help them in their 'wayfinding' or navigation from one part of the urban environment to another. Examples of different Shared Spaces are seen in photographs 2.1 -2.3.

Figure 2.2 Photograph of Shared Space in Drachten, The Netherlands

Figure 2.3 Photograph of Shared Space Stranraer, Scotland (Farrington, A. 2011)

Shared Space is becoming a feature of urban architecture and represents a departure from traditional, segregated areas for traffic and pedestrians in city and town centres. In order to fulfil the aim of forcing users to be more aware of each other, and to create a community area where users interact, segregation of traffic users does not occur. In the words of Ben Hamilton-Baillie, an architect involved in Shared Space design, "at the heart of Shared Space is the concept of integration" which he contrasts with the idea of separation, where different users of the urban environment utilise different parts of the road and pavement (Hamilton-Baillie 2006). The removal of the clutter associated with many current urban spaces is intended not only to be more aesthetically pleasing, but also, paradoxically, to be safer by forcing different users to be more careful in their approach to the area as discussed above.

Shared Space schemes are attractive to policy makers as they are relatively inexpensive to implement; they aim to reduce congestion; and they provide an increase in road safety. Simultaneously, in regions where the Shared Space design has been realised local people have felt an increased sense of community.

The concept of Shared Space can be said to originate primarily in The Netherlands where a leading innovator was Hans Monderman who worked to see its implementation, not only in The Netherlands, but also in several other European countries. In early 2004 the European Interreg IIIB road traffic project was launched. In this project Shared Surface urban areas were set up in several countries including The Netherlands, Belgium, Germany, Denmark and Suffolk County Council, England. In terms of the success of the Shared Space project, it was reported that the subsequent junction and street designs did indeed become safer and more aesthetically pleasing, and in many cases congestion was reduced.

However, there is some criticism of the evaluation of the show-case spaces; Sørensen, a Norwegian Senior Research Engineer, felt that although safety may be improved in one area, there was no "significant safety effect for whole city centres" – in other words accidents had "moved" elsewhere (Sørensen, 2011). He also stated that over time motorists may become familiar with the space and so pay less attention and drive faster. It will be necessary to revisit some of these spaces in a few years time to see if the improvement in safety has remained.

The ideas of Shared Space have been implemented in other countries such as the United States where, according to the architectural writer Philip Langdon (2008), the concept has been rolled out in "Seattle, Portland (Oregon), San Francisco, Santa Monica, and other cities on the West Coast; in Cambridge, Massachusetts, New York City, and other places in the East; and in scattered places in between, such as New Town at St. Charles, Missouri, and the South Main development in Buena Vista, Colorado".

In Australia, there are official areas designated as 'Shared Zones' in many towns. The legal regulations vary from state to state but are based on the Australian Road Rules, one of which is that vehicles must travel at less than 10km/h with pedestrians having right of way (NTC 2009). As observed by Gilles from the University of New South Wales, these rules are not always observed (Gillies, 2009). Figure 2.4 shows examples of pedestrians waiting for cars which are clearly not giving way, despite the obvious 'Shared Zone' sign visible in the picture.

Figure 2.4 Pedestrians waiting for cars despite the existence of Shared Zone (Gillies 2009)

In other places zones may not be completely described as Shared Space, but they do share many of the characteristics. Gilles describes these as "quasi shared spaces", and gives examples in Bankstown (Sydney), the approach road to the Sydney Opera House, and in Melbourne. They are areas where there is a change in road surface and where pedestrians feel freer to use the same space as the vehicular traffic (figure 2.5).

Figure 2.5 Photograph of Chapel Rd, Bankstown 'quasi shared' space. (Gillies 2009)

The National Council for the Blind in Ireland (NCBI) report that while Shared Spaces have not been fully implemented in Ireland, "bits and pieces of the original concept have been adopted" in, for example, O'Connell Street (Dublin) and Patrick Street (Cork). (NCIB 2011) see figure 2.6.

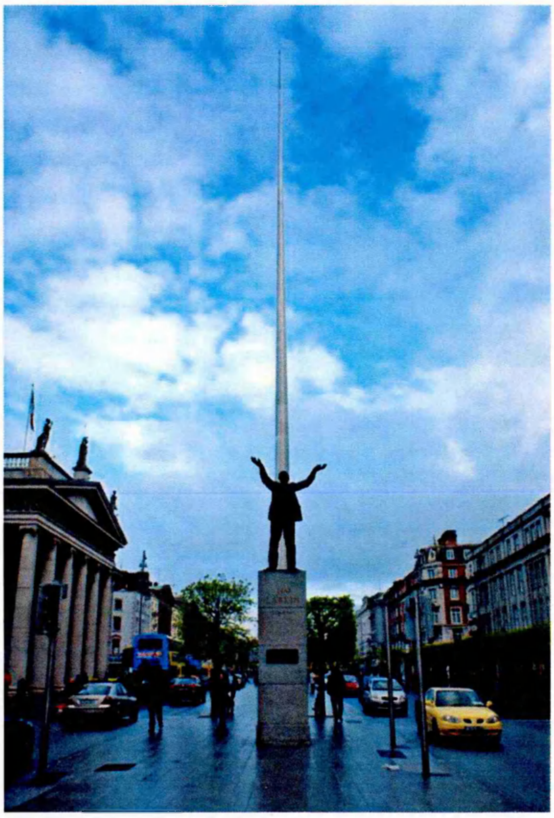


Figure 2.6 O'Connell Street, Dublin

Some aspects of Shared Surface design shown (Jaqian, 2009)

2.5 Home Zones

The Institute of Highway Incorporated Engineers define Home Zones as being "residential streets in which the road space is shared between drivers of motor vehicles and other road users, with the wider needs of residents (including people who walk and cycle, and children) in mind... motorists perceive that they should give informal priority to other road users" (IHIE 2002).

In 2001 a £30 million initiative was launched in England to promote the development of Home Zones. These areas are somewhat more prescriptive in their implementation than Shared Space with, for example, the speed limit being set at 20mph in England (10mph in most other European countries). It appears that the emphasis in England is on reduction of road traffic accidents in residential areas, whereas Shared Space has a somewhat more general aim, and is more concerned with the integration of different road users. Home Zones are more likely to use traditional traffic calming installations such as raised humps and chicanes, whereas Shared Space regions are more likely to include areas with very little or no separation of road users. The photograph (figure 2.7) shows speed bumps on a residential street in West Ealing which is part of a Home Zone. In this area pavements have been retained, and there is a clear distinction between the road and the footpath.

Figure 2.7 Home Zone: Speed bumps in Hastings Road, West Ealing (Five Roads Forum, 2011)

In figure 2.8 by contrast, there is no raised kerb and the pedestrian area is indicated by a different colour. In this example the pedestrian zones are clearly at the edges of the street. Motorists are meant to recognise the unusual road colouring as a marker that speed should be kept low, and that there may be other road-users sharing the space.

Figure 2.8 Portchester Home Zone (JMU Access Partnership, 2007)

2.6 Pedestrian Precincts

In the "Public Realm Policy Statement" issued by the "Joint Committee on Mobility of Blind and Partially Sighted People" pedestrian precincts are listed as the first of the "issues" about which recommendations are made (JCMBPS, 2008). Included in the description of difficulties blind and partially sighted people face, is the absence of non-visual orientation cues such as tactile pavements and "corduroy warning surfaces" such as should occur around a set of steps. Incidentally other problems included obstacles which had been moved into the line of pedestrian walkways such as street furniture, seating, advertising or even garbage.

In the survey on orientation mentioned earlier, several respondents stated that they found pedestrian areas difficult using descriptive words such as "daunting" to describe them and words such as "insecure" and "vulnerable" to describe their emotions when using them. One respondent mentioned the "random" spacing of street furniture and said it was difficult to follow a straight line (on foot) and another stated it was "easier to miss landmarks such as side-streets and openings". These anecdotal findings support the general statements of the JCMPBS.

It is interesting to note that the Department of Transport do have guidance on the use of tactile pavements in pedestrian precincts. They recommend, "a series of raised, flat-topped bars running in the direction of pedestrian travel" which can be described as corduroy paving (DfT 2011a). The pavement is meant to be placed in areas where there is no traditional pavement separating the building line from the carriageway. However, since this is a recommendation, there is no legal requirement to provide this type of paving. Furthermore, it does not appear to include indoor shopping centres where there is no carriageway, but where, nevertheless, people with visual impairment find it hard to be orientated. (An example of corduroy paving is depicted in figure 2.9 where it appears to separate the pavement from the road at a raised crossing, Stott, 2012).

Figure 2.9 Corduroy Pavement at a Crossing (Stott, 2012.)

2.7 Some Concerns Expressed by People with Disability

Many of the concerns raised in recent years have focused on Shared Space, and these are reported here. However, the issues apply to the other difficult urban environments just discussed.

As yet the use of Shared Space is only gradually being brought into the United Kingdom and, while safety aspects may be improved for most users, the community represented by the disabled feel that they may be at more of a disadvantage in these new spaces compared to those with full sight. Guide Dogs has expressed concern that blind people are avoiding such areas as they have little recognisable features to help them navigate. In addition, the nature of their disability means that they are unable to interact with other road users in the same way that sighted people can. One specific problem brought to light is to do with crossing from one side of a space to another. With the removal of the kerb a person with disability has one fewer marking to ensure that he or she is moving away from an edge at a right angle, and can no longer be sure that they will reach the other side in as short a time as possible. Guide dogs generally rely on pavement edges to do their work, and will stop at a kerb to ascertain what their owner wants to do. If kerbs are removed then this is problematic as, according to the NCBI, they cannot be trained to respond to tactile surfaces in the same way as they respond to kerb edges (NCBI, 2011). Furthermore, guide dogs become stressed if cyclists and cars pass close to them, as may well happen in a Shared Space area.

Guide Dogs has produced a report of the findings of two study groups: one group reported on the Shared Space architecture in two towns in Holland (Haren and Drachten) (Guide Dogs 2006b) while the second reported on the situation in the UK (Guide Dogs 2006a).

The first group, looking at the situation in Holland, found that there is a particular concern for blind users when shared surfaces are employed. Shared surfaces are that aspect of the Shared Space concept involving a surface common to both road users and pedestrians – in other words where there is no separate pavement or the pavement is on the same level as the 'road' surface. Such surfaces are consistent with the philosophy behind Shared Space, but are not essential for it. In any event, the report gave anecdotal evidence that blind people found this Shared Space environment difficult. Some spoke of the difficulty of losing their bearings when on a shared surface and the extra feeling of vulnerability they felt, as opposed to losing their way on a pedestrian-only surface. The report commented that "one participant, older than the others, reported that he no longer used shared surface areas unless he was with a sighted companion". This obviously goes against the 'sharing' principle of the Shared Space concept, and implies an inherent disadvantage associated with this particular type of architecture.

The UK group looked at various towns where there are shared surfaces and contacted 67 people with visual impairments and other disabilities who have experience using these surfaces. Akin to the Holland group, they found a particular concern over shared surfaces. Participants in the research felt more vulnerable and some reported feeling high levels of stress or exhaustion from the concentration involved. A further problem was that the participants felt they had not been consulted at the planning stage. This was a similar concern for the Holland group.

The RNIB support Guide Dogs in their concerns about the introduction of Shared Space. More recently both groups have campaigned against the incorporation of Shared Space ideas in London and the RNIB said in a statement, "We have consistently supported the lead given to this campaign by Guide Dogs and we will continue to oppose dangerous shared surface schemes" (RNIB, 2009b).

2.8 Comparison of Guide Dogs Design Principles and DfT Guidelines

Concurrent with the course of the current research there has been documentation produced by Guide Dogs, "Inclusive Streets: Design principles for blind and partially sighted people" (Guide Dogs 2010) and by the Department for Transport "Local Transport Note 1/11 Shared Space" (DfT, 2011b). David Cowdrey, Head of Campaigns for Guide Dogs, has also posted a comment on the Department for Transport document so dialogue continues (Cowdrey, 2011).

Table 2.1 shows some of the similarities and differences between the reports, and it is notable that there is considerable agreement on many of the issues. Guide Dogs lay more emphasis on confidence and feeling safe in the urban environment, whereas the Department for Transport (DfT) place somewhat more emphasis on reducing the dominance of vehicular traffic and increasing the social use of Shared Space streets. The DfT report includes more factors affecting people with disabilities other than visual impairment than that of Guide Dogs, as would be expected.

Issue	Guide Dog Recommendations	DfT Recommendations Table 2.1
Vocabulary used in report regarding urban design priorities	Safety Confidence Inclusively designed Feel safe and comfortable Independent use Disability Equality (quoting from the Disability Discrimination Act, 2005)	Stakeholder engagement Inclusive design Reduction of dominance of motor vehicles Sustainable design Balance between movement and place Pedestrian comfort Increased social activity in street Equality Act, 2010
Evidence Base	Reports from groups who travelled to Shared Space sites in the UK (Guide Dogs, 2006a), and the Netherlands (Guide Dogs, 2006b)	Stage 1: Appraisal of Shared Space (MVA, 2009). Shared Space: Operational Assessment (MVA, 2011a) Shared Space: Qualitative Research (MVA, 2011b).
Delineation of pedestrian routes and use of flat or shared surface	Kerbs and tactile surfaces should be used (in other words, no shared surfaces). Reference points such as kerbs and building lines are vital. Clear, continuous and readily identifiable routes for pedestrians needed – delineated from those of cyclists and motorists. Surface texture and/or visual contrast with band or changes in colour. Street furniture e.g. seating, bollards, planters, trees could be used with careful design	Reduction of demarcation between users. Side of street should be for pedestrians. Level surfaces are beneficial for narrow streets. Comfort space – area predominately for pedestrian use – **may** be beneficial when level surface used. If kerb-free design desirable, mitigating measures **may** be required. Tactile (corduroy) paving can be useful as a delineator strip (600-800mm). Tonal contrast as aid to navigation important (edge of carriageway, comfort zone). Complicated surface patterns should be avoided. **Comfort space may be added later if need not clear cut from start.** Trees, street sculpture, bollards, planters can be used.
Objects in pedestrian route	Unexpected features and obstacles should not be in circulation routes. Potential obstacles should have high visual contrast from background.	A-boards, tables, chairs may require local authority licence. Building line should be kept clear.
Pedestrianised Areas	Authorised vehicles only with max speed of 10mph. Designated stopping areas for vehicles. No cyclists. Tactile guidance paths and visual contrast should be used to give reference points.	Many Pedestrianised areas prohibit cyclists or allow during certain hours only.

Issue	Guide Dog Recommendations	DfT Recommendations
Traffic Speed	20mph in suburban areas (though 12mph would be better)	"Design speed" important – no more than 20mph <15mph preferable. Speed falls with less demarcation.
Layout	Logical	Scheme development is critical. Design for easy maintenance. Ladder-grid pattern helpful.
Crossings	Controlled (that is with lights, audible and tactile signals) Should be at ends of street and regularly along the street. Near bus stops and key destination points.	Uncontrolled – except for some busier Shared Space streets. Zebra crossings can be used but then legislation must be observed. Courtesy crossings useful – but no statutory requirement for drivers to give way. Such crossings can be clearly indicated (several ideas given including tonal contrast and tactile paving). Ladder-grid pattern helpful.
Lighting	Good lighting essential	Well designed lighting is important
Maintenance and Management	Regular cleaning and removal of obstacles.	Well maintained, even surfaces free from clutter and obstructions desired. Schemes should be reviewed. Post-scheme monitoring is important and needs funding. 3-year pre-scheme personal injury data required for baseline.
Consultation	Local people, disabled, older people and children need to be consulted during policy development; design, delivery and implementation of streets and external spaces	Needs of diverse range of people are to be properly considered at all stages of the development process. Engage with stakeholders including local access groups Useful to have local consultation with access groups – site specific solution regarding comfort zones
Eye Contact		Research showed no evidence of negotiation by eye contact. Cannot be relied upon. More subtle signals evident.
Education and Training	Should be training on mobility needs of disabled and equality issues	Familiarisation training for blind and partially sighted people when scheme opens

Table 2.1 Comparison of how issues are dealt with by Guide Dogs and Department for Transport

It can be seen from table 2.1 that there is a large degree of difference in the desired use of kerbs between the two reports. Where the Guide Dogs' report sees kerbs as being essential as a navigation cue, the DfT state that "level surfaces" may be beneficial in order to make the use of the street more flexible, particularly in cases where streets are narrow. Their report recommends that where "level surfaces" are used then a "comfort zone", for primarily pedestrian use, may be of benefit, and could be delineated by tonal contrast or corduroy tactile pavement. Although the balance of the report appears to encourage the use of a comfort zone and the delineation of such, it is not clearly recommended, and other parts of the report discuss reducing demarcation in order to slow down traffic. This could leave planners open to design level surfaces with no clear comfort zone.

At one point the DfT report suggests that a comfort zone could be added later if the need arises during monitoring. While monitoring is clearly to be commended, it would surely be better to design in a comfort zone from the start. Furthermore, if blind and partially sighted people simply avoid the space, then the need to have a comfort zone may not be recognised by later monitoring of the street.

Another area of disagreement is that of crossings. The Guide Dogs' report recommends that all crossings should be controlled and have audible and tactile warnings, and that there should be crossings near bus stops and other "key destination" points, as well as regularly along the street. The DfT place emphasis on "courtesy crossings" which are not controlled, and in fact at which there is no statutory obligation for motorists to stop. There are a number of recommendations in the report for how courtesy crossings are to be demarcated, and these include the use of tactile paving and tonal contrast, which will provide navigational cues for people with visual impairment.

Controlled crossings, the report suggests, could be used in Shared Space streets which are "busier", but this is not defined.

Both reports suggest that delineation of pedestrian routes within Shared Space can be made by the use of colour. The DfT report advises against the use of complex surface patterns as these can be difficult for people with partial sight to clearly detect. Figure 2.9 shows a photograph of New Road, Brighton, a Shared Space region completed in 2007, where a rather complex design has been used. It is not easy to see where there is a separation at the side of the road and possibly a comfort space.

Figure 2.10 Photograph of New Road depicting a rather complex surface pattern (Gillett, 2009)

The DfT report alludes to the use of zebra crossings in Shared Space areas, but points out that once a zebra crossing is implemented there is strict regulation regarding, not just the familiar black and white stripes, but also other road markings such as zig-zag lines and street furniture, such as orange globes. These regulations are set out in "The Zebra, Pelican and Puffin Pedestrian Crossings

Regulations and General Directions" (Legislation, 1997), and the road markings are seen in figure 2.10. The DfT suggest that this amount of road markings would be detrimental to the concept and appearance of the Shared Space, and may lead motorists to take less care or be less likely to give way at courtesy crossings. The question needs to be asked if legislation should be changed to allow more flexibility in the use of zebra crossings in Shared Space streets. Could the familiar black and white strips be used with the inherent statutory obligation on motorists and cyclists to give way to pedestrians, but without the requirement for the additional road markings and street furniture currently compulsory? Additional lights, audible and tactile cues could then be added at the discretion of the planners in consultation with local access groups.

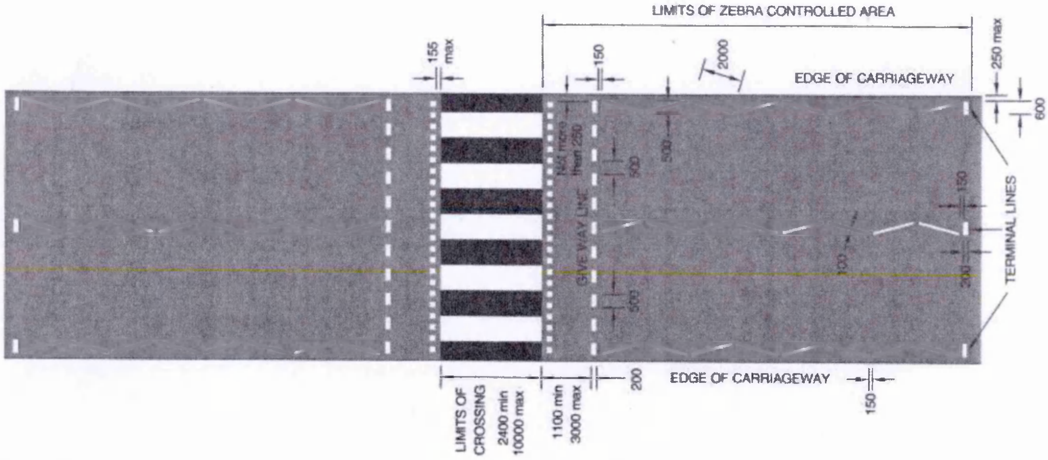


Figure 2.11 Regulation road markings at a zebra-crossing (Legislation, 1997)

In his comment on the DfT report, mentioned above, Cowdrey is critical of several points. He feels that there are omissions including lack of specific instructions about monitoring effective schemes. However, the DfT report does stipulate that schemes should be monitored and states that funding for monitoring and maintenance should be set aside. Perhaps clearer instructions as to how monitoring should be carried out would help.

Another omission pointed out by Cowdrey is to do with the increased height of some kerbs at bus stops. This is to allow easier access for wheelchair users and pushchairs. The increased height is perceived as a danger to blind and partially sighted people, who may not be aware of the increased step height when stepping off the kerb before alighting onto the bus. Translink, Northern Ireland, suggest that although "European cities have experimented with higher raised kerb areas at bus stops to achieve level access to floor height of low floor buses" such heights raise safety issues for pedestrians, as well as posing "serious damage risk to buses if a parallel approach is not consistently practical", and therefore this practice is not recommended in Northern Ireland (Translink, 2005). Other organisations have made similar remarks (for example Transport for Greater Manchester, 2007). Therefore the DfT may consider that this is more of a local issue and, in any case, there is no need for them to include commentary on kerb heights at bus stops as higher kerbs are not being widely implemented.

One aspect that the DfT do address is that of "eye contact", and they specifically state that their own research showed no evidence of negotiation between drivers and pedestrians by this means. Their evidence indicated more "subtle signs", such as drivers noticing pedestrians standing at the edge of a carriageway looking along the traffic as if they wanted to cross. The DfT insist that eye-contact should not be relied upon as a means of communication between motorists and pedestrians. Despite this, Cowdrey raises the issue in his comments on the DfT report arguing that "it can be hard for even sighted pedestrians to see into a vehicle". The issue of "eye contact" is clearly a contentious one, and this may be due to early advocates of Shared Space who used the term, perhaps unwisely, to describe the increased interaction between pedestrians, cyclists and motorists in such environments. It might have been more helpful to keep to an analogy used by Ben Hamilton-Bailie of skaters on an ice-rink, aware of and avoiding one another, even though moving in a fairly un-ordered way around on the ice (BBC, 2012). The point that Hamilton-Bailie and others were making is that when road users are segregated there is little to no interaction between them, to the extent that drivers are almost oblivious to the presence of pedestrians. In Shared Space it is arguably the motorist who has to adjust more as he or she is forced to take note of, and indeed give way to other road users. Motorists must use their eyes to anticipate the actions of pedestrians in a way that has become unprecedented in traditional roads and streets. So it may be agreed that it is not realistic for a pedestrian to catch the eye of the passing motorist – but the motorist should be more aware of the body language of the pedestrian – even if that pedestrian has a visual impairment.

2.9 Conclusion

The navigation of urban environments requires a specific skill set for people with visual impairment. The development of areas without traditional navigational cues and aids is a cause for concern for groups representing disabled interests. Individuals with visual impairment may lack confidence and feel vulnerable in such regions. Recommendations by the Department of Transport have been recently introduced, and although these emphasise the need to consult with local bodies, organisations representing the blind and partially sighted are not entirely satisfied. Still of specific contention is the idea of shared surfaces (level surfaces) where there is no kerb.

2.10 Research Questions Arising

The difficult urban environments discussed in this chapter, particularly the concepts of Shared Space and shared surfaces, are relatively recent. Aids to help people with visual impairment navigate their way through the built environment largely predate such environments, and a valid question to ask is whether existing assistive technologies form the basis of an urban environment navigational system that can tackle such areas.

In the United Kingdom guidelines about the development of Shared Space areas have now been released by the government, and although there are still areas of controversy within these, certain common features are likely to be built into new developments. If these features are consistent then it may be possible to gather information about them and feed that data to a person with visual impairment to help them navigate. A valid research question is, "How can any other information about the urban environment be gathered easily using suitable technology and fed to a blind or partially sighted person in a helpful way?"

Chapter 3 Literature Review

3.1 Introduction

This chapter looks at current and historical thinking on a number of themes that were taken into account during the research process. The themes are gathered here for convenience although aspects of each were encountered at different stages in the research development. The topics covered in this chapter are:

- Vision substitution History and Brain Plasticity
- Neural Networks
- Autonomous vehicle obstacle avoidance
- Human Computer Interaction (HCI) and user feedback case studies
- Mobility, orientation and aids used by sighted and blind individuals

Section 2.10 elucidated the research question concerning whether existing navigational aids were sufficient in a difficult urban environment, and this question is explored in the topic on mobility and orientation (Section 3.6). Other research questions arise during the course of each topic and are gathered in the conclusions at the end thereof.

3.2 Vision Substitution History and Brain Plasticity

In this section various papers and other publications which deal with the subject of sensory augmentation and navigation are discussed. Sensory Augmentation is treated first, and several areas of research looked at, to examine how successful current and prototype devices have been in helping people with visual impairment. To conclude, there is a look at progress on devices produced specifically for navigation or wayfinding.

3.2.1 The case for Sensory Augmentation

An important question for any device which gives additional information to an individual is the ability of the brain to usefully process that information. Several researchers have looked at devices which feed data to a person with visual impairment through other senses, particularly touch and sound. They have investigated how the brain manages this information. The following matrix (table 3.1) compares the approach of four investigators.

	Vision Substitution System	**Description of Brain Plasticity**	**Testing on a number of subjects**	**Nature of tests**	**Training mentioned**
Bach-y-rita (1984)	TV camera to tactile device on skin surface	Lengthy discussion of process	Unstated number but more than 2	Recognition of objects, tracking moving objects, manipulating objects requiring hand-eye coordination	Extensive training required (5 - 10 hours)
Kaczmarek 2004	Camera to electrostatic haptic device scanned by fingertips	Not mentioned	8 subjects	Testing comfort of haptic device. Comparison with abdominal pins	Some
Meijer 1992	Camera to audio	Described	Tested on individuals	Individuals trained to use device to recognise shapes	Training required
Maucher *et al.* 2000	Camera to tactile device scanned by fingertips	No mention	6 blind and 4 sighted (blindfolded)	Recognition of geometric images	Little training

Table 3.1Comparison of Approaches to Vision Substitution Research

The table shows that there are several differences in the approaches taken by the different research teams. Some, such as Bach-y-rita, spent considerable time training their volunteer subjects in using the equipment whereas others, such as Maucher, did not. Three of the four used tactile feedback with only Meijer's group relying on audio feedback. Half of the researchers considered the question of brain plasticity. All the groups tested their devices on volunteers. The following sections describe the approaches taken in more detail.

3.2.2 History

Before embarking on designing a prototype device using sensory augmentation or vision substitution, it was judicial to examine what other researchers had done. This section describes the development of sensory substitution devices with a particular look at the feedback mechanisms employed, as this was part of the research problem. From this problems encountered by the pioneers of vision substitution are described, together with their solutions.

Bach-y-rita carried out pioneering research in the area of vision substitution in the 1970s and 1980s (Bach-y-Rita 1984). In this work a television camera was connected through a signal processing system to a tactile pad placed on the back or abdomen of the subject or user. This pad consisted of a number of moving pins in a square array. Each pin was able to move independently and make contact with the underlying skin if the corresponding 'pixel' in the camera received light below a fixed threshold. In this way an image would be transferred from the camera to the subject's skin. Bach-y-Rita reported that the subject was able to discern an object from the pressure of the pins on his or her skin. A photograph of the original set up is shown in figure 3.1.

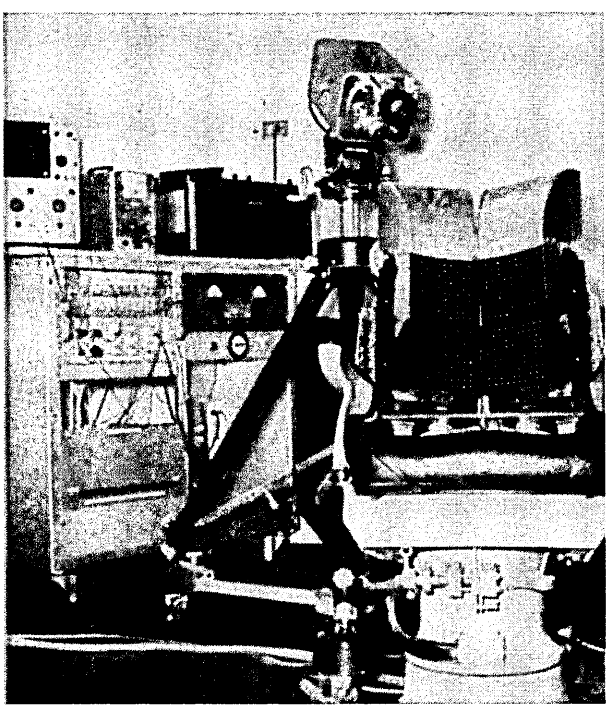


Figure 3.1Bach-y-rita's vision substitution system showing the camera and chair with moving vibrator rod pads (Bach-y-rita et al. 1969 page 963)

In the paper "Seeing with the Brain", Bach-y-Rita and Kaczmarek (2003)describe the history of the development of "Vision Substitution" devices and how these may be of use, not only for the visually impaired, but for any activity where sensory input is required, but not immediately available from the natural senses. They state that the brain is capable of handling the sensory information and processing it in such a way as to enable the user to form a mental 'picture' of the environment. The paper relates the experience of individuals who, after undergoing training, and, in response to the image of an object relayed to an array of vibrating pins on their abdomen, "reported experiencing the image in space, instead of on the skin". (In this case the array of vibrating pins was linked to a television camera). Furthermore, the subjects were able to analyse data about the object in the same manner that a sighted individual would have done, making "perceptual judgments using visual means of analysis, such as perspective, parallax, looming and zooming, and depth judgments".

3.2.3 Difficulties encountered

Difficulties are found in every system. Three of the problems that the pioneers in Vision Substitution encountered are outlined below:

1 The necessity to develop tools to test how effective a possible substitution system is - What should be the 'bench mark' for such a system?

2 The problem of understanding the ability of the brain to interpret one sense (tactile or auditory) and assign its input to that of another sense (visual).

3 For tactile systems, the development of generating tactile patterns over relatively large surface areas and the corresponding challenges to computing ability, mechanical interfacing and power supply. For auditory systems, the processing required to convert the signals to a useful form.

3.2.4 Overcoming the difficulties

The methods that different researchers have taken to analyse these problems vary. Dr Kurt Kaczmarek in his report "Electrotactile Display of Computer Graphics for Blind, Final Report", states as one of his aims that he wanted to compare electrostatic and electrotactile stimulation. A second stated aim is the comparison of the efficacy of an abdominal tactile display with one sensed by the finger tips (Kaczmarek, 2004). For both these aims Kaczmarek used one established system (the abdominal one) as the bench mark for a newer system (the fingertip device). At the same time he addressed the third problem outlined above, as the electrostatic solution used considerably less power than that developed by Bach-y-Rita's team.

Peter Meijer developed a system which translates a visual input to an audio signal. He found that a trained individual was able to use this signal to recognise shapes and gain a perception of their surroundings. In a paper presented at a conference in Tucson he makes reference to the ability of the brain to process auditory signals and reinterpret them as visual ones. He asked if it were possible to determine if the subject is truly seeing, or "if the brain is 'only' doing much extended and very sophisticated auditory processing to make good use of the camera sounds" (Meijer, 2002). As explained below this question was at least partially answered through the use of brain imaging techniques.

A possible improvement on Meijer's work was explored by Picton and Capp (2004). They noted that subjects using Meijer's system tended to require "considerable effort and concentration" due to the amount of information that was being received through the auditory channel. In their work Picton and Capp looked at reducing the amount of information through the use of stereo depth maps and cartoon depth maps. At the time the authors felt that processing speeds of computers would not be fast enough to complete the map generation and the sound pattern at typical video frame rates, though they felt that speeds of 1 or 2 frames per second (fps) might be achievable. (Typical video frame rates are 24 fps.)

Thorsten Maucher *et al.* (2000), face the problem of developing a large area tactile device in their paper "The Heidelberg Tactile Vision Substitution System". Their solution was to build a relatively large table top device but with only a small section, the part being scanned by the finger tips, active at any one time. The Heidelberg Virtual Tactile Display is shown in figure 3.2.

Figure 3.2 The Heidelberg Virtual Tactile Display showing the moveable pad and the tactile output unit together with a CMOS vision chip camera

3.2.5 Evaluation of the work

Comparing benefits and limitations, Kaczmarek compared his electrotactile system favourably, in terms of its ability to relay detail to the user effectively, with the work done by Bach-y-Rita. As stated earlier, Kaczmarek's research also addresses the problem of power consumption as the electrostatic array requires less power than the moving part array used previously. Evaluation of the work is limited in that the number of subjects who were tested is small: in the report sets of subjects numbered seven, eight or a maximum of nine which is, statistically, a small sample. It was also somewhat limited in the style of testing. Much of the testing reported in the paper was concerned with finding the optimum electrostatic signal, although some testing of pattern recognition was also carried out. Testing of ease of use and acceptability to the subject or user is not reported. Neither does the team focus on the problem of understanding the plasticity of the brain and how one sensory input is interpreted as another.

By contrast, Peter Meijer's work makes more of the actual and current benefit given to visually impaired people by using a substitution system based on sound. For example, at a conference in Tucson in 2002, Pat Fletcher, a blind user of the device, gave a talk based on her experiences in which she recounted the great use she was able to make of the system (Fletcher 2002). Furthermore, fMRI (functional Magnetic Resonance Imaging) scans indicated that the visual cortex was functioning in subjects using Meijer's device, and the authors of a presented poster based on the work of Meijer state, "Our results show that intensive training with a SSD [sensory substitution device] can lead to activation of visual areas in response to auditory stimuli that encode spatial information normally used to map retinotopic visual areas" (Merabet and Poggel, 2008). Figure 3.3 shows the fMRI scans of a subject before and after training with the SSD and clearly indicates that the visual cortex is active when the device is operating.

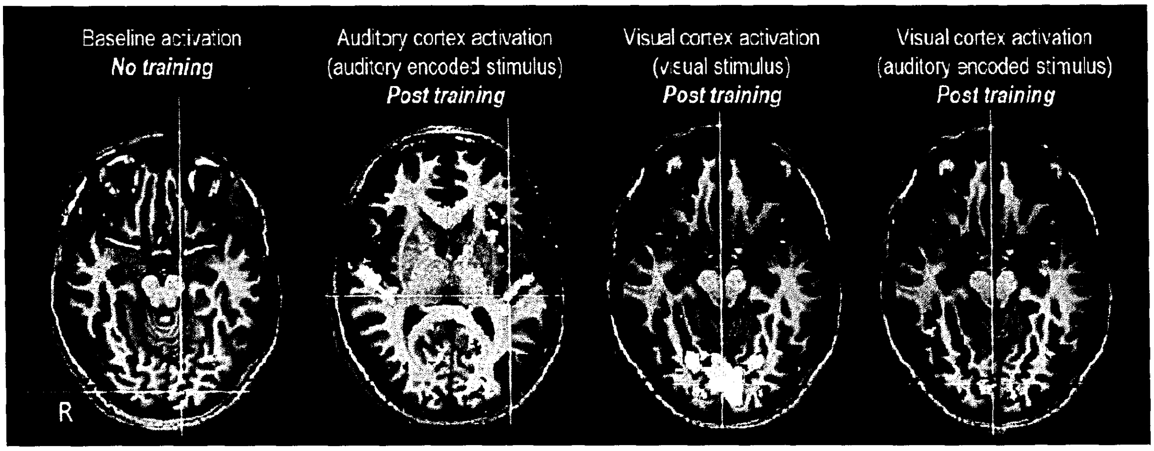


Figure 3.3 fMRI scans showing brain activity after training when a subject is using a sensory substitution device (third image from left (Merabet L., Poggel D. 2008).

On the other hand, evaluation of Meijer's work is limited in that there are few peer-reviewed papers available. The most recent advertised full paper on the associated web-site is a 1992 paper in which Meijer presents detail of the device, but gives little in the way of testing (Meijer 1992).

The ability of the visual cortex to interpret information in this way is also noted by other researchers such as Sadato who, during PET scans of blind children reading Braille, concluded that "in blind subjects, cortical areas normally reserved for vision may be activated by other sensory modalities" (Sadato *et al.*, 1996).

Thorsten Maucher and his team successfully built and tested a visual tactile display. The testing is clearly thought out and results between sighted and visually impaired subjects compared. As for Kaczmarek one limitation in the evaluation was in the sample size – 6 blind and 4 sighted subjects. A further limitation was in the interpretation of the results which showed that the blind subjects were able to perform one test faster and more accurately than the sighted (but blind-folded) subjects. Maucher, himself, points to further research as the way forward to find out where the strategies of a blind person are different from a sighted person (Maucher *et al.*, 2000). Finally, as for Kaczmarek's team, Maucher does not focus on understanding the plasticity of the brain.

3.2.6 Sensory Augmentation applied to Wayfinding

It would appear that for vision substitution, or perhaps more accurately sensory augmentation, to work successfully it is preferable for the device to be under the motor control of the subject. Charles Lenay found that his subjects only truly perceived an object in space when using a tactile vision substitution system when they were able to manipulate the camera themselves. Lenay (2001) observes, "the spatial localisation of a target requires dynamic sensori-motor coupling" for the subject. This is consistent with research carried out by Zelek on wayfinding where subjects use a haptic glove linked to a stereo camera which they themselves carry (strapped to their waist). Zelek's research gave the user feedback through a haptic glove and a human operator remotely enabled the user to sense stationary obstacles to the left, right or in front through the glove (Figure 3.4).

Figure 3.4 Vision Substitution system; "the camera is strapped to the waist, the haptic glove is worn on the non-carrying hand and the backpack contains the computer" (Zelek, 2005)

An obstacle course was set up using boxes in the laboratory and it was found "that the paths traversed by the test subjects using [the] device were as good if not better than the paths traversed with their existing aid [white cane or guide dog]" (Zelek 2005). This shows that navigation is possible through tactile feedback, however, Zelek's system did not actually use a computer system to process the video feed from the camera, but relied on a human operator to send signals to the haptic glove as the volunteer approached obstacles.

The work by Kurachi *et al.* (2005) involves external signals beamed to a subject with a suitable receiver. The signals could be radio, infra-red or RFID, and give information about waypoints on a specific path through an area. This system could be adapted for a Shared Space, but would lead to fixed paths for users to navigate and may not provide the freedom that the user may like.

3.2.7 Conclusions from Vision Substitution Literature

Some features of the research approaches discussed in this chapter are common. It is not easy to test a human interface device on a significantly adequate sample and the teams have limited themselves to relatively small sample sizes. Each team has majored on developing the technical equipment and researched into, and then solved, technical problems in different ways. The research into brain plasticity shows that it is possible for the brain to adapt to new information arriving through a familiar channel, but actually pertaining to a different one. This is discussed further in the section on human computer interfaces (section 3.6). It is also clear that the development of a visual substitution or sensory augmentation device necessitates a certain amount of training before the user is successfully able to benefit from it.

Sensory augmentation devices work best when under the control of the user and after training give useful information to the user. Many devices require extensive training and some interfere with other input signals (for example audible traffic noise). Nevertheless, it seems probable that a well constructed sensory augmentation device could give a user increased confidence about the terrain that he or she is travelling in, as shown by the work of Zelek, for example.

The research question that needs to be asked is, "Can a device be built which will be under the control of the user and which, with some training, will provide sufficient feedback to a blind or partially sighted user so as to increase their confidence when travelling through a difficult urban environment?"

3.3 Literature review of Neural Networks and Autonomous Vehicle Obstacle Avoidance Systems

The literature review of sensory substitution systems (section 3.2) shows that systems can be built which give feedback to blind and partially sighted users about their environment. Moreover, the work by Zelek showed the potential of such devices in the navigation of environments. Chapter 1 refers to the idea of providing feedback to a person with visual impairment about the surface they are travelling over, and the research described in this thesis investigated two separate methods of processing the video feed from a camera so as to provide such feedback. The first method was the use of neural networks to classify the surfaces and a literature review covering the history and characteristics of neural networks is described in section 3.4. The second method adapts a method used in autonomous vehicles, and so the literature review of neural networks is followed by one of autonomous vehicle obstacle avoidance systems (section 3.5).

3.4 Neural Networks

3.4.1 Introduction

Neural networks were explored in the research as a possible video processing sub-system to enable the video feed from the camera to be converted into useful information about the urban environment for the blind or partially sighted user. As a first step to developing a neural network appropriate to the problem, a literature review of the subject was carried out.

3.4.2 A Brief History of neural networks

The historical background to artificial neural networks can be said to stretch back at least to 1943 and the work of Warren McCulloch, a neuro-physiologist, and Walter Pitts, a mathematician (Abraham, 2002). They worked together to produce a set of axioms that explain hypothetical networks of artificial neurons (McCulloch and Pitts, 1943). Rosenblatt built on this work, and in 1958 published a paper describing a model for information storage in the brain – the "perceptron" (Rosenblatt, 1958). The fundamental idea of the perceptron was developed from the work by McCulloch and Pitts but, in contrast to their ideas, Rosenblatt proposed that the perceptron could "learn" by an adjustment of the internal weights linking the different neurons (Heaton, 2008, p XXXVII). However, the ability of Rosenblatt's perceptron to distinguish between patterns was limited to "linearly separable" input sets (Minsky and Papert, 1969) and the role of neural networks was not greatly advanced until the 1980s and the ability to experiment with the idea using digital computers.

In the 1980s development by Hopfield on a model of how the brain remembers information (Hopfield and Tank, 1985), coupled with a mathematical method of training a multi-layer network (known as back-propagation), led to a renewal in research into the operation and uses of neural networks (Warner and Misra, 1996). Pattern recognition machines were made using artificial neural networks, and programming languages such as C were used to develop algorithms to simulate and train the artificial neural networks using conventional digital computers. Figure 3.5 shows a three-layer network with inputs, outputs and some weights. The input into each node is multiplied by an adjustable weight, and if the total of all weighted inputs at any node rises above a given threshold then that node 'fires' by passing on an input to the next layer.

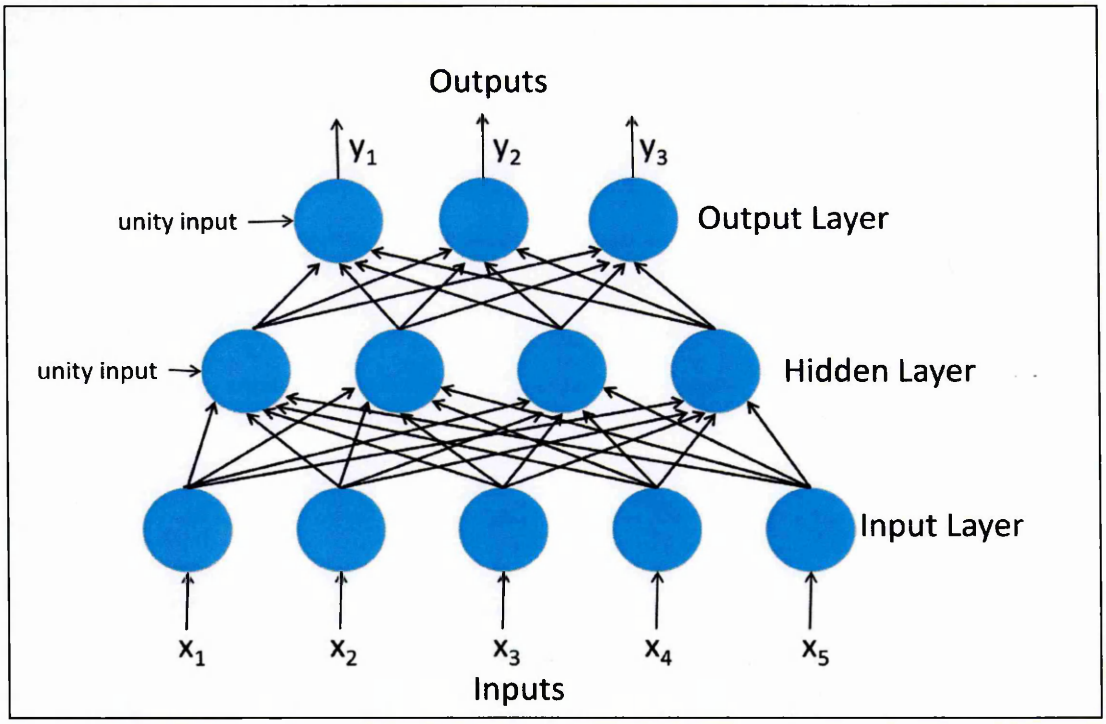


Figure 3.5 Diagram of Artificial Neural Network showing neurons, inputs, outputs and some internal weights

3.4.3 Training concepts, limitations and uses

Before they can recognise patterns, artificial neural networks must be presented with a training set of data, consisting of sets of inputs mapping to corresponding sets of known outputs, and then back-propagation, or another training method, can be used to adjust the internal weights between 'neurons' so that the output sets map to the correct input sets. The training is carried out iteratively, that is as each input-output match is considered the weights are adjusted a little to improve the mapping and then the next match analysed. This continues until all the input-output pairs are considered, and then the process is repeated. The repetitions continue until a threshold is reached where a sufficient number of input data sets in the training data match the corresponding output sets, without the need for weight adjustment. They can then be used on other input data sets to predict matches similar to the training set. There is a danger that the neural network will operate correctly with the training set but not with new data (outside the set) so ideally another set of known data should be used so that the efficacy of the network can be validated. Once validation has occurred then the network can be used on unknown data (Heaton, 2008 pp 43-47).

Work has to be done to ensure that the most efficient use is made of the number of inputs, the number of layers, and the number of iterations in the training process. There is as yet no precise way of determining these numbers in advance other than from examining what has worked well in the past, although there is a wealth of research into possible algorithms (Ripley, 1996). It may be necessary for the input data to be pre-processed in some way to provide suitable inputs for the neural network, and to do this effectively the designer must have a good knowledge of the data being examined (Johnson and Picton, 1999).

In their analysis of neural networks, Johnson and Picton state that neural networks are useful in situations where the input conditions can be mapped to known output conditions, even if the input conditions are non-linear or somewhat noisy. They suggest, however, that they are not reliable if the input conditions are likely to be rather different from their training sets (Johnson and Picton, 1999).

Over-optimistic claims have been made for neural-networks. For example in 1958 the Science News Letter reported of the perceptron that "only one more step of development, a difficult step, is needed for the device to hear speech in one language and then reproduce it ... in another language" (Science News Letter, 1958). Over fifty years later that "difficult step" has proved too much for neural networks working on their own. However, neural networks have found wide use in four particular areas (categories from Heaton 2008, p43):

1. pattern recognition, for example, signature recognition (Abikoye *et al.* 2011) ;
2. classification, for example in medicine (Yap J. *et al.*, 2010); in predicting quality of leather (Wang *et al.*, 2011)
3. prediction, for example of stock market movements (Vaisla and Bhatt 2010) and
4. optimisation, for example of engine parameters (Yap, W. *et al.*, 2012)

In at least one of the examples cited above, neural networks have been combined with other classifiers to make hybrid systems and these have been found to work well. More recent advances in neural networks have included the ability to self-organise and self-train, but these are beyond the scope of this thesis.

3.4.4 Conclusions from Neural Network Literature

An artificial neural network structure could be programmed using a personal computer or laptop and a suitable high-level language, such as C++ or Microsoft Visual Basic. The research question is whether such a program can classify surfaces, if appropriate pre-processing of images can lead to identifiable inputs, and whether that classification can be carried out at a sufficient speed, and provide enough detail, to be of use to a blind or partially sighted user.

3.5 Autonomous Vehicle Obstacle Avoidance

The aim of analysing the present surface over which a pedestrian is walking and alerting him or her if a surface change occurs can be fulfilled if every surface type is memorised and the surface is analysed rapidly. The current surface is then compared with the saved data-base of surfaces and any change leads to an alert. This is the basis of the neural-network method. The system must be able to distinguish between a change in surface, which should be notified to the user, and a change in ambient environmental conditions, which should not lead to notification. An example of the latter would be if a shadow falls across the surface.

An alternative approach to memorizing surface types and comparing the current one to a data base, is to analyse the difference between the current position and the pathway just in front of the pedestrian. If there is a sufficient difference then the pedestrian is given a warning that the surface is about to change. This approach takes inspiration from some of the current work on autonomous vehicles where a vehicle is required to find a path ahead and to avoid either a collision or driving off the road or pathway.

3.5.1 Immediate foreground analysis

Work by Lorigo, Brooks and Grimson at the Massachusetts Institute of Technology (MIT) emphasises the point that if a vehicle is currently on a suitable path then it is a reasonable assumption that the area immediately in front of the vehicle is clear and driveable (Lorigo *et al.* 1997). The principle being that the current region where the vehicle is situated must be 'safe' since the vehicle has indeed safely arrived there. By extension, if the regions surrounding the vehicle are examined and found to be the same as the current region then they can also be regarded as safe. Regions which are different should be avoided as they are a potential risk. (As far as the vehicle is concerned, the current region can also be tested with ultrasonic and other proximity detectors to ensure there are no immediate obstacles.) This idea finds practical use in vehicles which are required to fully or partially find a path for themselves such as the Mars rovers which are too far from Earth for an immediate remote feedback loop to work in the event of an obstacle (Biesiadecki and Maimone, 2006).

This principle was used by Katramados to develop a program which compared the immediate foreground in front of a vehicle with that of the path in the middle distance, as depicted in figure 3.6. The idea was that if the path ahead is the same as the immediate foreground then the vehicle is free to move forward, however, if the path is different from the foreground the vehicle should turn until a clear path is found (Katramados *et al.*, 2009). One problem, elucidated by Katramados in developing the software to look at the path ahead, is the problem of shadows and reflective surfaces, both of which would potentially appear as different surface types from the current surface. In order to deal with effects due to shadows and reflections obscuring the path ahead, the intensity invariant parameters from different colour spaces were utilised.

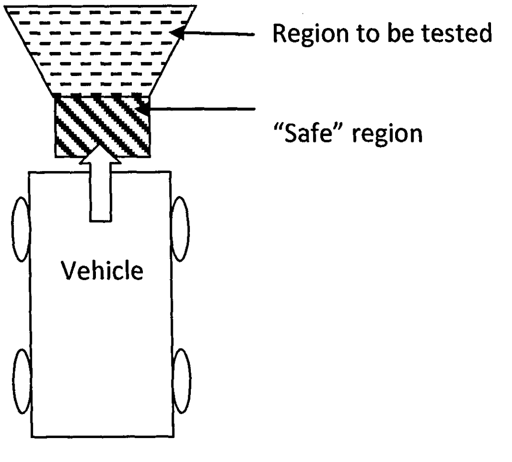


Figure 3.6: Vehicle compares 'safe region' in the immediate foreground with the surface in the path ahead

3.5.2 Colour Space Analysis

To analyse the current 'safe region' the RGB (red-green-blue colour space) input from the video feed must be converted into colour spaces best suited for the extraction of the required information. Work by Shan and by Wu among others shows that different colour spaces are better for the elimination of different details such as shadows and reflections (Shan *et al.*, 2007 and Wu *et al.*, 2007). Ideally, in the case of a pedestrian interested in surface change, the system should be insensitive to both shadows and to changes in the reflective properties of a surface caused by rain or dampness. Katramados suggests that a combination of three different colour spaces can help eliminate the artefacts caused by shadow and standing water or wet prints (Katramados *et al.*, 2009). The three spaces are HSL (hue, saturation, luminance), YCbCr (luma, blue-difference, red-difference) and LAB (lightness, A, B, where A and B are colour-opponent dimensions). Different attributes from each space are then taken to form an image which is relatively unaffected by changes in intensity typically caused by shadow or reflection.

Wu *et al.* suggest that combining the Cr and Cb colour space components of the YCbCr colour space can lead to a better system for detecting objects (in their case, vehicles) than simply using luminance values (Wu *et al.*, 2007). On the other hand, in their work *"Color space selection for moving shadow elimination"* Shan *et al.* (2007) study different colour spaces to find a combination of elements that will help eliminate different types of shadow. Katramados combines these approaches to develop an algorithm that suppresses the effects of shadow and reflection in the path being analysed in front of the autonomous vehicle. Figure 3.7 depicts an example where the photographer's shadow has largely been eliminated from the processed picture.

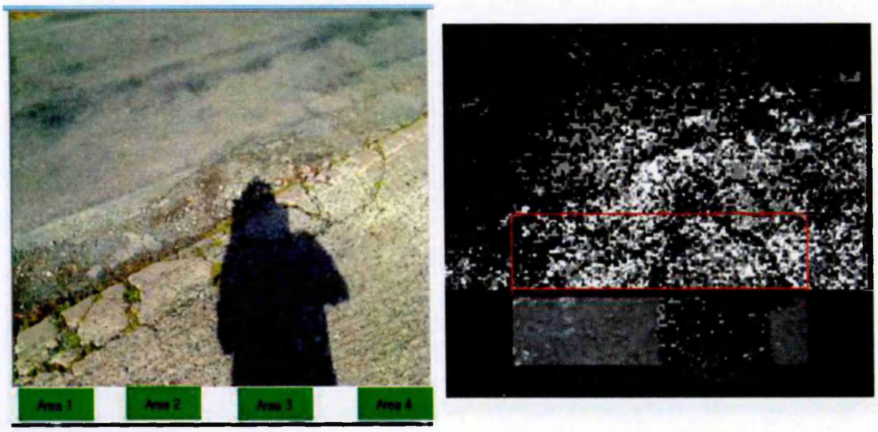


Figure 3.7 Near Elimination of shadow using colour space – the area in the lower rectangle is processed and the information used to help eliminate shadows in the upper area

3.5.3 Camera Noise Reduction

A second problem noted by Katramados is the existence of noise from the camera which again could be interpreted as an obstacle. In order to deal with this problem the program he developed looked for temporal invariant features. This builds on work by Kröese *et al.* (2000) who used optical flow to demonstrate how collisions could be avoided. Kröese postulated that mathematical extrapolation of the motion of the camera and the objects in an image field could lead to the prediction of a collision. To do this frame sequences were needed to develop the virtual paths followed by the camera and the objects. Although Katramados refers to this work, his method is somewhat different in that he does not look at extrapolating movement from frame sequences, but rather uses frame sequences to find features that are stable over time. This eliminates the problems of noise and camera blur without invoking the increased computational cost of calculating virtual paths using optical flow.

3.5.4 Conclusions from Autonomous Vehicle Literature

Colour spaces can be used to extract intensity invariant features from images of a pedestrian's path. The research question to be investigated is whether such features can be used to compare the current surface on which the pedestrian is standing with the path ahead so as to give a warning of potential change in surface type or upcoming obstacle.

3.6 *Human Computer Interfacing and user feedback*

3.6.1 Introduction

Any navigational system must have the means to communicate with the user of the system, and indeed there have been several different feedback systems used or investigated since Bach-y-rita's seminal work, discussed in section 3.2.2, which used an array of moving pins in contact with the user's skin. For example, sound has been used by researchers such as Zoltan in the development of an electronic travel aid (Zoltan, 2011), and in the voICe system developed by Peter Meijer, frequency and left-right position of sound information can be interpreted by the brain to produce a 'picture'. The ability of the long white cane to transmit vibrations to its user has been studied with a view to the development of an improved haptic system (Wong and Zelek, 2006), while Tang and Beebe (1998) explores an electrostatic system with feedback to either the skin of the fingers or the tongue. Research into haptic feedback systems have included systems not just for partially sighted and blind individuals, but also as a means to give more information to operators of delicate machinery, (for example Debus *et al.*, 2004).

As regards people with visual impairment there is an argument that feedback in the auditory channel should be kept to a minimum as they use their sense of hearing as a guide to everyday activities, such as navigation (Zelek *et al.*, 2003). The implication is that a tactile feedback system should be available. Discussion with Mr Murdoch from the Guide dog association confirmed that partially sighted and blind people do rely heavily on their hearing while navigating urban environments, and this was further backed up by conversations with people with visual impairment during the course of the current research.

This section looks at recent work on two systems that have particularly influenced the author in the development of a feedback system for people with visual impairment: The feelSpace Project which utilises a waist-belt system, and work on "Teaching Novices Correct Violin Bowing Technique" which formed part of the eSense project at the Open University.

3.6.2 The feelSpace Project

The aim of the feelSpace Project was to investigate if people could make use of an additional sense – in this case that of being aware of the direction of the Earth's magnetic field line at all times (König *et al.*, 2005). Although the project was not conceived of as being for navigation by partially sighted or blind people, the mechanism used in the project to feed information back to the subject has certain similarities to the concepts being considered during the current research.

The project was inspired by the work by Bach-y-rita and others alluded to in section 3.2 on brain plasticity, and in part was motivated by a desire to see how the brain would adapt to having an extra sense. One of the main objectives of the project was to see if having this additional sense could be integrated into a person's skill set, or whether it would be perceived as being part of another sense. Test subjects were given a belt to wear which had 13 vibrating 'tactors' attached to the inside of it, and which were interfaced through a computer system to a magnetic compass in such a way that at any one time the direction towards magnetic north was indicated by the continuous vibration of one of the tactors – the one closest to north. The subject was asked to wear the belt for 6 weeks and then went through a series of experiments to see if he or she had learnt how to utilise the information.

Figure 3.8 shows a cross section of the belt with the tactor nearest to magnetic north activated.

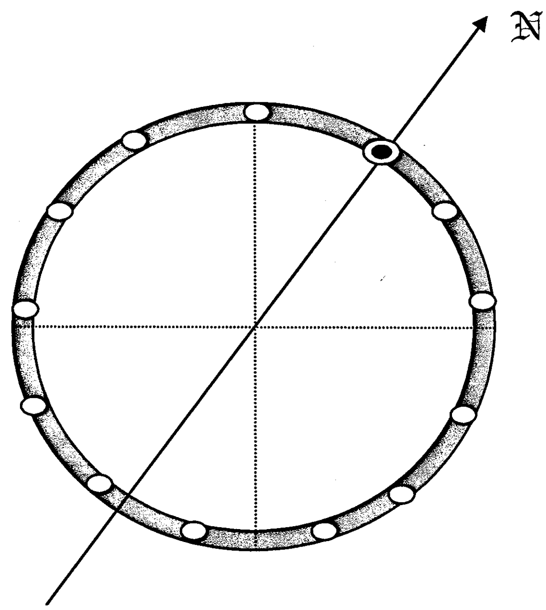


Figure 3.8 Cross section of feelSpace belt (diagram by researcher)

The belt was made of elasticated fabric and the tactors consisted of small electro-mechanical vibration motors glued into adapted plastic bottle tops for added robustness. Vibration frequencies and intensities were varied to find the most sensitive and acceptable values, which turned out to be 250Hz, although other frequencies worked well. It is reported that humans are most sensitive to the frequency range 200-250 Hz (Tan and Pentland, 1997, Jones and Sarter, 2008)). The team chose a commercially available orthopaedic belt on the basis that it could be worn for long periods without causing skin irritation.

Feedback to a user could have been through vibrators attached to other areas of the skin, for example, the shoulder (Toney *et al.*, 2003); the abdomen, as seen in work by Bach-y-rita above and other researchers such as Cholewiak *et al.* (2004), who carried out experiments to determine the effects of different variables on a vibrotactile display on the abdomen; or the forearm as seen in work by van der Linden, as explained in section 3.5.2 below. However, a belt around the waist was chosen for the feelSense project as it was felt that this gave clear unvarying orientation information as long as the user remains in an upright position, and the torso was perceived as being able to receive tactile information on a 360° basis.

To ascertain the success of the feelSpace project 12 subjects were involved in various experiments. Four had worn the belt for a period of six weeks and received training on how to use it, four received the training but did not wear the belt, and four more had neither belt nor training. The 12 subjects then took part in experiments in orientation in different environments. Some of the experimental evidence was inconclusive in terms of supporting all the original hypotheses of the project; however, in one particular experiment the belt performed well. This was a "homing experiment" where subjects were blindfold and had to find their way back to the start of a course by the shortest route (Stepankova *et al.*, 2003). The feelSpace researchers found that "at the end of training experimental subjects were making decreased errors only if their belt was on" clearly showing that the information from the belt was being accessed and utilised by the subjects to improve their performance (Açik and Kabisch, 2005).

3.6.3 Teaching Novices Correct Violin Bowing Technique using a vibrotactile fore-arm device

Research by a group at the Open University aimed to develop a system to help learners of the violin to develop a correct bowing technique (van der Linden *et al.*, 2009). This was part of a wider project on investigations into augmented reality called eSense which saw the development of several projects, all of which fed information back to experimenters about their environment in a tactile form (Bird *et al.*, 2008).

As for the feelSpace project described above, these eSense projects were not targeted at partially sighted or blind individuals, but rather aimed to increase the user's awareness of some particular environmental or personal variable or variables. The projects have produced variations of vibration motor 'displays' or feedback devices including a 5x4 array placed on the abdomen used for sensing the position of a rolling ball (Bird *et al.*, 2009). The work on the violin lessons led to a feedback 'jacket' containing seven vibration motors situated on the forearm and the torso. The forearm tactors were arranged to give feedback on the arm position during bowing, while those on the torso were arranged to give feedback on maintaining correct posture while playing the violin (van der Linden *et al.*, 2011).

The results of a user study of the "Music Jacket" showed that four novice violin players made significantly better progress in their bowing technique than four control subjects who were taught conventionally with the same number of training sessions. It is clear then that the vibration motors used were adequate and gave meaningful feedback to the users. The motors employed were 10-mm shaftless DC motors (310-101 Precision Microdrives) which were "firmly attached to the participants' arms using Velcro fasteners" (van der Linden *et al.*, 2011). The motors were interfaced to a computer using an Arduino microcontroller and two TLC 5940 chips (Texas Instruments, n.d.) capable of driving up to 32 vibration units using pulse width modulation. In correspondence with van der Linden about the current research, she emphasised the point that the motors need to be held against the skin, or light clothing over the skin, using some type of elasticated fabric. This ensured clear feedback. On the other hand when the motors were not held tight enough against the body their vibrations were not detected easily. Van der Linden's work shows how a vibrotactile device can be used to give helpful feedback to a user. This information was used in the research to help develop the feedback system used in the prototype navigational device.

3.6.4 Conclusions from HCI Literature

Many different types of non-visual feedback mechanisms exist and have been explored in research and written about in the literature. It was important that any prototype device built during the course of the research employed a mechanism that could be easily detected by the user, and that did not interfere with their cognisance of available cues for navigation. The feelSense project showed that a belt with attached vibrating motors could be used to give feedback about position of obstacles, or change in surface in a plane of orientation easily recognised by an individual walking on a horizontal surface. From van der Linden's work it was clear that the vibrating motors needed to be held against the body with sufficient tightness for the vibrations to be transferred to the skin. The frequency of the motors was ideally to be at around 200Hz, though a wider range was seen as acceptable. The research investigated if such a feedback provided a suitable mechanism for the navigation of urban environments.

3.7 Mobility, orientation and aids used by partially sighted and blind individuals

Chapter 2 dealt with how people with visual impairment navigate difficult urban environments and elicited some of the problems that they have in doing so. This section looks briefly at how such individuals navigate any environment and the cues that they use to do so before moving on to look at some navigation aids.

3.7.1 Mobility

In a survey of 1000 people with visual impairment carried out by the Visual Impairment Centre for Teaching and Research (VICTAR) at the University of Birmingham, 42% of respondents indicated that when they leave their home they do so on foot and 45% stated that they leave their home daily (Douglas *et al.*, 2006). In fact 93% state that they leave their house at least once a week and this percentage has gone up from 1991 when 80% stated that they would go out during the week (Bruce *et al.*, 1991). The rise in mobility, and the reliance on walking as a means of transport, emphasises the need to understand the methods used by blind and partially sighted individuals to orientate themselves, and to ensure that there are sufficient navigational cues available in public places. The VICTAR survey further states that 43% of visually impaired people would like to leave their home more often and 13% cited lack of confidence as being one reason why they would not venture out more. It is important then that public spaces do not exacerbate the lack of confidence by totally removing features that make such spaces navigable by them.

3.7.2 Navigation Skills and Primary Aids

Partially sighted and blind children must be taught to cross roads and negotiate the built environment for the same reasons that sighted children are – the safe completion of walking from A to B. Adults who develop visual impairment during or after childhood have to learn new skills for coping with navigation indoors and outdoors. Learning to use non-visual cues must be taught and then learnt through experience. Such cues include listening for traffic; hearing echoes from buildings; touching and recognising the feel of different surfaces underfoot. To these cues are added the use of primary navigational aids usually listed as the use of a long cane and ownership of a guide dog (for example as listed by the University of Gloucestershire in their guide to "Providing Learning Support for Blind or Visually Impaired Students Undertaking Fieldwork and Related Activities" (Gravestock, 2001)). According to Boone and Boone (1997) the first guide dog training school was founded in 1929 in Morris town, New Jersey while the use of the long cane became widespread during the late 1940s.

Institutions such as The Guide Dog Association and the Royal National Institute for the Blind (RNIB) provide training and information on how to make best use of primary aids and for whom each might be suitable. For example, people with some residual vision may not benefit from a long cane but might be advised to carry a symbol cane to alert drivers and others of their lack of visual acuity (RNIB 2012). Schools that cater for children with visual impairment will teach orientation and mobility to their pupils to enable them to live independent lives. Mobility is described as the ability to safely move around in any environment while orientation is described as "an awareness of space and an understanding of the situation of the body within it" (RNIB, 2012). In other words, orientation involves knowing where one is, and which way one is facing in relation to other features in the immediate and wider geographical location. People with visual impairment must create non-visual memories of routes and layouts of the environment if they are to learn orientation and this too requires training. Recent research suggests that the ability to create such memories and access them may have been underestimated in the past, and that in fact blind and partially sighted people can attain high levels of spatial awareness of routes in urban environments given sufficient training (Golledge *et al.*, 1999).

3.7.3 Secondary Navigational Aids

Secondary navigational aids include tactile maps (Erin, 2009); audible signage such as "Talking Signs" (Bentzen and Mitchell, 1995); various global satellite position systems, for example "The Trekker Breeze" (Phillips, 2011); and various electronic travel aids (ETAs) the majority of which work by actively transmitting a signal (usually infrared or ultrasonic) and interpreting how it is reflected from the environment. Baldwin (2003) writes about the development of robotic aids although these have not yet appeared in public life.

ETAs have been in development since the mid twentieth century and these have had various levels of success. Farmer and Smith (1998) have divided these into several types as follows. Type I aids provide a single output about objects, for example a hand-held ultrasonic obstacle detector ; Type II devices have a multiple output, for example the "Lasercane" which detected drop offs, such as kerbs, and overhead obstacles; Type III devices provide information about the wider environment as well as objects, for example by means of ultrasonic sweeping of the path ahead; and Type IV devices have artificial intelligence of which the "Sonic Pathfinder" was an early example (La Grow, 1999). Table 3.1 displays this information.

ETA s	Type I	Type II	Type III	Type IV
Features	Single output only	Multiple outputs	Primitive object identification possible	Uses A.I.
Detects	Objects	Drop offs and overhead obstacles	Objects and wider environment	Objects in front identified by system as they come near to user
Example	Hand-held ultrasonic detector	The "Lasercane"	The Sonic Guide (no longer available)	The Sonic Pathfinder

Table 3.2 Type of Electronic Travel Aids (after Famer and Smith, 1998)

The success of ETAs in the wider blind and partially sighted community has been rather limited and this has been put down to factors including complexity of use; difficulty of 'putting on' the device; lack of sufficient training; and, for some devices, an overwhelming volume of data being communicated to the user. As early as 1974, researchers were reporting that such devices should be simple to use adding that the user "should not have to 'get dressed up' in them"; and they should have either a tactile or auditory output and if the latter then the audio signals "should not block other aural cues" (Benjamin, 1974).

3.7.4 Conclusions from Literature on Mobility and Aids

People with visual impairment learn skills and techniques for navigating urban environments including the use of the primary orientation aids – the long cane or the guide dog. Any additional aid should complement the use of the primary aids and these skills, and furthermore they should be easy to learn and use. At the same time planners of the built environment need to ensure that suitable cues are embedded in the urban layouts they are designing so that blind and partially sighted pedestrians do not feel excluded from such spaces. The research question of whether existing assistive technologies form the basis of an urban environment navigational system is answered, and it can be concluded from the literature that primary aids are not of themselves sufficient in difficult urban environments, and that although secondary aids do exist these either do not work or have not been widely taken up by the partially sighted and blind community. The questions to be answered are

1 Would a person with a long cane or other assistive technology benefit from having information about the surface they are travelling along available as either tactile or audible feedback?

2 How can a device be constructed that will provide such feedback to the user?

3.8 Conclusions

Chapter 2 describes the problem of difficult urban environments and chapter 3 contains a literature review of relevant research. These two chapters laid the ground for the work carried out in the primary research through the discussion of the problem and the development of the research questions. The research questions are consolidated in the following list.

1 Do existing assistive technologies form the basis of an urban environment navigational system that can tackle difficult urban environments such as Shared Space? (Section 2.10)

2 How can any other information about the urban environment be gathered easily using suitable technology and fed to a blind or partially sighted person in a helpful way? (Section 2.10)

3 Can a device be built which will be under the control of the user and which, with some training, will provide sufficient feedback to a blind or partially sighted user so as to increase their confidence when travelling through a difficult urban environment? (Section 3.2.6)

4 Can a neural network program classify surfaces, if appropriate pre-processing of images can lead to identifiable inputs, and will that classification be carried out at a sufficient speed and provide enough detail to be of use to a blind or partially sighted user? (Section 3.4.4)

5 Can intensity invariant, colour space components be used to compare the current surface on which the pedestrian is standing with the path ahead so as to give a warning of potential change in surface type or upcoming obstacle? (Section 3.5.4)

6 Would a person with a long cane or other assistive technology benefit from having information about the surface they are travelling along available as either tactile or audible feedback? (Section 3.7.4)

7 How can a device be constructed that will provide such feedback to the user? (Section 3.7.4)

To these questions were added the following general ones:

8 Can the overall performance of a person with visual impairment making his or her way across the difficult urban environment be improved through knowing what surface they are on?

9 Can such a device indicate the angle at which the user was moving from one surface to another so that he or she could cross perpendicularly to the edge?

10 What constitutes a successful urban environment navigational system?

As explained in section 4.2, question 9 arose from discussions with the Public Policy Officer of Guide Dogs. Question 8 is a specific question about navigation and the use of information about surfaces. Question 10 is more general and looks for a quantifiable result in terms of what success means.

Chapter 4 Methodology

4.1 Introduction

The challenges of navigating difficult urban environments are explored in chapter 2 while chapter 3 looks at the historical background to vision substitution and at vision and pattern recognition together with human computer interfaces available for people with vision impairment. It is clear from these chapters that there is room to explore the possibility of developing a system that would provide feedback in real time to a user to help him or her to navigate a difficult urban environment. This chapter discusses the approach taken in the development of such a prototype and the decisions taken during the research, together with how the original question was explored. Included in the chapter are descriptions of a site visit to The Netherlands; the formation and distribution of a questionnaire; the development of an experiment to test the concept with the help of a blind volunteer; the possible approaches taken in using machine vision to analyse the problem; and the development of the feedback system which was to give a person with visual impairment valuable information. Posters presented during the course of the research show progress in thinking and can be found in appendix A.

4.2 Discussions with Guide Dogs

During the preliminary stages of developing the thesis, a discussion with a doctor in a low vision clinic at the Royal Victoria Hospital, Belfast led to a meeting with Mr Andrew Murdoch, the Public Policy Officer of Guide Dogs, Northern Ireland. As seen earlier, Guide Dogs have been involved in exploring the development of urban architecture in Europe and particularly the expansion of Shared Space as a system for controlling traffic. Mr Murdoch posed the question of whether or not machine vision could be employed as a solution to the problem of navigation of a difficult urban environment for a person with visual impairment. The issue of crossing a lane of traffic on a line perpendicular to the traffic direction was also discussed. This led to the research question, "Can a device be made that will indicate the angle at which the user was moving from one surface to another so that he or she can cross perpendicularly to the edge?

4.2.1 Conclusions from Discussion with Guide Dogs

From the initial discussions, together with an exploration of the literature on vision substitution and Shared Space, arose the formulation of many of the research questions which are outlined in the introduction and reiterated at the conclusion of chapter 3. Andrew Murdoch continued to be extremely helpful in the development of the research and the formulation of the questionnaire.

4.3 The Site Visit

As mentioned in chapter 2 in section 2.4 'Shared Space', Hans Monderman, a Dutch road traffic engineer and innovator, was instrumental in conceiving the concept of vehicular and pedestrian traffic negotiating a shared urban environment. He helped to design several urban regions in the Netherlands and, in June 2008, a visit was made to Drachten to investigate the concept of Shared Space in the field, and to attend the closing conference of the Interreg IIIB Shared Space project. The photograph in figure 4.1 shows a cyclist negotiating a roundabout in the Laweiplein Square, a famous Shared Space and the location of the conference.

Figure 4.1 Shared Space at Laweiplein Square, Drachten

Delegates at the conference were taken to two Shared Space areas and encouraged to walk around and observe at first hand the response of the traffic to them as they crossed the junctions on foot. It was clear that drivers were used to pedestrians moving relatively freely, and the drivers gave way to them. The issue of disabled users and particularly users with visual impairment was raised during the tour, and it became clear that some provision had been made for partially sighted and blind people. The photograph in figure 4.2 shows raised kerbing that has been maintained around the edge of Laweiplein Square to enable people with visual impairment to use it as an aid to navigation. At another Shared Space junction tactile pavement is incorporated into the design along with a zebra-crossing (figure 4.3). Delegates were informed that these features were not strictly part of the Shared Space concept, but were a concession to groups representing the needs of people with disabilities.

Figure 4.2 Raised kerb at the edge of the Laweiplein Square

Figure 4.3 Tactile pavement and zebra crossing at a Shared Space junction, Drachten

The site visit enabled the author to talk with some of the key personnel involved with the Interreg IIIB Shared Space project including Ben Hamilton-Baillie, who has been involved in the planning of some of the new designs of urban spaces in the United Kingdom including Hamilton Road, Felixstowe; he now "serves on a number of regional design panels, and is a visiting lecturer at the University of Bath and the University of the West of England" (Hamilton-Baillie, n.d.) He and other speakers stressed the "de-cluttering" and improved "civility" of the Shared Space areas while at the same time talked about the improved safety at each site. In a conversation that the author had with him, Hamilton-Baillie expressed his opinion that technological support to partially sighted and blind individuals may well be of benefit in helping them regain confidence in navigating Shared Space.

The sites that the delegates visited reflected good practice in that they were attractive; observed congestion was minimal; and cars, cyclists and pedestrians were using the spaces without apparent difficulties. On the other hand there were no people with disability observed using the spaces at the time of the delegates' visit to each.

Mention was made in the conference of groups representing people with disability and a representative from Suffolk County Council spoke about "engagement with the visually handicapped in Ipswich". While the problems articulated by the Guide Dog Association were recognised, it was felt that Suffolk County Council had consulted with them and made some concessions, specifically in the design of the Shared Space region in the Handford Road area of Ipswich (Pitchford, 2008).

4.3.1 Conclusions from site visit

It is clear that Shared Space is a logical step forward in terms of urban design and the examples visited were observed to be safe to different users. Groups representing the disabled had been considered, but there appeared to be more that needed to be done to restore confidence, particularly for partially sighted and blind people in their attitude towards Shared Space. Moreover, no people with visual impairment were observed using the Shared Space examples that the delegates were shown.

4.4 The Questionnaire

4.4.1 The Purpose of the questionnaire

The questionnaire was formulated to obtain primary evidence regarding the aids and cues that people with visual impairment use to navigate urban environments; to discover how the participants reacted when in a Shared Space or other difficult urban environment; whether they avoided such spaces; their emotional response in such areas; and finally, to ascertain whether they used the same or different cues in such a space. The results of the questionnaire were to be used to help the development of the prototype device, particularly in relation to the human computer interface (HCI) so that the feedback would be complementary to what the participants were already using in such areas.

4.4.2 The Questions

A full copy of the questionnaire can be found in appendix B. As mentioned above, the questionnaire was designed to ascertain what aids and cues the participants use in urban environments and so the first section (section A) asked questions about the ownership of canes and guide dogs as well as general questions on age and vision.

The second section (section B) dealt with the frequency with which the participant went out into a town or city and how often the participant would have been independent and reliant only on his or her aid, be it a cane or a guide dog. Questions were also asked about other external cues that might be used by the participant, such as traffic noise and changes in surface and gradient, and this constituted section C.

The fourth and fifth sections (D and E) dealt with unfamiliar urban environments with the fifth section concentrating on Shared Space and other difficult environments. Both these sections asked similar questions to those asked about familiar environments with the aim of establishing any change in pattern. Finally, participants were given an opportunity to comment on their findings.

The layout of the questionnaire was developed with help from Andrew Murdoch from the Guide Dog Association. It was important that the questionnaire would be in a format easily accessible to screen readers so that participants with visual impairment could access it.

4.4.3 Distribution of the Questionnaire

To establish a base for distribution the questionnaire and its objectives were described in the magazine of the Guide Dog Association in Northern Ireland, and readers invited to e-mail or telephone if they wanted to take part. However, the take up of this was small with only three people responding, and so the Royal National Institute for the Blind was contacted and they sent the questionnaire out to a selected group of recipients asking if they would be happy to take part in the research. The uptake was much higher and twenty five people sent their responses in by e-mail, post or telephone bringing the total up to twenty-eight. Although this number is still small, and is not an entirely random group of blind people from the United Kingdom, nevertheless the results still give a valuable insight into the participants' views and experiences of difficult urban environments.

4.4.4 Analysis and Use of the Results

A full discussion of the questionnaire including analysis of the results is given in chapter 5. As mentioned earlier, the results of the questionnaire fed into the design of the prototype. If it was found that most participants rely on audible cues when negotiating an urban environment, then it would be important to provide feedback in a way that doesn't interfere with this channel of information. (An alternative channel of feedback would be through the use of one or more vibrating motors attached either to a glove or a belt.) On the other hand if some participants were not so reliant on sound, then an audible beep or series of beeps of different pitches or duration might be a suitable means of feedback. The results were also used to confirm the need for further help for people with visual impairment in difficult urban environments.

4.5 Testing of the concept

As related earlier, part of the problem of Shared Space and other difficult urban environments for people with visual impairment is that some, and in some cases many, of the cues that they use for navigation are removed and so they lose confidence in their ability to use the space. It was hoped that using machine vision to give them some feedback about the environment would help to restore some confidence and lead to increased use of more difficult urban environments by blind and partially sighted individuals.

In order to ascertain whether information about the surface they are travelling on would be helpful or not, a concept experiment was proposed in which a blind or partially sighted volunteer would walk through a suitable environment and be given information about the surface, such as might be available from a prototype device. It could then be seen as to whether the information was helpful or not. Such an experiment was to be carried out safely and ideally should have compared the effect of the additional information in some quantitative way. However, it would still have been helpful to see of what value the information gained by a possible prototype would be, and specifically whether the movement from one surface type to another one would be a useful indicator for navigational purposes. Such a concept experiment was carried out and the results are described in more detail in chapter 5 (Section 5.5).

4.6 Development of a Prototype Device based on research

The main elements that were considered in developing a prototype device which could help a person with visual impairment to navigate a difficult urban environment were the following:

- elements for capturing information about the environment;
- elements for processing of the information;
- elements for the extraction of useful data from the information;
- elements for the feedback of the useful data to the user.

Each of these elements or subsystems could have been developed separately, but it was essential that they relate to the others in a coherent way. A brief outline of how each subsystem was considered and developed during the course of the research follows. More detail can be found in later chapters and explanations of how concepts and ideas changed somewhat during the research can be found there.

4.6.1 The input subsystem

It is clear that a person with visual impairment is not able to gain as much information from the visual field as a person without such impairment, and it is reasonable to take as a suitable input an image-capturing device such as a camera in the first instance. Different cameras were tested and the possibility of using light in the near infra-red spectrum was also considered. It was important to find a camera that could provide a real-time feed to the processing sub-system and, for this reason, cameras with either USB or 'Firewire' interfaces were considered. Finally, the position of the camera relative to the user was varied. At first it was thought that a fibre-optic cable could carry the image from the end of a long cane to a camera. This is shown in figure 4.4, an early concept sketch of a possible system. Details of the testing of several cameras and the decision where to finally mount the camera are found in chapter 6 "The Vision Subsystem".

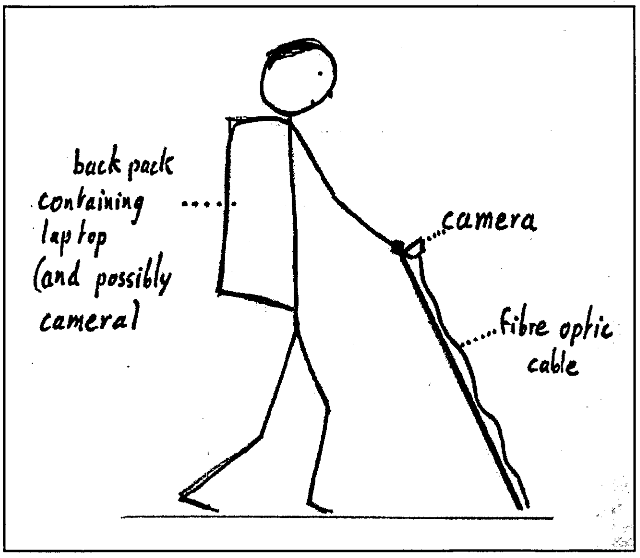


Figure 4.4 Early Concept Sketch of the Prototype

4.6.2 The processing and data-extraction subsystems

Considerations that had to be taken into account when deciding on suitable hardware and software for the processing subsystem, included the need for portability of the system as a whole and the necessity to rapidly build and test variations of software. As it was intended that it should be possible to test the prototype device in the field, a reasonably light-weight solution was required, and a light-weight, but relatively powerful, laptop computer was needed which had both USB and 'Firewire' interface ports so that different cameras could be attached and tested. The decision was made to purchase a Dell XPS M1330 machine with an Intel T5550 dual-core processor running at 1.83GHz and equipped with the Vista operating system.

Microsoft Visual Basic was chosen as the development language as it is a useful language for rapid application development (Clarke, 2001). During the research Microsoft upgraded to Visual Basic 2005 which reinstated some of the features that allow for such development that had been dropped from the .NET version (Langley, 2004). The final programs were written using the latest version then available, Visual Basic 2008. The decision to use Visual Basic.Net had an influence on the choice of camera as it became important to find a camera that could provide a video feed that could be processed using that language. This did not prove to be a problem once suitable drivers were found.

The purpose of the processing and data-extraction subsystems was to examine images of the surface being travelled on by the user and provide useful data to the user. The video feed therefore had to be sampled and analysis made at pixel-level. More information about the development of programs to process the captured images can be found in chapter 6 "The Vision Subsystem".

4.6.3 The feedback subsystem

The history behind feedback subsystems is examined in some detail in chapter 3 "The literature Review". Several researchers have used a variety of tactile and auditory systems and different ideas were considered in the course of the research. One idea was to use text-to-speech to give specific details about the type of surface being navigated at any time, for example the colour or the surface type. Another idea was to give audible beeps differing in frequency or duration to indicate different surfaces, or to use a form of haptic feedback in either a glove equipped with vibrating motors or a belt worn around the waist of the user. These different ideas are discussed further in Chapter 7 "The human computer interface".

4.7 Conclusions

The research questions have been set out in this and the previous chapters. In this chapter methodology of the research in arriving at a proof of the hypothesis has been outlined. The following chapters follow the story of the questionnaire; the development and testing of the prototype device; and how answers to the research questions came about through the development of the functioning prototype device, as well as further analysis of the questionnaire.

Chapter 5 Questionnaire and Concept Experiment Results

5.1 *Introduction*

Two important parts of the research which contributed to the development of the prototype device were the questionnaire and the concept experiment. The distribution and purpose of the questionnaire are discussed in section 4.3, and the idea of the concept experiment is described in section 4.4. This chapter summarises the aims of the questionnaire and provides an analysis of its results together with conclusions that can be drawn from them before describing the results of, and conclusions to be drawn from, the concept experiment.

5.2 *Summary of the aims of the Questionnaire*

The purpose of the questionnaire is discussed in section 4.3.3 and the aims are summarised below:

1. To find out how the participants navigate in familiar and unfamiliar urban environments.
2. To find out how much experience the participants had of Shared Space, Home Zones and pedestrian precincts (for simplicity these three regions are called "difficult urban environments" in this chapter, but not in the questionnaire).
3. To find the emotional response of the participants when in difficult urban environments.
4. To find out if the participants used other aids or cues for navigation in such settings.

5.3 *Results of the Questionnaire*

5.3.1 General Results

The full results can be seen tabulated in Appendix E. Out of 28 participants who completed the questionnaire, 25 said they had been in familiar environments, 22 in unfamiliar and 19 in either Shared Surface Streets; Pedestrian areas; or Home Zones. Of the latter group, a total of 15 reported an experience in a Shared Surface area – further breakdown is depicted in a Venn Diagram (figure 5.1).

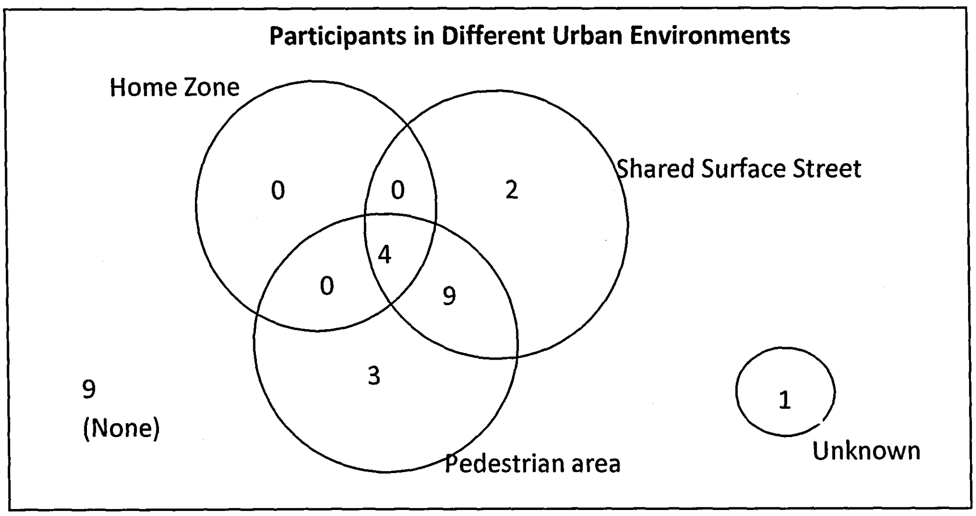


Figure 5.1: Venn Diagram showing experience of participants of Difficult Urban Environments

Participants were asked what cues they used in different areas. They were given a list of ten cues and instructed to place a tick against them for both familiar environments and less familiar ones. They were then asked about their experience in urban environments such as "Shared Surface Streets, Home Zones and pedestrian precincts". They were asked how they orientated themselves in these difficult urban environments but this time no list was provided. The responses are summarised in the table below (table 5.1).

Cue \ Environment	Familiar environment	Unfamiliar	'Difficult' Areas
Ask Directions	20	21	8
Change in Gradient	17	12	3
Change in surface	21	13	1
Familiarisation or partial knowledge of Route	24	11	4
General Listening	22	19	3
Long Cane	11	8	2
Other	7	9	1
Road Crossings	22	15	2
Specific Clues/ landmarks	18	17	7
Noises	14	9	2
Smells	12	8	0

Table 5.1 Table showing results of questionnaire as regards the cues used in different environments

Participants were also asked to describe their feelings when in a difficult area – 17 used words such as "anxious", "vulnerable", "uneasy", "frightened". They were also asked to describe the differences between familiar areas and the difficult areas. Two said they took longer to navigate difficult areas; two mentioned more difficulty with the guide dog; thirteen mentioned other specific problems with difficult spaces including knowing who has right of way; ascertaining which direction the traffic was moving; finding a safe place to stand; finding obstacles in the way or being more likely to lose one's bearings or miss side streets and openings.

5.3.2 Analysis of use of auditory channel

One purpose of the questionnaire was to establish to what extent the participants rely on their hearing to perform navigational tasks in urban environments. This is to establish whether audio feedback can be used by a navigational aid or if it should be avoided. The results of the questionnaire show that for familiar environments 22 out of the 28 participants use general listening to help them, and of those 14 listen out for specific noises to help them find their way. In percentage terms that is 78% and 50% respectively. In unfamiliar environments 19 out of the 24 participants who visit such environments use general listening skills, and 10 listen out for specific noises. This is 79% and 42% respectively.

These percentages, particularly for the general listening cue, show that hearing is very important for a significant majority of the participants involved in the task of finding their way. Consequently, it would be unwise to use an auditory feedback system which might detract from the ability to use such cues, and any prototype should be equipped with an alternative feedback system such as touch.

5.3.3 Statistical analysis of cues in familiar and unfamiliar environments

From the data collected it is clear that most participants who had been in a difficult urban environment felt uneasy, anxious or disorientated. It appears that they used much the same cues in both familiar and in unfamiliar environments and the Chi-Square Test is used below to ascertain whether or not the participants were influenced in their choice of cues, depending on whether they were in a familiar or an unfamiliar environment. The Chi-Square Test is used for two reasons. Firstly, it is a well recognised statistical test for non-parametric data, that is data where assumptions about any of the parameters of the distribution of the population cannot be reliably made (Owen and Jones, 1982, p. 330). Secondly, it is used in data such as that found here to compare the observed results of an experiment with those that would be expected if the results were left to chance. These expected results are calculated from probabilities of outcomes. A so called 'null hypothesis' is used which states that there is no difference between the observed and the probability calculated, expected results.

In this case the null hypothesis was that the participants' use of cues was independent of their environment; in other words that the participants were not influenced in their choice of cues depending on whether they were in a familiar or an unfamiliar environment.

To test the null hypothesis a table was drawn up to count how many individuals used the same cue in both environments (or did not use the same cue in both) and how many used one cue in one environment but not in the other. This was the observed pattern and is shown in table 5.2.

	Observed values	
	same	different
Ask Directions	21	3
Change in Gradient	20	4
Change in surface	18	6
Familiarisation or partial knowledge of Route	18	6
General Listening	22	2
Long Cane	22	2
Other	18	6
Road Crossings	20	4
Specific Clues/land	19	5
Noises	19	5
Smells	17	7

Table 5.2 Comparison of use of cues between environments

'Same' means that the cue was either used in both environments or else not used in both. 'Different' means that the cue was used in one environment but not the other.

To calculate the expected results a table was made showing which cues had been used in which environments. Then the expected results were calculated by using the formulae

E_same = (AC+BD)/N (expected number same use of cue or non-use of cue in both environments)

E_different = (AD+BC)/N (expected number different use of cue in two environments)

where A = number of participants who used the cue in a familiar environment

B = number of participants who did not use the cue in an familiar environment

C = number of participants who used the cue in a familiar environment

D = number of participants who did not use the cue in an unfamiliar environment

N= total number of participants who visited both environments = A+B = C+D

Table 5.3 shows this information.

	familiar		unfamiliar		expected same	expected different
	Used cue	Didn't use cue	Used cue	Didn't use cue		
Ask Directions	18	6	21	3	16.5	7.5
Change in Gradient	16	8	12	12	12	12
Change in surface	20	4	14	10	13.33333	10.66667
Familiarisation or partial knowledge of Route	22	2	18	6	17	7
General Listening	21	3	19	5	17.25	6.75
Long Cane	10	14	8	16	12.66667	11.33333
Other	9	15	11	13	12.25	11.75
Road Crossings	21	3	17	7	15.75	8.25
Specific Clues/land	18	6	17	7	14.5	9.5
Noises	13	11	10	14	11.83333	12.16667
Smells	11	13	8	16	12.33333	11.66667

Table 5.3 Observed usage of cues and expected same and different results

Finally a Chi Squared Test calculation was carried out as shown in table 5.4.

Observed values		Expected values		O-E		(O-E)^2/E	
same	different	same	different	same	diff	same	diff
21	3	16.5	7.5	4.5	-4.5	1.227273	2.7
20	4	12	12	8	-8	5.333333	5.333333
18	6	13.33333	10.66667	4.666667	-4.66667	1.633333	2.041667
18	6	17	7	1	-1	0.058824	0.142857
22	2	17.25	6.75	4.75	-4.75	1.307971	3.342593
22	2	12.66667	11.33333	9.333333	-9.33333	6.877193	7.686275
18	6	12.25	11.75	5.75	-5.75	2.69898	2.81383
20	4	15.75	8.25	4.25	-4.25	1.146825	2.189394
19	5	14.5	9.5	4.5	-4.5	1.396552	2.131579
19	5	11.83333	12.16667	7.166667	-7.16667	4.340376	4.221461
17	7	12.33333	11.66667	4.666667	-4.66667	1.765766	1.866667

Sums by column	59.33333	-59.3333	27.78642	34.46965

total	62.25608

Table 5.4 Chi Square Test calculations

Table 5.4 shows a total Chi Square Test value of 62.26. Since the number of degrees of freedom for this table is 11, the 5% significance value for rejecting the null hypothesis is 19.68. Indeed the 0.1% significance level is 31.26 so the null hypothesis can be reliably rejected.

The conclusion from this analysis is that the participants' use of cues in a familiar environment does influence their use of cues in an unfamiliar one, and the implication of this result is that partially sighted and blind people tend to use the same cues whether they are in a familiar urban environment or a non-familiar one.

Further analysis can be made by looking at each cue individually to ascertain whether or not it is used differently in unfamiliar environments compared with familiar environments. This involves testing each row separately using a chi square distribution on 1 degree of freedom. For example, with 'Ask Directions' the hypotheses would be

H_0 (null hypothesis): Whether people ask directions in a familiar environment does not influence whether they ask directions in an unfamiliar environment.

H_1 (alternative hypothesis): Whether people ask directions in a familiar environment does influence whether they ask directions in an unfamiliar environment.

In this case 1.23 + 2.7 (giving 3.93) is compared with a Chi Square on 1 degree of freedom and for this cue the null hypothesis at the 5% significance level (a value of 3.84) can be rejected. So the null hypothesis is rejected and the conclusion is that people will continue to ask for directions in an unfamiliar environment if they do so in a familiar one.

Table 5.5 shows this approach with calculations for all the cues and includes a column for testing if the null hypothesis can be rejected at the 5% significance level and at the 10% significance level.

familiar		diff		observed		expected		o-e		(O-E)^2/E		totals	5%	10%	
y	n	y	n	same	diff	same	diff	same	diff	same	diff		>3.84?	>2.706	cue type
18	6	21	3	21	3	16.5	7.5	4.5	-4.5	1.2273	2.7	3.93	Y	Y	Ask Directions
16	8	12	12	20	4	12	12	8	-8	5.3333	5.33	10.7	Y	Y	Change in Gradient
20	4	14	10	18	6	13.33	10.67	4.67	-4.67	1.6333	2.04	3.68	N	Y	Change in surface
22	2	18	6	18	6	17	7	1	-1	0.0588	0.14	0.2	N	N	Familiarisation or partial knowledge of Route
21	3	19	5	22	2	17.25	6.75	4.75	-4.75	1.308	3.34	4.65	Y	Y	General Listening
10	14	8	16	22	2	12.67	11.33	9.33	-9.33	6.8772	7.69	14.6	Y	Y	Long Cane
9	15	11	13	18	6	12.25	11.75	5.75	-5.75	2.699	2.81	5.51	Y	Y	Other
21	3	17	7	20	4	15.75	8.25	4.25	-4.25	1.1468	2.19	3.34	N	Y	Road Crossings
18	6	17	7	19	5	14.5	9.5	4.5	-4.5	1.3966	2.13	3.53	N	Y	Specific Clues/land
13	11	10	14	19	5	11.83	12.17	7.17	-7.17	4.3404	4.22	8.56	Y	Y	Noises
11	13	8	16	17	7	12.33	11.67	4.67	-4.67	1.7658	1.87	3.63	N	Y	Smells

Table 5.5 Individual rows tested using Chi Square Test

It can be seen from the table that for 6 out of the 11 cues the null hypothesis is rejected at the 5% significance level, and all but one rejected at the 10% significance level. The only cue that could be said to be uninfluenced in its use in an unfamiliar environment by whether the participant used the cue or not in a familiar one, is that of knowledge or partial knowledge of the route. This is somewhat to be expected as this cue is dependent on the familiarity of the location, so logically is going to be dependent on whether or not the location is familiar or not, rather than the participant's approach in a familiar setting. A similar argument might be put for the use of specific clues or landmarks, and here too the null hypothesis is accepted at the 5% significance level, if only just. The other two cues where the null hypothesis is accepted at this level are 'change in surface type' and 'road crossings'. The latter is arguably because the participant must have knowledge of the number of road crossings in order to use the cue effectively. In an unfamiliar environment he or she may have been told how many crossings to look out for, otherwise the participant could not use the cue. Change in surface is less clear, but could be related to using the surface type as a guide to finding familiar landmarks.

It could be argued that because each row has only one degree of freedom, Yates' Correction should be employed to prevent overestimation of the statistical significance. However, the correction is meant to be applied to 2x2 tables, which is not the case here, and in any case tends to overcorrect producing type II errors where the null hypothesis is not rejected when it should be (Haviland, 1990).

5.3.4 Statistical analysis of all three environments

Table 5.6 is a contingency table containing data from all three types of urban environments. Data has been grouped so that no cell contains a value less than 5 (Owen and Jones, 1982, p. 335). Cues that are similar in some aspects have been grouped together, for example, knowledge of the route has been grouped with use of road crossings and general listening has been grouped with specific noises (although smells and the 'other' category have also been added here).

Observed Cues	Familiar environment	Unfamiliar observed	Difficult Urban Environments	Totals
Ask Directions	20	21	8	49
Change in Gradient or surface or use of long cane	49	33	6	88
Familiarisation or partial knowledge of Route or road crossings	46	26	6	78
General Listening, noises, smells or other	55	45	6	106
Specific Clues/ landmarks	18	17	7	42
Totals	188	142	33	363

Table 5.6: raw results grouped

Table 5.7 shows the same categories of cues with values calculated using the ratios of the totals of each row and column. These are the values that would occur in each environment if there were no bias, and are known as the 'expected' values. Finally table 5.8 shows the Chi Square Test analysis in which the likelihood of the observed results differing from the 'expected' results is calculated.

Expected Cues	Familiar environment	Unfamiliar observed	Difficult Urban Environments	totals
Ask Directions	25.37741	19.16804	4.454545	49
Change in Gradient or surface or use of long cane	45.57576	34.42424	8	88
Familiarisation or partial knowledge of Route or road crossings	40.39669	30.5124	7.090909	78
General Listening, noises, smells or other	54.89807	41.46556	9.636364	106
Specific Clues/ landmarks	21.75207	16.42975	3.818182	42
Totals	188	142	33	363

Table 5.7 Expected values using ratios

(O-E)^2 / E	Familiar environment	Unfamiliar observed	Difficult Urban Environments	Totals
Ask Directions	1.13946	0.175086	2.821892	4.136439
Change in Gradient or surface or use of long cane	0.257274	0.058926	0.5	0.816199
Familiarisation or partial knowledge of Route or road crossings	0.777218	0.667326	0.167832	1.612376
General Listening, noises, smells or other	0.000189	0.301268	1.372213	1.67367
Specific Clues/ landmarks	0.647203	0.019792	2.651515	3.31851
Totals	2.821344	1.222398	7.513452	11.55719

Table 5.8 Chi Square Test calculations

Since table 5.8 is a 3 x 5 contingency table the number of degrees of freedom is given by 2 x 4 = 8.

The null hypothesis is that the cue used in each case is independent of the type of the environment.

The upper 5% point significance level of a chi square distribution with 8 degrees of freedom = 15.51, and the observed value (11.6) does not exceed this upper 5% point (15.51).

The conclusion is once again that blind and partially sighted people tend to use the same types of cues for navigation in difficult urban environments as they do in unfamiliar and familiar ones.

5.3.5 Conclusions of statistical analysis

Since the blind and partially sighted participants tend to use the same type of cues when navigating different urban environments, it is reasonable to assume that the population of blind and partially sighted people will find it difficult to adapt to urban environments where these cues have been removed.

It would therefore be considered unreasonable for planners to remove too many of these cues when planning new urban environments, and the needs of this group must be taken into consideration in designing any urban environment, whether it be a Shared Space; a Home Zone; a pedestrianised area in a town or city; or even an indoor Shopping Centre.

5.4 Overall Conclusions drawn from the Questionnaire

The aim of the questionnaire was to establish the methods that people with visual impairment use to navigate the urban environment. There is little doubt from the results that a wide range of cues are utilised, and that individuals learn to use these cues in the course of their daily lives. The skills learnt can then be transferred to new environments, but the questionnaire results show that the participants found it difficult to adapt to new environments, tending to rely on their learned skills. Furthermore, many of the participants felt vulnerable or unsafe in difficult urban environments (listed as "Shared Surface Streets, Pedestrian areas and Home Zones"). This finding confirms investigations carried out by the Guide Dog Association.

Finally, as stated in section 5.3.2, the sense of hearing is important to people with visual impairment as a means of finding their way in urban environments. Therefore a device to help partially sighted and blind people to find their way should provide information to them in a tactile form, or at least give them the choice of switching between auditory feedback and tactile feedback.

5.5 Testing the concept

5.5.1 Description of the concept experiment

In order to find out if a person with visual impairment could make use of surface information, a thought experiment involving a blind volunteer, Mr Eddie Chittick, was carried out. In this experiment, Mr Chittick walked across different surfaces in a local town and information about the surface on which he was walking was relayed to him by voice. Mr Chittick is a guide dog owner, although he also uses a long cane. The photographs show him in different parts of the town approaching surface changes (figure 5.2).

Figure 5.2 Photographs of Eddie Chittick crossing onto a different surface

Mr Chittick found that the concept of having information about the surface he was travelling on was different to his standard way of navigating using a guide dog. As discussed elsewhere (section 2.7), a guide dog is generally trained to take note of the kerb, and will keep moving forward until a junction is reached when it will stop and await instructions from owner. So the owner ignores surfaces altogether, and is primarily only concerned with junctions, knowing that the dog will avoid any obstacles in the way. A long cane, on the other hand, is used to convey information about the surface, and any obstacles, to the user. The user takes surface changes in texture into account while navigating.

In the case of the blind volunteer, Mr Chittick found that being told about surface changes would indeed have helped him not to step off a path (as seen in figure 5.2), and could be useful in finding a particular premise entrance where there was a distinct change in surface colour outside that entrance (figure 5.3).

Figure 5.3 Photograph showing change in surface colour outside a particular building.

The colour change could not be picked up by a long cane as there is no change in texture. The blind volunteer is being told about the surface change.

5.5.2 Conclusions from concept experiment

The concept experiment showed that information about the surface over which a blind or partially sighted person is walking could potentially be of benefit to him or her. Such information is not in itself going to provide enough guidance in a difficult urban environment to enable the person to navigate independently, but may well give sufficient support such that the user of a suitable device could have more confidence in their ability to traverse the space successfully.

Chapter 6 : The Vision Subsystem

6.1 Introduction

The vision subsystem consists of three main components, the camera, the video acquisition interface and the signal processing software. This section will discuss the cameras used in the research and the interface technology before considering the development of the two variations of the signal processing software in some depth. A block diagram of the vision subsystem is shown in figure 6.1.

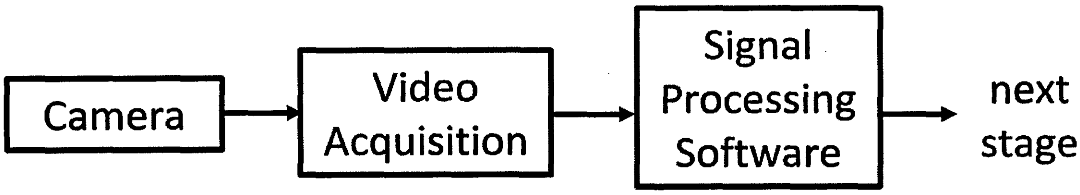


Figure 6.1 the Vision Subsystem

6.2 The Camera

6.2.1 Requirements

Various cameras were tested during the research in terms of quality, ease of use and resolution. The original idea was to provide the user with a camera which would communicate wirelessly with the user, and which would provide sufficient information so that the surface type could be identified from the video feed when processed by a software program. During experimentation it became apparent that the requirement for a clear image outweighed the requirement for wireless communication and so that criterion was dropped.

6.2.2 Camera 1: The 'Eyetoy'

Initial experiments were carried out with a simple webcam that was originally designed to work with the 'PlayStation 2' (PS2) – the 'Eyetoy'. This camera is made by Logitech and incorporates a low voltage CMOS sensor the OV7648 (Omnivision, 2005). The Eyetoy is capable of sending a 320×240 pixel, compressed video at 60 frames/sec in well-lit conditions and uses the universal serial bus (USB) port under the USB 1.1 protocol as an interface (Marks, 2010). The camera comes in a casing with blue and red LEDs at the front which are used in games. These gave unnatural lighting conditions when the camera was used as a webcam and were covered over with black tape (Figure 6.2).

Figure 6.2 The PlayStation 2 'Eyetoy' camera with tape over the LEDs

Drivers to allow the 'Eyetoy' to run with Windows XP and Windows Vista were readily available on the Internet. The camera worked well except when lighting conditions were relatively low. This resulted in a low contrast ratio and pictures of a grey, washed-out constituency.

6.2.3 Camera 2: Wireless Pinhole Camera

Due to the original, perceived requirement to have a wireless camera, the next camera that was used was the wireless pinhole camera supplied by Pakatak (see figure 6.3). This camera transmits data at a frequency of 2.6GHz and the dedicated receiver uses the USB interface (USB 1.1) to communicate with the laptop. The same program elements (drivers and interface subroutines) that were downloaded for the Eyetoy functioned with this camera. The resolution is 720 x 576 but the quality of the pictures was found to be not good.

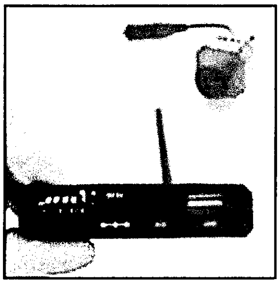


Figure 6.3 The wireless pinhole camera (Pakatak, n.d.)

6.2.4 Camera 3: Quick Cam Pro for Notebooks

Discussions with Dr George Redpath from CEM Tyco systems, a company that manufactures and maintains video surveillance equipment, helped with the decision to purchase a modern web-cam with moderate resolution but a reasonable ability to function at lower light levels. The Logitech "Quick Cam Pro for Notebooks" (figure 6.4) was found to work well with the video processing software.

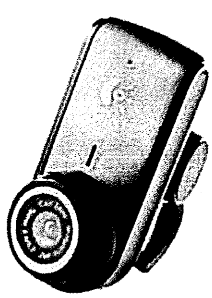


Figure 6.4 Logitech Quick Cam Pro for Notebooks

The Quick Cam Pro camera comes with built-in software 'RightLight' which automatically provides a balanced image under a variety of lighting conditions from dim background light to bright sun-light.

This function can be disabled but for most of the experiments carried out it was left on and performed well.

The camera supplies video at 30 frames per second, and the resolution can be set at 320 by 240, 640 by 480, or 960 by 720. It was found that 320 by 240 gave sufficient resolution for the software to process the image and produce a useful output, without the computer slowing down and becoming unable to keep up with the frame-rate. The camera interfaces with the laptop through the USB port under the USB 2.0 protocol.

6.2.5 Infrared Camera

One suggestion to help partially sighted and blind people to navigate was the painting of an infrared (IR) reflecting paint to mark out a safe route. This paint would be invisible to the naked eye, but would be picked up by a suitable camera; hence it would be in keeping with the philosophy of Shared Space as vehicle users would still perceive the area as part of the shared area. Another suggestion was that increased discrepancy between surfaces might occur in the IR spectrum, and so the neural network program might perform better. In order to carry out a preliminary investigation of these ideas the Eyetoy camera described above was adapted so as to be sensitive to near IR light.

Most CMOS sensors are already sensitive to the infrared spectrum in the near IR spectrum; up to a wavelength of around 1100nm (Gilblom and Yoo, 2004), and an IR cut-off filter is usually employed for normal visible light camera use. In order to adapt the Eyetoy camera it was necessary to remove the IR cut-off filter and replace it with a visible light filter. The IR filter turned out to be painted onto the lens and it had to be scrapped off which damaged the lens slightly. A makeshift visible light filter was made from cutting a 35mm photographic negative which had been overexposed. Although this method was not ideal, a working IR camera was made.

The IR camera was not found to give any improvement as far as the neural network program was concerned. It was also decided that painting a line, visible only to an IR camera, in a public area would be unsustainable as a solution to navigation in a difficult urban environment as it would be too difficult to maintain. The ideal solution to the navigation problem should not rely on measures which have to be applied specially to an area and which need to be repainted or reapplied.

6.3 Video Acquisition Interface

6.3.1 Introduction

The vision system requires that the video acquisition interface be transparent to the user. The ability of the camera to capture the image must be matched by the ability of the interface to feed the video signal seamlessly to the signal processing software. This section will consider the physical interface together with some of the drivers and software libraries available.

6.3.2 Firewire IEEE 1394

It was initially anticipated that the Firewire, IEEE 1394, interface might be used. According to "Vision Systems Design" this standard, developed by Apple, is one of the most popular interfaces for machine-vision (Wilson, 2008). Firewire provides a dedicated high-speed connection (up to 800 MBps for IEEE 1394a) which, unlike USB, will not vary in bandwidth since it does not rely on the computer CPU for management of its protocols (Q-Imaging, 2006). However, there are not many Firewire webcams available on the market and most web-cams available use the USB 2.0 port as explained below.

6.3.3 USB 2.0

The universal serial bus (USB) port is available in most laptops manufactured today and consequently most webcams use this standard as the interface. USB 2.0 allows up to 480 Mbps (so called "High Speed USB", Intel, n.d.). USB is designed as a master-slave interface channel and the bandwidth is not guaranteed. The host machine must determine a suitable bandwidth based on demand and usage. Generally, though USB is a suitable solution for webcams.

6.3.4 'Eyetoy' Drivers

In 1997 Microsoft released version 1.00 of AMCAP, a video capture and preview application. This was available together with the necessary drivers to run various webcams, including the 'Eyetoy', under the Windows Vista and Windows XP operating systems, and meant it could be interfaced to the laptop and to the Microsoft Visual Basic programs being written for the research. (Solutions to running the 'Eyetoy' camera with Windows 7 are available, but it is becoming increasingly difficult to find suitable drivers as the PS2 'Eyetoy' is now considered a legacy technology).

6.3.5 Other Drivers

Each of the other cameras that were experimented with came with their own drivers. The Logitech Quick Cam Pro for Notebooks drivers worked well in that it was possible to run the complementary 'settings' software at the same time as running the Visual Basic programs. Consequently, automatic light-level detection (a built in program called 'RightLight' referred to above) could be enabled as could the 'auto-focus' function and the 'zoom' function. A Microsoft camera, 'LifeCam HD-6000', also came with software called 'TrueColour' which promised to deliver bright images in a variety of lighting conditions. Unfortunately it was not possible to access this program while the Visual Basic programs were running, and so the Microsoft camera was abandoned in favour of the Logitech one.

6.3.6 Software Libraries

The software dynamic link library 'avicap.dll' is the Video for Windows Application Programming Interface (API) written by Microsoft and "supports streaming video capture and single-frame capture in real-time" (Microsoft, 2011). The library can be downloaded from various sites on the Internet, but Microsoft no longer supply it as they encourage programmers to use 'DirectShow'. The Visual Basic programs written during the research used the avicap library to preview the video from the cameras, and to provide the software with individual bitmap images which could then be analysed.

6.4 Signal Processing Software

6.4.1 Introduction

Some method for helping people with visual impairment to orientate themselves in a new urban environment is needed if traditional markers such as traffic crossings and even a raised pavement have been removed. It was envisaged that an indication of the surface over which the person was travelling would be a helpful cue in the navigation of the urban area. Any change in surface colour or type could well be indicative of a change in the designation of the region, and perhaps show that the person was moving into an area more likely to be frequented by vehicular traffic.

Two programs were developed and written by the author to process the signal from the video camera. The first, a neural network program, was designed to differentiate between surface types. The second was designed to inform the user if a change in surface type was about to occur as the user walked forward, and is a real time area comparison program. The neural network program was developed first and its development and performance are described in section 6.5 below. When this program proved to be limited in its ability to differentiate surface types in different lighting conditions a second approach was investigated – the area comparison program and this is described in section 6.6.

6.5 *The Neural Network Program*

6.5.1 Introduction

The development of software capable of categorising surface types was intended to help the user to navigate a difficult urban environment more easily as he or she received feedback on the ground surface type and notification of any changes in surface. As mentioned in chapter 2, neural networks have been used to classify different elements since the second half of the twentieth century and are particularly useful when the elements have characteristics that make classification difficult.

At first sight, as with many machine vision problems, the question of how to identify many surfaces seems relatively simple. However, images of surfaces of roads, pavements and other terrains will not have the same parameters in terms "scale, intensity and contrast to name only a few" (Stolpmann and Dooley, 2000). In other words before a system can start to categorize the images it is usually necessary to perform some pre-processing so that, for example, the images show the surfaces from the same height. Ojala and Pietikäinen mention that there are only a limited number of working texture analysis systems and attribute this to differences in "orientation, scale or other visual appearance" existent within the same texture and to the high levels of computational complexity required to solve some of the proposed systems. Pre-processing of images can help eliminate variations, however, computational power is required to do so and this can introduce a time delay.

A neural network was proposed as a solution with input data from the red, green and blue colour properties of the images of the different surfaces. Such an input set would then be independent of orientation of the image. In the current system, spatial scale would also be invariant as it was proposed that the camera would be the same height above each surface. (Initially the camera was to receive an image from the end of a long cane, as shown in figure 4.4, but this was later changed so that the camera would be attached to the user's belt). Illumination variance is another factor, and one that can be compensated for by performing a mathematical transform of the RGB data as seen in work by, for example Zickler *et al.* (2008). Alternatively, the effect of changes in lighting on the surface characteristics could be built in to the network memory so that the network would hold, not only a memory of different surface types, but also of those surfaces in different lighting conditions. This would be important if the user moved from an area of sunlight into an area of shadow while traversing the environment. It was proposed then to use a neural network to classify the different surface types that are commonly used in Shared Space, Home Zones and other difficult urban areas.

It was planned that different surface types would be stored as an array of weights in the neural network and their values determined through a training process before being used during normal run-time to ascertain a best match between the surface type under surveillance and that encountered during training. The neural network would save a wide variety of common surfaces and how they appear in different environmental settings (such as in different lighting conditions or when wet).

It was envisaged that as the user passed through the Shared Space, he or she would be warned when passing from one surface type to another. One long-cane user commented that a long-cane can pick up surface changes when there is a change of texture. However, urban environments commonly use the same material with different colours in different zones, and this is where a neural network could provide more information than a long-cane. Several versions of the neural network program were written and tested and these are explained in the following sections.

6.5.2 The first neural network program

To gain an initial understanding of the problem a number of close-up photographs of different surfaces were taken using a digital camera and the RGB (red, green, blue) colour space considered. It appeared that there was a significant difference in the RGB space between photographs of concrete, tarmac and paving stone. As a first step towards developing a more comprehensive neural network, the decision was taken to initially develop one which could differentiate between these three surface types. Figures 6.5, 6.6 and 6.7 show images of these three surface types and the RGB space for each. As mentioned above, the RGB colour space is useful as an input as it provides orientation invariance and since the camera is at a fixed height, spatial invariance as well.

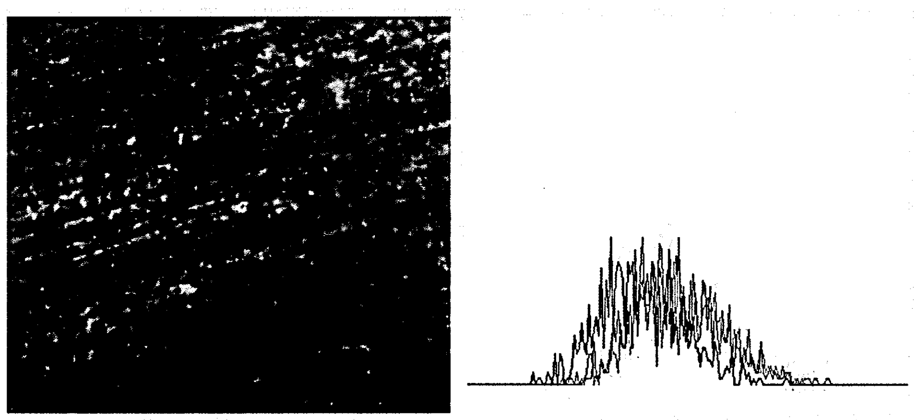


Figure 6.5 Concrete Path and RGB Graph

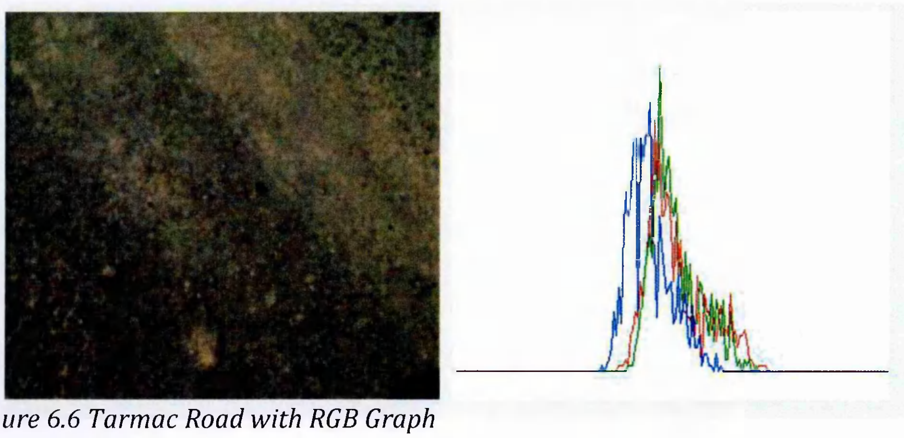


Figure 6.6 Tarmac Road with RGB Graph

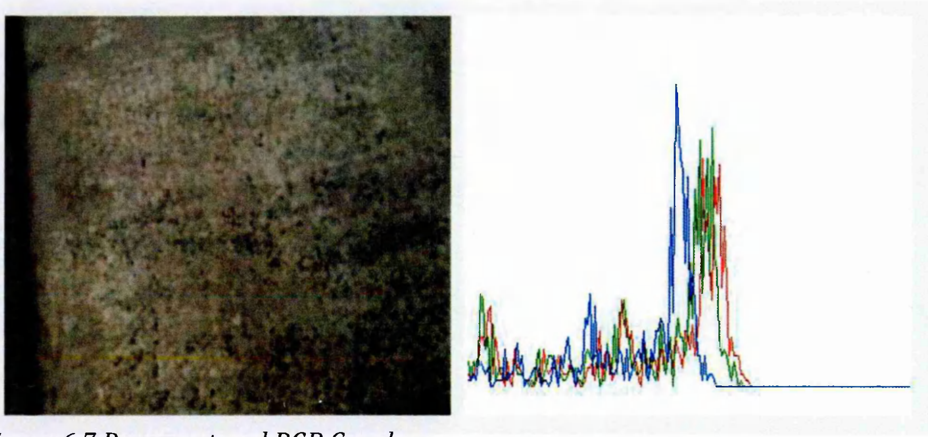


Figure 6.7 Pavement and RGB Graph

In order to design the neural network, decisions had to be taken regarding the number of inputs; the number of layers in the network; and the number of outputs. To allow some room for expansion four outputs were written into the software. Work by Heaton indicates that the number of sub-layers (hidden layers) required in a network depends to some extent on the complexity of the input data (Heaton, 2008, p152), but in general a network with fewer layers has the advantage of offering a somewhat faster processing time and a faster training period. Figure 6.8 depicts a generalised neural network with one hidden layer.

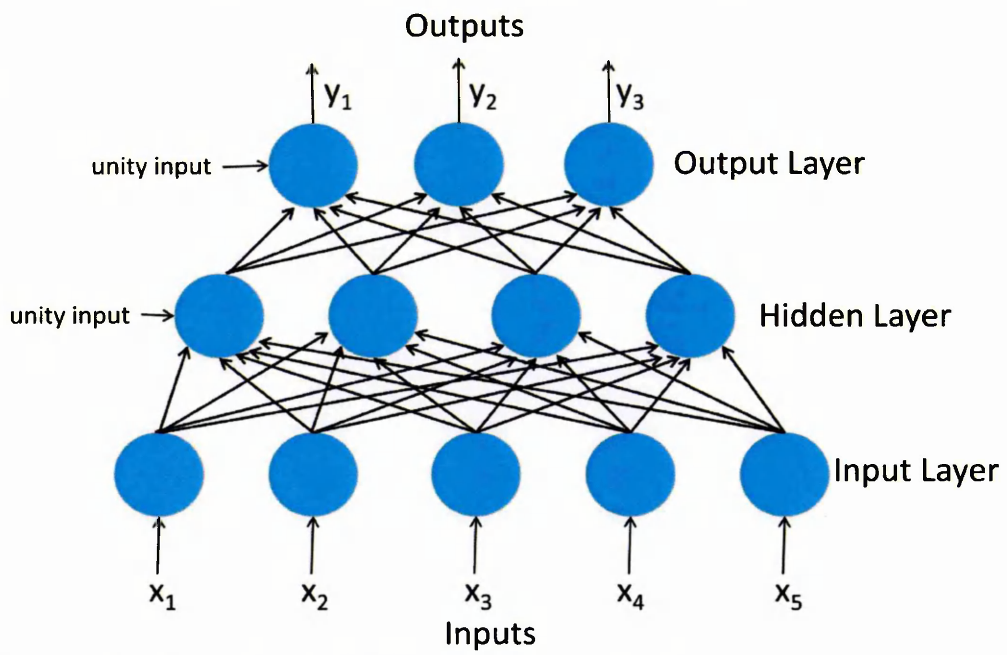


Figure 6.8 Diagram of a neural network with a hidden layer

Unity inputs are required for correct performance of the network

The first program written by the author consisted of three inputs; no 'hidden' or sub-layer; and the four outputs as explained above for concrete, tarmac, paving stone and other. The three inputs corresponded to the RGB (red green blue) components of the image under consideration, as derived from the 32-bit ARGB colour structure of each pixel. (The ARGB structure combines an 'alpha' channel representing opacity with the RGB information; the alpha channel was not used). The three inputs were assigned a value depending on the number of pixels in the image which showed a red, a green or a blue pixel saturation level greater than a set threshold value. RGB saturation varies between 0 and 255; the program took the threshold as a saturation level greater than 150.

This first program was written in Visual Basic using Microsoft Visual Studio Net 2003 and consisted of a simple network which used the RGB input from a selection of photographs of road surfaces. The program was able to compare photographs of different surfaces and derive the RGB components from each pixel. These were then fed as inputs into the neural network using a method known as back-propagation to train the network. The photographs were taken using a digital camera and loaded into the program to be analysed one at a time. Each pixel was analysed by the program and the RGB information extracted and counted.

The initial program was very limited in that training could only be done manually one photograph at a time, and the target outputs (concrete, tarmac or paving) had to be manually entered for each photograph; figure 6.9 shows the interface form of the initial program. The photographs were loaded individually, and then each pixel in the image analysed. A running total was kept of each red, green or blue argument for that pixel above the threshold. These three numbers were then used as the inputs to the network. During the training process the outputs 1, 2, 3 and 4 were set to 1 or 0 depending on the surface in the current photograph. Output 1 was set to 1 and the other outputs to 0 if the surface is concrete; output 2 was set to 1 and the others to 0 for tarmac and output 3 set to 1 and the others to 0 for paving. Backpropagation was used to adjust the weights. This had to be repeated many times using the training set of photographs to train the network. Once the network had been trained a new photograph could be loaded and the 'analyse' button pressed. The outputs were then calculated from the neural network. The output closest to 1 indicated the network's match to the photograph –in figure 6.9 output 2 is the highest value indicating that the network has matched the photograph to tarmac. However, the value is not that close to one (0.529), so the network does not appear to have performed that well. The network itself was very simple (three input and four output nodes), and it was thought that this was why the results were not clear. However, the program did provide an insight into the challenges of writing a neural network program as well as showing that weights could be set according to information derived from the photographs. The next step was to write a more complex program that could analyse information from a video feed.

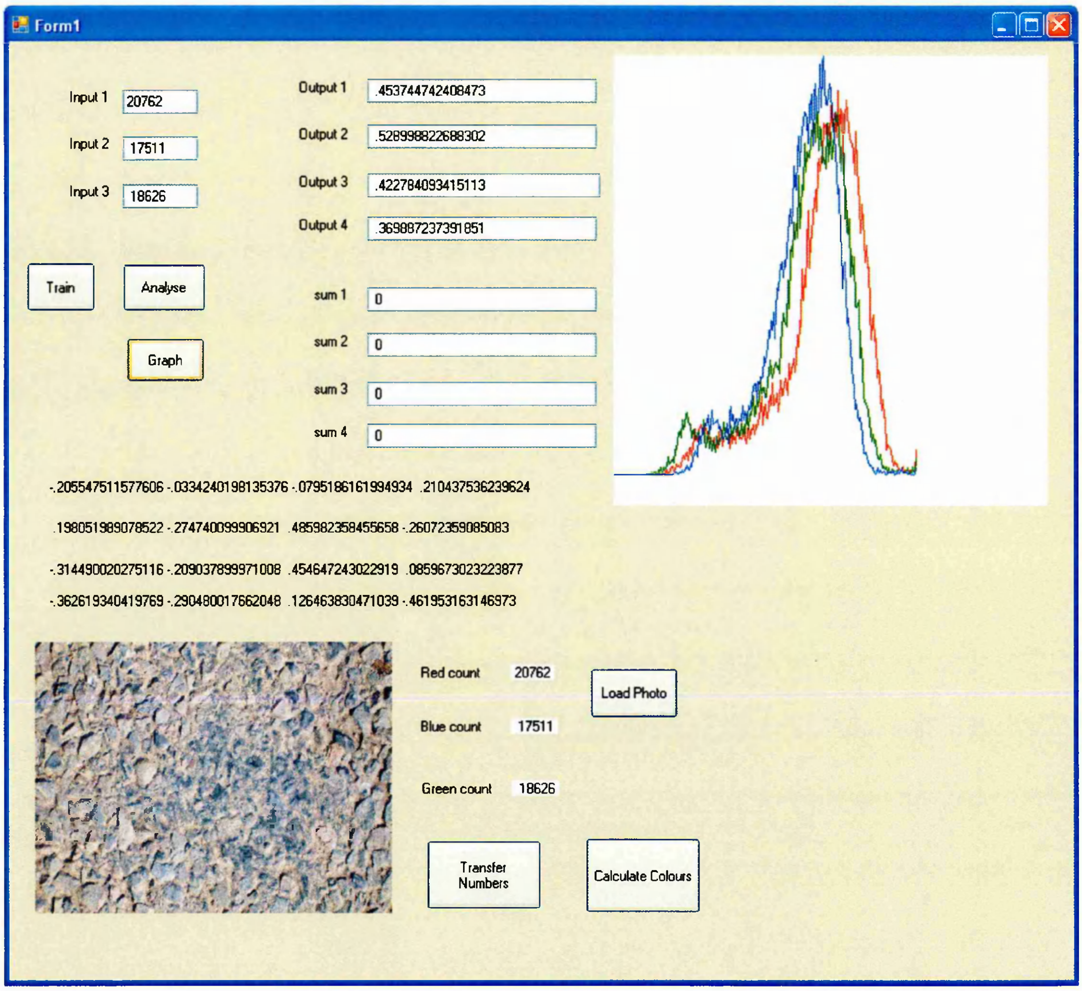


Figure 6.9 Simple neural network interface form

In training mode the photos are loaded using the "Load Photo" button, and the RGB information extracted using the "Calculate Colours" button. The results are loaded into the input text boxes at top left – the button "Transfer Numbers" carries out this step. Finally the "train" button can be pressed to perform back-propagation of the data and storage of the weights.

The four outputs correspond to chosen surface types – concrete (output 1), tarmac (output 2), paving (output 3) and a fourth which was to be assigned later. An output value close to unity indicates the surface type recognised by the network.

In run-mode the loaded photograph can be analysed by the network by pressing "Analyse" and the results appear in the Output text boxes.

At any time a RGB graph of the photograph can be produced by pressing "Graph"

6.5.3 The second neural Network program

The subsequent program, written using Microsoft Visual Basic 2005 Express, allowed for the analysis of individual frames captured from a web-cam. In training mode the surface currently being walked over was selected by using the drop down box (shown in figure 6.10). As before, the choices were concrete, tarmac or paving. When the 'Train' button was pressed the program captured a frame of the video feed depicting the current surface, and used this to calculate the input values for the backpropagation calculation in order to set the weights. In this program the inputs were drawn from only one of the three RGB colour components. It can be seen from figures 6.5 – 6.7 that the shapes of all three RGB histograms vary between the three surfaces, and initially the Red component was chosen as being convenient for the input data for the neural network. Experiments on all three colour components were carried out, and trial and error later showed the Blue component to be the most successful. The neural net had no sub-layer, but 255 inputs corresponding to the saturation values of the red component of the 32-bit ARGB colour structure of each pixel. Figure 6.10 shows training in progress with 'concrete' selected as the target output. To complete the training of the network the Train button had to be pressed many times for concrete before changing the drop down box selection to tarmac; walking over tarmac and pressing the 'Train' button before moving onto paving. The user could see the output values changing each time the train button was pressed, and this gave an indication of how complete the program was in the training process. It was found that at least 20 frames of concrete had to be captured for the concrete output to rise to over 0.7. This was then repeated as described for tarmac and paving.

Once trained, if the 'Analyse' button was pressed, the program would then use the weights to calculate the output as explained for the first program in section 6.5.2. The four output values are seen in the top right of the figure, and are for concrete, paving, tarmac and other. Currently the value nearest to 1 is the top value indicating that the closest match of the current image to those in the training set is concrete.

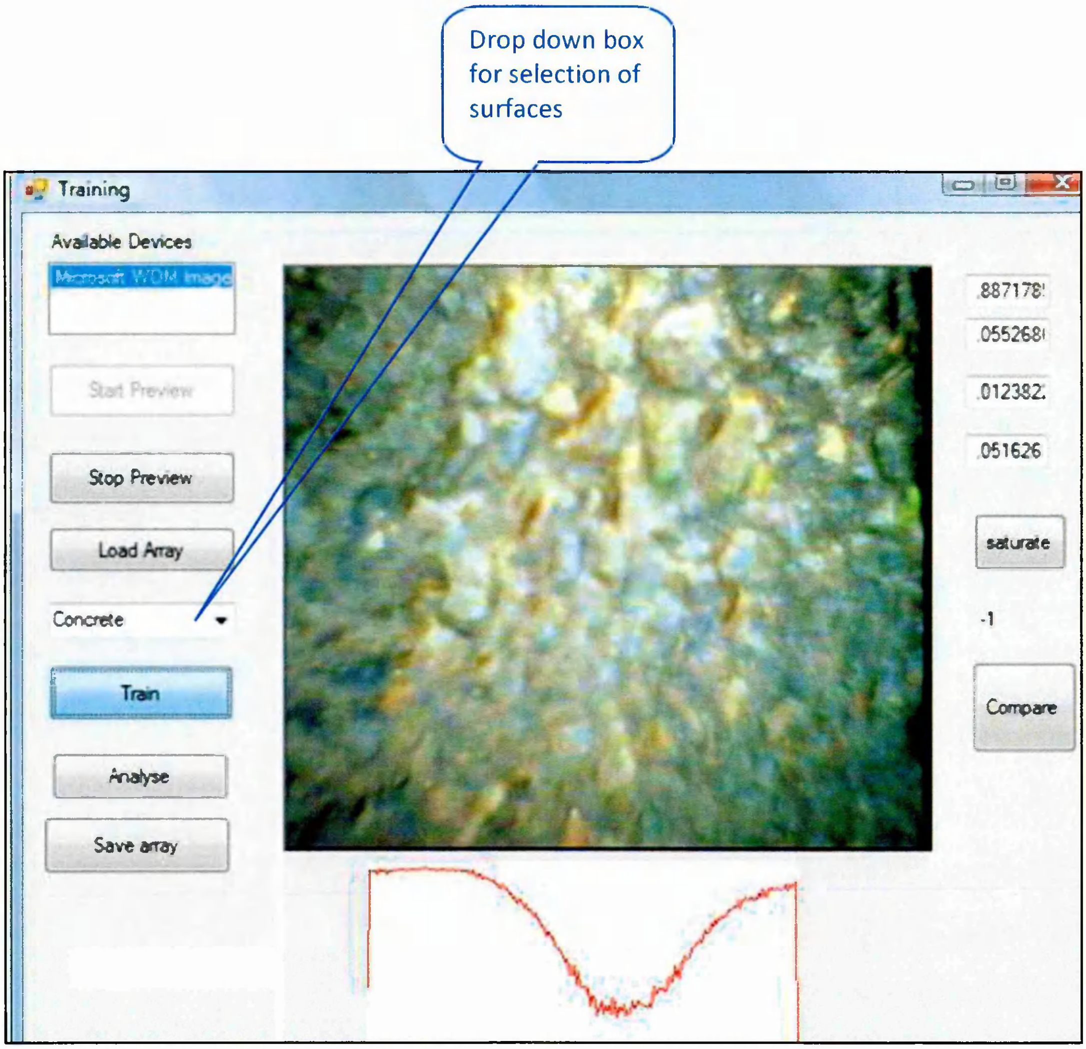


Figure 6.10 video feed neural network form

'Start Preview' allows the video feed to be displayed in the picturebox control. The drop down box (currently at 'Concrete') is used as the input during training while the text boxes (top right) work as the output boxes during analysis.

The difficulty with this program, as for the previous one, was that each frame in the training set had to be entered manually, in this case by walking over the target surface and pressing the Train button. This was very time-consuming. A development from the previous version was the inclusion of the 'Load array' and 'Save array' buttons – these allowed the variables corresponding to the weightings of the net to be saved and re-loaded when the program was next tested. Despite this innovation, the performance of the program was poor in that recognition rates of frames were uncertain at best. The network tended to regard each surface as being a close match to the one that it had just most recently been trained for, be that concrete, tarmac or paving.

On examination it was found that when training the network for one surface, say tarmac, the backpropagation calculations were having the effect of reducing the weightings learnt for a previously learnt surface, for example, concrete. It was thought that it would be better if, instead of the network learning by analysing 20 concrete images followed by 20 paving images and then 20 paving images, the different surface type images could be interlaced. This would mean the program would perform a backpropagation calculation on one concrete surface, the next on a tarmac surface, and then a paving surface before returning to concrete. In this manner it was thought that each calculation would have less damaging effect on connections already established. This method was explored in a later program.

Another problem was the nature of images which were found to be blurry. This was due to the position of the camera. At this time a wireless camera mounted at the end of a long-cane was used to provide the input as seen in figure 6.11. The camera was incorporated in a black box strapped to the bottom of the white cane, and to solve the problem of light level consistency, a high output LED powered by the same battery as the wireless camera was also placed in the box. The image from the camera was blurred in this position, and the camera was moved a little further up the cane to try to improve the quality of the video-feed.

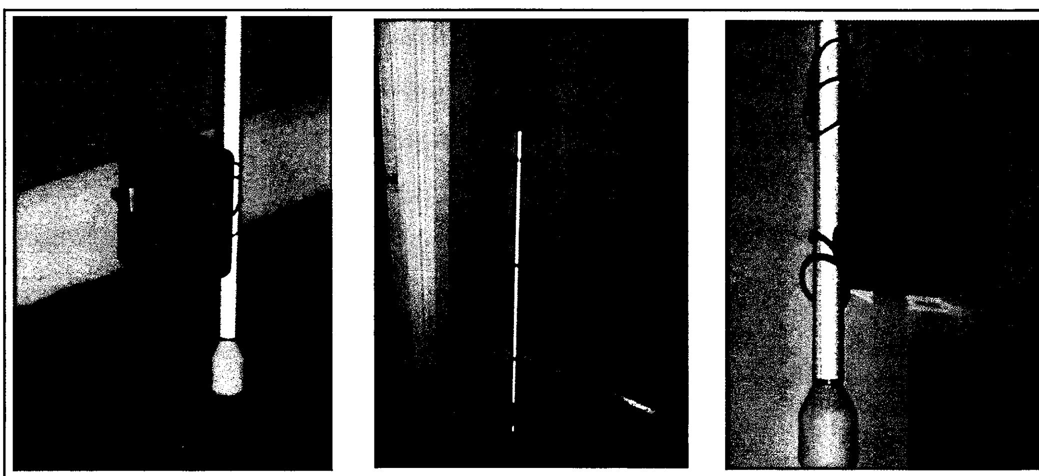


Figure 6.11 Images of the camera box crudely attached to the long white cane

6.5.4 The third and fourth neural network programs

The next step in programming was to allow video frames to be either captured from the video feed or loaded from (and saved to) file. Two extra buttons were added. The first 'Save pictures' allowed the user to capture the current frame and save it to a file. When the button was pressed the user was prompted to give the file a name and to save it to a file. By calling the image concrete1, concrete2 or concrete3 (or tarmac1, 2 or 3 or paving1, 2 or 3) it could then be used in an automatic training algorithm. The second button 'Automated training' initiated the automatic training routine as explained below.

The automatic training routine cycled through the stored images of the concrete, paving and tarmac surfaces, updating the weightings in the neural network as it proceeded. The program used three images of each surface, but interlaced the surface types so that it performed backpropagation calculations on first one concrete image, then one tarmac one, and then one paving one, before returning to concrete. It cycled through the 9 frames 200 times to set the weights. Using this technique, and after training, the program was able to correctly discriminate between surfaces in the training set of photographs. It was also able to discriminate between some 'live' surfaces being fed to it from the video feed.

In the fourth program an extra (hidden) layer of network nodes was added. Initially 20 hidden nodes were added so that the program consisted of 255 inputs (each of which could vary between 0 and the number of pixels in the image); 20 hidden nodes; and four output nodes. Unfortunately, the addition of the extra layer did not significantly improve the results. It was thought that this was partly due to the poor quality of the images and, in retrospect, it was also partly due to a program error which was not picked up on until the fifth network version (described below).

6.5.5 The Fifth Neural Network Version

It was at this time that the decision was made to purchase a better quality web-cam and the 'Quick Cam Pro for Notebooks' described in section 6.2.4 was purchased and used in subsequent programs.

Microsoft released Visual Basic 2008 Express, and the fifth neural network program was written by the author with a subroutine to help with the automated training of the network. The Interface form is shown in figure 6.12. When the 'Begin Photos' button is pressed the software allows 10 frames of each surface to be recorded and warns with an audible beep that the next surface should now be analysed. An external button was interfaced to the program and pressed each time a frame was to be captured. The external interface was built around a 'Motor Bee' package supplied by PC Control LTD. This package was chosen as it provides for up to four motor outputs as well as up to 6 digital inputs and is supplied with a DLL (dynamic link library) for incorporation into a Visual Basic program. The motor outputs were to provide the basis for a haptic interface in which vibration motors provide tactile feedback to a user with visual impairment. This interface was added later and more details on this, and the Motor Bee Interface board, are supplied in chapter 7.

Results from the neural network program were still not that clear and it was decided to test the system using the inputs from the eXclusive-OR (XOR) truth table. Heaton describes the testing of a neural network using an XOR logical truth table as "a sort of 'Hello World' application for neural networks" (Heaton, p 53). The idea is that, due to the nature of the data set, a two-input exclusive OR gate will require a hidden layer if a neural network is to be trained to recognise the function. Figure 6.12 shows the 'XOR test' button in place. (To activate the test normal inputs must be 'commented out' to prevent a subroutine called 'Sub ins()' from loading as this subroutine sets the inputs from the video frame currently in the picture box.) The button 'XOR test' ran the backpropagation algorithm 6000 times using the XOR truth table as the inputs and output. Once it had been trained in this way, the network was tested by entering 1 or 0 into the text boxes below the XOR test button. The output was displayed in the first text box (top right of interface). The expected results are shown in table 6.1.

IP(1)	IP(2)	Output text box
0	0	0
1	0	1
0	1	1
1	1	0

Table 6.1 XOR truth table test

Using this test, it was found that there were programming errors, including the omission of the unity input (as depicted in figure 6.8). The errors were fixed and the program found to be able to reproduce the XOR truth table. The program was then retested, and it was found that it was more reliably able to distinguish between the three surface types.

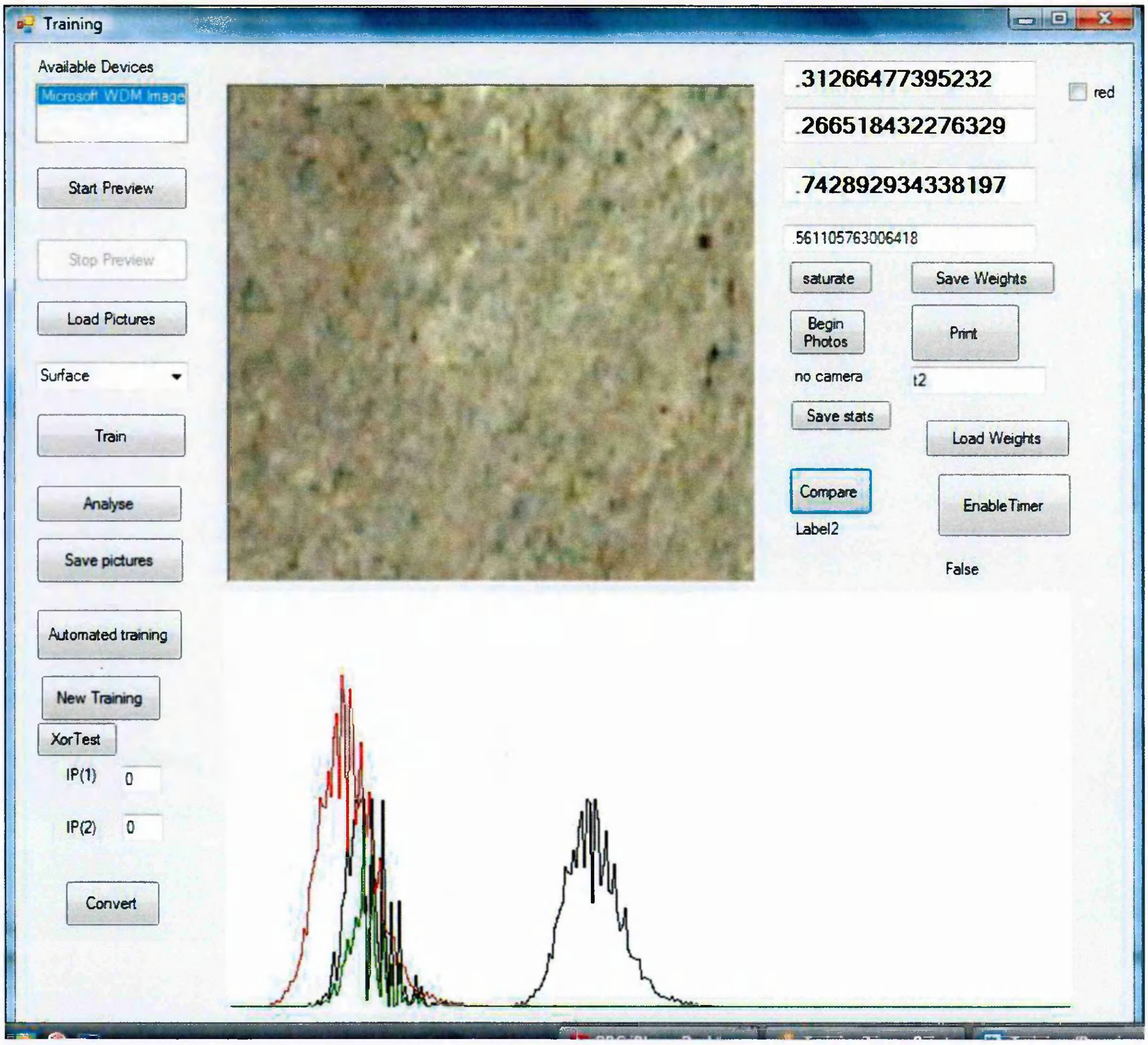


Figure 6.12 The Training Form

In this interface buttons are provided to allow photographs to be saved and loaded; a video feed to be employed; cues to be given for taking and saving a series of photographs; automated training to commence; as well as various tests including an "XOR test" of the neural network itself. The graphs show the red and green curves from the RGB components, and the Input values derived from them.

As before, this program included the ability to save, and then load, weights that had been calculated from images captured previously. One development in this program was the addition of a 'print' button which allowed for an image to be printed out, together with the graph of its RGB profile. A second development was the addition of a 'Save stats' (save statistics) button to allow information from the program's processing of the images to be saved to a comma separated variable (csv) file for later analysis. In this way it was possible to compare different light conditions; different camera angles; and eventually different input parameters.

6.5.6 The Final Neural Network

The program was refined further, and the interface form of the final version is shown in figure 6.13. Training in a new environment could be carried out by capturing 20 frames of three different surface types. The frames of each surface type were captured as before by pressing the external button connected to the Motor Bee Interface board. These were then used in an automated training subroutine which ran through the sixty images 200 times, adjusting the weights after each frame had been processed. The performance of the resulting neural network in distinguishing between the surface types could then be tested. Another button – 'Auto Test' was added which allowed the user to present a set of previously captured frames as the test set. This test set was then analysed with the current set of weights and the results saved in a csv file as before.

The analysis of the results was used to adjust the input parameters and re-run the training in an effort to improve upon the recognition ability. Trial and error was used to fine tune the network. The three colour components were each tested as the inputs – later hue and saturation were also used to see if they gave a better recognition rate. Rather than using all 255 inputs, a change was made so that the program looked at just a range of inputs (from IPmin to IPmax). The number of hidden layer nodes was varied. Appendix C contains the full listing of the neural network program in which the blue colour component was used for the inputs and the inputs range between a lower IPmin of 20 and a higher IPmax of 180; there are 20 hidden nodes (Nhid); and four output nodes as before.

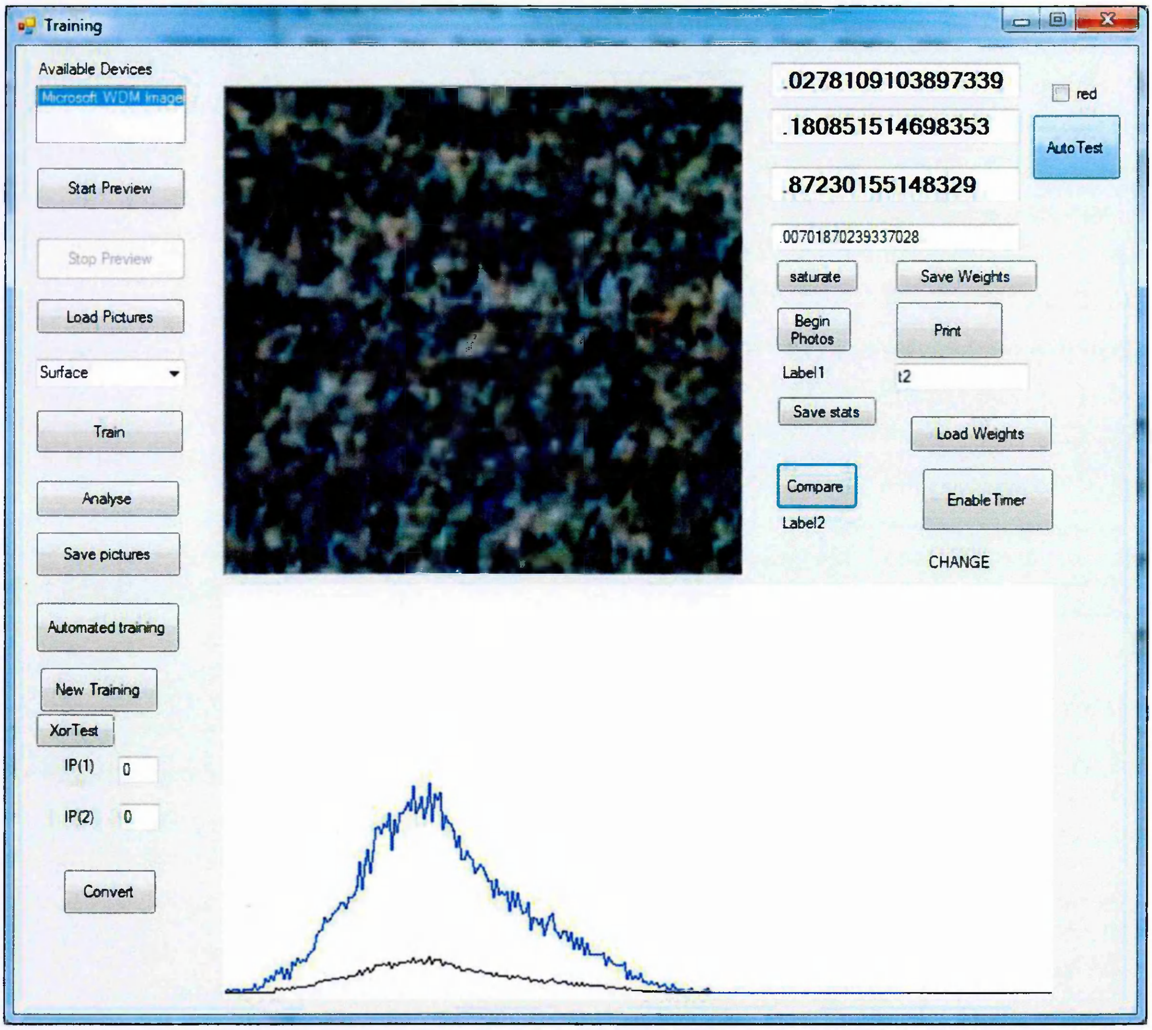


Figure 6.13 The Final Interface form

This final form shows the addition of the "Auto Test" button (top right) which produces a record of the analysis of photographs in a data test set.

6.5.7 Flowcharts depicting the operation of the network

Flow charts depicting the training and testing of the neural network are depicted in figures 6.14 and 6.15 respectively.

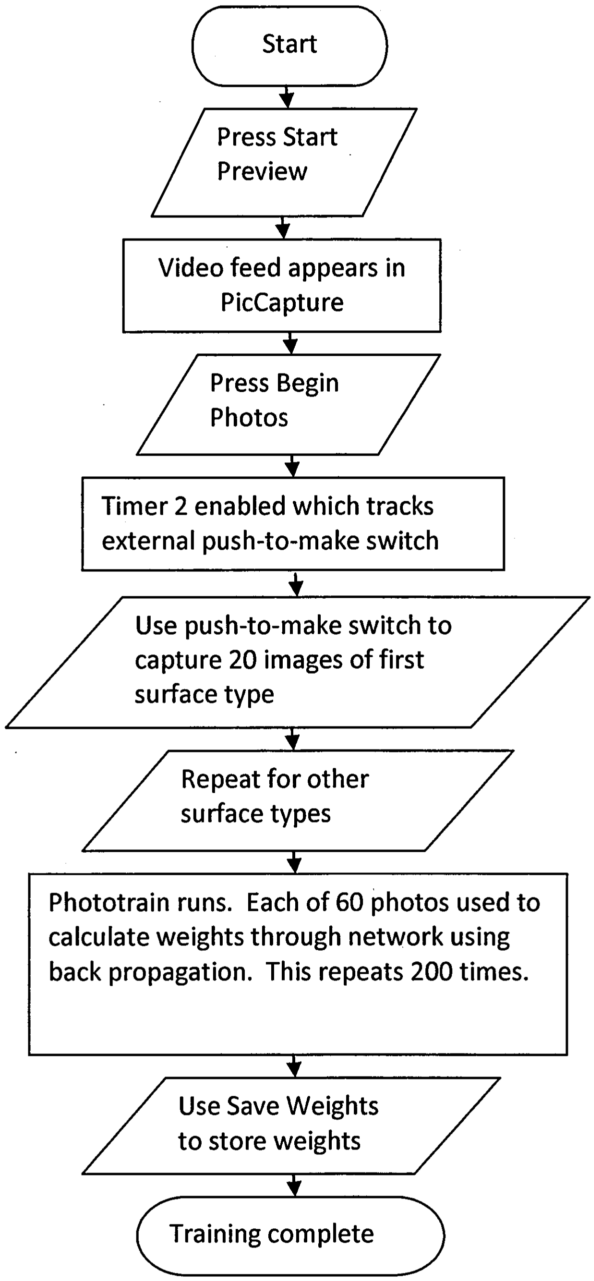


Figure 6.14 Flow Chart of Neural Network Training

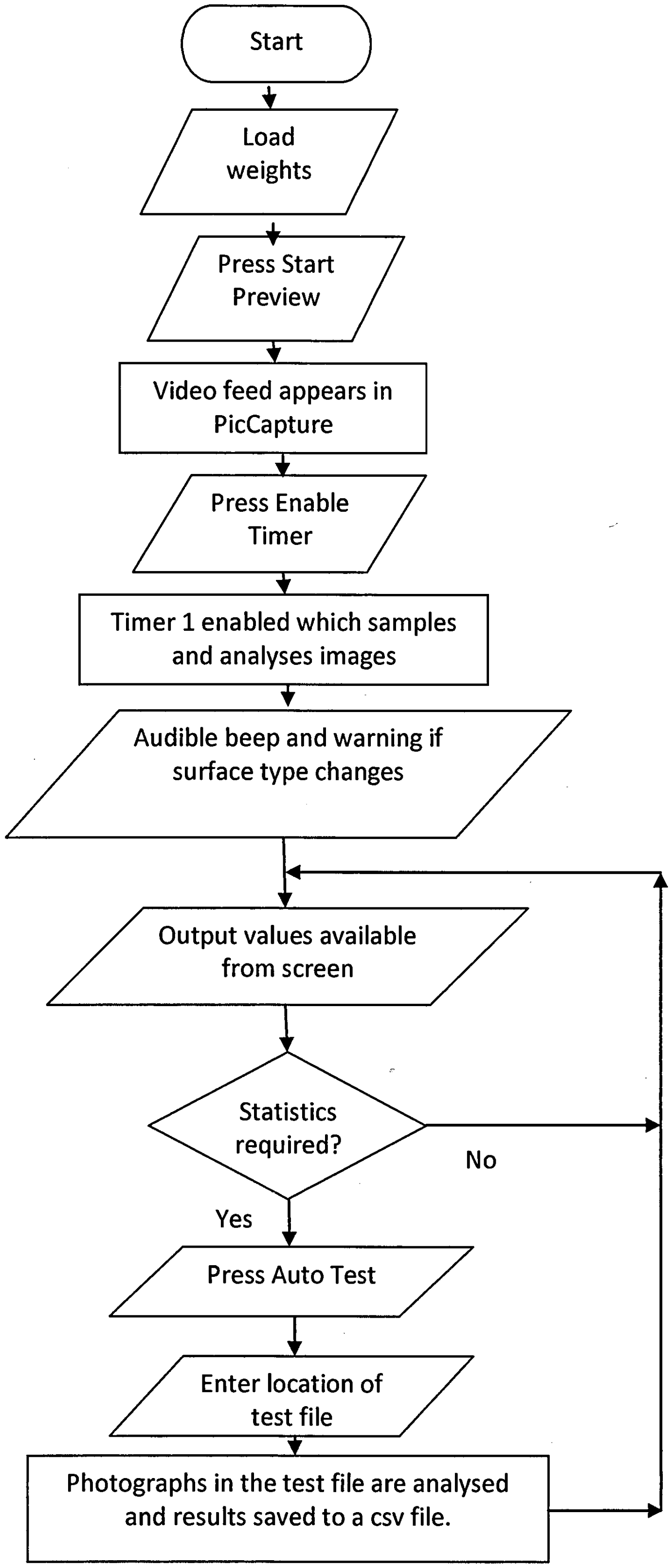


Figure 6.15 Flow Chart of Neural Network Operation

6.5.8 Results for the final Neural Network Program

Although the program was able to differentiate between the three surface types, the performance was poor even when the test data set consisted of sharp images. When the images were blurry recognition rates fell, and when different lighting conditions were encountered the program performance was worse again. For these reasons a second solution was considered which is referred to here as the area conversion program.

6.6 The Area Comparison Program

6.6.1 Introduction

As described above, the neural network software was found to be rather limited in its ability to differentiate between surface types, and was found to perform poorly when lighting conditions changed from those experienced in the test set. Reading accounts of how the problem of surface recognition had been tackled elsewhere led to an investigation of articles on autonomous vehicle direction finding.

The function of the area comparison program was to compare the immediate foreground of an image (the safe zone) with the area immediately behind it. To do this the program combined the outputs from several colour space and edge detection algorithms to form a composite image which was then analysed.

6.6.2 Description of the inspiration and potential problems

As described in the literature review in Chapter 3, autonomous vehicles share the same objective as people navigating an environment in that they have a requirement to find a safe path forward. The vehicles can use machine vision to scan the area ahead, and determine if it is a safe path or not. One method of finding such a path is to compare the part of the path that the vehicle is currently on with the area immediately ahead. The immediate foreground is taken as a 'safe region' and its characteristics analysed. The area that is just beyond the immediate foreground can be analysed and its features compared with the safe region to determine whether it is similar or different.

What becomes necessary is a system for analysing the 'safe region' and the region(s) just beyond in a way which will produce significant data, and which will do so rapidly while the 'user' (whether an autonomous vehicle or pedestrian) is moving forward. The system must take into account changes in lighting and shadows as they fall across the anticipated trajectory.

The part of the route just beyond the immediate foreground can be broken into regions so that the system can determine which part or parts of the anticipated route are similar to, and which are different from, the safe region. In this way the user can be instructed to move straight ahead or turn to the right or left or indeed be instructed that the path ahead comes to an end in its current form.

An autonomous vehicle program developed by Katramados (discussed in the literature review sections 3.5.1 to 3.5.3) was tested using a video feed from a high definition camera and found to cope with both reflections and shadows (Katramados *et al.*, 2009). The software was written in C++ so as to perform at maximum speed and it was initially unclear if the idea could be transferred to a webcam and a program written in Visual Basic.

An autonomous vehicle must have a link between the system analysing the path ahead and the decision making process which determines the path the vehicle is going to take. The information must be processed at a speed consistent with the speed of the vehicle if accidents are to be avoided. If the present path comes to an end then the decision must be taken whether to stop and await assistance, or whether to do an about turn and travel back on the path just taken.

In the case of a blind or partially sighted individual, he or she can be given the information about the surface type changes and then use this information to make the decision about whether to proceed or not. In the case of a user on a Shared Surface, or other difficult urban environment, a change in surface colour may indicate that the user is moving into a more dangerous zone and needs to take extra caution. Alternatively, it could be an indication that the zone is to be avoided.

6.6.3 Area Comparison Program Version 1

The first area comparison program was written by the author to test out the concept of comparing the immediate foreground with an area just ahead. The outputs from several colour space variables and edge detection algorithms across different areas were calculated to see which outputs and algorithms gave the best results in terms of providing a clear distinction between the area deemed 'safe' and an adjacent 'unsafe' area. In this context 'unsafe' implies a change in surface type, possibly a change in colour or texture, or the presence of an obstacle.

In the literature analysis chapter and section 3.5.2, it was seen that different attributes from three different colour spaces were useful in eliminating artefacts caused by shadow and standing water or wet prints. Area Comparison version 1 performed calculations on individual pixels to extract different colour space outputs and on groups of pixels to detect edges. The colour space outputs were saturation, hue and 'Chroma'. Saturation and hue were drawn from the HSL or hue-saturation-luminance colour space. Chroma is a term used to describe the combination of intensity invariant parameters from two different colour spaces: Pb (or Cb) and Pr (or Cr) from the YPbPr space (YPbPr standing for luma, blue-difference, red-difference) and A from the LAB space (LAB standing for Lightness, A, B, where A and B are colour-opponent dimensions). The exact ratios of Pb, Pr and A in the Chroma channel were determined by trial and error as explained below. Also explained are the mathematical algorithms used to calculate each colour space output from the RGB colour space.

Edges were detected in two different colour spaces. Firstly, in the RGB colour space the green and blue components were ignored and then the red monochromatic values in one pixel were subtracted from those in a neighbouring pixel. Secondly, the luma values from the YPbPr colour space were used in a similar way – subtraction of one pixel's values from that of a neighbouring pixel. This method of edge detection by subtraction of neighbouring pixels is basic but fast. Speed is important as the system must be able to keep up with the moving user.

An extra channel was obtained by compressing the image using a simple pixellation technique. The image was broken down into 5 x 5 blocks and the average R, G and B values taken over all 25 pixels. The average was then used as a pixel in a new image of reduced size.

Figure 6.16 shows the interface form of the first area comparison program. The same interface and software library (avicap) was used to obtain the video feed from the web cam as was used for the neural network programs described earlier.

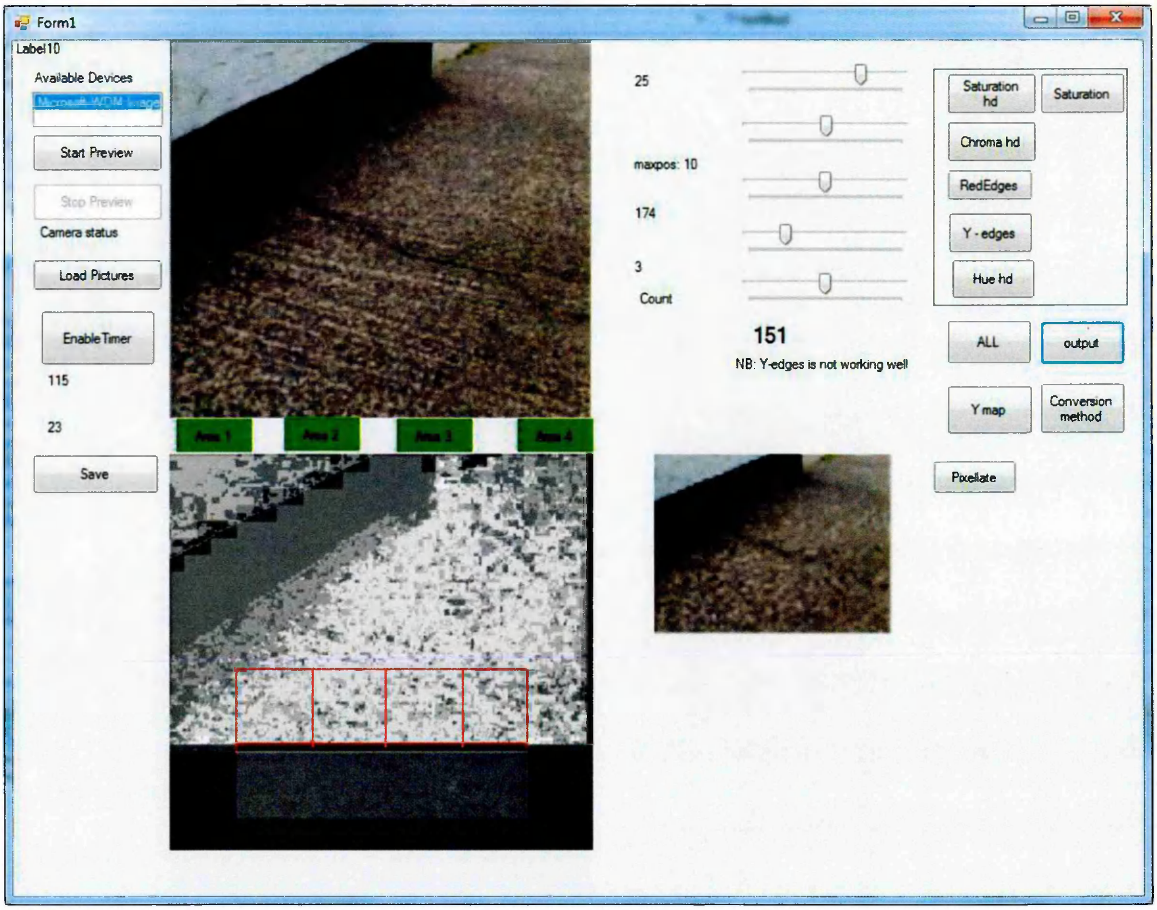


Figure 6.16 Area Comparison Form version 1

Sliders vary the weighting of each of the colour space variables. The lower picture box shows the composition of the superposed images from each of the channels. In this image the area within the red rectangle is mostly white, indicating it is of the same surface type as the safe zone.

The figure shows three separate images which have all been derived from one frame of the video feed. The top image shows the frame as captured by the web-cam. The smaller bottom right image is the pixellated image. The lower image shows the result of the current analysis. It is a composite image which has been made up by adding and normalising the results of the weighted colour outputs and edge detection algorithms. It can be seen that the whole image has been processed. In the bottom of the image there is a grey rectangle surrounded by a black border. This grey rectangle is in the zone close to the camera and is deemed to be safe. The area immediately above this safe zone is the area that the user is about to move onto, if he or she were walking forwards. This area is divided into four regions, depicted as four red rectangles. A series of four labels can be seen; one for each region, labelled 'Area 1', 'Area 2' 'Area 3' and 'Area 4'. These are currently green in colour.

A typical colour space output histogram is seen in figure 6.17. The horizontal axis is the saturation parameter on a scale of 0-255. The vertical axis is the frequency of pixels. In this figure the majority of the pixels in the zone have a parameter close to 180. This would be typical of a colour space histogram found in the safe zone, and in order to compare its properties with that of other zones, the program would calculate the modal parameter value (maxpos), which in this case is 180. When examining other pixels outside the safe zone, if a particular pixel has a colour space parameter within a threshold of 10 from the maxpos value, then it would be deemed as being safe according to that parameter. In the actual program similar calculations were run on all the chosen colour space outputs and then the results amalgamated.

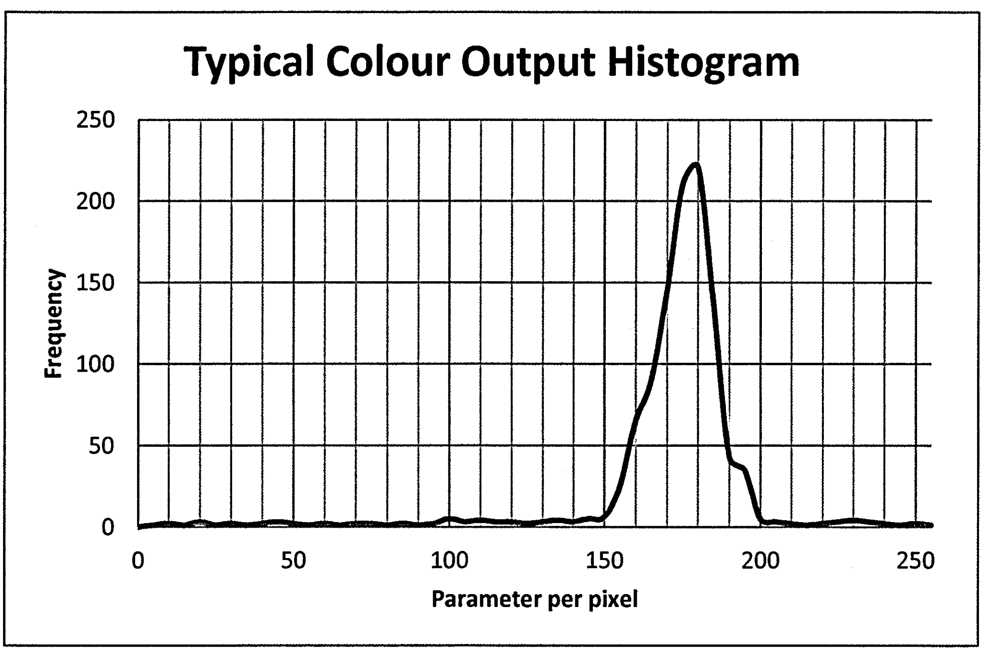


Figure 6.17 Typical colour space histogram

The percentage content of each channel could be adjusted manually by the user using the sliders depicted in figure 6.16. If there was a significant difference between the safe zone and any one of the four regions then a warning in the form of a visual and audible signal was given to the user. The background of each of the four labels ('Area 1', 'Area 2' and so on) changed from green to red when that area was different from the safe zone. At the same time each region had a distinct audible tone which sounded in the event of the surface being different. (In the second version of the program these signals are complemented with a set of four vibrating motors.) By way of illustration, figure 6.18 shows the same form as that of figure 6.16, but with a different image – in this case tarmac containing an area that has been patched. The figure shows only the saturation component of the image. In this case the two labels on the left of the image are red, indicating that these areas are unsafe.

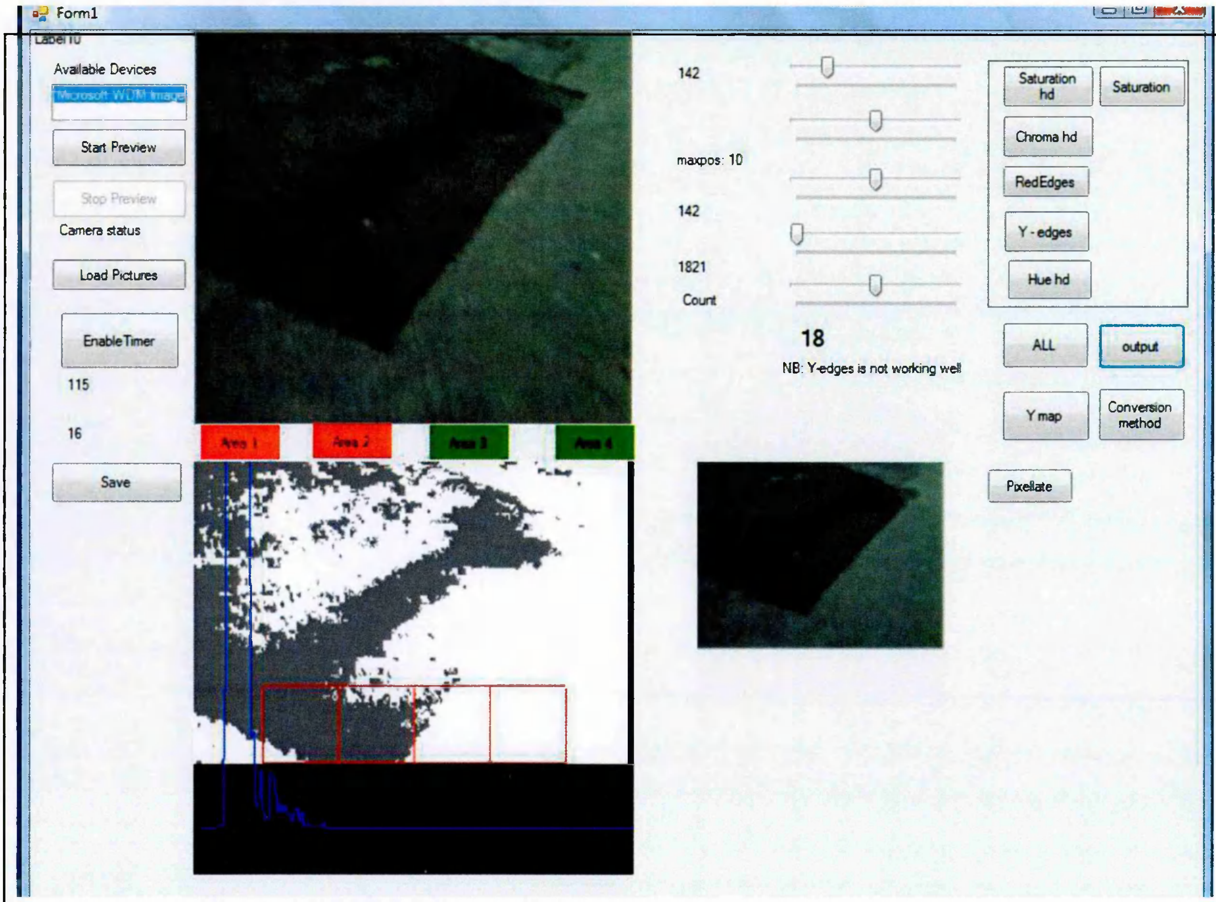


Figure 6.18 Area Comparison Form: the Saturation hd (higher definition) button has been selected before the output button was pressed. The lower picture box shows the saturation channel of the image. The two left areas both contain significant amounts of grey, and so these two areas are deemed unsafe and the labels are shown in red. There would be an accompanying audible tone.

The function of each of the buttons on the interface form is described below after a description of how the RGB colour information for each pixel was extracted by the program en masse using a method called BitmapDirect

6.6.4 En masse calculation of colour space outputs using BitmapDirect

The program relied on real-time extraction of colour information from the safe zone and comparison with the colour information in each of the four areas to be tested. Initially this extraction was carried out using the same method as had been used in the neural network programs discussed above. However, it was found that this method was too slow to do this effectively. Code segment 6.1 shows the old method and the loop that was used to extract the red component from pixels in the image in the neural network program.

```
With bmap
      For i = 0 To .Height - 2 Step 2
            For j = 0 To .Width - 2 Step 2
      'get pixels and examine red component - step 2 so every other pixel
      'examined
                  pixel1 = .GetPixel(j, i)
                  red = pixel1.R
                  RedCount(red) = RedCount(red) + 1
            Next
      Next
End With
      For j = 1 To 255
            IP(j) = (RedCount(j) - 150) / 150
            If IP(j) > 1 Then IP(j) = 0.999
            If IP(j) < -1 Then IP(j) = -0.999
      Next
```

Code Segment 6.1 Colour Extraction in Neural Network Program

In this loop the GetPixel method is called for every pixel that is examined (the program actually examines one in every four pixels to save time). Since the image bitmap must be repeatedly sampled this function call becomes slow on aggregate. A better method is to sample the complete image and map all the information from all the pixels to memory. The analysis can then be carried out on the memory map, which can then be updated en masse between the processing. In order to carry this out a separate class 'BitmapDirect' was written. The class was adapted from a Sobel edge detector program written by a programmer (Cowles, 2008) which in turn was based on a program found in a Microsoft development network library (MSDN, n.d. b). The class is reliant on the 'lockbits' method which 'locks' the bitmap into system memory so that it can be accessed dynamically. Once the bitmap has been mapped into memory the colour information can be extracted.

6.6.4.0 Buttons on the interface form

The following explains the function of each of the buttons controlling the extraction.

6.6.4.1 Pixellate

As described above, the image is broken down into 5 x 5 blocks and the average R, G and B values taken over all 25 pixels. The average is then used as a pixel in a new image of reduced size.

6.6.4.2 Saturation HD

HD stands for higher definition; in other words it is the original image that is to be examined and not the 'pixellated' version. The saturation of each pixel in the safe zone (shown as a dark rectangle in figures 6.16 and 6.18) is obtained. The saturation value is acquired from the BitmapDirect Class using the function GetSaturation which in turns calls the VB.Net color method getsaturation. The maximum saturation value is recorded – both the amount (stored in the variable 'max') and the saturation at which this amount occurs (stored in the variable 'maxpos'). This gives the point in the saturation histogram against which pixels in the areas to be examined can be compared. A graph of the saturation histogram for the safe zone is plotted in blue.

The remaining image is examined pixel by pixel and if the saturation level is within a set value of the maximum saturation (maxpos) then the pixel is coloured white, otherwise it is coloured grey. The SetPixel function from the BitmapDirect method is used.

6.6.4.3 Chroma HD

This button is similar to the Saturation button, but this time the Chroma level is examined and compared. As explained above, Chroma is used to describe the combination of intensity invariant parameters from two different colour spaces: Pb and Pr (from YPbPr) and A (from LAB). Pr and Pb for each pixel are obtained by a mathematical calculation involving the red, green and blue component values; the equations used are

Pr = 128 + 112.439 / 256 x Red - 94.154 / 256 x Green - 18.285 / 256 x Blue

Pb = 128 - 37.945 / 256 x Red - 74.494 / 256 x Green + 112.439 / 256 x Blue

The LAB colour space parameters are calculated from the XYZ colour space, and this is obtained by a translation of the RGB colour space. The transformations are handled by the function GetA which is printed below:

```
Public Function GetA(ByVal x As Integer, ByVal y As Integer) As Double
        'Code taken from http://www.easyrgb.com/index.php?X=MATH&H=02#text2
        ' to find XYZ colour space - but I only need X and Y
        Dim var_R As Double = (GetRed(x, y) / 255)      'R from 0 to 255
        Dim var_G As Double = (GetGreen(x, y) / 255)    'G from 0 to 255
        Dim var_B As Double = (GetBlue(x, y) / 255)     'B from 0 to 255
        Dim XX, YY, AA As Double

        If (var_R > 0.04045) Then
            var_R = ((var_R + 0.055) / 1.055) ^ 2.4
        Else : var_R = var_R / 12.92
        End If

        If (var_G > 0.04045) Then
            var_G = ((var_G + 0.055) / 1.055) ^ 2.4
        Else : var_G = var_G / 12.92
        End If
        If (var_B > 0.04045) Then
            var_B = ((var_B + 0.055) / 1.055) ^ 2.4
        Else : var_B = var_B / 12.92
        End If

        var_R = var_R * 100
        var_G = var_G * 100
        var_B = var_B * 100

        XX = var_R * 0.4124 + var_G * 0.3576 + var_B * 0.1805
        YY = var_R * 0.2126 + var_G * 0.7152 + var_B * 0.0722
        'Z = var_R * 0.0193 + var_G * 0.1192 + var_B * 0.9505
```

```
        'Hunter LAB equations from
http://www.easyrgb.com/index.php?X=MATH&H=05#text5
        If YY = 0 Then
            AA = 0
        Else
            AA = 17.5 * (((1.02 * XX) - YY) / Math.Sqrt(YY))
        End If

        '(H)L = 10 * sqrt( Y )
        '(H)a = 17.5 * ( ( ( 1.02 * X ) - Y ) / sqrt( Y ) )
        '(H)b = 7 * ( ( Y - ( 0.847 * Z ) ) / sqrt( Y ) )

        Return AA

    End Function
```

Chroma is then calculated by combining the values of Pb, Pr and A. The weightings of each were determined by trial and error, and it was found that for the most effective elimination of shadow, the A value had to be multiplied by 4 to obtain a value comparable to the Pb and Pr values.

The subroutine 'Private Sub `BtnChromaHD_Click`' calculates the Chroma value, and again the max and maxpos values for the safe zone are found and used to plot a Chroma map of the rest of the image. Finally, a Chroma histogram is plotted.

6.6.4.4 Red Edges

By shifting the image by one pixel, and subtracting the R values of each (from the RGB colour space), edges can be detected. This subroutine calculates the density of red edges in the safe zone and compares that with 10x10 pixel blocks in the rest of the image. If the densities are a close match the complete pixel block is coloured white, otherwise it is coloured black. This gives a fairly 'blocky' result, as can be seen in the diagonal edge of the processed image of the wall in figure 6.16.

6.6.4.5 Y Edge

The same process is carried out as for the 'Red Edges' method, except that the starting image is the chroma map. This method thus compares the density of edges in the chroma map of the safe zone with 10x10 pixels in the rest of the image. This process did not appear to work well and was later abandoned.

6.6.4.6 Hue hd

This button is similar to the Saturation hd and Chroma hd buttons already mentioned. In a manner similar to the saturation function, this function calls on the VB.Net color function gethue. In this case the hue of each pixel in the safe zone is measured and the max and maxpos variables compared with the pixels in the rest of the image. A green graph of the hue histogram is drawn. In accordance with Katramados, it was noted that this graph sometimes contains two separate peaks indicating that two colours may be prominent in the safe zone (Katramados *et al.* 2009).

6.6.4.7 Saturation

Performs the same task as 'Saturation HD', but works on the pixellated image. This was created to ascertain whether it would be a useful channel to include in the final program. It was discarded as not providing any significant extra information.

6.6.4.8 ALL

This button combines the images of all the channels (except for the pixellated saturation one) according to weightings set by the sliders. Six different output maps are generated, one for each channel, and one to hold the final bitmap. This is not a fast routine, and also seems to contain a memory leak as the program will eventually stop working if allowed to run in timer mode. Each pixel generated by this button will be lighter, if it is similar to the safe zone in terms of the attributes chosen, and darker if it is different from the safe zone.

6.6.4.9 Output

This button analyses the area above the safe-zone and divides it into four separate regions. Each region measures 55 x 20 (a total of 1100 pixels). The region is deemed clear if it has largely the same parameters as the safe-zone, otherwise it is marked as being not clear and a warning is issued. To determine whether it is the same or not, the subroutine counts up the number of lighter pixels in each region (lighter meaning that the R value is greater than a threshold of 100). If there are fewer than 1000 lighter pixels then the warning is given for that area.

6.6.4.10 Y Map

This was used to experiment with the different colour space components and produces an image to show a representation of which ever data set has been programmed in - the hue data, for example.

6.6.4.11 Other functions

In addition to the buttons outlined above, the area comparison program allowed for continuous testing of the video feed through the 'Enable timer' button. With the timer enabled the program continuously calculates the maximum values of each component in the safe area and determines whether the four zones have similar parameters. As an alternative to testing the video feed, previously saved frames can be tested using the 'Load Picture' button. The 'save' button allows for the capture of the current frame from the video feed.

6.6.5 Results of testing

The program demonstrated that it was possible to compare the colour spaces of the safe zone and the adjacent areas in real time. The sliders were used to ascertain the best combination of input channels to use in the next implementation of the software – Area Comparison 2.

6.6.6 Area Comparison Program Version 2

The second area comparison program is a much faster adaptation of the first version and is a development of Version 1. Version 2 also uses the BitmapDirect class but combines the different colour space outputs by using only one bitmap. By contrast, in the first version a separate bitmap was used for each output, and this slowed the program down. In addition, Area Comparison 2 only analyses the safe-zone and the four test regions; the rest of the image is ignored.

The same video buttons, 'load picture' button and labels are present on the interface form as were on the version 1 form. The hue, chroma and saturation channels are analysed and compared as before. However, in this version the program looks for two hue peaks which allows for two prominent colours in the safe zone to both be used. It was found by using the sliders in the first version that the edge detection channels did not significantly contribute to the performance of the program and these were dropped in the second version.

In version 1 of the program each colour space output was calculated for the whole image and then the weightings applied (as determined from the sliders). This meant that the image was scanned 6 times. In this version the safe zone is scanned once, and the peak values and positions of all the colour channels recorded in that one scan. Each test region is also scanned once and the hue, chroma and saturation values compared with the safe-zone maxima readings. A series of IF... END IF statements then add a weighted value to the colour of the output pixel if the test pixel lies close to the maxima values. The greater the number of matches the lighter the output pixel becomes.

Finally, the resultant test areas are scanned and the number of lighter pixels counted. 1000 is again used as the threshold for a clear test area so that if less than 1000 lighter pixels are found, then the area is deemed unsafe. The different weightings and the threshold can be varied to give the best results. The warnings are interfaced to switch on a vibration motor using the Motor Bee interface board as described in chapter 7.

The second area comparison program was found to perform well on still images and with a relatively slow to medium walking pace. It was somewhat resilient to light intensity changes and shadows were ignored to a degree, though in bright sunlight performance was degraded. However, it was fast enough to keep up with the video feed; it produced real time information on the area immediately in front of the user; and, unlike the neural network program, it did not have to be trained. Appendix D contains the final version of the Area Comparison program (with a modification explained in section 7.7). More detailed testing of the program is described in chapter 8.

6.7 Summary

This chapter has examined the vision subsystem discussing the video acquisition interface and the signal processing software. The development of the neural network program and the area comparison programs were explained and some indication of their effectiveness in solving the problem of processing the camera image was developed. The human computer interfaces of both programs (and all their various prototypes) was touched on and this is developed in more detail in the next chapter while chapter 8 goes into more detail on the testing of the software.

Chapter 7 : Human Computer Interface

7.1 Introduction

The human computer interface unit of the system is required to give an output to a blind or partially sighted person. Since a visual output is of limited or no value, the decision must be made as to whether a tactile, audible or combined tactile and audible output is preferable. Discussions with people with visual impairment showed that they tend to use the audible channel to listen to traffic noise, and would prefer not to have this interfered with by additional information. However, as explained in the literature review, there is conflicting evidence on this and it was decided to make both forms of output optional.

The purpose of the prototype is to let the user know if the surface they are walking on is about to change. The user could be notified by anything from the sound of a simple 'beep' warning that there is a change just ahead, to a running commentary on the surface type in front of the user. A tactile warning could be provided by one or more vibrating motors or shafts placed at a suitable point or points on the skin. The vibrators could be incorporated into a glove worn on one hand or into a belt worn around the waist (figure 7.1). A flow diagram of the human computer interface is shown in figure 7.2.

Figure 7.1 Possible tactile interfaces – glove or belt

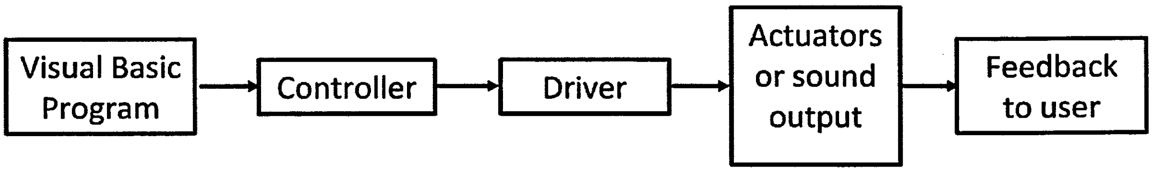


Figure 7.2 Flow diagram of the human computer interface components

7.2 The Neural Network Program HCI

The human computer interface for the neural network program consists of an audible output relayed to the user through the computer's loud-speakers. A simple beep indicates that the artificial neural net has detected a change in surface type. Once a surface has been reached by the user the output of the neural array is stored as a value between 0 and 1 in the array 'y(k)', where the variable 'k' represents the current output channel; the program has four output channels used to distinguish between surfaces. Three channels were used in tests to distinguish between concrete, paving and tarmac surfaces; the fourth was available for future expansion but never used.

Photographs of the different surfaces were used to train the neural network by backwards propagation and the setting of weights. Analysis of new surfaces by forward propagation yielded output values from which the program determined the nearest 'fit'. Figure 7.3 shows the program 'form' when one of the training photographs is analysed. In this case a concrete surface has yielded output values of 0.952, 0.024, 0.065 and 0.012 as can be seen in the top right of the screen-shot. The first three of these values represent the match to concrete, paving and tarmac respectively; the closer the value to unity the closer the match with the training set. The fourth value has not been assigned to a surface type and is near zero as expected. Since the photograph was in the training set, it is not surprising that it matches very closely. A surface type change will be reflected in a change in these outputs. For example, with a change to tarmac the figure for concrete drops to nearly zero and that for tarmac increases to nearly one.

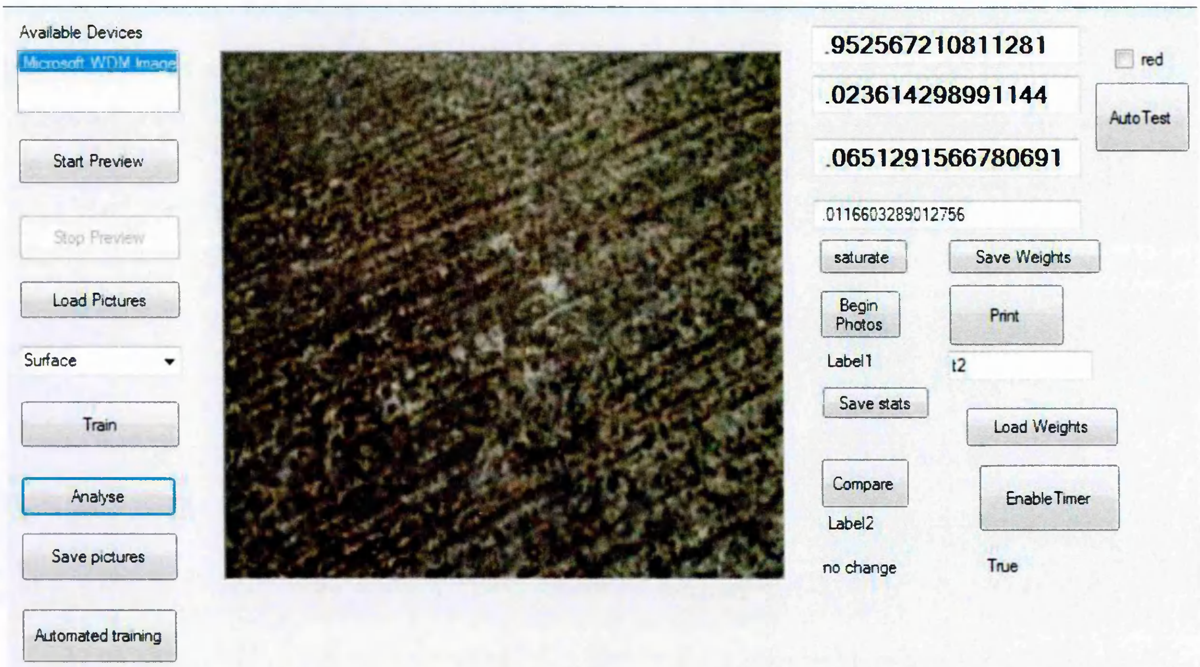


Figure 7.3 Main form from the neural network program showing the analysis of a concrete surface photograph.

The section of code that decides if a sufficient change has occurred is shown below (code segment 7.1):

```
Dim changeflag As Boolean = False
    For k = 0 To 3
       returnValue = Math.Abs(y(k) - track(k))
       If returnValue > 0.5 Then changeflag = True
       track(k) = y(k)
       If recordStats Then statsOutput(k, statsCount) = y(k)
    Next
    If changeflag Then
       Label6.Text = "CHANGE"
       Label3.Text = "CHANGE"
       System.Console.Beep(1000, 5)
    Else
       Label6.Text = "True"
       Label3.Text = "no change"
    End If
```

Code segment 7.1 Detecting a change in the surface type parameters

In this code the variable array 'track(k)' is used to store old values of the outputs and the Boolean variable 'changeflag' is set if any of the four output channels has changed by more than a threshold, set at 0.5. The 'Beep' command is used to give an audio output to the user. The command 'beep' has two parameters – tone (or pitch) and duration. Microsoft state, "The pitch and duration of the beep depend on [the] hardware and system software" (MSDNa, n.d. a). This is acceptable for the prototype, as is the fact that the sound is played through the laptop's built-in speakers, although the sound can be played through headphones if required. A short experiment revealed that the command 'Beep (1000, 500)' yielded a tone of 31 seconds so the command used here, 'Beep(1000, 5)', produces a high pitched tone of duration 0.3 seconds – sufficient for the user to sense a change in surface.

Discussions with members of Guide Dogs led to the conclusion that a short beep is acceptable as an indication of change of surface, but that this could be accompanied by haptic feedback in the form of a vibration. In principle, a different vibrating motor could be used for each of the four output channels, or alternatively just one motor could be used to indicate any change in surface. It would also be possible to incorporate a text-to-speech output informing the user which surface was currently on view. However, in the discussions it was felt that this would be too much information for the user, and in any case there would not usually be time for the walking user to gather this information before he or she had moved forward two or three steps.

The vibrating motor circuit was not incorporated in this program. However, the system explained in section 7.4 used for the final area-comparison prototype program could have been incorporated, and it would be easy to include an extra line in the code segment above to turn on and off a vibrating motor.

7.3 *Final Prototype program HCI*

For the Area-Comparison program it was decided to use both sound and vibrating motors in the prototype with the option of turning off either output channel. The vision processing software detects a change in the surface just ahead in four areas – far right, immediate right, immediate left and far left. These four areas are linked to a tone and a vibrating motor. The pitch of the tones varies across the areas, and the position of the motor on the body reflects the spatial position of the area. Hence, if a change in surface type is detected towards the far right of the path, then a higher tone is used, and a vibrating motor pad on the right side of the person is activated. Figure 7.4 shows a change in surface type on the right hand side of the picture.

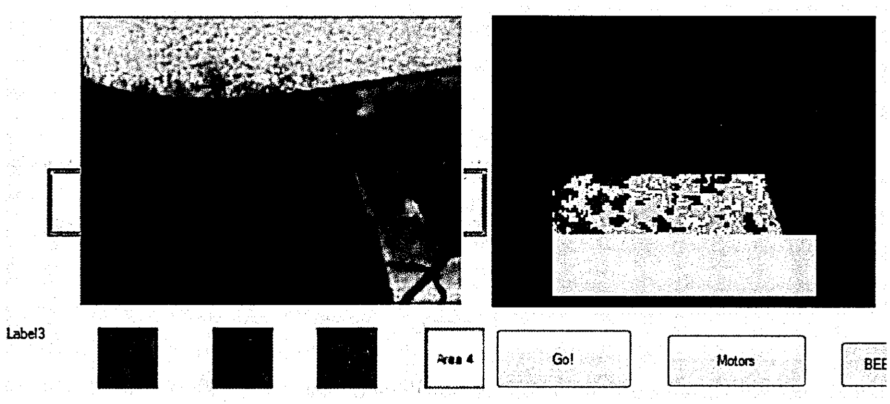


Figure 7.4 photo with change of surface clearly on RHS (left hand image)

Area 4 is associated with a higher pitch while area 1 is associated with a low pitch and areas 2 and 3 have intermediate pitches as illustrated in the table below (table 7.1).

Area	Command	Description of tone
1. Far left	Beep(200,10)	0.6 second lowest tone
2. Near Left	Beep(400,10)	0.6 second lower tone
3. Near Right	Beep(600,10)	0.6 second higher tone
4. Far Right	Beep(800,10)	0.6 second highest pitch

Table 7.1 Pitches and durations of tones across the four areas

As explained in the vision subsystem chapter, the software analyses the chrominance, saturation and hue levels in each area. If the maximum of at least two of these does not match the position of the maximum in the 'safe zone', immediately in front of the person, then that area is marked as potentially unsafe. The software then enables the appropriate tone or tones to be sounded and the corresponding vibrating motor or motors to be turned on as shown in the section of code below (code section 7.2).

```
'area 1
  count = 0
  For x = 50 To 105
     For y = 130 To 179
        If col(x, y) > 50 Then count = count + 1
     Next
  Next
  If count < 1000 Then
     If noise Then Beep(200, 10)
     LbArea1.BackColor = Color.Red
     o2 = 1 And BeeON  'motor 2 = speed 2; note area 1 is motor 2 because
 of belt
  Else
     LbArea1.BackColor = Color.Green
     o2 = 0
  End If
```
Code segment 7.2 Examination of left hand side area

This section of code counts how many pixels in the far left area (area 1) have an assigned colour value ('col(x,y)') greater than fifty. Each pixel has already been assessed in terms of to what extent it differs from the safe area and this colour value is a measure of that difference. For example, if the hue of a pixel matches the average hue in the safe zone then the pixel at that point is given a value according to a rating system. A similar calculation is carried out for saturation and chrominance. If a sufficient number of pixels in the zone show values outside the average levels, so that the threshold is not reached (currently set at 1000), then the zone is deemed unsafe and the 'If count <1000' condition is met. The variables 'noise' and 'BeeOn' are Boolean variables set by the user, and used to mask the output statements so as to turn on or off audio and motor outputs. A red block of colour is used to give a visual indication of a surface change in that area. This is principally of value to the programmer as a quick test that the program is functioning as expected. Figure 7.5 below shows the output of an earlier version of the final program.

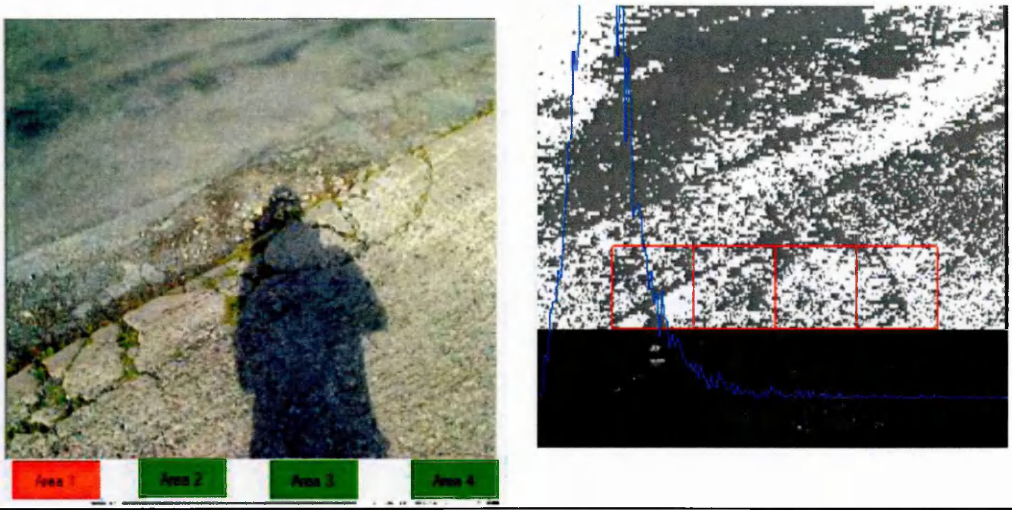


Figure 7.5 Shadow elimination

The left hand image shows a photograph of path with a shadow. The right hand image shows the processed image where the shadow has been largely eliminated. The "safe zone" is the lower rectangle surrounded by black pixels. The four squares outlined in red are the areas being analysed. The blue graph superimposed is the saturation count showing a maximum. If enough of the pixels in each zone have saturation values remote from the position of the maximum value in the safe area then that zone is deemed as unsafe. In the image it can be seen that the far left zone is unsafe – this is due to the hue variation rather than the illustrated saturation.

7.4 The Haptic Output

7.4.1 Introduction

The haptic output subsystem consists of a controller, drivers and actuators attached to a suitable item of clothing that the user can wear. It must be portable, including any extra power supply that needs to be carried, and it must be capable of being interfaced with the laptop being used for the prototype device. Finally, it must be capable of alerting the user to changes in the surface through the sense of touch alone. Figure 7.6 shows the components of the haptic output.

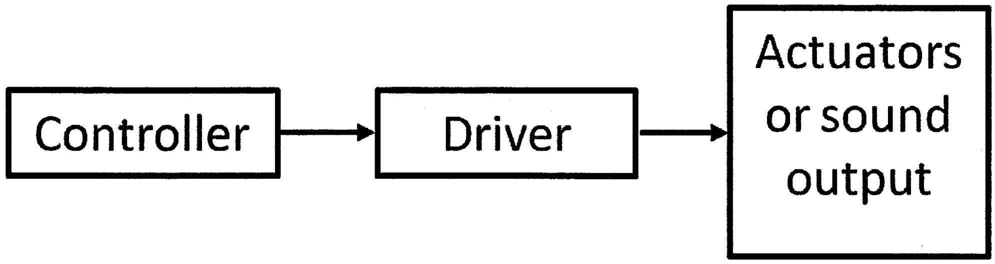


Figure 7.6 Haptic output components

7.4.2 The Controller and Driver

Several interface boards are available which can drive actuator devices and be controlled using software on a laptop. The 'Motor Bee' was chosen as it does not require an external power supply and it is supplied with a dynamic link library enabling commands to be embedded into the Visual Basic program.

The 'Motor Bee' is an interface board supplied by 'PC-Control' which can control up to four d.c. motors and a servo. There are an additional four digital outputs and six digital inputs available. The board communicates with a laptop using the USB port, but allows for an external power-supply of up to 24 volts for the motors. The speed of the four d.c. motors can be controlled in one direction, or bi-directional control for two motors can be realised. Multiplexing might allow the use of more motors, but four is sufficient for the current project. To achieve motor control there are two integrated circuits on the Motor Bee board – the microcontroller and an H-bridge driver, as can be seen in figure 7.7

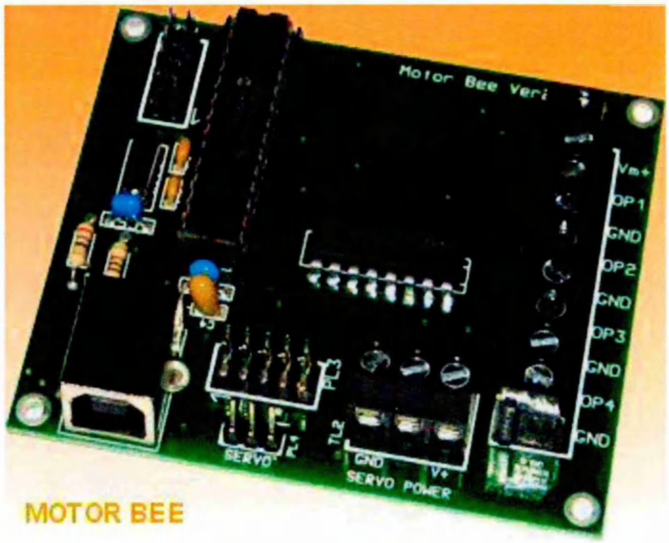


Figure 7.7 The Motor Bee board showing the two integrated circuits

The microcontroller on the Motor Bee board is a Microchip PIC16C765-I/SP, which is a reduced instruction set computer (RISC) device. It follows the Harvard architecture style, that is the program information and the data are accessed by the CPU using separate data buses. The program bus is 14 bits wide allowing all instructions to be completed in one clock cycle (with the exception of a program branch). The data bus is 8 bits wide which is sufficient for the project. The microcontroller contains two dedicated pulse width modulation (PWM) modules with parameters set up to allow a resolution of 10 bits (Microchip, 2000). It is a one-off programmable device meaning that the routines used by the Motor Bee have already been translated into machine code and are stored in the microcontroller memory. These routines cannot be altered, but are accessed through the commands in the dynamic link library.

The second integrated circuit on the Motor Bee board, the L293D, is an H-bridge driver which is described as "a high voltage, high current four-channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as ... DC and stepping motors)" (STMicroelectronics, 1996). In other words up to four direct current motors can be driven in one direction. The driver will support a maximum of 36 volts and a current of up to 60mA, which is sufficient for the Precision Motor vibration motors used in the prototype, provided they are not run continuously.

7.4.3 The Actuators

The haptic output actuators can consist of vibrating motors or shafts which are placed on the skin surface, or by direct electro-epidermal simulation. The latter involves using small amounts of electric current to cause a tingling sensation on the skin and, as mentioned in the Literature Review, this has been explored by Kaczmarek and others (Kaczmarek, 2004). The most promising tactile surface for this method has been the tongue due to the high density of nerve endings found there, and work is still ongoing to investigate this (Williams *et al.*, 2011). For this project, direct electro-epidermal simulation was not considered as the amount of feedback necessary is not large, although in theory, a tongue display could be employed.

Vibrating motors have the advantages of being cheap and readily available as they are widely used in mobile phones and video-game products. Shaftless d.c. motors purchased from Precision Motor Drives were used to provide the haptic feedback for the prototype. These are lightweight (1.2 grams), small (10mm diameter with a height of 3.4mm) and have a starting voltage of 2.3 volts (Precision Microdrives, 2011). These characteristics, coupled with a current draw of 65mA when turned on, make them suitable for use in the project.

As mentioned in the Literature Review, two projects in particular led to the decision to use an elasticated belt to hold the vibration motors. Work by Carl *et al.* on the device used in the 'feelSpace' project involved the use of an elasticated lumber support which held vibration motors close to the skin (Carl *et al.*, 2005 page 82-84). The position of magnetic north was picked up by apparatus which then caused one of the eight vibration motors in the belt worn by the subject to indicate the direction of magnetic north. The vibration motors were distributed about the waist in a horizontal plane so that an upright person will be orientated such that magnetic north will be in a direction lying in this plane. In experiments conducted by the group the subject wore the belt while active during the day and took it off only at night. This implies that the subject was able to tolerate wearing the belt for a protracted period and, given the success of the project, that he or she was able to adequately perceive the vibrations and distinguish one vibrating motor from another.

The second influential project was one involving helping learners of the violin to be aware of the position of the bow when drawing it across the violin strings. This was part of the 'e-sense' project (Linden *et al.*, 2009). The rationale behind this project is that learners need to develop an appropriate bowing action and this takes training. Vibration motors were used to signal to the learner that he or she was approaching the advisable limits of the bowing action. Correspondence with one of the researchers, Rose Johnson, indicated that the vibration motors needed to be pulled against the skin by an elasticated material if the user was to perceive them adequately. This is in keeping with the feelSpace project where an elasticated orthopaedic support was used, the *Thuasne Cemen 2900 orthopedic belt for women* (Carl *et al.*, 2005 page 83). Coincidently, the 'Music Jacket' made for the e-sense project used the same 10mm shaftless dc motors purchased from the same supplier (Precision Motor Drives) as were purchased and employed in the current research prototype.

The right hand photograph of Figure 7.1 depicts the four vibrating motors attached to a standard (non-elastic) belt. Also shown are the webcam, battery and Motor Bee board. Figure 7.8 shows the elasticated lumber support belt which was purchased for this project with the vibrating motors attached by Velcro strips.

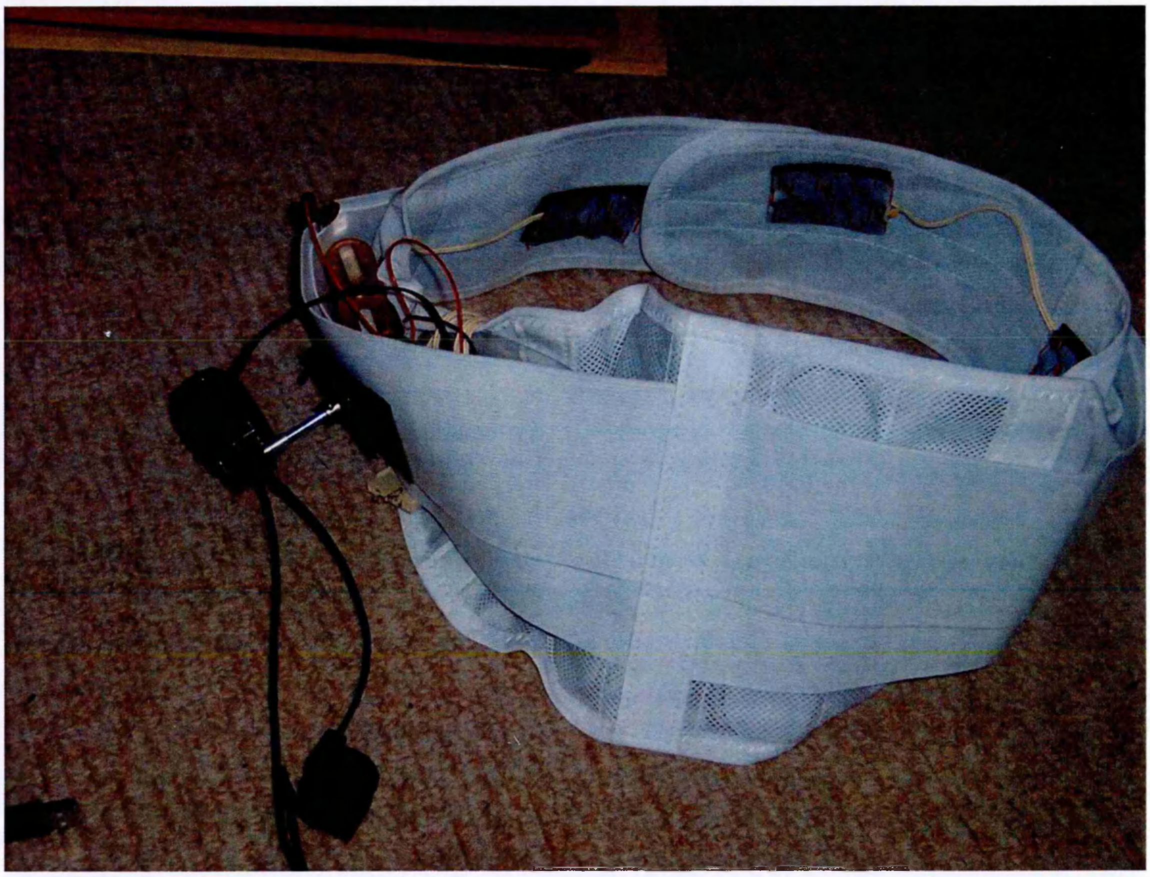


Figure 7.8 Elasticated belt (lumber support) with vibration motors attached by Velcro strips.

7.4.4 Communicating with the Controller

As described above, the 'Motor Bee' board comes supplied with a dynamic link library (DLL) which means it is possible to incorporate commands to communicate with the board from within a Microsoft Visual Basic program. The three functions used are declared in the declarations of the main form and are:

1 Declare Function InitMotoBee Lib "mtb.dll" () As Boolean 'mtb.dll is copied in windows/system32 as a library file

2 Declare Function Digital_IO Lib "mtb.dll" (ByRef inputs As Integer, ByVal outputs As Integer) As Boolean

3 Declare Function SetMotors Lib "mtb.dll" (ByVal on1 As Integer, ByVal speed1 As Integer, ByVal on2 As Integer, ByVal speed2 As Integer, ByVal on3 As Integer, ByVal speed3 As Integer, ByVal on4 As Integer, ByVal speed4 As Integer, ByVal servo As Integer) As Boolean

The board is then initialised by the command:

```
InitMotoBee()
```

which is included in the subroutine Form1_Load so that the board is ready to go as soon as the program loads the main form. The Motor Bee variables are declared in the following code:

```
'motorbee settings:
    Dim s1, s2, s3, s4 As Integer
    Dim o1, o2, o3, o4 As Boolean
    Dim sv As Integer
    Dim inputs As Integer
    Dim outputs As Integer
```

where s1 – s4 are the integer variables representing the speed of the motors on the scale 0-255; the Boolean variables o1 –o4 represent the four outputs which are either on or off; sv is available to be used for an external servo and 'inputs' and 'outputs' are used for the digital input and output ports.

In the area-conversion program, each of the vibration motors is connected to a separate output and controlled using o1 – o4. In the sample code above, the servo motor attached to the front of the belt on the left was controlled using the variable o2. The speeds were all set at 215, which is almost at the maximum, as this gave a vibration that was easy to perceive.

Once all four areas have been compared to the safe zone, and the motors variables set as either true or false, a command was used to output the data to the driver and so switch the motors on or off. The command used was:

SetMotors(o1, s1, o2, s2, o3, s3, o4, s4, sv).

The speed of the motors is determined using pulse width modulation (PWM) where the mark-space ratio of a square-wave signal is varied to provide more or less average power to the motor. The onboard microcontroller handles the conversion of the s1-s4 value to the appropriate mark-space duty cycle.

7.5 External Input

The incorporation of the Motor Bee interface board allowed for the use of digital inputs and one of these was used to connect to an external switch. This switch provided input for both the neural network program and the area comparison program.

7.5.1 Neural Network – capturing multiple photographs

In order to train the neural network, it was necessary to provide a large number of photographs which could be analysed by software and used to provide inputs, so that the weights can be set using back-propagation. A button was added to the form of the program which when pressed allowed for the capture and saving of photographs from the video feed (Illustrated in Figure 7.9). Once saved, the photographs could be analysed and data extracted.

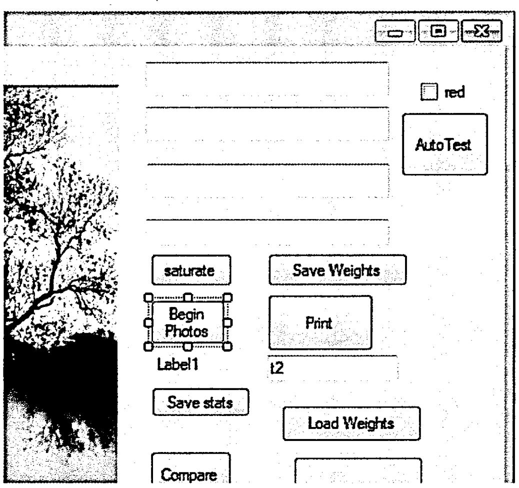


Figure 7.9 showing button for beginning the automatic photographing

The subroutine to allow the photographing to start is shown below (code segment 7.3)

```
Private Sub Btn_BeginPhotos_Click(ByVal sender As System.Object, ByVal e As
 System.EventArgs) Handles Btn_BeginPhotos.Click
    photoNo = 1
    Timer2.Enabled = True
    Label1.Text = "PHOTO"
End Sub
```

Code segment 7.3 showing initialisation of automatic photograph acquisition

This code starts “Timer2” and resets the variable “photoNo” to 1. Timer 2 is then used to interrogate the external button and activate a block of code when the button is pressed. The relevant code is included below (code segment 7.4).

```
Digital_IO(inputs, outputs) ‘read/write to Motor Bee board

          'on each press of external button a photo is saved in the auto folder
If ((inputs AND 1) = 0) Then
  DIP1.Checked() = True
  PhotoSave() ' save current video as a photo

  If photoNo Mod 20 = 0 Then
    System.Console.Beep(500, 100)
  Else
    System.Console.Beep(1000, 100)
  End If
  photoNo = photoNo + 1
Else
  DIP1.Checked() = False
      End If

If photoNo > 60 Then     'once collection is complete the training is started
```

[...]

Code segment 7.4 - interrogation of external button and saving of photos

In this code each time the external button is pressed input 1 goes low, and so ‘inputs AND 1’ becomes false. A check box on the form displays a tick and there is a short beep to indicate that the program has recognised the button being pressed. The subroutine is designed to take three batches of twenty photographs, one batch for each of the three areas: concrete, tarmac and paving. When the last photograph in each batch is acquired a lower-pitched beep is sounded. The subroutine ‘PhotoSave()’ opens a file and saves the current photograph. It assigns a name to each photograph depending on the value of ‘photoNo’, and in this way photographs are labelled according to the region being covered. In order for the system to work successfully the user is required to take 20 photographs of the concrete area, followed by 20 of the tarmac area, and finally 20 of the paved area. In fact any three disparate surface types could be used as the training set.

Once the 60 photographs are analysed the program then runs through the training subroutines adjusting the weights accordingly and then Timer2 is disabled. The neural network can then be tested and the weights saved.

7.5.2 Area Comparison – toggling the outputs

Software code was written that allows the user to switch between the following options in sequence by simply pressing the external button:

1. neither audio output nor vibration motors on
2. audio output only
3. motors only
4. both motors and audio

The section of code that allows for these settings is shown below (code segment 7.5):

```
Digital_IO(inputs, outputs) 'read/write to Motor Bee board
 'detect if button pressed and board connected
    If ((inputs AND 1) = 0 And inputs AND 2 = 1) Then
      Beep(1000, 10)
      switch = (switch + 1) Mod 4
      Label3.Text = switch
      Select Case switch
        Case 0
          noise = False
          BeeON = False
        Case 1
          noise = True
          BeeON = False
        Case 2
          noise = False
          BeeON = True
        Case 3
          noise = True
```

```
        BeeON = True
    End Select
  End If
```
Code segment 7.5

In this code 'switch' is a public integer variable used to track how many times the external switch has been pressed. The Boolean variables 'noise' and 'BeeON' are the same public variables accessed from the form buttons as explained above. This means that the outputs can be controlled either by the external switch or the form buttons. The 'if' condition checks if the external button which is connected to input 1 with a pull-up resistor has been pressed or not. It is also necessary to check if the interface board has been connected or not, and this is done using input 2. If the board is not present then input 2 will read false, and this section of code is ignored. This has the effect of allowing the program to be tested without connecting the interface board so that the vision subsystem and processing of live video, or saved photographs, will still function.

Once the board is connected a pull up resistor connected between input 1 and 5 volts ensures that the physical input 2 is brought high so that the variable 'inputs' will show that input 2 is now on.

7.6 Summary of HCI

In the prototype area comparison program, the human computer interface allows the user to obtain information from the video signal processing program in the form of either audio tones or through vibration motors attached to a belt. An external switch gives the user the ability to toggle the outputs on or off.

The external switch is used with the neural network program to facilitate the acquisition of multiple photographs needed to train to the network.

In both cases the Motor Bee board is used to interface the Microsoft Visual Basic program with the external input and outputs.

7.7 Area Comparison Final Version and Variations

Testing with the area comparison program revealed a problem with timing and speed of walking. As explained, to detect a surface change there must be a difference between the surface in the immediate foreground and that in the test zones. However, if the user passes rapidly across an interface between two surface types, then the system does not have time to respond to the change. As discussed in chapter 8 the average walking speed is approximately 5 km/hour, and at this speed the system does not always indicate a change in surface. As a consequence it was decided to include a fifth area beyond the four test zones. This gives the system more time to detect a surface change. Figure 7.10 shows this variation. Translating this fifth area to a suitable feedback signal involves the use of a fifth tone – higher than the others – and a new pattern of vibration. In this case all four motors are switched on but with a lower frequency to denote that the obstacle change is further away.

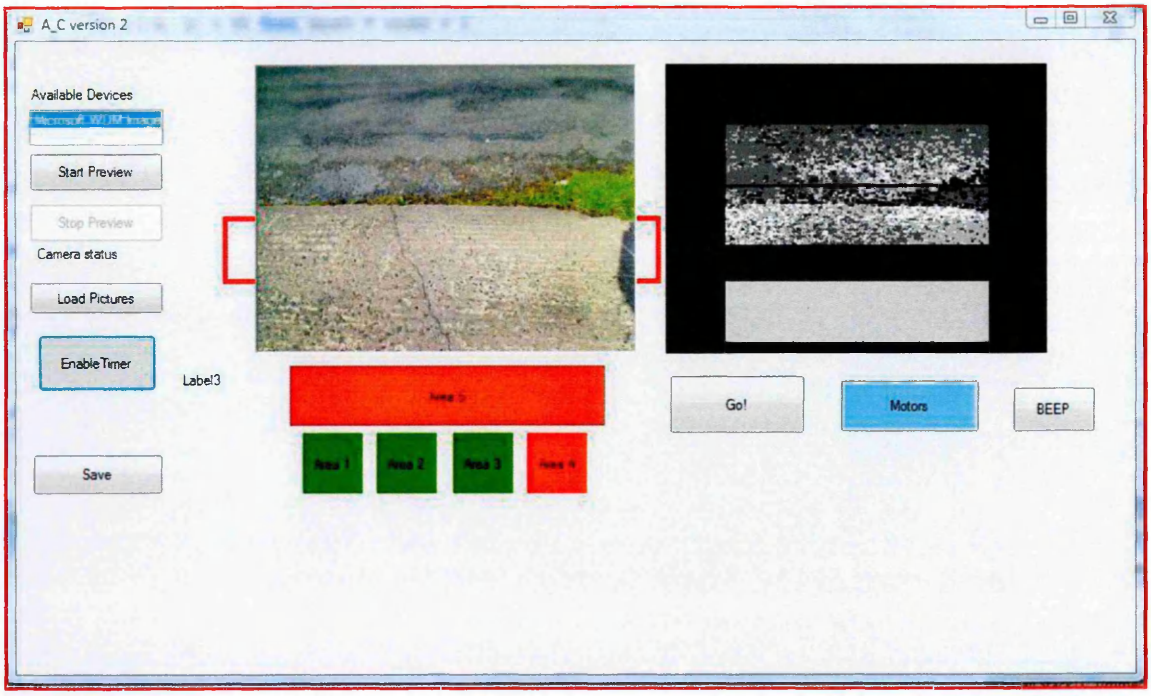


Figure 7.10 Area Comparison Form with five areas.

The right hand test area contains sufficient grass to denote a change in surface whilst ahead of the immediate four areas there is a change from concrete to road surface.

A second variation involved changes to allow for the detection of a straight line running parallel to the direction of travel of the user. To achieve this variation, the area of the safe zone was decreased in width. This meant that if the line was crossed then it would be more likely to be detected. Using this system a line could be kept on one side of the user. Figure 7.11 shows this variation being used to follow a line formed by a change in colour of paving in a pedestrian precinct. The user could position him or herself to the right of the line and use the feedback from Area 1 to keep the line to his or her left. If the user strays too far to the right, then Area 1 will stop giving feedback as the area is no longer different from the narrow safe zone. If the user strays too far to the left, then Area2 will start to show as being active. Such a system could be of benefit in difficult urban environments when a distinction is made between pedestrian comfort zones and other areas by a change in colour.

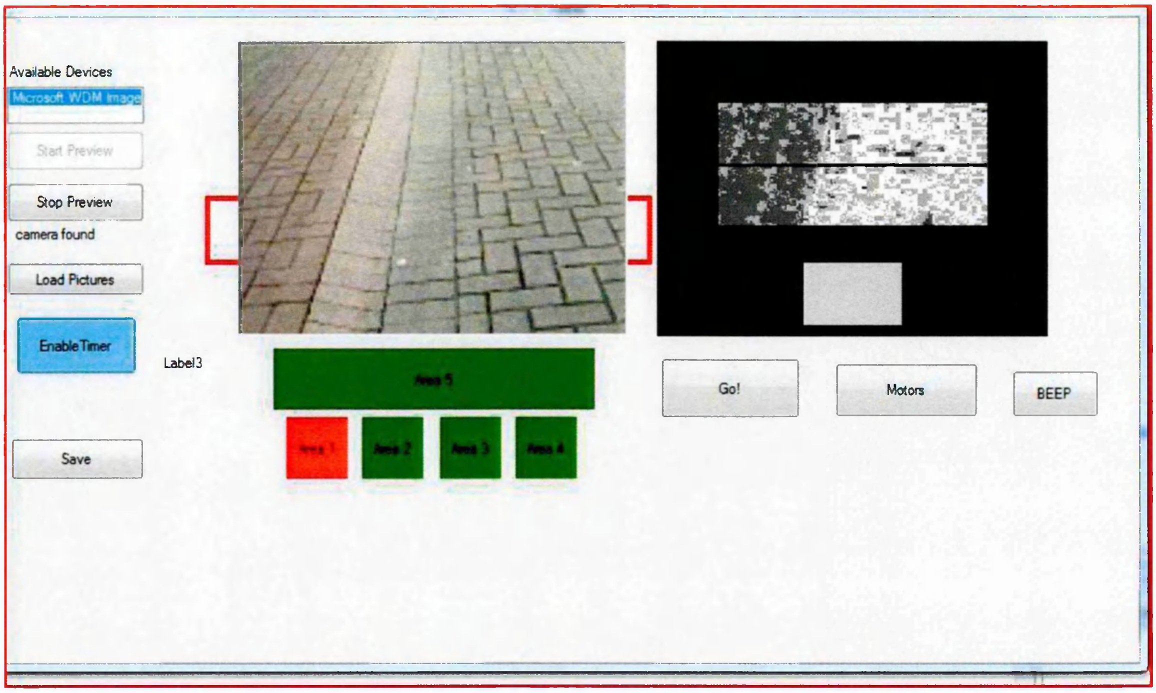


Figure 7.11 Area Comparison Form adjusted to enable user to follow line

Chapter 8 : Experiments to test the efficacy of the prototype in detecting a change in surface colour

8.1 Introduction

The purpose of these experiments is to ascertain if the device is capable of warning the user that the surface type he or she is walking over is about to change. Four experiments were carried out, three indoors and one outside. The indoor experiments were carried out in The George Rzevski Complexity Laboratory in the Mathematics, Computing and Technology Faculty of the Open University in Milton Keynes. The outdoor experiment was carried out in Hibernia Street, Holywood, County Down.

The indoor experiments included a method of calculating the speed of the experimenter to ensure that this variable remains consistent across all trials. As there is likely to be some variation in speed the correlation of the ability of the device in detecting the surface edge with speed can be analysed. Average walking speeds are very dependent on terrain but in "Research on Road Traffic" are quoted as being around 3.4 mph or 5.5 km/h (HMSO, 1965). However, there is some variation in the literature, and the 2003 "Manual on Uniform Traffic control Devices" stipulates a normal walk speed of 1.2 metres per second (cited by LaPlante *et al.*, 2004). This equates to a rather slower speed of 2.7mph or 4.3km/h which, as shall be seen, is closer to the walking speeds attained in the laboratory setting. It is in any case to be expected that indoor ambulatory travel is slower than outdoor walking speeds.

8.2 Indoor Experiments

In these experiments the prototype device was used by the researcher to detect the presence of a change of surface type. Three pieces of A1 card were laid out to make a dark grey rectangle of dimensions 1782 x 841 mm which was placed on the blue-grey carpet as the surface to be detected. Markings were taped onto the carpet at 500mm intervals to provide a means to measure the displacement of the pedestrian testing the device. Figure 8.1 shows a photograph of the experimental set-up including the insert which displays the time in hours, minutes, seconds and thirtieths of a second. A reasonably accurate walking speed can therefore be calculated from this time and the displacement readings of the test subject taken from the taped markings.

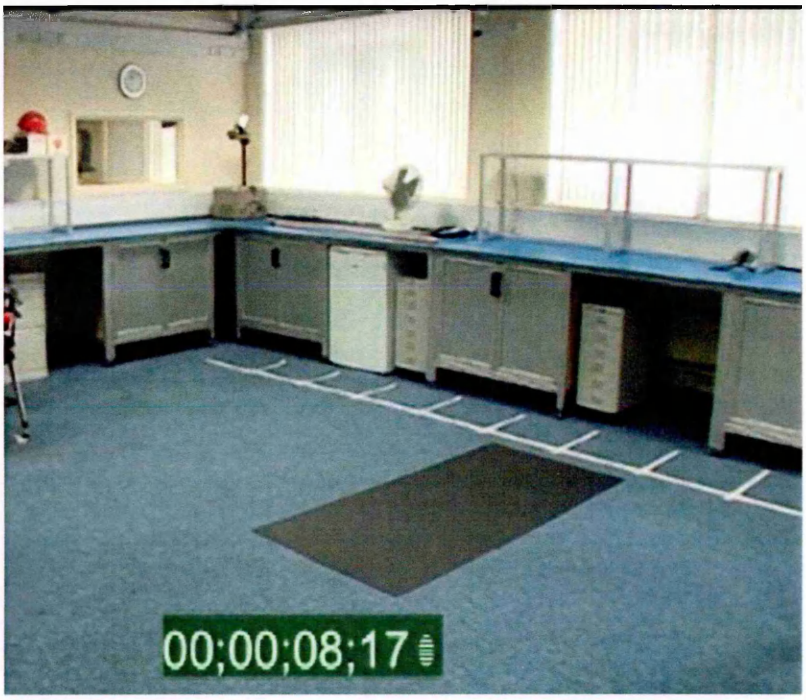


Figure 8.1 The George Rzevski Complexity Laboratory set up for the experiment

The experimenter attached the camera to the belt and walked steadily across the carpet and over the card so that the card was crossed perpendicular to the direction of travel. The audible feedback was used to indicate the two surface changes - the change from carpet to card, and then the second change from card back to carpet. As described earlier, the device compares the current surface with regions just ahead of the experimenter. In this experiment there are 5 test zones and a full response was judged to have occurred if at least 4 out of the 5 zones responded, making for a clear 'bleep'.

False positives were noted and the experiment was repeated ten times with the experimenter walking in the same direction across the card each time (walking from left to right as depicted in the photograph, figure 8.1.) The experimenter can be seen looking at the screen of the laptop which shows the path and gives visual confirmation that the surface change has occurred. The results were recorded in tabular form.

The experiment was conducted three times with different lighting conditions and with 10 trials in each experiment. In the first experiment normal daylight was supplemented by overhead indoor lighting so that the room was well lit (figure 8.2). In the second experiment the lights were switched off and, as it was an overcast day, the room was rather dark (figure 8.3). In the third experiment a second experimenter, Professor Johnson, illuminated the path ahead using a torch (figure 8.4). Figure 8.5 shows the interface of the program at the point where the user has just approached the edge of the rectangle. Without the main lights on, the device produced some false positives due to camera noise; however, these only occurred three times during the actual experimental walks. Decreased ambient light will cause camera noise to be increased although more advanced cameras (including the one used in the experiment) have built-in software to help reduce this (Logitech, 2012).

Figure 8.2 Experiment 1 Normal Lighting

Figure 8.3 Experiment 2 Dim Lighting

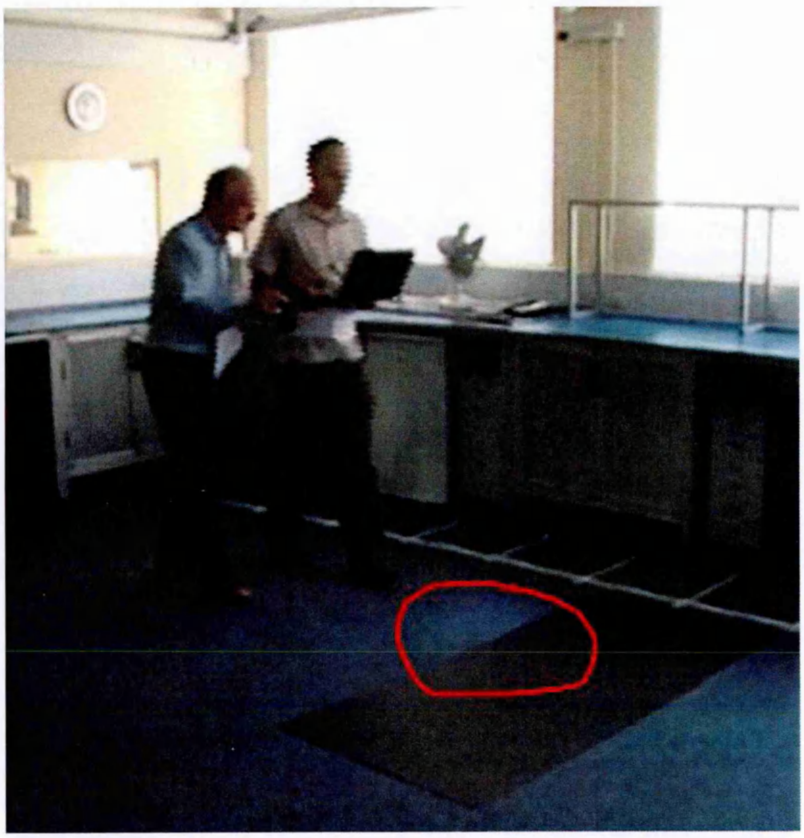


Figure 8.4 Experiment 3 Path lit using torch

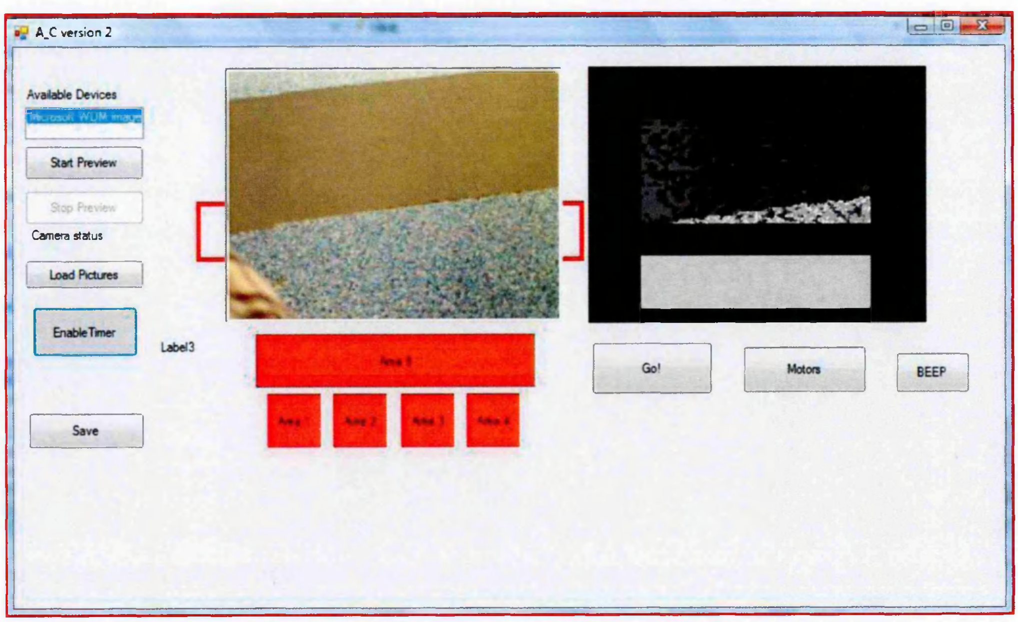


Figure 8.5 Program Interface showing screen as the surface is reached by the device

8.3 *Results of indoor testing*

The full tabulation of results including the calculations of the speeds can be seen in appendix F while a summary table is shown below (Table 8.1).

Experiment	Average Speed km/h	Clear On Transitions	Clear Off Transitions	Uncertain Transitions	Undetected Transitions	False Positives
1. Normal Lighting	3.56	8	10	1	1	0
2. Lights off	3.18	3	5	3	9	1
3. With Torch	3.78	5	7	6	2	2

Table 8.1 Results of Indoor Experiments

The first experiment, with the lights on, resulted in no false positives; 8 clear 'On' transitions; and 10 clear 'Off' transitions. The speed of walking varied from 0.8 m/s (2.9km/h) to 1.1 m/s (4km/h) and the two unclear transitions were both detected at the two highest speeds. One transition was not detected at all; and one did not give a full indication across all five test zones.

In the second experiment, with the lights off, there was one false positive; there were only 3 clear 'On' transitions and 5 clear 'Off' transitions. There were 3 uncertain and 9 undetected transitions. The speed of walking varied from 0.7 m/s (2.5 km/h) to 1.1m/s (3.9km/h), but there appears to be little correlation between walking speed and success of detecting the transitions, with one clear trial occurring at 3km/h (trial 7), and one walk with no detection of transitions occurring at just under 3km/h (trial 8).

The third experiment showed an apparent improvement over the second one. There were 2 false positives, but 5 clear 'On' transitions and 7 clear 'Off' transitions and only 2 undetected transitions. The average walking speed was slightly higher than for experiment 2 – up to 4.1km/h (ranging from 0.9m/s or 3.3km/h up to 1.1m/s or 4.1km/h). So in spite of the slightly higher walking speeds the torch has made a perceptible difference. As in experiment 2 there is little correlation between speeds and success of transition detection; indeed the highest speed in this experiment gave a completely successful trial (trial 3).

8.4 The Outdoor Experiment

This experiment was located in Holywood, Co. Down at a site where there is a change in paving colour but not texture. The area under test is shown in figure 8.6 and is one of the zones used in the test of concept with the blind volunteer.

Figure 8.6 Photograph of area used in outdoor experiment

Weather conditions may have been expected to cause some variance in the results as it was a breezy day with clouds passing across the sun. Hence, some walks occurred in bright sunshine and others in shade. As before the experimenter carried out ten trials and walked in the same direction recording the transition from the general pavement colour to the lighter area outside the building. Figure 8.7 shows the interface from the device program at the point where the experimenter is passing from the dark pavement to the lighter one.

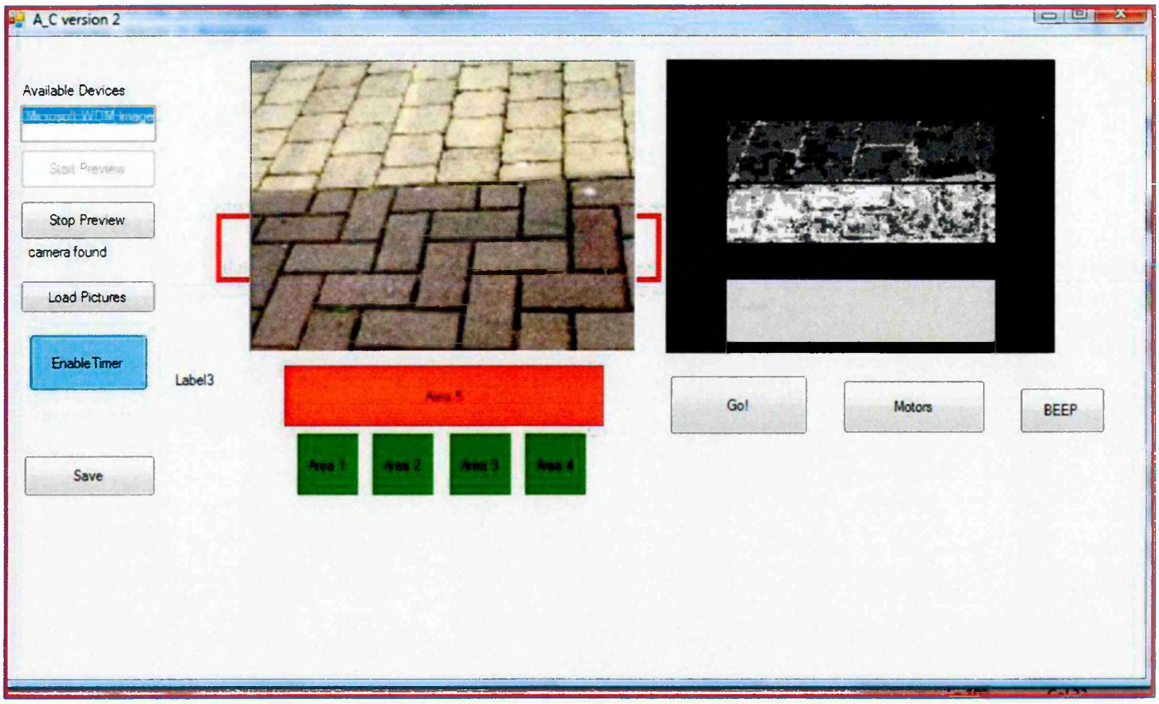


Figure 8.7 Output from Program Interface as user passing from one surface colour to another

8.5 Results of Outdoor Test

Table 8.2 shows a summary of the results while the full results can be found in appendix F (F.4).

Experiment	Clear On Transitions	Clear Off Transitions	Uncertain Transitions	Undetected Transitions	False Positives
4. Outdoor	9	9	2	0	4

Table 8.2 Results of Outdoor Experiment

For the outdoor experiment there were 4 false positives, an increase on those occurring during the indoor experiments. There were 9 clear 'On' transitions, and 9 clear 'Off' transitions with no undetected transitions. Three of the walks were in bright sunlight, and these gave rise to one of the less clear transitions, but also to false positives in each walk. The walks carried out during cloud cover caused just one false positive and one less clear transition.

The increased tendency toward false positives during sunlight is thought to be caused by the production of dark shadow on one side of the path, and reflected sunlight on the other. This can be seen in figure 8.8. A further 5 walks were carried out on a separate day when the area was brightly lit by sunshine. On that occasion there were clear transitions both on and off the lighter pavement for each of the 5 walks, but false positives occurred on each walk caused by the researcher walking close to the building. It is possible that false positives could be reduced by adjusting the sensitivity of the camera.

Figure 8.8 Strong sunlight has produced reflected light and shadows

8.6 Conclusions

The walking speed in all the indoor experiments is slightly below the average walking speed discussed above. Nevertheless, in good lighting the device is able to successfully detect transitions from one surface colour to another with a high degree of reliability. In bright sunlight the device functions well although it is more prone to false positives.

Chapter 9 : Conclusions and Further Work

9.1 Introduction

This chapter bring the thesis to a conclusion. A very short summary of the research is provided before the research questions are reviewed. Some further work is considered and finally contributions of the thesis to wider research areas are considered.

9.2 Brief Summary of the Research and Review of the research questions

The research has shown that users do rely on a variety of non-visual clues when navigating an urban environment. The questionnaire confirmed that people with visual impairment find some urban areas such as Shared Space, Home Zones and pedestrian districts difficult. The participants described such areas as "frightening" and "disorientating" and used words such as "vulnerable", "very uneasy" and "stressed" to describe their feelings when navigating such a space. The questionnaire showed that blind and partially sighted people find it hard to adapt to unfamiliar environments and have a propensity to utilise the same methods of navigation that they use in familiar, everyday settings.

The thesis has also shown that it is possible to interface a webcam to a laptop to provide a system that will discriminate in real time between the surface that the user is currently on and the one about to be walked on. The system suffers from some flaws in that it doesn't react well in strong sunlight, as shadows are not distinguished well from surface changes. However, the work done does show that meaningful feedback can be provided to the user.

Question 1: Do existing assistive technologies form the basis of an urban environment navigational system that can tackle difficult urban environments such as Shared Space?

This question was analysed in the literature review, and the conclusion was that primary aids are not of themselves sufficient in difficult urban environments, and that although secondary aids do exist, these either do not work, or have not been widely taken up by the partially sighted and blind community (section 3.6.4).

Question 2: How can any other information about the urban environment be gathered easily using suitable technology and fed to a blind or partially sighted person in a helpful way?

A prototype device was built on the premise that information about the surface over which the user was travelling would be of use to a blind or partially sighted person. This is further explained in the answer to research question 6. Chapter 4 and particularly section 4.5 describes the methodology behind the development of the prototype device, and how it would employ machine vision and provide audio and tactile feedback for a user. Chapters 6 and 7 describe the different sub-systems used in the prototype in more detail.

Question 3: Can a device be built which will be under the control of the user and which, with some training, will provide sufficient feedback to a blind or partially sighted user so as to increase their confidence when travelling through a difficult urban environment?

Section 3.2.6 concluded, from the literature review on vision substitution research, with the statement that it seemed probable that "a well constructed sensory augmentation device could give a user increased confidence about the terrain that he or she is travelling in". During the research a prototype device was constructed, as described in chapters 6 and 7, while the results of the experiments in chapter 8 indicate that the device does indeed have the potential to provide useful information to the user. The fact that the camera moves with the user is important in helping the user to understand the feedback being relayed to him or her.

Question 4: Can a neural network program classify surfaces, if appropriate pre-processing of images can lead to identifiable inputs, and will that classification be carried out at a sufficient speed and provide enough detail to be of use to a blind or partially sighted user?

It was hoped that a neural network could be constructed that would be able to classify surfaces so as to warn a user that he or she was about to cross onto a different surface type. As described in section 6.5.6, the performance of the network as developed in the research was not sufficient to be reliable, particularly when light conditions changed. It would be incorrect to state categorically that neural networks are incapable of classifying surfaces fast enough and accurately enough to be of benefit to a person with visual impairment. However, it is conceivable that the amount of information that the network would have to store in terms of surface types in different light conditions would make the system impractical in terms of speed or processing power at least for a portable device.

Question 5: Can intensity invariant, colour space components be used to compare the current surface on which the pedestrian is standing with the path ahead so as to give a warning of potential change in surface type or upcoming obstacle?

This question was asked in section 3.4.4, and is answered by the work carried out in developing the prototype device as set out in chapters 6 and 7. Section 6.6 describes the development of the Area Comparison Program which uses colour space components to analyse the immediate foreground and compare that with the path just beyond. The experiments as outlined in chapter 8 show that this system is capable of warning the user about potential change in surface type.

Question 6: Would a person with a long cane or other assistive technology benefit from having information about the surface they are travelling along available as either tactile or audible feedback?

Section 5.5 describes the concept experiment in which Mr Chittick, a 50 year old man registered blind, took part and walked across various surfaces and was informed when the surface changed. The experiment showed that this information could be useful to him in identifying where one particular landmark or feature was found. The surface change was one of colour only and so would not have been picked up by a long cane.

Question 7: How can a device be constructed that will provide such feedback to the user?

The literature review chapter of the thesis discusses various human computer interfaces used (section 3.2), and goes on to expand on two projects that influenced the design of the prototype device (section 3.5). The prototype device was equipped with both audible and tactile feedback as explained more fully in chapter 7 sections 7.3 and 7.4.

Question 8: Can the overall performance of a person with visual impairment making his or her way across the difficult urban environment be improved through knowing what surface they are on?

As discussed in the answer to question 6, the concept experiment demonstrates that a blind or partially sighted person can benefit from having information about the surface they are travelling. It would be necessary to do some further experimentation in situ to demonstrate that this information will enhance the navigational performance of people with visual impairment. A possible experiment is discussed below under Further Work.

Question 9: Can such a device indicate the angle at which the user was moving from one surface to another so that he or she could cross perpendicularly to the edge?

As explained in section 7.3, the prototype device tests four areas situated just beyond the immediate pathway (far right, immediate right, immediate left and far left). More work is needed on how these four regions might be used to orientate a user so that they are positioned perpendicular to the surface interface. This is discussed further under further work.

Question 10: What constitutes a successful urban environment navigational system?

The answer to this question has to do, not only with function, but also with convenience of use. In section 3.6.3 reference is made to the fact that existing electronic travel aids have had limited uptake amongst blind and partially sighted users because of various factors including complexity of use; difficulty of "putting on" the device; lack of sufficient training; and, for some devices, an overwhelming volume of data being communicated to the user. It follows that a successful device should be simple to take up and put on, and transmit only a limited amount of information. Training in any new device should be provided and there should be a systematic way of doing this.

As regards function, a successful system will be one that helps a blind or partially sighted person overcome the existent barriers to use that are currently presented at difficult urban environments, including Shared Space. Appendix G shows 15 different Shared Space sites, the majority of which appear to be laid out in such a way that navigation could be enhanced through the use of a device such as the prototype described in this thesis.

The prototype device described in this thesis needs further refinement and testing before it could be regarded as a successful device.

9.3 Further Work

9.3.1 Experiment to ascertain whether device increases confidence

Although it has been shown that the prototype device described in this thesis can give helpful information to a blind or partially sighted person about the surface they are walking on, further work would be required to establish that the overall performance of a person with visual impairment, making his or her way across a difficult urban environment, might thereby be significantly improved. A suitable experiment would be to conduct an experiment with a number of volunteers in an actual environment, such as a Shared Space or a simulation of it, where it is known that some difficulty in terms of confidence in navigation is encountered. The volunteers would be asked to navigate the environment following a specific route, and a record made in terms of time taken and feelings of stress. Some training in the use of the device would be required and then the volunteers asked to repeat the navigational exercise. Ideally a second control group would carry out two complete navigations of the same route, but without the use of the device on the second run to ascertain how much any change in confidence was due to familiarity with the route, since the volunteers would already have had one run through.

9.3.2 Experiment to determine if device can help in crossing perpendicular to the edge

Research question 9 asked if the prototype device could indicate the angle at which the user was moving from one surface to another, so that he or she could cross perpendicularly to the edge. This question was raised as users of a long cane use a cane to determine the edge of a kerb, and then by tapping the cane to the right and left of themselves determine the line of the kerb, so that they can move perpendicularly away from it. Moving perpendicularly away gives the best chance of arriving at the opposite side of the crossing in as short a distance as possible. The removal of the kerb prevents the cane user from using this technique.

It is possible for the user to stand in one position at the edge of a kerb or other interface such that all four areas show as unsafe. If he or she then orientates the camera by turning to the left and right then it is possible to obtain a reaction from the far left zone and then the far right zone. This action can be used to give a rough measure of the line of the kerb. However, more work is needed on whether this is a reliable method, and it is also likely that more zones may have to be incorporated into the device to achieve the aim of crossing a shared surface area at right angles to the vehicular path.

9.3.3 Developing a successful Difficult Area navigational system

The prototype device can give useful information to a blind or partially sighted person about the surface he or she is travelling on. Work is needed to refine the program and obtain the best possible settings in terms of sensitivity to shadows and changes in surface colours. Ideally it should be possible to individualise these settings so as to make the device suit the personal requirements of individual users.

Rather than the user carrying around a laptop, the program would need to be written for a more convenient platform, such a mobile phone. It should be possible to use the camera in the mobile phone to acquire the video feed, and then the processor in the phone to carry out the signal processing. Communication with the microcontroller driving the vibration motors could be conducted wirelessly, either using Bluetooth or Wi-Fi. The mobile phone could be fastened to the vibration motor belt using a suitable attachment. The vibration motor belt itself needs some redesigning to make it more aesthetically pleasing and more robust.

The prototype can be used to detect surface colour change. As mentioned above such colour changes are used in some Shared Space regions to denote a comfort zone which is safer for people with disability, but there needs to be an increased nationwide consistency in the use of colours and surface textures if technological devices, such as the prototype described here, are to be of benefit to a wide number of people.

9.4 Contributions and original work of the thesis

Original work carried out in the research includes:

- The development of a questionnaire to ascertain the use of non-visual cues in different urban environments. The results of the questionnaire have been analysed using a valid statistical method.
- The development of a prototype navigational aid, which is shown by experiment to be of potential benefit to blind and partially sighted users when navigating difficult urban environments.

The thesis has added a voice to the discussion of Shared Space development in the United Kingdom and made valid proposals regarding their implementation.

The thesis has made the following contribution in the field of machine vision and vision substitution.

- The concept of colour space analysis to the problem of obstacle detection and surface change warning has been applied in the context of navigation for blind and partially sighted individuals.

9.5 Final Conclusion

Machine vision devices can be helpful to people with visual impairment. This thesis describes the construction of one device which could conceivably be used successfully to increase a blind or partially sighted person's confidence when navigating a difficult urban environment, such as Shared Space. The prototype navigational aid described has been achieved using low cost equipment, and could be migrated to a smaller computing platform such as a mobile phone with suitable adaptations. It is intended that the aid would be a secondary aid and, as such, act as a complement to the user's primary aid, be it a long cane or a guide dog. If comfort zones are marked with a contrasting colour, and careful use is made of colour and architectural features, then the barriers that present to blind and partially sighted people navigating difficult urban environments can be much reduced through the use of this technology.

References

Abikoye, O. C., Mabayoje, M. A., Ajibade, R.,(2011) Offline Signature Recognition & Verification using Neural Network, *International Journal of Computer Applications*; Dec2011, Vol. 35, p44-51

Abraham, T. H. (2002), (Physio)Logical Circuits: The Intellectual Origins Of The Mcculloch–Pitts Neural Networks, *Journal of the History of the Behavioral Sciences*; Winter 2002, Vol. 38 Issue 1, p3-25, 23p Accessed from http://www.kip.uni-heidelberg.de/vision/publications/vtd.html (Heidelberg University site) Jan 2006

Açik, A. and Kabisch, C. (2005) "Orientation in Natural Environment" Appendix H of feelSpace Final Report , available from http://cogsci.uni-osnabrueck.de/~feelspace/downloads/feelSpace_finalReport.pdf accessed August 2012

Bach-y-Rita, P. (1984) "Tactile Vision Substitution System", Chapter from *Cognition and Motor Processes,* Ed. By Prinz, W. And Sanders A.F., Springer-Verlag, Berlin, Heidelberg

Bach-y-Rita , P. and Kaczmarek, K. (2003) "Seeing with the Brain" *International Journal Of Human–Computer Interaction*, 15(2), p 285–295; Lawrence Erlbaum Associates, Inc.

Bach-y-Rita, P. and Kercel, S. (2003) "Sensory substitution and the human-machine interface", *Trends in Cognitive Science";* December, 7 12, p541-p546

Bach-y-Rita, P., Collins, C. C., Saunders, F., White, B., & Scadden, L. (1969),Vision substitution by tactile image projection, *Nature*, 221, 963–964.

Baldwin D., (2003), Wayfinding Technology: A Road Map to the Future, *Journal of Visual Impairment & Blindness*; Oct., Vol. 97 Issue 10, p612-620

BBC (2012) "Thinking Streets" Broadcast January 2012 Available at http://www.bbc.co.uk/programmes/b018xs8t accessed August 2012

BDD (n.d.) Brenner Dietrich Dietrich: "Diplomingenieure Freie Architekten Büro Für Städtebau Freiburg" (Office of qualified architectural engineers, Freiburg) available from http://www.bdd-stadtplanung.de/buero0.html accessed December 2012

Beale, L; Field, K; Briggs, D; Picton, P; Matthews, H. (2006) "Mapping for wheelchair users: Route navigation in urban spaces", Cartographic Journal; MAR, 43; 1; p68-p81

Benjamin, J. M., 1974, "The Laser Cane", *Bulletin of Prosthetics Research*, Fall; p 443-450

Bentzen, B.L., Mitchell, P.A., (1995) Audible signage as a wayfinding aid: Verbal Landmark versus Talking Signs, *Journal of Visual Impairment & Blindness*; Nov/Dec Part 1 of 2, Vol. 89 Issue 6, p494

Biesiadecki, J.J. and Maimone, M.W. (2006) "The Mars Exploration Rover surface mobility flight software: Driving ambition," *Proc. IEEE Aerospace Conf.*, Big Sky, MT available from http://www-robotics.jpl.nasa.gov/publications/Mark_Maimone/biesiadecki_maimone06.pdf accessed July 2012

Bird, J., Holland, S., Marshall, P., Rogers, Y. and Clark, A. (2008) "Feel the force: using tactile technologies to investigate the extended mind", *Proceedings of Devices that Alter Perception* (DAP08), pp. 1-4.

Bird, J., Marshall, P. and Rogers, Y. (2009), "Low-Fi Skin vision: a case study in rapid prototyping a sensory substitution system", *Proceedings of the 2009 British Computer Society Conference on Human-Computer Interaction*, 1-5 Sept 2009, Cambridge, UK

Boone, D. and Boone, C. (1997) "Beyond city sidewalks", *American Rehabilitation,* Autumn/Winter, Vol. 23 Issue 3/4, p2

Bruce, I., McKennell, A., Walker, E. (1991) *Blind and partially sighted adults in Britain: the RNIB survey, Volume 1,* HMSO, London.

Byrd, T. (2007), "Deaf Space" available from http://www.gallaudet.edu/communications_and_public_relations/gallaudet_today_magazine/deaf_space_spring_2007.html last accessed December 2012

Cholewiak, R.W., Brill, J.C. and Schwab, A (2004), "Vibrotactile localization on the abdomen: Effects of place and space", *Perception & Psychophysics*, 66 (6), 970-987

Carl, C., Kringe, T., Lörken, C., Märtin, R., (2005)"Hardware" chapter from "feelSpace, Report of a Study Project", University of OsnabrÄucket pages 82-84 available from www.ee.lamar.edu/eelabs/documents/feelSpace_finalReport.pdf, *accessed Nov 2011*

Clarke, D (2001) "Visual Basic and the future of Web-application development", *Computer*, Volume: 34 , Issue: 8, Page(s): 16 - 18

Cowles, J. (2008) "Sobel Edge Detector in VB.NET", available from http://visualcore.com/index.php/2008/03/sobel-edge-detector-in-vb-net/ accessed Feb 2012

Cowdrey, D., (2011) "Shared Space – Did the DfT go far enough", available from http://www.guidedogs.org.uk/news/blogs/taking-the-lead/shared-space-did-the-dft-go-far-enough/?tx_ne_blogs[showUid]=6269&cHash=a9e34a70c213d2e1d2ec0a4eb064ef3e Accessed July 2012

Crosier, A. (April 2009) "A summary of research into the barriers and experiences of 'shopping' and 'money matters' among blind and partially sighted people" , accessed July 2011 from http://www.rnib.org.uk/livingwithsightloss/Documents/go_shop_summary.doc

Debus,T., Jang, T., Dupont,P. and Howe, R. (2004)"Multi-Channel Vibrotactile Display for Teleoperated Assembly", *International Journal of Control, Automation, and Systems*, vol. 2, no. 3, pp. 390-397, September

DfT, 2011a Department of Transport, "Inclusive Mobility" available from http://www2.dft.gov.uk/transportforyou/access/peti/inclusivemobility_pge_4-.html?page=4 accessed July 2011.

DfT 2011b Department of Transport, "Local Transport Note 1/11 Shared Space", Published by TSO (The Stationery Office), Norwich and available online from assets.dft.gov.uk/publications/ltn-01-11/ltn-1-11-quantitative.pdf accessed July 2011

Disability Discrimination Act, (2005) Available from http://www.legislation.gov.uk/ukpga/2005/13/contents, accessed December 2012

Douglas, G., Corcoran, C. And Pavey, C. (2006), "Network 1000, Opinions and circumstances of visually impaired people in Great Britain: report based on over 1000 interviews", available from http://www.rnib.org.uk/aboutus/research/reports/2010/network_1000_opinions.doc accessed August 2012

Erin, J. (2009), "Travelling by Touch: How Useful are Tactile Maps", *Journal of Visual Impairment and Blindness,* May, Vol. 103 Issue 5, p265-266

Equality Act, (2010) Available from http://www.legislation.gov.uk/ukpga/2010/15/contents, accessed December 2012

Farmer L, Smith D. (1998) "Adaptive technology" , in: Blasch B, Wiener W, Welsh R, editors. Foundations of orientation and mobility. 2nd Ed. New York: American Foundation for the Blind; p. 231-59.

Farrington, A. (2011) Image Copyright Andy Farrington. This work is licensed under the Creative Commons Attribution-Share Alike 2.0 Generic Licence. Image available at http://www.geograph.org.uk/photo/2556070 accessed August 2012

Five Roads Forum (2011)"Pictures of Changes Underway" available from http://www.fiveroadsforum.org/inprogress/index.htm accessed August 2012

Fletcher (2002): "Seeing with sound: A journey into sight" available from http://www.seeingwithsound.com/tucson2002.html accessed Jan 2006

Gilblom, D. and Yoo, S.K. (2004): "Infrared and ultraviolet imaging with a CMOS sensor having layered photodiodes", *Proceedings- SPIE The International Society For Optical Engineering,* Vol. 5301, pages 186-192

Gillet, P. (2009) Image Copyright Paul Gillett. This work is licensed under the Creative Commons Attribution-Share Alike 2.0 Generic Licence. Image available at http://www.geograph.org.uk/photo/1297201 accessed August 2012

Gillies, A. (2009) "Is the road there to share? Shared Space in an Australian Context" Thesis Project; University New South Wales. Available from http://www.fbe.unsw.edu.au/schools_and_engagement/resources/_notes/5A2_44.pdf. Accessed July 2011

Golledge, R.G., Blades, M., Kitchin, R. and Jacobson, D. (1999) Final Report - Understanding geographic space without the use of vision. NSF (National Science Foundation) Report SBR95-14907. pp. 348

Gravestock, P. (2001) "Providing Learning Support for Blind or Visually Impaired Students Undertaking Fieldwork and Related Activities" Available from http://www2.glos.ac.uk/gdn/disabil/blind/ch10_1.htm . Accessed August 2012

Guide Dogs (2006a), "Shared Surface Street Design Research Project the Issues: Report of Focus Groups". Available from http://www.alanhunt.co/pdf/Report of UK Focus Groups.pdf, accessed June 2012

Guide Dogs (2006b), "Shared Surface Street Design: Report of focus groups held in Holland". Available from http://www.alanhunt.co/pdf/Report_of_Holland_Focus_Groups.pdf, accessed June 2012

Guide Dogs (2010) "Inclusive Streets: Design principles for blind and partially sighted people". Available from http://www.guidedogs.org.uk/fileadmin/gdmain/user/What_we_do/Shared%20Surfaces/Documents/Inclusive_Streets_Design_Principles_booklet_Guide_Dogs_2010.pdf accessed August 2012

Hamilton-Baillie, B. (2005) "Shared Space – Safer Streets" available at http://www.hamilton-baillie.co.uk/_files/_publications/8-1.pdf accessed August 2012

Hamilton-Baillie, B. (2006) Cited in Appendix 1 of *Shared Surface Street Design Research Project; The Issues: Report of Focus Groups* Published by Guide dogs and available at http://community.stroud.gov.uk/_documents/23_Shared_Surface_Street_Design_Research_Project.pdf accessed August 2012

Hamilton-Baillie, B. (n.d.) "about hamilton-baillie associates" available at http://www.hamilton-baillie.co.uk/index.php?do=about . Accessed July 2012.

Haviland, M.G. (1990) "Yates's correction for continuity and the analysis of 2 x 2 contingency tables", *Statistics in Medicine*, Volume 9, Issue 4, pages 363–367, April

Heaton, J. (2008) "Introduction to Neural Networks for C#", 2nd edition, St Louis, Heaton Research

Heylighen, H., Devlieger P. "in dialogue with (dis)-Ability", *The International Society of City and Regional Planners*, 43rd Congress 2007. Available from http://www.isocarp.net/Data/case_studies/1001.pdf accessed July 2012

HMSO (1965) Road Research Laboratory "Research on Road Traffic", Ed. JB Behr, pp. 505, Hopfield. J.J. and Tank D.W. (1985) "'Neural' Computation of Decisions in Optimization Problems", *Biological Cybernetics*, 52, 141-152

IHIE, (2002), *"Home Zone Design Guidelines"*, pub: by The Institute of Highway Incorporated Engineers (IHIE), (ISBN 0954287509).

Intel, (n.d.), "Intel Universal Serial Bus (USB) Frequently Asked Questions (FAQ)", available from http://www.intel.com/content/www/us/en/io/universal-serial-bus/universal-serial-bus-faq.html accessed Jan 2012

Interreg IIIB, (2003), "Shared Space - Reconciling people, places and transport", available from http://northsearegion.eu/iiib/projects/details/&tid=14 accessed 28 July 2008

Jaqian (2009) Photograph of O'Connell Street Dublin & Jim Larkin, (file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license) available from http://en.wikipedia.org/wiki/File:O%27Connell_Street_Dublin_%26_Jim_Larkin.JPG accessed August 2012

JCMBPS, (2008) Joint Committee on Mobility of Blind and Partially Sighted People, "Policy statement on Public Realm (and access considerations)" available from http://www.google.co.uk/url?sa=t&source=web&cd=1&sqi=2&ved=0CCYQFjAA&url=http%3A%2F%2Fwww.jcmbps.org.uk%2Ffileadmin%2Fgdba%2Fdownloads%2FJCMBPS%2FJCMBPS_2008_updates%2FPolicy_Statement_on_the_Public_Realm.doc&rct=j&q=%22pedestrian%20precincts%22%20access%20disabled%20partially%20sighted&ei=Was9Tpf4G8WF-wbGx7nTDg&usg=AFQjCNES75nAtyVQp4TzksMYotm3gFjVQg&sig2=eTHY318DT_brJxMa8lI9Pg&cad=rja accessed July 2011

JMU Access Partnership, (2007) "Designing for Disabled People in Home Zones" Available from http://www.homezones.org.uk/public/guidance/documents/finalreport30-06-07.pdf accessed August 2012

Johnson, J. and Picton, P. (1999) "Mechatronics, Concepts in Artificial Intelligence, Volume 2, Designing Intelligent Machines", Butterworth-Heinemann, Oxford, in association with the Open University

Jones, L. A., & Sarter, N. B. (2008), "Tactile displays: guidance for their design and application", *Human Factors*, 50, 90–111

Kaczmarek 2004: "Electrotactile Display of Computer Graphics for Blind, Final Report", accessed from the University of Wisconsin website at http://kaz.med.wisc.edu/Publications/2004-Kaczmarek-ETH-Final-Report.pdf Accessed January 2006

Katramados, I., Crumpler S., Breckon, T.P. (2009) "Real-Time Traversable Surface Detection by Colour Space Fusion and Temporal Analysis", *Proceedings International Conference on Computer Vision Systems*, Springer, Volume 5815, pp. 265-274

König, P., Açik, A. Bernhard, B.C., Carl, C., Dombrowe, I., Honey, C., Jansen,L., Kabisch, C., Kringe, T., Lörken, C., Märtin, R., Nagel, S., Saal, H., Stefaner, M., Stößel, C., (2005) feelSpace - Report of a Study Group. *Proceedings of the 6th Meeting of the German Neuroscience Society / 30th Göttingen Neurobiology Conference 2005*, Eds: H. Zimmermann and K. Krieglstein, 1 Suppl.: 434B. Neuroforum, Göttingen, Germany available from http://cogsci.uni-osnabrueck.de/~feelspace/downloads/feelSpace_finalReport.pdf accessed August 2012

Kröese, B.J.A., Dev, A., Groen, F.C.A. "Where are you driving to? Heading direction of a mobile robot from optical flow", *Image and Vision Computing* 18(5), 415–424 (2000)

Kupers R, Ptito M, 2011, "Cross-modal brain plasticity in congenital blindness: Lessons from the tongue display unit", *i-Perception* 2(8) 748, Available from http://i-perception.perceptionweb.com/journal/I/article/ic748 accessed April 2012

Kurachi K, Kitakaze, S., Fujishima, Y., Watanabe, N., Kamata, M (2005), "Integrated Pedestrian Guidance System Using Mobile Device, *12th World Congress on ITS*. San Francisco. USA. Available from http://www.csun.edu/cod/conf/2005/proceedings/2210.htm Accessed July 2008.

La Grow, S., (1999), "The use of the Sonic Pathfinder as a secondary mobility aid for travel in business environments: a single-subject design", *Journal of Rehabilitation Research and Development*, Vol. 36 No. 4, Oct)

Langdon, P. (2008), "'Shared-space' streets cross the Atlantic" ,New Urban Network, available from http://newurbannetwork.com/article/%E2%80%98shared-space%E2%80%99-streets-cross-atlantic, accessed July 2011

Langley, N. (2004), "Visual Basic eases programmers' path for creating Windows apps", *Computer Weekly,* November, p60.

LaPlante, J. N., Kaeser, T.P., (2004) "The Continuing Evolution of Pedestrian Walking Speed Assumptions", *Institute of Transportation Engineers. ITE Journal 74. 9* Sept, pp 32-40.

Legislation (1997), "The Zebra, Pelican and Puffin Pedestrian Crossings Regulations and General Directions" Available from http://www.legislation.gov.uk/uksi/1997/2400/schedule/1/made accessed August 2012

Lenay C. (2001) "The Constitution of Spatiality in Relation to the Lived Body: a Study based on Prosthetic Perception", *Conference on the Emergence and Development of Embodied Cognition* (Beijing, China: August 2001); available from: http://www.ifi.uzh.ch/ailab/people/lunga/Conferences/EDEC2/CharlesLenay.pdf Accessed January 2007

Logitech, (2012) "RightLight™ 2 technology. Be seen in a better light" Available from http://www.logitech.com/en-us/webcam-communications/webcams/6600 accessed August 2012

Lorigo L. Brooks R., Grimson E. (1997) "Visually-Guided Obstacle Avoidance in Unstructured Environments", In *IEEE Conference on Intelligent Robots and Systems*, volume 1, pages 373–379, Sept.

Marks, R. (2010) "EyeToy, Innovation and Beyond" available from http://blog.us.playstation.com/2010/11/03/eyetoy-innovation-and-beyond/comment-page-2/#comment-478157, accessed Jan 2012

Maucher,T., Schemmel, J. and Meier, K. (2000): "The Heidelberg Tactile Vision Substitution System", Paper presented at the 6th *International Conference on Tactile Aids, Hearing Aids and Cochlear Implants*, ISAC2000, Exeter, May 2000 and at the *International Conference on Computers Helping People with Special Needs*, ICCHP2000, Karlsruhe, July.

Matthews, H., Beale, L., Picton, P., Briggs, D., (2003) "Modelling Access with GIS in Urban Systems (MAGUS): Capturing the Experiences of Wheelchair Users", Area. Mar., 2003, Vol. 35, Issue 1, p34-45, 12p.; Blackwell Publishing; The Royal Geographical Society (with the Institute of British Geographers)

McCulloch, W.S. and Pitts, W., (1943), "A Logical Calculus of the Ideas Immanent in Nervous Activity", *Bulletin of Mathematical Biophysics,* 5: 115-137

Meijer, P. (1992)," An Experimental System for Auditory Image Representations", *IEEE Transactions on Biomedical Engineering*, Vol. 39, No. 2, pp. 112-121, Feb 1992

Meijer, P.(2002), "Seeing with sound for the blind: Is it vision?" Paper presented at "*Toward a Science of Consciousness*" Tucson, AZ. April 8-12, 2002. Available at http://www.seeingwithsound.com/tucson2002.html, accessed January 2006

Merabet L. , Poggel D (2008) "Activation of visual cortex using crossmodal retinotopic mapping", Poster presented at the *14th Annual Meeting of the Organization for Human Brain Mapping* (OHBM) June 15-19, 2008, Melbourne, Australia. Available from http://www.artificialvision.com/hbm2008.html accessed July 2008

Microchip (2000) http://ww1.microchip.com/downloads/en/devicedoc/41124c.pdf accessed November 2011

Microsoft, (2011) "About Video Capture", available from http://technet.microsoft.com/en-us/query/dd742882 , accessed February 2012

Minsky, M. and Papert, S. (1969) *Perceptrons: An Introduction to Computational Geometry*, The MIT Press, Cambridge MA

MSDN, (n.d. a) "Interaction.Beep Method", available from http://msdn.microsoft.com/en-us/library/microsoft.visualbasic.interaction.beep.aspx, accessed February 2012

MSDN, (n.d. b.) "Bitmap.LockBits Method", available from http://msdn2.microsoft.com/en-us/library/5ey6h79d.aspx, accessed February 2012

MVA Consultancy (2009) *Stage 1: Appraisal of Shared Space.* DfT available online from http://www.dft.gov.uk/publications/shared-space-appraisal accessed August 2012

MVA Consultancy (2011a) *Shared Space: Operational Assessment.* DfT Available online from http://assets.dft.gov.uk/publications/ltn-01-11/ltn-1-11-quantitative.pdf accessed August 2012

MVA Consultancy (2011b) *Shared Space: Qualitative Research.* DfT Available online from http://assets.dft.gov.uk/publications/ltn-01-11/ltn-1-11-qualitative.pdf accessed August 2012

NCBI (2011), National Council for the Blind of Ireland "The impact of shared space on people who are blind or vision impaired" available from :http://www.ncbi.ie/news/public-policy-and-campaigns/the-impact-of-shared-space-on-people-who-are-blind-or-vision-impaired, accessed July 2011

NTC (National Transport Commission). (2009). Australian Road Rules. Available from http://www.ntc.gov.au/ViewPage.aspx?documentid=00794, accessed July 2011

Ojala, T., Pietikäinen M. and Harwood, D. (1996) A comparative study of texture measures with classification based on feature distributions, Pattern Recogn., 29 (1) , pp. 51–59

Omnivision (2005), "Data sheet on OV7648 (colour) and OV7148 (black and white)" available from http://www.comedia.com.cn/OV%20SENSOR%20PB/30W/OV7648_PB%20(1.7).pdf, accessed Jan 2011

Owen, F. and Jones, R. (1982), *"Statistics, Second Edition"*, Polytech Publishers Ltd. Stockport, Ashworths Print Services, Bury, Lancs.

Pakatak (n.d), "Wireless Pinhole Camera", available from http://www.pakatak.co.uk/shop/product.asp?strParents=72&CAT_ID=100&P_ID=264, accessed Jan 2012

Phillips, C., (2011), "Getting from Here to There and Knowing Where: Teaching Global Positioning Systems to Students with Visual Impairments", *Journal of Visual Impairment & Blindness,* October, Vol. 105 Issue 10, p675-680

Picton, PD; Capp, MD. (2008) "Relaying scene information to the blind via sound using cartoon depth maps", Image And Vision Computing; APR 1; 26; 4; p570-p577

Pitchford, J. (2008), "Engagement with the visually handicapped in Ipswich" in *Shared Space, Partner Publication* available at http://www.senza.nl/userfiles/files/SharedSpacePartnerpublica.pdf, accessed July 2012

Precision Microdrives, (2011) http://www.precisionmicrodrives.com/new-vibration-motors/310-101-10mm-coin-vibrator-motor accessed January 2012

Q-Imaging (2006) 'Firewire vs USB 2.0' available from http://www.qimaging.com/resources/pdfs/FirewireUSB.pdf accessed Jan 2012

Ripley, B.D. (1996) "Pattern Recognition and Neural Networks", Cambridge University Press, Cambridge, U.K. pp 168-173

RNIB, (2009a), Royal National Institute for the Blind report on living with sight loss, available from http://www.rnib.org.uk/livingwithsightloss/documents/go_shop_conditions.doc. Accessed July 2011

RNIB, (2009b) Royal National Institute for the Blind, "Press Statement condemning the introduction of a shared surface scheme at Exhibition Road", available from http://www.rnib.org.uk/aboutus/mediacentre/mediareleases/media2009/Pages/media release26Oct2009.aspx accessed Sept 2011

RNIB, (2012) "Mobility and independence: School Age", available from http://www.rnib.org.uk/professionals/Documents/mobility_and_independence_at_sch ool.doc accessed August 2012

Rosenblatt, F. (1958) "The perceptron: a probabilistic model for information storage and organization in the brain", *Psychological Review* 65, pp. 386-408

Sadato, N., Pascual-Leone, A., Grafman. J., Ibañez, V., Deiber, M.P., Dold, G., Hallett, M., (1996) "Activation of the primary visual cortex by Braille reading in blind subjects", *Nature* (380):526-528

Science News Letter, (1958) Vol. 74. No. 3. Jul 19, Published by Society for Science and the Public

Shan, Y., Yang, F., Wang, R. (2007) "Color space selection for moving shadow elimination", *Fourth International Conference on Image and Graphics* (ICIG), pp. 496–501

Shared Space (n.d.) Interreg III Project available from http://www.shared-space.org/ last accessed June 2010

Sørensen M.W.J. (2011) "Shared space in Norway (...and in Europe)" , available at www.nvfnorden.org/lisalib/getfile.aspx?itemid=4649, accessed June 2012

Stepankova, K. , Pastalkova, E. , Kalova, E. , Kalina, M. , and Bures, J. (2003) "A battery of tests for quantitative examination of idiothetic and allothetic place navigation modes in humans", *Behavioural Brain Research*, 147: 95-115

STMicroelectronics (1996), "Push-pull four channel driver with diodes", L293D data sheet available from http://www.datasheetcatalog.org/datasheet/stmicroelectronics/1330.pdf accessed December 2012

Stolpmann A. and Dooley L. S., (2000) "A Texture Classification System Using Statistical and Soft-Computing Methods", published in "Texture Analysis in Machine Vision", Editor Pietikäinen, (Series in *Machine Perception Artificial Intelligence* Volume 40) World Scientific Publishing Co. Pte. Ltd. Singapore

Stott, R. (2012) Image Copyright Robin Stott. This work is licensed under the Creative Commons Attribution-Share Alike 2.0 Generic Licence. Image available at http://www.geograph.org.uk/photo/2978539 accessed August 2012

Tan, H.Z. and Pentland, A. (1997) "Tactual displays for wearable computing", *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, page 84

Tang, H. and Beebe, D. "A Microfabricated Electrostatic Haptic Display for Persons with Visual Impairments", *IEEE Transactions On Rehabilitation Engineering*, Vol. 6, No. 3, September 1998

Texas Instruments, (n.d.) "16 Channel LED Driver with Dot correction and Grayscale PWM Control" Data sheet available from http://www.ti.com/lit/ds/symlink/tlc5940.pdf, accessed December 2012

Toney, A. , Dunne,L., Thomas,B. and Ashdown, S.P., (2003), "A shoulder pad insert vibrotactile display", *SeventhIEEE International Symposium on Wearable Computers*, pages 35-44, White Plains, New York, 2003. IEEE Computer Society.

Translink (2005), "Bus Stop Design Guide", available from http://www.planningni.gov.uk/index/policy/supplementary_guidance/guides/busstop-designguide.pdf accessed August 2012

Transport for Greater Manchester (2007) "Design Guidelines for Bus Stops in Greater Manchester" available from http://www.tfgm.com/upload/library/07_0650_bus_stop_guidelines.pdf accessed August 2012

V & A (2010) Victoria and Albert Museum "aberrant architecture - Architecture Residents 2010 Week 6 | School Report: 15.02.10" available at http://www.vam.ac.uk/b/blog/aberrant-architecture/week-6-school-report-150210 last accessed December 2012

Vaisla, K., Bhatt, A. (2010) "An Analysis of the Performance of Artificial Neural Network Technique for Stock Market" , *International Journal on Computer Science & Engineering*; 2010, Vol. 1 Issue 6, p2104-2109

van der Linden, J., Schoonderwaldt, E., Bird, J. (2009), "Towards a Real-Time System for Teaching Novices Correct Violin Bowing Technique", *IEEE International Workshop on Haptic Audio visual Environments and Games*

van der Linden, J., Schoonderwaldt, E., Bird, J., Johnson, R. (2011), "MusicJacket—Combining Motion Capture and Vibrotactile Feedback to Teach Violin Bowing", *IEEE Transactions on Instrumentation and Measurement,* Volume: 60 , Issue: 1, Page(s): 104 – 113

Wang, Y.J., Picton, P., Turner, S., Attenburrow, G. , (2011) *"Predicting Leather Handle Like An Expert By Artificial Neural Networks"*, Applied Artificial Intelligence*; 25; 2; p180-p192*

Warner, B., Misra, M. (1996), "Understanding Neural Networks as Statistical Tools", American Statistician; Nov., 1996, Vol. 50 Issue 4, p284-293, 10p.

Williams, M.D., Ray, C.T., Griffith, J., De l'Aune, W., (2011) The Use of a Tactile-Vision Sensory Substitution System as an Augmentative Tool for Individuals with Visual Impairments, Journal of Visual Impairment & Blindness, January, pages 45-50.

Wilson, A. (2008), 'Word on the Wire', Vision Systems Design, 13, 8, pp. 55-62, Business Source Complete, EBSCOhost, accessed 27 January 2012.

Wong, F., Zelek, J.S., (2006) "Tactile and Inertial Patterns from a Long White Cane", *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics,*

Wu, J., Yang, Z., Wu, J., Liu, A.(2007), "Virtual line group based video vehicle detection algorithm utilizing both luminance and chrominance", *2nd IEEE Conference on Industrial Electronics and Applications (ICIEA),* May 2007, pp. 2854–2858

Yap, J.S. , Lim. C.P., Mohamad-Saleh, J., (2010), "An enhanced generalized adaptive resonance theory neural network and its application to medical pattern classification", *Journal of Intelligent & Fuzzy Systems*; 2010, Vol. 21 Issue 1/2, p65-78

Yap, W. K., Ho. T, Karri, V (2012), "Exhaust emissions control and engine parameters optimization using artificial neural network virtual sensors for a hydrogen-powered vehicle", *International Journal of Hydrogen Energy*; May2012, Vol. 37 Issue 10, p8704-8715

Zelek, J.S., Bromley S., Asmar, D. and Thompson, D (2003) "A haptic glove for tactilevision sensory substitution for wayfinding"; *Journal of Visual Impairment and Blindness, special issue on technology*, 97(10):621–632, October

Zelek, J. (2005) "Seeing by Touch (Haptics) for Wayfinding" Available from : http://stargate.uwaterloo.ca/~jzelek/papers/1281049.fdf, accessed January 2007

Zickler, T.; Mallick, S.P.; Kriegman, D.J; Belhumeur, P.N.(2008) Color subspaces as photometric invariants INTERNATIONAL JOURNAL OF COMPUTER VISION; AUG,; 79; 1; p13-p30

Zoltan, H. (2011), "Hardware and Software Design of a Prototype Electronic Travel Aid for Visually Impaired", Journal of Electrical & Electronics Engineering; Oct2011, Vol. 4 Issue 2, p41-46

Appendices

Appendix A: Poster presentations –snap-shots of research concepts

Two posters were presented of the research and these reveal progress in the research at each time. Figure A.1 shows a poster presented at an Open University "Research Student Event" in 2006 for which the poster won a runner-up prize. The poster gives no indication of the specific area of "difficult urban environments" but depicts the interest of the researcher into how information about the environment can be acquired and used to stimulate the brain, and possibly even the visual cortex of a blind or partially-sighted individual. The breakdown of vision substitution into a number of subsystems is shown in the central horizontal flow-chart and questions are asked about the best way of feeding back useful information through to the user.

The second poster, shown in figure A.2 won a prize at the Open University Poster Competition in 2009 and went on to represent the University at the vitae Midlands hub regional poster competition in the University of Nottingham on 7 July 2009. Several judges gave it a high score but it did not win a prize. It is interesting to note the progression in thinking from the previous poster. The intention to use a wireless camera at the end of a long cane to capture images of the surface is depicted as is the move to the more definite problem of navigation of Shared Space.

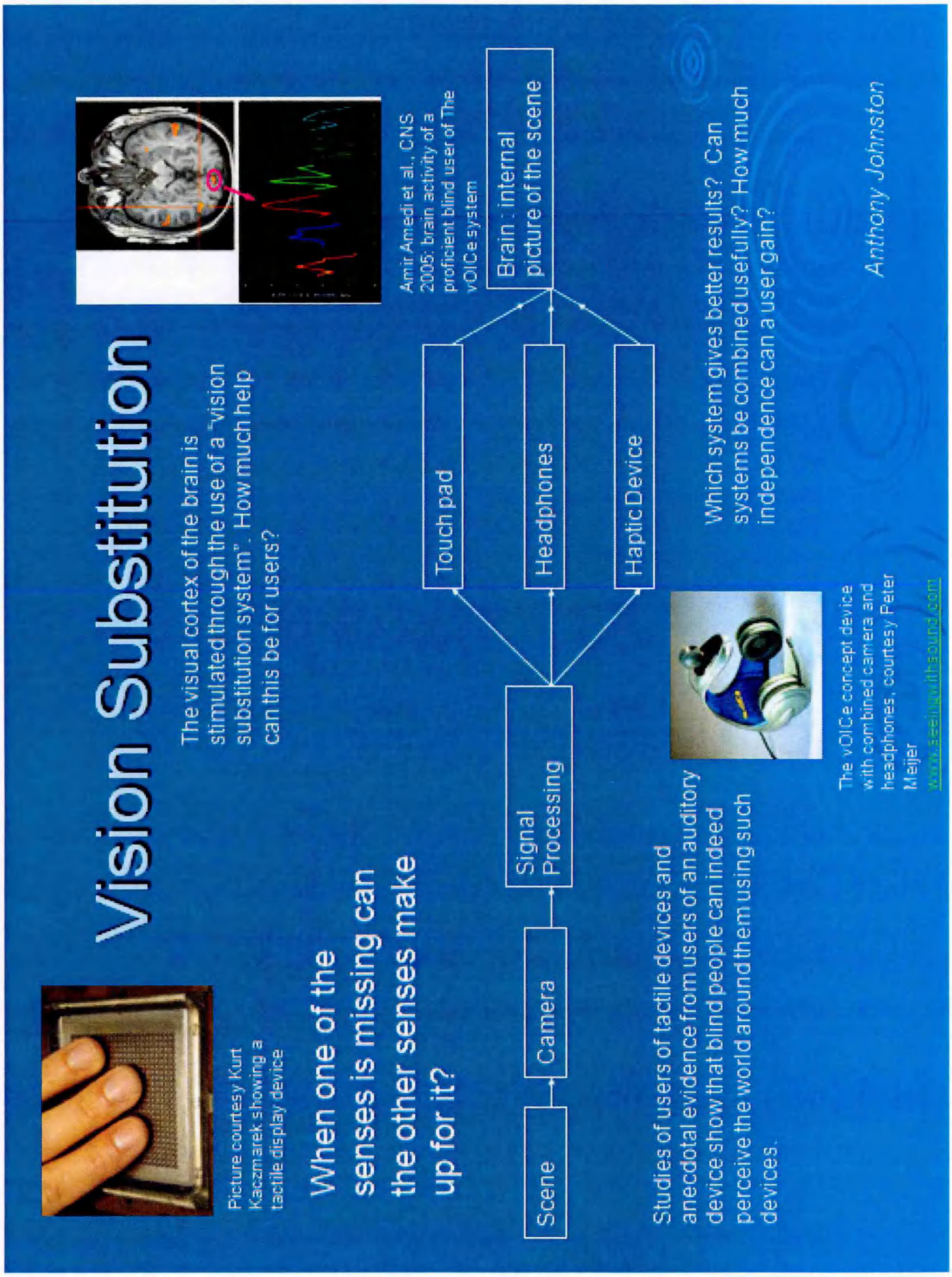


Figure A.1 Poster of Research Concept presented at Open University Poster Competition 2006

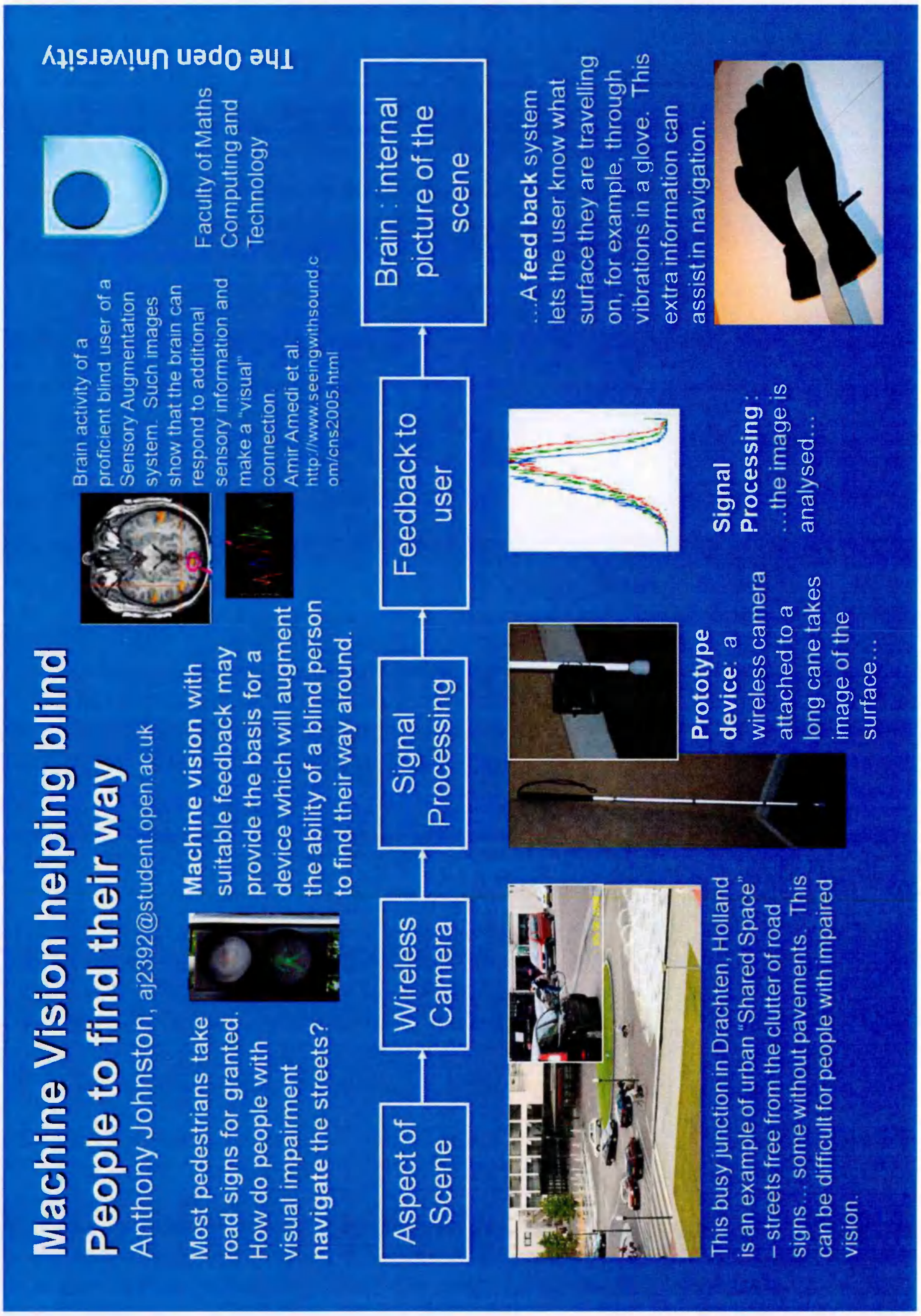


Figure A.2 Poster of Research Concept presented at Open University Poster Competition 2009

Appendix B: Questionnaire

A) Questions about you and your level of vision

What age are you?

Under 16

17 – 25

26 – 34

35 – 44

45 – 54

55 – 64

65 – 74

Over 75

Are you registered…

Blind?

Partially sighted?

Don't know

Which of the following mobility aids do you use?

Sighted Guide

Symbol cane

Guide Cane

Long cane

Guide Dog

Other

If you use a long cane
How long have you been trained to use a long cane?

Less than 1 year

1 – 3 years

4 – 6 years

7 – 9 years

10 – 19 years

20 – 29 years

30 years or more

If you use a guide dog
How long have you been a guide dog owner?

Less than 1 year

1 – 3 years

4 – 6 years

7 – 9 years

10 – 19 years

20 – 29 years

30 years or more

B) Questions about finding your way around your town or city

How often would you go out with someone acting as a sighted guide?

More than once a day

Almost every day

Once or twice a week

Once or twice a month

Seldom

Never

How often do you go out independently using your long cane?

More than once a day

Almost every day

Once or twice a week

Once or twice a month

Seldom

Never

How often do you go out independently using your guide dog?

More than once a day

Almost every day

Once or twice a week

Once or twice a month

Seldom

Never

C) When you are travelling independently

Which of the following do you use to remain orientated and know where you are going? (Could be more than one)

Change in gradient

Change of surface

General listening skills (traffic, voices etc)

Specific noises

Specific smells

Specific clues or landmarks

Road crossings

Information provided by the long cane

Knowledge or familiarisation of route

Ask directions from general public

Other

D) Questions about unknown or unfamiliar areas

How often would you travel independently in an unfamiliar area?

Almost everyday

Once or twice a week

Once or twice a month

Once a twice a year

Never

Other

If Never, do you ever travel in unknown or unfamiliar areas when there is someone with you?

Yes

No

If you are travelling independently in an unknown or unfamiliar area:

Which of the following do you use to remain orientated and know where you are going? (Could be more than one)

Change in gradient

Change of surface

General listening skills (traffic, voices etc)

Specific noises

Specific smells

Specific clues or landmarks

Road crossings

Information provided by the long cane

Partial knowledge of route

Ask directions from general public

Other

E) Questions about Shared Surface Streets, Pedestrian areas &Home Zones

Do you have experience of travelling independently in any of the following areas?

Shared surface streets

Pedestrian areas

Home Zone

If so, where is the place or places? ….

How does travelling independently in such areas make you feel?

What are the main differences when travelling independently in these areas compared to other areas?

How do you remain orientated and know where you are going?

F) Comments

Is there anything you want to add about what helps you find your way around?

Appendix C: Neural Network Program

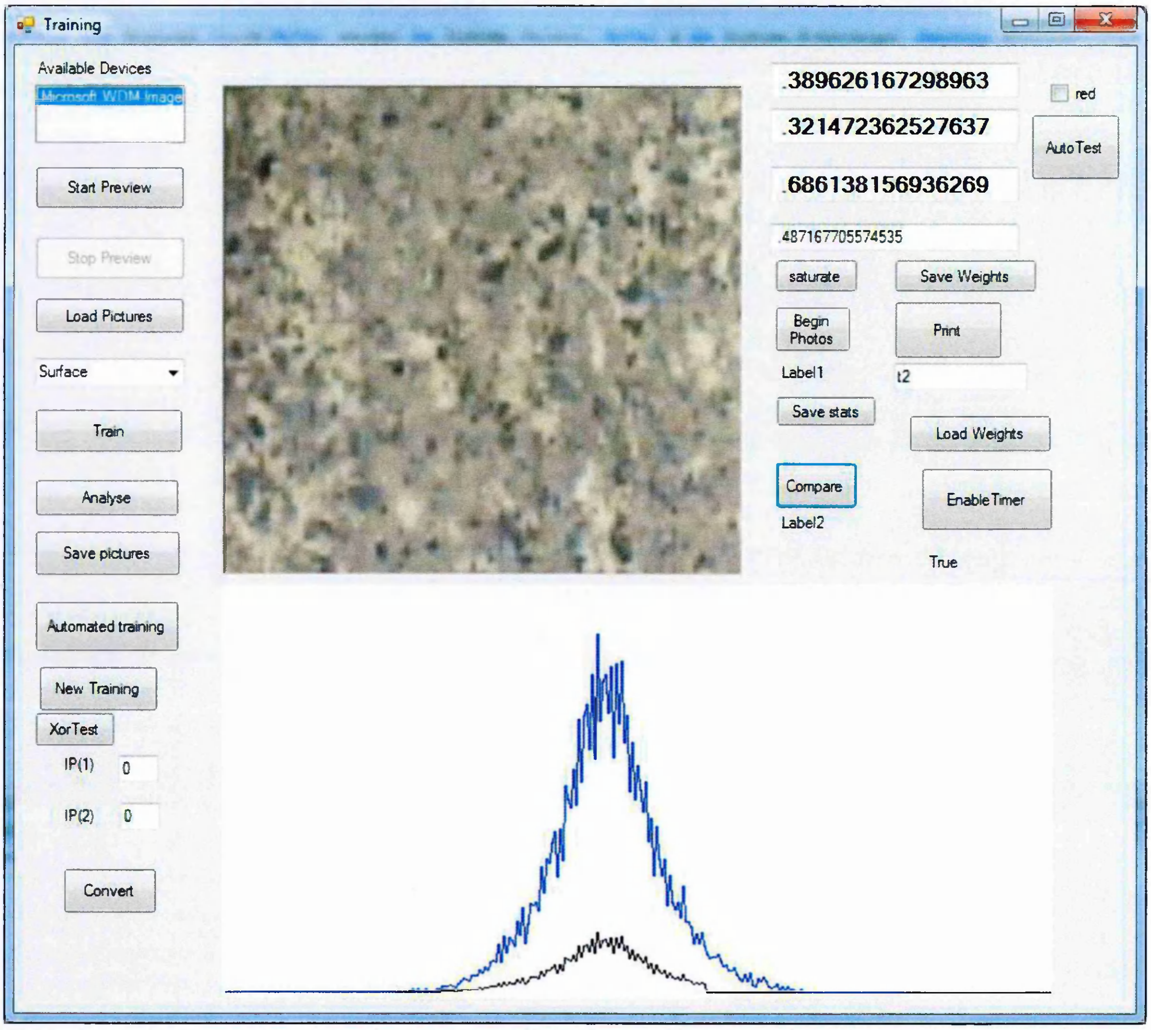


Figure C.1 Neural Network Program Interface

Program:

```
Imports System.Runtime.InteropServices
Imports System.Drawing.Graphics
Imports System.Runtime.Serialization.Formatters.Binary.BinaryFormatter
Imports System.Console

Public Class FrmTraining
    Inherits System.Windows.Forms.Form

    'initial public variables defined
    Public IPmax As Integer = 180 '(must be 254 or smaller)
    Public IPmin As Integer = 20 '(must be 1 or greater)
    Public IP(360) As Double ' though only up to IPmax used
    'Public weight(3, 255) As Double
    Public Sum(3) As Double 'sum of hiddens time weights
```

```
    Public OP(3) As Double 'desired or analysed OP
    Public y(3) As Double ' claculated output of OP unit
    Public RedCount(255) As Integer
    Public Nhid As Integer = 20 ' number of hidden units, max is 50
    Public autoI As Integer

    Public wh(50, 359) As Double ' hidden weights (Nhid, IPmax)
    Public wo(3, 50) As Double 'output weights      (3, Nhid)
    Public hidden(50) As Double 'sum of inputs times weights (Nhid)
    Public Bluecount(255) As Integer
    Public Greencount(255) As Integer
    Public Huecount(360) As Integer
    Public Satcount(360) As Integer
    Public BrightCount(360) As Integer
    Public h(50) As Double 'calculated output of hidden unit (Nhid)
    Public training As Boolean = False
    Public track(3) As Double 'used to see if outputs have changed
    Public photoNo As Integer = 1 'tracks photos taken

    Dim recordStats As Boolean = False ' flag to save data for analysis
    Dim statsCount As Integer = 1
    Dim statsOutput(3, 31) As Double

    'Public Shared Function FromStream(ByVal stream As IO.Stream) As Image
    'Following code is for reading image from camera
    Const WM_CAP As Short = &H400S
    Const WM_CAP_DRIVER_CONNECT As Integer = WM_CAP + 10
    Const WM_CAP_DRIVER_DISCONNECT As Integer = WM_CAP + 11
    Const WM_CAP_EDIT_COPY As Integer = WM_CAP + 30
    Const WM_CAP_SET_PREVIEW As Integer = WM_CAP + 50
    Const WM_CAP_SET_PREVIEWRATE As Integer = WM_CAP + 52
    Const WM_CAP_SET_SCALE As Integer = WM_CAP + 53
    Const WS_CHILD As Integer = &H40000000
    Const WS_VISIBLE As Integer = &H10000000
    Const SWP_NOMOVE As Short = &H2S
    Const SWP_NOSIZE As Short = 1
    Const SWP_NOZORDER As Short = &H4S
    Const HWND_BOTTOM As Short = 1

    Dim bmap As Bitmap
    Dim iDevice As Integer = 0 ' Current device ID
    Dim hHwnd As Integer ' Handle to preview window

    Declare Function SendMessage Lib "user32" Alias "SendMessageA" _
        (ByVal hwnd As Integer, ByVal wMsg As Integer, ByVal wParam As
Integer, _
        <MarshalAs(UnmanagedType.AsAny)> ByVal lParam As Object) As Integer

    Declare Function SetWindowPos Lib "user32" Alias "SetWindowPos" (ByVal
hwnd As Integer, _
        ByVal hWndInsertAfter As Integer, ByVal x As Integer, ByVal y As
Integer, _
        ByVal cx As Integer, ByVal cy As Integer, ByVal wFlags As Integer)
As Integer

    Declare Function DestroyWindow Lib "user32" (ByVal hndw As Integer) As
Boolean

    Declare Function capCreateCaptureWindowA Lib "avicap32.dll" _
        (ByVal lpszWindowName As String, ByVal dwStyle As Integer, _
        ByVal x As Integer, ByVal y As Integer, ByVal nWidth As Integer, _
```

```
        ByVal nHeight As Short, ByVal hWndParent As Integer, _
        ByVal nID As Integer) As Integer

    Declare Function capGetDriverDescriptionA Lib "avicap32.dll" (ByVal
wDriver As Short, _
        ByVal lpszName As String, ByVal cbName As Integer, ByVal lpszVer As
String, _
        ByVal cbVer As Integer) As Boolean

    'declarations for motorbee
    Declare Function InitMotoBee Lib "mtb.dll" () As Boolean
    'mtb.dll is copied in windows/system32 as a library file
    Declare Function Digital_IO Lib "mtb.dll" (ByRef inputs As Integer,
ByVal outputs As Integer) As Boolean
    Declare Function SetMotors Lib "mtb.dll" (ByVal on1 As Integer, ByVal
speed1 As Integer, ByVal on2 As Integer, ByVal speed2 As Integer, ByVal on3
As Integer, ByVal speed3 As Integer, ByVal on4 As Integer, ByVal speed4 As
Integer, ByVal servo As Integer) As Boolean

    Private Sub FrmTraining_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'load main form, initialise buttons
        BtnSave.Enabled = True
        btnStart.Enabled = True
        BtnStop.Enabled = False

        'InitMotoBee()     ' initialise the MotorBee
        'initialise weights
        IP(0) = 1.0
        Dim i, j As Integer
        For i = 0 To 3
            track(i) = 0
            For j = 0 To Nhid
                wo(i, j) = (5 - Rnd(1) * 10) / 10
            Next
        Next
        For i = 0 To Nhid
            For j = 0 To IPmax
                wh(i, j) = (5 - Rnd(1) * 10) / 10
            Next
        Next
        For i = 1 To Nhid
            h(i) = 0.5
        Next
        h(0) = 1

        'initialise camera
        LoadDeviceList()
        If lstDevices.Items.Count > 0 Then
            btnStart.Enabled = True
            lstDevices.SelectedIndex = 0
            btnStart.Enabled = True
        Else
            lstDevices.Items.Add("No Capture Device")
            btnStart.Enabled = False
        End If

        BtnStop.Enabled = False
        BtnSave.Enabled = True
        picCapture.SizeMode = PictureBoxSizeMode.StretchImage
```

```
    End Sub
    Private Sub LoadDeviceList()
        Dim strName As String = Space(100)
        Dim strVer As String = Space(100)
        Dim bReturn As Boolean
        Dim x As Integer = 0
        '
        ' Load name of all avialable devices into the lstDevices
        Do
            '   Get Driver name and version
            bReturn = capGetDriverDescriptionA(x, strName, 100, strVer,
100)
            ' If there was a device add device name to the list
            If bReturn Then lstDevices.Items.Add(strName.Trim)
            x += 1
        Loop Until bReturn = False
    End Sub

    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click
        'starts video feed
        iDevice = lstDevices.SelectedIndex
        OpenPreviewWindow()
    End Sub
    Private Sub OpenPreviewWindow()
        Dim iHeight As Integer = picCapture.Height
        Dim iWidth As Integer = picCapture.Width
        ' Open Preview window in picturebox
        hHwnd = capCreateCaptureWindowA(iDevice, WS_VISIBLE Or WS_CHILD, 0,
0, 640, _
            480, picCapture.Handle.ToInt32, 0)
        ' Connect to device
        If SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0) Then
            'Set the preview scale
            SendMessage(hHwnd, WM_CAP_SET_SCALE, True, 0)
            'Set the preview rate in milliseconds
            SendMessage(hHwnd, WM_CAP_SET_PREVIEWRATE, 10, 0)
            'Start previewing the image from the camera
            SendMessage(hHwnd, WM_CAP_SET_PREVIEW, True, 0)
            ' Resize window to fit in picturebox
            SetWindowPos(hHwnd, HWND_BOTTOM, 0, 0, picCapture.Width,
picCapture.Height, _
                  SWP_NOMOVE Or SWP_NOZORDER)

            BtnSave.Enabled = True
            BtnStop.Enabled = True
            btnStart.Enabled = False
            Label1.Text = "camera found"
        Else
            ' Error connecting to device close window
            DestroyWindow(hHwnd)
            Label1.Text = "no camera"
        End If

    End Sub

    Private Sub BtnStop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnStop.Click
        ClosePreviewWindow()
        BtnSave.Enabled = False
```

```
        btnStart.Enabled = True
        BtnStop.Enabled = False
    End Sub
    Private Sub ClosePreviewWindow()
        ' Disconnect from device
        SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0)
        ' close window
        DestroyWindow(hHwnd)
    End Sub

    'NB Rather than pick up inputs everysecond - for training only
necessary when training or analyse button pressed

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        'every tick of timer the image is processed
        Dim returnValue As Double
        ins()
        outs()
        graph()
        Dim changeflag As Boolean = False
        For k = 0 To 3
            returnValue = Math.Abs(y(k) - track(k))
            If returnValue > 0.5 Then changeflag = True
            track(k) = y(k)
            If recordStats Then statsOutput(k, statsCount) = y(k)
        Next
        If changeflag Then
            Label6.Text = "CHANGE"
            Label3.Text = "CHANGE"
            System.Console.Beep(1000, 5)
        Else
            Label6.Text = "True"
            Label3.Text = "no change"
        End If

        If recordStats Then
            If statsCount < 30 Then statsCount = statsCount + 1
            If statsCount = 30 Then
                recordStats = False
                Timer1.Enabled = False
                Label6.Text = Timer1.Enabled
                Savestats()
                statsCount = 1
            End If
        End If
    End Sub
    Private Sub ins()
        'calculates blue component and sets inputs according to the
threshold
        Dim data As IDataObject
        Dim bmap As Bitmap
        Dim max As Double = 0
        Dim threshold As Double = 300
        ' Copy image to clipboard
        SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
        ' Get image from clipboard and convert it to a bitmap
        data = Clipboard.GetDataObject()

        If (Not training) Then  'And
data.GetDataPresent(GetType(System.Drawing.Bitmap))) Then
```

```
            bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)),
Image)
            'use image from camera if present otherwise use image in
picturebox(ie loaded)
        Else
            bmap = picCapture.Image
        End If
        If btnStart.Enabled Then bmap = picCapture.Image

        Dim i, j As Integer
        Dim DispX As Integer = 1, DispY As Integer = 1
        'Dim hue, sat, bright As Integer
        Dim pixel1 As System.Drawing.Color
        Dim red1, green1, blue1 As Integer
        '  Obtain counts from current frame
        For i = 0 To 255
            'RedCount(i) = 0
            Bluecount(i) = 0
            'Greencount(i) = 0
        Next
        ' For i = 0 To 360
        'Huecount(i) = 0
        'Satcount(i) = 0
        'BrightCount(i) = 0
        'Next
        'get pixels and examine red component - step 2 so every other pixel
examined
        With bmap
            For i = 0 To .Height - 2 Step 2
                For j = 0 To .Width - 2 Step 2
                    '
                    pixel1 = .GetPixel(j, i)
                    'red1 = pixel1.R
                    'green1 = pixel1.G
                    blue1 = pixel1.B
                    'hue = pixel1.GetHue
                    'sat = pixel1.GetSaturation * 359
                    'bright = pixel1.GetBrightness * 359

                    'RedCount(red) = RedCount(red) + 1
                    Bluecount(blue1) = Bluecount(blue1) + 1
                    'Greencount(green) = Greencount(green) + 1
                    'Huecount(hue) += 1
                    'Satcount(sat) += 1
                    'BrightCount(bright) += 1
                Next
            Next
        End With
        For j = IPmin To IPmax ' use IPmax for RGB! use 100 for hue 2/1/12
            'If RedCount(j) > max Then max = RedCount(j)
            If Bluecount(j) > max Then max = Bluecount(j)
            IP(j) = Bluecount(j) / 1800 '1100
            'IP(j) = Huecount(j) / 1200
            If IP(j) > 1 Then IP(j) = 1

          '103680 is the max. colour saturation (needs calibration)
          '1036 gives better differentiation between surfaces but in bright
          'light may cause overload!
        Next
        'For j = 101 To 180 ' delete these lines or use IPmax for RGB with
a different colour
```

```
        'IP(j) = Satcount(j - 90) / 900
        'If IP(j) > 1 Then IP(j) = 1
        'Next
        'MsgBox(max)
        'End If

    End Sub
    Private Sub BtTrain_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtTrain.Click
        ins()
        Dim iteration, i, j, k As Integer
        Dim delta(3) As Double
        Dim Hdelta(Nhid) As Double
        Dim SumD As Double = 0
        Dim whNew(Nhid, IPmax)
        Dim woNew(3, Nhid)

        'set outputs according to ComboBox (CBoxSurface)
        For i = 0 To 3
            OP(i) = 0
        Next
        If CBoxSurface.Text = "Concrete" Then
            OP(0) = 1
        ElseIf CBoxSurface.Text = "Tarmac" Then
            OP(1) = 1
            OP(0) = 0
        Else : OP(2) = 1
        End If

        For iteration = 0 To 1
            'set outputs to 0
            For k = 0 To 3
                Sum(k) = 0
            Next
            For j = 1 To Nhid
                hidden(j) = 0
            Next
            For j = 1 To Nhid
                hidden(j) = hidden(j) + wh(j, 0) * IP(0)
                For i = IPmin To IPmax 'not worth changing to incorporate
sat as well as hue
                    hidden(j) = hidden(j) + wh(j, i) * IP(i)
                Next
                h(j) = 1.0 / (1.0 + 1 * Math.Exp(-hidden(j)))
            Next
            'sum each output using current weights
            'each weight multiplied by the hidden
            For k = 0 To 3
                For j = 0 To Nhid
                    Sum(k) = Sum(k) + wo(k, j) * h(j)
                Next
                y(k) = 1.0 / (1.0 + 1 * Math.Exp(-Sum(k)))
                'calculate delta for each block
                delta(k) = y(k) * (1 - y(k)) * (OP(k) - y(k))
            Next
            'adjust weights
            ' 0.2 is arbitary and perhaps too large?
            For k = 0 To 3
                For j = 0 To Nhid
                    wo(k, j) = wo(k, j) + 0.01 * delta(k) * h(j)
                Next
```

```
        Next
        For j = 1 To Nhid
            hidden(j) = 0
        Next
        'sum each hidden using current weights
        'each weight multiplied by the input
        For j = 1 To Nhid
            hidden(j) = hidden(j) + wh(j, 0) * IP(0)
            For i = IPmin To IPmax
                hidden(j) = hidden(j) + wh(j, i) * IP(i)
            Next
            h(j) = 1.0 / (1.0 + 1 * Math.Exp(-hidden(j)))
            SumD = 0
            For k = 0 To 3
                SumD = SumD + wo(k, j) * delta(k)
            Next
            Hdelta(j) = h(j) * (1 - h(j)) * SumD
        Next
        'adjust weights
        For j = 1 To Nhid
            whNew(j, 0) = wh(j, 0) + 0.05 * Hdelta(j) * IP(0)
            For i = IPmin To IPmax
                whNew(j, i) = wh(j, i) + 0.05 * Hdelta(j) * IP(i)
            Next
        Next
        'Now simultaneously update weights
        For j = 1 To Nhid
            wh(j, 0) = whNew(j, 0)
            For i = IPmin To IPmax
                wh(j, i) = whNew(j, i)
            Next
        Next
    Next
    outs()
End Sub

Private Sub BtAnalyse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtAnalyse.Click
    Dim returnValue = 0
    ins()
    outs()
    For k = 0 To 3
        returnValue = Math.Abs(y(k) - track(k))
        If returnValue > 0.5 Then
            Label6.Text = "CHANGE"
            Label3.Text = "CHANGE"
            track(k) = y(k)
            'System.Console.Beep(1000, 500)
            k = 3
        Else
            Label6.Text = "True"
            Label3.Text = "no change"
            track(k) = y(k)
        End If

    Next
End Sub
Sub outs() ' MIGHT NEED CHANGED 9/6/09
    Dim i, j, k As Integer
    For k = 0 To 3
        'set outputs to 0
```

```
            Sum(k) = 0
        Next
        h(0) = 1
        For j = 1 To Nhid
            'set hidden to 0
            hidden(j) = 0
        Next
        'sun each hidden using current weights
        For j = 1 To Nhid
            hidden(j) = hidden(j) + wh(j, 0) * IP(0)
            For i = IPmin To IPmax 'needs ajusted to incorporate hue as
well as sat 2/1/12 **
                hidden(j) = hidden(j) + wh(j, i) * IP(i)
            Next
            h(j) = 1 / (1.0 + 1 * Math.Exp(-hidden(j)))
        Next

        'sum each output using current weights
        For k = 0 To 3
            For j = 0 To Nhid
                Sum(k) = Sum(k) + wo(k, j) * h(j)
            Next
            y(k) = 1.0 / (1.0 + 1 * Math.Exp(-Sum(k)))
            'calculate delta for each block
        Next
        'MsgBox(Sum(0))
        'MsgBox(y(0))
        Txt_OP1.Text = Str(y(0))
        Txt_Op2.Text = Str(y(1))
        Txt_Op3.Text = Str(y(2))
        Txt_Op4.Text = Str(y(3))
        'Label1.Text = Str(OP(2)) + " " + Str(Sum(2)) 'CBoxSurface.Text
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim pixtotal As Integer = 0
        Dim data As IDataObject
        Dim bmap As Bitmap
        ' Copy image to clipboard
        SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
        ' Get image from clipboard and convert it to a bitmap
        data = Clipboard.GetDataObject()
        If data.GetDataPresent(GetType(System.Drawing.Bitmap)) Then
            bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)),
Image)
        Else
            bmap = picCapture.Image

        End If

        Dim i, j As Integer
        Dim DispX As Integer = 1, DispY As Integer = 1
        Dim pixel1 As System.Drawing.Color
        '  Obtain counts from current frame
        With bmap
            For i = 0 To .Height - 2 Step 2
                For j = 0 To .Width - 2 Step 2
                    '
                    pixel1 = .GetPixel(j, i)
                    pixtotal = pixtotal + 1
```

```
            Next
        Next
        MsgBox(.Height & " x " & .Width)

    End With

    Label1.Text = Str(pixtotal)
End Sub

Private Sub BtnLoad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnLoad.Click
    ' Displays an OpenFileDialog so the user can select a photograph
    Dim openFileDialog1 As New OpenFileDialog()
    Dim returnValue As Image

    openFileDialog1.Filter = "Data|*.*"
    openFileDialog1.Title = "Select the picture"
    ' Show the Dialog.
    If openFileDialog1.ShowDialog() = DialogResult.OK Then
        'Read picture in file using Deserialization
        Dim readFile As IO.FileStream
        readFile = IO.File.OpenRead((openFileDialog1.FileName()))

        returnValue = Bitmap.FromFile(openFileDialog1.FileName())

        picCapture.Image = returnValue
        ' SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0)
        'RedCount = CType(formatter.Deserialize(readFile), Array)
        readFile.Close()
        '*****THE FOLLOWING WAS FOR PLOTTING GRAPH WHEN PICTURE WAS
UPLOADED - NO LONGER USED ******
        'Dim max As Integer = 0
        'Dim k As Double = 0
        'For i = 0 To 255
        'If RedCount(i) > max Then max = RedCount(i)
        'Next
        'Txt_OP1.Text = Str(max)
        'max = 200

        'k = 300 / max
        'Txt_Op2.Text = Str(k)
        'For j = 0 To 255
        'RedCount(j) = RedCount(j) '* k
        'Next
        'Dim g As Graphics = PictureBox2.CreateGraphics
        'Dim myPenr As New Pen(Color.Red)
        'Dim myPeng As New Pen(Color.Green)
        'Dim myPenb As New Pen(Color.Blue)
        'myPenr.Width = 1
        'g.Clear(Color.White)
        ' For i = 0 To 254
        'g.DrawLine(myPenr, i, RedCount(i), i + 1, RedCount(i + 1))
        'Next i
    End If
End Sub

Private Sub BtnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnSave.Click

    Dim saveFileDialog1 As New SaveFileDialog()
    Dim bmap As Bitmap
```

```
        Dim data As IDataObject
        ' Copy image to clipboard
        SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
        ' Get image from clipboard and convert it to a bitmap
        data = Clipboard.GetDataObject()

        If data.GetDataPresent(GetType(System.Drawing.Bitmap)) Then
            bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)),
Image)
        Else
            bmap = CType(picCapture.Image, Image)
        End If

        saveFileDialog1.Filter = "Data|*.*"
        saveFileDialog1.Title = "Save the picture (and graph)"

        ' Show the Dialog.
        If saveFileDialog1.ShowDialog() = DialogResult.OK Then
            'save picture using Serialization object
            Dim saveFile As IO.FileStream
            saveFile = IO.File.Create(saveFileDialog1.FileName())
            Dim formatter As
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            bmap.Save(saveFile, Drawing.Imaging.ImageFormat.Bmp)
            'formatter.Serialize(saveFile, bmap)
            'formatter.Serialize(saveFile, RedCount)
            saveFile.Close()
        End If
    End Sub


    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        graph()
        Label3.Text = ""
    End Sub

    Private Sub graph()
        'button COMPARE
        'Plots graph from current image on camera
        ins()

        Dim max As Integer = 0
        Dim i As Integer
        Dim g As Graphics = PictureBox2.CreateGraphics
        'PictureBox2.Height = 300
        'PictureBox2.Width = 256

        Dim myPenr As New Pen(Color.Red)
        Dim myPeng As New Pen(Color.Green)
        Dim myPenb As New Pen(Color.Blue)
        Dim myPenbck As New Pen(Color.Black)
        myPenr.Width = 1
        g.Clear(Color.White)

        For i = 1 To 359 'IPmax - 1
            g.DrawLine(myPenbck, i * 2, 300 - Convert.ToInt32(IP(i) * 150),
i * 2 + 1, 300 - Convert.ToInt32(IP(i + 1) * 150))
```

```
            ' g.DrawLine(myPenb, i * 2, 300 -
Convert.ToInt32(BrightCount(i) / 2), i * 2 + 1, 300 -
Convert.ToInt32(BrightCount(i + 1) / 2))
            'g.DrawLine(myPenr, i * 2, 300 - Convert.ToInt32(Satcount(i) /
4), i * 2 + 1, 300 - Convert.ToInt32(Satcount(i + 1) / 4))
            'g.DrawLine(myPeng, i * 2, 300 - Convert.ToInt32(Huecount(i) /
20), i * 2 + 1, 300 - Convert.ToInt32(Huecount(i + 1) / 20))
            If i < 255 Then g.DrawLine(myPenb, i * 2, 300 -
Convert.ToInt32(Bluecount(i) / 2), i * 2 + 1, 300 -
Convert.ToInt32(Bluecount(i + 1) / 2))
        Next i
    End Sub


    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        'Phototrain("c:\users\anthony\new folder\photo\photos\concrete1")
        'load up pictures and run through training automatically
        Dim openFileDialog1 As New OpenFileDialog()
        Dim returnValue As Image
        Dim l As Integer
        Dim delta(3) As Double
        Dim Hdelta(Nhid) As Double
        Dim SumD As Double = 0
        Dim whNew(Nhid, IPmax)
        Dim woNew(3, Nhid)
        Dim whA = 0.1
        Dim woA = 0.1
        'Dim readFile As IO.FileStream
        'Dim formatter As
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

        training = True
        For Me.autoI = 1 To 200
            If autoI Mod 10 = 0 Then Out.WriteLine(autoI)
            'concrete
            'Read picture in file using Deserialization
            Dim readFile As IO.FileStream
            readFile = IO.File.OpenRead(("c:\users\anthony\new
folder\photo\photos\concrete1"))
            Dim formatter As
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            returnValue = CType(formatter.Deserialize(readFile), Image)
            'stream = openFileDialog1.OpenFile()
            picCapture.Image = returnValue
            'RedCount = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
            'set outputs for concrete
            OP(0) = 1
            OP(1) = 0
            OP(2) = 0
            OP(3) = 0

            trainheart(3)
            'tarmac
            'Read picture in file using Deserialization
            readFile = IO.File.OpenRead(("c:\users\anthony\new
folder\photo\photos\tarmac1"))
```

```
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            returnValue = CType(formatter.Deserialize(readFile), Image)
            'stream = openFileDialog1.OpenFile()
            picCapture.Image = returnValue
            'RedCount = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
            'set outputs for tarmac
            OP(0) = 0
            OP(1) = 1
            OP(2) = 0
            OP(3) = 0
            trainheart(3)

            'pavingstone
            'Read picture in file using Deserialization
            readFile = IO.File.OpenRead(("c:\users\anthony\new
folder\photo\photos\paving1"))
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            returnValue = CType(formatter.Deserialize(readFile), Image)
            'stream = openFileDialog1.OpenFile()
            picCapture.Image = returnValue
            'RedCount = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
            'set outputs for paving
            OP(0) = 0
            OP(1) = 0
            OP(2) = 1
            OP(3) = 0

            trainheart(3)
            'concrete2
            'Read picture in file using Deserialization
            readFile = IO.File.OpenRead(("c:\users\anthony\new
folder\photo\photos\concrete2"))
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            returnValue = CType(formatter.Deserialize(readFile), Image)
            'stream = openFileDialog1.OpenFile()
            picCapture.Image = returnValue
            'RedCount = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
            'set outputs for concrete
            OP(0) = 1
            OP(1) = 0
            OP(2) = 0
            OP(3) = 0

            trainheart(3)
            'tarmac2
            'Read picture in file using Deserialization
            readFile = IO.File.OpenRead(("c:\users\anthony\new
folder\photo\photos\tarmac2"))
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            returnValue = CType(formatter.Deserialize(readFile), Image)
            'stream = openFileDialog1.OpenFile()
            picCapture.Image = returnValue
            'RedCount = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
```

```
            'set outputs for tarmac
            OP(0) = 0
            OP(1) = 1
            OP(2) = 0
            OP(3) = 0
            trainheart(3)
            'pavingstone2
            'Read picture in file using Deserialization
            readFile = IO.File.OpenRead(("c:\users\anthony\new
folder\photo\photos\paving2"))
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            returnValue = CType(formatter.Deserialize(readFile), Image)
            'stream = openFileDialog1.OpenFile()
            picCapture.Image = returnValue
            'RedCount = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
            'set outputs for paving
            OP(0) = 0
            OP(1) = 0
            OP(2) = 1
            OP(3) = 0

            trainheart(3)
            'concrete3
            'Read picture in file using Deserialization
            readFile = IO.File.OpenRead(("c:\users\anthony\new
folder\photo\photos\concrete3"))
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            returnValue = CType(formatter.Deserialize(readFile), Image)
            'stream = openFileDialog1.OpenFile()
            picCapture.Image = returnValue
            'RedCount = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
            'set outputs for concrete
            OP(0) = 1
            OP(1) = 0
            OP(2) = 0
            OP(3) = 0

            trainheart(3)
            'tarmac3
            'Read picture in file using Deserialization
            readFile = IO.File.OpenRead(("c:\users\anthony\new
folder\photo\photos\tarmac3"))
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            returnValue = CType(formatter.Deserialize(readFile), Image)
            'stream = openFileDialog1.OpenFile()
            picCapture.Image = returnValue
            'RedCount = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
            'set outputs for tarmac
            OP(0) = 0
            OP(1) = 1
            OP(2) = 0
            OP(3) = 0
            trainheart(3)

            'pavingstone3
```

```
            'Read picture in file using Deserialization
            readFile = IO.File.OpenRead(("c:\users\anthony\new
folder\photo\photos\paving3"))
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            returnValue = CType(formatter.Deserialize(readFile), Image)
            'stream = openFileDialog1.OpenFile()
            picCapture.Image = returnValue
            'RedCount = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
            'set outputs for paving
            OP(0) = 0
            OP(1) = 0
            OP(2) = 1
            OP(3) = 0
            trainheart(3)
        Next
        training = False
    End Sub

    Private Sub trainheart(ByVal no_op) 'no_op is no of outputs being
examined
        Dim i, j, k As Integer
        Dim delta(3) As Double
        Dim Hdelta(Nhid) As Double
        Dim SumD As Double = 0
        Dim whNew(Nhid, IPmax) As Double
        Dim woNew(3, Nhid) As Double
        Dim whA As Double = 0.1 '/ (autoI * autoI)
        Dim woA As Double = 0.1 '/ (autoI * autoI)
        h(0) = 1
        ins() 'suppress for XOR test
        For iteration = 0 To 0
            'set outputs to 0
            For k = 0 To no_op
                Sum(k) = 0
            Next
            For j = 1 To Nhid
                hidden(j) = 0
            Next
            For j = 1 To Nhid
                hidden(j) = hidden(j) + wh(j, 0) * IP(0)
                For i = IPmin To IPmax
                    hidden(j) = hidden(j) + wh(j, i) * IP(i)

                Next
                h(j) = 1 / (1 + 1 * Math.Exp(-hidden(j)))
                'MsgBox(h(j))
            Next
            'sum each output using current weights
            'each weight multiplied by the hidden
            For k = 0 To no_op
                For j = 0 To Nhid
                    Sum(k) = Sum(k) + wo(k, j) * h(j)
                Next
                y(k) = 1.0 / (1.0 + 1 * Math.Exp(-Sum(k)))
                'calculate delta for each block
                delta(k) = y(k) * (1 - y(k)) * (OP(k) - y(k))

            Next
            'adjust weights
```

```
            ' 0.2 is arbitary and perhaps too large?
            For k = 0 To no_op
                For j = 0 To Nhid
                    woNew(k, j) = wo(k, j) + woA * delta(k) * h(j)
                Next
            Next
            'Out.WriteLine(wo(0, 1))
            For j = 1 To Nhid
                hidden(j) = 0
            Next
            'sum each hidden using current weights
            'each weight multiplied by the input
            For j = 1 To Nhid
                hidden(j) = hidden(j) + wh(j, 0) * IP(0)
                For i = IPmin To IPmax
                    hidden(j) = hidden(j) + wh(j, i) * IP(i)
                Next
                h(j) = 1.0 / (1.0 + 1 * Math.Exp(-hidden(j)))
                SumD = 0
                For k = 0 To no_op
                    SumD = SumD + wo(k, j) * delta(k)
                Next
                Hdelta(j) = h(j) * (1 - h(j)) * SumD
            Next
            'adjust weights
            For j = 0 To Nhid
                whNew(j, 0) = wh(j, 0) + whA * Hdelta(j) * IP(0)
                For i = IPmin To IPmax
                    'If (whA * Hdelta(j) * IP(i)) < 0 Then MsgBox(whA *
Hdelta(j) * IP(i))
                    whNew(j, i) = wh(j, i) + whA * Hdelta(j) * IP(i)
                Next
            Next
            'Now simultaneously update weights
            For j = 0 To Nhid
                wh(j, 0) = whNew(j, 0)
                For i = IPmin To IPmax
                    wh(j, i) = whNew(j, i)
                Next
            Next
            For k = 0 To no_op
                For j = 0 To Nhid
                    wo(k, j) = woNew(k, j)
                Next
            Next
        Next
        'outs()
        'Label1.Text = Str(k)
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        'XOR test set up for two inputs, two layers, one output
        Dim l As Integer
        Dim delta(3) As Double
        Dim Hdelta(Nhid) As Double
        Dim SumD As Double = 0
        Dim whNew(Nhid, IPmax)
        Dim woNew(3, Nhid)
        Dim whA = 0.1
        Dim woA = 0.1
```

```
        IPmax = 2
        IPmin = 1
        Nhid = 2
        IP(0) = 1
        OP(1) = 0
        OP(2) = 0
        OP(3) = 0
        For l = 1 To 6000
            Out.WriteLine(l)
            'If Math.DivRem(Me.autoI, 100, l) = 1 Then Label3.Text =
Me.autoI
            'set inputs = 0,0; output = 0
            IP(1) = 0
            IP(2) = 0
            OP(0) = 0
            trainheart(1)
            'set inputs = 1,0; output = 1
            IP(1) = 1
            IP(2) = 0
            OP(0) = 1
            trainheart(1)
            'set inputs = 0,1; output = 1
            IP(1) = 0
            IP(2) = 1
            OP(0) = 1
            trainheart(1)
            'set inputs = 1,1; output = 1
            IP(1) = 1
            IP(2) = 1
            OP(0) = 0
            trainheart(1)
        Next
    End Sub

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TextBox1.TextChanged, TextBox2.TextChanged
        IP(0) = 1
        IP(1) = Val(TextBox1.Text)
        IP(2) = Val(TextBox2.Text)
        outs()
    End Sub

    Private Sub BtnSaveWeights_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnSaveWeights.Click
        Dim saveFileDialog1 As New SaveFileDialog()
        saveFileDialog1.Filter = "Data|*.*"
        saveFileDialog1.Title = "Select the weights table"
        ' Show the Dialog.
        If saveFileDialog1.ShowDialog() = DialogResult.OK Then
            'save weights using Serialization object
            Dim saveFile As IO.FileStream
            saveFile = IO.File.Create(saveFileDialog1.FileName())
            Dim formatter As
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            formatter.Serialize(saveFile, IPmin)
            formatter.Serialize(saveFile, IPmax)
            formatter.Serialize(saveFile, Nhid)
            formatter.Serialize(saveFile, wh)
```

```
            formatter.Serialize(saveFile, wo)
            saveFile.Close()
        End If
    End Sub

    Private Sub BtnLoadWeights_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnLoadWeights.Click
        ' Displays an OpenFileDialog so the user can select a weights table
        Dim openFileDialog1 As New OpenFileDialog()
        openFileDialog1.Filter = "Data|*.*"
        openFileDialog1.Title = "Select the weights table"

        ' Show the Dialog.
        If openFileDialog1.ShowDialog() = DialogResult.OK Then
            'Read weights in file using Deserialization
            Dim readFile As IO.FileStream
            readFile = IO.File.OpenRead((openFileDialog1.FileName()))
            Dim formatter As
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter
            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            IPmin = CType(formatter.Deserialize(readFile), Integer)
            IPmax = CType(formatter.Deserialize(readFile), Integer)
            Nhid = CType(formatter.Deserialize(readFile), Integer)
            wh = CType(formatter.Deserialize(readFile), Array)
            wo = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
        End If
    End Sub

    Private Sub BtnTimer_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnTimer.Click
        Timer1.Enabled = Not Timer1.Enabled
        'Timer2.Enabled = True
        Label6.Text = Timer1.Enabled
    End Sub

    Private Sub BtnPrint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnPrint.Click
        'print each picture C1 - C10, T1-T10, P1- P10
        'Read picture in file using Deserialization

        Dim prn As New Printing.PrintDocument
        Dim margins As New System.Drawing.Printing.Margins(100, 100, 100,
100)

        prn.PrinterSettings.PrinterName = "Epson stylus CX5400"

        AddHandler prn.PrintPage, AddressOf Me.PrintPageHandler
        prn.DefaultPageSettings.Margins = margins
        prn.Print()
        RemoveHandler prn.PrintPage, AddressOf Me.PrintPageHandler
    End Sub
    Private Sub PrintPageHandler(ByVal sender As Object, ByVal args As
Printing.PrintPageEventArgs)

        Dim openFileDialog1 As New OpenFileDialog()
        Dim returnValue As Image
        Dim Fname As String = ""

        openFileDialog1.Filter = "Data|*.*"
```

```
        openFileDialog1.Title = "Select the picture"
        ' Show the Dialog.
        If openFileDialog1.ShowDialog() = DialogResult.OK Then
            'Read picture in file using Deserialization
            Dim readFile As IO.FileStream
            Fname = openFileDialog1.FileName()
            readFile = IO.File.OpenRead(Fname)

            returnValue = Bitmap.FromFile(Fname)

            picCapture.Image = returnValue
            ' SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0)
            'RedCount = CType(formatter.Deserialize(readFile), Array)
            readFile.Close()
        End If

        graph()
        Dim myPenr As New Pen(Color.Red)
        Dim myPeng As New Pen(Color.Green)
        Dim myPenb As New Pen(Color.Blue)
        Dim myPenbck As New Pen(Color.Black)
        Dim myFont As New Font("Microsoft San Serif", 14)

        For i = IPmin To IPmax - 1
            args.Graphics.DrawLine(myPenbck, i * 2, 200 -
Convert.ToInt32(IP(i) * 150), i * 2 + 1, 200 - Convert.ToInt32(IP(i + 1) *
150))
            args.Graphics.DrawLine(myPenb, i * 2, 300 -
Convert.ToInt32(Bluecount(i) / 4), i * 2 + 1, 300 -
Convert.ToInt32(Bluecount(i + 1) / 4))
            args.Graphics.DrawLine(myPenr, i * 2, 300 -
Convert.ToInt32(RedCount(i) / 4), i * 2 + 1, 300 -
Convert.ToInt32(RedCount(i + 1) / 4))
            args.Graphics.DrawLine(myPeng, i * 2, 300 -
Convert.ToInt32(Greencount(i) / 4), i * 2 + 1, 300 -
Convert.ToInt32(Greencount(i + 1) / 4))

        Next i
        args.Graphics.DrawString(Fname, New Font(myFont,
FontStyle.Regular), Brushes.Black, 50, 50)
    End Sub
    Private Sub AxWindowsMediaPlayer1_Enter(ByVal sender As System.Object,
ByVal e As System.EventArgs)

    End Sub

    Private Sub BtnConvert_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnConvert.Click
        Dim openFileDialog1 As New OpenFileDialog()
        Dim returnValue As Image
        Dim bmap As Bitmap
        Dim FName As String = ""

        openFileDialog1.Filter = "Data|*.*"
        openFileDialog1.Title = "Select the picture"
        ' Show the Dialog.
        If openFileDialog1.ShowDialog() = DialogResult.OK Then
            'Read picture in file using Deserialization
            Dim readFile As IO.FileStream
            FName = openFileDialog1.FileName()
            'readFile = IO.File.OpenRead((openFileDialog1.FileName()))
```

```
            readFile = IO.File.OpenRead(FName)
            Dim formatter As
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

            formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            returnValue = CType(formatter.Deserialize(readFile), Image)
            picCapture.Image = returnValue

            readFile.Close()
        End If
        'save picture as a bit map
        Dim saveFileDialog1 As New SaveFileDialog()
        bmap = CType(picCapture.Image, Image)

        FName = FName & ".bmp"
        Dim saveFile As IO.FileStream

        saveFile = IO.File.Create(FName)
        bmap.Save(saveFile, Drawing.Imaging.ImageFormat.Bmp)

        saveFile.Close()
    End Sub

    Private Sub BtnAutoTrain_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnAutoTrain.Click
        'load up pictures and run through training automatically
        ' Phototrain("c:\users\anthony\new folder\photo\collection\")
        Phototrain("c:\users\anthony\new folder\photo\auto")
    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
        'on each press of external button a photo is saved in the auto
folder
        'once collection is complete the training is started

        Dim inputs As Integer
        Dim outputs As Integer
        Digital_IO(inputs, outputs)
        If ((inputs And 1) = 0) Then
            DIP1.Checked() = True
            PhotoSave() ' save current video as a photo
            If photoNo Mod 20 = 0 Then
                System.Console.Beep(500, 100)
            Else
                System.Console.Beep(1000, 100)
            End If
            photoNo = photoNo + 1
        Else
            DIP1.Checked() = False
            'o1 = 0
        End If
        'SetMotors(o1, s1, o2, s2, o3, s3, o4, s4, sv)
        If photoNo > 60 Then
            Label1.Text = "TRAINING      "
            Timer2.Enabled = False
            Phototrain("c:\users\anthony\new folder\photo\Auto\") '
automatically run through training on acquired photos
            Label1.Text = "              "
        End If
```

```
    End Sub

    Private Sub Btn_BeginPhotos_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Btn_BeginPhotos.Click
        photoNo = 1
        Timer2.Enabled = True
        Label1.Text = "PHOTO"
    End Sub

    Sub Phototrain(ByVal dir As String)
        Dim openFileDialog1 As New OpenFileDialog()
        Dim returnValue As Image
        Dim l As Integer
        Dim delta(3) As Double
        Dim Hdelta(Nhid) As Double
        Dim SumD As Double = 0
        Dim whNew(Nhid, IPmax)
        Dim woNew(3, Nhid)
        Dim whA = 0.1
        Dim woA = 0.1
        Dim di As New IO.DirectoryInfo(dir)
        Dim aryFi As IO.FileInfo() = di.GetFiles("*.bmp")
        Dim fi As IO.FileInfo
        Dim readFile As IO.FileStream
        'Dim readFile As IO.FileStream
        'Dim formatter As
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter
        training = True
        For Me.autoI = 1 To 200
            If autoI Mod 10 = 0 Then Out.WriteLine(autoI)
            Dim strFileSize As String = ""
            For Each fi In aryFi
                readFile = IO.File.OpenRead(fi.FullName())
                returnValue = Bitmap.FromFile(fi.FullName())
                picCapture.Image = returnValue
                'RedCount = CType(formatter.Deserialize(readFile), Array)
                readFile.Close()
                'set outputs for concrete
                If fi.Name.Contains("Con") Or fi.Name.Contains("con") Then
                    OP(0) = 1
                    OP(1) = 0
                    OP(2) = 0
                    OP(3) = 0
                ElseIf fi.Name.Contains("Tar") Or fi.Name.Contains("tar")
Then
                    OP(0) = 0
                    OP(1) = 1
                    OP(2) = 0
                    OP(3) = 0
                Else
                    OP(0) = 0
                    OP(1) = 0
                    OP(2) = 1
                    OP(3) = 0
                End If
                trainheart(3)
            Next
        Next
        training = False
    End Sub
```

```
    Sub PhotoSave()
        Dim saveFileDialog1 As New SaveFileDialog()
        Dim bmap As Bitmap
        Dim data As IDataObject
        Dim di As String = "c:\users\anthony\new folder\photo\Auto\"
        Dim FName As String = ""
        ' Copy image to clipboard
        SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
        ' Get image from clipboard and convert it to a bitmap
        data = Clipboard.GetDataObject()
        If data.GetDataPresent(GetType(System.Drawing.Bitmap)) Then
            bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)),
Image)
        Else
            bmap = CType(picCapture.Image, Image)
        End If
        If photoNo < 21 Then
            FName = di & "concrete" & Str(photoNo) & ".bmp"
            'save picture using Serialization object
        ElseIf photoNo < 41 Then
            FName = di & "tarmac" & Str(photoNo - 20) & ".bmp"
        Else
            FName = di & "paving" & Str(photoNo - 40) & ".bmp"
        End If
        Dim saveFile As IO.FileStream
        saveFile = IO.File.Create(FName)
        bmap.Save(saveFile, Drawing.Imaging.ImageFormat.Bmp)
        saveFile.Close()
    End Sub

    Private Sub Btn_Statistic_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Btn_Statistic.Click
        recordStats = True 'sets flag picked up by timer 1
    End Sub

    Private Sub Savestats()
        Dim saveFileDialog1 As New SaveFileDialog()
        Dim FName As String = ""
        saveFileDialog1.Filter = "Data|*.*"
        saveFileDialog1.Title = "Select the file for storing 30 outputs"
        ' Show the Dialog.
        If saveFileDialog1.ShowDialog() = DialogResult.OK Then
            'save data using Serialization object
            FName = saveFileDialog1.FileName() & ".csv"
            Dim saveFile As IO.FileStream
            saveFile = IO.File.Create(FName)
            'Dim formatter As
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter
            'formatter = New
System.Runtime.Serialization.Formatters.Binary.BinaryFormatter()
            Using writer As System.IO.StreamWriter = New
System.IO.StreamWriter(saveFile)
                For k = 0 To 3
                    For m = 1 To 30
                        writer.Write("{0:0.0000} ,", statsOutput(k, m))
                        'formatter.Serialize(saveFile, statsOutput(k, m),)
                    Next
                Next
            End Using
```

```
            saveFile.Close()
        End If
    End Sub

    Private Sub BtnAutoTest_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnAutoTest.Click
        'test the photographs one at a time
        'determine the outputs and save in a file
        Dim openFileDialog1 As New OpenFileDialog()
        Dim returnValue As Image
        Dim l As Integer
        Dim di As New IO.DirectoryInfo("c:\users\anthony\new
folder\photo\2Jan12 blurry")
        Dim aryFi As IO.FileInfo() = di.GetFiles("*.bmp")
        Dim fi As IO.FileInfo
        Dim readFile As IO.FileStream
        Dim statsCount As Integer = 1
        Dim statsOutput(60, 4) As Double
        Dim photoname(60) As String
        Dim total(3) As Double
        Dim average(3) As Double
        Dim SD(3) As Double 'standard deviation

        For Each fi In aryFi 'go through each picture stored in the
directory
            readFile = IO.File.OpenRead(fi.FullName())
            returnValue = Bitmap.FromFile(fi.FullName())
            picCapture.Image = returnValue
            readFile.Close()
            ins()
            outs()
            For k = 0 To 3 ' prepare output matrix to hold calculated
outputs
                statsOutput(statsCount, k) = y(k)
            Next
            If fi.Name.Contains("Con") Or fi.Name.Contains("con") Then
                statsOutput(statsCount, 4) = 1
                'l = MsgBox("Concrete Press Return for next")
            ElseIf fi.Name.Contains("Tar") Or fi.Name.Contains("tar") Then
                statsOutput(statsCount, 4) = 2
                'l = MsgBox("Tarmac Press Return for next")
            Else
                statsOutput(statsCount, 4) = 3
            End If
            photoname(statsCount) = fi.Name
            statsCount = statsCount + 1
        Next
        'Do calculations!
        'concrete:
        For k = 0 To 3
            total(k) = 0
        Next
        For m = 1 To 20
            For k = 0 To 3
                total(k) = total(k) + statsOutput(m, k)
            Next
        Next
        For k = 0 To 3
            average(k) = total(k) / 20
        Next
```

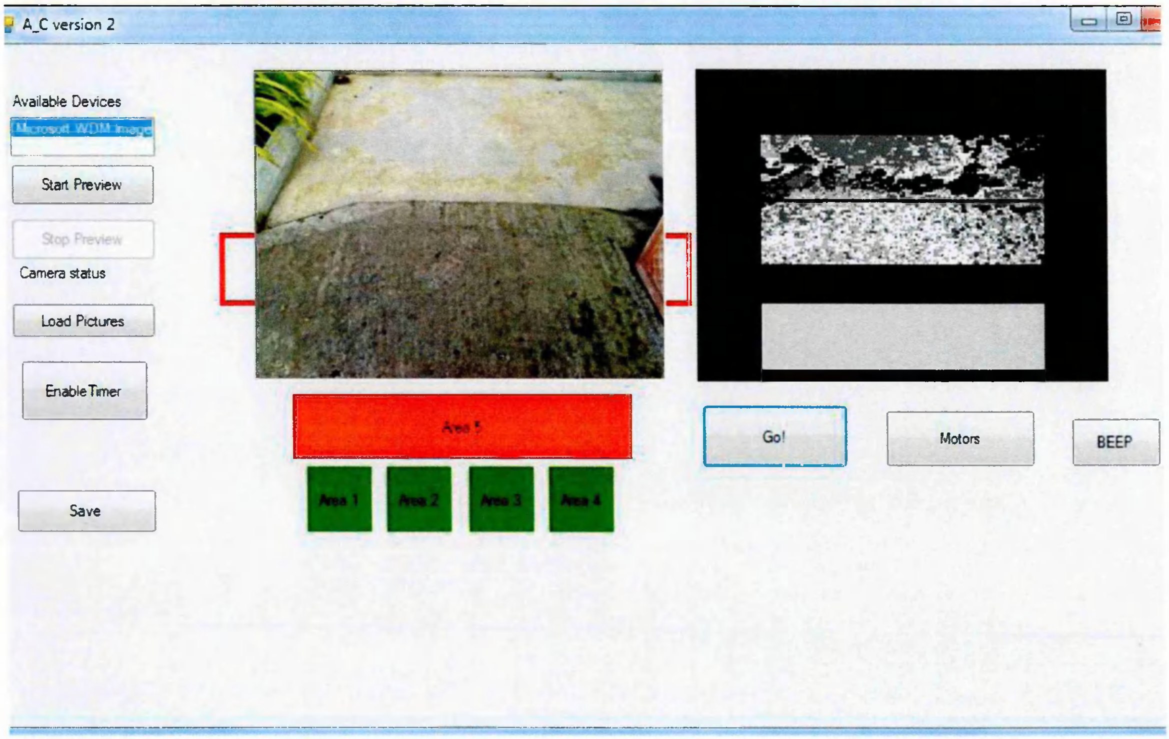
```
        'Now save the output
        Dim saveFileDialog1 As New SaveFileDialog()
        Dim FName As String = ""
        saveFileDialog1.Filter = "Data|*.*"
        saveFileDialog1.Title = "Select the file for storing 30 outputs"

        ' Show the Dialog.
        If saveFileDialog1.ShowDialog() = DialogResult.OK Then
            'save data as a coma seperated variable file
            FName = saveFileDialog1.FileName() & ".csv"
            Dim saveFile As IO.FileStream
            saveFile = IO.File.Create(FName)

            Using writer As System.IO.StreamWriter = New
System.IO.StreamWriter(saveFile)
                For l = 1 To 19
                    For m = 1 To 3
                        For k = 0 To 4
                            writer.Write("{0:0.0000} ,", statsOutput(l + 20
* (m - 1), k))
                        Next
                        writer.Write(photoname(l + 20 * (m - 1)))
                        writer.Write(",")
                    Next
                    writer.WriteLine()
                Next
            End Using
            saveFile.Close()
        End If

    End Sub
End Class
```

Appendix D: Area Comparison Program



D.1 Main Form

```
Imports System.Runtime.InteropServices
Imports System.Drawing.Graphics
Imports System.Console
'investigation of safe area in foreground

Public Class Form1
    'Public Shared Function FromStream(ByVal stream As IO.Stream) As Image
    'Following code is for reading image from camera
    Const WM_CAP As Short = &H400S
    Const WM_CAP_DRIVER_CONNECT As Integer = WM_CAP + 10
    Const WM_CAP_DRIVER_DISCONNECT As Integer = WM_CAP + 11
    Const WM_CAP_EDIT_COPY As Integer = WM_CAP + 30
    Const WM_CAP_SET_PREVIEW As Integer = WM_CAP + 50
    Const WM_CAP_SET_PREVIEWRATE As Integer = WM_CAP + 52
    Const WM_CAP_SET_SCALE As Integer = WM_CAP + 53
    Const WS_CHILD As Integer = &H40000000
    Const WS_VISIBLE As Integer = &H10000000
    Const SWP_NOMOVE As Short = &H2S
    Const SWP_NOSIZE As Short = 1
    Const SWP_NOZORDER As Short = &H4S
    Const HWND_BOTTOM As Short = 1
    Public switch As Integer 'keep track of beep/motor

    Dim bmap As Bitmap
    Dim iDevice As Integer = 0 ' Current device ID
    Dim hHwnd As Integer ' Handle to preview window
    Public photoNo As Integer = 1 'tracks photos taken

    Declare Function SendMessage Lib "user32" Alias "SendMessageA" _
```

```
        (ByVal hwnd As Integer, ByVal wMsg As Integer, ByVal wParam As
Integer, _
        <MarshalAs(UnmanagedType.AsAny)> ByVal lParam As Object) As Integer

    Declare Function SetWindowPos Lib "user32" Alias "SetWindowPos" (ByVal
hwnd As Integer, _
        ByVal hWndInsertAfter As Integer, ByVal x As Integer, ByVal y As
Integer, _
        ByVal cx As Integer, ByVal cy As Integer, ByVal wFlags As Integer)
As Integer

    Declare Function DestroyWindow Lib "user32" (ByVal hndw As Integer) As
Boolean

    Declare Function capCreateCaptureWindowA Lib "avicap32.dll" _
        (ByVal lpszWindowName As String, ByVal dwStyle As Integer, _
        ByVal x As Integer, ByVal y As Integer, ByVal nWidth As Integer, _
        ByVal nHeight As Short, ByVal hWndParent As Integer, _
        ByVal nID As Integer) As Integer

    Declare Function capGetDriverDescriptionA Lib "avicap32.dll" (ByVal
wDriver As Short, _
        ByVal lpszName As String, ByVal cbName As Integer, ByVal lpszVer As
String, _
        ByVal cbVer As Integer) As Boolean

    'declarations for motorbee
    Dim BeeON As Boolean = False 'used as mask so that user can select
haptic output
    Dim noise As Boolean = True 'used as mask so that user can select audio
output
    Declare Function InitMotoBee Lib "mtb.dll" () As Boolean
    'mtb.dll is copied in windows/system32 as a library file
    Declare Function Digital_IO Lib "mtb.dll" (ByRef inputs As Integer,
ByVal outputs As Integer) As Boolean
    Declare Function SetMotors Lib "mtb.dll" (ByVal on1 As Integer, ByVal
speed1 As Integer, ByVal on2 As Integer, ByVal speed2 As Integer, ByVal on3
As Integer, ByVal speed3 As Integer, ByVal on4 As Integer, ByVal speed4 As
Integer, ByVal servo As Integer) As Boolean

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'load main form, initialise buttons
        BtnSave.Enabled = True
        btnStart.Enabled = True
        BtnStop.Enabled = False
        'initialise camera
        LoadDeviceList()
        If lstDevices.Items.Count > 0 Then
            btnStart.Enabled = True
            lstDevices.SelectedIndex = 0
            btnStart.Enabled = True
        Else
            lstDevices.Items.Add("No Capture Device")
            btnStart.Enabled = False
        End If

        BtnStop.Enabled = False
        BtnSave.Enabled = True
        picCapture.SizeMode = PictureBoxSizeMode.StretchImage
```

```
        InitMotoBee()      ' initialise the MotorBee
    End Sub

    Private Sub LoadDeviceList()
        Dim strName As String = Space(100)
        Dim strVer As String = Space(100)
        Dim bReturn As Boolean
        Dim x As Integer = 0
        ' Load name of all avialable devices into the lstDevices
        Do
            '   Get Driver name and version
            bReturn = capGetDriverDescriptionA(x, strName, 100, strVer,
100)
            ' If there was a device add device name to the list
            If bReturn Then lstDevices.Items.Add(strName.Trim)
            x += 1
        Loop Until bReturn = False
    End Sub
    Private Sub BtnStop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnStop.Click
        ClosePreviewWindow()
        BtnSave.Enabled = False
        btnStart.Enabled = True
        BtnStop.Enabled = False
    End Sub
    Private Sub OpenPreviewWindow()
        Dim iHeight As Integer = picCapture.Height
        Dim iWidth As Integer = picCapture.Width
        ' Open Preview window in picturebox
        hHwnd = capCreateCaptureWindowA(iDevice, WS_VISIBLE Or WS_CHILD, 0,
0, 640, _
            480, picCapture.Handle.ToInt32, 0)
        ' Connect to device
        If SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0) Then
            'Set the preview scale
            SendMessage(hHwnd, WM_CAP_SET_SCALE, True, 0)
            'Set the preview rate in milliseconds (note this was at 66 - I
have put it much lower)
            SendMessage(hHwnd, WM_CAP_SET_PREVIEWRATE, 10, 0)
            'Start previewing the image from the camera
            SendMessage(hHwnd, WM_CAP_SET_PREVIEW, True, 0)
            ' Resize window to fit in picturebox
            SetWindowPos(hHwnd, HWND_BOTTOM, 0, 0, picCapture.Width,
picCapture.Height, _
                    SWP_NOMOVE Or SWP_NOZORDER)

            BtnSave.Enabled = True
            BtnStop.Enabled = True
            btnStart.Enabled = False
            Label1.Text = "camera found"
        Else
            ' Error connecting to device close window
            DestroyWindow(hHwnd)
            Label1.Text = "no camera"
        End If
    End Sub

    Private Sub ClosePreviewWindow()
        ' Disconnect from device
        SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0)
        ' close window
```

```
        DestroyWindow(hHwnd)
    End Sub

    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click
        'starts video feed
        iDevice = lstDevices.SelectedIndex
        OpenPreviewWindow()
    End Sub

    Private Sub BtnLoad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnLoad.Click
        ' Displays an OpenFileDialog so the user can select a photograph
        Dim openFileDialog1 As New OpenFileDialog()
        Dim returnValue As Image

        openFileDialog1.Filter = "Data|*.*"
        openFileDialog1.Title = "Select the picture"
        ' Show the Dialog.
        If openFileDialog1.ShowDialog() = DialogResult.OK Then
            'Read picture in file using bmap
            Dim readFile As IO.FileStream
            readFile = IO.File.OpenRead((openFileDialog1.FileName()))
            returnValue = Bitmap.FromFile(openFileDialog1.FileName())
            picCapture.Image = returnValue
            ' SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0)
            readFile.Close()
        End If
    End Sub

    Sub SafeDispose(ByRef d As IDisposable)
        If Not d Is Nothing Then
            d.Dispose()
        End If
    End Sub

    Private Sub BtnGo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnGo.Click, Timer1.Tick
        SafeDispose(CType(Me.PictureBox2.Image, IDisposable))
        Me.PictureBox2.Image = Nothing
        'weightings (total=100)
        Dim wsat As Integer = 33
        Dim whue As Integer = 33
        Dim wchrom As Integer = 34
        'Dim wedge As Integer = 33

        'corners of safe area
        Dim a As Integer = 50 '120 for narrow '10 * 5 'x1  (50)
        Dim b As Integer = 180    'y1
        Dim c As Integer = 270 '200 for narrow '54 * 5 'x2 (270)
        Dim d As Integer = 230    'y2

        'y-co-ordinates of 4 zones
        Dim Y1 As Integer = 100
        Dim Y2 As Integer = 149

        Dim count As Integer
        'Dim , count2 As Integer
        Dim i As Integer
        Dim total As Integer = 500 'trackbar total
        Dim maxsat As Integer = 0
```

```
        Dim maxhue As Integer = 0
        Dim maxChrom As Integer = 0
        Dim maxChrompos As Integer = 0
        Dim maxsatpos As Integer = 0
        Dim maxhuepos As Integer = 0
        Dim maxpos2 As Integer = 0
        'Dim edge(100)
        Dim bmap As Bitmap

        'motorbee settings:
        Dim s1, s2, s3, s4 As Integer
        Dim o1, o2, o3, o4 As Boolean
        Dim sv As Integer
        Dim inputs As Integer
        Dim outputs As Integer
            s1 = 215 'speed of motor
            s2 = 215
            s3 = 215
            s4 = 215
        Digital_IO(inputs, outputs) 'read/write to Motor Bee Board
        'detect if button pressed and board connected
        'If ((inputs And 1) = 0) Then 'And inputs And 2 = 1
        'Beep(1000, 10)
        'switch = (switch + 1) Mod 4
        'Label3.Text = switch
        'Select Case switch
        '    Case 0
        'noise = False
        'BeeON = False
        '    Case 1
        'noise = True
        'BeeON = False
        '    Case 2
        ' noise = False
        ' BeeON = True
        '     Case 3
        ' noise = True
        ' BeeON = True
        ' End Select
        ' End If
        Dim data As IDataObject
        If (Not btnStart.Enabled) Then
            SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
            ' Get image from clipboard and convert it to a bitmap
            data = Clipboard.GetDataObject()
            bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)),
Image)
            'use image from camera if present otherwise use image in
picturebox(ie loaded)
        Else
            bmap = picCapture.Image
        End If

        Dim out1 As New Bitmap(bmap.Width, bmap.Height,
Imaging.PixelFormat.Format32bppArgb)
        Dim bOut1 As New BitmapDirect(out1)
        Dim bIn As New BitmapDirect(bmap)
        Dim iH As Integer = bmap.Height - 1
        Dim iW As Integer = bmap.Width - 1
        Dim sat As Integer
        Dim hue As Integer
```

```
        Dim Chrom, AA, Pb, Pr As Integer
        'Dim red, green, blue As Integer
        'Dim red1, red2 As Integer
        Dim SatCount(360)
        Dim hueCount(360)
        Dim ChromCount(360)
        For i = 0 To 360
            SatCount(i) = 0
            hueCount(i) = 0
            ChromCount(i) = 0
        Next
        Dim col(320, 240) As Integer
        'Do sat and hue together
        Using bOut1
            Using bIn

                For y As Integer = b To d   'inImg.Height - 1
                    For x As Integer = a To c   'inImg.Width - 1
                        sat = bIn.GetSaturation(x, y) * 255
                        hue = bIn.GetHue(x, y) * 255 / 360
                        'bOut1.SetPixel(x, y, (Color.FromArgb(sat, sat,
sat)))
                        SatCount(sat) += 1
                        If SatCount(sat) > maxsat Then
                            maxsatpos = sat
                            maxsat = SatCount(sat)
                        End If
                        hueCount(hue) += 1
                        If hueCount(hue) > maxhue Then
                            maxhuepos = hue
                            maxhue = hueCount(hue)
                        End If
                        Pb = bIn.GetPb(x, y) '* 255
                        Pr = bIn.GetPr(x, y) '* 255
                        AA = Math.Min((bIn.GetA(x, y) + 160) * 255 / 160,
255)
                        'Chrom = 0.1 * (Pb + Pr + 8 * AA)
                        'Chrom = 0.25 * (Pb + Pr + 4 * AA)
                        Chrom = 1 / 6 * (Pb + Pr + 4 * AA)
                        col(x, y) = Chrom 'so col(x,y) contains chrom map
within the safe-zone
                        ChromCount(Chrom) += 1
                        If ChromCount(Chrom) > maxChrom Then
                            maxChrompos = Chrom
                            maxChrom = ChromCount(Chrom)
                        End If

                        bOut1.SetPixel(x, y, col(x, y))
                    Next
                Next
                maxhue = 0
                For i = 0 To 360
                    If System.Math.Abs(maxhuepos - i) > 20 Then
                        If hueCount(i) > maxhue Then
                            maxhue = hueCount(i)
                            maxpos2 = i ' allows for a secondary hue peak
ie two key colours in the safe zone
                        End If
                    End If
                Next i
```

```
                For x = 50 To 270
                    For y = (Y1 - 50) To Y2
                        sat = bIn.GetSaturation(x, y) * 255
                        hue = bIn.GetHue(x, y) * 255 / 360

                        Pb = bIn.GetPb(x, y) '* 255
                        Pr = bIn.GetPr(x, y) '* 255
                        AA = Math.Min((bIn.GetA(x, y) + 160) * 255 / 160,
255)
                        'Chrom = 0.1 * (Pb + Pr + 8 * AA)
                        'Chrom = 0.25 * (Pb + Pr + 4 * AA)
                        Chrom = 1 / 6 * (Pb + Pr + 4 * AA)
                        If sat > maxsatpos - 10 And sat < maxsatpos + 10
Then
                            col(x, y) = col(x, y) + wsat
                        End If
                        If (hue > maxhuepos - 5 And hue < maxhuepos + 5) Or
(hue > maxpos2 - 5 And hue < maxpos2 + 5) Then
                            col(x, y) = col(x, y) + whue
                        End If

                        If Chrom > maxChrompos - 5 And Chrom < maxChrompos
+ 5 Then
                            col(x, y) = col(x, y) + wchrom
                        End If

                        bOut1.SetPixel(x, y, col(x, y) * 255 / 100)
                    Next
                Next
                For x = 50 To 270
                    For y = Y1 To Y1 + 1
                        bOut1.SetPixel(x, y, 0)
                    Next
                Next

            End Using
        End Using
        Me.PictureBox2.Image = out1
        Me.PictureBox2.Refresh()

        'area 1
        count = 0
        For x = 30 To 85 '50 To 105
            For y = Y1 To Y2
                If col(x, y) > 50 Then count = count + 1
            Next
        Next
        If count < 1000 Then
            If noise Then Beep(200, 10)
            LbArea1.BackColor = Color.Red
            s1 = 215
            o2 = 1 And BeeON  'motor 2 = speed 2; note area 1 is motor 2
because of belt
        Else
            LbArea1.BackColor = Color.Green
            o2 = 0
        End If

        'area 2
        count = 0
        For x = 106 To 160
```

```
            For y = Y1 To Y2
                If col(x, y) > 50 Then count = count + 1
            Next
        Next
        If count < 1000 Then
            If noise Then Beep(400, 10)
            LbArea2.BackColor = Color.Red
            s1 = 215
            o1 = 1 And BeeON  'motor 2 = speed 2; note area 2 is motor 1
because of belt
        Else
            LbArea2.BackColor = Color.Green
            o1 = 0
        End If
        'area 3
        count = 0
        For x = 161 To 215
            For y = Y1 To Y2
                If col(x, y) > 50 Then count = count + 1
            Next
        Next
        If count < 1000 Then
            If noise Then Beep(600, 10)
            LbArea3.BackColor = Color.Red
            s1 = 215
            o4 = 1 And BeeON  'motor 4 = speed 4; note area 3 is motor 4
        Else
            LbArea3.BackColor = Color.Green
            o4 = 0
        End If
        'area 4
        count = 0
        For x = 236 To 299 '216 To 270
            For y = Y1 To Y2
                If col(x, y) > 50 Then count = count + 1
            Next
        Next
        If count < 1000 Then
            If noise Then Beep(800, 10)
            LbArea4.BackColor = Color.Red
            s1 = 215
            o3 = 1 And BeeON  'motor 3 = speed 3; note area 3 is motor 4 on
belt
        Else
            LbArea4.BackColor = Color.Green
            o3 = 0
        End If

        'area 5
        count = 0
        For x = 50 To 270
            For y = (Y1 - 50) To (Y2 - 50)
                If col(x, y) > 50 Then count = count + 1
            Next
        Next
        If count < 4000 Then
            If noise Then Beep(1000, 10)
            LbArea5.BackColor = Color.Red
            s1 = s2 = s3 = s4 = 150
            o1 = 1 And BeeON
            o2 = 1 And BeeON
```

```
            o3 = 1 And BeeON
            o4 = 1 And BeeON
        Else
            LbArea5.backcolor = Color.Green
        End If

        SetMotors(o1, s1, o2, s2, o3, s3, o4, s4, sv)
    End Sub

    Private Sub BtnTimer_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnTimer.Click

        Timer1.Enabled = Not Timer1.Enabled
        If Timer1.Enabled Then
            BtnTimer.BackColor = Color.SkyBlue
        Else : BtnTimer.BackColor = Color.LightGray
        End If
    End Sub

    Private Sub BtnMotors_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnMotors.Click
        BeeON = Not BeeON
        If BeeON Then
            BtnMotors.BackColor = Color.SkyBlue
        Else
            BtnMotors.BackColor = Color.LightGray
            SetMotors(0, 1, 0, 2, 0, 3, 0, 4, 0) 'o1, s1, o2, s2, o3, s3,
o4, s4, sv ie switch off all motors
        End If
    End Sub

    Private Sub Btn_beep_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Btn_beep.Click
        noise = Not noise
        If noise Then
            Btn_beep.BackColor = Color.SkyBlue
        Else : Btn_beep.BackColor = Color.LightGray
        End If
    End Sub

    Private Sub BtnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnSave.Click
        Dim saveFileDialog1 As New SaveFileDialog()
        Dim bmap As Bitmap
        Dim data As IDataObject
        Dim di As String = "c:\users\anthony\new folder\photo\3Jan12\"
        Dim FName As String = ""
        '
        ' Copy image to clipboard
        '
        SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
        '
        ' Get image from clipboard and convert it to a bitmap
        '
        data = Clipboard.GetDataObject()

        If data.GetDataPresent(GetType(System.Drawing.Bitmap)) Then
            bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)),
Image)
        Else
            bmap = CType(picCapture.Image, Image)
```

```
        End If

        FName = di & "test" & Str(photoNo) & ".bmp"
        photoNo = photoNo + 1

        Dim saveFile As IO.FileStream
        saveFile = IO.File.Create(FName)
        bmap.Save(saveFile, Drawing.Imaging.ImageFormat.Bmp)
        saveFile.Close()
    End Sub

End Class
```

D.2 BitmapDirect Class

```
Option Strict On
Option Explicit On

Public Class BitmapDirect
    Implements IDisposable

    Dim bmpData As System.Drawing.Imaging.BitmapData
    Dim ptr As IntPtr
    Dim rect As Rectangle
    Dim bytes As Integer
    Dim rgbValues() As Byte
    Dim Kb As Single = 0.114
    Dim Kr As Single = 0.299

    Private bmp As Bitmap

    '// WARNING: This class was designed for 24bpp image format only
    '// Reference: http://msdn2.microsoft.com/en-us/library/5ey6h79d.aspx

    '// Usage: Create a new BitmapDirect object before working on data
    '//         make desired changes to pixels via Get/Set pixel methods
    '//         dispose BitmapDirect, which unlocks bitmap bits making it
usable

    Public Sub New(ByVal bitmap As Bitmap)
        Me.New(bitmap, New Rectangle(0, 0, bitmap.Width, bitmap.Height))
    End Sub

    Public Sub New(ByVal bitmap As Bitmap, ByVal area As Rectangle)
        Me.bmp = bitmap
        rect = area

        '// 24 bpp RGB is forced here intentionally
        bmpData = bmp.LockBits(rect, _
                    Drawing.Imaging.ImageLockMode.ReadWrite, _
                   Imaging.PixelFormat.Format24bppRgb)

        Dim dx As Integer = bitmap.Width

        bytes = bmpData.Stride * rect.Height
        ptr = bmpData.Scan0
        ReDim rgbValues(bytes - 1)

        ' Copy the RGB values into the array.
        System.Runtime.InteropServices.Marshal.Copy(ptr, rgbValues, 0,
bytes)
```

```
    End Sub

    Public Sub SetPixel(ByVal x As Integer, ByVal y As Integer, ByVal
intensity As Byte)
        rgbValues(bmpData.Stride * y + x * 3) = intensity
        rgbValues(bmpData.Stride * y + x * 3 + 1) = intensity
        rgbValues(bmpData.Stride * y + x * 3 + 2) = intensity
    End Sub

    Public Sub SetPixel(ByVal x As Integer, ByVal y As Integer, ByVal c As
Color)
        '// notice, alpha color data is ignored
        rgbValues(bmpData.Stride * y + x * 3) = c.R
        rgbValues(bmpData.Stride * y + x * 3 + 1) = c.G
        rgbValues(bmpData.Stride * y + x * 3 + 2) = c.B
    End Sub

    Public Function GetPixelIntensity(ByVal x As Integer, ByVal y As
Integer) As Single
        Dim c As Color = GetPixel(x, y)
        Return 0.333F * (CSng(c.R) + c.G + c.B)
    End Function

    Public Function GetPixel(ByVal x As Integer, ByVal y As Integer) As
Color
        Return Color.FromArgb(rgbValues(bmpData.Stride * y + x * 3), _
                              rgbValues(bmpData.Stride * y + x * 3 + 1),
_
                              rgbValues(bmpData.Stride * y + x * 3 + 2))
    End Function
    Public Function GetRed(ByVal x As Integer, ByVal y As Integer) As
Integer
        Return rgbValues(bmpData.Stride * y + x * 3)
    End Function
    Public Function GetGreen(ByVal x As Integer, ByVal y As Integer) As
Integer
        Return rgbValues(bmpData.Stride * y + x * 3 + 1)
    End Function
    Public Function GetBlue(ByVal x As Integer, ByVal y As Integer) As
Integer
        Return rgbValues(bmpData.Stride * y + x * 3 + 2)
    End Function
    Public Function GetSaturation(ByVal x As Integer, ByVal y As Integer)
As Single
        Dim pix As Drawing.Color
        pix = GetPixel(x, y)
        Return pix.GetSaturation 'this now calls the VB.NET color.method
    End Function
    Public Function GetHue(ByVal x As Integer, ByVal y As Integer) As
Single
        Dim pix As Drawing.Color
        pix = GetPixel(x, y)
        Return pix.GetHue 'this now calls the VB.NET color.method
    End Function

    Public Function GetPr(ByVal x As Integer, ByVal y As Integer) As Double
        'Return 0.5 * (GetRed(x, y) / 255 - GetY(x, y) / 255) / (1 - Kr)
        Return 128 + 112.439 / 256 * GetRed(x, y) - 94.154 / 256 *
GetGreen(x, y) - 18.285 / 256 * GetBlue(x, y)
    End Function
```

```
    Public Function GetPb(ByVal x As Integer, ByVal y As Integer) As Double
        'Return 0.5 * (GetBlue(x, y) / 255 - GetY(x, y) / 255) / (1 - Kb)
        Return 128 - 37.945 / 256 * GetRed(x, y) - 74.494 / 256 *
GetGreen(x, y) + 112.439 / 256 * GetBlue(x, y)
    End Function

    Public Function GetY(ByVal x As Integer, ByVal y As Integer) As Double
'Single
        'Return Kr * GetRed(x, y) / 255 + (1 - Kr - Kb) * GetGreen(x, y) /
255 + Kb * GetBlue(x, y) / 255
        Return 16 + 65.738 / 256 * GetRed(x, y) + 129.057 / 256 *
GetGreen(x, y) + 25.064 / 256 * GetBlue(x, y)
    End Function

    Public Function GetA(ByVal x As Integer, ByVal y As Integer) As Double
        'Code taken from http://www.easyrgb.com/index.php?X=MATH&H=02#text2
        ' to find XYZ colour space - but I only need X and Y
        Dim var_R As Double = (GetRed(x, y) / 255)      'R from 0 to 255
        Dim var_G As Double = (GetGreen(x, y) / 255)    'G from 0 to 255
        Dim var_B As Double = (GetBlue(x, y) / 255)     'B from 0 to 255
        Dim XX, YY, AA As Double

        If (var_R > 0.04045) Then
            var_R = ((var_R + 0.055) / 1.055) ^ 2.4
        Else : var_R = var_R / 12.92
        End If

        If (var_G > 0.04045) Then
            var_G = ((var_G + 0.055) / 1.055) ^ 2.4
        Else : var_G = var_G / 12.92
        End If
        If (var_B > 0.04045) Then
            var_B = ((var_B + 0.055) / 1.055) ^ 2.4
        Else : var_B = var_B / 12.92
        End If

        var_R = var_R * 100
        var_G = var_G * 100
        var_B = var_B * 100

        'Observer. = 2°, Illuminant = D65 - may have something to do with
lighting?
        XX = var_R * 0.4124 + var_G * 0.3576 + var_B * 0.1805
        YY = var_R * 0.2126 + var_G * 0.7152 + var_B * 0.0722
        'Z = var_R * 0.0193 + var_G * 0.1192 + var_B * 0.9505

        'Hunter LAB equations from
http://www.easyrgb.com/index.php?X=MATH&H=05#text5
        If YY = 0 Then
            AA = 0
        Else
            AA = 17.5 * (((1.02 * XX) - YY) / Math.Sqrt(YY))
        End If

        '(H)L = 10 * sqrt( Y )
        '(H)a = 17.5 * ( ( ( 1.02 * X ) - Y ) / sqrt( Y ) )
        '(H)b = 7 * ( ( Y - ( 0.847 * Z ) ) / sqrt( Y ) )

        Return AA

    End Function
```

```
#Region "Disposable"

    Private disposedValue As Boolean = False        ' To detect redundant
calls

    ' IDisposable
    Protected Overridable Sub Dispose(ByVal disposing As Boolean)
        If Not Me.disposedValue Then
            If disposing Then
                ' TODO: free managed resources when explicitly called
                ' Copy the RGB values back to the bitmap
                System.Runtime.InteropServices.Marshal.Copy(rgbValues, 0,
ptr, bytes)

                ' Unlock the bits.
                bmp.UnlockBits(bmpData)
            End If

            ' TODO: free shared unmanaged resources
        End If
        Me.disposedValue = True
    End Sub

#Region " IDisposable Support "
    ' This code added by Visual Basic to correctly implement the disposable
pattern.
    Public Sub Dispose() Implements IDisposable.Dispose
        ' Do not change this code.  Put cleanup code in Dispose(ByVal
disposing As Boolean) above.
        Dispose(True)
        GC.SuppressFinalize(Me)
    End Sub
#End Region
#End Region

End Class
```

Appendix E: Tabulated results of questionnaire

E.1 Basic Data

Basic Data					
ID	Age	Registered Blind?	Mobility Aids	Long cane training	Guide Dog Owner years
1	55-64	Yes	Long Cane, Sighted Guide, Symbol Cane	1-3 Years	None
2	45-54	Partially Sighted	None	None	None
3	<16	Partially Sighted	None	None	None
4	45-54	Yes	Long Cane, Sighted Guide	1-3 Years	None
5	55-64	Yes	Long Cane	>30	None
6	35-44	Partially Sighted	Long Cane, Sighted Guide, Symbol Cane		None
7	45-54	Partially Sighted	Guide Dog, Symbol Cane	None	
8	35-44	Yes	Guide Dog, Long Cane, Other, Sighted Guide	>30	10-19 years
9	55-64	Yes	Guide Dog, Long Cane, Sighted Guide	>30	>29
10	45-54	Yes	Long Cane, Sighted Guide	4-6	None
11	45-54	Yes	Guide Cane, Guide Dog, Long Cane, Sighted Guide, Symbol Cane	1-3 Years	None
12	55-64	Partially Sighted	Symbol Cane	None	None
13	35-44	Yes	Guide Dog	None	10-19 years
14	35-	Yes	Guide Dog, Long Cane	20-29	10-19 years

Basic Data					
ID	Age	Registered Blind?	Mobility Aids	Long cane training	Guide Dog Owner years
	44				
15	35-44	Yes	Long Cane	4-6	None
16	55-64	Yes	Other	None	None
17	35-44	Yes	Guide Dog, Sighted Guide	None	1-3 years
18	55-64	Yes	Guide Cane, Guide Dog, Long Cane, Other, Sighted Guide, Symbol Cane	10-19	20-29
19	45-54	Yes	Guide Dog, Sighted Guide, Symbol Cane	None	10-19 years
20	55-64	Yes	Sighted Guide, Symbol Cane	None	None
21	26-34	Yes	Symbol Cane	None	None
22	55-64	Yes	Guide Dog, Long Cane	7-9	4-6 years
23	45-54	Yes	Symbol Cane	None	None
24	35-44	Yes	Guide Dog, Long Cane	10-19	20-29
25	45-54	Yes	Guide Dog, Long Cane, Sighted Guide	10-19	10-19 years
26	55-64	Yes	Guide Cane, Guide Dog, Long Cane, Sighted Guide, Symbol Cane	20-29	20-29
27	55-64	Yes	Guide Dog, Long Cane	20-29	20-29
28	65-74	Yes	Guide Dog	None	>29

E.2 Navigation in Familiar Areas

Navigation					
ID	With sighted guide	With cane only	With dog only	Orientation & finding way familiar	Other
1	Once or twice a week	Seldom	Never	Ask directions, Familiarisation of route, Specific clues/landmarks	
2	Once or twice a week	Never	Never	General Listening, Other, Road Crossings	Satnav
3	Once or twice a month	Never	Never	Change in gradient, Change in surface, Familiarisation of route, General Listening, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	
4	> Once a day	Never	Never	Familiarisation of route, Long cane info	
5	Once or twice a month	Almost Every day	Never	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Long cane info, Road Crossings, Specific clues/landmarks, Specific noises	
6	Seldom	Never	Never	Change in gradient, Change in surface, Familiarisation of route, General Listening, Road Crossings, Specific clues/landmarks, Specific noises	
7	Once or twice a month	Never	Almost Every day	Familiarisation of route	
8	Once or twice a week	Once or twice a month	Almost Every day	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Long cane info, Other, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	GPS, travel announcements

Navigation					
ID	With sighted guide	With cane only	With dog only	Orientation & finding way familiar	Other
9	Once or twice a week	Once or twice a month	> Once a day	Change in gradient, Change in surface, Familiarisation of route, General Listening, Long cane info, Other, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	GPS
10	Once or twice a week	Almost Every day	Never	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Long cane info, Other, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	research, pre-plan
11	Never	Never	Never	Ask directions	
12	Seldom			Ask directions, Familiarisation of route, General Listening, Road Crossings, Specific clues/landmarks	
13	Once or twice a month	Never	> Once a day	Ask directions, Change in gradient, Change in surface, General Listening, Long cane info, Road Crossings, Specific noises, Specific smells	
14	Almost Every day	Once or twice a week	> Once a day	Ask directions, Change in surface, Familiarisation of route, General Listening, Other, Specific clues/landmarks, Specific noises	
15	Once or twice a week	Almost Every day		Change in gradient, Change in surface, Familiarisation of route, General Listening, Long cane info, Road Crossings, Specific clues/landmarks	
16	Almost Every day	Never	Never	Other	never out independently
17	Never		> Once a day	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Other, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	Trekker Breeze GPS (Humanware)

Navigation					
ID	With sighted guide	With cane only	With dog only	Orientation & finding way familiar	Other
18	Seldom	Seldom	> Once a day	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Other, Road Crossings, Specific noises, Specific smells	air currents from shops/passages, echoes, talking gps - gps on phone, Wayfinder,
19	Once or twice a week		Almost Every day	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	
20	Almost Every day	Never	Never	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Road Crossings, Specific clues/landmarks	
21	Once or twice a month	Never	Never	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Road Crossings, Specific clues/landmarks	
22	Once or twice a week	Almost Every day	> Once a day	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Long cane info, Road Crossings, Specific noises, Specific smells	
23	Never	Never	Never	Ask directions, Change in surface, Familiarisation of route, Road Crossings, Specific clues/landmarks	
24	Seldom	Once or twice a week	> Once a day	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Long cane info, Road Crossings, Specific clues/landmarks	
25	Once or twice a week	Seldom	Almost Every day	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Long cane info, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	

Navigation					
ID	With sighted guide	With cane only	With dog only	Orientation & finding way familiar	Other
26	Once or twice a week	Seldom	Almost Every day	Ask directions, Change in gradient, Change in surface, Familiarisation of route, General Listening, Long cane info, Road Crossings, Specific noises, Specific smells	
27	Seldom	Seldom	Once or twice a week	Ask directions, Change in surface, Familiarisation of route, General Listening, Other, Road Crossings, Specific smells	crossing buttons/rods
28	Seldom		Almost Every day	Ask directions, Change in surface, Familiarisation of route, General Listening, Other, Road Crossings, Specific clues/landmarks	dog remembers

E.3 Navigation in Unfamiliar Areas

Unfamiliar areas				
ID	How often visited (independent)	Visit with guide?	Navigation in Unfamiliar Areas	Other
1	1 or 2 a year	No	Ask directions, Partial knowledge of route, Specific clues/landmarks	
2	1 or 2 a year	No	General Listening, Other, Road Crossings	Satnav
3	1 or 2 a year	Yes	Change in gradient, Change in surface, General Listening, Road Crossings, Specific clues/landmarks, Specific noises	
4	Never	Yes	Change in surface, Specific clues/landmarks	
5	1 or 2 a year	No	Ask directions, Change in gradient, Change in surface, General Listening, Long cane info, Other, Partial knowledge of route, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	Position of sun

Unfamiliar areas				
ID	How often visited (independent)	Visit with guide?	Navigation in Unfamiliar Areas	Other
6	1 or 2 a year	Yes	Ask directions, Change in surface, General Listening, Road Crossings, Specific noises	
7	Never	Yes		
8	1 or 2 a week	No	Ask directions, Change in gradient, Change in surface, General Listening, Long cane info, Other, Partial knowledge of route, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	GPS, travel announcements
9	1 or 2 a month	No	Ask directions, Change in gradient, Change in surface, General Listening, Long cane info, Other, Partial knowledge of route, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	Whatever info u can get, GPS
10	1 or 2 a year	No	Ask directions, Change in gradient, Change in surface, General Listening, Long cane info, Other, Partial knowledge of route, Road Crossings, Specific noises, Specific smells	research, pre-plan
11	Never	Yes		
12	1 or 2 a year	No	Ask directions, General Listening, Road Crossings, Specific clues/landmarks	
13	1 or 2 a week	No	Ask directions, General Listening, Other, Partial knowledge of route	
14	1 or 2 a month	No	Ask directions, General Listening, Other, Partial knowledge of route, Specific clues/landmarks	
15	1 or 2 a month	No	Ask directions, Change in gradient, Change in surface, General Listening, Long cane info, Partial knowledge of route, Road Crossings, Specific clues/landmarks	
16	Never	Yes		
17	1 or 2 a month	No	Ask directions, Change in gradient, Change in surface, General Listening, Other, Partial knowledge of route, Road Crossings, Specific	GPS Satnav

Unfamiliar areas				
ID	How often visited (independent)	Visit with guide?	Navigation in Unfamiliar Areas	Other
			clues/landmarks, Specific noises, Specific smells	
18	1 or 2 a week	No	Ask directions, Change in gradient, Change in surface, General Listening, Other, Partial knowledge of route, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	Pattern of steps. Indents on pavement, side doors
19	1 or 2 a week	No	Ask directions, Change in gradient, Change in surface, General Listening, Partial knowledge of route, Road Crossings, Specific clues/landmarks, Specific noises, Specific smells	
20	1 or 2 a year	No	Ask directions, Change in gradient, General Listening, Partial knowledge of route, Road Crossings, Specific clues/landmarks	
21	1 or 2 a year	No	Ask directions, Other, Partial knowledge of route, Specific clues/landmarks	take taxi if don't know where going
22	1 or 2 a week	Yes	Ask directions, Change in gradient, General Listening, Long cane info, Partial knowledge of route, Road Crossings	
23	Never	Yes	Ask directions, Change in surface, Partial knowledge of route, Road Crossings, Specific clues/landmarks	
24	1 or 2 a month	No	Ask directions, Change in gradient, Change in surface, General Listening, Long cane info, Partial knowledge of route, Road Crossings, Specific clues/landmarks	
25	1 or 2 a year	No	Ask directions, Long cane info, Other	Rely on sighted assistance in unknown area
26	Never	Yes		
27	1 or 2 a year	No	Ask directions, Change in surface, General Listening, Partial knowledge of route, Road Crossings	

Unfamiliar areas				
ID	How often visited (independent)	Visit with guide?	Navigation in Unfamiliar Areas	Other
28	1 or 2 a month	No	Ask directions, Change in surface, General Listening, Partial knowledge of route, Specific clues/landmarks, Specific noises, Specific smells	

E.4 Difficult Urban Environments

Difficult Urban Environment					
ID	Difficult Urban Environment	Location	Feelings	Differences	Orientation comments
1	Home Zone, Pedestrian areas, Shared surface streets	Cardiff, Exeter, Truro, Rhayader, Llandindod Wells, Birmingham	Anxious	Uncertainty over right of way, stopping when crossing?	take long time
2	Home Zone, Pedestrian areas, Shared surface streets	Motherwell, Glasgow	Very nervous, vulnerable	Wider pavements better but can lose bearings	Satnav - can go wrong
3	Shared surface streets	Aylesbury	Very unsafe, not clear where roads are, fast taxis	More unsettling, more hesitant, longer time to cross	partial knowledge, status cane, sounds and sights of local places
4	Home Zone, Pedestrian areas, Shared surface streets	Colchester	With guide a little anxious; more confident previously with guide dog	need to concentrate more with cane	familiar route- feel ground, listen; unfamiliar with dog previously by counting kerbs etc
5	None	None			
6	None	None			
7	None				
8	Pedestrian areas, Shared surface streets	Peterborough, York	Nervous,	no traffic boundary, drivers and cyclists less careful and aware, guide dog needs more commands	Difficult, keep close to shoreline
9	Pedestrian areas, Shared surface streets	Reading	more care needed, frightening	difficult to know which direction vehicle travelling	walk along side of street as if there were a pavement

Difficult Urban Environment					
ID	Difficult Urban Environment	Location	Feelings	Differences	Orientation comments
10	Pedestrian areas, Shared surface streets		vulnerable to traffic,	slower and more reliant on help from others	patience, ask for help
11	None	Truro, St Austell, Plymouth	Scared	train barriers, relying on public	ask people
12	Pedestrian areas, Shared surface streets	local	confusing, disorientating	cyclists on pedestrian areas	concentration, use of stick
13	Pedestrian areas, Shared surface streets	Farnborough, Guilford, Woking, Redhill	vulnerable re skateboarding and cycling, don't feel safe	less tactile and defined guides, no rules	guide dog, asking for help
14	Pedestrian areas	pedestrian high streets	confident with guide dog, challenging with long cane		working straight lines with curbs and guide dog, ask
15		Newcastle centre	anxiety, pedestrians not looking where going	difficult to follow straight line, random street furniture, poor flow of pedestrians	noise from shop fronts, tactile paving, paving type differences,
16	None				
17	Pedestrian areas, Shared surface streets	local town	very uneasy; vulnerable to traffic	v disorientated due to sound echoes from lorries and buses	guide dog, asking public
18	Pedestrian areas, Shared surface streets	Melbourne, Sloan Square, indoor shopping centres, Kingston town centre	need to pay attention, need experience dog, daunting	indoors sounds masked, lifts stairs can't be found, GPS doesn't work	plan route, keep to building line
19	None				
20	None				
21	None				
22	Home Zone, Pedestrian areas, Shared surface streets	Aberdeen station, Perth, St Devenick's Mews Home Zone, Cults,	Frightened, insecure, vulnerable	orientation difficult, Guide dog is hesitant and uncertain	stick to building line, ask for help
23	Pedestrian areas, Shared surface streets	Bury	fine by day, stressed at night	vulnerably and stressed	as before
24	Pedestrian areas, Shared surface streets	Bury	fine by day, stressed at night	vulnerably and stressed	as before

Difficult Urban Environment					
ID	Difficult Urban Environment	Location	Feelings	Differences	Orientation comments
25	None				
26	Shared surface streets		a little apprehensive	unsure which area is safe	
27	Pedestrian areas			wait for assistance	shared space impossible without assistance
28	Pedestrian areas	Coleraine	disorientated near buildings, anxious when vehicles moving	easier to miss landmarks such as side streets and openings	dog who knows route, try to use as though pavements present i.e. stay near building line

E.5 Comments

Comments			
ID	Comment 1	Comment 2	Comment 3
1	Pre-Planning helps	Allow lots of time	
2	Cone system in Glasgow helps	Power chair and vibrations can be confusing	
3	Use landmarks	Verbal instructions from friends	
4	Knowledge of route essential	Waiting for dog	Won't try new route independently
5	Retain mental map	Has own website!	
6	Sufficient road lighting	no obstacles	bus with voice location
7	Only go out independently with guide dog	Will go anyway with sighted guide	
8	Wayfinding services increasingly important and affordable especially in unfamiliar places	NOTE - probably means GPS	
9	use GPS system -within a few feet of destination	dog is by far best help	

Comments			
ID	**Comment 1**	**Comment 2**	**Comment 3**
10	research	less busy time	ask for help confidently while being friendly
11	travel to places known previous to sight loss	carrying a cane helps others understand	
12	counting number of steps	concerned eyesight might get worse	
13	street furniture staying in same place	friend to explain route	GPS is not exact enough
14	people generally helpful especially with guide dog		
15	People helpful in larger cities	in quieter areas hard to find people to ask	
16	none		
17	shared surface experience is v stressful-puts guide dog at risk	bus drivers feel stress if they see someone with mobility problem or a small child	
18	GPS phone v helpful; plinted map	tubes, buses talk inside - but not outside to announce number/route	large open parks and indoor shopping centres difficult
19	I can not underestimate the work that my Guide dog actually does		
20	none		
21	in Italy found different acoustics due to architecture made it impossible to live there as planned. Avoided crossing roads.		
22	extractor fans, café sounds and smells or Lush are great landmarks		
23	I ask for assistance and use bus route from home to town		
24	get assistance from Guide Dog Assoc		

Comments			
ID	Comment 1	Comment 2	Comment 3
	when need to undertake new routes in my area		
25	pre-plan every journey	navigational aid give incorrect info so would prefer to use own senses	
26	senses - shaded/non-shaded; wind		
27	Can't take risks with Guide Dog	prefer vibration feedback to beep	
28	guide dog makes one less aware of obstacles, not so good when looking for a landmark		

Appendix F: Experiments

F.1 Indoor Experiment 1 Lights on

Trial	Distance	Start time	End Time	Speed	km/hr	Certainty on	off	False positive
1	3	3.43333333	6.6	0.947368	3.410526	**	**	n
2	2.25	20.1333333	22.43333	0.978261	3.521739	**	**	n
3	2.75	33.4666667	35.96667	1.1	3.96	*	*	n
4	2.5	50.6333333	53.76667	0.797872	2.87234	**	**	n
5	2.5	5.96666667	8.933333	0.842697	3.033708	**	**	n
6	3	20.8666667	23.53333	1.125	4.05	?	**	n
7	2.5	39.2333333	41.86667	0.949367	3.417722	**	**	n
8	2.5	52.8666667	55.4	0.986842	3.552632	**	**	n
9	2.5	11.2333333	13.33333	1.190476	4.285714	**	**	n
10	2.5	24.9	27.5	0.961538	3.461538	**	**	n
				Average	3.556592			

Notes

1) Distances and time read from video of experiment. Distances are marked on the floor.

2) Certainty of device indicating a move onto or off the surface was graded as

   a) ** certain,
   b) less certain,
   c) ? unsure,
   d) n not indicated.

3) False positives are where the device indicates a surface change which is not actually there

F.2 Indoor Experiment2 Lights off

Trial	Distance	Start time	End Time	Speed	km/hr	Certainty on	off	False positive
1	2.5	29.8	32.56667	0.903614	3.253012	**	**	?
2	2.5	44.0333333	46.36667	1.071429	3.857143	n	**	n
3	2.5	59.9333333	62.7	0.903614	3.253012	n	*	n
4	2.5	14.4333333	17.1	0.9375	3.375	n	n	n
5	2.5	29.2666667	31.8	0.986842	3.552632	n	n	n
6	2.5	48.1333333	51.66667	0.707547	2.54717	**	**	n
7	2.5	6.76666667	9.766667	0.833333	3	**	**	n
8	2.5	21.0666667	24.1	0.824176	2.967033	n	n	n
9	2.5	35.9333333	38.86667	0.852273	3.068182	n	*	n
10	2.5	51.3333333	54.36667	0.824176	2.967033	*	**	n
				Average	3.184022			

F.3 Indoor Experiment 3 with Torch

Trial	Distance	Start time	End Time	Speed	km/hr	Certainty on	off	False positive
1	2.5	40.8333333	43.56667	0.914634	3.292683	**	**	n
2	2.5	56.0666667	58.5	1.027397	3.69863	**	**	n
3	2.5	10.8	13	1.136364	4.090909	**	**	n
4	2.5	27.2	29.46667	1.102941	3.970588	*	**	n
5	2.5	40.3333333	42.6	1.102941	3.970588	n	**	(n)
6	2.5	55.0333333	57.43333	1.041667	3.75	**	**	n
7	2.5	9.46666667	11.9	1.027397	3.69863	*	*	n
8	2.5	22.6333333	25.2	0.974026	3.506494	**	**	**
9	2.5	39.0666667	41.43333	1.056338	3.802817	*	*	n
10	2.5	53.2333333	55.5	1.102941	3.970588	n	*	*
				Average	3.775193			

F.4 Outdoor Experiment

Trial	Certainty on	off	False positive
1	**	**	y
2	**	*	n
3	*	**	n
4	**	**	n
5	**	**	n
6	**	**	n
7	**	**	?
8	**	**	n
9	**	**	y
10	**	**	y

Appendix G: Shared Space Photographs

G.1 Observation and Categorisation of Photographs

In this section various photographs are examined and features recorded.

1.0 Germany

Photographs from Brenner Dietrich Dietrich: Diplomingenieure Freie Architekten Büro Für Städtebau Freiburg (Office of qualified architectural engineers, Freiburg BDD, n.d)

1.1.1 Haslach

Shared Surface Y	
Street composition	Cobbled
Pavement composition	Cobbled
Pvmnt-strt interface	visible drainage channel
Objects on pavement	car, flower pots, lamp-post
Location of pedestrians	road and pavement

1.1.2 Haslach

Shared Surface Y	
Street composition	Cobbled
Pavement composition	Cobbled
Pvmnt-strt interface	visible drainage channel
Objects on pavement	flower pots, chair lamp-post
Location of pedestrians	road and Pavement
Additional stepped pavement with	tables/chairs

1.2.1 Wolfach

Shared Surface Y	
Street composition	Cobbled
Pavement composition	Cobbled with linear stones
Pvmnt-strt interface	visible change in surface
Objects on pavement	barrels in channel, chair, bench
Location of pedestrians	Pavement and on linear stone channel

1.2.2 Wolfach

Shared Surface Y
Street composition Cobbled
Pavement composition Cobbled
Pvmnt-strt interface visible drainage channel
Objects on pavement barrels in channel, litter bin, chair, bench
Location of pedestrians Pavement only

2.0 Holland

Photographs from Shared Space site (shared space, n.d)

2.1 Oudehaske

Shared Surface Y
Street composition Clinker bricks
Pavement composition small paving stones
Pvmnt-strt interface clear colour change
Objects on pavement tree, bus shelter
Location of pedestrians Pavement only

2.2 Oldeberkoop

Shared Surface Y
Street composition Clinker bricks
Pavement composition small paving stones
Pvmnt-strt interface clear colour change
Objects on pavement chairs, menu board, tables
Location of pedestrians none seen

2.3 Makkinga Village

Shared Surface	No
Street composition	Clinker bricks
Pavement composition	small paving stones
Pvmnt-strt interface	clear colour change, kerb
Objects on pavement	rocks, bench, bin, tree
Location of pedestrians	none seen

2.4 Donkerbroek

Shared Surface	No
Street composition	tarmac
Pavement composition	small paving stones
Pvmnt-strt interface	clear colour change, Kerb, bollards
Objects on pavement	wheelie bin, tree
Location of pedestrians	none seen
Informal crossing area seen	

2.5 Wolvega

Shared Surface	Yes
Street composition	clinker bricks
Pavement composition	small paving stones
Pvmnt-strt interface	change in colour and orientation, bollards
Objects on pavement	scooter
Location of pedestrians	on crossing
Large pedestrian crossing	

2.6 Oosterwolde

Shared Surface	Yes
Street composition	clinker bricks
Pavement composition	small paving Stones, larger ones
Pvmnt-strt interface	2 changes in colour, some railings
Objects on pavement	bench
Location of pedestrians	road

2.7 Drachten

Shared Surface	Yes but raised inner walkway
Street composition	asphalt
Pavement composition	small paving
Pvmnt-strt interface	unclear change in colour,
Objects on pavement	fountain, cyclist,
Location of pedestrians	pavement
Informal pedestrian crossing	

2.8 Haren

Shared Surface	Yes
Street composition	asphalt
Pavement composition	asphalt
Pvmnt-strt interface	bolards, corduroy stones,
Objects on pavement	trees, lamp-post
Location of pedestrians	pavement
Informal pedestrian crossing	

3.0 United Kingdom

3.1 London: Exhibition Road, Kensington (V & A, 2010)

Shared Surface	Yes
Street composition	grey and white stones
Pavement composition	same
Pvmnt-strt interface	none
Objects on pavement	trees, lamp-post
Location of pedestrians:	pavement
Informal pedestrian crossing	

3.2 Stranraer, Scotland (Farrington, A. 2011)

Shared Surface	Yes
Street composition	clinker bricks
Pavement composition	paving stones
Pvmnt-strt interface	white paving
Objects on pavement	trees, bin, band-stand
Location of pedestrians:	pavement

3.3 New Road, Brighton (Gillett, 2009)

Shared Surface	Yes
Street composition	clinker bricks
Pavement composition	paving stones
Pvmnt-strt interface	white paving
Objects on pavement	trees, bin, band-stand
Location of pedestrians:	pavement

G.2 Analysis of Photographs

Shared Surface present	13
Raised kerb	2
Street composition same as pavement?	5
Street different from pavement	10
Pavement Street interface by colour	4
Interface by kerb	2
Interface by surface change	4
Interface by channel	4
no interface	1

Table G.1

In the sample of 15 Shared Space sites, the majority have a shared surface and an interface between the "street" where vehicular traffic mainly flows and the "pavement" which is safer for pedestrians. Blind and partially sighted people could benefit from a device such as the one described in the thesis in these circumstances.