Strategies for image visualisation and browsing

Janjusevic, Tijana

QUEEN MARY, UNIVERSITY OF LONDON
SCHOOL OF ELECTRONIC ENGINEERING
AND COMPUTER SCIENCE

STRATEGIES FOR IMAGE VISUALISATION AND BROWSING

Thesis submitted to University of London
in partial fulfilment of the
requirements for the degree of
Doctor of Philosophy

Tijana Janjusevic
Supervisor: Prof. Ebroul Izquierdo

London, 2010.

Za mog batu Dusana

Acknowledgements

I would like to thank professor Ebroul Izquierdo for giving me the opportunity to pursue the PhD in Multimedia and Vision Research Group at Queen Mary. Apart from the knowledge I gained, this experience enabled me to work with some very smart people and discuss my ideas with them on everyday basis. Therefore, I would like to say big thanks to my co-workers and friends from MMV Group for making my Phd an amazing experience in both research and human aspects.

I also had the honour to work with exceptional colleagues within the EU project RUSHES. I would like to thank in particular Leticia Fuentes Ardeo, Mikel Frutos Hernandez and journalists at EiTB for priceless help during the experimental evaluation. My gratitude also goes to professor Riccardo Leonardi for his creative thoughts which lead to interesting new idea reported in this thesis.

I would like to state my big gratitude to my friends and colleagues Krishna Chandramouli and Giuseppe Passino for their extensive patience during our discussions and their great friendship which means a lot to me.

I would like to thank all my friends for being great people.

Special thanks goes to my love, my friend and my colleague Sergio Benini for giving me his support in every sense of that word and being there for me.

Last but not least I would like to thank my parents, Dragica and Relja for teaching me the right things, and giving me love and support throughout my life. I would also like to thank my brother Dusan, and the rest of my family for all their love.

To all the others who influenced my life, thanks for your help.

"Our field is not about making pretty pictures. It is about helping people with the complex tasks involved in data analysis and understanding."
John T. Stasko '03

Abstract

Strategies for image visualisation and browsing

The exploration of large information spaces has remained a challenging task even though the proliferation of database management systems and the state-of-the art retrieval algorithms is becoming pervasive. Significant research attention in the multimedia domain is focused on finding automatic algorithms for organising digital image collections into meaningful structures and providing high-semantic image indices. On the other hand, utilisation of graphical and interactive methods from information visualisation domain, provide promising direction for creating effcient user-oriented systems for image management. Methods such as exploratory browsing and query, as well as intuitive visual overviews of image collection, can assist the users in finding patterns and developing the understanding of structures and content in complex image data-sets.

The focus of the thesis is combining the features of automatic data processing algorithms with information visualisation. The first part of this thesis focuses on the layout method for displaying the collection of images indexed by low-level visual descriptors. The proposed solution generates graphical overview of the data-set as a combination of similarity based visualisation and random layout approach.

Second part of the thesis deals with problem of visualisation and exploration for hierarchical organisation of images. Due to the absence of the semantic information, images are considered the only source of high-level information. The content preview and display of hierarchical structure are combined in order to support image retrieval. In addition to this, novel exploration and navigation methods are proposed to enable the user to find the way through database structure and retrieve the content.

On the other hand, semantic information is available in cases where automatic or semi-automatic image classifiers are employed. The automatic annotation of image items provides what is referred to as higher-level information. This type of information is a cornerstone of multi-concept visualisation framework which is developed as a third part of this thesis. This solution enables dynamic generation of user-queries by combining semantic concepts, supported by content overview and information filtering.

Comparative analysis and user tests, performed for the evaluation of the proposed solutions, focus on the ways information visualisation affects the image content exploration and retrieval; how efficient and comfortable are the users when using different interaction methods and the ways users seek for information through different types of database organisation.

Contents

List of Figures

List of Tables

GLOSSARY

BMU - Best Matching Unit
CIF - Common Intermediate Format
DPI - Dots Per Inch
HAC - Hierarchical Agglomerative Clustering
IR - Information Retrieval
VIR - Visual Information Retrieval
LCA - Leading Cluster Analysis
LUV - perceptually uniform color space used in computer graphic
MCB - Multi Concept Browser MDS - Multi Dimensional Scaling
MPEG - Moving Picture Experts Group
PCA - Principal Component Analysis
RF - Relevance Feedback
SOM - Self Organising Maps
QCIF - Quarter Common Intermediate Format
TSVQ - Tree-Structured Vector Quantisation

Chapter 1

Introduction

Development of multimedia technologies along with the availability of cheap digital recording and storage devices caused the tremendous growth of digital image collections. With the constant explosion of available digital content, despite the proliferation of the database management systems and the state-of-the art retrieval algorithms, the topic of accessing relevant information from huge image collections is still an active issue. Significant research attention in the multimedia domain is focused on automatic indexing and organising of digital image repositories to support rapid and reliable image retrieval. Once the content is indexed and organised, the question is how to transform numerical or textual information into a suitable visual form for user to gain insight into the data-set. In fact, an adequate graphical representation of data information can enable efficient content query, access and management.

By definition, *Information Visualisation* visually represents abstract data to reinforce human cognition, thus enabling the viewer to gain useful knowledge about the internal structure of the data and causal relationships in it [81]. Employment of information visualisation for creation of smart interfaces has the ability to support the users in exploratory browsing and query, accessing adequate content and most importantly to accompany the dynamic evolution of user information needs. Efficient support can be achieved by enabling the user to find patterns in indexed and/or organised complex data-sets.

The specific aim of this thesis is to apply information visualisation techniques to address the problem of *semantic gap* in image retrieval. The issue of semantic gap is a consequence of existing inability to extract full image semantics directly from images. In other words it is the difference between the complex semantic level of human cognition and directly derived low-level image visual information. In particular, the work presented here addresses the problem of visualising and exploring indexed and organised image collections, for getting insight into the specific content domain, to understand organisational patterns and thus manage the content itself.

Despite the lasting existence of *semantic gap* most of the retrieval systems are ignoring the potential of information visualisation techniques and methods for lessening or solving the stated issue. Developing the techniques to improve capabilities of retrieval systems most researchers still focus more on the domain of statistical analysis and pattern recognition.

In fact, if automatic indexing and organisation methods managed to succeed in extracting relevant information accurately on their own, visualisation would be useless [196]. Since these automatic methods are not foolproof, using visualisation can be highly justified.

Apart from the direct involvement of the user, there are two important advantages of visual data exploration over statistics or machine learning (recognised as standard automatic data mining techniques). First is the fact that visual data exploration is able to deal with highly non-uniform and noisy data, and second, that it is intuitive and requires no understanding of complex mathematical or statistical algorithms or parameters [98].

In general, visual data analysis allows faster data exploration and enables users to achieve good results even if the automatic algorithms fail. However, in order to be efficient and useful, visualisation and interaction solutions have to be meaningfully designed and combined.

In particular, for efficient management of huge data collections, several issues need to be adequately treated by information visualisation:

1. Which graphical representation of large data-set can efficiently and intuitively convey structural organisation and useful information for data management?;

2. What visualisation and interaction methods can reduce or overcome the existence of the semantic gap in retrieval systems?;

3. What access and exploration methods are able to keep up with the dynamics of changes in user information needs.

Considering this, the visualisation and exploration methods proposed here aim at tackling previously stated issues by:

- Providing an intuitive overview of the image collection using similarity based data visualisation;

- Enabling access and exploration of relevant images in low-level structures of image indices using visual representation and interaction;

- Neutralising the faulty performance of the automatic algorithms using novel query and visualisation strategies;

- Including methods to accompany dynamics by which user is adapting his/her information needs.

1.1 Context of the thesis

Within image retrieval there is a certain border which statistical analysis and pattern recognition tools have not been able to cross yet. Results obtained by advanced automatic or semi-automatic algorithms can save a lot of human effort during the retrieval process, but there is still a huge void between their achievements and human requirements.

The idea behind this work is that employing information visualisation as an intermediate tool might positively influence the efficiency of retrieval systems. Instead of inferring the semantics of user needs by processing lower level of information, using information visualisation techniques and tools it is possible to meet the user "halfway". This approach would allow combining the power of processing algorithms with human cognition abilities. The solution for visual data exploration, proposed in this thesis, uses the results of image indexing and organisation algorithms for visualisation and exploration of image data-sets, aiming to improve the overall performance of the image retrieval system.

Providing support for accessing image databases requires two main stages: storage and retrieval [82]. In the storage phase images are processed to extract features which describe their visual content. These features are then used for indexing and organisation. There are two general types of image indexing techniques: the content-based (or low-level, which adopts primitive features such as colour, texture, etc.) and semantic. The content indexing approach extracts visual features directly from images and stores them in numerical form. The information which automatic algorithms can extract directly from multimedia content are: colours or (distribution of colours), image texture, shapes, image transforms in the frequency domain (e.g. Discrete Cosine Transform or DCT, etc) and other similar characteristic which machine algorithms know how to process. On the other hand, the way automatic processes understand image texture is not something a common person can grasp very easily. The term *texture* by itself most likely does not mean anything to people outside multimedia domain, not to mention DCT. So how can we expect a user to find image(s) based on texture if not familiar with such term? To solve this problem image analysis and understanding algorithms are employed to produce image indices on higher semantic level, thus providing more meaningful descriptions from the user point of view.

This second type of image indexing, semantic, is achieved either by running automatic algorithms or by manual annotation. Usually the automatic procedures employ pattern recognition techniques to classify images into one predefined category or classes and image semantics are derived using knowledge basis. On the other side, manual annotation generated by humans is rarely available, due to the time and efforts needed for such task.

Irrespective of the semantic indexing approach the problem with generated indexes is their low reliability, inconsistency and in cases where automatic algorithms are used, the fact that they often do not conform to a standard language. Query on a human level would be specified in form of "I want to find an image of a sailing competition" or "I need an image with waterfalls". This level is far from the numeric description of image colour properties or "high confidence" that image contains a *boat*. Stated facts cause the existence of the so-called *semantic gap* between the capabilities of image processing algorithms and the requirements of the *human user*.

One attempt to reduce the gap is done by organisation of image indices according to some specified criteria. The goal of these algorithms, such as *flat* and *hierarchical clustering*, is to group similar images together thus improving the efficiency of the retrieval process. However, the fact that these algorithms are fallible introduces a significant impact on the overall retrieval system achievement in which they are employed.

After analysing contemporary systems for image retrieval from the user perspective the significant lack of visual data exploration support was detected. For this reason the search for the semantic-gap solution was transferred into domain of information visualisation. The main reason was the fact that visual data exploration enables a high level of user involvement in the retrieval process, something that was not emphasised enough so far. Involving human cognitive abilities as opposed to a fully automated system might be able to solve or lessen current retrieval issues and this is where information visualisation can incorporate it's abilities.

1.1.1 Focus of the thesis and research approach

The task of assisting the human in understanding the data properties and features, the way it can be easily perceived and exploited is a complex one. Integration of several scientific fields into a computer-based information visualisation creates powerful combination of knowledge, able to provide significant assistance to the user in human-computer environment to achieve stated task.

Computer science is first important element of information visualisation. It's influence is realised through presence of computer graphics and human-computer interaction (HCI) in information visualisation. Another relevant aspect is the way humans perceive displayed information. This is the reason for the presence of the psychological domain. Semiotics, or the study of signs, enables the meaningful use of symbols to visualise and transfer the knowledge. Design and art provide their "fine" impact on the final visual representation in terms of visual elements used, their chosen features, etc.

Generally speaking, information visualisation is visual representation of large-scale collections of numerical information. This is done in order "to see what lies within, determine the answer to a question, find relations, and perhaps apprehend things which could not be seen so readily in other forms" [63].

This thesis focuses on the visual and navigational strategies for dealing with image repositories when different types of information about the content is available. In this work existing content information, extracted by automatic and semi-automatic processing algorithms, is combined with information visualisation in order to present the content and information relevant to the user and specific task. The goal is to to exploit different results provided by state-of-the art multimedia processing algorithms and highly involve the user in order to supersede their shortcomings.

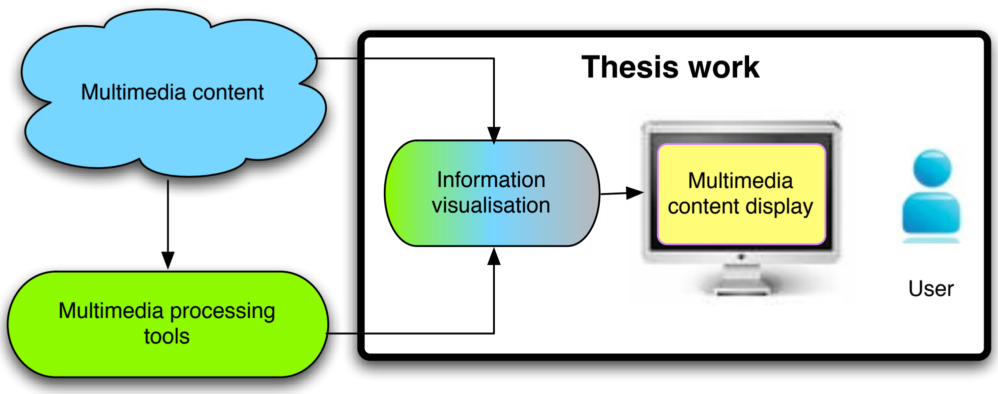

Figure 1.1: Focus of the thesis.

Depending on the way images are indexed there are various visualisation and exploration solutions. In some cases visualisation methods developed for abstract data are not directly applicable in case of images due to the importance of image properties. For example, images should not be displayed as small points in space, since their complete semantic information is then lost. This makes layout more complex since the size of data object has an impact on visual layout. Using information visualisation for image database presentation and exploration thus requires fine tuning depending on the available information and user task.

The first part of this thesis deals with an image data-set where indices are low-level visual image features. The problem addressed here is how to define an intuitive layout solution for interactive exploration of such image set. As opposed to standard visualisation solution in image domain, which aim to represent the image collection as a faithful 2-dimensional embedding of high-dimensional features, this work took a slightly different approach. Instead of applying dimensionally reduction techniques directly on the high-dimensional image indices, a combination of partitional clustering and dimensionality reduction method is proposed for image layout generation. A ranked random approach preserves local image similarity while similarity embedding based on dimension scaling conveys global similarity of images in the data-set. The generated layout provides content overview where exploration is supported by interactive visualisation technique.

Low-level image indices organised into a hierarchical structure require a specific presentation and access solution and this issue is tackled in the second part of this thesis. Such organisation contains no semantic information and thus raises the question of visualisation and above all exploration strategies. Particular visualisation solution proposed by this work displays the structure and the content preview to support user's navigation decisions. However, the highly important question is how to access the relevant content (which is one or more images). If user looks for an image in hierarchicaly organised data-set, without knowledge about the exact location of the content, some initial exploration has to be performed. This initial stage is modelled and implemented here as novel application functionality. The application takes the user through various levels of hierarchy in order to enable the retrieval of at least one "interesting image". Initial and general navigation is supported by different interaction strategies that can be used in combination or alone depending on user preferences. Visual and interactive methods aim at supporting dynamic changes in user's information needs and knowledge gathering.

In the third part of this thesis, visualisation of image content indexed by semantic labels is tackled. The layout of images developed in the first thesis part was employed and later adapted according to the properties of semantic indexed data-set. What started as a layout of semantic *visual concepts* evolved into a prototype multi-concept query and browsing tool. The pre-defined classes populated by any classifier are combined with user cognitive abilities to create broad spectre of queries. Query specification is supported by providing different information to the user about the content using information visualisation approaches. Again various interactive methods are used for supporting the user while performing his/her task.

1.2 Thesis contribution

The contribution of this thesis can be summarised through three work sections as stated in previous chapter. As the first part, Chapter 4 deals with an image set indexed with low-level, content-based image indices, a problem of graphical image layout was initially addressed. With a distinctive focus on spatial graphical positioning of images in a collection, the contribution of the first part of the thesis are:

- A proposal for a novel similarity-based layout method for displaying the overview of the image collection, using combination of clustering, Multi-Dimensional Scaling and ranked Gaussian technique;
- A comparison of the proposed layout with related similarity-based MDS layouts of the image collection, using the example of colour based image indices (MPEG-7 Colour Layout). The comparison has been performed by observing the "semantics" of image locations in both solutions.

The second part of the thesis contribution is related to hierarchicaly organised image collection and is described in Chapter 5. The issues identified and addressed are ways to visualise and access the content in the hierarchy when semantic image information is absent. In order to tackle these issues the contribution of this thesis includes:

- Proposal for a novel access strategy which adopts statistical analysis principles for navigation through the content hierarchy;
- Implementation of the hierarchical browsing and visualisation tool where user navigation is supported by visualisation of the repository structure and related content preview. The tool incorporates a novel combination of interaction techniques for supporting user navigation and search;
- The "smart" retrieval strategy has been proposed in order to exploit the similarity properties of hierarchical organisation for retrieving set of relevant content items;
- Reports of the hierarchical browser evaluation where behavioural patterns and strategies taken by users are noted. Results of professional and non-professional user experiments (in their working environment) give insight into types of information users found important and useful for accomplishing the retrieval task in real scenarios.

The third set of thesis contributions specified in Chapter 6, emerged from the goal of creating a visual interactive environment for image retrieval based on set of semantic indices. Combining results of image classifier and information visualisation related contributions include:

- Implementation of the visual multi-concept browser prototype for specifying a wide range of queries by integrating classification algorithm results and human cognition using information visualisation and interaction tools. The goal of the developed application is to support the dynamic and visual query formulation by combining a limited set of semantic image categories;

- Proposal for a potentially effective information visualisation for supporting the concept selection in multi-concept browser;
- Discussion of a potentialy effective information visualisation technique for providing an overview of the query space in a multi-concept browser. The query space displays the sub-set of the entire image collection filtered by user concept selection and aims at enabling content exploration without complete relying on classification results;
- Results of evaluation which include comparison of the proposed method with two standard retrieval approaches as well as users satisfaction and general feedback.

1.3 Thesis outline

This thesis is organised as follows:

Chapter 2 gives an overview of techniques and tools found in the domain of information visualisation. The distinctive focus is on the methods for multi-dimensional data visualisation and browsing, as well as representation and navigation techniques suitable for hierarchicaly organised content.

Chapter 3 discusses interface solutions of existing image retrieval system and the level on which they employ information visualisation and interaction techniques. The review is performed according to visualisation complexity of interfaces starting from standard grid based displays. Content access is regarded as a separate topic since it is considered that information visualisation can play an significant role in that particular domain.

In Chapter 4 the spatial, similarity based layout of image collection is proposed. This solution integrates clustering and layout techniques for visualising image collections. A standard similarity based visualisation technique based on Multi-Dimensional Scaling (MDS) is used to layout the centres of the cluster while ranked Gaussian generation of image positions ensures fast displaying of images. Spatial distortion is an interactive technique employed to support exploration of the displayed content in more detail. The proposed layout was compared with analog MDS mapping of images which is one of the most common similarity based visualisation approaches. The comparison was performed with the goal of assessing the difference in information conveyed in both cases.

Chapter 5 discusses the work conducted for the development of a novel exploration, navigation and access technique for a hierarchy of images. Particular hierarchical structure of data-base is created based on low-level visual information. This meant that visualisation and navigation had to be performed with total absence of semantic information. The novel access method models the random behaviour of the user in the exploration stage using “biased random jumps” through the hierarchy. The navigation through the structure is supported with structure and content preview and several interaction methods, such as sequential access, “jump-to-leaf” and detailed node exploration. The evaluation scenario was designed with the goal of assessing the usefulness and efficiency of the visualisation and exploration techniques proposed as well as to observe the strategies and impressions of the user during the engagement with the system.

Novel query formulation methods and an exploration-based image retrieval strategy is proposed in **Chapter 6**. The developed visual system enables the user to visually combine semantic concepts for specifying broad range of queries based on a small number of extracted image indices. Search for relevant images is supported by spatial visualisation, detailed content examination on request and various other interactive elements. The evaluation procedure aimed at testing both system performance from image retrieval aspects as well as user satisfaction.

Thesis concludes with **Chapter 7** where the obtained results are discussed and conclusions made regarding the proposed solutions. The same chapter specifies considered future research directions.

Chapter 2

Information Visualisation

Combining human knowledge and perception with the power of data processing tools can make a huge contribution to the process of data mining and retrieval. In fact, a visual representation of information creates a connection between the system and its user thus providing the insight into the data, enabling the user to infer the existing patterns and interactively achieve a task. Once data is visually rendered, a user is able to perform a visual data *exploration*: a process in which user gathers information about the system and the data thus increasing his/her domain knowledge. While exploring, user's information needs adapt according to the data and the system, which corresponds to the dynamic nature of data mining.

Information visualisation is the discipline which studies graphical methods for efficiently presenting information to the user. The objective of visualisation is to improve understanding of the data being presented and to efficiently direct the user towards fast and successful task accomplishment. This is achieved by generating a meaningful graphical layout for displaying information and employing methods which enable user information management. Layout is "the way that something is arranged" [147]. It establishes the overall appearance, relative importance, and relationships between displayed graphic elements. On the other hand, as stated at [201], "interaction techniques are used in information visualisation to overcome various limitations in screen real estate and bounded cognition". In order to use visualisation methods for dedicated systems, such as image retrieval, a detailed analysis of existing approaches in information visualisation is required. For this reason in this chapter an overview of visualisation methods and applications is given, with a special focus on techniques relevant for this thesis: visualisation of multivariate and hierarchical data, 2D visual techniques combined with interaction methods such as distortion and so on.

2.1 Taxonomy of data

The fact that data is present in wide range of different contexts such as programming languages, data mining and communications, leads to different perceptions and definitions of this term. Data taxonomy stated here, refers to the properties of data observed from the visualisation perspective [98].

According to [98] it is possible to identify six distinctive types of data to be visualised:

- one-dimensional data;
- two-dimensional data;
- multi-dimensional data;
- text and hypertext;
- hierarchies and graphs;
- algorithms and software.

For some data this classification might be equivocal depending on it's visual interpretation. For example, although a text document can be considered as textual data, it can also be considered as multi-dimensional type if observed as a vector of terms.

Image collection can be classified into more than one stated data category depending on the image abstraction used. Here "abstraction" is considered to be a "result of generalisation by reducing the information content of a concept or an observable phenomenon, to retain only information which is relevant for a particular purpose" as stated in [202].

In case the information is extracted directly from the image, it will usually have the form of a feature vector which is a multi-dimensional numeric representation of image graphical content. In case of manual annotation, a set of textual descriptions will provide semantically meaningful image abstraction. Abstraction obtained by e.g. partitional or hierarchical clustering will provide advanced structural information which can be used to visualise images.

For this reason, throughout this chapter images and related abstraction forms are discussed within appropriate data types. In particular, special attention is given to **multi-dimensional data** and **hierarchies**, since they are image abstraction forms addressed in this thesis.

2.1.1 One-dimensional and two-dimensional data

From the perspective of information visualisation, when speaking of one, or two-dimensional data it is either referred to data with one/two explicit properties or one/two dimensions extracted from a bigger set of dimensions for visualisation purposes. Temporal data is typical example of one-dimensional data type. Examples of visualisation solutions for temporal data presentation are *ThemeRiver* [72] and *Timeline* [78]. Figure 2.1 shows and example of *ThemeRiver* visualisation used for displaying topics in collection of Fidel Castro's speeches, interviews and articles in the period from end 1959 to mid 1961. The popularity of article topics can easily be followed using this visualisation. Figure 2.2 is a screen shot of BBC web-site showing British History using *Timeline* visualisation. Two examples of interface are shown here: first showing the entire historical timeline divided according to important periods and historical events (e.g. reign of Tudors, Civil War and revolution, etc.); and the second displaying the focus on one smaller time interval so called "Vikings and Anglo-Saxons" where three historical events occurred.

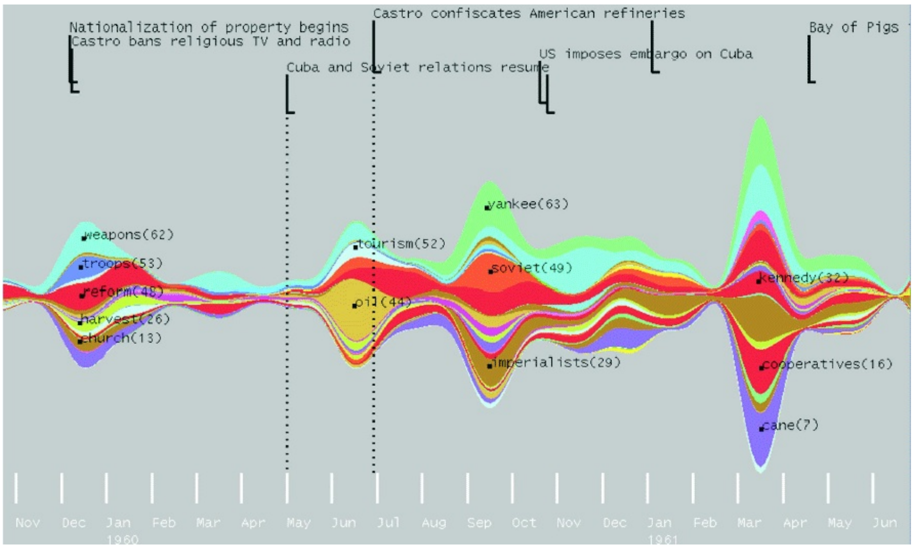

Figure 2.1: *ThemeRiver* visualisation of themes in collection of Fidel Castro's speeches, interviews and articles from end 1959 to mid 1961.

Another example of one-dimensional visualisation is *WordCount*, a dynamic animated visualisation of the most common 86800 words in the English language [70] shown in Figure 2.3. The words in the display are ranked in order of commonness. Each word is scaled to reflect its frequency relative to the words that precede and follow it, giving a visual barometer of relevance. The size of the word indicates its frequency of it's use, or in other words the more it is commonly used the bigger is the word's displayed size.

Two dimensional data can be easily visualised with two dimensional plots or his-

Figure 2.2: BBC British History Timeline visualisation found at BBC web-site at [12].

Figure 2.3: *WordCount* visualisation of the 86,800 most frequently used English words.

tograms. One histogram example is shown in Figure 2.4 which displays distribution of scores for one final exam. As seen, it is an effective way of conveying information regarding the exam results and students' performance. Another typical example of 2D data visualisation is the visualisation of geographical data rendered using maps. Probably the most famous example is one of Google Maps shown in Figure 2.5.

2.1.2 Multidimensional data

Multidimensional data is by definition a data which is described as set of values corresponding to more than one dimension. Examples of such data are tables from relational databases with large number of columns. This type of data is very interesting from the visualisation perspective since there is no direct way of mapping it to the screen. Since it is only possible to visualise data in two or three dimensional space, different methods are applied for transforming the high-dimesional information into suitable presentation form and are given in Section 2.2.

When talking about this type of data it is useful to mention here that, depending on the way an image abstraction is obtained, visualisation of an image collection can

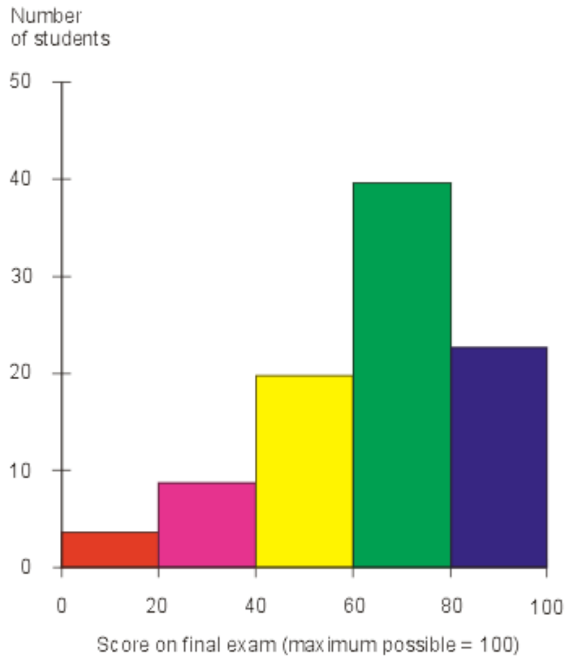

Figure 2.4: Histogram of student's final exams scores [189].

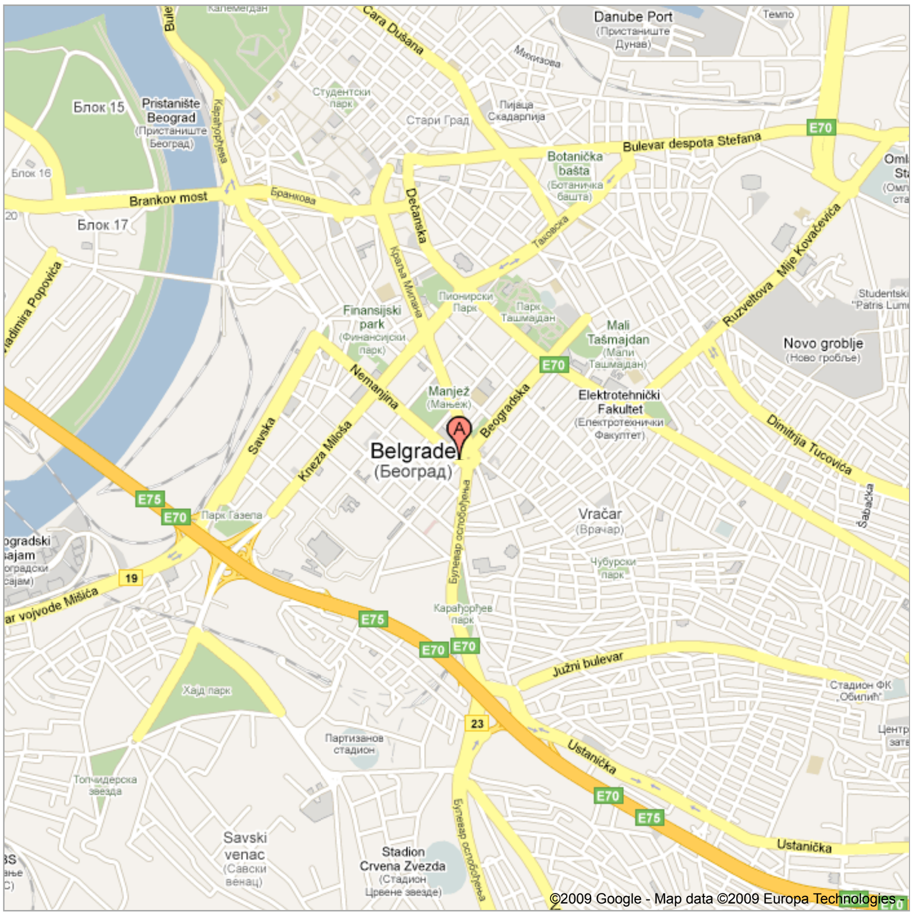

Figure 2.5: Google Map visualising the geographical position of Belgrade, capital of Serbia.

be seen as visualisation of a multidimensional data set.

Multidimensional image indexing

Image index is an abstraction of one image, capable of representing certain properties such as distribution of colour or semantic image content. Visual indexing techniques are based on features such as colour, texture and shape extracted directly from an image. The extracted information is stored in a form of a *feature vector*. An image feature vector is an n-dimensional vector of numerical values that represents the content of that particular image. This form of image representation is useful since almost all processing algorithms, especially in machine learning, require a numerical representation of objects as an input. The feature values describe inner characteristics of region or pixels which can not be described by unique scalar value. For example, in the Colour Histogram the features correspond to colours and the feature values are numbers of pixels with that particular colour. On the other hand Colour Layout is an MPEG descriptor which "effectively represents the spatial distribution of color of visual signals in a very compact form" [93]. From the information perspective a feature vector F_n is n-dimensional form of information and can be represented as:

$$F_N = \{f_1, f_2, \ldots, f_N\} \tag{2.1}$$

where f_i is the i-th feature. The feature vector can based on a single image property, such as colour, or can be weighted combination of several properties (e.g colour and shape).

Apart from visual features, another standard type of image feature are semantic indices (most frequently specified in textual form). Due to the fact that image can be described with one or more semantic terms it is correct to classify this information as multi-dimensional.

2.1.3 Text and hypertext

To be able to distinguish this data type from other forms, "text" has to be clearly defined. It can be defined as a "written alphabetical form of natural language" [209]. From the perspective of visualisation when talking about text data, it is mostly referred to textual documents. Hypertext is text which is "connected" with other text (on the Web or in the database) which can be immediately accessed by humans. The statistical and semantic nature of text, expressed by the frequency and context of words, enable different visualisation approaches. The goal of text visualisation is to transform text information into an adequate visual representation.

Visualisation tools aim to enable exploration of text documents by for example, visually highlighting key terms from a document [50] as shown in Figure 2.7. On the other hand, understanding the relationships between the documents terms might be an effective way to gain new insights [43]. As seen in Figure 2.6, visualisation shows terms found in documents as different coloured circles. Circles are connected by lines in case when terms co-exist in the same document. Observing relationships between terms, users can learn about the content of documents and find out for example, that large number of documents containing term "America" contains also terms "Qaeda", "Iraq" and "Congress" (as shown in the same figure). It can be easily inferred that the content of the documents is mostly politic and war related. The authors in [209] emphasise the benefits of visual representation of documents by context abstraction and specialisation of text documents. This spatial representation can then be explored and documents accessed through it.

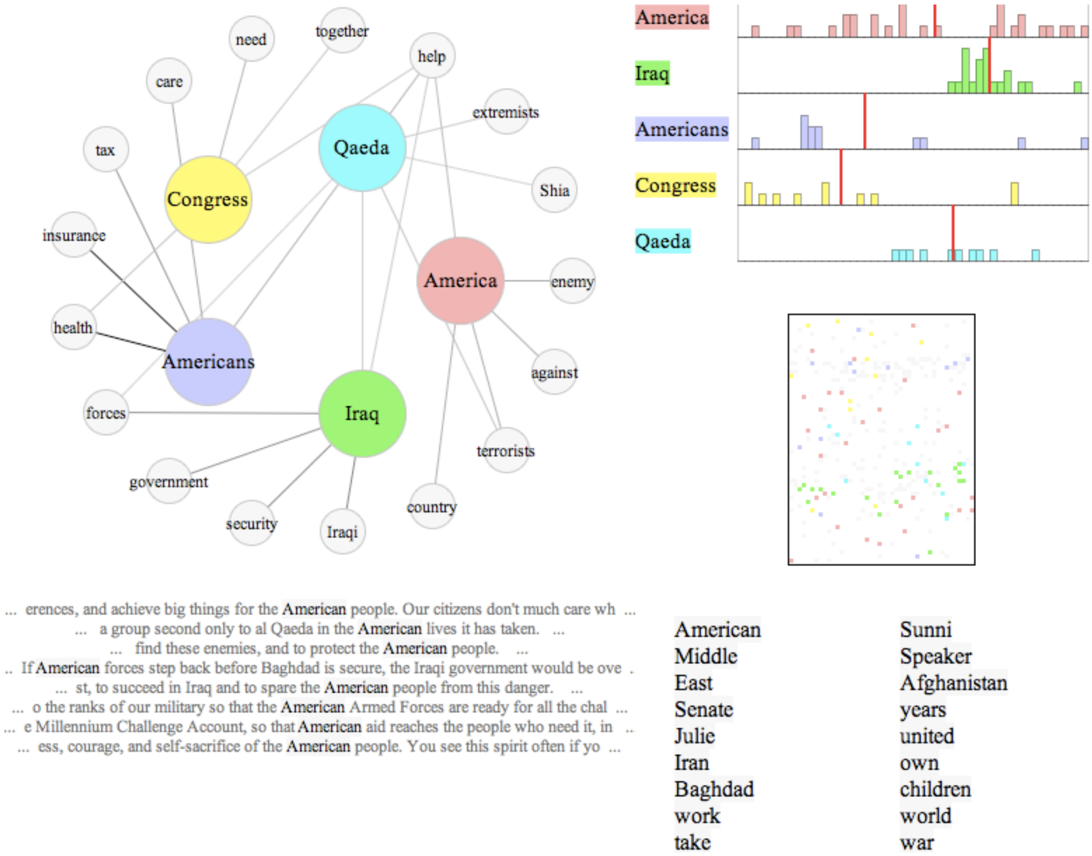

Figure 2.6: Visualising the relations between the items of the document collection [43].

2.1.4 Hierarchies and graphs

In case when the goal of visual representation is to convey the relationships between data objects, these relationships can be formalised using graphs and graph

Figure 2.7: Visualising the headlines with BigSpy [50] .

theory. On the other hand, when these relationships span over multiple levels, hierarchical structure is a more adequate representation method.

A graph is a mathematical structure used to model pairwise relations between objects in a data collection. Graphs consists of objects represented as nodes which are mutually connected by edges, where edges represent the relationships between the nodes. One of the biggest issues in graph visualisation is how to generate the graph layout when number of graphs nodes and edges is large.

There is a set of visualisation and interaction solutions for graph display which address this and similar graph drawing issues [139], [9], [134] and [135]. Outside of theoretical graph environment, graphs can be used as a representation of complex structures such as social networks or web-pages [136].

On the other hand, a data hierarchy is an arrangement of data items in which the data relations can be defined through multiple levels. It is an inherent structure where each data item, or node, has a single parent node (except for the root node). Nodes can have sibling nodes with whom they share the same parent node, as well as child nodes. Hierarchical structures are a quite common type of data organisation and huge number of visualisation solutions have been developed for their presentation [150], [103], [6], [19], [66] and [114].

Hierarchical organisation of images

When talking about images, hierarchical structures are obtained in most cases using a hierarchical clustering algorithms. The process of hierarchical clustering consists of multiple partitioning steps, which may run from a single cluster containing all objects to N clusters each containing a single object. There are two types of hierarchical clustering: *agglomerative* methods, which proceed by series of fusions of N objects into groups, and *divisive* methods, which separate N objects into successively finer groups.

In both cases the result is a hierarchical structure S organised on n levels, $\mathcal{S} = \{L_1, L_2, \ldots, L_n\}$. Based on this organisation, relationships between nodes on consecutive levels can be noted. Parent-child relations which exist in hierarchical structures enable data organisation into more manageable units. In image retrieval and visualisation it is an effective way of breaking information into more segments thus reducing the information overload on the user. On the other hand, observing several clusters on one hierarchy level provides information on similar and dissimilar data or image items. Thus enabling the user to focus on one sub-group of images.

2.1.5 Algorithms and debugging

In software engineering visualising algorithmic schemes can significantly improve understanding at how the specific algorithm works, what are the results and how they can be used. This is especially true when dealing with one algorithm for the first time. Drawing work flow is one of the first actions people usually take when planing the software development. It shows the idea, points to possible drawbacks and potential mistakes while planning. On the other hand, without programming platforms such as Eclipse [53] it would be very hard to follow the written code or more importantly to debug the code in case of malfunction.

2.2 Visualisation techniques

Based on the graphical representation used existing visualisation techniques can be classified into one of the following categories [98]:

- standard 2D and 3D techniques;
- geometrically transformed displays;
- iconic displays;
- dense pixel displays;
- stacked displays.

Not all visualisation techniques can be used for every data type in the taxonomy given in the previous section. For example, some techniques have limitations in number of data properties they can display. Within this section the focus is given to visualisation techniques suitable for multidimensional and hierarchical data types, such as dimensionality reduction methods, projections and tree views, respectively.

2.2.1 Standard 2D/3D displays

The simplest data display forms are x/y plots, bar charts and grids. Chart visualisation is frequently used method for generating intuitive display of some results. One typical chart is shown in Figure 6.15 displaying results of the test results for five users. Each users results have distinctive colour. After gathering the answers to 14 questions, the results are stored in a table and visualised in a chart. It is clear that this visual representation enables quick understanding of the value distribution over several data items, an estimation of minimum and maximum value, estimation of mean value and so on. The displayed bar chart is generated based on a table given in Figure 2.9 of numerical values, whose examination would require more time for obtaining the same conclusions.

As discussed in Chapter 3 a grid is a common visualisation method adopted for image browsing and retrieval. In some applications, standard displays are still proving to have advantage over highly-sophisticated visualisation techniques that are too complicated for the user. A grid display is an "easy-to-use" visualisation method due to it's visual simplicity and clarity of representation. For example, grid arrangement in image visualisation prevents overlapping of displayed images, thus enabling easy examination of their visual content. However, to easily examine one image in order to acknowledge it's content, image size has to be carefully chosen. The problem is that for hosting such image sizes a grid must have limited number of cells. Due to the common sizes of image collections this implies extraction of limited image set to be displayed and existence of images not shown to the user. If a huge number of images are displayed, the grid looses it's important feature - easy understanding of image content. On the other hand, grid does not convey large amounts of information such as relationships between images displayed, connection with non-displayed images (in case of limited sized grid) and so on. For this reason in some cases grid visualisation is not very efficient if used as a standalone visualisation method.

2.2.2 Geometrically transformed displays

Geometrically transformed displays is a set of visualisation solutions which use graphical interpretations of data dimensions and the data values per dimension. Among those methods a frequently used approach is *Parallel Coordinates* [83]. This visualisation method treats all data dimensions uniformly by representing them as

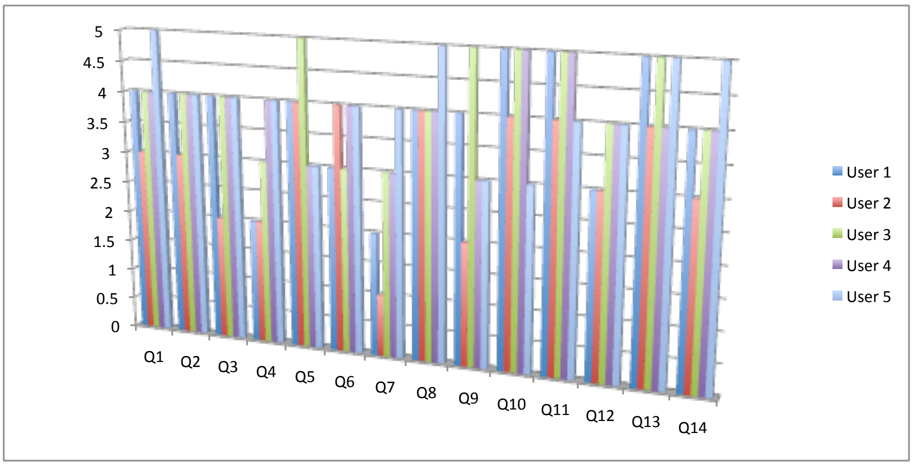

Figure 2.8: Bar chart display.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14
User 1	4	4	4	2	4	3	2	4	4	5	5	3	5	4
User 2	3	3	2	2	4	4	1	4	2	4	4	3	4	3
User 3	4	4	4	3	5	3	3	4	5	5	5	4	5	4
User 4	4	4	4	4	3	4	3	4	3	5	5	4	4	4
User 5	5	4	4	4	3	4	4	5	3	3	4	4	5	5

Figure 2.9: Table of numerical values used for bar chart visualisation in Figure 6.15.

axes positioned in parallel. The axes can be vertical or horizontal. One example with vertical axes arrangement is shown in Figure 2.10. Each data object is visualised as a line passing through all the axis. The place of intersection with axis corresponds to the data value in the related dimension. This method of data visualisation aims to provide quick comparison between different dimensions of information.

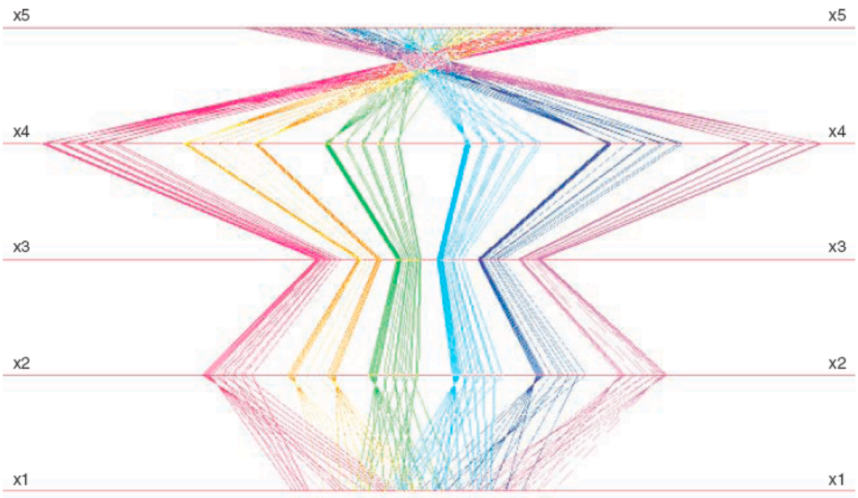

Figure 2.10: Parallel Coordinates.

A general issue with this method lies in the visualisation of a large number of data. In such cases, lines are cluttered and users can not easily distinguish the properties of individual data objects nor correctly identify clusters of correlated data. A variation of this approach is adopted in [94] where axes are arranged on a circle on a two dimensional plane with equal angles between axes (angles can be modified interactively).

Apart from frequently used Parallel Coordinates, other visualisation methods classified in this group are *Prosection Views* [65] (compositions of sections and projections), *Hyperslice* [195] and plots of high dimensional data or *Scatterplots* [5]. *Scatterplots* are visual matrices which show relationships among several variables taken two at a time. Scatterplot matrices can reveal dependencies within the data, cluster sets, and particular outliers. One example of a Scatterplot matrix is given in Figure 2.11.

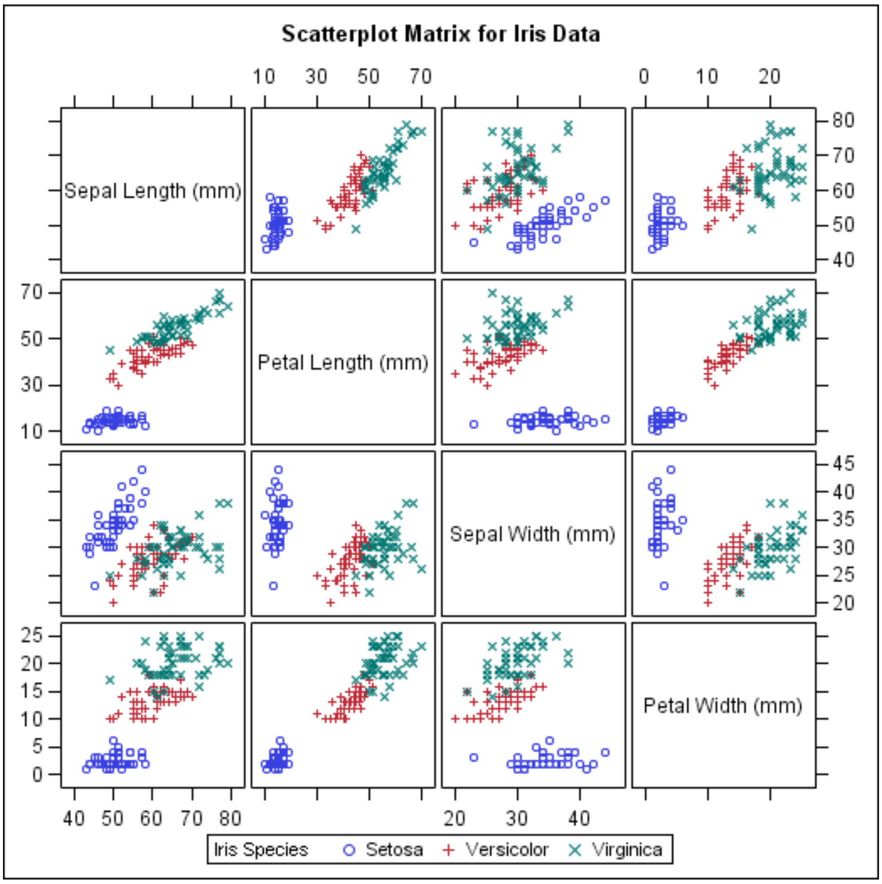

Figure 2.11: Scatterplot matrix.

The stated methods could be used in image visualisation in case the focus of visual representation is the distribution of values through existing dimensions. These can be useful tools for image processing experts and researchers to follow the processes and examine the results of their algorithms. However, for image retrieval important information are images themselves and these methods are not very efficient due to the existence of the *semantic gap*.

Collections of data can be interpreted as structures of information. Such data structures can be *linear* and *non-linear*. A data structure is considered linear if every item is related to its previous and next item (e.g.array, linked list). On the other hand a non-linear data structure is the one in which every item has relationships to many other data items (e.g, n-ary tree). In addition to this, projection techniques can be linear and non-linear. These two types of projections are discussed in further text through the algorithms used in information visualisation.

Linear projection techniques

Due to the fact that the number of dimensions in multi-dimensional data is larger than the number which can be visualised directly, intermediate algorithms are required for adaptation. These algorithms should reduce the number of dimensions and embed the structure detected within the multi-dimensional data-set. Some of the techniques used are *Principle Component Analysis (PCA)* [127] and Exploratory Projection Pursuit (EPP) [62].

PCA is a transformation technique that aims to identify the main factors according to variances in the data set. By visualising these factors, main trends inside the data can be identified and understood. The PCA provides a transformation of the original data space in such a way that the first coordinate of the resulting principle component space would resemble most of the variance in the data set, the second variable most of the remaining variance, and so on.

Sometimes the projection onto the principal components does not show the structure of the data set. Exploratory Projection Pursuit (EPP) is a set of techniques designed to identify structure in high dimension data sets. It is a technique for finding projections which maximise some statistic over the data set.

Non-linear projection techniques

As one of a non-linear dimensionality reduction method, Multi-Dimensional Scaling (MDS) is frequently used method for embedding the relations between multidimensional data objects into a 2D or 3D visualisation space [130], [128], [132] and [208]. The purpose of the MDS is to provide visual representation of the similarity patterns between the data objects. One of the typical examples of MDS application in data visualisation is the case of representing distances between the several US cities as shown in Figure 2.12.

The MDS algorithm minimises the ratio of differences between inter-point distances in the high-dimensional space and in the projected low-dimensional space.

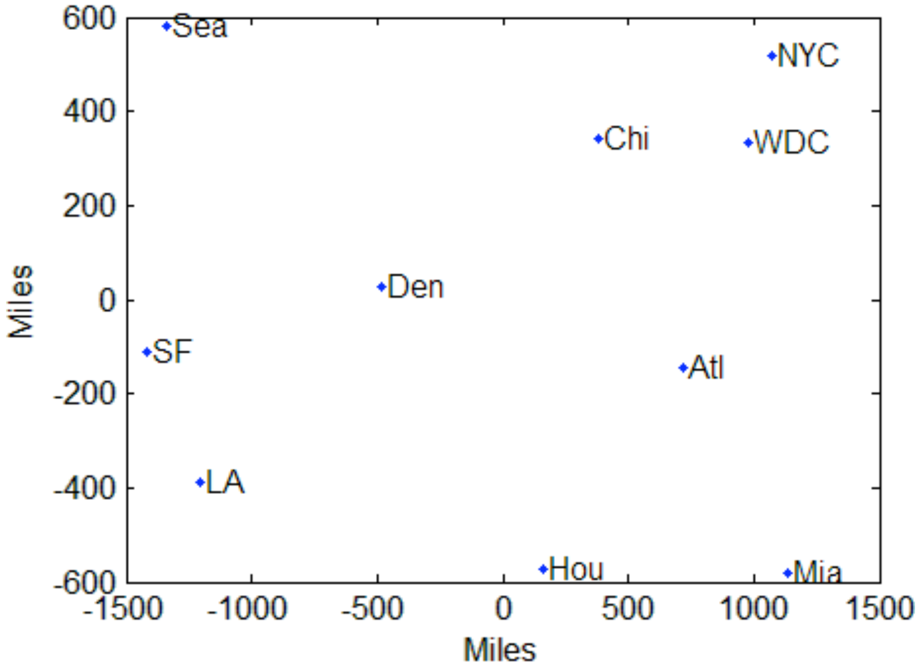

Figure 2.12: Visualisation of distance between US cities using MDS projection algorithm (taken from http://www.mathworks.com).

There are various computational techniques which are used to perform MDS calculations. The first MDS algorithm was introduced in psychology by Torgerson [191] and is nowadays referred to as classical MDS. It produces the layout based on eigenvectors and linear combination of dimensions and has an $O(N^3)$ procedure for generating layouts. N is the number of multi-dimensional data elements. Apart from it's cubic complexity, the algorithm is not scalable. Additional data items can not be added to the layout without re-running the MDS process and re-calculating all the distances. The same property is related to the method proposed by Kruskal [110]. Here the objective is to directly embed the Euclidean distances as spatial distances on the screen typically by minimising a stress function. In [30] an iterative MDS layout algorithm has been proposed with the complexity of $O(N^2)$ in order to reduce the complexity of the MDS layout algorithm and improve the layout scalability. This, so called "basic-spring" approach, performs an iterative displacement of points and calculates the displacement vector for each point to minimise the difference between the original and the embedded distances. In general cases, when spring-based algorithms are used, the relationships between objects are modelled by a spring. The length of the relaxed spring is the distance between two objects in the high-dimensional space. Similar objects will be pulled closer together whereas dissimilar one will be repelled away form each other. The final layout corresponds to the state of system equilibrium. In [131] the authors use the combination of k-means clustering approach (Appendix A) and spring based MDS algorithm. The random set of $\sqrt{N}$ elements is taken from the data-set and it is used as a seed for k-means algorithm. Then the spring algorithm is applied between the K-means centroids. A combination of a clustering algorithm and MDS is used also in [15], where cluster analysis is used for accessing the structural significance of groups of data objects and their layout order. In [208] the authors propose a steerable and progressive MDS algorithm for visualising large set of points. The goal is to perform the MDS region by region where "region" refers to rectangular areas also called *bins*. The developed approach addresses the complexity of MDS when dealing with large number of points by iteratively placing data objects on the screen. This method is developed for the data which is represented as points on the screen, thus it's direct application would not be suitable for image visualisation. However, one potential adaptation in order to apply this approach for image collection visualisation is relating *bins* with image clusters and adapting the approach according to this premise.

Apart from MDS, there are non-linear techniques developed on the basis of PCA and adapted for the representation of non-linear data. There are several works addressing this issue such as generalised PCA [95], kernel PCA [165] and the principal curves [71]. In principal curves the goal of constructing a principal curve is to project the set onto a non-linear manifold.

In addition to the stated methods, there are also methods: *Locally preserving projection* (LPP) algorithm [73], Locally Linear Embedding [152] and *Isomap* [190], a non-linear methods for mapping the high-dimensional data.

Neural networks

Inspired by research on the structures of the brain, artificial neural networks are computer simulations of brain-like systems. The basic units of the artificial neural network are referred to as the *neurons*, following the analogy with the human brain. Each neuron (or node) of the neural network is associated with a combination of weights which determines the systems response to an input stimulus. This way an input x leads to a predictable output y. The weights associated with each node are result of a network *training process*. Training process is usually initialised by assigning random weights to each node (the network state refereed to as network without any knowledge). Further learning process can be supervised or unsupervised depending on the way in which the learning process is conducted. Supervised learning is influenced externally by human or some other source, while in case of unsupervised learning network learns exclusively from the data itself.

The most common neural network used for data visualisation are Self-organising Maps [107], [116] and [211]. Self-Organising Map (SOM) is a biologically inspired method with the ability to map similar or closely related objects into neighbourhood cells of the networks. As a result, displayed information is automatically grouped and abstracted so that the topological order of data features is preserved. A Self Organising Map consists of a regular grid arrangement of M neurons where each neuron is assigned with a k-dimensional weight vector.

$$w_i = [w_{i1}, w_{i2}, ..., n_{ik}], i \in (0, M) \tag{2.2}$$

The dimensionality of the vectors corresponds to the dimensionality of the input data.

The training step consists of taking a k-dimensional data sample from the input set where an input data vector is given as:

$$x = [x_1, x_2, ..., x_k] \tag{2.3}$$

The first training step is finding the best matching node for this data item. The winning node c is the one whose weight vector w_i is most similar to the input vector x.

$$||x - w_c|| = min||x - w_i|| \tag{2.4}$$

The weight vector of the best matching node is adjusted to match the input vector even more, following the rule given by

$$w_i(t+1) = w_i(t) + h_{ci} \cdot (x - w_i(t)) \tag{2.5}$$

The scalar factor $h_{ci}(t)$ is a so called "neighbourhood function" which is usually defined as a Gaussian curve, decreasing from the neighbourhood centre node to the outer limits of the local area given as:

$$h_{ci} = \alpha(t) \cdot exp(-\frac{||r_c - r_i||^2}{2\sigma(t)^2}) \tag{2.6}$$

where $\alpha(t)$ is the scalar value of "learning rate" , r_c and r_i are the locations of units c and i on the map grid, respectively, and $\sigma(t)$ is the radius of the area input vector has influence on (the neighbourhood radius at time t). Since all the nodes of the local neighbourhood are adapted according to the input vector, the resulting arrangements will lead to an implicit clustering of the data. Although the SOM is not primarily a clustering tool the resulting visualisation will show groups of similar items localised in one area. However the classic SOM does not distinguish borders between clusters. Colour encoding of network nodes is one of the visualisation techniques used to distinguish difference between close data elements and to convey additional data information. One such example is displayed in Figure 2.13. Each black dot marks a different map area and colour is used to display the density or the clustering tendency of the documents. Deeper blue areas have the lowest density.

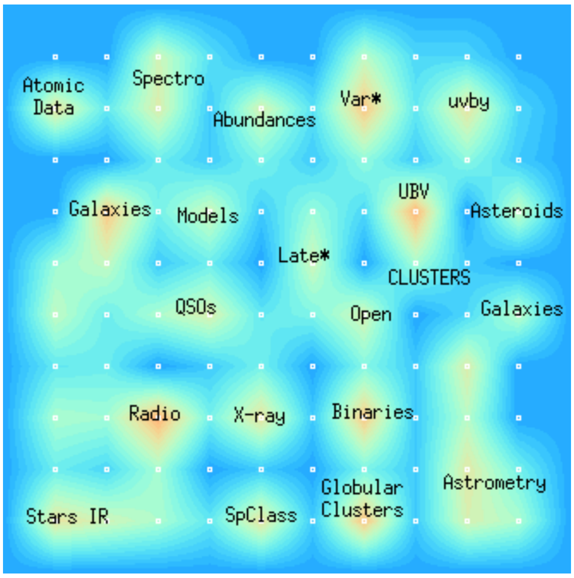

Figure 2.13: Kohonen Self-Organizing Map visualisation of keywords associated to the catalogues (taken from vizier.inasan.ru/vizier/).

SOM's are also used for image visualisation, in order to show similarity between images according to image features (primitive or semantic). One example of image collection visualisation is shown in the Figure 2.14.

Figure 2.14: SOM visualisation of images (taken from http://www.generation5.org).

2.2.3 Iconic displays

Iconic displays use icons, their shapes and colours to represent properties of multidimensional data on the display. Data characteristics or features are mapped onto geometric (shape, size, orientation etc) or non-geometric icon attributes (colour, texture etc). A visualisation solution named *Star-Glyphs* [59] generates an individual icon related to a specific data object. A star glyph is composed of equally spaced radii, as many as the number of attributes in the data table, starting from the glyph centre. The lenght of the radius is proportional to the value of the data attribute where the rightmost radius corresponds to the first attribute and mapping is continued clockwise. This icon technique has two important issues: the number of features it is able to visualise and the difficulty for comparison between two icons separated in space. The example of different data icons are shown in the Figure 2.15. The *Chernoff faces* technique presented in [40] allows conversion of multivariate data to cartoon faces, the features of which are controlled by the variable values as shown in Figure 2.16. The author uses facial features to represent trends in the values of the data, not the specific values themselves. This makes this visualisation not very efficient for precise data analysis. However, it can be a helpful visual tool for determining which sections of the data were of particular interest. The example where faces are described by 10 facial characteristic parameters is shown in Figure 2.17. This means that for data visualisation the following parameters can be used: 1. head eccentricity, 2. eye eccentricity, 3. pupil size, 4. eyebrow slant, 5. nose size, 6. mouth shape, 7. eye spacing, 8. eye size, 9. mouth length, and 10. degree of mouth opening.

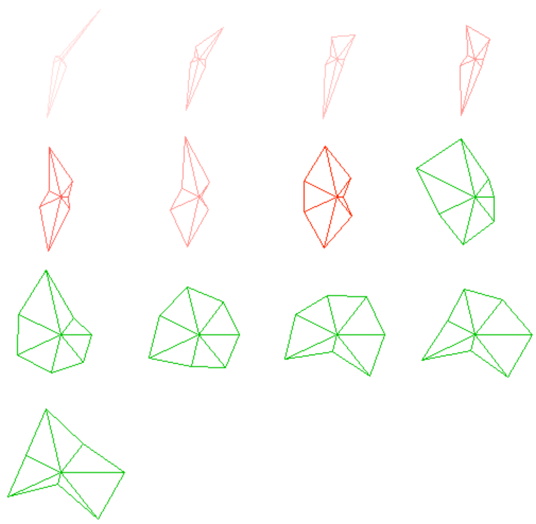

Figure 2.15: Star Glyph visualisation of multi-variate data [205].

Another type of iconic-based visualisation generates textures like in *Stick Icons*

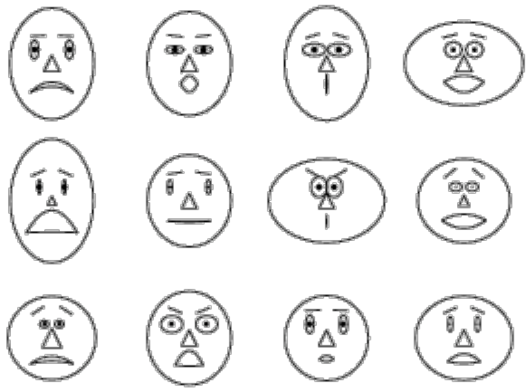

Figure 2.16: Chernoff faces visualisation [38].

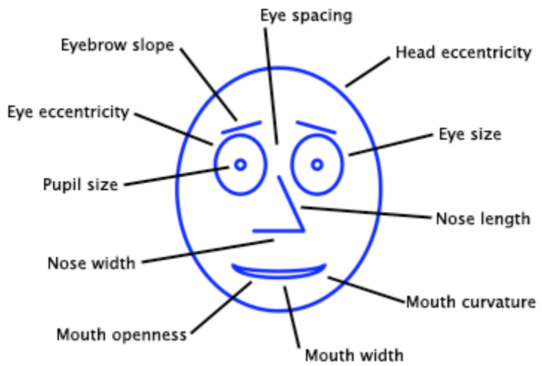

Figure 2.17: Facial parameters used for visualisation of multidimensional data [39].

[143]. In this example, each feature is mapped onto the icon by defining the angle of the *limb* of the icon as shown in the Figure 2.18. Number of dimensions that can be displayed with this example is 4-5.

One of the well known iconic display is InfoCrystal [179] where documents are filtered and visualised according to the selected query criteria. Icons are used for representing both criteria and documents. Icons are placed in-between the criteria icons depending on their relevance and also colour and shape encoded to convey additional information about the documents they represent. A example of InfoCrystal visualisation is shown in the Figure 2.19.

2.2.4 Dense pixel displays

The idea of the dense pixel display techniques is to use all available space on the display for visualising multidimensional data by mapping each data value into one pixel as show in Figure 2.20. The issues related to this group of visualisation approaches as well as their applications are discussed in details in [101]. Due to the use of pixels for data value representation these visualisation techniques allow the display of the largest amount of data. Pixel-based methods partition the display into

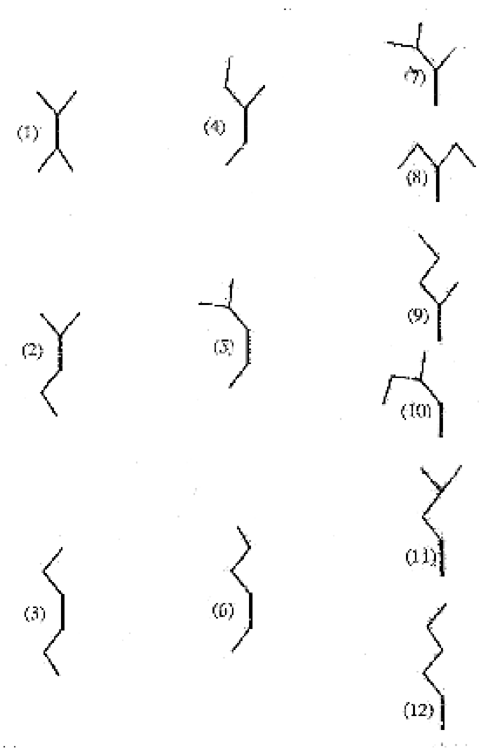

Figure 2.18: Family of Stick Icons with four limbs arranged according to data attributes [143].

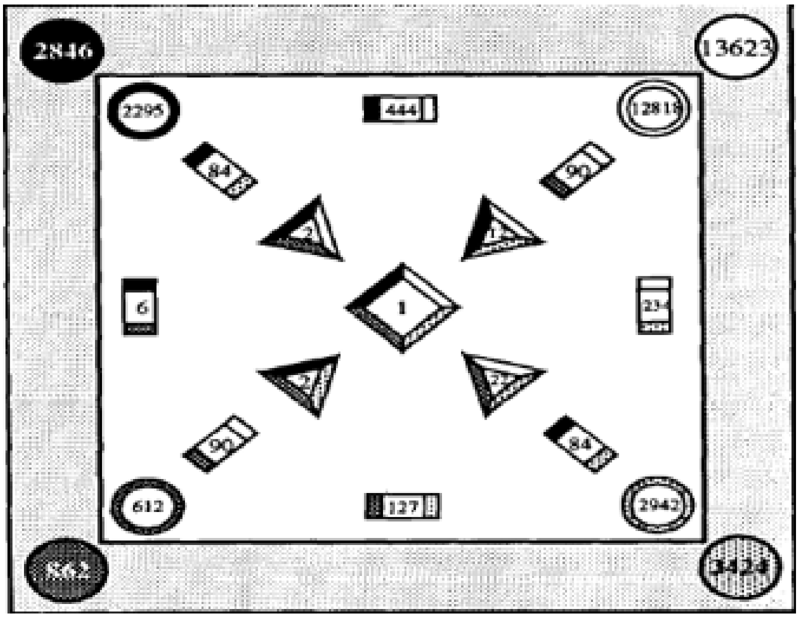

Figure 2.19: Example of InfoCrystal visualisation [179].

sub-windows where each sub-window represents one dimension. Since data values in each dimension are presented by a single pixel, different values are presented by the use of colour. For example green for the lowest and red for the highest data value. Also, shades of grey can be used for the same purpose. The information about the relationships between the dimensions can be detected by observing corresponding regions in sub-windows. One of the issues related to this type of visualisation is the limited number of colours for presenting data values.

Examples of the different pixel-based visualisation methods can be found in [102], [100] and [8].

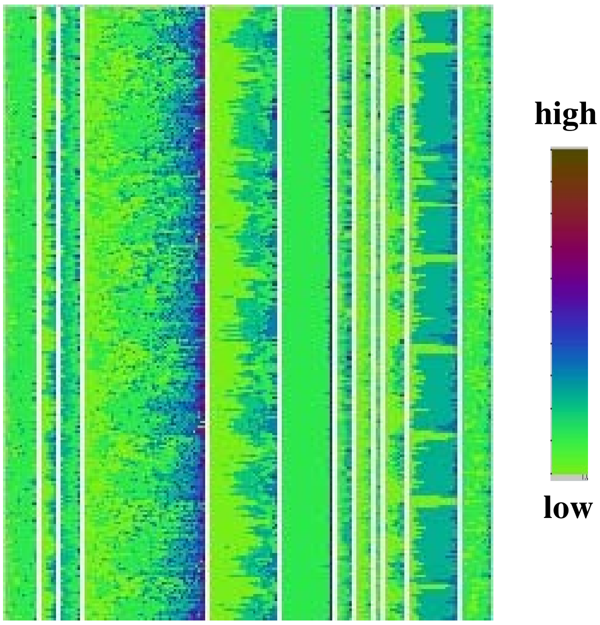

Figure 2.20: Dense pixel visualisation of multi-dimensional data [99].

This method is utilised in image visualisation in the work presented in [212] where images are mapped as pixels in blocks representing various data dimensions (visual features, semantic labels and so on). Each image is represented by one pixel with appropriate colour corresponding to the image value in this dimension. This technique is useful for visualising properties of entire collection throughout dimensions. Just by observing different blocks/dimensions certain patterns can be noted.

2.2.5 Stacked displays

Stacked displays are visualisation solutions which aim at maximising the effective use of the visualisation space by display multidimensional data using their hierarchical properties.

One set of solutions perform an initial partitioning of the entire visualisation space into separate areas. The next step is populating areas with information allocated to that area. For example, one area can be used for presentation of one image cluster. The best known example of such visualisations is the *TreeMap* [89]. *TreeMaps* partition the visualisation space according to the weight value of corresponding node in the hierarchy. The weight value of one node can be related to the number of elements in the node or some other visualisation criteria. The TreeMap visualisation example is displayed in Figure 2.21. As opposed to rectangular areas in *TreeMaps*, employing *Voronoi Treemaps* for space partitioning [13] allow creation of areas with various shapes. One such visualisation example is shown in the Figure 2.22.

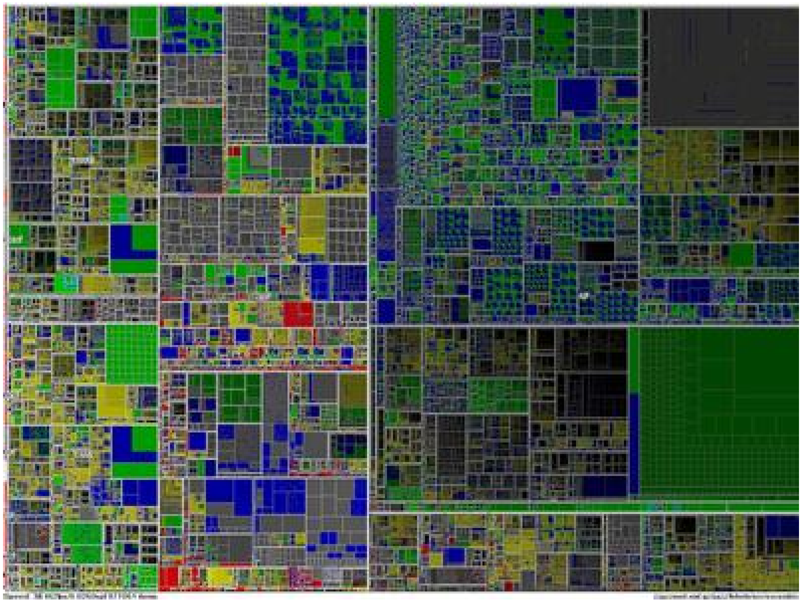

Figure 2.21: Visualisation of one million data items using TreeMaps [188].

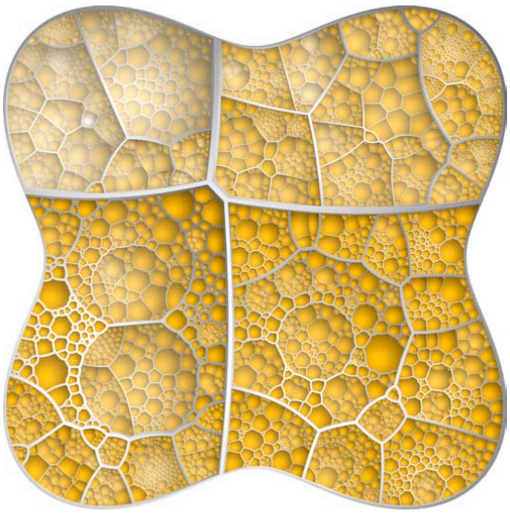

Figure 2.22: Voronoi Treemap visualisation of 4075 nodes at 10 hierarchy levels [13].

ConeTrees are another example of stacked displays [150] where 3D visualisation is used for displaying hierarchically organised data. Although *ConeTrees* do not partition the display space the aim is also to use most of it. One example of *ConeTree* visualisation is given in Figure 2.23.

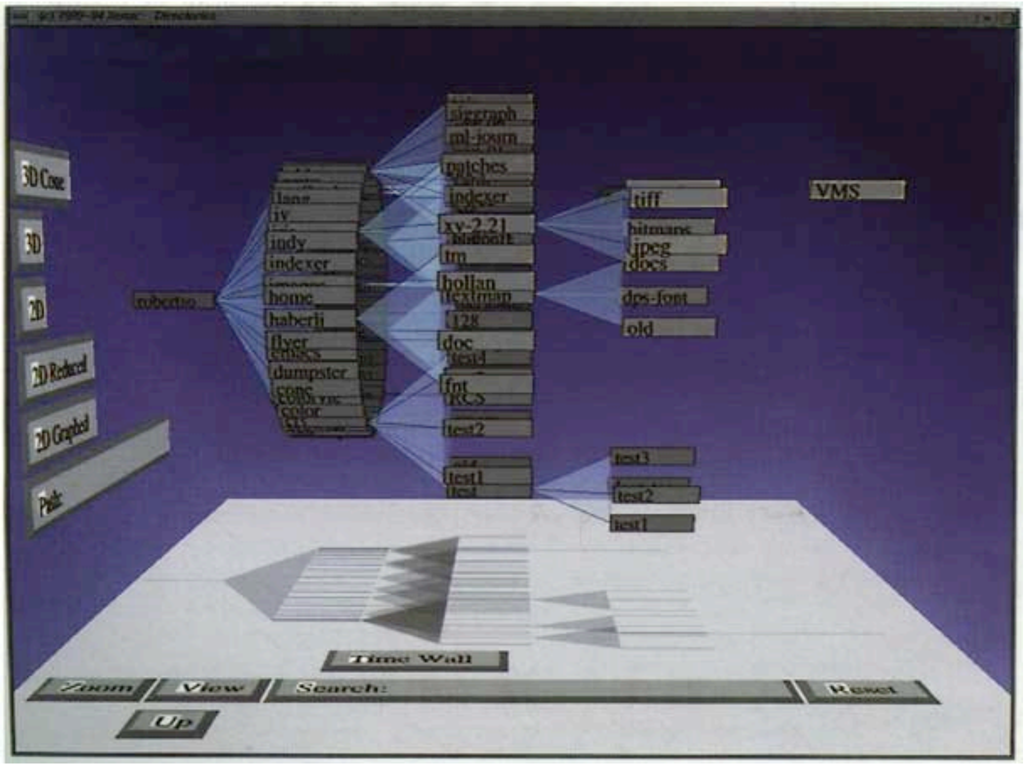

Figure 2.23: Visualising hierarchy using *ConeTree* (taken from infovis.net).

2.3 Interaction and distortion techniques

As stated in [98], for effective data exploration there are cases when visualisation techniques have to be combined with appropriate interaction methods. Generally speaking, interaction methods allow the user to actively interact and change the visualisation in order to accomplish his/her task. Interaction techniques refer to any way of interaction from simple clicking on the interface button, reaction to mouse movements up to more complex actions such as visualisation view changing and so on. Distortion, as one of the complex methods, alternates the original shape of visual representation or data objects. It is used in information visualisation to allow the user to focus on the details while the global overview is preserved.

In [98] the author makes a classification of the interactive visualisation techniques into the following types:

1. Dynamic projections;
2. Interactive filtering;
3. Interactive zooming;
4. Interactive distortion;
5. Interactive linking and brushing.

2.3.1 Dynamic projection

For exploring the data space where data has a huge number of dimensions, sometimes changing data projections can be useful for the user. Instead of visualising only two or three dimensions, or performing the dimensionality reduction the system can support the user in moving through a sequence of projections as in Grand Tour [11]. Another example of dynamic projection techniques can be found in *gGobi* [185] a system evolved from well known *XGobi* which also employed dynamic projection methods [184]. The main issue of dynamic projection approaches is the case of large numbers of data dimensions which are then difficult to trace [98].

2.3.2 Interactive filtering

When dealing with data-sets containing large numbers of data items, it is difficult to examine everything at once. For this reason, interactive filtering techniques such as *browsing* and *query*, interactively partition the data-set into sub-sets and enable focusing or details requesting on these sub-sets. Browsing involves exploration and direct selection of the wanted subset. In other words it is "a scanning activity that allows users to build a cognitive map of a domain" [177]. With query, user specifies properties of the subset to be extracted. The use of these methods depends on two things: the user's ability to specify his/her information needs within the specific data domain and the type of support provided by the the application (or system).

Browsing is a very useful tool in image retrieval systems especially in the initial stage where user has no domain knowledge. It is an activity which assists content exploration and is useful in cases where the user does not know how how to specifically express his/her information needs in the given environment. The extensive survey of browsing models in image retrieval is given in [75]. From an application perspective, browsing is usually supported by displaying the images themselves in which case user is aware of the visual content of images.

Query is a formal statement of a user's information needs that is matched with the content and information from the database. They are supported either by keyword search without having any contact with the actual images (query-by-text), choosing one semantic category which is relevant to the task (query-by-concept) and/or by selecting one image from the displayed set (query-by-content). Queries which use browsing as the initial method for finding the relevant content, which is then used as query example (so called query-by-example) are utilised for both content and concept-based queries.

On the other hand, there are visual-based interaction tools such as *MagicLenses* which perform filtering of data. MagicLenses change the way data is visualised and depending on their property, they can provide different data aspect. One MagicLense is used to filter data with specific properties. For example, one potential application for image retrieval is using Magic Lenses for connecting semantic and content-based retrieval by filtering semantic query results by their low-level properties. If the system retrieves a result of a query *car* a MagicLense could be used for filtering or emphasising red cars. It is also possible to combine several filters/lenses which overlap. In that case every filter will be applied and the integral result shown.

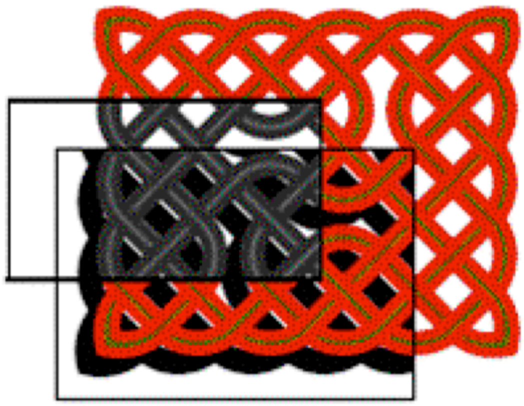

Figure 2.24: Magic Lense: the lense visually emphasises part of the display area.

2.3.3 Interactive zooming

Zooming is a widely adopted interaction technique for providing details of the requested data. When user initialises query or browsing, the first view in most applications is a general view (or an overview) of the whole data-set. This view is important for the user to gain knowledge of the content and detect patterns and correlations within the data set. Since the overview is usually a very compact presentation of the data-set, it can be difficult for the user to find the relevant data at a glance. Zooming enables the user to spatially enlarge the data objects or regions, thus supporting the learning of details. One of the standard examples for a zoomable interface is the one presented in [17]. Another commonly used application, which supports one form of zooming, is "Adobe Acrobat Reader" where zooming of the document pages is available using the overview window functionality. Treemaps [19] can also be combined with zooming functionality. Authors in [44] discuss the zoomable interfaces and their critical issues in more details.

2.3.4 Interactive distortion

Distortion techniques aim at presenting the global overview and local focus on the screen at the same time. It provides information about the whole structure of a data set and simultaneously supports user focus on a smaller area of data. Interactive distortion techniques are most commonly referred to as "focus + context" techniques and one of the first comprehensive reviews on these techniques was provided in [115].

One of the first application of visual distortion is for Bifocal Display [178] where data objects not in focus are compressed uniformly in the horizontal direction (in case of one dimension bifocal display) or in the horizontal and vertical direction (in case of two dimensional bifocal display).

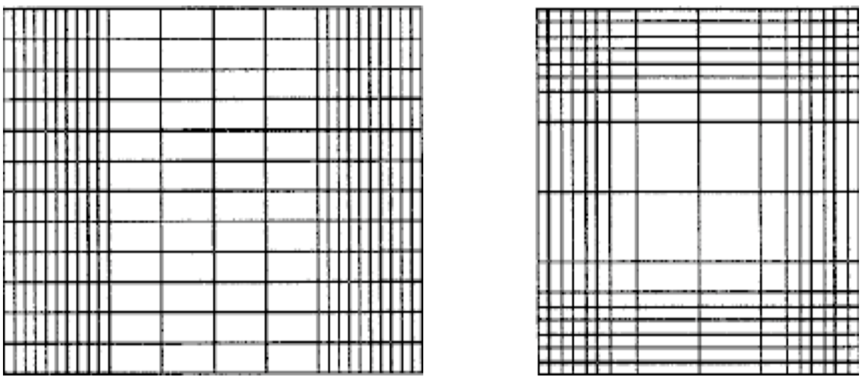

Figure 2.25: Bifocal display: distortion in one and two dimensions

Another, probably more frequently used method in information visualisation is Fisheye View [64]. In this method, the more relevant information is less distorted than the less relevant information. Two issues are addressed with this approach: reducing the time for accessing a large structure and leaving out details the farther away from the focus point. One example of distortion application on a tube map is presented in the Figure 2.26.

In work presented in [64], distortion of data items is accomplished using formula for degree of interest (DOI)

$$DOI(x|y) = API(x) - D(x, y) \tag{2.7}$$

where x is the data element, y is the current focus element, $API(x)$ is the *a priori* interest in object x, and $D(x, y)$ is distance between x and y (spatial or semantic).

In [159], authors extend the fisheye distortion to graphs and maps using Euclidean distances between graph-vertices and map coordinates, respectively. In [18] a compact presentation of a calendar view is supported by interactive distortion for viewing details per day. Another application of this interaction technique was pro-

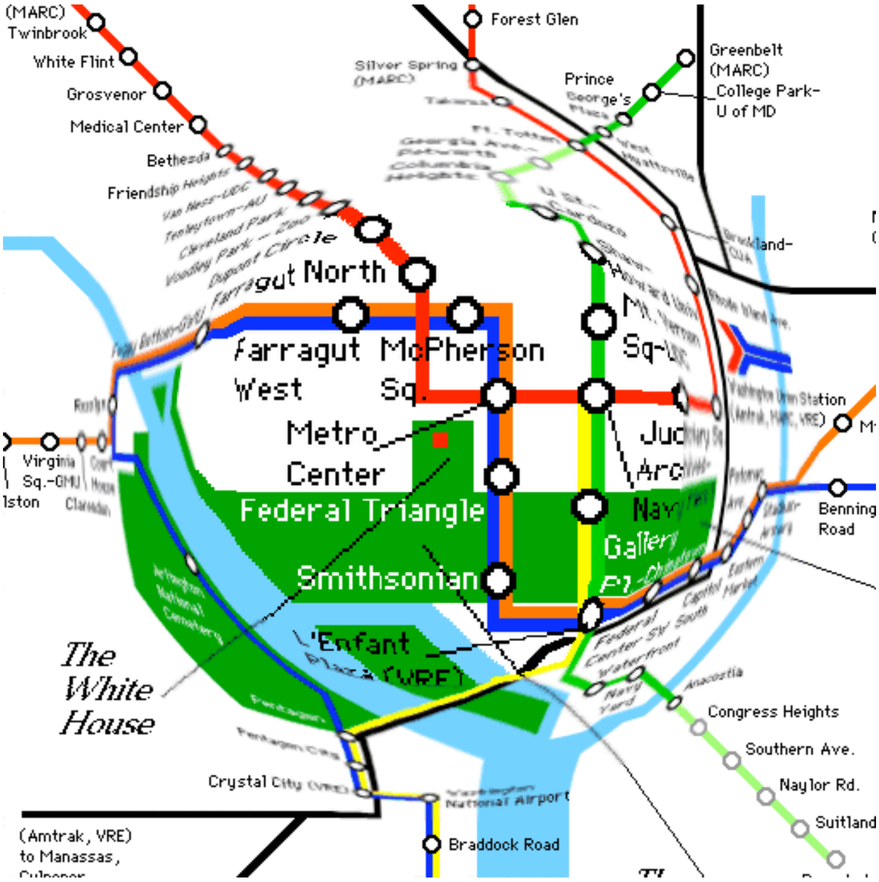

Figure 2.26: Fisheye distortion of the Washington tube map [126]

posed in [114], where distortion is used for focusing on parts of multi-level hierarchy which is mapped on a hyperbolic plane. One graph distortion application can be found in [139].

2.3.5 Interactive linking and brushing

Brushing is an interactive technique used for highlighting or sometimes deleting a subset of data in multi-perspective data visualisation spaces. It is commonly used together with a linking method [185] and [184], so the highlighted area or a subset is also highlighted in every linked visualisation space. One of the examples is linking and brushing applied to a scatter plot matrix. When user interacts with one part of the matrix, the selected data is modified, usually by changing colour, and this reflects in all the cells containing the same data. This way the user can observe different data properties in the same visualisation view.

2.4 Evaluation of visualisation approaches

Evaluation of information visualisation solutions is a very important research topic [145], [124], [91] where very little work was done before 1994 and huge efforts have been invested after 2005. To emphasise this fact, in [91], the authors show a diagram of the retrieved documents by Google Scholar related to the keywords "information visualisation" and "usability" for the period until year 2005. On the other hand, new visualisation solutions are constantly reported but not always accompanied by evaluation of their performance and/or usability reports. In around 60% of the papers proposing a new visualisation technique (or an adaptation of an old one) there is no mention of the evaluation process [55].

The main reason for the previously stated situation is the fact that evaluation in information visualisation is not a straightforward task. There is a number of issues related to various aspects such as which evaluation method to use, how to formulate the tasks, how to retrieve useful feedback from the users and others, which have been actively discussed in [145], [108], [55].

Some say that a visualisation solution has to be judged in the context of the task it aims to assist. If visualisation is considered as a collection of methods, techniques and tools applied to satisfy a certain need, it can be evaluated in terms of *effectiveness* and *efficiency* [196]. Effectiveness by definition is a measure of "success or achievement of the results that you want" whereas efficiency is the "property of using resources well without wasting it "[147]. In other words, in information visualisation evaluation, effectiveness refers to quality of proposed solution/application and efficiency refers to quantity or speed.

The goal of visualisation is to provide insight into content of the data-set, organisational patterns and so on. Using visual display of information users can easily see things which would hardly be noticed in tabular forms or by examining element per element in a image collection. However, evaluating the efficiency of information in terms of information gathered by the user is not simple. The level of acquired information about the data-set cannot be accessed in a straightforward way. It can be estimated by observing the users actions and quantifying the consequence of wrong decision [196].

In [145], the author classifies evaluation strategies into four standard methods:

1. Controlled experiments comparing design elements. This type of evaluation study compares specific interface components [2] or the way information is mapped onto a graphical display [84];

2. Usability evaluation of the tool. This type of evaluation provides users with feedback on the tool, in any intermediate development stage or for the final

interface [183]. They help developers and researchers to discover problems or improve aspects which were initially overlooked, but actually important for the users.

3. Controlled experiments comparing two or more tools. These techniques are used to compare a new tool against state-of-the art solutions or to compare performance of existing ones [105]. It is applicable in cases where the same type of solutions already exist (which is not always the case).

4. Case studies of tools in realistic settings. This type of evaluation involves *professional users* in a professional environment, performing real tasks using the given tool. These tests provide realistic effectiveness and usefulness assessment of the tool. According to [145] these are the least common types of studies. Such evaluation has been performed in [193], where the authors discuss the use of complex visualisation solutions by expert meteorological forecasters.

On the other hand, at Information Visualisation Wiki [201], the classification of evaluation visualisation techniques identifies two main categories and related subcategories:

1. Analytic evaluation or an expert analysis based on formal analysis models. This type of evaluation is performed not by the users but by the people specialised in evaluation of such solutions. It includes:

   - Heuristic/expert evaluation, where experts act as less experienced users and describe the potential problems they foresee for real users [192]. Experts are given the software, typical user profile and tasks. Based on this they test the application and provide report with details of advantages and disadvantages and problems of the software.
   - Cognitive walk-through, where experts performs a specific task using a prototype.

2. Empirical evaluation, based on the user tests. They can be further distinguished between quantitative and qualitative studies:

   - Quantitative studies, which consist of an analysis of hypotheses tested through direct measurements usually conducted by controlled experiment. Controlled experiment is a type of experiment in which an observer tests a hypothesis by studying the effects brought on by alterations to a certain parameter.

- Qualitative studies, involves the analysis of questionnaires, interviews and observing users behaviour and decisions while using the system. It can be done by creating a focus group for discussing the specific part of research or by interviewing the users to understand their impressions about the evaluated tool.

In practise, most visualisation solutions are evaluated by non-expert users [2], [84], [183], [105], [193], [91]. When talking about non-experts in evaluation two types of users can be identified: *professional users* who are not experts in visualisation or the interface domain, which test the solutions in the professional environments performing real tasks. The second type of users are also referred to *end-users* (i.e. in some cases *home-users*). In visualisation evaluation, most of the tests are performed with home users who are usually students or laboratory colleagues, since reaching professional users at their work is mostly unfeasible.

Nevertheless, laboratory based tests give valuable initial usability results. They help remove most of the interface issues and development bugs; provide experience regarding the properties of the tool, and help discovering elements that were overlooked during development.

Generation of tasks on which the evaluation is performed depends on the purpose of the system and it's users. They have to be simple, since most of the users will use the tool for the first time and the time for accomplishing the task is relatively short. For example, within TRECVID Retrieval Evaluation [187] the evaluation time was 15 minutes before year 2008 and 10 minutes starting from 2008. However, user tasks should be selected in such way that they range from less difficult to more difficult. It is noticed that users appreciate when they are given the least difficult one first, since it helps them to relax and approach the new system with positive attitude. If they are given difficult tasks at once, they could become frustrated with the tool, thus immediately discarding the optimistic approach for evaluation (which can influence the results).

During the evaluation there are two ways of compiling and reporting performance results. One is to report the overall performance for a combined set of tasks given during evaluation. The second approach is to distinguish the results per task which can help distinguish the situations in which the system performs better from situations where it under-performs.

During the evaluation procedure leaving users to explore the data freely, outside the task, can be very significant for the evaluation and further development of a proposed solution [145]. If users are motivated they can point out some properties of the tool which were missed by the developers. In fact, this happened while testing the hierarchical visual browser described in Chapter 5 of this thesis. Apart from using standard interaction and navigation methods explained by developer for finding one image of a plane, users discovered additional navigation method which enabled finding more than one image of a plane (which was beyond the task description).

One of the standard outputs of user tests is the time needed for accomplishing the given task [145]. For providing access to a broader group of users, interfaces are usually hosted on servers and accessed through Internet. Depending on the available connections it can influence the interaction process in terms of speed. Interfaces are supposed to provide fast exploration support, but their speed does not always depend on the developed application. Graphical support depends heavily on the computers users have at their work places or homes. More detailed usability issues are addressed in [145].

Comparing prototype tools with state-of-the-art can be difficult due to the different data-sets used and their public un-availability. This is particularly true for interfaces developed for image retrieval. For this reason a huge number of researchers use generic data-sets such as Corel [45] or TRECVID [186] set. In order to create benchmark data-set to help developers and researchers compare their ideas and solutions, an information visualisation contest has been organised under the name *InfoVis Contest* (web link for 2009 competition can be found at [80]).

As previously mentioned, another problem in the evaluation procedure is that users do not spend a lot of time using the system. User tests are usually limited in time and the initial tutorial phase is often not enough for users to understand the properties and ideas of the tool.

Research methodologies have been proposed for unifying evaluation of information visualisation tools [169], [23]. The authors emphasise the importance of long-term user involvement with the tools as well as their engagement in the development process. However in realistic cases this is rarely feasible.

One of the issues in visualisation evaluation is the number of users involved in user testing. It is considered that it has to be large enough to "account for inter-subject and inter-topic differences" [198]. An extensive study on number of people and the types of tasks used in user evaluation is given in [91]. The authors also provide information regarding the sizes of test collections, which vary from 100 up to more than 14000 items.

As stated at the beginning of this section, despite all efforts to establish criteria for evaluation procedures such as: which methods should be applied, number of participants and data sets to be used, this goal is still far from being accomplished. The spectra of developed visualisation solutions and the particularities of information they try to display graphically is getting wider and wider very fast, which increases the difficulty of recognising appropriate evaluation methods. Problems of finding people or paying people[1] to perform user tests prevents the possibility of having a constant number of participants in evaluation phase. In addition to already stated issues copyrights of used data-sets do not allow utilisation of the same data for comparing visualisation solutions. For all previously given reasons most existing evaluation procedures are adapted according to individual capabilities and circumstances.

In this chapter, existing methods and issues in information visualisation have been discussed. Since the work performed within this thesis concerns the application of visualisation approaches for image management and retrieval, evaluation of proposed solution utilised some of the discussed evaluation methods. In other words, overview of the techniques presented here provides basis for evaluation procedures adopted in this thesis and given with related proposed solutions Chapters 4, 5 and 6.

[1] Some authors pay for the user test e.g. [183]

Chapter 3

Related work

The evident trend in a large number of image management and retrieval systems is strong focus on improving the background algorithms without considering smart ways of involving the human cognition on the higher level. One of the promising directions is believed to be "information visualisation", as a user-oriented approach for conveying various types of information.

Although some image management systems integrate information visualisation tools and methods, the full potential of this combination is not exploited yet. The requirements for development of an efficient and effective system are still more investigated from the machine perspective and than the user's one.

It is important however, to define the task of visualisation and user-system interaction as user-oriented retrieval aids. Visualisation and interaction methods used for retrieval, should be designed according to task user needs to accomplish but also match the content characteristics and the way content is indexed. For this reason in Section 3.1 different types of user tasks are analysed as well as search methods performed for the task accomplishment.

In order to investigate on what level information visualisation is currently involved within image management and retrieval systems, an investigation of existing systems has been performed. As a result, the related report has been compiled and presented in this chapter.

Starting from simple, grid-based solutions, the methods for generating a graphical layout of images are given in Section 3.2. Various grid-based solutions found in image systems are reported in Section 3.2.1. Evolving from the simple lattice, more informative visual representations have been developed to convey the relations between the content of a collection. Similarity based visualisation is investigated as the most common way of presenting the similarities or dissimilarities between the objects in a data-set irrespective of their nature. Existing image systems adopting this visualisation approach are given in Section 3.2.2. Examples of hybrid visual solutions which combine several different visualisation methods are given in Section 3.2.3

The absence of efficient methods for content access and exploration makes this content useless (if user is not able to retrieve it). For this reason, image retrieval and browsing systems employ various interaction and access methods which are discussed in Section 3.3. This is considered as a particularly important aspect of an user-oriented system since providing information visually can simplify and speed-up user access to information. Among the various solutions presented, attention is given to solutions for accessing and interactively exploring image hierarchies.

3.1 Natures of user tasks in image retrieval

In [148] authors define image retrieval "as an information need clarification process consisting of task description, needs expression both in verbal and visual terms, and exploratory search for targets and their alternatives".

Following such definition the first, basic level in image retrieval is the information seeking goal. In image retrieval we can make a distinction between a very specific and well defined goal, when the user is looking for a specific image knowing the content of the database, and the exploratory intention, when the user has a vague search idea and/or not an adequate knowledge of the content

On the second level we have the user activities that need to be performed in order to achieve the goal. These actions fall into the so called information seeking part which can be defined as a "fundamental human activity in the process of gathering information and building knowledge" [22].

There is a significant research performed trying to define the models of information seeking [122], [41], [16]. In [122] author defines three types of phases in information seeking: lookup, learn and investigate. The same author recognises two distinctive retrieval strategies: directed and exploratory search, first being more related to lookup, and the later "especially pertinent to the learn and investigate activities".

In the situation in which the users can describe their image needs abstractly, and have a good knowledge of the database contents, the **directed search** method is natural and effective [148]. In such case there is a clearly defined goal (what is being looked for) and what is expected to be returned within the result set.

The known "item search" in TRECVID [186] is defined as a search task that "models the situation in which someone knows of a video, has seen it before, believes it is contained in a collection, but doesn't know where to look". Even if it reefers to videos the same stands for image retrieval. Example of the TRECVID 2010 user task is: Òfind the video of sun setting into the clouds". The user might use a keyword such as "su'n' or "clouds" in order to execute the search.

Directed search can provide very good results, depending on the performance and features of the system, as well as the search method used. As stated in [41], considering some other tasks such as a need to report on "typical people activities on a sunny day in a rural areas", directed search might not be the best way to retrieve the content.

As defined in [122] "lookup tasks are suited to analytical search strategies that begin with carefully specified queries and yield precise results with minimal need for result set examination and item comparison".

Apart from the initial requirements for the user task, directed search has to be adequately supported by the system. The queries that user request from the system have to return meaningful results which, in case of image collections, is not a simple task, due to the well known semantic gap.

Another problem is that true end users of image collections, not the multimedia experts, often do not have full understanding of what dimensions are available for searching.

All this indicates that there is a need for alternative search support which can be used in three situations [207]:

- when the user does not have the knowledge or contextual awareness to formulate queries or navigate complex information spaces;
- when the search task requires browsing and exploration of the content;
- when indexing of available information is inadequate.

Thus, an alternative or complementary search method to direct search is the so called **exploratory search**, defined in [122], which is a combination of query and browsing. The exploratory search process consists of multiple iterations between the user and the system during which user gathers knowledge about the system and the content. The results of each iteration are dynamicaly evaluated and user is able to adapt his/her strategy according to this evaluation and new gathered knowledge. This type of search is more flexible, allowing user to refine his information needs during the retrieval process.

The benefit of exploratory search is explained in [148] where it is stated that: "serendipitous findings are thought provoking and educational. For example, the user has never thought of using an image of Einstein to represent intelligence. However, quite often users got false hits because the image collectionÕs taxonomy is not externalized into visual structures to allow accurate formulation of the keyword query".

Authors in [148] consider exploratory search better for common users in contrary to directed search which is more suited to professional users whose needs for images can be more precisely formulated in their queries.

Directed search systems focused more on the system core: content classification, concept detection and so on. The interface for such systems are usually not demanding in terms of visual features and interactive support.

On the other hand, exploratory search requests more from the interface since user needs to be able to interact with the system more than just writing the search keyword. There are various information visualisation solutions which support exploratory search. Interactive search is strongly determined by how information is structured and by the interaction style presented in the user interface [29]. Overview techniques have been used in some document retrieval systems (*Starfield* [3] and *FilmFinder* [4]), text retrieval (*Tilebar* [74]), and visualisation of a collection of images (*DynamicTimeline* [113]). These techniques proved efficient in helping users navigate in a complex information space, orient themselves, zoom in on interesting details, and select the targeted items. In addition to this, overview techniques allow users to easily locate, compare and evaluate data items. There are other visualisation methods that support image exploratory search. In [213] authors propose an interface which allows users to navigate explicitly along conceptual dimensions that describe the images. Visualisation of the content structure is another method to support exploratory search [112] [109].

Considering the complexity of the search and retrieval process our conclusion is that retrieval should combine directed and exploratory search in order to provide flexibility of retrieval and support various types of users and their tasks.

3.2 Graphical layouts of image collections

For a significant period of time, most image retrieval systems ignored the fact that providing information about the data in meaningful ways can improve system performance. This would come from the fact that visualising more information for the user would increase the human involvement in the process for which that particular system is used. However, most such systems employed simple visual representations, mostly grid-based, presenting images without any additional info. This section provides examples of various image visualisation solutions, from simple to more sophisticated ones.

3.2.1 Grid base visualisation

A grid, as a visual representation, is mostly used in two cases: for fast (apparently random) displaying of images [199], [203] or for showing a ranked list of images [167]. In the first case, a grid is a simple method to allocate portions of the display space for placing images without visual structure. In the second case, it is used to display a ranked set of images relevant to the specific user query. In such applications, a top left corner contains image with highest similarity to the query, whereas the bottom right is most dissimilar. This type of interface is the most common in image retrieval systems.

In [24], [37] and [109] the authors use grid based visualisation for displaying a preview of the content organised by hierarchical clustering algorithms. As an example, the screen-shot of the interface proposed in [24] is shown in Figure 3.1. A grid of images is one layer of hierarchical structure. Initially, it is a top hierarchical layer where large image is the first image of the grid (representing one cluster). It can be seen from the figure that the user has no information regarding the structure, how many levels there are in the hierarchy, how many images in each node and so on. Such information could provide useful support to the user.

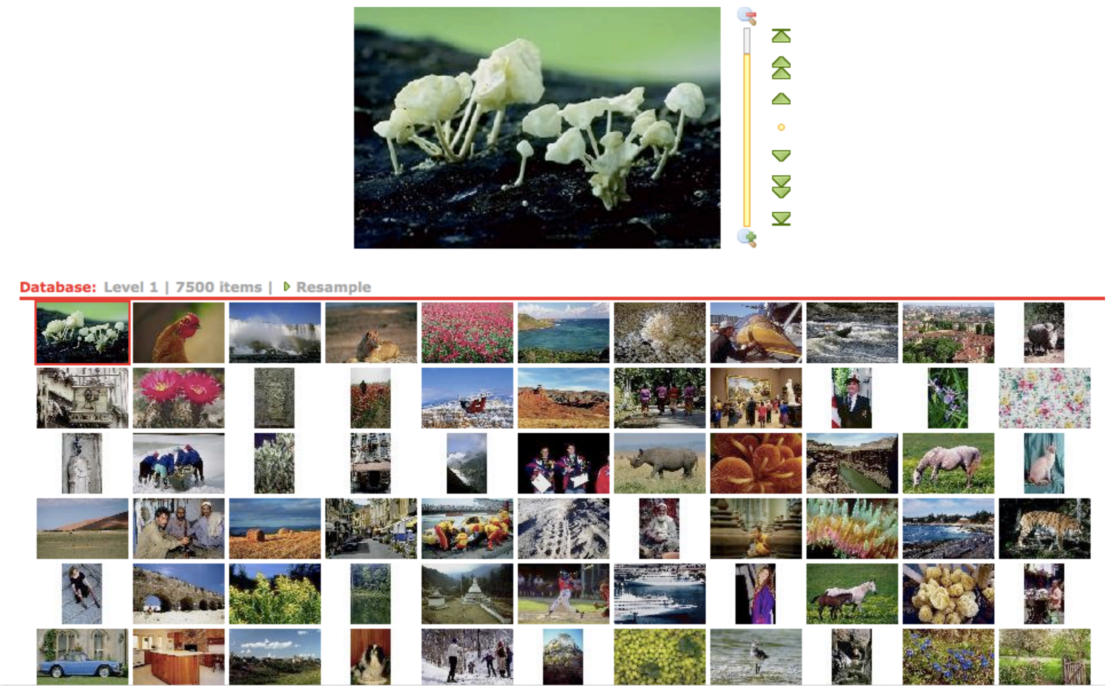

Figure 3.1: Screen shot of the Navidgator interface for navigation through the content hierarchy [24].

The grid has been also employed for visualising hierarchical structure where semantic information about image content is available. In [57] using hierarchical tree structure of a classifier, authors obtain semantic labels on several levels. Similar key-frames are then clustered together and one level visualised in a grid.

The authors in [214] propose similarity based grid visualisation for image placement within the grid where different sections of grid show images retrieved as relevant according to different criteria (specific semantic category, specific low level feature and so on). In [17] a novel visualisation approach was used for browsing personal image collections. The author implements *Quantum Treemaps* and *Bubblemaps* as algorithms for creating the graphical layout. Several grids displaying different image groups are placed together in order to use all available visualisation space. This was achieved by partitioning the graphical display according to the number of existing groups, and later by populating all the available space within each area. The number of areas is equal to the number of pre-calculated image clusters in the collection. The screen-shot of this solution is shown in Figure 3.2.

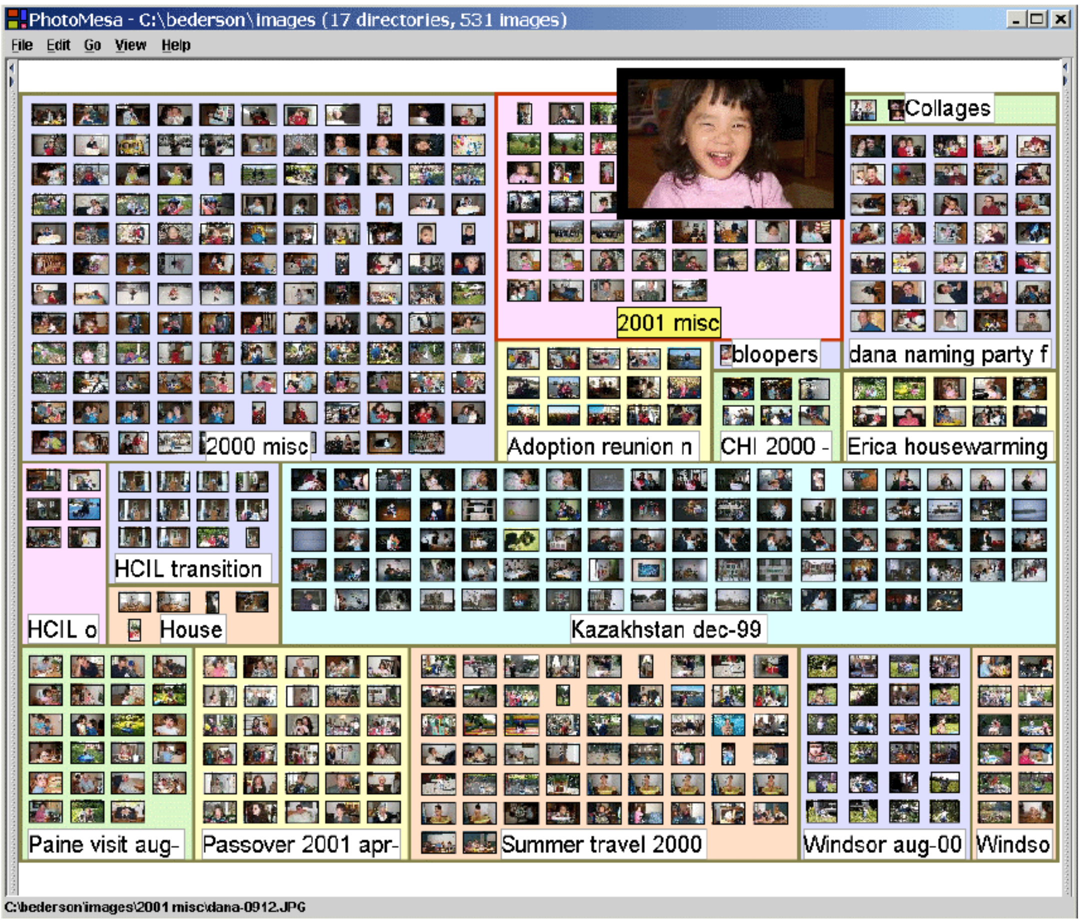

Figure 3.2: Treemaps in PhotoMesa image browsing system [17].

The well known *Mediamill* image/video retrieval system [210], consists of four interactive visualisation solutions that aim to present various perspectives of available information. Three of the four system interfaces have a grid-base visual display expanded to a 3-dimensional space that is achieved by mapping a grid on a sphere. The *Concept Browser* presents the grid of the concept hierarchy, where the user needs to select concepts and descend into the hierarchy. *Cross Browser* visualises one concept thread along the temporal axe. The *Sphere Browser* visualises the temporal dimension on the horizontal axis and related semantic threads of each image (video shot) on the vertical axis. The *Rotor Browser* presented in [175] is another advanced browsing solution which enables the user to see several semantically connected threads of one video shot. The visualisation is again achieved by mapping the grid on a sphere. The *Cross* and *Rotor* browsers from Mediamill system are shown in the Figure 3.3.

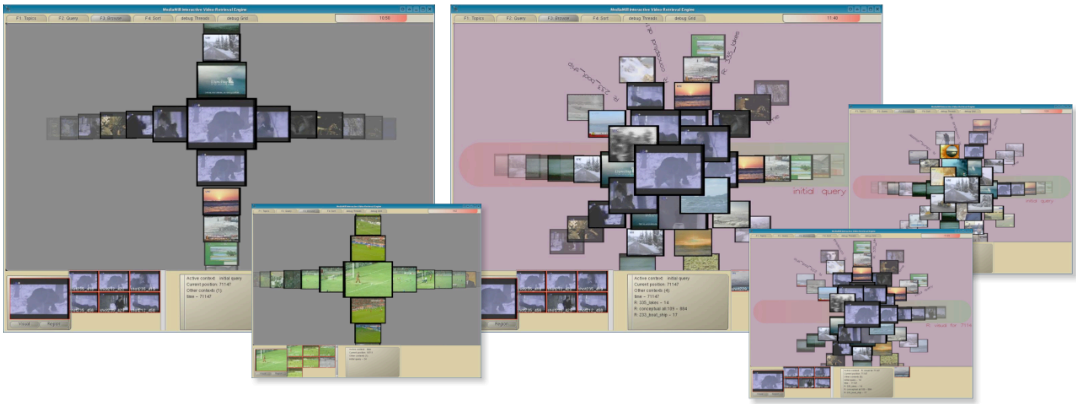

Figure 3.3: A screen shot of Cross and Rotor browser [172].

Although the idea of the grid is to provide a simple visualisation without overlapping displayed objects, very little information about the collection or the content can be displayed. For this reason the user in most cases does not gain any additional knowledge about the content from the grid itself. As a consequence, much time can be lost in navigating through the collection if the visualisation is grid based. Instead, following the same organisational idea behind the use of pattern recognition algorithms, content can be "visually organised" according to some feature. In order to provide more information about the content, especially in the form of the mutual relationships, instead of grid, visual solutions which convey relational content information have been employed.

3.2.2 Similarity based visualisation

As stated previously, a grid can convey a confined amount of information and as such it is not always applicable, especially in cases where more information needs to be presented. For example, one important type of information about the image data-set is the structure which can be presented by displaying relations between the elements of the data-set. They can be single level or multi-level (hierarchical). In addition to this, the hierarchy can be observed as a structure in two dimensions where one dimension is the data relationships on one hierarchical level and the second are parent-child relationships. The relations between data objects or images, are in most cases represented by similarities or dissimilarities between them. Similarity between images can be obtained by comparing features extracted from images directly, using text descriptions or semantic annotations.

One of the earliest visualisation ideas for displaying relations between data objects was the approach presented in [85], where k features were extracted for each data item and the item was mapped into a k-dimensional space accordingly. Since k-dimensional space cannot be directly mapped into a 2-dimensional display, these high-dimensional features had to be transformed in an appropriate presentation form. Different transformation methods for displaying such high-dimensional data are discussed previously in Chapter 2.

From the visualisation perspective, the intermediate or transitive solution between the grid and similarity-based visualisation approaches (which use spatial distribution of objects), is the use of neural networks (NN) for mapping and displaying image database content. Neural networks is a good approach for displaying topology or local neighbourhood of images within the data-set as explained in Section 2.2.2. As stated previously, within NN a Self-Organising Map (SOM) is a frequently used method in information visualisation [117], [106], [96] and [49]. The benefit of SOM's is their ability to perform the vector quantisation and dimensionality reduction at the same time.

The first proposed application which used SOM in image indexing was done in [215]. In [69] the authors applied SOM to image database visualisation and retrieval. The authors in [168] use image colour information for image clustering through a hierarchy of self-organising maps. The upper left corner of the map area shows blue colour that means that they represent images with a dominant blue colour. A user looking for a sky image can explore this region to find an item containing dominant blue in the upper part of the image.

In [25] the authors present PicSOM, a system for image retrieval based on tree structured SOM. In the initial stage the system displays a random set of images chosen from the observed collection. The randomly selected initial set aims at representing the whole collection as well as possible. The user selects the relevant images and, based on them, PicSOM scores the corresponding Best Matching Units (BMU's) with a positive value inversely proportional to the number of relevant images. BMU of any irrelevant images are assigned with negative values. The Figure 3.4 displays the layer of the Tree Structured SOM (TS-SOM), visualisation where visually similar images in means of spatial edge frequency and orientation are mapped closer together.

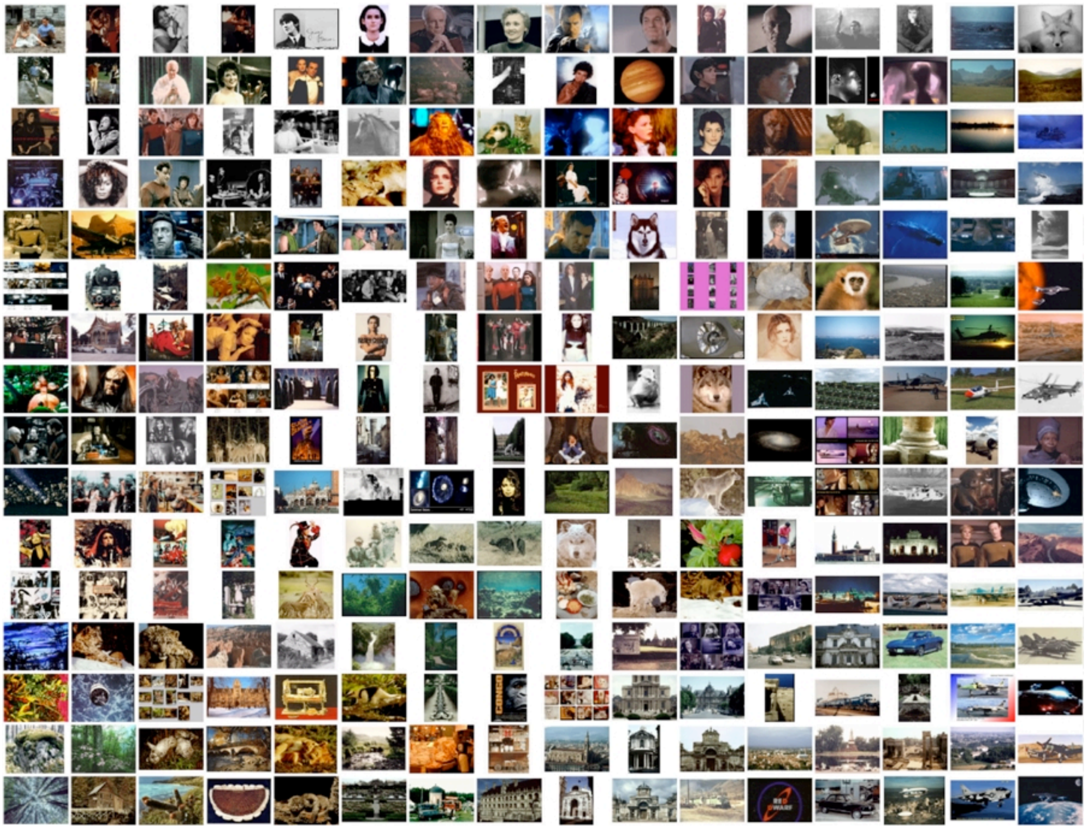

Figure 3.4: 16x16-sized SOM trained with Edge Histogram descriptors [25].

An advanced visualisation using SOM was applied in VideoSOM [14], where the authors developed an interface for browsing the video content using image video representatives (i.e. key-frames). Key-frames are processed and clustered using SOM and the results visualised as a hexagonal grid. Every grid cell defines one cluster and the cluster is represented by the most representative key-frame in that cluster. Cluster cells are framed with shades of green stating the number of key-frames allocated in the specified cluster.

For the spatial distribution of data objects on the screen and improved visual distinction of displayed items, statistical analysis tools (e.g., PCA and MDS) have been widely used in the visualisation community. Their application is in the domain of dimensionality reduction and mapping of multidimensional data into lower dimensional spaces (2D or 3D in case of visualisation) is given in Chapter 2. These algorithms attempt to embed data items into a low dimensional space while preserving the characteristics and relationships between data existing in multi-dimensional space.

In [127], the proposed visualisation called PCA Splat (PCS), displays mutual similarities based on the low level visual features that were extracted from images. The layout is generated based on object similarity to a query image. Similarities are calculated using image feature vectors and presented using a PCS based reduction approach. The proposed layout preserves both similarities to the query image but also all mutual similarities between the retrieved image set. In this work, attention was directed towards the problem of image overlap. It was initially concluded by the authors, that the layout which used linear PCA for mapping the images, produced too much overlap. For this reason authors developed an optimisation method in order to improve the layout and reduce content occlusion. In the improved version, images are re-positioned to reduce the overlap, while preserving as much as possible of the relations between images. The proposed method seeks the optimal value of a cost function dependant on two factors: first, the level of overlap between images and second, the level of deviation from the original image placement obtained using PCA. The optimisation goal is to find both position and image sizes that minimise this cost function.

In [58] similarities between retrieved images are preserved using the Kernel Principle Component Analysis (KPCA) used for mapping the images onto a hyperbolic plane. The authors applied PCA on different multidimensional image features leading to different visualisation results, as shown in the Figure 3.6 and Figure 3.7.

As stated previously in Chapter 2, the most commonly used non-linear projection method for data visualisation is *Multidimensional Scaling* (MDS). Although it is considered non-repeatable and slow, the advantage of this algorithm is the ability to provide low-dimensional representation based only on similarities or dissimilarities between data objects. It is frequently used for image visualisation (as explained in the following text) since it is able to visualise various types of relationships between images.

The application of MDS for image database visualisation and navigation was first proposed in [153]. In this work the authors propose a solution for navigating through the image collection based on colour similarities between images. Embedding of the similarity information was performed by the MDS algorithm. The obtained spatial 2-dimensional embedding provides an overview of an entire image database which is useful information for the user when exploring the content.

In [151] MDS is again used as a dimensionality reduction method for image layout. The authors use a similarity measure which takes into account both global image properties, such as colour and texture, as well as spatial layout of the image

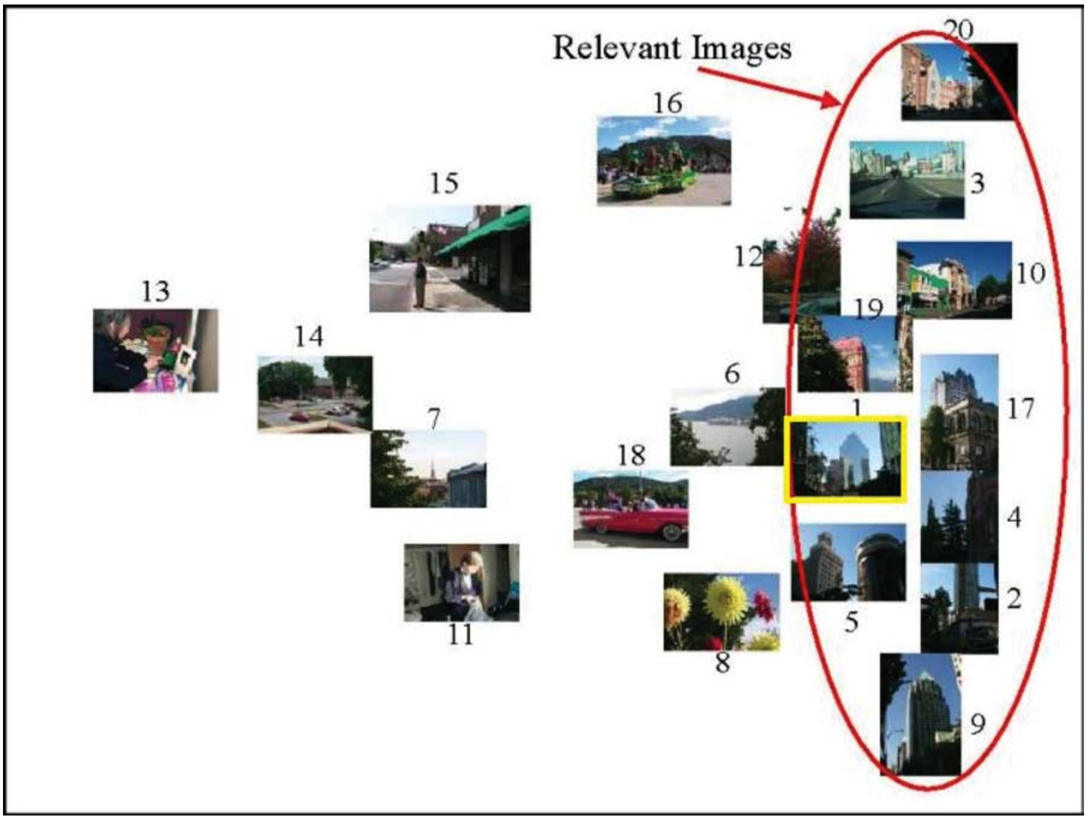

Figure 3.5: PCA Splat visualisation of 20 retrieved images [127].

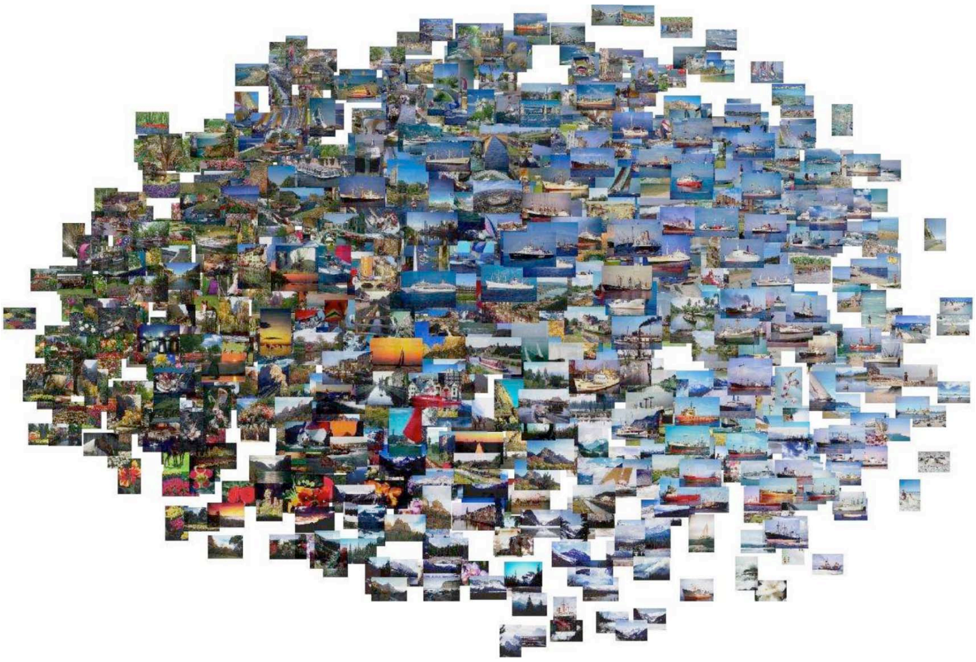

Figure 3.6: Kernel PCA projection of images retrieved as a result of a keyword query [58].

regions. The use of a similarity based approach has been explicitly motivated by the study that authors performed previously, which indicated that user might locate the images faster if the set is visualised according to the similarity (and not in a random way).

In some cases [119], MDS is also used to organise the images returned as a result of a query. In [180] and [181] the authors propose a combination of hierarchical organisation and MDS for supporting image retrieval. Each cluster in the hierarchy is represented with one image and MDS is applied only on those images.

In [125] the 3-dimensional MDS ordering is used for mapping the dissimilarities between images into the distances on the screen. MDS is good technique for embedding the image similarity information on the screen. Apart from structure preservation, the benefit of this approach is that the input information can be any form of similarity between images which provides some scalability of the approach.

Apart from PCA and MDS, other algorithms were used for embedding image similarities into their visual representation. In [35] the similarity between images is determined based on three visual features. For displaying image similarities, separate *Pathfinder networks* are generated for each of these features. A Pathfinder network is a method where the network structure is derived from proximities of pairs of examined data elements [166]. These proximities can be obtained from similarities, correlations, distances, conditional probabilities, or as any other relationship met-

Figure 3.7: Kernel PCA projection of images with different subset of features (with respect to Fig. 3.6).

ric between elements. For these particular Pathfinder networks, the authors use a triangular inequality condition to eliminate redundant links and provide a database summary.

The work presented in [118] proposes a graphical layout which allows modification of one parameter which scales the layout between similarity-based spatial one and a grid. The specific parameter is scaled according to the user request. This feature can be useful since it enables user to change view and observe relationships between the images (similarity-based layout) or perform easy and detailed examination of images (grid).

In [137] the authors focus on the problem of visualising collections with a very large number of images and emphasise the importance of three visualisation requirements:

1. *overview*;

2. *structure preservation*;

3. *visibility*.

In the case of large image databases it is not possible to display all images on the screen at the same time without reducing them in size (down to point-based representation). This is due to the limited display size of the devices, the so called *visualisation space constraint*.

In order to keep the sizes of displayed images large enough for the content to be understandable, the displayed set has to contain a limited number of elements. This subset should provide the overview of the collection thus its selection is very important.

The authors argue that a random selection of this subset is not a good option since it does not contain representatives for the whole collection. They propose the use of the k-means clustering algorithm for extraction of the representative set. The entire collection is grouped into clusters and cluster representatives taken for display. The mapping technique used by the authors is ISOSNE algorithm which combines ISOMAP [190] and SNE [77] approaches. This visualisation solution is also used in *GalaxyBrowser* [210] for supporting similarity based browsing of images. The third requirement considers the overlap of displayed images and their mutual occlusion in that case. The overlap problem might "hide" images so that user miss them while browsing. One good aspect of this solution is that it provides a window which shows the location of the selected subset within the entire collection. This way the user can focus on one part of the collection using a graphical overview. Another promising method found in this work is the selection of the image representative set using k-means (or other clustering algorithm).

In [76] a visualisation is used to display the results of a search. Image results are visualised in a form of a spiral where the distance from the centre is proportional to the distance between the query and the specific image. The authors also propose similarity based visualisation of neighbourhood images in the result set. The images are visualised according to their similarity to the query where the most similar image is positioned in the centre of the screen. The image relevance is also size-encoded where more significant images have bigger size.

The particular aspect of this work is the fact that it emphasises the importance of initial access method in case user does not want or does not know how to formulate a query. They use a grid to display the so called *high-connectivity hubs* of the constructed nearest neighbourhood network. This way the user is provided with the collection overview and can initialise a query by clicking on one of the images.

Another method used for image visualisation is spring-based visualisation, where similarities are modelled by springs, which is used mostly for visualising graphs. Their purpose is to position the nodes of a graph into a two dimensional or three dimensional space with certain edge lengths corresponding to relationships between data objects, while having as few crossing edges as possible. In [51] authors use mass-spring simulation to display relations between images. Similarity is based on image meta-data in which every image is associated with one or more string descriptors or tags. Springs exist between any two images that share a tag.

In [42] authors developed an *ImageVibe* interface where the placement of image object is determined based on *points of interest* (POIs). Each POI represents property of data object (colour, orientation or similar) relevant to the user. The user can select POIs and data objects will be placed according to their relationships with this set of POIs. A screen shot of ImageVibe interface is shown in the Figure 3.8.

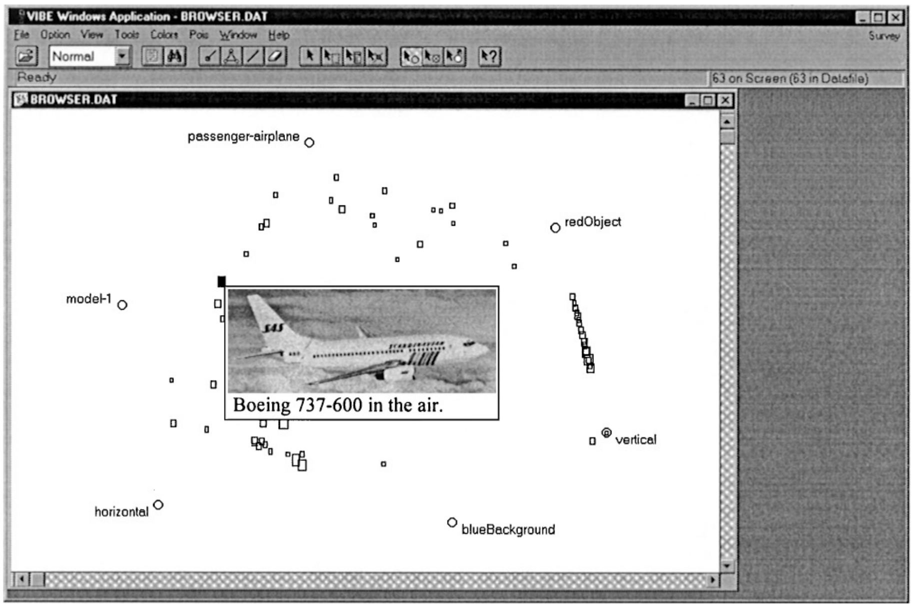

Figure 3.8: A screen shot of ImageVibe interface proposed in [42].

3.2.3 Hybrid visualisation solutions

Authors in [212] employ information visualisation for browsing and retrieval on a higher level in the proposed *semantic image browser*. They combine multiple visualisation techniques such as MDS and grid layout for displaying semantic image information as shown in Figure 3.9. The browser employs techniques such as MDS based image layout, relational display of the dimensions as well as various interaction tools for browsing and retrieval. The particular issue they focus on is providing image and content overviews, along with relationships between the dimensions. For showing what images exist in the collection, image miniatures are mapped in the 2D space using MDS embedding of all images.

Users can select images of interest through this display and then choose between several display approaches. The content overview is provided by displaying the distribution of detected concepts by means of *glyph* based visualisation and MDS. Each element or dimension is shown as a block and each image as a pixel in the

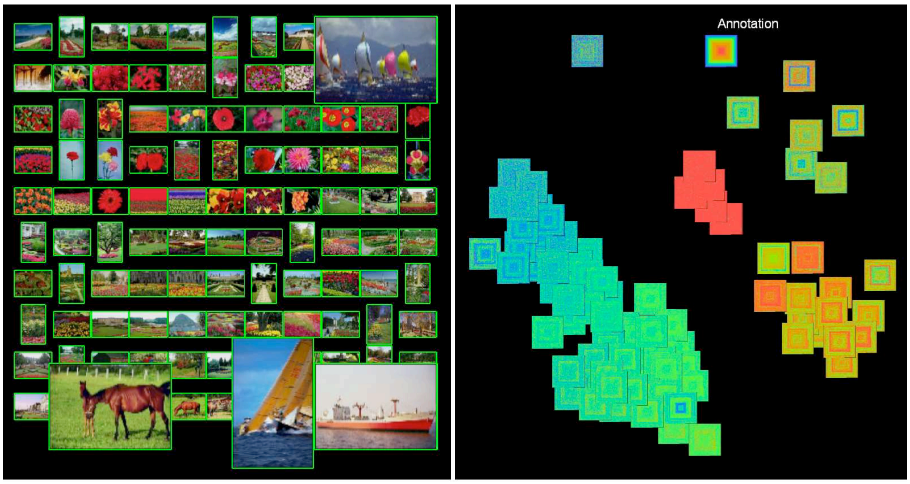

Figure 3.9: A screen shot of *Semantic Image Browser* proposed in [212].

block. The colour of image pixels depends whether specific content is detected in the image or not. For example, green colour means that the image represented by this pixel has certain content, whereas red pixel colour indicates an absence of the same content.

Apart from directly visualising the semantic relationship and information about images in one data-set, some research efforts are invested in visualising a higher semantic level of organisation - ontology. In most ontology visualisation the focus is on displaying relationships between semantic textual labels rather than images. For this reason most of the visualisation techniques used for ontologies use a tree or graph representation for displaying these semantic structures. Protege [140] offers a Windows Explorer like tree view of the ontology. OntoViz [171], IsaViz [144] are graph base browsers, SpaceTree [146] is a tree based visualisation of ontologies. CropCircles [204] shows innovative visualisation solutions using a hierarchy of circles to show relations.

3.3 Access and interaction methods

As stated in [109], when dealing with a huge database it is not feasible to navigate linearly through the content. Various statistical analysis and pattern recognition methods are used for generating content structures thus reducing the effort needed for content retrieval. However, generating structure without adequate means of their utilisation is useless.

Apart from algorithms for image organisation, efficient access and management of the content, using structural information can improve overall performance. Employing tools for supporting user interaction and exploiting human knowledge in the query process, can significantly influence the overall achievement of image management and retrieval systems. For this reason, in this section, various access and interaction methods employed for the stated purpose are discussed.

From the user's point of view, there is an important difference between two main interaction methods in image retrieval: content browsing and direct query. The most important factor which implies the preference is the level of user's knowledge about how to formulate the query and what to expect from the content set. If the user has clear and precise information needs, feels comfortable with the system and understands how to formulate a query, direct query is the most straightforward way to interact with the system. However a query is not always the best method in case the stated conditions are not met. In this sense, content browsing is often a preferable form of interaction, at least at the beginning, since it enables the user to build a query step by step, adding new search directions, thus reducing the searched content set. In fact, according to [123] browsing is a strategy that can be utilised to get familiar with the system and understand what is there, how to find it and where. While browsing, the user performs exploring and searching action, where searching direction can be changed dynamically, depending on the immediate results obtained from the system. In this section, access and interaction methods are discussed from different perspective. Both query and browsing can be supported by various visualisation methods.

3.3.1 Access

Information visualisation components, such as graphical layout and interaction, play an important role even at the very beginning of image management and retrieval processes. For this reason, the level of employment of information visualisation and interactive methods in one image retrieval system can be observed most easily through the way user query specification is supported. For example, systems that rely only on automatic algorithms do not need advanced interface solutions.

In systems with direct access, the user directly specifies information needs without an initial interactive phase such as exploration or browsing. In such cases, the level of information visualisation and interaction employment is usually low since the system relies on machine processing performance and not on the user. Examples of this access type are: pure query-by-text (or by key-word) after which the system returns a set of retrieved results; making a sketch of a query [60], [174], [32], [33] and [97]; or importing an external image into the system and requesting similar content [162], [133].

In the framework proposed in [1], both key-word and image queries can be used for query specification. For the text-based queries, after the user specifies one word, similar segments are retrieved through automatic-speech-recognition transcripts. In cases of query-by-example, the user can set one image as a query and the system is able to retrieve images showing similar concept vectors. However, in order to select an image suitable for query-by-example, the user has first to find the image within the repository or to use an image which does not belong to the database.

Another retrieval system based on a visual-query is described in [162]. Here the query is initialised by selecting one or more images which can originate from some external source (*e.g.*, from Google Image search). The user can then select objects inside the image, or specify which is the feature (colour or texture) or the visual-characteristic that makes it a good query image.

Apart from the systems where the level of user's involvement is relatively low, there are systems which aim at extending the user's involvement by allowing initial content exploration. In such systems, existing data structures should be used by the user for accessing the relevant images in the collection. From the perspective of the system, all images are equally relevant, but usually it is not possible to present all of them at the same time due to display limitations. Of course, observing from the side of the user, some images are more relevant than the others. In order to understand the information needs of the user there has to exist a certain *access point* which enables the user's initial engagement with the system. It will enable the user to specify what is a specific query task and which content is considered relevant. having this in mind, we believe that visualisation layout and interaction methods are able to provide significant methods of user support.

As an introductory information about the content, most retrieval systems provide an initial view of the database either by extracting the collection representative images [200], [217] or by performing random sampling [24]. By browsing through the displayed set of images, the user is enabled to select a certain element as an access point into the collection. This element can be an image in systems employing query-by-content [24], [26], [181] or a semantic element (concept) in case of query-by-concept [176]. In some cases the user should select both positive and negative examples in order to inform the system about the context of his query [157].

In [200] and [137] the authors perform clustering of images and then select a certain set of images to be displayed in the initial view. The user can then access the content by selecting a relevant cluster.

With a strategy similar to [28], by selecting one of the available low-level features, visual content is accordingly clustered, and depending on the current size of the content set, a fraction of the segments is selected as a content preview. The user can then decide to select a subset of clusters that seem to be relevant and discard the others, or repeat clustering on the current content set using another feature. A possible drawback lies in the fact that the user might not be completely aware of the searched content. On the other hand, the selection of a relevant feature to perform the clustering is not trivial.

In cases where semantically higher level information is available and the image collection is organised and labelled by semantic concepts, the overview can be provided using detected concepts as in [176]. Here authors suggest that the large lexicon of semantic concepts is very important for effective query-by-concept. In their proposed system they provide semantic access to video archives through 101 detected concepts (aircraft, food, night fire, outdoor and so on). This system allows three ways of query specification:

- query-by-direct concept for queries related to exactly one lexicon concept (if looking for aircraft select concept *aircraft*);
- query-by-sub concept which is again concept from the lexicon (select concept *football* when looking for sport);
- query-by-super concept where users can use concepts semantically related to the required one (select *animal* when looking for *ice-bear*).

However, search topics not covered by any of the concepts in the lexicon are not supported by query-by-concept, a user has to use other types of queries such as keyword or content-based ones.

In [21] the authors extract the visual dictionary from the content as a set of visual representatives. The idea is to create a meaningful user-friendly representation in form of visual elements. The interface simulation uses macro-icons for representing visual elements and the user access relevant content by selecting one micro-icon. In [104] the authors propose a system where a user's involvement is an important retrieval element. However, most of the retrieval elements are still automatic. After specifying the text-based query, the user is presented with a set of recommended concepts extracted from the list of available ones. But the recommendation is again an automatic process which can not cover the wide span of user information needs and predict in all cases the most suitable way for query specification.

The system presented in [47] does not allow a user to perform interactive querying. Instead, it provides a probabilistic model of the user's actions. So the query is intelligent from the automatic perspective but limited on the user side. However to be able to infer the user's task, some form of initial access has to exists. The problem of this solution is that interface contains only 9 images by which user needs to initialise the query.

In cases where content is organised into a hierarchical structure the access method strongly depends on the way images are indexed. When images are indexed by non-semantic information most interface solutions do not directly display structural views of the database [181], [24].

The absence of semantic information about the content and presented structural information about the data-set, makes retrieval using such systems difficult. The solutions [24], [164] and [109] support hierarchical browsing for enabling the user to initialise the retrieval process using low level complexity of the user interface where the exploration is mostly initialised by selecting one of the displayed images. In [24] the hierarchical levels are indicated numerically and the user can descend into the hierarchy only sequentially. However applying such solutions to large data collections and considering the limited display space, it is not clear how to initialise the exploration if no significant or query-similar image is in the first visualised set. In case nodes of hierarchy have textual labels a solution is somewhat simpler. A standard tree representation of elements in the hierarchy can be used for displaying the structure, as in the Windows Explorer application.

Higher levels of user involvement in initialising his/her task can be realised by asking the user to focus on image regions instead of the whole images. In [46] a user can choose the colours of the regions to specify the query.

In [34] the authors use user interaction with the system for query specifications for creating *semantic visual templates*. A visual template is a set of icons or example scenes/objects that represent the semantics of a template it is associated with. The icons are animated sketches made by users when specifying a query.

In [173] the system allows users to specify a complex query through iterative, direct interaction with the graphical display. The system works with multidimensional data by grouping the data into categories creating *facets* and displaying *facets* as distinctive bubbles on the screen. By clicking on the bubbles the user interactively performs the query thus performing initial access.

3.3.2 Interaction methods

Different interaction methods used in general information visualisation applications are explained in detail in Chapter 2. Here, the interaction methods employed in state-of-the-art image systems are discussed. It is important to understand that, due to the exploratory nature of the search process, the information needs of a user can change considering the new knowledge gained through the use of the system [194]. Also, once the user becomes familiar with the content of the collection, query specification is easier and more focused. After an initial stage, the user should acquire enough knowledge of the content and the patterns of retrieval and organisation in the system.

Due to unfamiliarity with the content and the system the user-task will rarely be accomplished in the first or second interaction step. Usually it will take more communication phases with the system. The success of these phases depends on the image processing part of the system but also from the interaction support provided by the system interface.

In most image retrieval systems interaction is based on a relevance feedback (RF) mechanism where user retrieves a sets of initial images, marks them as “positive” or “negative” and sends his/her “opinion” to the system [125], [154], [167], [138]. It is usually done by clicking once on positive and double-clicking on negative image examples, or ticking the image check-box and so on. Most systems use this information to adapt the content relevance criteria and return new set of images. The relevance feedback procedure is usually repeated several times until the retrieved results are satisfactory.

On the other hand, in [194] the authors construct the query using outputs of multiple iterations of human-computer communication. This approach allows the query to change and adapt according to the level of knowledge the user gains in each step.

In order to initialise the user-system communication process, most interfaces display a large number of images in the same visualisation view. Since their visual content is not understandable at once, due to the size or occlusion, user might need assistance in examining the presented images. Without seeing the visual content, it is very hard to decide what is relevant for the query. Some image systems employ *zooming* as an interaction method. Zooming is standard technique for requesting more detailed view of one visual element. In [125] and [163] the user can use zooming and rotation of a 3D visualisation space to understand the content better. In [17] zooming enables detailed view of a square area containing more images. In [212] the user can double click on image or *zoom in/zoom out* or *pan* the image display.

Another approache which provides direct interaction with the layout for eliminating clutter of images on the screen is image reordering [212], or changing the scale factor of the layout which influences the image distribution [118]. For changing the visibility of displayed content sizes of images can be reduced or enlarged by the user [212].

Changing the positions of images for improving the layout or updating image annotation can be achieved using *drag and drop* functionality [212]. It enables the user to select an image and to change its location on the screen. Adjusting positions of images can also adjust the similarity metrics in the image index space [138]. In [157] the user can move the displayed images in order to create a visual concept and group images he/she finds relevant together. The user can create concepts and drag images into concept boxes thus performing the annotation. Annotation of more than one item ("bulk" annotation) can be performed by selecting multiple images and clicking on the meta-data attribute as in [112].

In hierarchy based visualisations, interaction is mostly limited only to sequential access, where a user has to navigate from level to level [24]. A second approach is the direct selection of one node, if visual representation of the structure is available [112]. In [37] *similarity pyramids* are used for image organisation. The user can move along the x or y directions of one pyramid level in a panning motion to search for image clusters of interest. In [24] the hierarchical levels are displayed by numbers and the user is descending into the hierarchy sequentially by selecting a relevant image.

Sequential access is the most used method of exploring hierarchies. Depending on the information the user has while accessing level by level, it can be more or less efficient. A potential drawback is a hierarchy without semantic information which is an especially important case in image retrieval. In the case of image collections indexed by visual indices, the semantic information is low (or non-existent), which makes sequential navigation very difficult to employ. How should the user decide where in which direction to continue?

Colour encoding and highlighting are visualisation methods which support visual understanding and dynamically emphasise results of human-computer interaction. Colour encoding is used in [173] to visually distinguish different information categories. In [214] different colours are used to distinguish different *Visual Islands*. In [51] the authors visualise all the images from the set. The user can highlight and remove images depending on a selected tag. All images that do not have the selected property are dimmed when the user points the mouse over a tag.

For quick browsing through images and requesting details of some image region *distortion* is an efficient interaction method [212], [118], [157]. Distortion dynamically follows user's movements and changes the layout of images. It helps users to see both the local detail and the global context information at the same time. The region or image in focus will be enlarged while the position of the images that are further away from the focus will appear slightly distorted and small.

In this chapter, an extensive review has been performed in order to examine information visualisation algorithms and tools used for visual image management. It is evident that a significant number of systems employ visual graphical interfaces with interactive support in order to help users to accomplish the tasks. However, the level of integration of information visualisation techniques and tools is still relatively low. The review has been used for identifying drawbacks and missing elements of image management and retrieval system, which are addressed in the following chapters of this thesis.

Chapter 4

Similarity-based overview of an image collection

As previously explained two important parameters in image retrieval are the "user task" and the "search strategy". While the user task or goal can be well defined or can remain a vague information quest, the search process can be classified as directed or explorative [148]. Apart from Trecvid-like situations where "someone knows of a video, has seen it before, believes it is contained in a collection, but doesn't know where to look" and where the direct search is efficiently supported by the system, there are many situations in which the user task is accomplished through the combination of direct query and explorative search. In most cases this implies an iterative process in which the user starts querying the system and the system returns a set of results after which the same process continues until the task is accomplished. The size of the result set returned by the system depends on the query, and is retrieved following the correspondence between the query and the image indexes.

Without focusing on why certain image features are used for image indexing and retrieval, the focus is on "how" this set is presented or visualised in order to help the exploratory search based on the specific content and the available information about the same content.

The proposed visualisation solution addresses the visualisation of colour image collection for explorative part of the user search process. It can be used for visualising the entire image collection if the user search starts with browsing, or used as an aid to display the resulting set after the search is performed in some other way (key-word based, query by example etc.).

It is clear that image visualisation as a retrieval aid depends on the underlying retrieval system, on the user task as well as the type of content over which retrieval is performed.

Since visualisation should support the retrieval and the way content is indexed, here we present visualisation of image set based on colour since colour is one of the most widely used visual features in image and video retrieval [36]. This is especially true where the content of the collection is diverse and the retrieval application is dedicated to the wide range of users and queries. This is due to the fact that, colour features are relatively robust to changes in the background colors and are independent from image size and orientation [170].

As an example of content set where the proposed visualisation can be used for retrieval is the large set of images professional photographs make during one Italian wedding. The standard set of images includes pictures made at the home of the bride and groom, pictures made during the church celebration, pictures of the couple in the open space after the church (garden of a villa, park or similar), pictures of the wedding celebration inside a restaurant or in the garden, and finally photos taken during the evening. In this case, we could distinguish several image groups based on the dominant colour of the location where they were taken. For example, church and indoor environment will be very different from green dominated pictures taken in the park or the garden of the restaurant. Considering this, clustering and visualising images based on colour feature can be a fast and effective method to focus on certain image subset in order to find, for example, images taken during the church ceremony. This is particularly useful when bride and the groom need to make an image selection to make a photo album of the wedding.

However, application is intended to be scalable to other visual-based image indices without loss of generality. In cases where the collection is dedicated for e.g. architects and the retrieval of some specific content, it might be more important to distinguish between regular and irregular shapes rather than colours. In this case edges could also be an important feature to base image indexing on. Similar stands for retrieval of medical images where magnetic resonance as well as various scanners produce grey scale images and the image retrieval has to rely on features different from colours.

The evaluation, performed in form of comparison with related MDS-based layouts, demonstrates the quality of the overview provided by the proposed solution in terms of graphical layout and conveyed semantics as explained in Section 4.5.

4.1 The idea

Nowadays a wide range of low-level information, usually arranged in the form of feature vectors, can be easily derived from images due to a variety of automatic algorithms [156]. However, this information is not easily interpreted by humans since for example, an array of numbers does not say much about the visual content of one image, but it is easily interpreted by the machines and are used to extract similarities between images. Based on these similarities, a visual representation of the collection can be generated. Even-though the user might not understand the rules of similarity by which the data is visualised, he/she can notice the patterns in the visual representation.

For example, taking the MPEG-7 Color Layout descriptor as low-level feature and using standard similarity based visualisation approach such as Multidimensional Scaling (MDS) the result obtained will be one large "cloud of images" as the one shown in Figure 4.1. This example displays 700 images from the Corel collection [45]. Due to the size of the collection and scaling of image sizes, image details are not directly visible. However, even if the user does not know the criteria adopted for image organisation in the screen, just by glancing at this layout it is easy to understand that colour is the organisational factor. Although the layout does not explicitly show the information about which images contain *water*, just by glancing at the visualised "cloud", humans can conclude that images with water can be most probably found in the blue region on the right.

Applying MDS directly to the whole data-set will surely provide extensive similarity information. In addition to this, it will also provide localisation of similar images together thus enabling the user to focus on a certain areas while exploring the content. However, such "full MDS" approach can be a non-optimal choice considering the aim of visualisation. For example, authors in [151] analysed the effect of MDS colour-based, similarity visualisation, and concluded that a "cloud" of images does not adequately distinguish images with similar colours.

In order to emphasise the similarities but at the same time distinguish dissimilarities between images, a combination of MDS and a clustering algorithm is used to generate graphical overview of the image collection.

As stated previously, clustering algorithms partition a set of objects into groups or clusters so that objects within one cluster would have high similarity between each other and low similarity to objects in other clusters, thus providing distinction between different groups of images in the collection. Adequate visualisation of such clusters can visually separate the content in a better way than when a "full MDS" approach is adopted. Taking the previous example with MPEG-7 Color Layout descriptor and performing image clustering, the obtained set of clusters can be visualised as shown in the Figure 4.2. In this example, similar images are clustered together and similarities between clusters are stresses by larger inter-cluster distances (providing more intuitive visualisation solution from the one proposed in Figure 4.1).

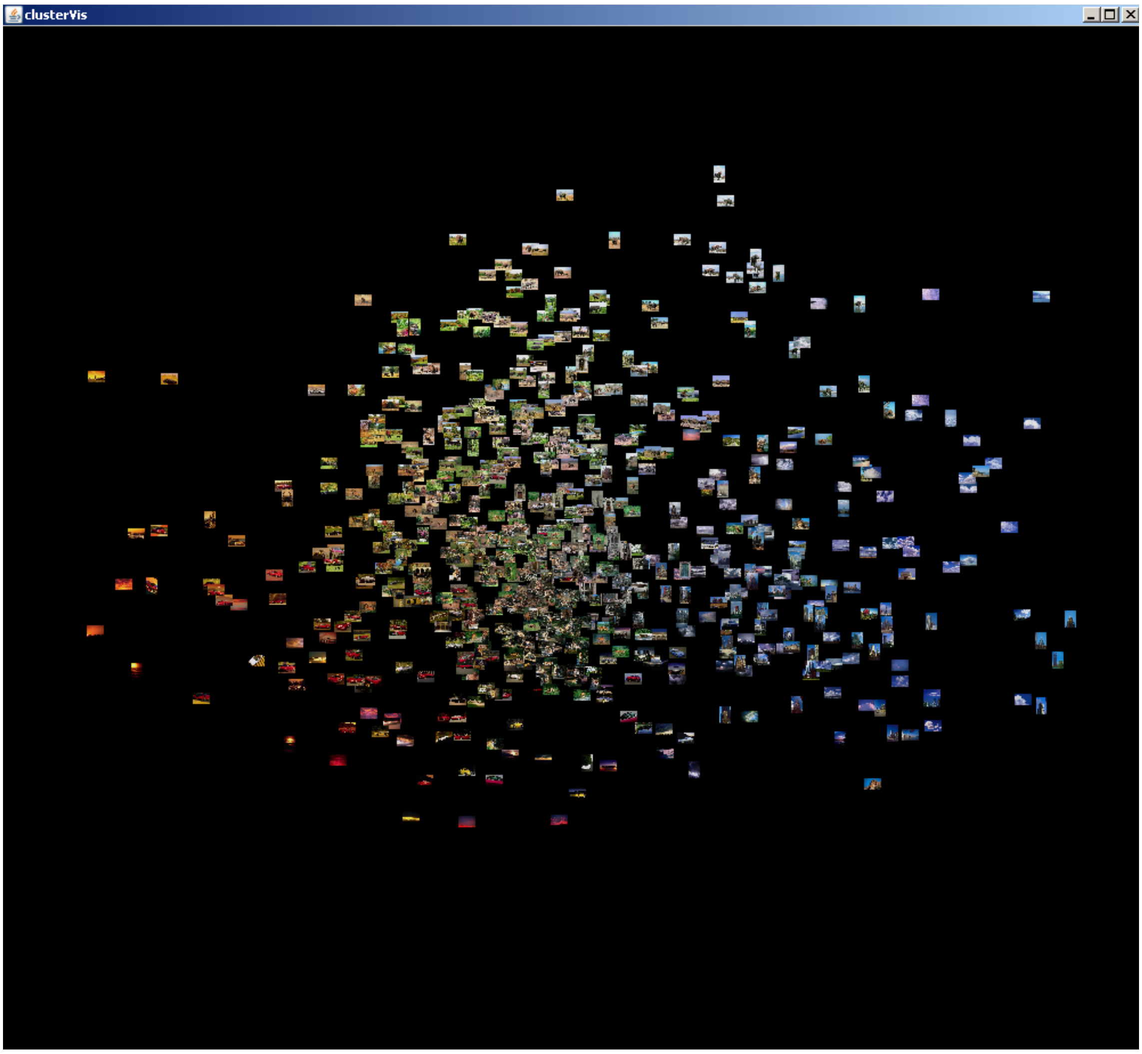

Figure 4.1: MDS similarity-based layout of 700 Corel images using MPEG-7 Colour Layout descriptor.

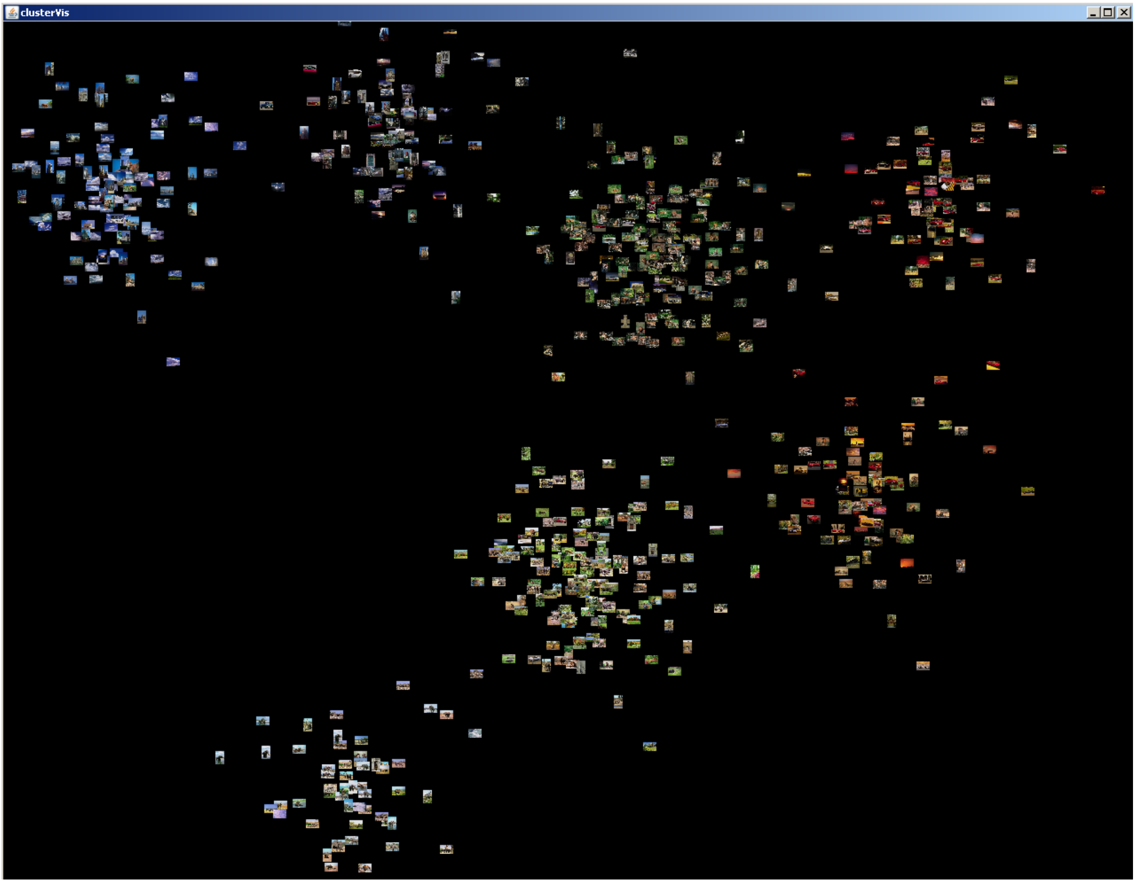

Figure 4.2: Similarity-based visualisation of clustered image collection using MPEG-7 Colour Layout.

The layout displayed in Figure 4.2 is a final proposed solution developed through several steps: in Section 4.2 the initial idea for generating the image overview is stated; Section 4.3 discusses the embedding of the global information using MDS for placing cluster centroids; in Section 4.4 three layouts embedding global and local similarities are described.

4.2 Maximising inter-cluster distances and random layout

As reported in [87], the idea was to generate a graphical layout so that the user can understand that there are several distinctive image groups in the data-set, while maximising the use of the display space. For this reason, in analysing the results of the clustering algorithm two type of information are found important: the positions of cluster centroids and distribution of the n images among clusters. Following these information categories, the layout algorithm explained in this section consisted of two generation steps:

- Placing the cluster centroids in order to maximise their inter-distances;
- Placing the rest of the cluster images within local areas around cluster centroids.

4.2.1 Positioning cluster centroids

Following the idea of "scattering" content over the screen and using all the available display space, the initial solution adopted was to find the most distant positions on the screen for given set of points. For example, the ideal maximum-distance placement for 5 points in a 2D space would be the one shown in Figure 4.3.

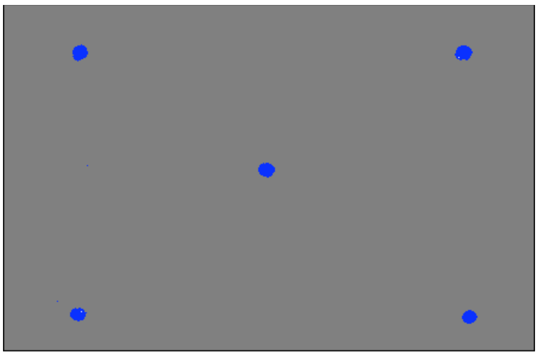

Figure 4.3: Maximum-distance placement of five points in a 2D space

The centroid placement problem was defined as follows: finding the geometrical configuration of k points (k cluster centres) in a limited 2D visualisation space that maximises the distance between these points. The initial distribution of points was the result of a simple algorithm which generated a random set of k points, through iterative process, aiming at finding the optimal one. The centroid locations were obtained as follows:

- Step 1: k points $p_i = (x_i, y_i), i = 0, 1, \ldots, (k-1)$ were generated according to a uniform random distribution and stored in a matrix $P = \{p_i\}, i = 0, 1, \ldots, (k-1)$. The constraints on the random generator were set according to the dimensions of the computer display;
- Step 2: The matrix of distances between all the points was calculated as:

$$D = \{d_{ij}\} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4.1}$$

- Step 3: The minimum distance value was found as

$$C_{iter} = min\{d_{ij}, i \neq j\} \tag{4.2}$$

- Step 4: Compare the obtained minimum value C_{iter} with the stored minimum value C_{min} (or the value from the previous iteration C_{iter-1} in case iter=1). If the value C_{iter} is greater than the existing stored minimum value C_{min} the old value is replaced with the new value, along with the corresponding matrix P ($P_{min} = P_{iter}$).

After a number of iterations which ensured that no substantial improvement can be obtained, the saved value of the matrix P_{min} would give the set of centroid positions.

It is interesting to note, that the goal is not to maximise the average distance value but to find the configuration with maximal minimum distance. The results will provide configuration of points with maximised inter-distances.

These positions were used for generating the graphical layout of cluster centroids as shown in Figure 4.4, where the image representatives of each cluster were placed at the locations obtained for the centroid of the respective cluster.

4.2.2 Random layout generation

For displaying the rest of the image collection, according to the results of clustering, the $N - k$ images are placed on the screen according to following procedure:

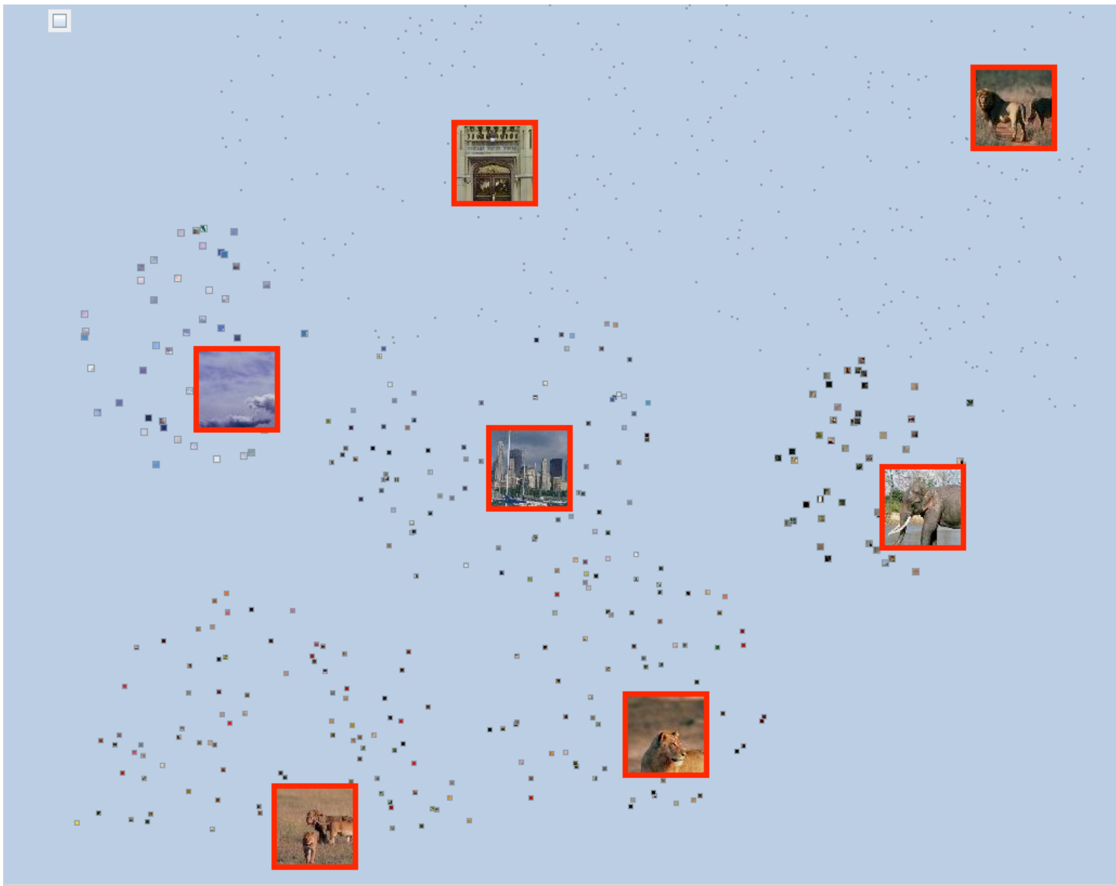

Figure 4.4: Placing the centroids of image clusters by maximising inter-cluster distances.

a $(n-k) \times 2$ location matrix L is generated off-line as a set of uniformly distributed points in a limited 2D space. A visual example of a generated 700×700 location matrix is shown in Figure 4.5.

After generating the centroid positions, each image is assigned to the cluster it belongs to in a position which is computed as follows: for all clusters, all images in that cluster are ranked according to their similarity s to the cluster centroid, by computing the Euclidean distance between the respective feature vectors. In particular, the similarity between image i and cluster h centroid c_h is obtained as:

$$s_{ih} = \sqrt{(\vec{f_i} - \vec{f_{c_h}})^2} \tag{4.3}$$

where $\vec{f_i}$ and $\vec{f_{c_h}}$ are the feature vectors of image i and centroid c_h, respectively.

In addition to this, the points stored in the location matrix L are also ranked according to their Euclidean distance to each centroid c_h where the point with least distance from centroid is stored as first.

The distances between images in one cluster and the centre of that cluster are then mapped in the corresponding ranked positions preserving the similarity information between the centroids and images. In other words, the location points closer

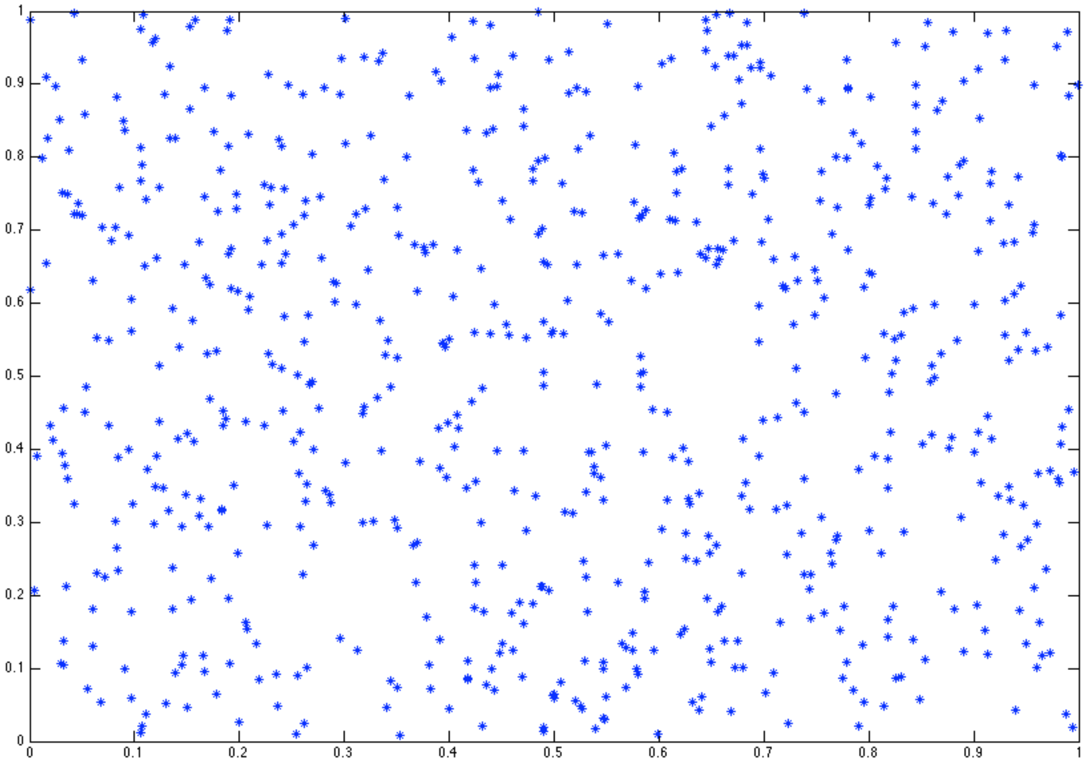

Figure 4.5: Layout with 700 randomly generated positions according to uniform probability distribution.

to the centre of the *visual cluster* are occupied by the images which are more similar to the cluster centres. For example, if an image i, which belongs to cluster h, is the most similar to the centroid c_h, it would be placed at the closest pre-defined location for cluster h. This way the local or intra-cluster similarities are partially preserved or in other words, this solution preserves image-to-centroid similarities while "sacrificing" image-to-image similarity (the least important ones considering clustering inaccuracies) in order to lower the layout complexity.

For the example shown in Figure 4.4, a k-means clustering algorithm was used (algorithm is described in Appendix A) with Colour Layout descriptor as the feature vector. K-means is a fast clustering algorithm where the number of clusters has to be previously specified. Since the initial layout solution employs k-means clustering, two clustering issues need to be emphasised here:

- The results of the applied clustering approach are deeply influenced by the accuracy of the low-level representation;
- The specified number of clusters is a parameter which hugely influences the generated content structure and its visual representation.

These two issues can greatly influence the expected distribution of visual content through the clusters: it can happen that relevant content can not be found in the expected cluster of images and that the user does not have an idea where else (in which other cluster) it can be found; the second issue can cause again a non-optimal grouping of images with effects similar to previous ones.

In order to address these issues using information visualisation, instead of maximising the distance between the *visual cluster* centres, an alternative approach is to provide information through the spatial relations between clusters. This way the global similarity information will be better preserved and the user can infer his/her own the connection between content clusters: if the content is miss-classified it is very likely that the user finds it in some of the neighbour clusters (using displayed global similarity).

4.3 Preserving inter-cluster similarity relationships employing MDS

The hypothesised layout solution described here is that an adequate visualisation of the similarity information can disguise the two, previously described drawbacks of clustering. For example, in the case when the initial number of clusters is bigger than the optimal one, clusters that contain similar content can be placed closer together so that erroneous number of clusters will not have a negative effect on the user exploration. On the other hand, an efficient interactive exploration can enable the user to move from group to group and locate "miss-clustered" images.

Here, similarity information is embedded in the layout on both levels identified previously (i.e. global and local). In addition to partial local similarity preservation, using the ranked random layout described in previous section, the solution presented here also preserves global similarity. Here, this global, inter-cluster similarity information is preserved by mapping the inter-cluster distances into screen positions using the Multidimensional scaling (MDS) algorithm.

Between two considered MDS algorithms, metric and non-metric (explained in Appendix A), the non-metric was employed here for two reasons:

1. It is computationally more efficient;
2. It preserves the ranking order of the distances.

Monotonocity constraints given in non-metric MDS preserve the ranking order of distances so that larger distances in the embedded space will correspond to larger dissimilarities in the high-dimensional space, which is exactly the information that the proposed visual representation aims at conveying.

The graphical layout generated using non-metric MDS for 700 images from Corel collection [45] (clustered using k-means) is shown in Figure 4.6

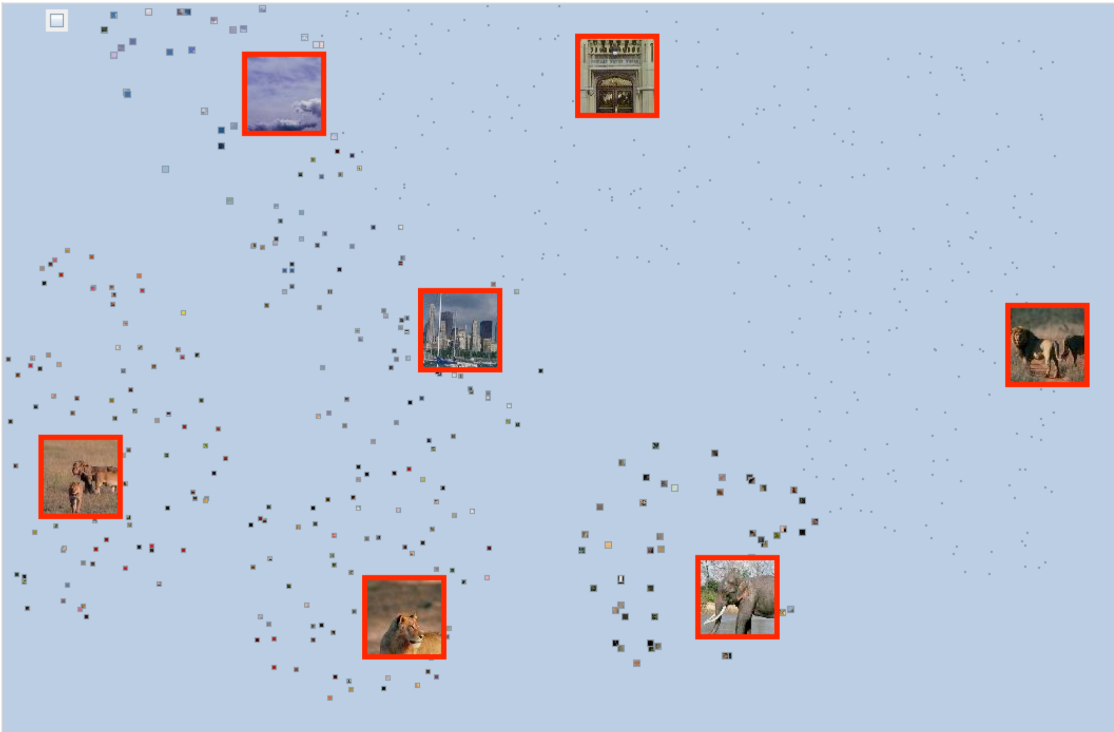

Figure 4.6: Layout of 7 *visual clusters* using non-metric MDS for centre placement and ranked uniform layout for images in each cluster.

This solution employs the visual size encoding of images which means that the clusters with more images are represented with objects smaller in size. This is useful to display all images with less overlap and to provide quick visual information on the cluster size.

The set of points generated for centroid placement are normalised values in the range $[0, 1]$, which are then scaled according to the size of the visual display. This ensures the optimal use of limited visualisation space.

Observing the overall approach for image placement (global plus local similarity embedding methods) it was clear that the distinction between different image clusters were not satisfactory. In other words, as a result of the uniform ranked random layout, different clusters were overlapping on the screen. The approach described in the next section aims at addressing this issue.

4.4 Preserving similarity relationships between images

After applying non-metric MDS algorithm for placing the cluster centres, the improvement in the visual layout in terms of global similarity information presented were evident. However as said before, the question regarding the optimal placement of non-central images around the cluster centres remained an open issue. As previously explained, some images belonging to different clusters were overlapping so that different image clusters were not visually distinguishable enough.

Considering this, further efforts for improving the data-set layout advanced in two directions:

- Examining the other similarity mapping techniques for placing the cluster centres;
- Improving the method for positioning non-central images on the display.

As mentioned before, MDS is a good approach for preserving inter-image relationships but its exponential complexity creates an important issue. Trying to map n images using MDS, will introduce costly processing operations since the computational complexity of MDS algorithms goes from $O(n^2)$ for standard MDS algorithms to sub-quadratic complexity of the algorithm proposed in [131]. If k-means clustering is performed in the original high-dimensional information space[1] the algorithm complexity is reduced to $O(nkm)$ where n is the number of images, k the number of clusters and m the number of iterations. But an even more important question is, how are the results in terms of the generated image layout?

The work presented here shows several approaches for placing the images using different versions of the MDS algorithm, as well as their combination with random layout. It also aims at showing how the content layout is influenced when preserving inter-cluster distances and partially "ignoring" intra-cluster image similarities. Placing the cluster centroids according to their mutual distances retains the relationship between the clusters. The more similar clusters are one to another the closer together they will be on the screen. Considering the assumption that one cluster contains similar images, the objective is to preserve the global relationships between all the items. The user should then able to identify main topics of the content quickly and efficiently, just by glancing at the this spatial representation of clusters.

For producing the layout with previously described characteristics, three methods for graphical layout were investigated:

1. Placing the cluster centroids using classic non-metric MDS and iteratively embedding the rest of the (whole) image collection using incremental MDS (described in Appendix A). This approach is described in Section 4.4.1;
2. Placing the cluster centroids using classic non-metric MDS and positioning images on a cluster base (local similarities) using classic non-metric MDS algorithm. This layout solution is described in Section 4.4.2;

[1] Feature, semantic or any other space

3. Positioning the cluster centroids using non-metric MDS and placing images on the cluster base randomly within localised cluster areas. This third approach is described in details in Section 4.4.3.

4.4.1 Combining classic non-metric MDS with iterative MDS

In order to present as accurately as possible both global and local similarities and provide scalability of the graphical layout in terms of adding new image elements, a combination of classic and iterative MDS algorithms have been employed. After placing the cluster centres using classic non-metric MDS, the layout for the rest of the image collection was calculated according to the iterative, spring based MDS algorithm proposed in [15].

The spring based solution proposed in [15] aimed at addressing the computational complexity of the classic MDS algorithm by performing incremental iterative layout of data objects. Here the performance of this algorithm is also observed in terms of the layout it generates and the flexibility it provides. The relaxed spring between two points is given by the dissimilarity between corresponding data objects. Each spring has an energy, and the loss function of the system is the sum of energies for all the springs.

In the first generated layout, centroids of *visual clusters* are positioned according to the non-metric MDS, and they are set as anchor points for iterative, spring MDS. Position p_i for each image i was determined by minimising the energy of the spring system respective to the established anchor points.

The spring based MDS approach was employed in the proposed layout development since it allows a set of points to be initially fixed independently from the rest of the collection. This feature enables the application of this spring based MDS for the proposed layout. After anchor points are fixed, scaling can be performed in "one by one" manner, or by so called *single scaling*. This scalable property of the algorithm, is an additional advantage if compared to classic MDS. In fact, in case that there is one new image to be displayed, after the initial set is layed out, it is possible to determine the new position based on the existing object placements without re-calculating the positions for the entire displayed set (as with classic MDS).

Applying the spring based system in the single scaling mode, where cluster centres are the anchor points, has a big drawback in terms of the generated layout. If data objects placed in the previous iteration do not adjust their position at each step, the layout will have a sparse structure. The example layout for 700 Corel images, obtained without re-positioning previously placed points is shown in the Figure 4.7.

The comparison in the Figure 4.7 was performed between "full MDS" (blue

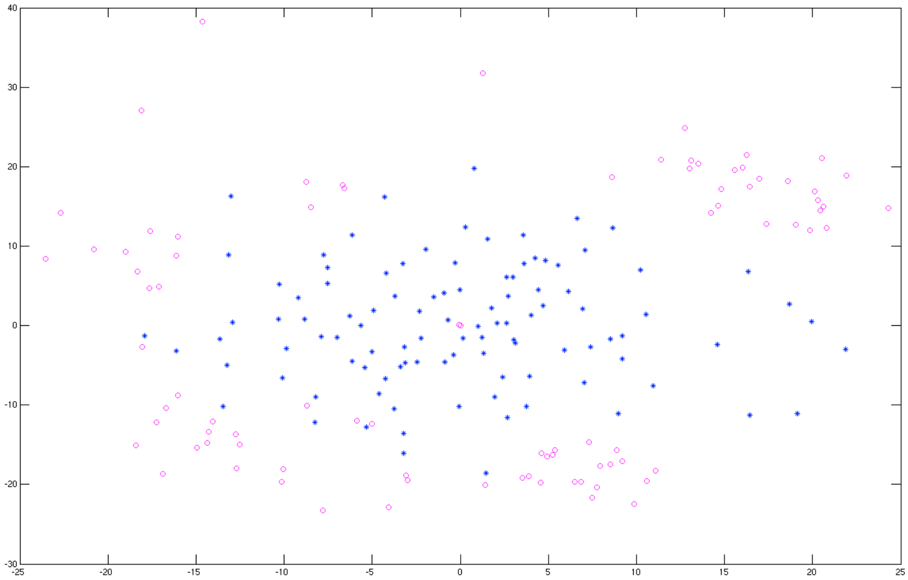

Figure 4.7: "Full MDS" layout (blue) versus non-adjusting single scaling spring-MDS layout (magenta).

points), calculating all distances in the data-set, and the spring based MDS (magenta) where centres were placed first, followed by the iterative placement of the rest of the images (and no re-positioning).

As can be seen from the figure, this approach does not "allow" a new point to arrive closer than the relaxed string distance, which in case of positioning using cluster centres does not produce a satisfying layout. In such cases new image objects are repelled on the outskirts of clusters which prevents relationships to be accurately displayed.

4.4.2 MDS centroid placement with cluster-based MDS

Considering that an iterative spring-based method does not provide a layout which optimises the available display space, further improvements and research were required. First, all centres of *visual clusters* were allocated using non-metric MDS, thus preserving global similarity. The second step was to apply the MDS for embedding the complete intra-cluster similarities, image-to centroid and image-to-image.

This approach, reduces the complexity of calculating the full set of distances for the entire data-set. In particular, the complexity of finding the locations for placing cluster centroids is $O(k^2)$, for classic MDS, where k is the number of clusters/centroids. In addition to this, the complexity of per-cluster-MDS is in total

$$\sum O(n_h^2), \tag{4.4}$$

where n_h is approximate number of images in cluster h, obtained as:

$$n_h = \frac{n}{k} \tag{4.5}$$

and k is the fraction of the total number n of the images in the data-set. This approach implies, that in case a new image needs to be displayed the new positions of images are re-calculated within one cluster only. The example of layout generated in this way is shown in Figure 4.8.

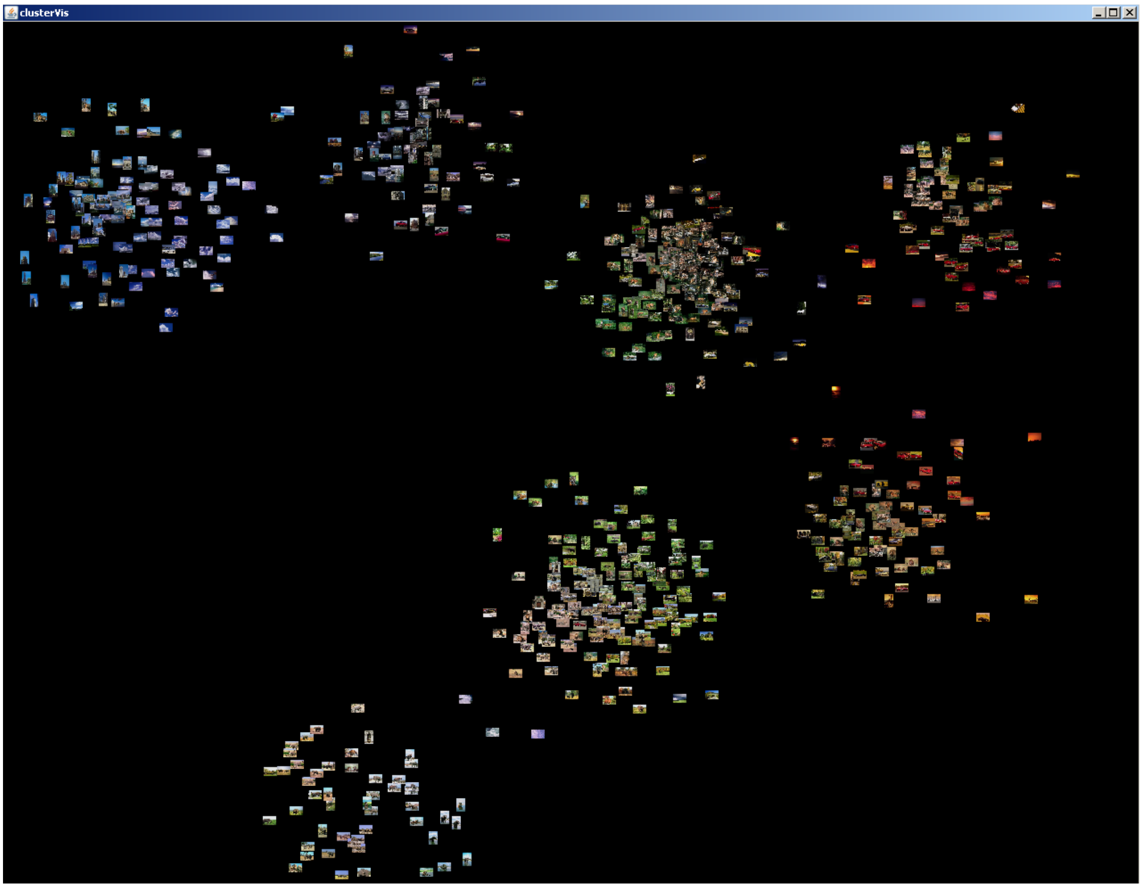

Figure 4.8: Embedded inter and intra-cluster similarities using non-metric MDS for 700 Corel images.

4.4.3 ClusterVis - an MDS centre placement with ranked Gaussian random cluster layout

It has to be emphasised here, that apart from multimedia based researchers, the common user is not interested in exploring the values of the low level descriptors (such as MPEG-7 Colour Layout). On the other hand, it is highly likely that the user would try to locate one or more image objects based on their semantic content. With this rationale, the motivation for deviation from standard similarity based image mapping performed within this thesis has been justified. Following this opinion the fifth layout approach, named *ClusterVis* is proposed and implemented. Here the method for global similarity presentation from previous approach is kept, with the modification regarding the method for populating the local cluster areas. In other words, the locations of cluster centroids have been obtained based on inter-centroid similarities using non-metric MDS. On the other hand, a random layout is generated for placing images within local areas (creating *visual clusters*). This approach differs from the one described in Section 4.3 since it prevents images belonging to different clusters to mix by localising the random positions.

In order to explain better the motivation for ClusterVis layout approach, three facts have to be highlighted here:

- Similar images are expected to be in the same cluster;
- Dissimilar images are expected to be located in different clusters;
- Even in the case of inaccurate clustering results (in terms of semantics) the user should be able to visually locate image of interest.

Considering stated observations, the goals of the proposed solution are: to display different clusters as distinctive groups on the display, to place images of one cluster close to each other, and to provide a quick method for the exploration of the collection through this overview.

The stated goals and experience gained through previous implementation results and related issues, influenced the shape of the third layout solution proposed here. Again, cluster centroid positions are calculated based on inter-cluster distances using non-metric MDS algorithm. As previously explained, this preserves the information about the cluster similarities and provides global relational information. Once the centroid locations are specified, circular local areas with radius r_h around them are defined, for mapping the image objects.

These areas (*visual clusters*) are then populated by positioning images according to bi-variate normal distribution whose average values are:

$$\sigma_{x_i} = \sigma_{y_i} = \sigma = \frac{r_h}{3} \tag{4.6}$$

$$\mu_{x_h} = x_{c_h} \tag{4.7}$$

$$\mu_{y_h} = y_{c_h} \tag{4.8}$$

where the x_{c_h} and y_{c_h} are the co-ordinates of the centre of cluster c_h, respectively and σ standard deviations. The radius r_h is a scalable value which depends on the size of the display device and number of images per cluster. Using a bi-variant normal distribution, the 99.7% of images will be located (as in Section 4.3) within the radius of the area which 3σ, while the remaining ones will be randomly placed at the distance r_h. Ranked order of image placement will ensure that the more similar images to the cluster centroid are located closer to the cluster centre. This way only inter-image similarities per cluster are ignored. While placement of images within the limited *visual cluster* area prevents overlapping of different clusters (problem of solution in Section 4.3).

The ClusterVis layout of 700 Corel images distributed in 7 clusters is showed in Figure 4.9.

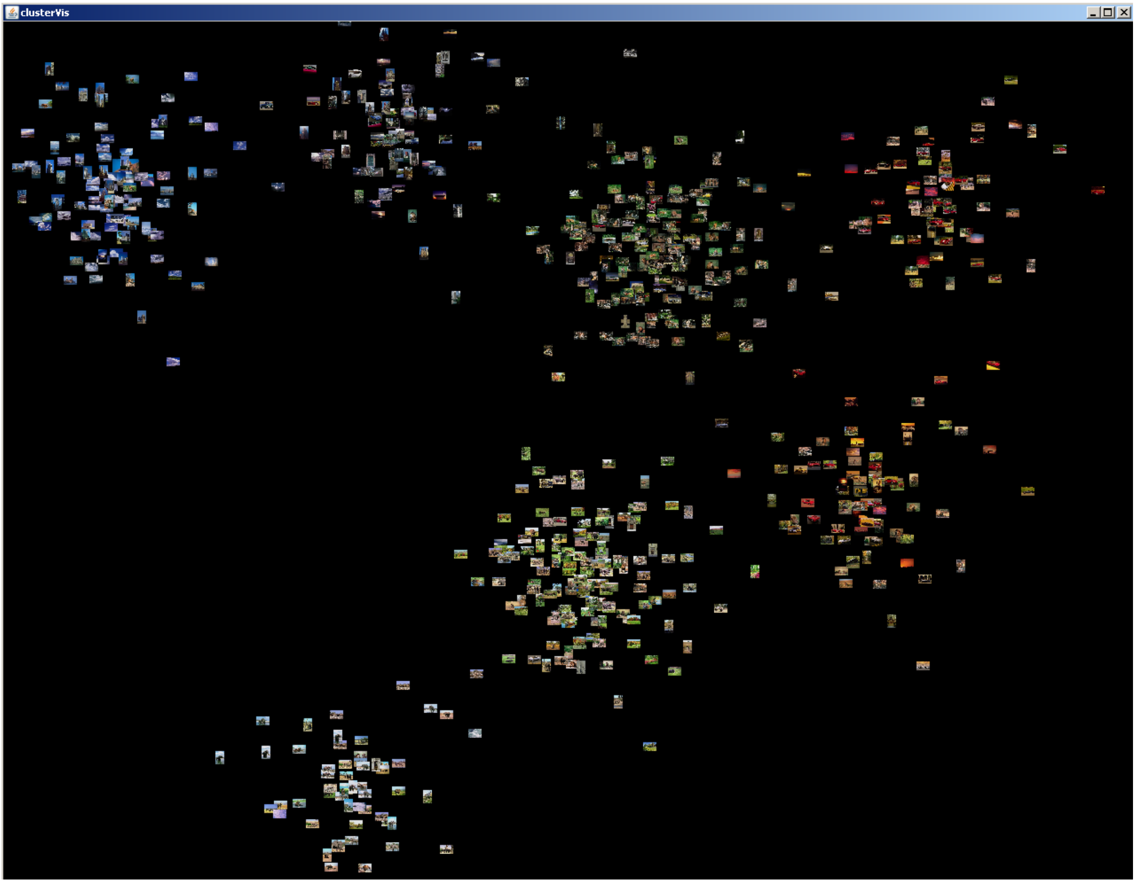

Figure 4.9: ClusterVis layout of 700 images from Corel collection using combination of MDS and ranked Gaussian placement.

In addition to providing a graphical layout with the overview of the image collection, additional interaction methods need to support user content exploration, especially considering the fact that images are represented by mini-icons. Between zooming and distortion (explained in Chapter 2), which are both interaction methods suitable for dynamically obtaining quick details, the choice fell on distortion since it is considered here as a more suitable and quicker method for providing fast image details. In fact, user does not have to position himself exactly on the image, like in case of the zooming, in order to enlarge it, and instead, a quick "run over" image set is enough.

Although the radius of the *visual clusters* is a scalable parameter its value can not be changed by the user (in the current application). Useful functionality would be to give the user the option to modify the radius of the visual cluster thus zooming into the content on the global level. It can be done in two ways: by including the scale button where user can change the radius between the maximum and minimum value; and by zooming in directly on the cluster where allocation of the layout area for every cluster will change in order to provide more space for the group of interest. The screen shots of radius values $r = 50c$ and $r = 60c$ are displayed in the Figures 4.10 and 4.11, respectively. The c is a constant which depends on the display dimensions.

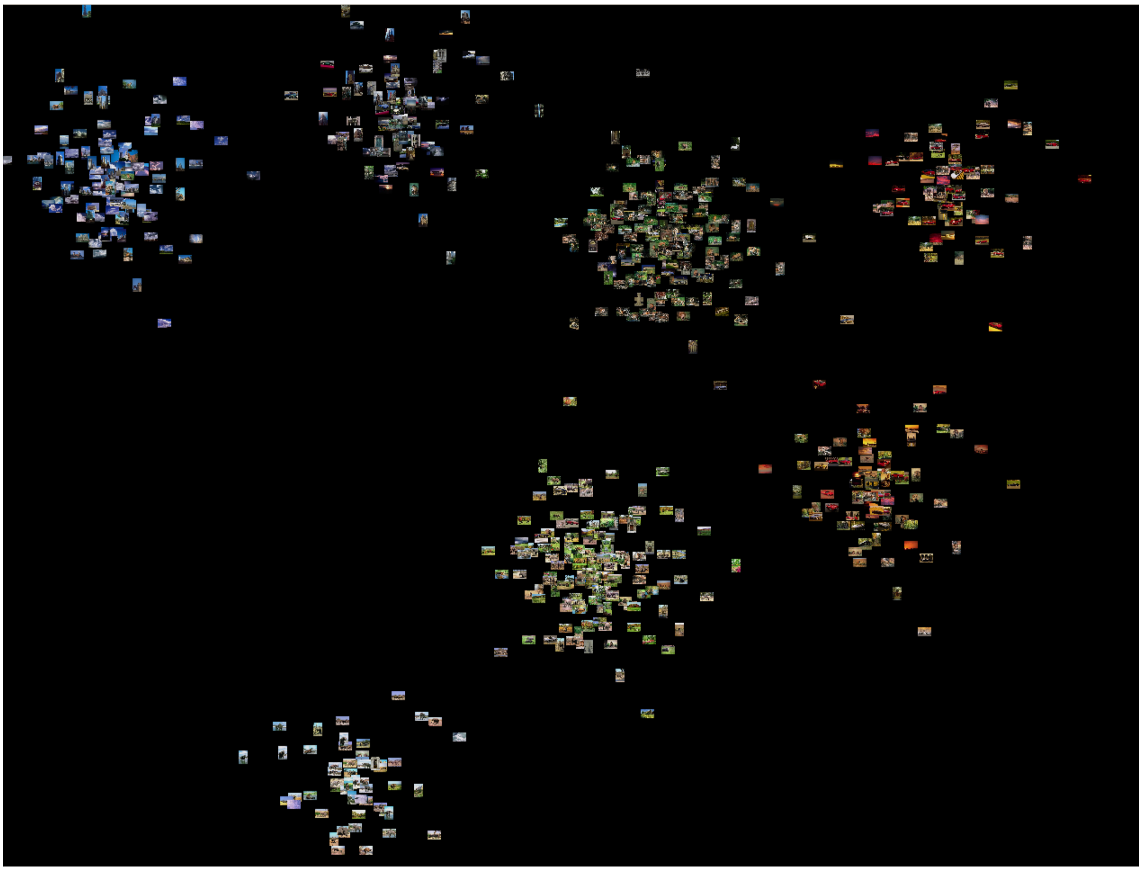

Figure 4.10: Screenshot of the ClusterVis layout with the *visual cluster* radius $r = 50c$.

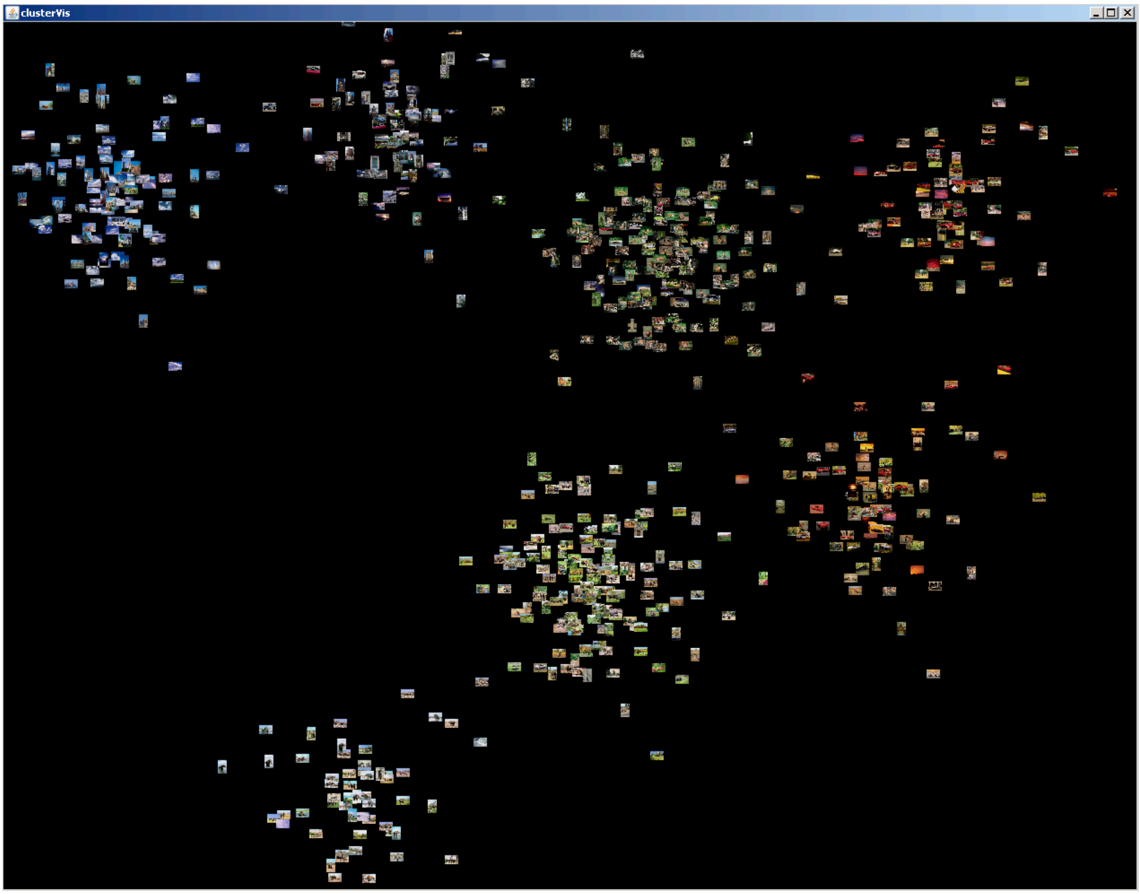

Figure 4.11: Screenshot of the ClusterVis layout with the *visual cluster* radius $r = 60c$.

4.5 Evaluation

The main objectives of the evaluation were to determine the grade of complexity of the proposed layout and assessing whether the generated visual content overview is able to provide a meaningful match between the system and the "real word". Specifically, a quantitative comparison was used for evaluating the layout algorithm complexity, while an informal qualitative analysis was applied for assessing the visual presentation of the content.

In the first part of the evaluation we compared the proposed layout complexity with state-of-the-art MDS-based layout algorithms. As a comparison metric the algorithm complexity value was used. After performing a quantitative analysis of layout complexities, results presented shown that the lower complexity of the proposed algorithm leads to a faster display of images, which is an important feature for a broad spectrum of image retrieval systems where rapid image display is required. In terms of usability, in particular for what concern the visual aid to retrieval, a formative evaluation was performed by analysing the visual properties of the display. In the specific, the quality of visualisation was assessed by examining the spatial arrangement of images on a semantics base. Considering that the layout was generated on the base of low level features, applied heuristic type of evaluation enables to investigate how well is visualised the match between the system and the real world [182]. This part of evaluation was performed by subjective comparison of the proposed layout with equivalent MDS-based visualisation approaches (which adopt a similarity based layout based on colour).

The first segment of evaluation compared the performances of all analysed layouts in terms of the algorithm complexity, similarity preservation and overview provided. In the second part of evaluation, making a parallel into the visual domain was done and the evaluation of the proposed layout was performed by comparing it with the common methods for similarity based visualisation in terms of inter and intra-cluster information provided with the proposed layout.

4.5.1 Comparison of discussed layout algorithms

Table 4.1 shows a comparison between the previously discussed layouts in terms of: method for placing the cluster centroids; method for placing images in each cluster; complexity (ticked if they have low computational complexity); inter-cluster similarity preservation; intra-cluster similarity preservation which includes: image-to-centroid and image-to-image similarities; and visual summary (if it provides intuitive collection overview).

No.	global	local	low complexity	inter-cluster	image-centroid	image-image	visual summ.
1	-	-		x	x	x	
2	max dist.	uniform rank	x	x			x
3	MDS	uniform rank.	x	x	x		x
4	MDS	spring MDS	x	x	x		
5	MDS	MDS		x	x	x	x
6	MDS	normal rank.	x	x	x		x

Table 4.1: Comparison of proposed layout solution in terms of: complexity, inter-cluster, image-to-centroid and image-to-image similarity preservation and provided visual overview in terms of main collection topics.

It can be seen that the last layout method, which combines clustering, MDS and ranked normal random placement outperforms most of the other layouts. The only "comparable" layout is the 4th one, where MDS is used separately for placing centroids and later for mapping images per cluster. However, this layout preserves image-to-image intra-cluster similarities at the cost of higher algorithm complexity.

4.5.2 Global similarity comparison

As previously stated, a common user[2] is not interested in examining the feature vectors of images or any other low-level form of image interpretation. What a user wants to find is information on the level of high semantic such as “lion”. This part of evaluation focuses on testing the proposed layout in terms of image locations from the global perspective. In other words, compare the location of images between the standard similarity based visualisation approach, such as “full MDS” and the ClusterVis layout proposed in Section 4.4.3.

Considering this, the first set of evaluation concerns the comparison between collection overview generated by applying “full” MDS and the ClusterVis layout. The two layouts are shown in the Figure 4.12.

Examining both layouts, the inferred conclusion that different groups of images can be identified based on their dominant colour. It is our oppinion that the user can initially utilise the same strategy of exploration directed by the colour within both layouts. The only visual difference is that, due to different generation procedures, the ClusterVis layout is rotated compared to the “full MDS” layout. In ClusterVis the “blue” cluster (with images of sky, cloud and/or water) is positioned in the upper left corner while the same set of images is in “full MDS” visualisation spread across the right side.

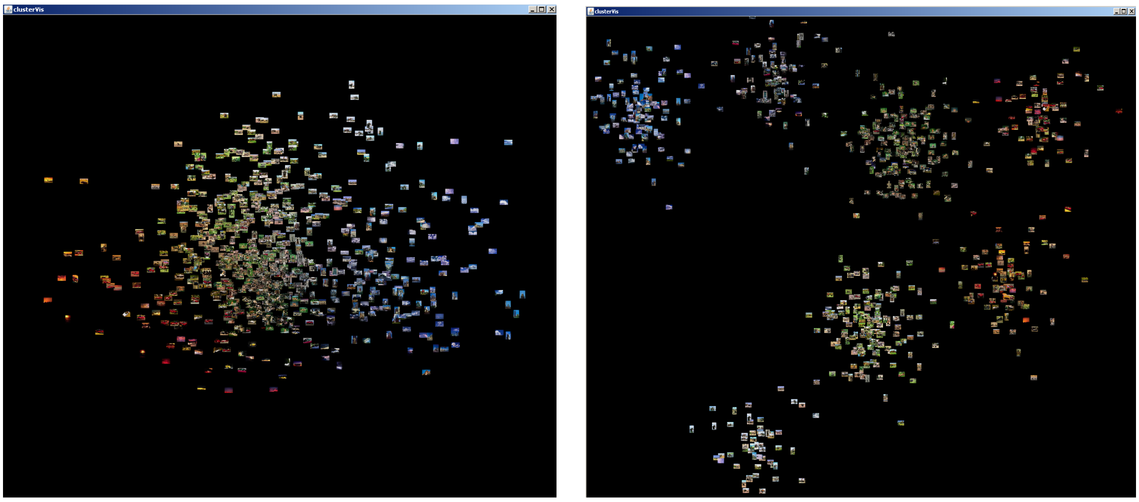

Figure 4.12: “Full MDS” vs ClusterVis layout of 700 Corel images.

The **first test** of comparing semantics conveyed, consisted in examining the positions of image group in the compared layouts. Image groups were selected within the “full MDS” layout after which their position was located in the ClusterVis layout. Although the positions of various image groups were compared, the most interesting

[2] A common user is a person who has no in-depth knowledge on multimedia and is interested in multimedia only from semantic perspective

are the groups whose position in “full MDS” layout are on the “crossings” between different parts of the MDS cloud. For example, as seen in the Figure 4.13 the group between blue and the central (green) region.

Here two examples of such comparison are displayed. Figures 4.13 and 4.14 show the positions of the first selected image group, found in the MDS cloud in between blue and the central groups, identified as a “mostly green coloured region”. Several images (with IDs 212094, 191081, 212019, 212034, 212068 and 29082) were identified inside the first test region for comparison between layouts. The same images were then located within the ClusterVis. It can be noticed that in both layouts these images are placed together and both times in-between a “blue” and “green” set of images.

The comparisons for the second selected group of images, for both “full MDS” and ClusterVis is shown in Figure 4.15 and 4.16, respectively. As previously, a number of images (105015, 150093, 150048, 150049 and 108031) was abstracted from the region 2. In this case, there is a difference between the two layouts since these images are not localised together in the ClusterVis implementation. The region 2 is in the MDS cloud in between “red” and “green” zone, whereas in the ClusterVis same images are in clusters containing green or mixture of red and green colour. However, observing in more details it was noticed that the image 108031 (showing tiger, water and green vegetation) was in the MDS cloud placed with cars, buildings and some images with water. In ClusterVis the image had a similar neighbourhood but in addition to this, contained semantically similar images (with tigers and lions). In addition to this, this group is in between the green and red part of MDS cloud whereas in the ClusterVis they are separated between green and red clusters. This indicates better separation of “less similar” coloured images.

Based on these examples, it can be inferred that global similarity of image content is preserved by preserving inter-cluster similarities. In other words, visual separation of content into clusters arranged according to their mutual similarity will not negatively influence image positions. Even so, by separating distinctive groups (based on colours) it enables easier location of images that are “hidden” in central part of MDS cloud.

The **second test** consisted of finding multiple random images in the collection and locating them in both layouts. For example, the image of lion (ID=105032) was located first in the “full MDS” layout as shown in the Figure 4.17. The image location is circled in red. Figure 4.18 displays the location of the same image in the CLusterVis layout highlighted with red circle line. As evident from the shown screen shots, the position of this example image is in both cases located in “blue” regions. This implies that from the visual point of view the information is interpreted and

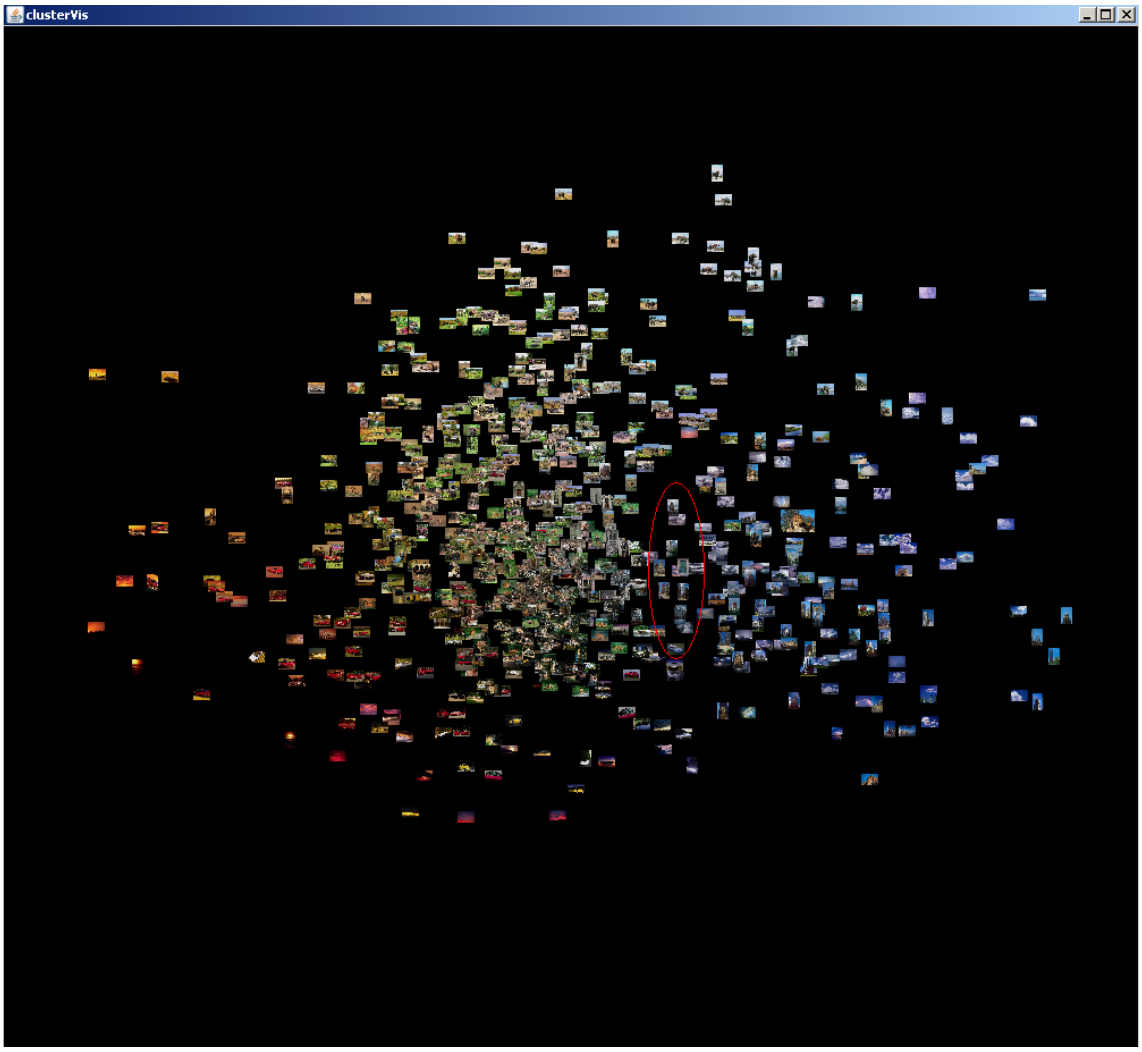

Figure 4.13: Image group 1 located in “full-MDS” layout (circled in red).

placed equivalently. Examining the same effect through other examples we consider that these two layout are more-or-less equivalent.

The **third test** was the comparison between the neighbourhood of images in both layouts. This was investigated from semantic perspective, since random layout approach might have influenced the neighbour images of one image. Again, several images are taken randomly and their positions located in both layout solutions. Then, the images located in their proximity were analysed in terms of their semantic content.

Example of this analysis is shown with Figures 4.19 and 4.20. These figures show a distorted view of the image 105032 and its neighbours in the complete MDS and ClusterVis layout, respectively.

It was inferred based on visual analysis that the standard similarity-based layout in this example does not show superiority over proposed ClusterVis layout. The results of the rest of the analysis performed, showed similar results, thus it was

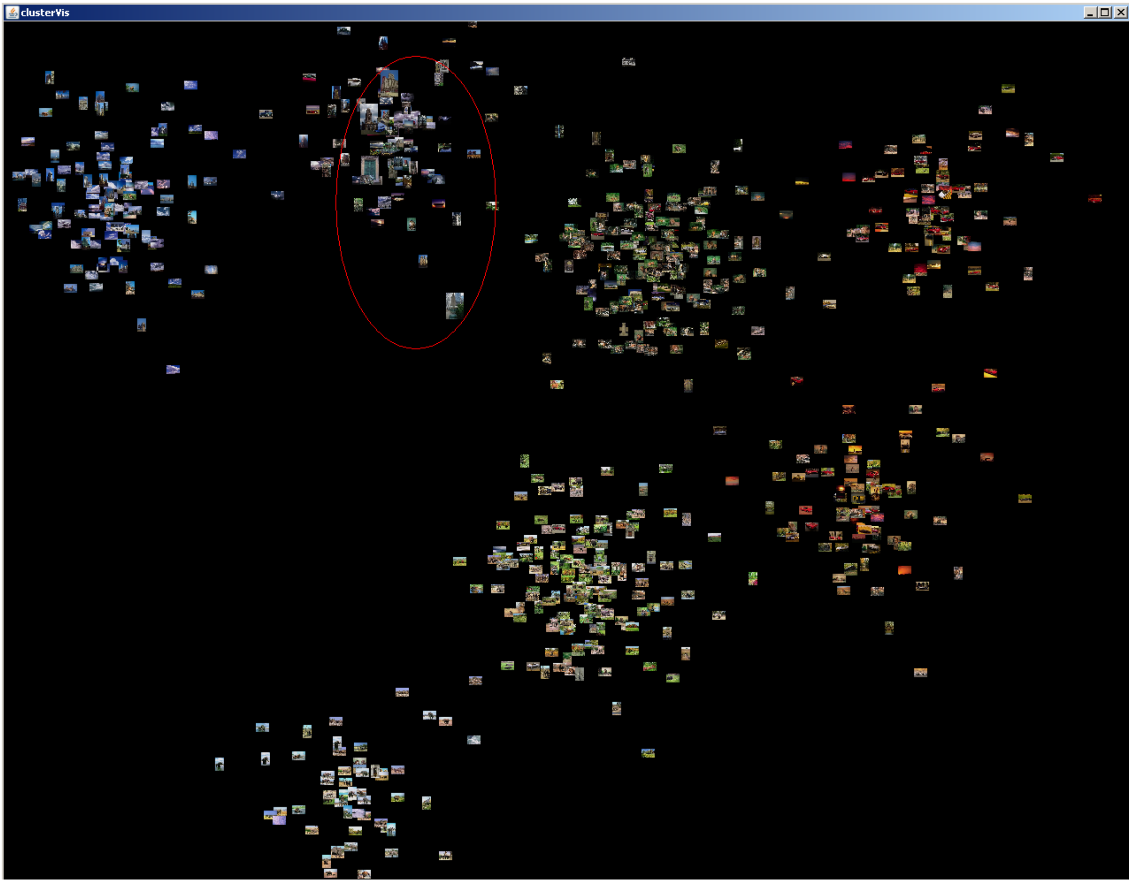

Figure 4.14: Image group 1 located in ClusterVis layout (circled in red).

considered that proposed ClusterVis solution is highly competitive to the “full-MDS” layout even in this context.

4.5.3 Local similarity analysis

The second part of the layout evaluation was done to establish how does the ranked normal random mapping of the images in one cluster affect the local similarity distribution. In other words, how different is the image position in one cluster in cases of MDS-per-cluster layout and ranked random layout per cluster. MDS-per-cluster refers to the approach described in Section 4.4.2 where MDS is used separately to place cluster centroids and then for images per each cluster.

The first part of the performed comparison is a general comparison between layouts. Comparing the layouts in Figure 4.21 and Figure 4.22 there is no obvious difference. Both layouts show several image groups with more-or-less distinctive dominant colours. It can be concluded that the image collection overviews provided with compared layouts visualise the collection equally.

The second part of layout comparison is performed by observing the position of selected images in both layouts. Since the ClusterVis adopts ranked random layout

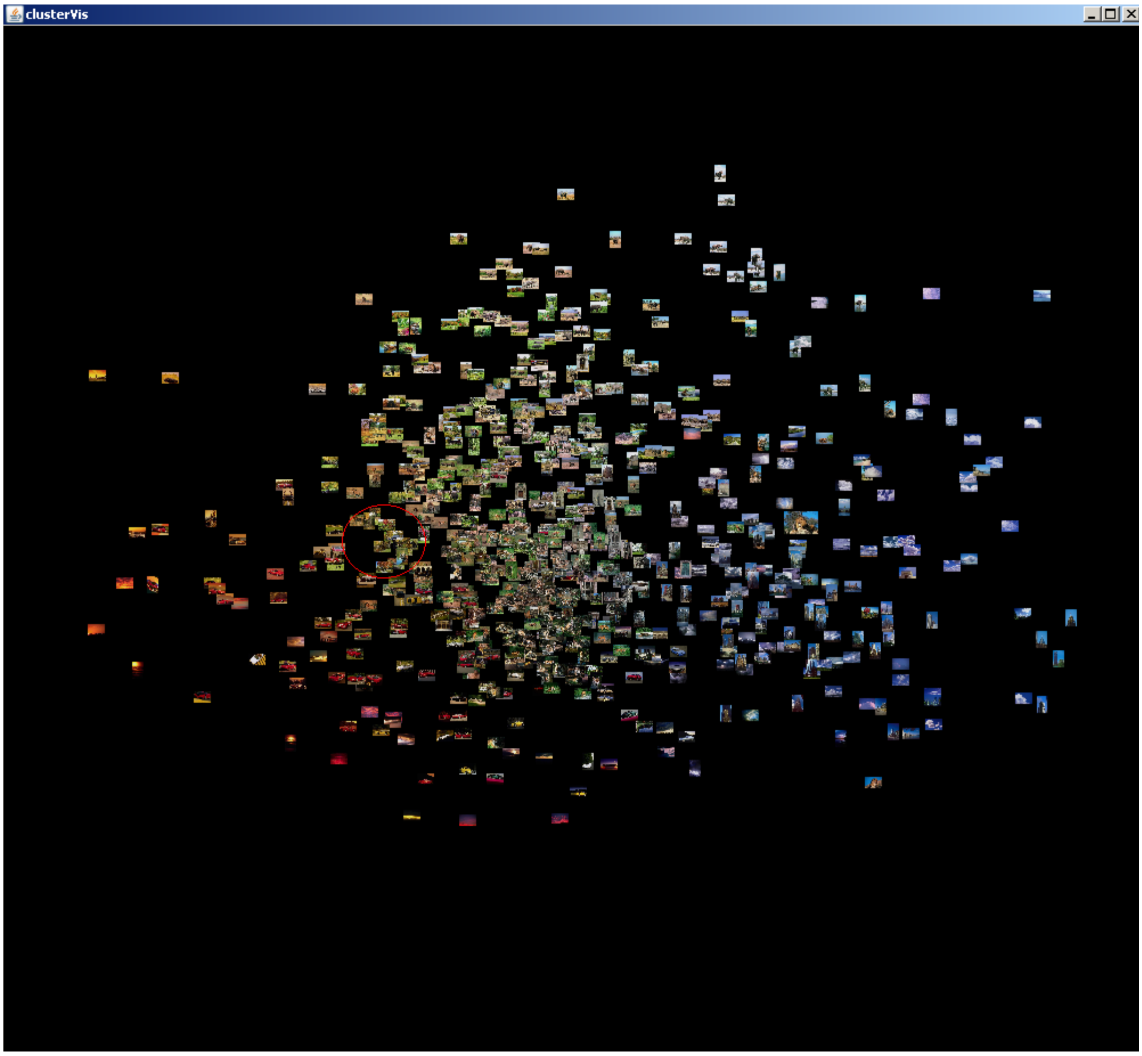

Figure 4.15: Image group 2 located in "full-MDS" layout (circled in red).

approach which ommits the inter-image similarities it was interesting to see how does this affects the image layout. Through several examined examples it was concluded that when examining one cluster no significant difference was noted. Figure 4.21 and 4.22 display the the positions of one example image (with ID 212017) which is bordered with red circle. Figure 4.21 displays the layout where positions in one cluster are generated using "MDS-per cluster" and the Figure 4.22 layout where placement per cluster is random. This image was randomly selected from the cluster. It can be seen from the screen shot that in both cases the image is on the outskirts of the cluster which means that the difference in its location is ignorable or in other words, that two layout place one random image "quasi" equally. It can inferred that ClusterVis approach, although does not preserve similarities entirely, like the first one, does not loose any important information.

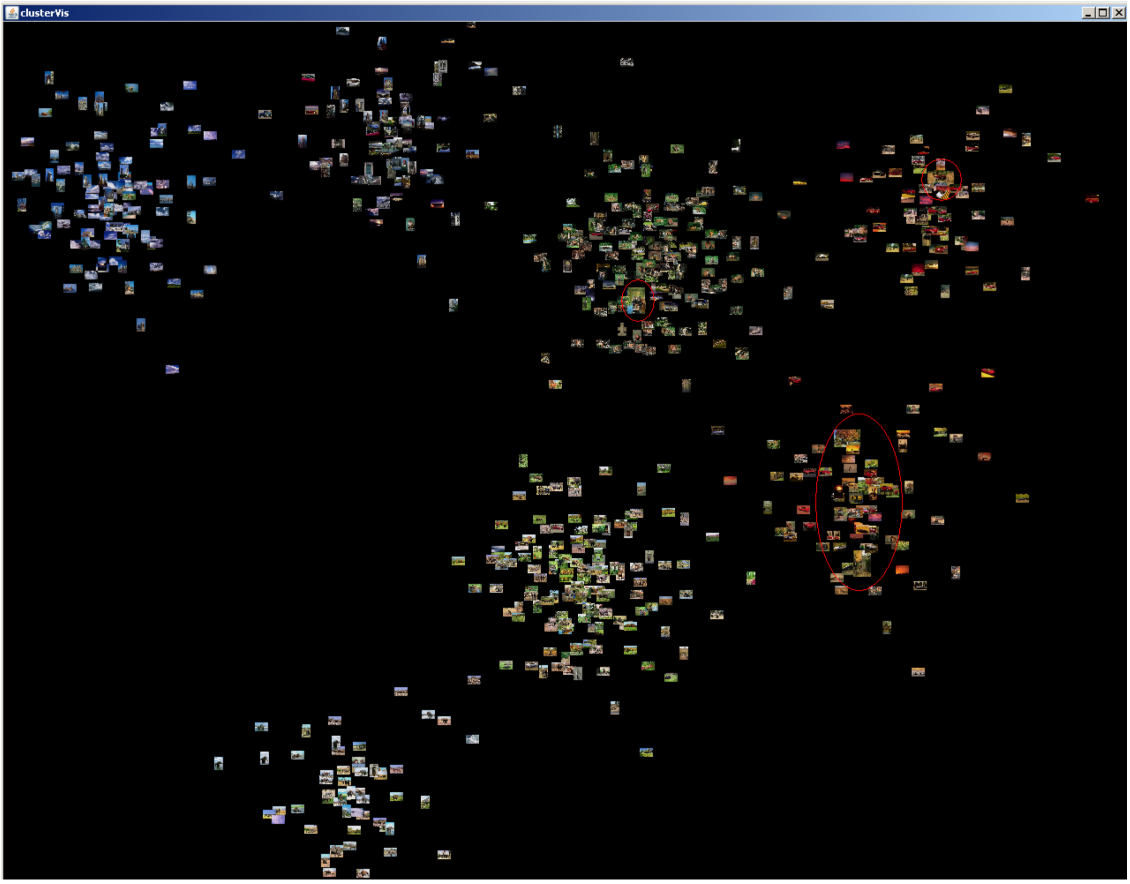

Figure 4.16: Image group 2 located in ClusterVis layout (circled in red).

This chapter has been dedicated to the discussion regarding several similarity-based graphical layouts and their ability to provide an intuitive overview of the image collection. Starting from standard 'full-MDS" layout approach, various alternative method were analysed and novel layout method developed.

Results obtained by the performance comparison clearly indicates the reduced algorithm complexity of the proposed ClusterVis over the other analysed layout methods. This conclusion is directly supported by a layout complexity measure which enables an objective quantitative comparison of each approach. The quality of layout, in terms of the visual content overview it provides, was assessed using a qualitative comparison with related state-of-the-art similarity-based visualisation approaches. The inferred conclusions are that different groups of images can be identified based on their dominant colour and that the user can initially utilise the strategy of exploration directed by the colour within both layouts. A similar approach was used for evaluating the preservation of local and global similarity information. With the goal of analysing if the proposed solution changes significantly the position of images with respect to their semantic neighbourhood, a qualitative comparison was performed between ClusterVis and related state-of-the-art visualisation solutions. In this case the inferred outcome is that, preserving the inter-cluster

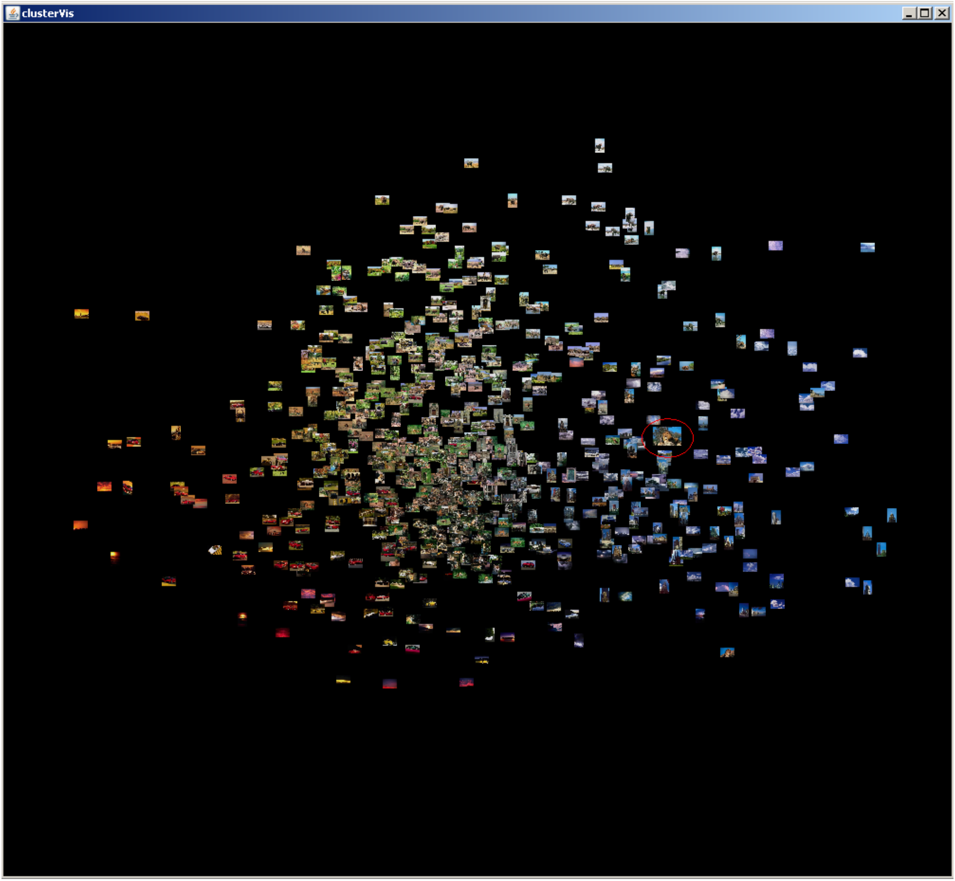

Figure 4.17: The position of image 105032 in "full MDS" layout.

similarities preserves the global similarity of image content. and from the visual point of view the content is placed equivalently. Based on the visual analysis of local image arrangement (in a cluster) we consider that ClusterVis approach, although does not preserve inter-image similarities, does not loose any important information in terms of image placement.

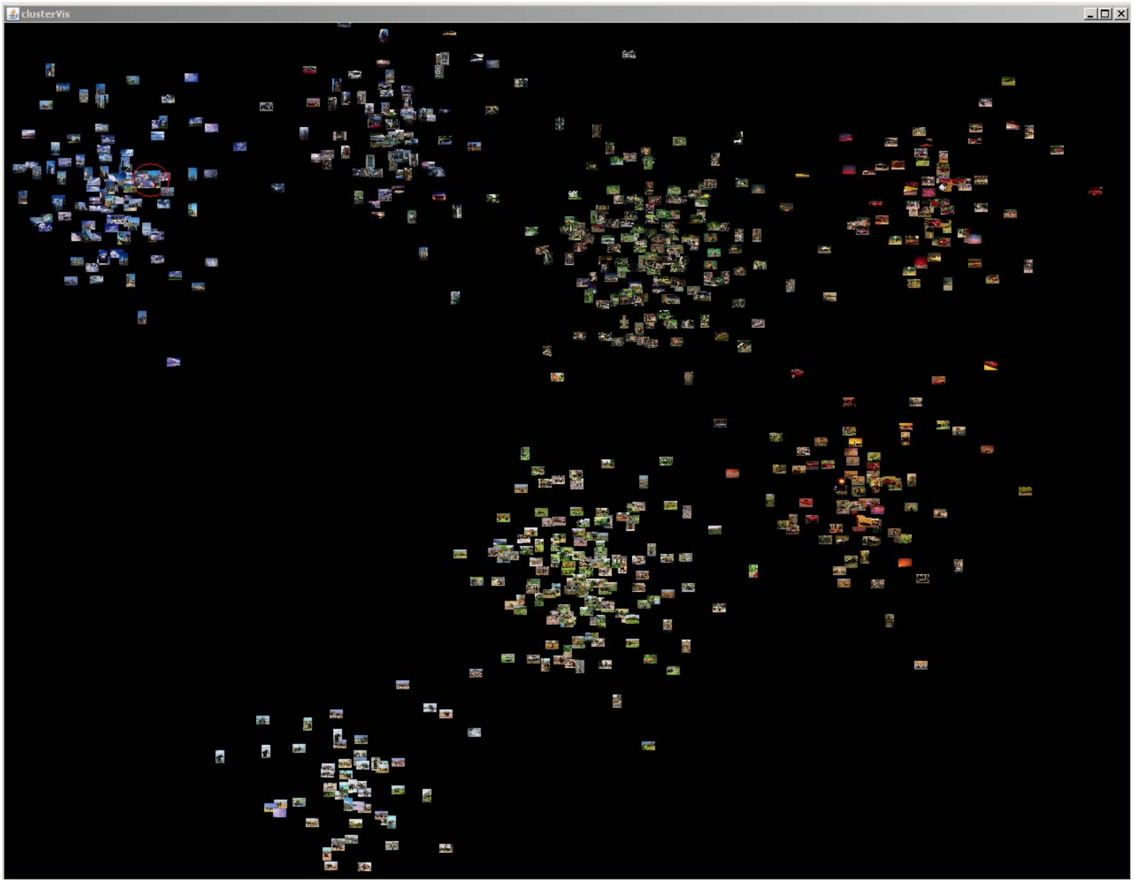

Figure 4.18: The position of image 105032 in ClusterVis layout.

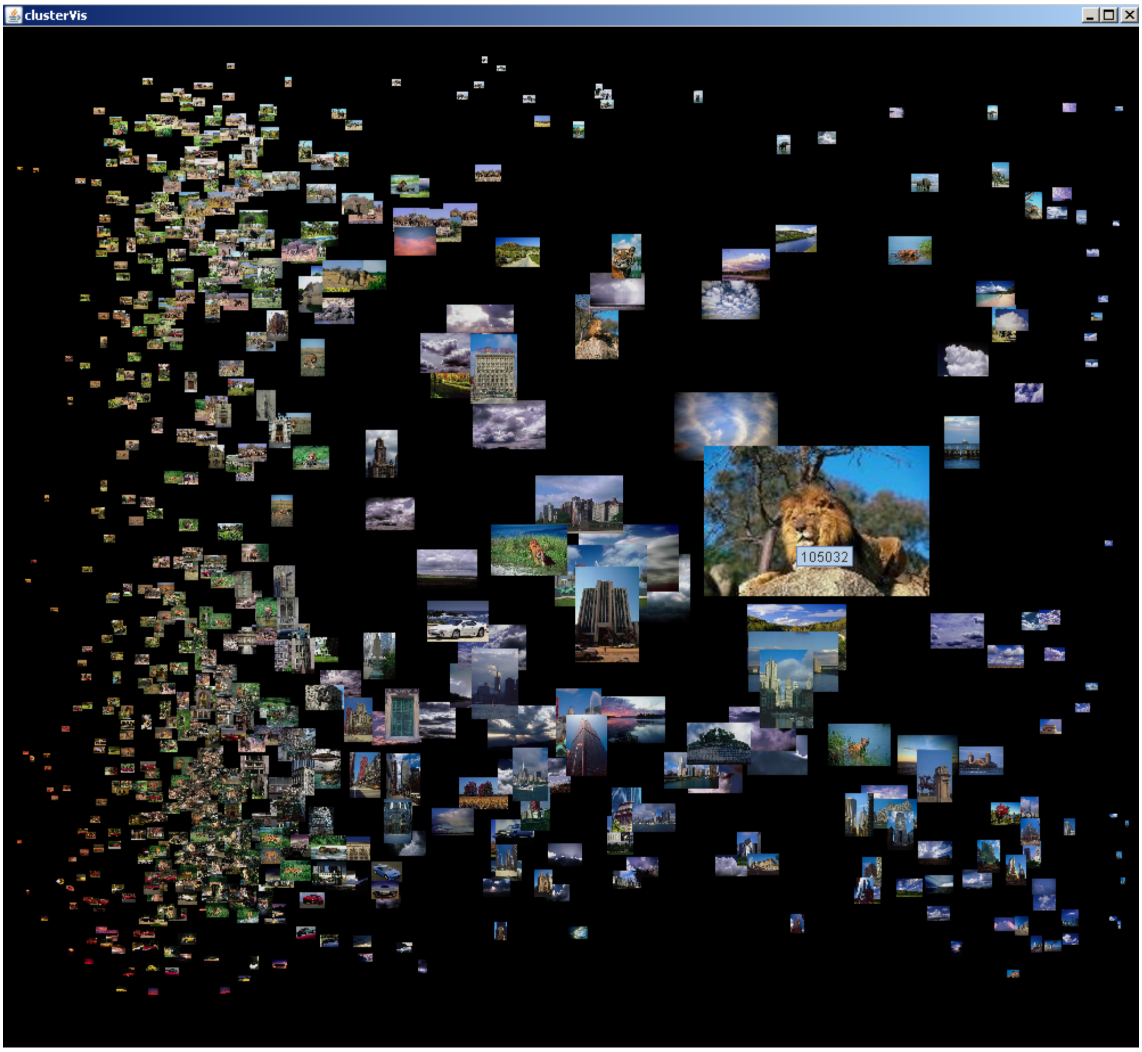

Figure 4.19: Neighbour pictures of image 105032 with lion in “full MDS” layout.

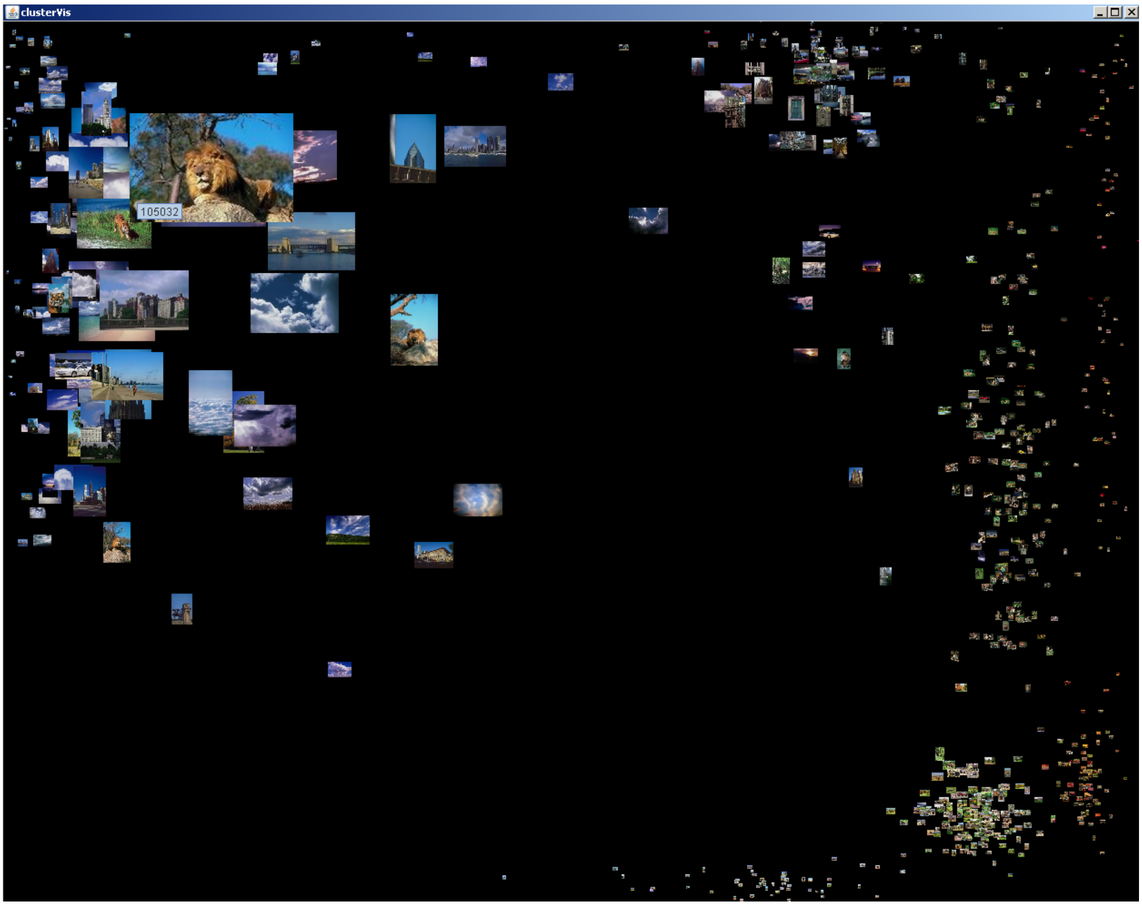

Figure 4.20: Neighbour pictures of image 105032 with lion in ClusterVis layout.

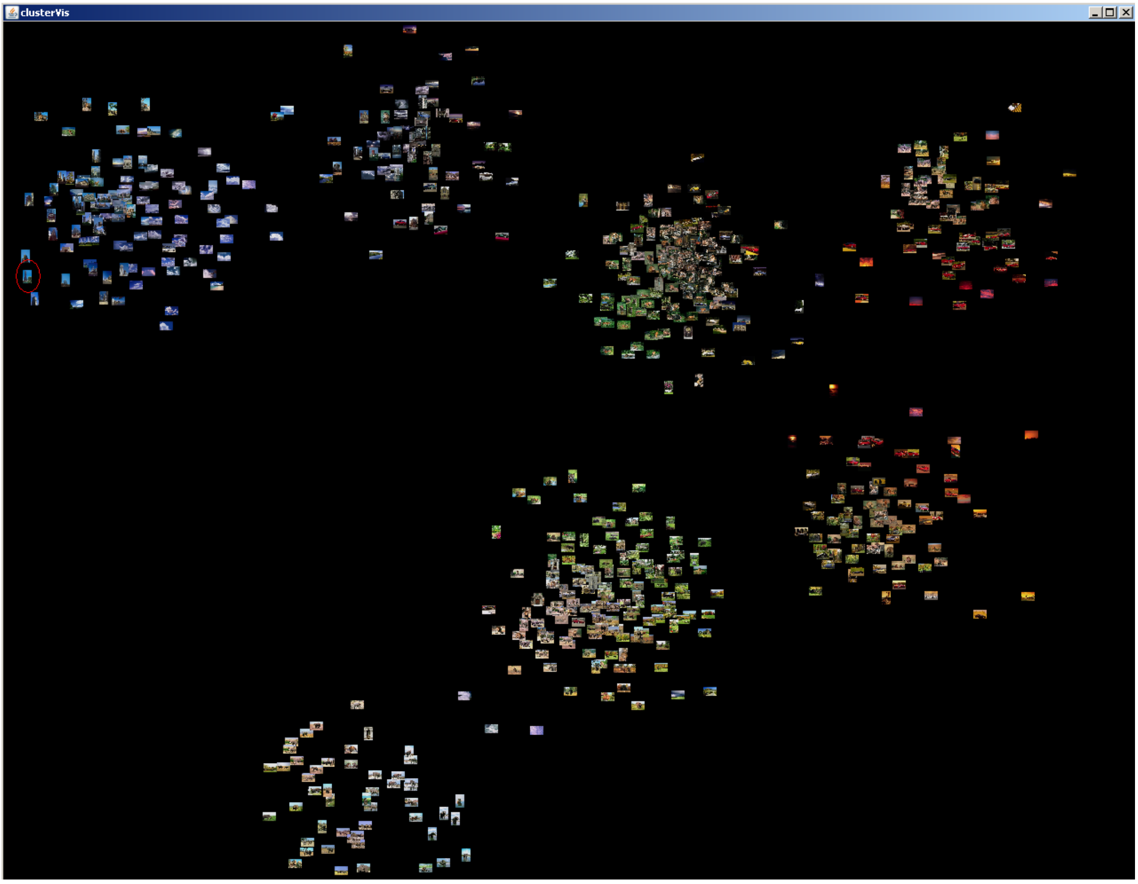

Figure 4.21: The position of image 212017 in MDS-per-cluster layout.

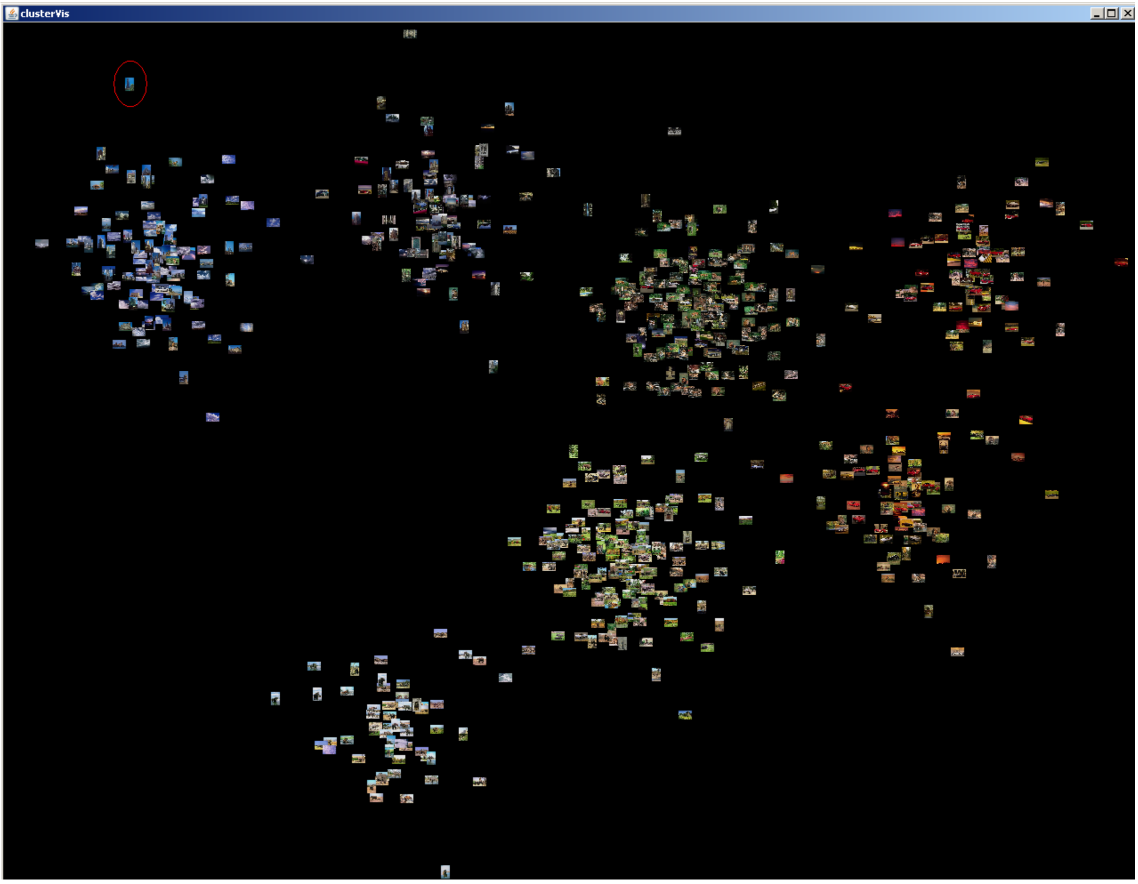

Figure 4.22: The position of image 212017 in ClusterVis layout.

Chapter 5

Hierarchical visual browser

The Hierarchical browsing application described in this chapter was developed as a part of the dedicated system for "delivery of and access to raw media material (rushes), as well as its reuse in the production of new multimedia assets" [61].

The developed system was intended to support two types of users, professional and home users, in accomplishing retrieval task over the large broadcaster archive of videos. Semantic-based retrieval tool, also implemented within the RUSHES system, was not able to cover all expected users tasks/queries since the content consisted of mainly un-annotated video material. For this reason, an additional retrieval tool was required development to enable content based exploration.

The developed complementary retrieval method is based on low-level visual image features and uses hierarchical clustering algorithm for its organisation.

It is important to note that the analysed content was a collection of raw videos, full of repetition so that image representatives (key-frames) extracted for their summary are often very similar among each other. Considering the quantity of content in the broadcaster's archive and its repetitive nature, a hierarchical organization of such content is helpful since it organises the content in smaller organizational structure which should assist the retrieval and it organises the repetitive content by means of collocation.

For example, in the case of video archives, a professional user could efficiently browse all "sport" segments if these are stored separately from the "news" (see Figure 5.1). Then, on a lower level, it might be helpful for exploration to separate video material coming from different sports, such as "tennis" and "football"; similar considerations can be iteratively repeated through lower levels.

On the other hand, the provision of a data-base content preview (or summary) as a collection of representative images (or key-frames in case of videos), fully matches with the browsing strategy, as they both allow for fast exploration of large unknown content sets, without the requirement to specify a query in advance. In addition

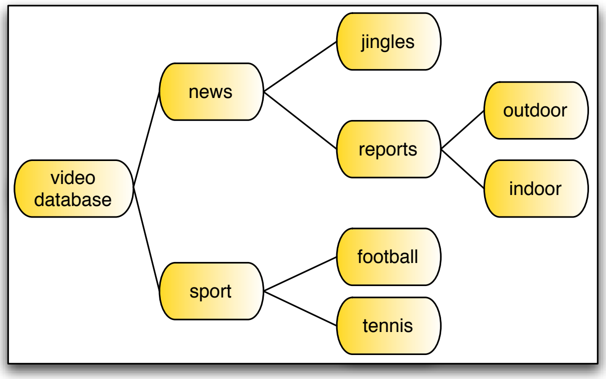

Figure 5.1: Hierarchical arrangement of a video/image database.

to this, the availability of hierarchical previews is relevant since (i) it supports the user in quickly gaining an overview of the unknown content set, (ii) it organises the material by similarity, and (iii) it selects representative content for visualisation.

Since the content was mainly un-annotated the search using Hierarchical browser was a combination of explorative search and query-by-example. The user tasks defined in RUSHES project are given in [10] and as stated there 'based on the performed analysis and the overall architecture of the RUSHES system, gathered information was combined in order to define requirements and use cases that capture the user input and provide a view on the functionality of the retrieval application". According to this, Hierarchical browser supports professional user requirement number 5: "*Users think that it would be useful to search for further clips based on a set of pre-selected images, by selecting one image and requesting the application to search for clips that contain images similar to the selected one*" (explorative and query-by-example) and user requirement number 6: "*the users want to have the possibility of searching for specific visual content, as for example: landscape, cityscape, outdoors, indoors, sunset, etc (explorative query)*". Based on list of user requirements the same document lists examples of more detailed user tasks derived from user requirements.

Considering the type of the content, content organisation and specified user requirements, the proposed Hierarchical Browser provides visual and interactive tools that support explorative search and retrieval of content based on visual similarity. In other words, the image retrieval is supported by visual content examination and content-based search.

For example, journalist making a documentary about the football club "Atletic Bilbao" needs to retrieve video material taken in the past regarding related to the topic. Since the material is mainly un-annotated he can perform a content based, explorative search in order to find the material for his documentary. Knowing that "Atletic Bilbao" shirts are striped red and white he can focus on that detail while exploring the content. Once he finds one image with " Atletic Bilbao" player he can perform similarity based search using the hierarchical content organisation.

The user types considered are both users who know that database contains relevant content but do not know exact location, but also users with no knowledge of the content looking if there is something relevant to their task. Considering user profiles application targets both experts and non-experts (professional and home users).

5.1 Requirements for managing hierarchically organised data

An ideal arrangement of images or videos in a hierarchical tree such as shown in Figure 5.1, where each node is labelled with one *semantic* category, would ideally enable a fast access and a complete understanding of the database structure. In case of having semantic labels associated with each data item, the creation of meaningful groups of data content would not impose extreme difficulty. The absence of semantic information (or image meta-data) requires a different approach based on low level, visual features.

The proposed hierarchical summary is therefore obtained by a visual clustering of the N images (or key-frames) of the database.

Each level L_i of the hierarchy, from root L_1 to level L_n, contains the whole set of N images (or key-frames) organised in an increasing number of visually similar clusters, *i.e.* $L_i = \{C_{i,1}, C_{i,2}, \ldots, C_{i,\chi_i}\}$ where χ_i is the cardinality of L_i, as shown in Figure B.1. Such organisation enables structured exploration: once the user identifies an interesting image (key-frame), he/she can interactively request more similar content from the same cluster, or refine his/her search by descending into the hierarchy, thus restricting the scope of his/her quest.

Even if the grouping of similar content is based on visual similarity rather than semantics, the proposed arrangement assists the browsing process by reducing the semantic gap between low-level features and high-level concepts familiar to the user.

Considering the hierarchical organisation of the content, the most straightforward way of presenting it to the user is applying a hierarchical visualisation approach which provides, at the same time, an easy initialisation of the browsing. A number of information visualisation techniques for displaying hierarchical structures have been proposed in the past and are discussed in Chapter 2. These visualisation solutions, though helpful for solving the problem of visual clutter for large numbers of tree-nodes, does not always serve as efficient aid for the user, since the excess complexity of the user interface sometimes induces an additional obstacle for performing the browsing task. Regarding the proposed visualisation, the first aim is therefore to keep a low-level of complexity, while preserving the information about the structure.

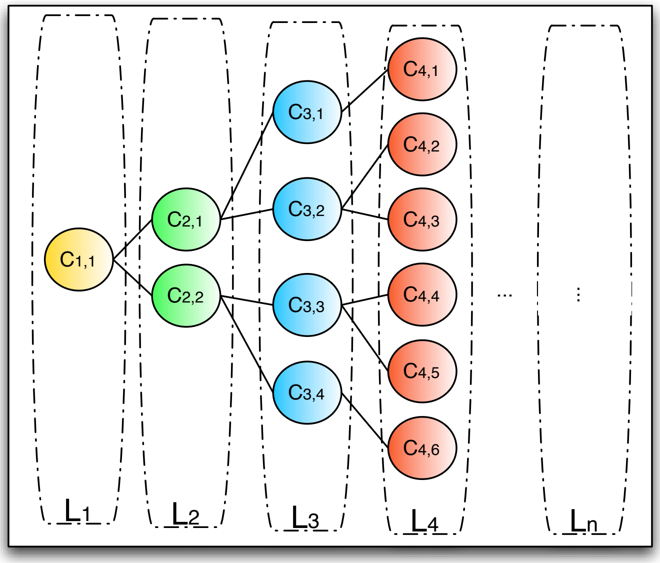

Figure 5.2: Structure of the hierarchical preview.

As explained previously in Chapter 3, most hierarchicaly-based solutions for image retrieval have low level user support especially for initialisation of the content exploration. The second aim is therefore to assist user's initialisation of browsing or content access.

Standard approaches such as limiting the set of the displayed images or reducing the image sizes in order to present them all [24], [164] and [109] negatively affect the beginning of the browsing, since the user might not find the relevant image or their size is too small to understand the content. For this reason the visualisation solution proposed here aims at providing visual support for the user, so that he can understand where to find items of interest (and to reduce the time for browsing initialisation) by displaying a tree map of the content space together with content previews. The tree map is generated as a graphical representation of the hierarchical organisation of the repository while content preview is set of images extracted from each node the user is exploring at specific moment. This is particularly important considering that due to the existence of *semantic gap*, images as visual objects carry the most important information.

5.2 Visualisation of global and local information

Considering the hierarchical organisation of the image repository, there are two levels of information to be displayed: global and local. Global information refers to the structure of the data-set whereas local implies the content and information related to one node. For this reason graphical user interface (GUI) consists of two windows which enable two views: (i) global, which enables exploration of the content

Figure 5.3: Screen-shot of the interface. In the upper window: the tree view of the database structure, with the coloured search path to the selected node; the two buttons "Explore database" for random access functionality and "New set of key-frames" for requesting more content from the same node. In the bottom window, the grid with node preview images is displayed.

by interactive search and (ii) local, which shows the preview of the explored content. A screen-shot of the whole user interface is shown in Figure 5.3.

5.2.1 Visualising global information

Global, structural information is displayed for three reasons:

1. To support the user's understanding of the organisation of image repository;
2. To enable the user to track his/her location within the structure;
3. To see the parent-child relationships between the content.

The method for displaying the repository structure used in this work is *tree visualisation* as shown in Figure 5.4. The tree view is chosen due to it's broad application for representing hierarchical structure which implies the fact that large number of users are familiar with it's meaning. The interactive properties of the tree display are discussed in Section 5.3.

The tree visualises the hierarchical preview of the collection and in case of large data-set it might consist of a huge number of nodes and branches. Due to the

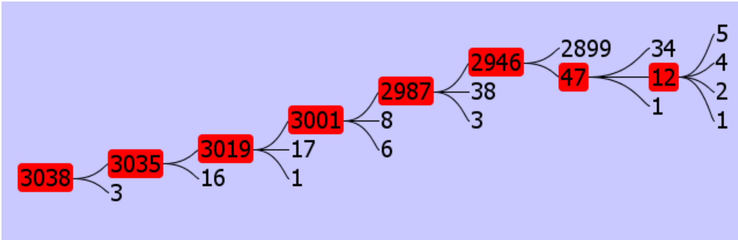

Figure 5.4: Tree view of the database structure (upper window).

limited display dimensions, visualising the entire tree at all times is space-consuming and unnecessary. The absence of semantic information implies that no important information can be conveyed by displaying the whole tree, especially in case when the nodes close to root are explored. An optimal solution adopted here, is to dynamically change the number of displayed levels. A node can be expanded (upon the user request) to reveal its child-elements, if any exist, or collapsed to hide departing branches and child-elements. The user does not see the entire tree while browsing, but only the set of nodes relevant to currently selected one. The number of levels displayed at the same time is adaptable and can be changed at any time.

To highlight the user's search path and parent elements of currently selected node, all nodes starting from the root $C_{1,1}$ to the currently explored node $C_{i,j}$ are colour-encoded. Considering that there is no semantic meta-data on the observed data collection, the only information potentially useful to the user is the number of images contained in a node. This information is visualised as a numeric node label (see Figure 5.4).

5.2.2 Local preview of a node content

As previously stated, the global information related to data-set is accompanied by local information in form of a content preview. The content preview is considered essential due to primitive level of image index information. In such case the only semantic, relevant information are images themselves. By viewing images the user becomes aware of the content domain.

When clicking on a certain node, the selected item in a tree is coloured in red, the next level child-elements are shown, and the related preview is accessed and shown in the lower window, as shown in Figure 5.5. The set of images belonging to the currently visited node is displayed on a grid. Strategies for accessing the preview and the selection of the preview images are described in Section 5.3.

The content preview is shown in a grid with adaptable number of cells where the choice of a number influences image size in a cell. The goal is to optimise the number of cells so that image size allows the user to see the visual content summary without

Figure 5.5: Preview images of the currently selected node (lower window).

the need of any additional interaction (zooming, etc.). The initial implementation had 21 cells which was marked as too little when the initial professional evaluation was performed (more details are given in the Section 5.4).

The aim of the visual summary is to show a set of representative images for each node in the hierarchy. There are several approaches to obtain the representative set such as showing the image centroid [164], or showing one image for each child node [109]. However, they cannot guarantee optimal selection of a representative image set. For example, observing the strategy adopted in [109] in cases when there are two nodes, only 2 images will be displayed and the display space wasted even though there are more images belonging to this node. A similar approach to the one presented in [24] is adopted in this work. The preview set of images are randomly extracted from each selected node. However, the potential drawback is the probability that content of some child-node will not be represented but on the other hand processing time is shorter. In addition to this the goal is to represent the content of a node itself, since the exploration of child nodes can be performed very quickly just by clicking at them. This implies that the stated issue is compensated by interactive support.

Although a restricted number of images can not completely represent the content of a node, this approach avoids information overflow or inadequately re-sized images. Again interactive interface elements can help to avoid information loss. In case the user wants to perform further exploration on the same node, he/she can request for additional content, and a new random set is then extracted from the node and displayed.

5.3 Random exploration and content access

Since image databases are in most cases too large to be explored in a single step, support needs to be provided for a "virtual tour" through the repository. Process of information seeking requires a meaningful initialisation and the ability to change according to new knowledge acquired by the user regarding both the structure and the content. In an environment with no semantic information for efficient exploration the user should be able at least (i) to find one item that interests him/her to start the navigation with (ii), to identify successive steps by which to proceed and (iii) to easily access and grasp the repository content.

In order to meet these needs, novel solutions regarding (i) random exploration, (ii) interaction methods and (iii) content access, are proposed here.

5.3.1 Random exploration

Analysing user interfaces for content-based image retrieval one of the evident issues is initialising the search/query for relevant content. In case an interesting image is not presented at first, most users often perform some "random" attempts at exploration, if possible. In a large number of content-based retrieval systems this is the only way to locate some content that might suit their information needs. These attempts are usually performed using interaction tools designed for other purposes and no designated support is provided by the system in this stage (in a large number of cases found in literature). In other words: the user is "on his/her own".

In order to address the stated issue, a novel way of identifying interesting search directions is proposed here. It is accomplished by providing a novel interactive functionality, called *biased random exploration*. The proposed functionality is designed specifically in order to imitate user random behaviour in situations where potentially fruitful directions cannot be easily identified. The adjective "biased" comes from the fact that the behavioural model is not completely random, since it tries to "help" the user, as explained below.

So, in case that the content of a currently explored node is of no interest for the user, the application supports random selection of different node by *biased random exploration* functionality. After the node is selected its preview is displayed in the bottom window providing new, potentially helpful knowledge to the user.

Initial "random" selection of a new node was obtained in a two-step process. In the first step one among the n available levels in the hierarchy was randomly selected; the second "random" selection picks one of the existing nodes belonging to the previously chosen level. This means that the algorithm accesses one level L_i of the hierarchy according to a discrete probability density function which is uniformly distributed among all levels, so that:

$$p(L_i) = \frac{1}{n}\,,\ \ \forall i \in [1, \ldots, n]. \tag{5.1}$$

The previous implementation employed random selection even in the second step. After the level L_i has been selected one node belonging to this level is chosen according to a discrete probability density function which is uniformly distributed among all nodes on the same level, *i.e.*

$$p(C_{i,j}) = \frac{1}{\chi_i}\,,\ \ \forall j \in [1, \ldots, \chi_i]. \tag{5.2}$$

where χ_i is the cardinality of level L_i. As described in the evaluation part in Section 5.4, this two-step strategy enables the user to find in a reasonably short time interesting new search directions, even in case of huge databases.

The very first implemented strategy, a one-step random selection of a node $C_{i,j}$ according to a probability density function uniformly distributed among all nodes of the hierarchy (so that $p\left(C_{i,j}\right) = 1/\sum_{i=1}^{n}\chi_i$) has been discarded after a few experiments. In this case in fact, since the cardinality of levels increases going from the root level towards the leaves, the probability to pick a leaf-node would be much higher than that to select a more populated node at a lower level. The quantity of information obtained by viewing the preview of a leaf-node is smaller than the quantity of some higher node in the hierarchy.

In order to improve the quantity of information the user obtains in each step, the initial "random" selection is substituted by a "biased random" node selection. As stated previously, the change is in the second step of the process. In the improved approach, instead of selecting one node in L_i according to a uniform, discrete probability density function a "biased-random" selection step can associate a probability value to a node in this level, on the basis of one or more of the following criteria:

1. The quantity of information associated to the subset;
2. The previous selection of the subset during the same search procedure;
3. The propagated information about the user decisions during previous searches;
4. The subset relevance to another subset that was judged previously by the user;
5. The number of subsets.

Each of the factors stated above contribute to the overall probability P_M of the subset S_M being selected.

The influence of quantity of information (factor one) is calculated as the average distance between the images in each cluster. The cluster containing more diverse content will have a bigger quantity of information, thus be more useful to the user in terms of new knowledge obtained. This way, subsets with more diverse content (i.e. with higher entropy), will be more likely selected than those with less informative content.

The factor two, involves nodes previously selected by the user and discarded due to the lack of interest. Such node can be assigned with either null or very low probability of being selected again. This factor takes two things into account:

- the possibility of an error in the subset assessment by the user, if the user is not sure that the content of the subset is relevant;
- the fact that not all images from a subset are displayed, due to their large number and limited space for their visual display.

In the similar way, factor three considers the subsets found interesting in previous user sessions. If the search cases are related the higher probability will be assigned to this subset in curent user session.

The factor four refers to the number of child nodes one node contains. The node, will have more diverse content if having more child nodes, considering the principles of hierarchical clustering.

As a further advantage, each subset can be endowed with a value of interest which can be set by the user, for example using interactive visual slider or similar solution displayed close to the subset representation. This will influence the value of this node but also the probability values of the related nodes. By modifying this value of interest, the user can modify thus update, the level of probability of a selected subset. At the same time the probabilities P of related subsets, in term of similarity, will also be promoted or degraded with respect to the current probability value of the examined node.

In the last tested version of the hierarchical browsing application, described in Section 5.4.2, only factor one influences the "biased random" jumps through the structure. The rest of the described functionalities will be implemented and tested in the future.

5.3.2 Jump-to-leaf and content access

Hierarchical visualisation solutions use single images to represent the node content and start the browsing (*e.g.,* [24]) or display the set of images which are the gateways to the related lower hierarchical levels. In the hierarchical browser, proposed within this thesis, the user can start browsing and access the content through

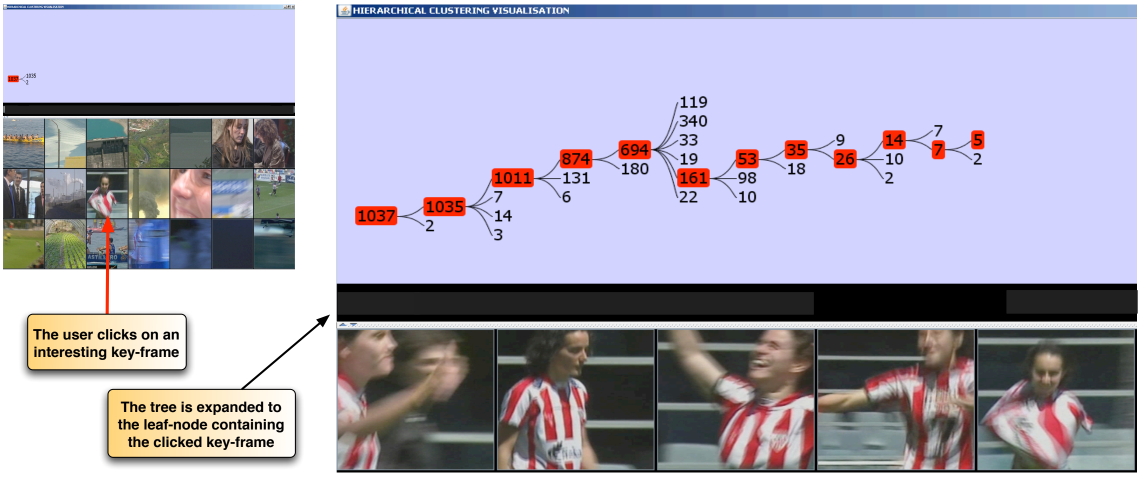

Figure 5.6: By clicking on a key-frame of interest, the tree is expanded from the root to the leaf node containing the key-frame itself and the leaf preview is shown in the lower window.

both a global and local view (upper and lower interface window) due to their interactive properties. The access can be performed by node or direct image selection.

By clicking on any node $C_{i,j}$ in the tree visualisation, the selected node becomes the focus of the tree visualisation and the preview of this node content is displayed. At that point the user can perform *sequential access* to other nodes, moving backward and/or forward through the tree to refine his/her search. The dynamic change in the tree visualisation enables the user to track his/her current position inside the database, thanks to the colour encoding of the search path. As shown in the Figure 5.3 the focus node is the last red node in the tree view and its content preview is displayed in the grid below. The user can follow the “red path” towards the root node to find more similar images or go deeper into the hierarchy.

A novel functionality, called *jump-to-leaf*, is an additional advantage which this browsing solution provides. If the user clicks on one interesting image f $(1 \leq f \leq N)$, either at browsing initialisation or at any moment of his/her exploration, the leaf-node $C_{n,k}$ containing the selected image becomes the node in focus. In this case, the database tree is expanded until the last level L_n, all nodes from $C_{1,1}$ to the leaf node $C_{n,k}$ are colour-encoded to highlight the search path, and the preview of the leaf node $C_{n,k}$ is shown on the grid, as shown in Figure 5.6. Starting from the leaf node containing f at level L_n, the user is then able to climb backward the hierarchy and sequentially access red encoded nodes at levels L_{n-1}, L_{n-2}, $\ldots$, in order to refine his/her search and obtain the largest set of similar images that interests him.

This direct access to a node at level L_n, especially if combined with the backward sequential climbing, improves the effectiveness when the user’s aim is to retrieve more similar images.

If the number of images in the node in focus is larger than number of grid cells the user can ask for a new set of images to be displayed. This is especially useful in case the user wants to perform further exploration of the same node. During the evaluation it was noted that some users preferred to stay in the root node and ask for a new set as opposed to interactive navigation through the tree. At the moment there is no memory in the process, so the application extracts a random set from the entire node without distinguishing images which had already been displayed. The "memory" of the extraction process can be easily added and it is expected to reduce time needed to find relevant image or images.

Within the proposed application the user can use any interactive method at any time. Random exploration can be used as initial strategy but also in any other retrieval stage. It can be combined with sequential access and new content set request.

5.4 Evaluation

The evaluation performed for assessing the hierarchical browser usability was conducted in two stages. Initial user tests were conducted over a defined set of retrieval tasks following the user tasks defined in [10]. In this stage all users performing the tests were under the direct supervision of the application developers. The supervised sessions ensured that users have spent a certain time using the application and have become familiar with all application tools and properties. All users performed some example tasks at the beginning of their test session in order to get their "hands on" the application. The usability of the novel random tool was conducted as part of the usability test done for the whole application. None of the users were either aware of the principles and the technical ideas behind the random exploration, nor were they directed to give special attention to the random tool. The specifications of the random interaction instrument were given together with the sequential access tool as an application feature that can be used for the database exploration.

Therefore the initial evaluation had a formative character since it was done in the intermediate stage of interface development. It aimed at assessing the potential application problems and gathers users' opinions to be used in further stages of development. Information was gathered through the capture of user logs and by questionnaires. User logs collected the time duration for each user task as well as the interaction methods used. Based on the time information a mean time per task has been calculated which can be an indicator of the retrieval efficiency. As stated in [161] the mean time-per-task measurements during formative evaluations are useful since "it might not tell you exactly what the problem is, but it can help tell you where there is a problem".

In terms of user satisfaction, the first stage of evaluation aimed at determining how well was the random method accepted (or not) by the users when performing a retrieval task over a rushes database. This goal was assessed by observing users during the user sessions.The goal was also to see how much random exploration method was utilised by the users during the same process by collecting the quantitative information through user logs. The results of the tests also aimed at indicating if random exploration was successfully used within one hierarchical browsing/retrieval application (for certain user tasks) in combination with standard sequential method for navigation through the hierarchy. This was assessed through a number of completed retrieval tasks.

The second stage of the evaluation was equivalent to the first one in terms of the evaluation methodology used. The difference is that the second evaluation was performed over a larger group of users (55 in total) which consisted of news-(professional users) and non-experts (home users). The evaluation was partially observed by the application developers and completely observed by people participating in RUSHES project (thus familiar with the application idea and project goals). Tests were partially lab-based and partially performed in real-working environments of professional broadcaster. The evaluation assessed the whole application using some retrieval user cases defined on the project level [D5]. Again, user logs of each retrieval session were stored, containing the time duration for each user session (per retrieval task), the number of times each interaction method was used, and the total number of steps used for accomplishing each retrieval task. Based on these log files the goal was to analyse the interaction methods preferred by users, as well as the time needed to accomplish each task. The observation of the favorite interaction methods relates in particular to the preference between a standard sequential access and the proposed novel random method.

Time measuring was performed to verify if the application can be used for retrieval and if by using it users would complete the tasks in a reasonable time. However, these results should not be used for quantitative evaluation since tests were partially performed remotely over the internet, thus having influential value differences. In addition to the user log analysis, a questionnaire was filled by users in order to assess visualisation aspects of the proposed application interface, and in the specific whether it is intuitive and easy to use the application interface, having in mind both visual and interactive properties.

In order to rate the usability issues not assessable by log-based analysis, partially controlled user sessions were conducted for observing the interaction with the system [7] in order to perform a qualitative analysis. This aimed at analysing, among other:

- the grade of complexity of all interaction tools - users asking for help on how to perform some actions;

- the alternative tools provided for supporting the retrieval, in order the user not to get frustrated - user not knowing what to do next in order to continue the search;

- other user feedback - statements about interesting functionalities of the interface, combination of interactions which leads to successful retrieval in shorter time.

In the end of the retrieval sessions, questionnaires were used to collect not only general usersÕ opinions, but also views on specific points, thus providing subjective data to be analysed in a quantitative way [7].

5.4.1 Initial professional evaluation

After implementing the first hierarchical browser prototype the application was tested by professional users in order to validate the idea of random access and exploration. For this reason the evaluation criteria were the satisfaction and usefulness of proposed application. Measuring the level of user satisfaction with the application, was performed among journalists within the EiTB [54] television centre, simulating as close as possible the real user scenario in users' working environment. Journalists were introduced to the application and the evaluation tasks by a short tutorial and demonstration. After a short trial period used to make the users feel comfortable with the application, the evaluation process was initiated.

A grid showing 21 key-frames has been chosen for testing as a common search scenario for all users. Journalists were given two professional use-cases derived from the ones defined within the EU funded project RUSHES [10], *i.e.*:

- **User-case 1:** Find rushes key-frames containing aeroplanes (see Figure 5.7);

- **User-case 2:** Find rushes key-frames showing a flood scene (see Figure 5.8).

After performing the specified tasks, EiTB journalists were asked to fill a questionnaire (summarised in Table 5.1) for rating their satisfaction with the most important aspects of the proposed solution. The design of the questionnaire enables the user to mark their level of agreement with the provided statements. Marks were given using the typical format of five-level Likert scale as shown in Table 6.14. Likert scale [27] is a psychometric scale commonly used in questionnaires, and widely used scale in visualisation surveys [149], [56], [79], [90].

Obtained evaluation results are given in Figure 5.9 as a diagram of the user marks per each question.

Figure 5.7: Examples of key-frames answering User-case 1.

Figure 5.8: Examples of key-frames answering User-case 2.

No.	Statement
1.	It is easy to use the interface
2.	Interaction with the interface is comfortable
3.	It is easy to understand the functionality of the interface
4.	It is easy to understand the functionality of each button
5.	The tree in the upper window is helpful for understanding the database structure
6.	The interface is well organised
7.	The organisation of the visualisation display is clear
8.	The tree in the upper window is helpful for browsing and accessing content
9.	The key-frames displayed in the bottom window provide a good overview of the node content
10.	Random exploration is helpful for browsing when I don't know where to find the desired content
11.	The interaction method is intuitive
12.	The colour of the tree nodes helps me to understand my position in the database
13.	The interface is pleasant to use

Table 5.1: Questions from user questionnaire.

After filling the questionnaire the journalists stated positive and negative aspects of the application, and gave personal comments on potential improvements. The

Mark	Answer
Strongly disagree	1
Disagree	2
Neither agree nor disagree	3
Agree	4
Strongly Agree	5

Table 5.2: Marking system in Likert scale.

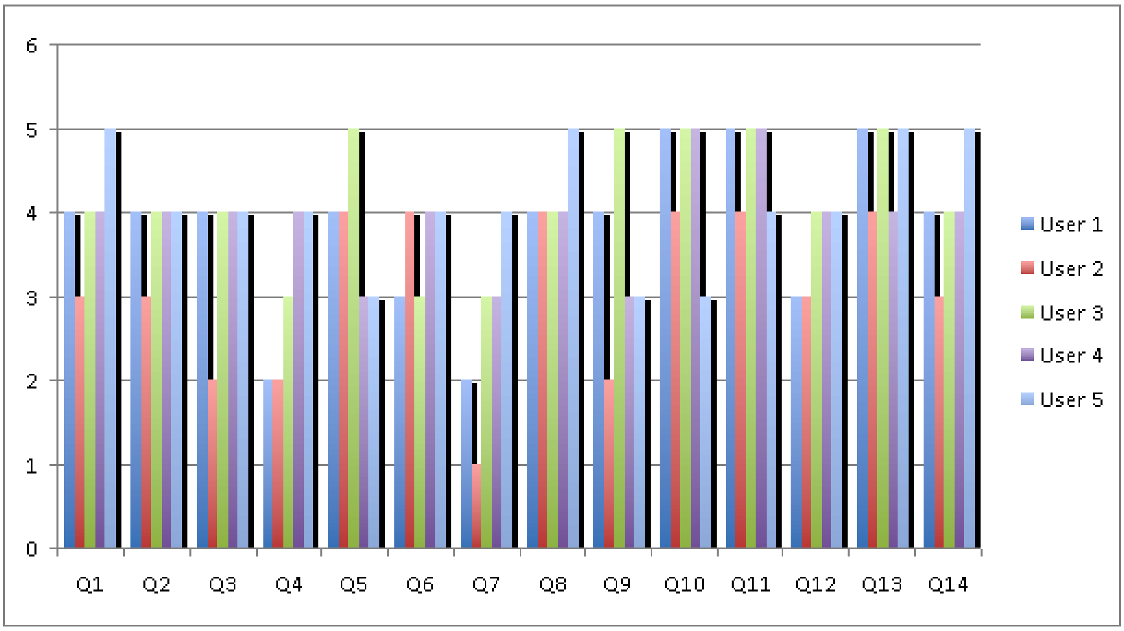

Figure 5.9: Marks stating the user satisfaction with the application.

most frequent positive and negative aspects are listed in Table 5.3 and Table 5.4, respectively.

As it is evident in Figure 5.9, the novel random access exploration has been highly appreciated by the journalists as an useful tool for browsing, especially when they did not know where to find the desired content. The same positive comment on random access was also one of the most cited among the reported beneficial aspects of the application (in Table 5.3).

Regarding the potential improvements, EiTB journalists also suggested to display the basic information about the video segment represented by the selected keyframe, such as video title, date and duration. In addition to this, they stated the importance of adding the option of going a step backward in the navigation process, in case the chosen direction of exploration is not fulfilling their expectation. Finally they suggested to integrate a video-player directly in the application interface.

No.	Positive remarks
1.	Comfortable tool for browsing clip databases
2.	Random exploration of nodes is helpful to find content without text-based search
3.	Preview key-frames are good representation of the node content

Table 5.3: Most frequently cited positive remarks.

No.	Negative aspects
1.	Add more information about the clips
2.	Buttons should have icons and tool-tips
3.	Displayed key-frames should be adaptive to node size

Table 5.4: Most frequently cited negative remarks.

Home-user evaluation of application efficiency

Initial evaluation of the system efficiency was measured in terms of the time needed for accomplishing the given tasks. System testers were five non-professional users, chosen among the members of the Queen Mary, Multimedia and Vision Research lab (MMV), working in various areas of multimedia development. They were unfamiliar with the content repository and were not part of the developing team for this application. Each user was first introduced to the system by explaining the functionality and properties of the application. After the users felt comfortable and stated having sufficient understanding of the application, three sets of tests were performed. Test users were asked to find key-frames containing a plane, scenes with a flood and a sheep from the test database. These tasks were selected as examples with different level of dificulty. For example the rushes repository there are 3.37 % of key-frames contain a plane, 1.64 % with flood and 2.12 % with a sheep. Although the main task of this experiment was measuring the time per task, user behaviour and types of interaction they performed were noted, for future application improvements.

Table 5.5 shows the average time per user needed for successfully completing the task. Time was measured per task. User behaviour and preferred interaction methods were measured by counting the number of times the user employed each tool. The relevant values were: number of random access steps, number of sequential access steps, number of requests for a new set of key-frames from the same node and number of clicks on a single key-frame in the lower application window for a *jump-to-leaf* functionality The total number of steps are computed as a sum of all previously

User no.	Average time [s]	Total steps	Random steps	Sequential steps	New set of key-frames	*Jump-to -leaf*
1	8.67	2	0.67	0	0	1.33
2	14.40	3.33	0	0	3	1
3	48.30	15.33	9	5.33	0	1
4	12.75	3.33	0	0	2.33	1
5	22.39	10.67	5.33	1.33	3.33	1.33
Avg.	**21.30**	**6.93**	**3.00**	**1.33**	**1.73**	**1.13**

Table 5.5: Efficiency test: average results per user.

stated interaction methods applied by each user. The results in Table 5.5 show that all three tasks were accomplished in a reasonably short time and successfully.

During the evaluation process we noticed different behavioural patterns with different users. This is due to the fact that the application offers three methods of interaction: random access, sequential access and *jump-to-leaf* through the hierarchy. The variations in behaviour and navigational patterns can be summarised as:

- Some users employed the random access exploration, but if the given key-frame was not found after few attempts, the sequential access was used instead;
- Several users preferred to stay in the root node and continuously ask for new sets of key-frames of the root summary (lower window). After seeing something that might be visually similar to the requested key-frame, the users clicked on the key-frame, jumped to the leaf-node and started going backward through the hierarchy.

5.4.2 Final evaluation of hierarchical browsing tool

The evaluation of the last interface prototype was performed within the RUSHES project [61]. It consisted of both professional and home-user evaluation with the goal of testing *usability* and *acceptability* of the developed prototype. The proposed hierarchical browser was tested as an integral part of the RUSHES system and details of complete procedure and results are reported in [155][1].

These evaluation sessions consisted of guided usage scenario and tool explanation performed by a project member. Since the evaluation was distributed over several premises an "Evaluation guide" was compiled within the RUSHES project and is given in [155]. The tool specification provided by the project member emphasised

[1] The results were gathered and processed by Pedro Concejero from Telefonica Investigacion y Desarrollo (TID), Spain as part of the RUSHES evaluation

innovative aspects of the proposed solution. The questionnaire was defined so that these novel aspects are investigated in terms of usefulness and acceptability.

Professional evaluation

The professional user evaluation was conducted in ETB premises where the final implementation of the proposed hierarchical browser was tested as part of the RUSHES system. A total of 15 professionals participated as users in this evaluation stage. The gendre distribution within the professional users is given in the Table 5.6 and displayed in the Figure 5.10.

Partner	Female	Male	Total
Journalists	3	2	5
Archivists	4	2	6
Script content	1	3	4
Total	**8**	**7**	**15**

Table 5.6: Genre distribution of professional users.

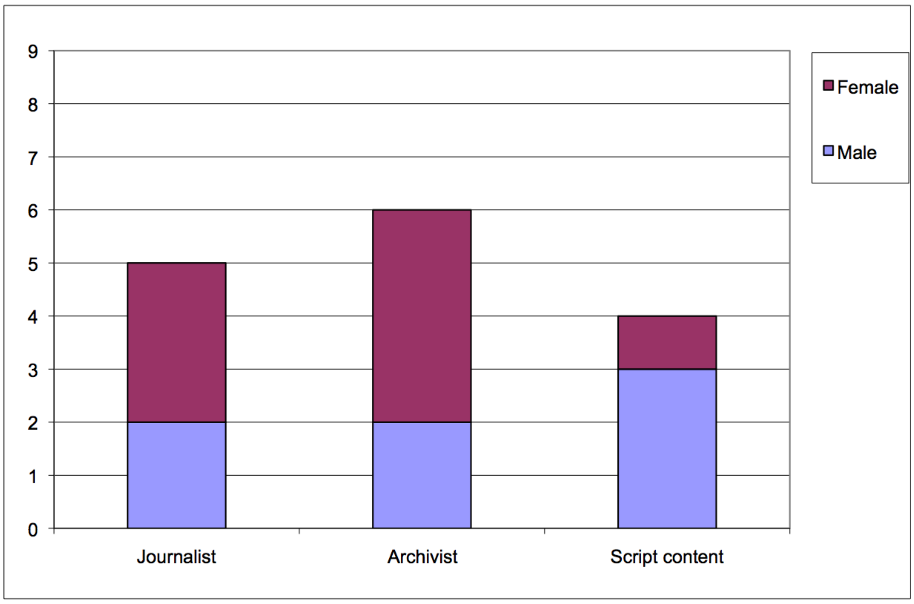

Figure 5.10: Genre distribution of professional users.

Among these users there were three different occupations: journalist, archivist and script content editors. As reported in [155] within this evaluation stage three testing groups have been organised to test and evaluate the whole RUSHES system (including hierarchical visual browser). Professional groups used were:

- Test group 1 - journalist testing group, which consisted of three young journalists and other two expert members working mainly in the news area;

- Test group 2 - archivist, formed by the archivist chief and five archivist, all with an extensive expertise in the cataloguing tasks;
- Test group 3 - script content editors, group of mainly young people whose occupation consists of editing and producing script content programs.

Figure 5.11 and Figure 5.12 show the differences in rating of the visual browsing tool depending on the user group. It can be concluded, based on the Figure 5.11, that archivist rated this tool lower than script editors. One of the possible deductions might be that there is a strong influence of age in terms of excepting novel methods for interactive exploration.

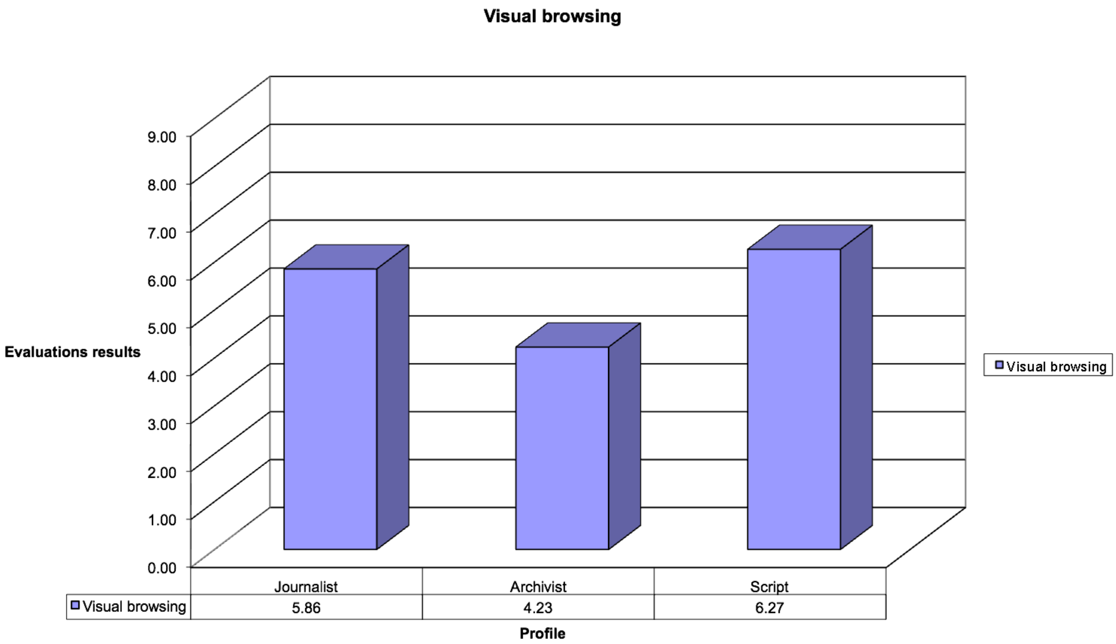

Figure 5.11: Visual browsing evaluation results.

Figure 5.13 shows the cumulative results of the questions answered by professional users (VB refers to the questionnaire questions related to hierarchical visual browser). The complete RUSHES questionnaire is given in [155].

Home-user evaluation

Home-user evaluation was performed by each partner in RUSHES within the evaluation period of the project. The proposed system was tested at premises of project partners by people who were not involved in the research or development RUSHES project in any way.

A total of 49 people participated in system testing in this evaluation part, among which 70 % of the group were below 29 years old. As given in [155] within the

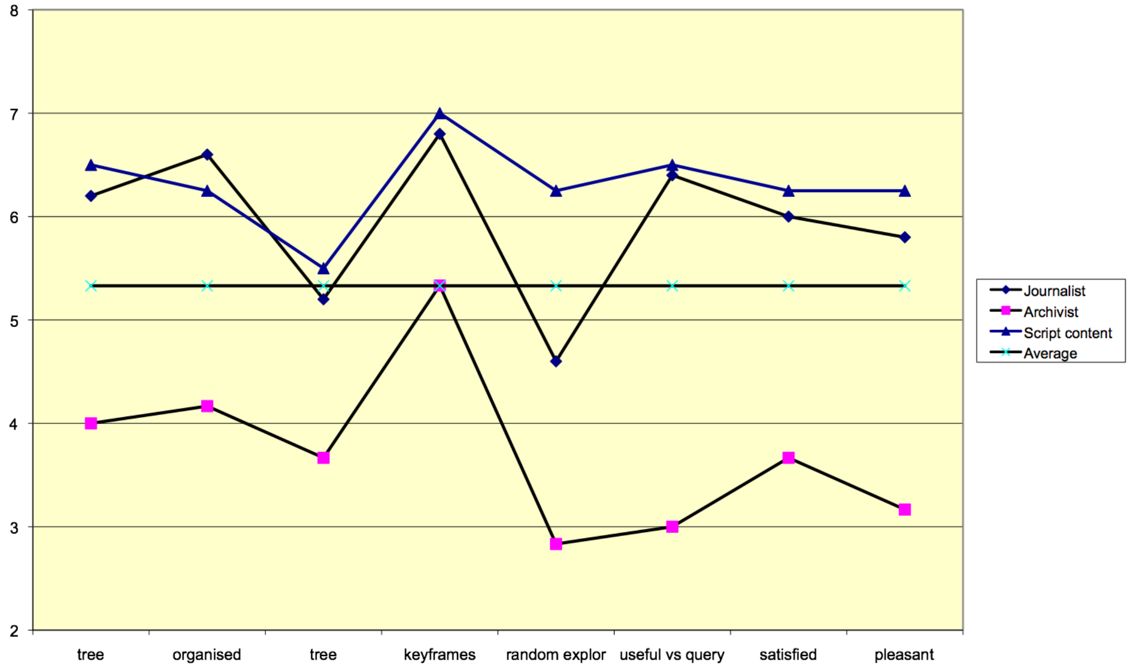

Figure 5.12: Statistical analysis of visual browser results.

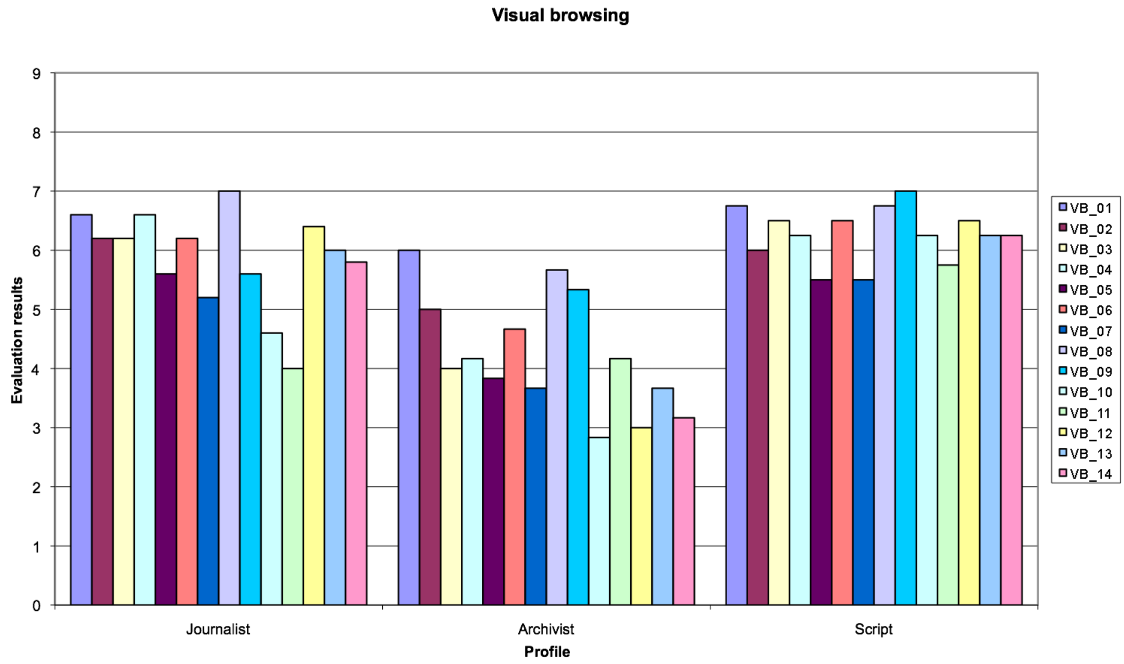

Figure 5.13: The plot of results based on users' answers to the user questionaire.

home users who performed the evaluation, a distinction was made between more experienced and less experienced users with multimedia databases and indexing systems.

Partner	Female	Male	Total
EiTB	5	4	9
TID	3	3	6
BRUNEL	2	4	6
QMUL	1	4	5
HHI	0	9	9
ATC	1	5	6
VICOMTECH	1	7	8
Total	**13**	**36**	**49**

Table 5.7: Genre distribution of home-users per RUSHES partner.

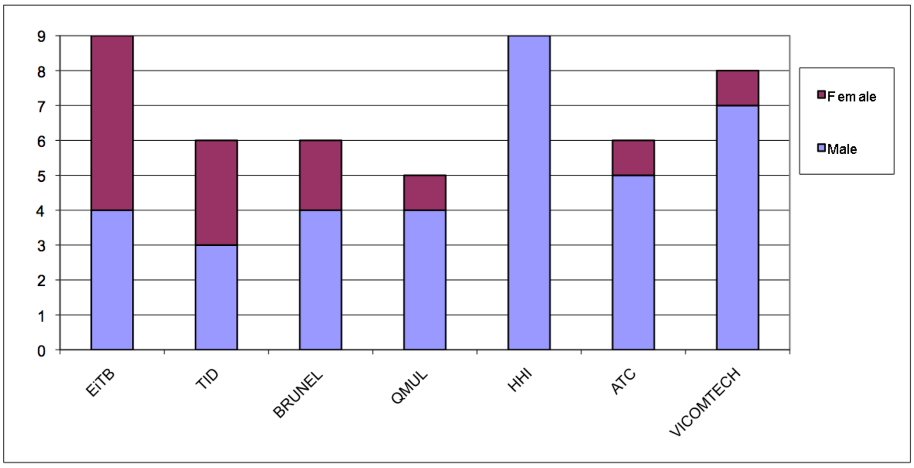

Figure 5.14: Genre distribution of home-users per RUSHES partner.

Preferred methods of interaction

During the user evaluation stage of the RUSHES project, user logs of visual hierarchical browser utilisation were stored for later analysis. These results are used to observe the behaviours of the users while using the application, in particular in terms of preferred interaction methods and utilisation of random exploration and access tool. The total number of processed user logs were 55 which consisted of both professional and home user results.

Figure 5.15 shows distribution of interaction methods used within 55 users. It can be seen that the random access approach was utilised in total more than any other interaction method. The sequential access folows as second most used method. The second figure shows distribution of sequential and random access over different users.

It can be concluded that users used sequential and random access more or less equally. In addition to this, general conclusion made based on observations of users' behaviour, is that the diverse methods of interaction were highly appreciated. In fact, not once the user was stuck at the point where he/she did not have an idea

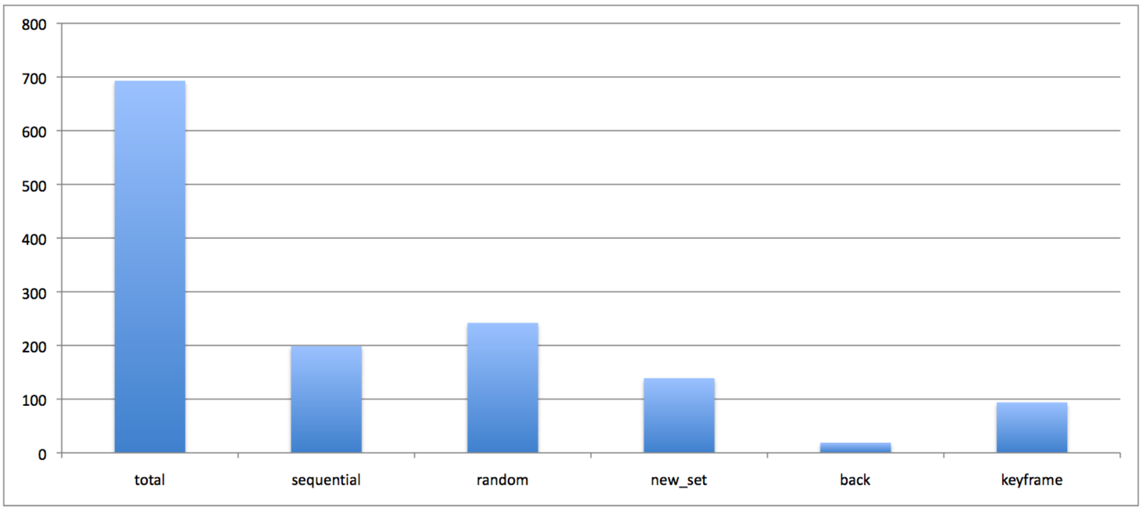

Figure 5.15: The number of different interaction methods used: total, sequential, random, requesting new set of preview images, going step backwards and selecting one image/keyframe from the grid.

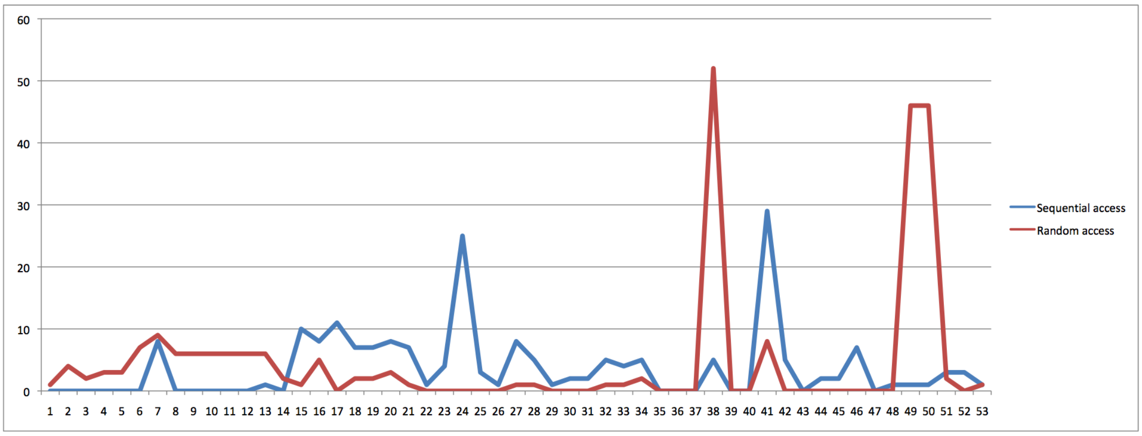

Figure 5.16: Distribution of number of sequential and random access methods over different users.

how to proceed while trying to accomplish the task of retrieval. If compared to such situations that happen with standard retrieval systems, this conclusion is seen as a promising sign in terms of effectiveness.

In this chapter a description of a visualisation application for browsing and searching through a *rushes* repository has been given. This approach can be easily expanded to any other data-set. As explained, the proposed solution combines information visualisation techniques for structured navigation of the database with hierarchical summarisation methods for fast content grasping. The novelty of the presented approach lies in modelling the user's navigational behaviour by a "biased random" exploration functionality which is useful for identifying new search directions in the case of unfamiliar databases. Interactive navigation through the content is supported by browsing or direct access, either by the image/key-frame selection or by sequential access to node previews.

In order to perform the evaluation of the hierarchical browser, a log-based study [4.6] of user actions during the retrieval was used. The goal was to see the grade of appreciation of the users regarding the random interactive tool for performing the exploratory search, and to analyse the user preferences in terms of exploration access methods. Time stamped logs recorded the number of times each interaction method was used as well as the duration of each retrieval session. We interpreted a click on a image as an objective indicator of relevance to the users' information goal, thus assessing if and after how long users achieved their goals. In the end of the retrieval sessions, questionnaires were used to collect not only general users' opinions, but also views on specific points, thus providing subjective data to be analysed in a quantitative way [7]. Based on the gathered user logs it can be seen that two supported navigational techniques, i.e., sequential and random, were used equally in average. This is considered as a positive result since, as a novel application, the random exploration did not induce negative retrieval effects, in terms of results or user comfort. We consider that in case the user did not feel comfortable with the random exploration, they would be able to rely only on the sequential exploration. Based on the user logs, we recorded that this happened with only a few of them. Because of the application testing was performed over the internet, the gathered time information can provide only indicative information for a quantitative analysis. Based on the performed evaluation, results confirmed that the proposed application successfully assists professional and home users in finding their way through the initially unknown content repository, where random exploration was considered helpful for browsing in cases when user does not know where to find the desired content. As an inferred conclusion, we believe that new methods based on explorative search and supported by statistical models, will be more and more considered and studied for application in real retrieval systems.

Chapter 6

Multi-concept browsing and retrieval tool

In this chapter a developed Multi-Concept retrieval application is described. The idea of the Multi-Concept browser is to combine directed and exploratory search based on a set of semantic concepts extracted from the image collection. The goal of a concept-based image retrieval system is to “discover images pertaining to a given concept in the absence of reliable metadata” [48]. For retrieval purposes, the content of an image collection is usually previously analysed in order to determine a set of relevant semantic terms which can represent content and be used for query. Once the relevant concepts are identified, semi-automatic or automatic algorithms are used for assigning labels to images, thus indexing the image collection. This provides a retrieval system based on a semantically indexed image collection. On the basis of such indexing, an interface application is proposed here which uses visual and interactive tools to enable a non-expert user to retrieve image(s) from an unknown image collection.

The proposed application supports directed and explorative search in different stages of retrieval. Directed search is supported for query specification and is enabled by possibility of interactive combining of available concepts. The idea behind concept combination was to assist the user to define the context associated to his/her information needs.

For example, in standard retrieval system in order to find image which semantic content is related to concept ”grass” besides using only a concept “grass” to examine all related images, this application provides the option of searching for more specific topic based on the same set of semantic terms. This means that user can search for green areas in urban environment combining ”grass” and ”building” without examining separately all images of ”grass” and later all images with ”building”. In addition to this the proposed application avoids development of a new concept detector to cover query ”green areas in urban environment”.

Query specification using concepts can be used for directed search giving the user a way to initialise the query. Once the system returns a set of images corresponding to the user query, the user can perform the explorative search in order to find image of interest or gather some information on how to reformulate the query. This set of images are displayed in a meaningful way, for example based on the concept distribution, so that the user can easily evaluate the results and continue or change the query. With this aim the visualisation solution displays images grouped according to concepts detected in each of them.

As an example of user task for which this application can be useful is: ”User needs to find image of a park fountain in the modern city”. This is the case of directed search, since the user has a very specific information need but lacks the knowledge of the content. Having available concepts such as ”water”,”building”, ”grass” and ”rock” the user can directly query the system, which will retrieve images that have these concepts in common. And using MC browser user does not have to examine each of them separately. In addition to this, system will show also additional images which does not contain all of the selected concepts broadening the retrieved set due to the standardly present classification error. In other words, in MC browser application user uses directed search to start the retrieval process combining concepts using human logic and later performes exploratory search using the retrieved image set.

6.1 The motivation

Apart from generating indices based on visual information extracted from images (such as colours, textures, shapes, etc.), indexing can be also based on higher level semantics. This means that images can be described by labels such as *grass*, *sky* and *building*. Such semantic information about the image content is more useful for browsing, exploration and/or retrieval than low level visual features of images. However, this type of index is generated either by running automatic (or semi-automatic) algorithms or performing manual annotation so that in cases of large, or even medium data-sets, manual semantic annotation is only partial or non-existing. This is due to the fact that manual annotation, although providing semantic descriptions on the high level, is also time consuming and not a straight-forward process (annotation is biased and depends on the user's knowledge of the domain). It is also true that there are “hundreds of objects in most of the images that could be referenced, and each image has a long list of attributes” as stated in [142]. The same authors state that manual annotation is usually limited since users perform minimal annotation relevant to their current task. Based on these considerations it can be concluded that manual labelling in most cases does not cover a wide set of queries and that images have to be re-annotated for expanding the ones supported. Another option is to employ the machine-based algorithms (semantic concept detectors) to "transform" the low-level, visual information into more advanced form in terms of semantics. They employ different techniques, such as pattern recognition approaches, to classify images into one of the predefined categories (or classes). Since such obtained information is still in a form far from the human level of interpretation, there is usually a need for "something-in-between".

Retrieval based on pre-defined concepts is supported in most systems by a *query-by-concept* approach. In standard semantic retrieval scenarios concept detectors are developed in order to cover the content and queries related to the data-set. In case the retrieval system relies only upon the machine detectors, the retrieval performance will be strongly influenced by the accuracy of the applied automatic annotation. This is especially true if the user has no "influence" on the system (e.g. provide feedback for results improvement, engage with the system to understand how the results are generated, etc.). On the other hand, the environment where query formulation, exploration and retrieval of images are supported by visualisation and interaction tools, can utilise the results of automatic image indexing and improve its overall performance by increasing the level of user involvement. Utilising human "higher cognition" [68] provides a means for the optimal use of the available semantic information and to extend the level of human-computer integration. In other words, instead of adapting a classifier for each new query requested by the user, existing classification results can be re-used and combined by humans.

In addition to this, the queries are not always defined in the same way by different users. For example, some users consider *open landscape* as an image without wild or any other animals, some include both; *modern city* can be a query for city panorama or more focused street view full of cars. For this reason having one set of concepts and enabling the user to combine them aims at creating flexible image management and retrieval environment.

Here, two types of concepts are distinguished: mid-level, such as *grass*, *lion*, *building* and high-level, such as *wild life*, *modern city view* and *boat race*. Following this definition and considering the issues stated above, this part of the thesis proposes a Multi Concept browser tool which provides visual and interactive support for image exploration and retrieval based on the results of mid-level semantic detectors.

In particular, the proposed Multi Concept (MC) browser includes:

- support for visual formulation of broad range of queries;
- content filtering;

- multiple information visualisation components;
- combined interaction methods for user support.

The developed Multi Concept browser, consists of two interconnected interface windows shown in Figures 6.1 and 6.2. Figure 6.1 is the screen-shot of the interface window for the query specification. The functionalities and elements of this window are described in Section 6.2. Figure 6.2 shows the screen shot of the latest version of the visualisation solution for content exploration and retrieval. Since the development of this part was performed in two stages, the initial solution is reported in Section 6.3, whereas the latest implementation is described in Section 6.4. The evaluation set-up, metrics and results obtained are stated in Section 6.5.

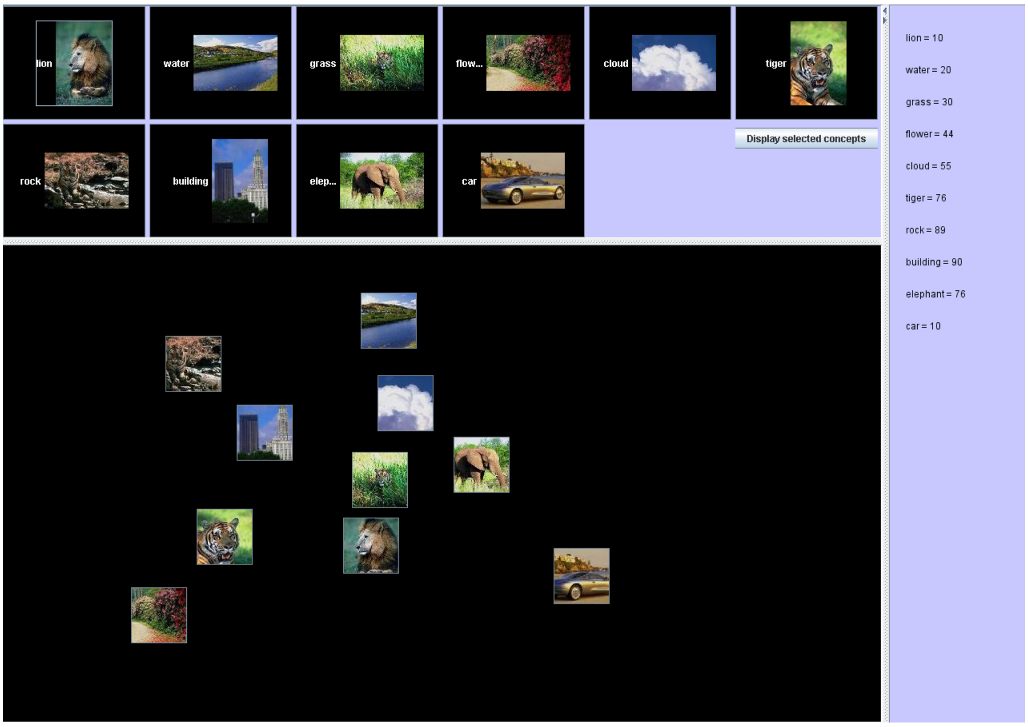

Figure 6.1: First MC browser application interface window for interactive visual query specification.

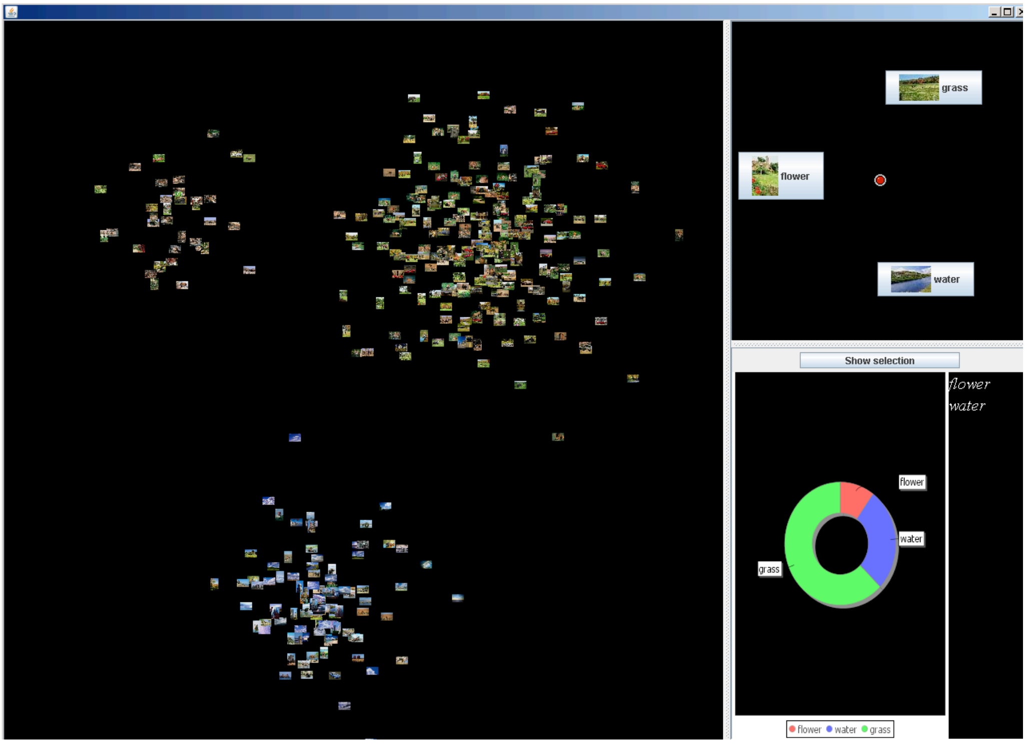

Figure 6.2: Second MC browser application interface window for displaying images filtered by the user query.

6.2 Visual query specification

The information obtained as a result of several concept detectors on the same data-set can be seen as multivariate content information. Defined semantic categories are mutually overlapping since more concepts can be detected in the same image or in other words, one image can belong to several concepts.

This integrated, multivariate information can be presented in a simple tabular form as shown in Table 6.1 where the images from Corel collection are labelled (using MOO approach stated in Appendix C) according to the concepts they belong to (Flower, Grass, Rock, Water, Building, Car, Cloud, Elephant, Lion and Tiger, respectively).

F	G	R	W	B	C	C	E	L	T	Image
img1	1	0	0	1	0	1	1	1	1	1
img2	0	1	1	1	1	1	1	1	1	1
img3	0	1	0	0	0	0	0	0	0	0
img4	1	0	0	1	0	1	0	1	0	1
img5	0	0	0	0	1	1	1	0	1	1

Table 6.1: Tabular form listing images from the data-set and the detected semantic concepts in each of them.

Presenting results to the user in a tabular form is not an optimal presentation method in case of an image collection, for several reasons stated here:

1. A table does not show the image itself and it is know that "A picture is worth a thousand words";
2. A tabular representation does not provide useful information which can help the user to initialise the query;
3. Browsing and/or retrieval capabilities based on tables are limited and slow.

A potential solution is to develop an intuitive visual-based system which would use these results in more efficient way.

6.2.1 Query specification using mid-level concepts

The mid-level concepts detected in the database, such as *grass*, *water* or *building*, can help the user to find images of vegetation, water and unspecified populated area by directly selecting the appropriate concept. However, due to the gap between the abilities of the automatic (or semi-automatic) detectors and realistic users' requirements, an advanced method for "higher level" query specification is needed. For example, for finding images of a *rural garden* there might not be available an ad-hoc concept detector which can label images as a rural garden. Even if there was an efficient detector for *rural garden*, not every user would define the same concept equally. Further on, there could be issues with detector performance in terms of, e.g. accuracy.

For addressing these issues, the proposed framework aims at providing a high level of flexibility in the query specification as well as increased user involvement during the retrieval process by allowing the user to visually combine mid-level concepts. Extracted mid-level concepts which can be combined for query specification are called *query classes*. In case of employing M concept detectors for image indexing, the proposed framework will contain a set of m query classes:

$$Q = \{q_1, q_2, \ldots, q_m\} \tag{6.1}$$

and the images containing the terms corresponding to a specific combination of query classes are found relevant and displayed by the system.

This query approach can be compared to textual search where the user specifies requested semantic content by means of words. However, there are a few properties of the proposed visual-based query approach, which are not available in such systems. These properties are a specific contribution of information visualisation and are the following:

- The user can instantly realise which query key-words are supported and understood by the system;
- Each key-word term is "explained" by an image;
- Information about the concept relationships is displayed;
- Content exploration support provides fast understanding of the accuracy and appropriateness of the query specification.

The developed MC browser supports both simple and high level queries. Simple queries are the ones issued by a single concept selection. For example, if the user needs to retrieve images with *grass*, he/she would select only the concept *grass*. Apart from direct use of one of the available mid-level concepts, the user can select any combination of query classes for constructing a high level query. If the user is looking for *panoramic city view* or *park in the city* the selection of concepts *building* and *cloud* or in second case *grass*, *flower* and *building* can result in meaningful retrieval. This way the user is able to include his/her cognition into the query specification process based on the visualised content information.

6.2.2 Visual table of contents

The list of supported query classes is displayed as a set of image icons providing a "visual table-of-contents" for the database and an interactive query entry point for the user. As previously stated, the user can select one query class or combine several available concepts for building more complex queries.

A simple table of contents, like a textual list of existing concepts in a data-set, does not necessarily provide insight into the content of the database. Such insight can be very significant to the user especially when dealing with one data-set for the first time with no knowledge of the content. Therefore information such as the number of images of one query class and the relationships between different query classes can provide a "hint" about the type of images contained in the data-set.

For the stated reason three interface components are used here to visualise different information about the content:

- A *Concept grid*: it contains names and example images of the supported query classes;
- A *Concept relationship display*: it visualises similarity relationships between concepts in the data-set;
- A *Concept list*: it displays the number of images for each concept.

The initial visualisation view with the three distinctive interface elements is shown in Figure 6.1. The task of each component is given in detail in the text below.

Concept grid

Names and example image icons of the detected concepts are displayed in a interactive grid as shown in the Figure 6.3. The grid is chosen due to its simplicity and its adequacy for presenting the list of available concepts without overlap and confusion. Each query class is represented with a manually selected image icon and the corresponding textual label. Both image and text are used for two reasons:

1. The textual label shows how concepts are named by the system;

2. The image shows an example of which concept is labelled as such.

Figure 6.3: Interactive concept grid for query specification displaying list of query classes.

The user can click and select classes according to his/her query, thus invoking content filtering and display. At this point the user can select query classes he/she considers relevant for a query which cannot be represented by a single available semantic concept. Additional filtering support can be provided by enabling the "AND NOT" type of query, where a user could specify which concepts he/she does not want to see in images. The results expected are twofold: in case of a "good" mid-level concept detector, filtering should provide good results in terms of *precision*. However, bearing in mind the actual performance of the automatic annotation algorithms, excessive filtering can "hide" some data/image items which are actually good results, or in other words fail in terms of *recall*.

Concept relationship display

Although the user is allowed to interact with the system in more iterations, the goal is always to accomplish the task in a short period, thus achieving high task efficiency. For this reason it is important to present as much meaningful information to the user as possible, in particularly when the user is dealing with a data-set for the first time and has no previous knowledge of the content.

The number of images in which one semantic concept is detected defines the size of the query class. The class size provides basic but important information about the content of the specific data-set. The user can see that there is, for example, a large number of images where the concept *building* is detected, without exploring the content.

Additional useful information about the content can be inferred by examining the relationships between concepts. Concept relationships can provide a useful content summary by showing for example, which semantic categories have a huge number of images in common. The relationship between concepts *building* and *cloud* can "hint" on type as the images where a building is detected. If these concepts are strongly related, meaning that they have large number of images in common, the user can conclude that most of images with buildings contain panoramic views with sky and clouds. On the other hand, if *building* shares a lot of images with *grass*, the user can assume that there are images of rural gardens or city parks in this image collection.

In this framework the inter-concept relationship information is displayed as a spatial concept distribution, as shown in Figure 6.4. Concepts are placed according to their mutual similarities, where the similarity metric (i.e. the spatial distance) is inversely proportional to the number of images that the two concepts have in common, as given in:

$$d_{C_i,C_j} = \frac{k}{N_{C_i,C_j}} \tag{6.2}$$

where d_{C_i,C_j} is the "distance" between concepts C_i and C_j; k is a constant and N_{C_i,C_j} is the number of images where both concepts C_i and C_j are detected. This means that the image sets that share more elements will be placed closer together in the display. This concept mapping is achieved by using a non-metric MDS algorithm for embedding distance information.

The distance between the selected concepts infer the relative number of images that the system will filter and present to the user as a concept intersection. If the user selects two concepts which are positioned close to each other the filtering will give a larger image group. By selecting two distant concepts, the images labelled with both semantic concepts will be few, or none in case concepts are very different.

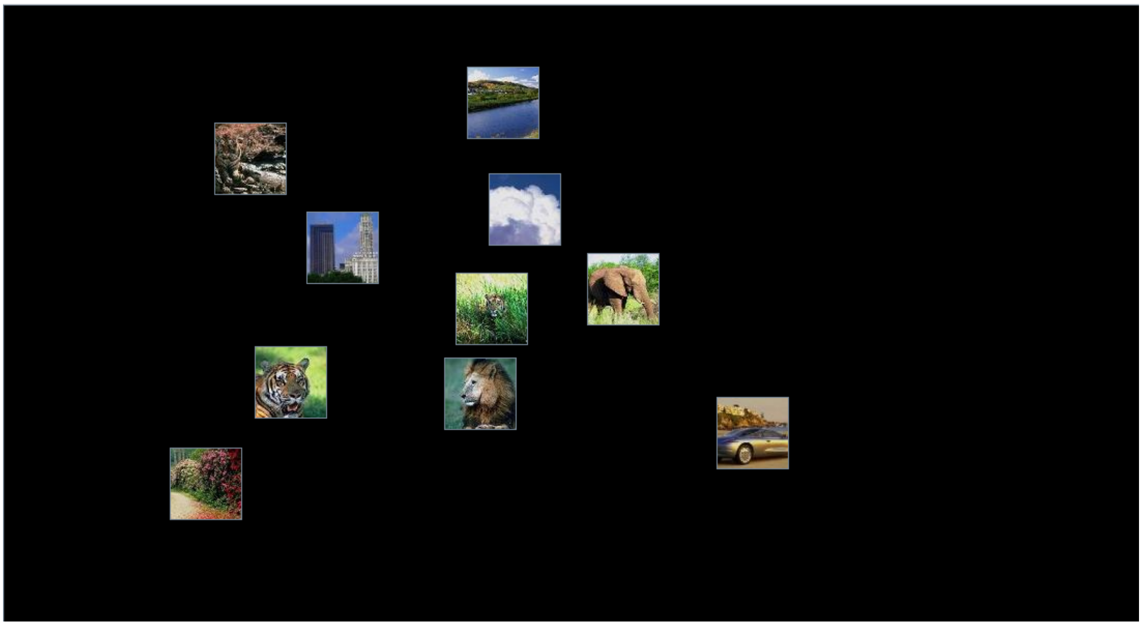

Figure 6.4: Relationship display showing inter-concept similarities in the data-set.

Concept list

The index of available query classes is also displayed in the form of an interactive text list as shown in Figure 6.5. The list is an alternative index of semantic classes which provides additional information regarding the number of images in which each concept was detected. The size of the image icon within the *concept relationship display* can also visually convey size information about the concepts. But it would not be more intuitive than the explicit number connected with the concept name such as presented in the list. The list interactivity is realised using interactive distortion of the list elements. This functionality provides the scaleability feature for displaying a large number of items in the list (if the number of query classes is large).

Using all three interface components and the information they represents the query specification is intuitive, quick and user friendly. In addition to this, the user does not have to browse through a large amount of images or go through series of iterations with the interface just to initialise the query process. After a desirable combinations is selected, the system performs content filtering and displays the results in the so called *query space*.

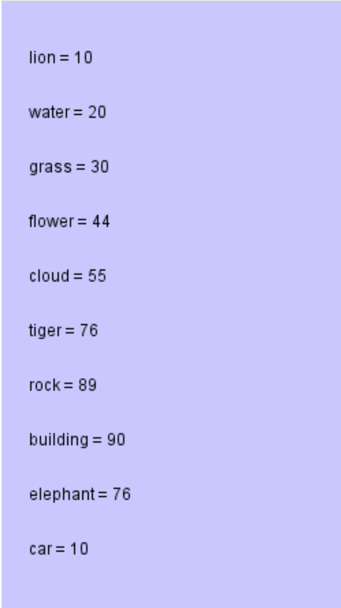

Figure 6.5: Interactive list showing available query classes and the number of images per each class.

6.3 Initial visualisation approach for the image data-set exploration

The initial implementation of a solution for image exploration aimed at displaying the "query space" by visualising all possible combination of selected query classes. For example, a selection of three semantic concepts would create seven distinctive groups as shown in the Figure 6.6, which would be visualised as seven *visual clusters* of objects in the screen.

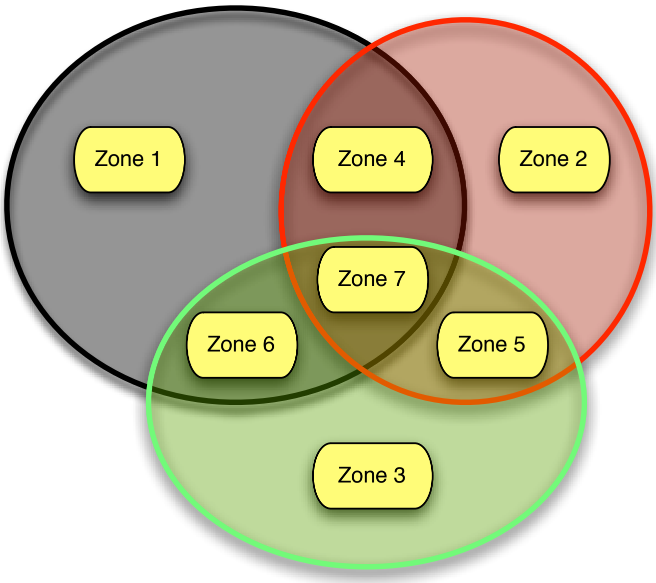

Figure 6.6: Diagram of the zones in the query space obtained using three concepts.

Zones 1, 2 and 3 would correspond to images in which only one concept is detected; Zones 4, 5 and 6 would contain images with two concept labels; while the central Zone 6 would contain images labelled with all query concepts.

As explained previously, showing different zones as distinctive "visual clusters" the goal was to allow the user to identify the results of automatic labelling and explore different sets separately thus avoiding information overflow. While browsing through sets of images with different index combinations the user could "pick up" organisational patterns which can assist in finding the relevant content faster.

Visualising all possible combinations of query indices as separate *visual clusters*, could also help the user in case of under-performance of the underlying automatic indexing algorithm. In fact, if there is an image of *grass* and *flower* but only *grass* is detected by a machine algorithm, this image will not be in the intersection of grass and flower as expected, but in the zone corresponding to a single concept images of *grass*. In such cases, to avoid confusion which a large image set can create when presented wrongly, separating them visually would ideally prevent the clutter of data objects on the screen.

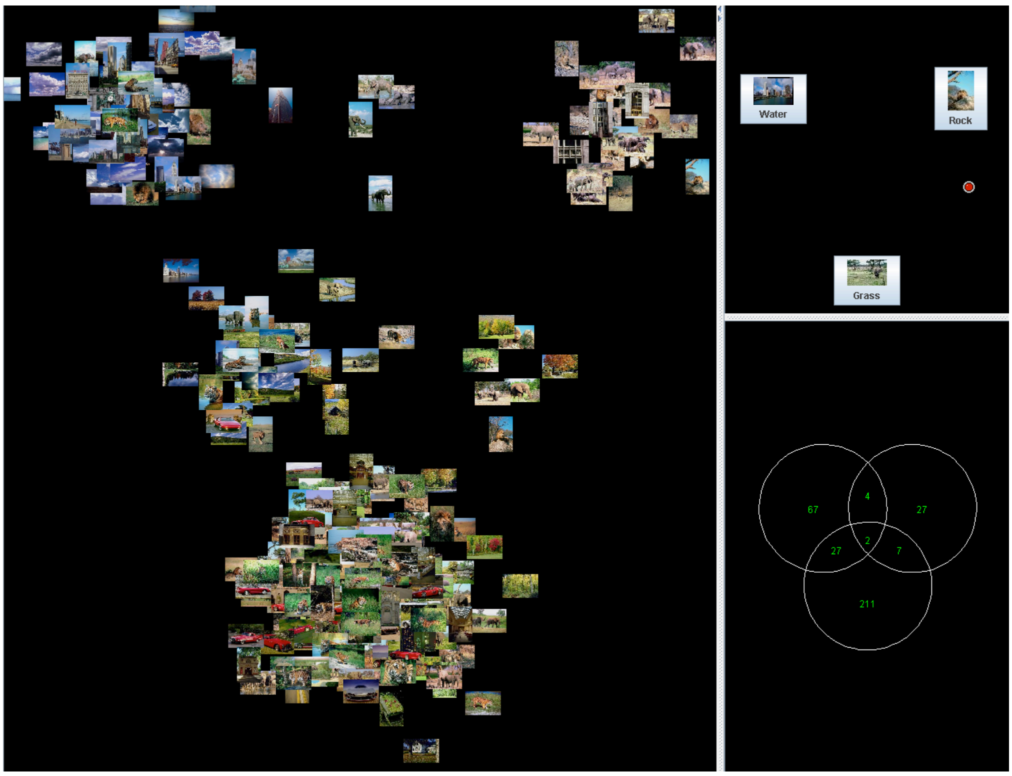

Figure 6.7: The initial visualisation solution for displaying the query space of three concepts. The left interface component is referred to as *content space* since it is used for displaying images. The*concept map* is the upper left element displaying selected concepts. The bottom left part of the display contains the *concept diagram*, showing intersections of query classes.

As visible in the Figure 6.7 the first implementation of the interface consisted of three components. The interface component occupying left part of the display is the *content space*, which is explained in the Section 6.3.1, where the content of the selected query classes is presented by filtering a subset of images from the database.

The upper right part is a display of a *concept map*, a component which serves as query space overview and provides content browsing support. This interface component is adopted also within the second prototype, thus explained in detail in Section 6.4.2. Finally, the bottom right part is the *concept diagram* which displays the concepts intersection by using Venn diagram. The benefits and drawbacks of this element are discussed in the Section 6.3.2.

6.3.1 Content space

As stated previously, the visualisation solution described in this section proposes to display the image set filtered by the user query using *visual clusters*. Each visual cluster corresponds to one zone in Figure 6.6. Considering this, the content filtering of the query defined selecting, e.g. query classes q_i, q_j and q_k, is performed in the following manner: *visual cluster* of the Zone 1 will contain images indexed as $q_i = 1$, $q_j = 0$ and $q_k = 0$, Zone 2 as $q_i = 0$, $q_j = 1$ and $q_k = 0$ and Zone 3 $q_i = 0$, $q_j = 0$ and $q_k = 1$. Intersection zones have a combination of indexes e.g. $q_i = 1$, $q_j = 1$ and $q_k = 0$. The images in the central *visual cluster* representing Zone 7 are indexed as $q_i = 1$, $q_j = 1$ and $q_k = 1$.

As a result of the content filtering multiple distinctive image groups are identified and should be presented to the user. Since the user should understand that there are distinctive image sub-sets, the position of each *visual clusters* depends on the image set it represents (in terms of combination of indices). For example, *visual cluster* corresponding to zone 7 in the Figure 6.6, will have the central position, whereas the zones 4, 5, and 6 will depend on the positions of zones 1, 2 and 3, as explained below.

Two types of *visual clusters* are distinguished here:

1. *Visual clusters* where all images are labelled by a single concept;
2. *Visual clusters* where images are labelled with k concepts, $(k = 2, \ldots, n)$.

The layout of the images is achieved in several consecutive steps. First, an area A dedicated to *visual cluster* i is determined using the formula:

$$A_i = A_{tot} \cdot \frac{N_i}{N_{tot}} \tag{6.3}$$

where A_{tot} is the total available visualisation area (depending on the display device and part allocated for *content space*), N_i is the number of data items belonging to the *visual cluster* i and N_{tot} is total number of data items in the collection. This allocated area, A_i, is represented as a circle with the diameter R_i.

In the second step, *visual cluster* c containing images with all selected concepts will be placed in the centre of the display area occupying the area A_c with the radius R_c.

The next step is dedicating an area for single-label *visual clusters* by specifying the display positions of their centres. In order to provide a flexible environment, in which the *query space* can contain an arbitrary number of query classes, positioning of the centres had to be a scalable method. For this reason, centres were placed according to a circular layout (as shown in Figure 6.8) which provides a high level of scalability. In particular, the scalability is achieved by using a set of concentric circles, which enables the selection of a large number of available positions along circular lines, as well as a flexible distance from the common centre.

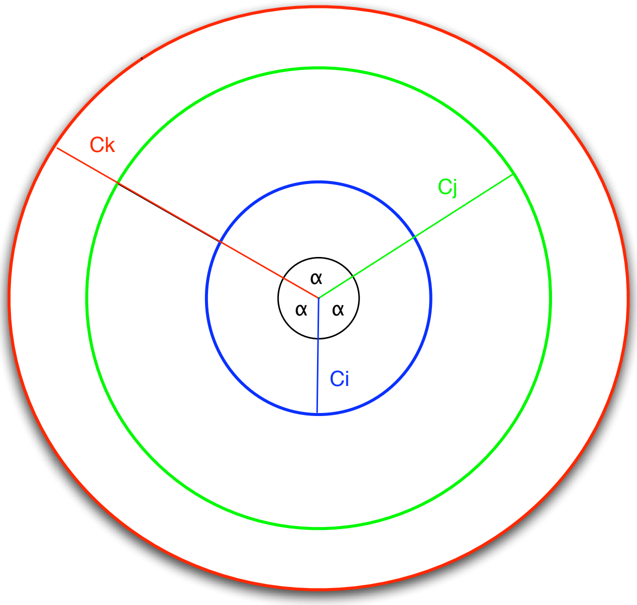

Figure 6.8: Circular layout method for placing the *visual clusters* according to their number and size.

The number of pure concepts n influences the angle α of their positions on the circular layout according to:

$$\alpha = \frac{2\pi}{n} \tag{6.4}$$

The example shown in the Figure 6.8 has 3 clusters on the circle border so the angle α is equal to $\frac{2\pi}{3}$. The radius of the circle on which the *visual cluster* will be placed, depends on the size of this cluster and the size of the central cluster. The sizes of the clusters are expresed by their radiuses, so that the *visual cluster* i will be placed on the circle with radius C_i given by:

$$C_i = R_c + R_i \tag{6.5}$$

where R_c is the radius of the central cluster and R_i radius of the cluster i.

In the example given by the Figure 6.8 the clusters i, j and k will be positioned at circles with radiuses C_i, C_j and C_k. It is evident that the sizes relationships of respective clusters are $N_i < N_j < N_k$.

After placing the centres of the *visual clusters* with single-label images, the centres of multi-label *visual clusters* are calculated as “centres of mass” between the single-label concept centres according to:

$$\vec{P} = \frac{\sum m_i \vec{p}_i}{\sum m_i} \tag{6.6}$$

where $\vec{p}_i$ is the position of single labelled cluster i and m_i is the mass of the cluster i proportional to the cluster size N_i. Within each *visual cluster* i images are placed randomly according to a normal distribution where

$$\sigma = \frac{R_i}{3} \tag{6.7}$$

so that the 99.7% of images will be located within 3σ distance from the centre. This provides visual distinction of different image groups by localising the images in one area.

Although for the optimal use of a display space, radius R_i is calculated proportionally to the allocated area A_i as:

$$R_i = \sqrt{\frac{A_i}{\pi}} \tag{6.8}$$

it can later be adjusted depending on the quality of generated layout. For example, the radius can be reduced if it would provide a better distinction between different *visual clusters*. Depending on the value of the radius for each concept area, the distinction between the image groups will be increased or decreased. Smaller radii will invoke more overlap between the displayed items, but the existence of different image groups will be more clear. A larger value of the radius will produce a sparse layout and a better overview of the images.

This initial filtering and layout approach produced meaningful layouts in the case of a small sized image collection (less than 300 images) and a small number of concepts selected by the user. However, scaling the image set up to 700 images elicited new challenges for displaying the content within the visible display area. An increased number of images caused merging of the visual clusters to the level iat which they could not be recognised as different. Another consideration was

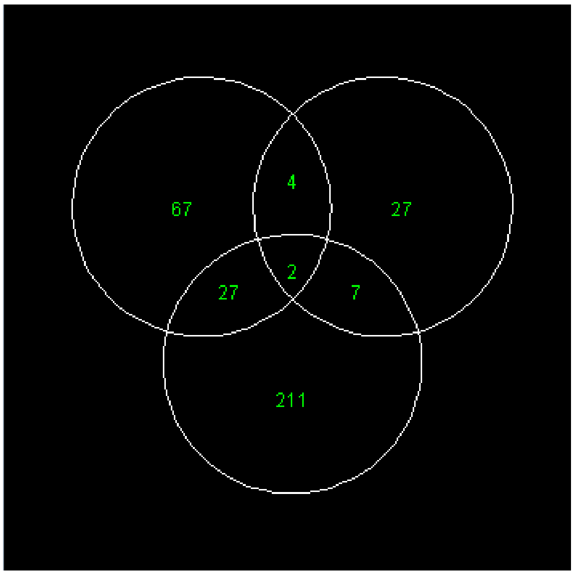

Figure 6.9: Venn diagram displaying distribution of content for three query classes.

the increased number of concepts combined for query specification. For this reason further improvements focused on lowering the quantity of displayed content and employing an interactive technique for extensive exploration, as explained later in the Section 6.4.1.

6.3.2 Concept diagram

For visualising relationships and overlapping between the filtered query classes, a Venn diagram representation was adopted as the most intuitive one where each region created by the diagram contours corresponds to a specific set of data items.

A Venn diagram is a graphical representation of n sets obtained using n closed curves, where all intersection areas are displayed (even the empty ones). Usually, when Venn diagrams are used, if any of the area is an empty set, the corresponding part of the diagram is shaded. Instead of shading the area, here each zone is labelled with the number corresponding to size of the specific set, as a more intuitive displaying method. Labelling the zones with the size values enables a user to quickly gather information about the related set sizes just by glancing over the diagram.

In case the user combines three concepts to define the query, the diagram will consist of three sets as shown in Figure 6.9. Data items belonging to all query concepts are placed in the intersection between the three sets and the labels indicate the number of items in the set.

Venn diagrams can be useful in a large number of query cases. In fact it has been demonstrated that users mostly formulate key-term queries with no more than 4 terms [92]. In such cases, construction and drawing of Venn diagram provides an intuitive graphical representation and the information it conveys is simple to understand. In theory, even more than 4 contours can be represented using Venn diagrams although the graphical representation becomes somewhat complex.

However, in-order to extend the usability of the proposed framework and create flexibility in browsing and retrieval, the constraint on the number of query concepts was a tight constraint. For this reason, investigating various visualisation solutions born from Venn diagram representation, such as InfoCrystal [179], the decision was to find alternative visual and interactive solutions or elaborate and expand the diagram-based ones.

The later adopted solution, explained in Section 6.4.3, employs simpler visualisation with the increased interactive functionality for replacing the Venn diagram.

6.4 Final visualisation solution for image exploration and retrieval

Based on the initial implementation of the interface for image exploration and the related issues discussed in the previous section, further improvements have been performed in order to address the stated issues of the proposed framework. The graphical solution, as shown in Figure 6.10, for visualising the query space in implementation explained in this section, again consists of several integrated visual components. These interface elements complement each other with visual and interactive functionalities. Each of the components aims at presenting different information related to the "query space" and enables several dynamic exploration methods.

Element 1 is the *Content space* where the information displayed are images visually arranged into groups (*visual clusters*). The space is highly interactive and enables exploration, detailed image content examination as well as retrieval of specific image items. The features of this interface component are specified in details in Section 6.4.1.

Element 2 is the *Concept space* with dual functionality: it provides an overview and navigation support in the query space. The purpose and its specific functionality are explained in details in Section 6.4.2.

Element 3 is the *Concept chart* which displays the selected concepts and the distribution of images in the "query space". It substitutes for the previously explained approach with Venn diagrams for the reasons stated in Section 6.3.2. Interactive functionality of the chart diagram enables sub-filtering of the "query space" content. It can be used in case the user wants to examine pairs of concepts within the selected set and see the images in their intersection. The information presentation approach as well as the interaction functionality are specified in Section 6.4.3.

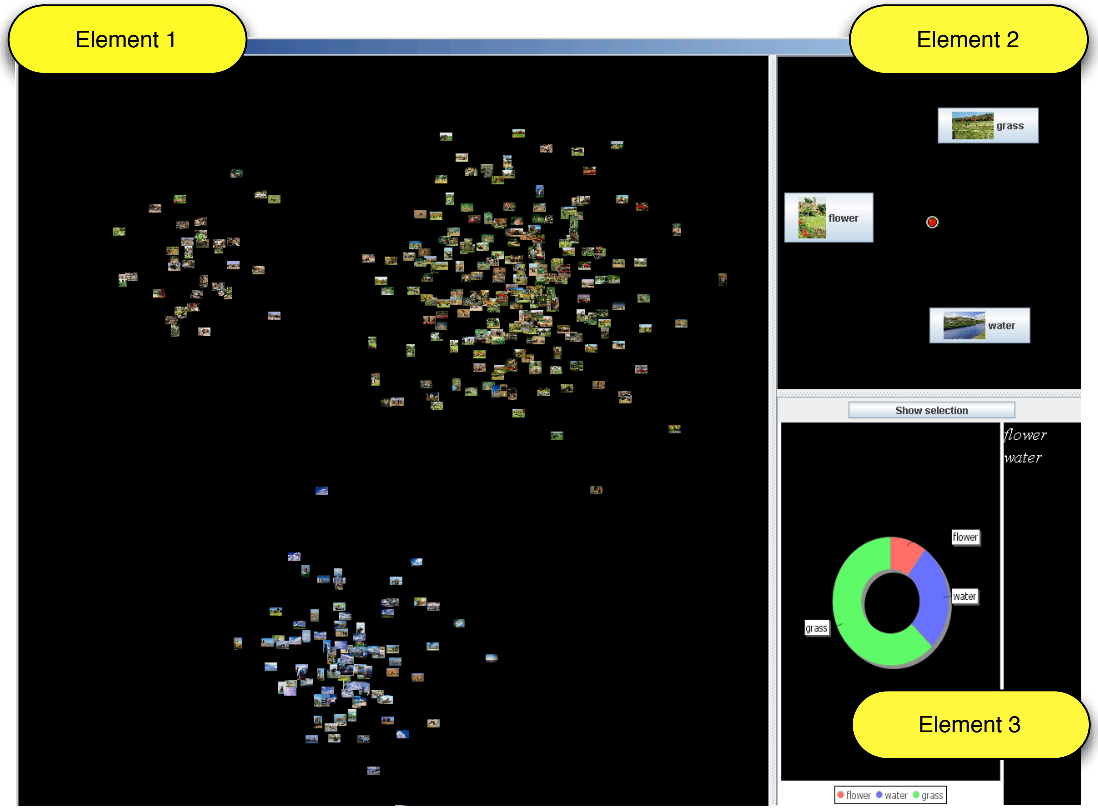

Figure 6.10: Visualisation of the *query space* for image exploration with three interface elements: *Content space*, *Concept space* and *Concept chart*.

It allows the user to adopt any of potential tactics, shows more content in case a user can not find content where expected without duplicating the images on display.

6.4.1 Content space

Observing the displayed grid of available query classes (Figure 6.3), the user can choose one or more concepts to specify the query. After the selection is performed, the system filters the content in preparation for its display. This filtering should enable the presentation of the data in a way suitable for the query and meaningful to the user.

Lets assume that when a user selects several concepts at the same time he/she has one of the two expectations:

- Case 1 - all selected query classes should be detected in the relevant content (q_i AND q_j AND $\ldots$ q_k);
- Case 2 - relevant content can be labelled with one or more selected concepts (q_i OR q_j OR $\ldots$ q_k);

The Case 1 would mean filtering out only the images where all concepts are detected, whereas in the Case 2 all images which contain at least one of the selected concepts are filtered.

In order to choose the method for filtering (between AND and OR) three facts are important to be considered:

- The user does not know which way to specify the query in order to obtain good results;
- The user does not know what kind of content there is in the data-set;
- The results of the classification are not reliable.

In order to address the stated issues by providing flexibility of the framework, filtering based on Case 2 is adopted. Filtering images that have at least one selected query class is considered here a better choice (initial premise) since it: (i) would display broad range of images as a result of a single query, (ii) would enable gaining wide knowledge of the content domain and (iii) the large number of miss-classified images will not stay “hidden”.

Graphical layout

After filtering the images in the manner stated previously, this extracted image set needs to be visualised in a meaningful way. In order to organise image content on the screen and reduce information overload, this visualisation solution employs *visual clusters*. However, in order to improve the visualisation solution of the *content space* specified in the previous section, the clutter of the visual display is addressed here by reducing the number of image groups on the screen. The question is how to organise the filtered set of images within smaller number of visual entities.

Having binary information about the images and concepts as shown in Table 6.1 does not provide much assistance for arrangement of images for their visual display. In other words, there is no indication if one image is more prototypically *grass* or more *flower* thus no semantic logic is exists for assigning images to one *visual cluster*. The solution adopted here is based on the order in which the user is making selection of query classes for query specification. For example, the user who is looking for the *city panorama* will choose the concepts in the following order:

- building
- sky
- car

The *building* is considered concept one (C1), *sky* is concept two (C2) and *car* is concept three (C3). The respective visualisation diagram is shown in the Figure 6.11.

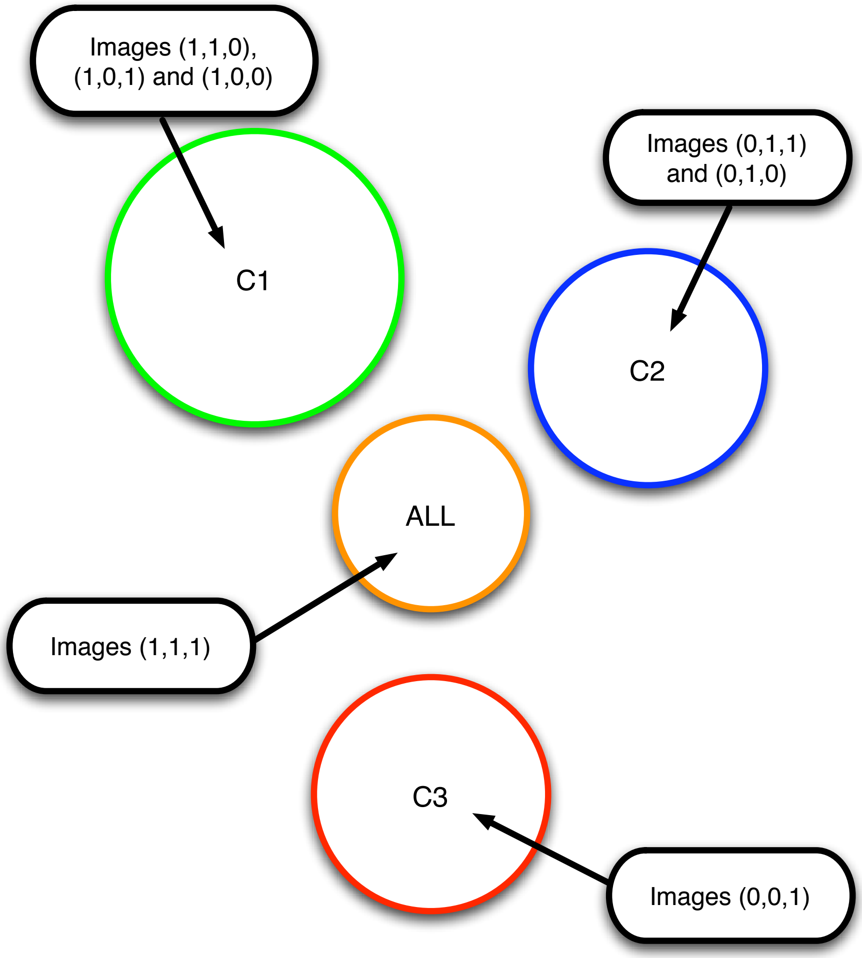

Figure 6.11: Layout diagram showing image placement in *visual clusters* in case of three selected query classes.

Following the example given in Figure 6.11, in case of 3 selected concepts the content visual arrangement is performed in a following manner:

1. Central *visual cluster* (ALL) will contain images that have all the selected concepts detected and is placed in the centre of the display;
2. First selected concept (C1) will contain images where C1 is detected and they do not belong to cluster ALL;
3. Second selected concept (C2) will contain images where C2 is detected but which are not in C1 and ALL;
4. Third concept (C3) will contain images where C3 is detected and not already placed in any of the previous clusters.

This way a sequential set will not contain images allocated to previous ones. It avoids duplication of images so the user does not have to examine the same content more than once. Since all images are available or displayed depending on their number there is no information loss when using this approach.

In cases where the confidence level would be specified for each concept and each image, this information could be used for a more intelligent approach. The images could be assigned to the concept with the highest confidence level during the image visual arrangement process.

As already stated, images are visualised as distinctive *visual clusters* are placed in spatial positions in order to be visually distinguishable. In particular, they are positioned following the logic of graphical presentation of the set intersections, e.g, the zone of intersection is in between the involved sets. Following this approach, a *visual cluster* with images containing all query terms is placed in the central display position.

Adopting the solution from the previous version of the *content space* the *visual clusters* are positioned using a circular layout approach, as shown in Figure 6.8.

The Figure 6.12 displays the case where the user has selected three concepts: *grass*, *flower* and *water*. As said, the content is filtered and four data sets are created. These four sets are mapped into four visual clusters: grass, flower, water; and intersection of the three. The user can then examine sets of images in *grass*, *flower* and/or *water* separately or directly focus on the images containing all three.

The visual representation of the four visual clusters (instead of three single concepts) should not negatively influence the retrieval process or increase the number of displayed images. On the contrary, it can reduce the retrieval time by directing user's focus to the visual cluster containing images labelled with all the selected concepts.

If the relevant content is not found within the multi-labelled images, the query is easily re-directed to other *visual clusters* in the same visualisation window. The fact that the concepts are meaningfully combined by the user, implies the increase of the probability of finding relevant image/images within displayed content without initialising a new query.

Methods of interaction

Displaying a large number of images means reducing their size to the point when immediate understanding of image visual information is hard (or not easily achievable). On the other hand, keeping a meaningful size, would produce an overflow of information on the screen. For balancing these methods two interaction methods are employed here:

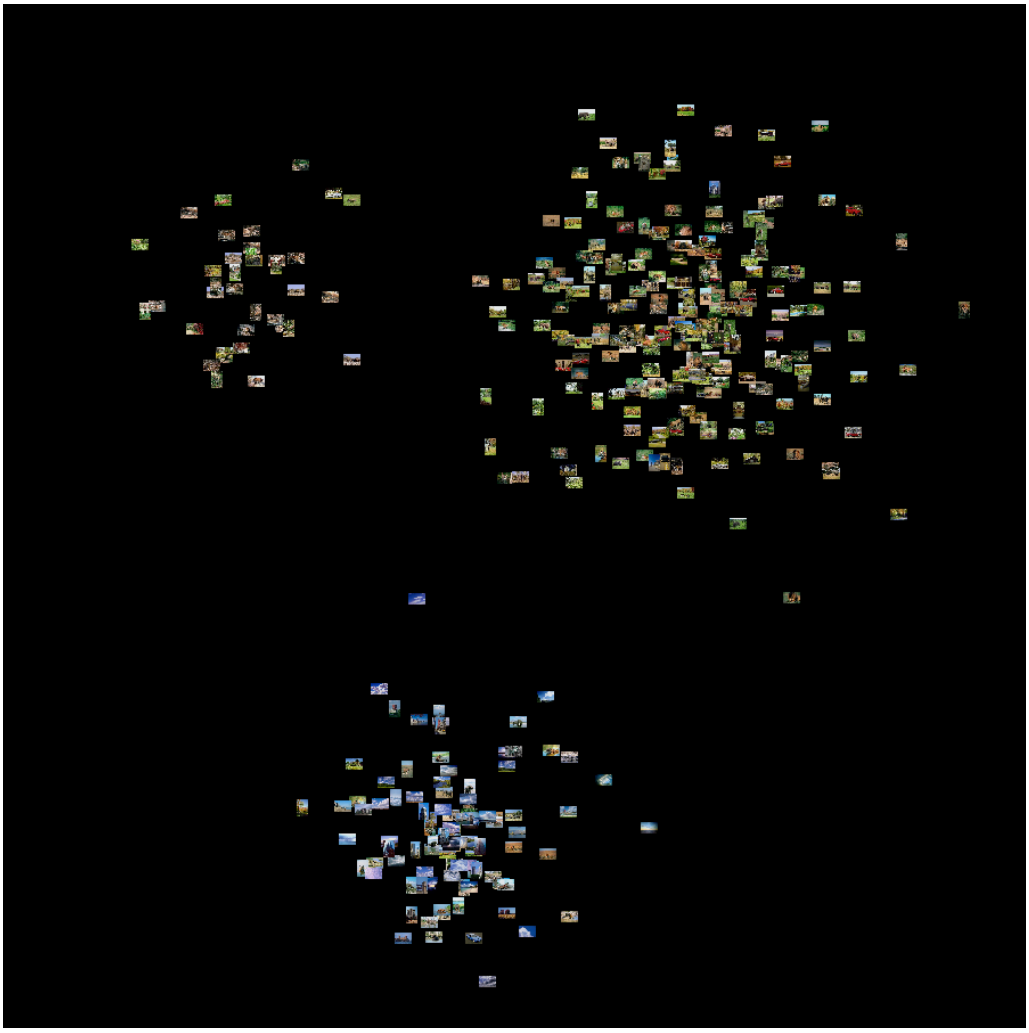

Figure 6.12: Graphical layout example of three selected query classes.

- distortion;
- “click-to-clear”.

Interactive distortion technique is employed to dynamically modify the visual space in which the content is mapped, thus provides “focus+overview”. A Fisheye distortion technique [160] used here, enables user to magnify the area of interest while minimising other data. By moving through the displayed images the user explores the content and increases his/her domain knowledge. Since distorted space can induce confusion with the user, the global position within the *content space* can be followed using the *concept map* in the Element 2 (Figure 6.10). In fact, the red dot icon in the the concept map is synchronised with users movements in the *content space*. This feature allows the user to understand which visual cluster is currently being explored. One example of this feature is shown in Figure 6.14 where the concept *grass* is currently explored. It can be seen that the red dot in the Element 2 is on top of the *grass* image icon.

In order to improve the retrieval support this framework provides to the user, a novel interactive functionality is added to this part of the interface. This feature is refereed as "click-to-clear" provides a means for marking the relevant images while at the same time, deals with the image overlap. In case the user finds one relevant image, by clicking on it this image, it disappears liberating the space it previously occupied. This feature enables iterative changing of the visualisation space making the visual set representation more clear. Once clicked upon, the image ID is automatically stored in the database as a retrieval result for this user session.

6.4.2 Concept space

Changing query directions too often, when dealing with large image database, can be frustrating for the user especially in the first moments of dealing with the system. If non of the relevant images can be found for a long time user satisfaction will decrease. This is one of the reasons why displaying a large set of images in one place is good in this framework. This is especially true when an interactive method such as distortion, enables content management in terms of easy exploration. However, if displays change focus, the user can get lost in understanding which image group he/she is exploring at the moment and which *visual cluster* content is enlarged by distortion. For this reason the upper right part of the *query space* visualisation is dedicated to the window with a dual functionality. First, it shows the concepts which the user choose to construct the *query space*. Second, it serves as a graphical navigation support during the *content space* exploration.

A concept map is usually defined as a simple graphical representation of complex information. Here it is used to preserve the global information regarding the concept set or "query space" the user is exploring at the moment. A basic version of the concept map is used for graphical representation of the subset filtered by the query as shown in Figure 6.13.

Once the user selects n concepts, a map is created where each concept is represented with an icon selected from the relevant data set and a textual label naming the concepts.

The *concept map* is not interactive by itself, but it assists the dynamic exploration process by following the interaction in the first window of the interface. Apart from an overview of what the user is currently exploring this interface component also serves as a navigation map of the content space during user exploration.

"Distortion", which is used for image exploration and examination of content details can be a too complex interaction method in some cases. The initial user testing showed that a distorted view can sometimes cause a loss of orientation for the user while exploring. For this reason as a user moves and explores the items in

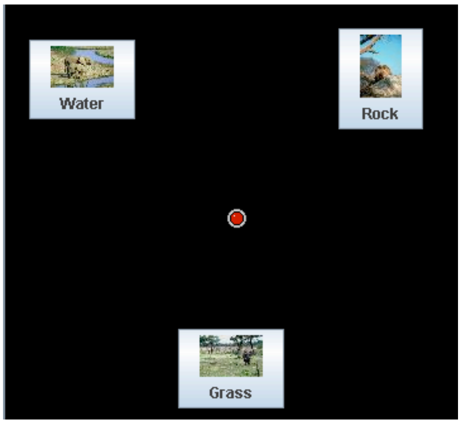

Figure 6.13: *Concept map* of the query space for visual identification of selected concepts and additional navigation support.

the content space, the red dot icon is moving between the concepts in the concept map accordingly. This is shown in the Figure 6.14.

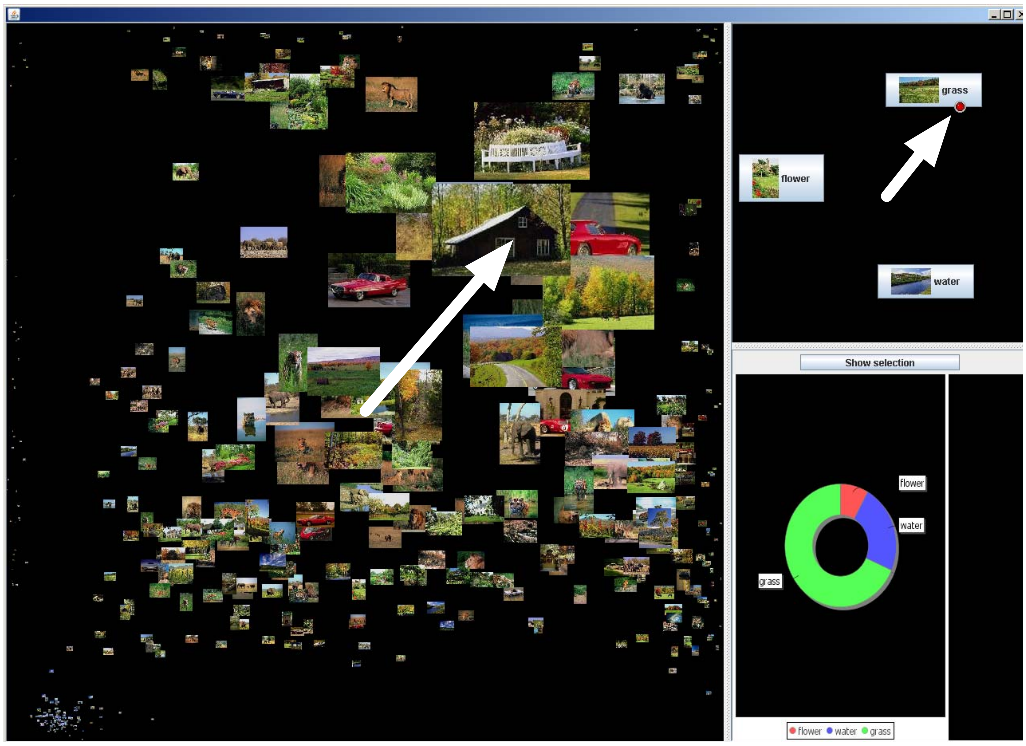

Figure 6.14: The navigational support for image exploration provided in the concept map.

While exploring the images in the *grass* set, the red dot icon will be positioned on (or close to) the *grass* icon in the *concept map*. During the exploration of the images in which all query terms co-exist, the red dot will be visible in the centre of the concept map.

6.4.3 Concept chart

Finding an appropriate visualisation technique for displaying the relationship and the content distribution between concepts in the query space was not an easy task. The initial implementation involved Venn diagrams (as described in Section 6.3.2) with a restricted number of concepts to be combined. however, Venn diagrams have the issue of being too complex for representing more than 4 sets. As stated before, after examining more advanced solutions based on diagram sets representation, such as InfoCrystal [179], the conclusion was that the complexity of such visualisation was higher than its benefits. For this reason it was decided to employ a visually simpler solution in combination with interaction methods which would provide information upon specific user request. Following such a line of thinking, the required visual representation aims at displaying different concepts in distinctive manner, presenting information about the number of image items in each concept and the number of items in the intersection of all concepts.

Charts are in most cases simple and easy to understand graphical representations of data distribution. One of its forms is a pie chart which has a circular shape and it represent parts of a whole in form of different sectors, creating a full circular shape. This visualisation solution is considered adequate for the proposed framework, due to its scalability and constrained display space. The scalability refers to the ability to support an arbitrary number of query classes.

Each chart section represents one concept from the *query space* and all wedges provide the overall overview of the observed data or data-set. The sectors are in most cases differentiated with different colours. However, colour encoding cannot be considered completely scalable. In situation where chart has excessive number of segments the shades of the same colour would have to be used which is not a good visualisation approach. Within this MC browser framework, it is considered that the number of mid-level semantic concepts which can be successfully detected by machines is limited. In addition to this, the number of these simple concepts selected for one query specification is not expected to be big (especially if observing the key-word query specification which usually does not exceed 4 words). For these reasons the number of colours used for pie chart creation is small enough to use distinctive colours for different chart segments.

In the proposed visualisation solution the variation of the bar chart, a *doughnut chart* is used as shown in Figure 6.15. The reason for this is the fact that the void in the middle of the dough can be related to the “all concept” image set for displaying relevant information (e.g. number of images).

As can be seen in the same figure, each segment is assigned with textual label, which corresponds to the name of the concept. The size of the segment is determined by the number of images labelled with this concept. Users can click on any of the segments in order to visualise the images where these concepts are detected. It can be useful for exploring the subset of the query space.

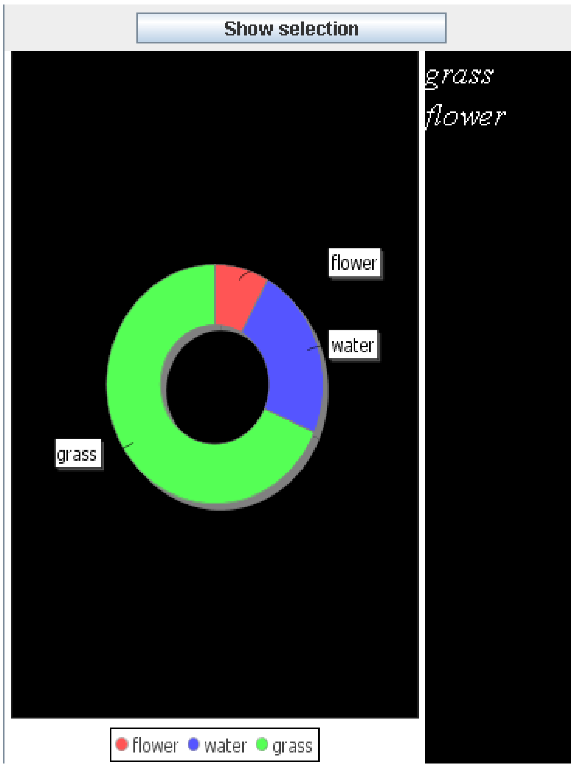

Figure 6.15: The bar chart showing distribution of images through selected concepts.

6.5 Evaluation

The Multi Concept browser application is based on a classifier which provides a semi-automatic machine based core. The goal of this application is to build up on the core and extend the usability of the generated image meta-data on a broad set of retrieval tasks. Since the application uses visualisation for image retrieval, both visualisation and retrieval properties were assessed within the evaluation.

In the image retrieval domain, a standard solution for extending the set of user queries in one retrieval system is to develop new concept detectors (classifiers) to cover new user queries (thus defining new tasks). Since we propose a system which, due to its interactive visual nature, allows the user to compose new queries in a flexible way without extending the concept set, the evaluation was mainly performed as a comparison between these two methods. Using the same users for testing both approaches, i.e. the semiautomatic machine based classifiers and the multi concept browser, we made sure that the human parameter does not change (as much as it is possible). With the same users and the same user tasks, we tested both systems for retrieving images. In addition to that, the multi concept browser was also compared to a relevance feedback method.

The specific goals of the evaluation were:

- To compare the retrieval performance of the proposed application with standard machine-based retrieval method (classifier);
- To compare the retrieval performance of the proposed application with a standard user-oriented retrieval method (relevance-feedback);
- To test the user satisfaction with the system.

In order to compare the performance of the system we used a summative evaluation technique relying on the standard retrieval evaluation metrics, mainly based on precision and recall measures, as explained in the Chapter 6 of the thesis. In order to access the user satisfaction with the system a formative evaluation was conducted using questionnaires. Different questions were answered at different stages of the evaluation. Those regarding the application functionalities and the realisation were given at the end of a 40 min user session. All questions were in the form of statements, with which users had to agree or disagree. All provided statements were offered with a positive connotation.

During the evaluation, all users' behaviors were observed in order to draw out potential problems with the application, collect negative opinions about features and methods used, register users' preferences, frustrations and so on.

6.5.1 Evaluation metrics

Since the proposed Multi Concept (MC) browser is an user-oriented tool for visual exploration and retrieval of images, the selected evaluation metrics should assess usefulness, user satisfaction and system performance. For evaluating the usefulness and user satisfaction a questionnaire was compiled and given in Appendix C. Users answered the questions thus providing their opinion and giving feedback on the proposed solution. On the other hand, since the proposed MC browser tool is developed for the purpose of image retrieval it's efficiency is evaluated based on evaluation metrics used in the retrieval domain.

Various methods for evaluating the performance of information retrieval systems have been proposed and used. Standard measures are based on defined queries and existence of ground truth information regarding these queries.

The most comonly used methods for evaluating retrieval system performances are *precision* and *recall*. Precision is by definition ratio between the number of relevant retrieved data items and the total number of retrieved data items.

$$\text{precision} = \frac{|\{\text{relevant retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \tag{6.9}$$

Recall is ratio between the number of relevant retrieved data items and total number of relevant items in the data-set thus recall measures the rate of successful task accomplishment.

$$\text{recall} = \frac{|\{\text{relevant retrieved documents}\}|}{|\{\text{total relevant documents}\}|} \tag{6.10}$$

F-measure is considered as a measure of trade-off between precision and recall. In general, greater precision decreases recall and greater recall leads to a decreased precision. Its formula is:

$$F_\beta = \frac{(1+\beta^2)\cdot \text{precision}\cdot \text{recall}}{\beta^2\cdot \text{precision} + \text{recall}} \tag{6.11}$$

In case *precision* and *recall* are weighted equally $\beta = 1$, if $\beta > 1$ precision is favoured and if $\beta < 1$ recall is favoured.

6.5.2 Initial system evaluation

As stated before, the initial testing of the proposed framework was performed on the data-set containing 700 images from Corel data-set [45]. As a result of the first retrieval stage the entire data-set has been automatically labelled using the 10 pre-defined concepts. The retrieval efficiency of the system was tested on four high-level queries: *open landscape*, *modern city view*, *wild life* and *rural gardens*. Relevant images in the database were manually annotated with same four concepts for establishing the ground truth.

Evaluation procedure

In order to initially evaluate proposed system and determine further improvements, the user tests were conducted with 3 users for 4 high level queries. In total 12 test results have been obtained as well as additional users' feedback regarding the proposed system. User tasks were to find and select as many images as possible, relevant to four given queries: *open landscape*, *modern city view*, *wild life* and *rural garden*. The number of images in each category (according to the ground truth) is given in Table 6.2.

During the user testing the focus was on several aspects relevant to the system purpose as well as performance. Apart from evaluating the efficiency of the proposed

	Open landscape	Modern city view	Wild life	Rural garden
Ground truth	139	58	300	77

Table 6.2: Number of images in 4 test concepts according to ground truth.

Task	Simple concepts used	Recall
	User 1	
Open landscape	water, grass	41
Modern city view	building, cloud	67
Wild life	lion,tiger,elephant	42
Rural garden	flower,water,grass	82
	User 2	
Open landscape	water	13
Modern city view	building	55
Wild life	lion, rock, grass, tiger, elephant	77
Rural garden	flower	16
	User 3	
Open landscape	water, grass, cloud, car, elephant	42
Modern city view	cloud, building, car	83
Wild life	lion, tiger, grass, elephant, rock	70
Rural garden	flower, water, grass	79

Table 6.3: Results of the initial user evaluation per user and per query given in form of system recall.

system, an additional observation was made on which combination of concepts different users used for query specification. For future reference we also examined users behaviour and comments during the test sessions.

Efficiency of system was measured as recall of the performed retrieval conducted within the time interval of 10 min.

Evaluation results

Results of User 1, User 2 and User 3 initial evaluation regarding the system efficiency are given in Table 6.5. The table shows combinations of concepts for each query and the obtained recall values.

It can be noticed that the number of simple concepts combined affects the retrieved results. The users which requested more simple concepts retrieved more relevant images.

As it can be observed from Table 6.5, the proposed system is able to achieve good performance in helping the user to retrieve abstract query terms from images. Moreover, comparing the results of the Users 1 and 3, it can be concluded that when multiple concepts are employed in assisting the retrieval, better performance can be obtained within the same time limitations.

6.5.3 Second stage performance evaluation

For the final evaluation of the proposed visual-based retrieval framework a repository of 3500 images was used. All images were manually annotated and those indices used as a ground truth information. Within this data-set, the same set of 10 query classes were identified: *flower*, *grass*, *rock*, *water*, *building*, *car*, *cloud*, *elephant*, *lion* and *tiger*. Apart from these 10 mid-level semantic concepts, images were also annotated for 8 high level semantic queries/concepts: *city street*, *mountain view*, *boat*, *waterfalls*, *flower fields*, *wild life*, *modern city view* and *rural garden*. The quantity of images according to semantic indices in the ground truth are given in Table 6.4.

The designed test for the proposed visual image retrieval framework consisted of two complementary parts which aimed at evaluating system performance and user satisfaction, or in other words test system efficiency and effectiveness. System efficiency was evaluated through the comparison of the proposed visualisation framework with two standard types of system in image retrieval:

- Retrieval of high level queries based on Support Vector Machine (SVM) classification results;

- Content-based retrieval combining classification and Relevance Feedback mechanism (RF).

The set-up of the first evaluation environment is shown in Figure 6.16 where q are the query classes (or mid-level concepts) used in the MC browser (e.g. grass, building, car) for specifying high-level queries hq (e.g. boat, waterfalls, city street). In the first case, semi-automatic classifier is used to classify images into mid-level concepts q and in the second case into high-level concepts hq.

For example, for the query “boat” the classifier (SVM) was trained for the concept of boat and then ran over the entire data-set. The resulting set of this classification was used for comparison with the results obtained using the proposed Multi Concept browsing framework. The details and results of this evaluation are given in Section 6.5.3.

In the second part of comparison, RF is used due to the fact that it is more user-oriented since it gathers the users “opinion” through iterative user-computer communication. For this comparison the used system configurations are shown in Figure 6.17 and results are reported in Section 6.5.3.

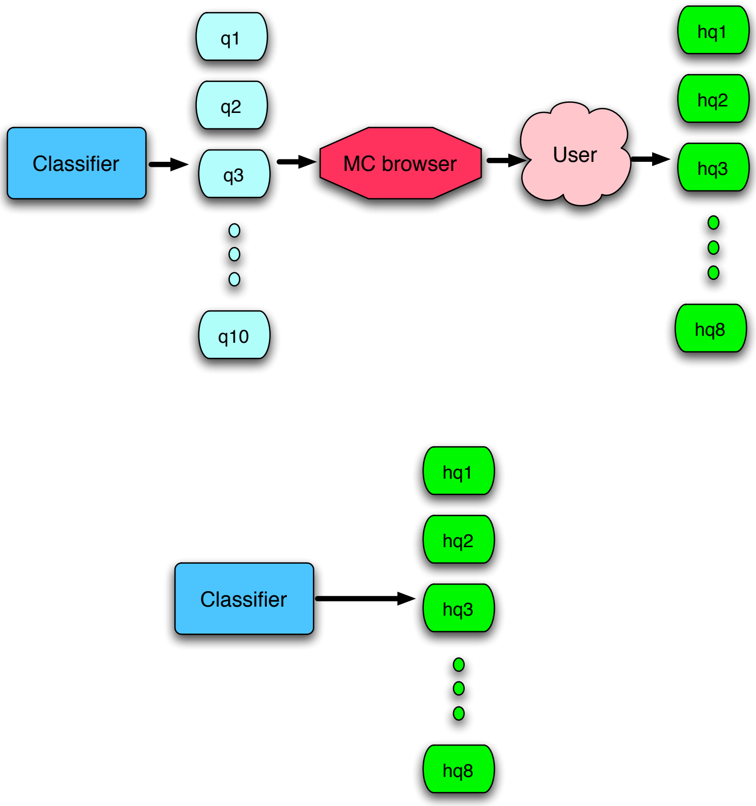

Figure 6.16: The diagram of evaluation set-up for comparing performance between MC browser and the SVM classifier.

CHANGE In order to examine the usability-related aspects of the MC browser user interface for image retrieval, a formative inspection usability evaluation method called "cognitive walkthrough" was used. As stated in [141] "Cognitive walkthrough is usability inspection method to evaluate a design for ease of learning by exploration", and it is usually performed by interface designers or his/her peers. The goal is to model the real usersÕ behaviour while trying to perform the user task using the interface tools. In addition to this, the user satisfaction results and feedback were obtained using the designed user questionnaire given in Appendix C. The questions in the given questionnaire aimed to evaluate important features of the interface and evaluate user satisfaction with the proposed solution.

Image indexing for performance comparison

As said before, the first part of the performance evaluation compared the results achieved using the Multi Concept browser and the classification results of the SVM for the same high-level queries. Both mid- and high-level semantic image indices available in the proposed framework were generated by same classifier. For generating semantic image indices for the 18 selected mid-level and high-level concepts

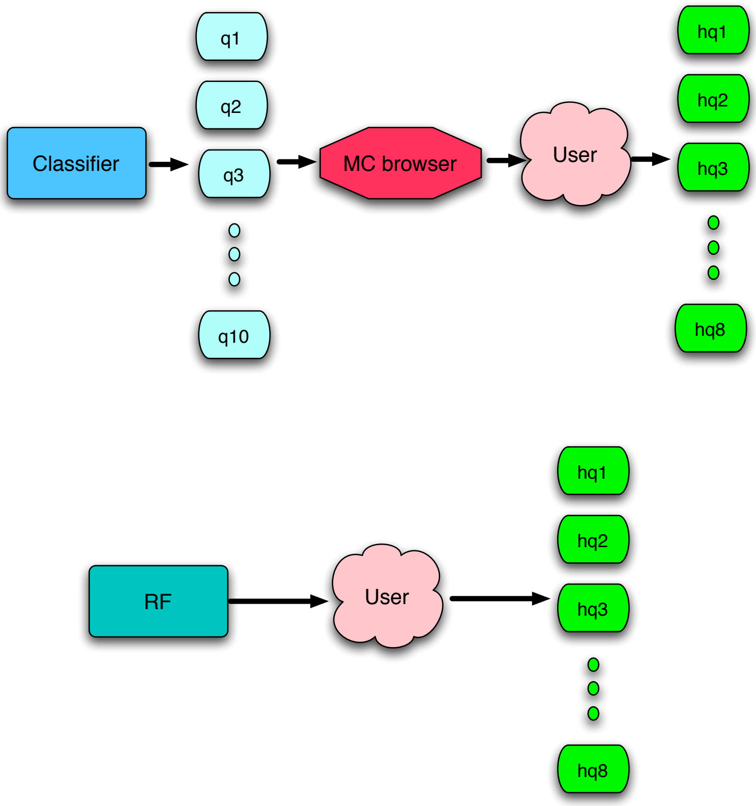

Figure 6.17: The diagram of evaluation set-up for comparing performance between MC browser and RF framework.

Support Vector Machine (SVM) classification algorithm was used. Classification was performed based on MPEG-7 Colour Layout and Edge Histogram visual features. The implementation of the SVM used here is SVMlight [88], described in Appendix C.

The SVM classification generated three set of results for all 18 semantic concepts. This was achieved using 3 different sized training sets. In the first case the training set consisted of 10% of positive examples relative to ground truth and the same number of negative examples. In the second case this number was equal to 30% of positively annotated images and in the third case it was raised to 50%. Recall of three classifies for all 18 semantic concepts, SVM with 10% training set, SVM with 30% and SVM with 50% training set are given in Table 6.5, respectively.

Evaluation procedure

User tests for efficiency evaluation were performed with 8 users with various experience levels in image retrieval domain.

Standard statistical data was collected and processed before the evaluation started and they are given in Tables 6.6 and 6.7. Table 6.7 shows users' previous experience

Concepts	Ground truth [no images]
lion	102
water	1081
grass	1064
building	1271
car	182
cloud	708
rock	922
tiger	102
elephant	109
flower	730
boat	234
flower fields	32
modern city view	168
rural garden	182
mountain view	598
waterfalls	15
wild life	322
city street	191

Table 6.4: Number of images annotated using 18 concepts in ground truth set.

with image retrieval systems (Google images, Yahoo image search, etc.), while in the second column and users' previous experience with the image retrieval tools and algorithms (image indexing, annotation, classification, relevance feedback, etc.) in the last column. As seen in Table 6.8, mark 0 corresponds to no previous experience and mark 5 to high previous experience.

Comparing the Multi Concept browser with SVM

After answering initial set of question given in "Pre-evaluation" part of the questionnaire given in Appendix C users were given an explanation about the motivation and functionalities of the proposed framework. Following an initial tutorial, users were given a trial period to get more familiar with the tool. For most users it consisted of performing one sample query for 3-5 minutes. Once users stated that they were comfortable with the tool, the evaluation procedure was initiated.

As previously stated, a set of 4 queries were given in order to test the proposed system on various query levels. Each query was performed by combining a set of concepts and users were completely free to select any combination. The combinations used for each query are also noted as part of the evaluation results.

The user was instructed to select images considered relevant, by clicking on them.

Concepts	SVM 10%	SVM 30%	SVM 50%
		Mid-level concepts recalls	
lion	0.853	0.902	0.882
water	0.67	0.757	0.82
grass	0.97	1.0	0.97
building	0.686	0.787	0.5
car	0.4	0.55	0.54
cloud	0.613	0.76	0.839
rock	0.557	0.684	0.804
tiger	0.784	0.892	0.892
elephant	0.679	0.917	0.963
flower	0.809	0.830	0.88
		High-level concepts recalls	
boat	0.624	0.765	0.808
flower fields	0.375	0.75	0.843
modern city view	0.47	0.756	0.815
rural garden	0.791	0.841	0.901
mountain view	0.766	0.838	0.873
waterfalls	0.6	0.933	0.933
wild life	0.67	0.838	0.907
city street	0.686	0.775	0.906

Table 6.5: Recall of SVMlight classifier for mid-level and high-level semantic concepts for 3500 images.

Female	Male	Age span
4	5	22-34

Table 6.6: Users' statistical data.

User	Exp. with img. retrieval sys.	Exp. with img. retrieval tools and app.
1	3	3
2	0	0
3	5	5
4	4	4
5	4	1
6	2	1
7	4	4
8	5	5

Table 6.7: Users' previous experience with image retrieval systems, tools and algorithms (marked from 0 to 5).

Mark	Answer
No previous experience	0
Very little experience	1
Little experience	2
Some experience	3
Good experience	4
Extensive experience	5

Table 6.8: Marking system in Likert scale for user experience with image retrieval systems and tools.

User	boat	flower fields	city street	waterfall
1	63	50	86	12
2	47	43	87	13
3	62	50	130	15
4	60	117	97	15
5	59	41	88	11
6	104	51	101	8
7	40	113	91	12
8	89	86	160	6

Table 6.9: Number of retrieved images per user for 4 high-level queries using MC browser.

User selections were stored and the results compared with the SVM high-level classification results in terms of recall and F-measure. As part of the evaluation regarding the visual and interactive solution "post-evaluation questions" were answered by users. Results were analysed and described as final user satisfaction evaluation.

The indexing results used for testing the proposed retrieval tool were SVM results with the smallest training set (using 10% of positives from the database, corresponding to the first column in Table 6.5).

Table 6.9 shows the numbers of retrieved images using Multi Concept browser per user and per query. It is interesting to note that although the ground truth value for *flower fields* is 32, all users found more than 32 images relative to this query. This is clear evidence of the fact that even when there is manual annotation of one semantic concept, the interpretation of the concept by the users is different than the annotation person. This fact also influenced recall values for this query as marked with in Table 6.10.

Due to the different natures of the proposed system and the classifier, direct comparison is not possible. Within the proposed framework, the user is the one selecting relevant images, thus precision of the system is equal to 1. In order to have a common measure for both systems, the F-measure was used to compare their

Query	MC browser [av.]	SVM 10%	SVM 30%	SVM 50%
Boat	0.437	0.228	0.249	0.264
Flower fields	1*	0.029	0.039	0.055
City street	0.714	0.2	0.219	0.221
Waterfall	0.873	0.018	0.022	0.016

Table 6.10: Comparison of f-measure between MC browser and SVM for four high-level queries (* marks the case where user found more images than ground truth).

Query	MC browser [av.]	SVM 10%	SVM 30%	SVM 50%
Boat	0.28	0.624	0.765	0.808
Flower fields	1*	0.375	0.750	0.843
City street	0.556	0.686	0.775	0.906
Waterfall	0.775	0.600	0.933	0.933

Table 6.11: Recall results comparison between MC browser and SVM for four high-level queries (* marks the case where user found more images than ground truth).

performances. On the other hand, considering the fact that precision of the proposed system is 1, it was considered necessary to compare also the recalls of two systems.

Table 6.10 shows the average F-measure per query as a result of the MC browser and F-measures of all 3 SVM classifiers used. As can be seen here, when comparing the F-measure of two systems the proposed tool outperforms the SVM classification for all queries. In order to provide a more balanced comparison, in addition to F-measure also recall of two approaches is compared. In Table 6.11 a comparison regarding retrieval recalls are given between the proposed MC browser and the three SVM classifiers.

Comparing the Multi Concept browser with RF module

The second evaluation method for testing the system performance was the comparison with a standard interaction method used in image retrieval system, the Relevance Feedback (RF). RF is a query-by-content type of retrieval system where users refine and accomplish queries through multiple iterations. The screen-shot of the application used for this comparison is shown in Figure 6.18. The images bordered with green are marked as relevant and red as non-relevent by the user. Blue border marks the images selected by the user as “exact match” to the query (query boat in this example). One noticeable drawback of the RF system, is that when facing high level queries, the user does not know when he/she will initialise the query. In other words, in which iteration the user will find a suitable positive example to initialise the query. It might happen that user has to go through 10 iterations and see the first relevant image in the 11th iteration. In such cases, the user can be discouraged in using the system. Table 6.12 shows the number of retrieved images per user and per query while the comparison of average number of retrieved images when using MCB and RF per query is given in Figure 6.19.

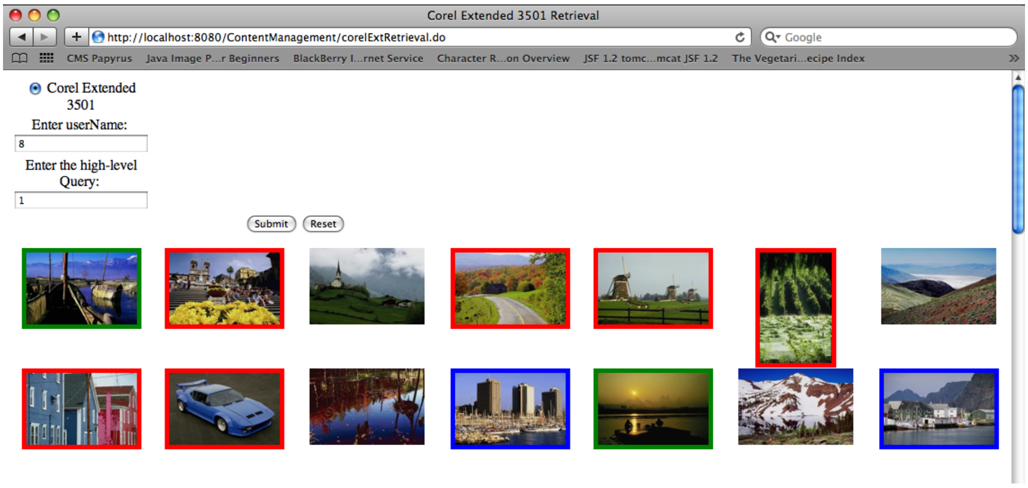

Figure 6.18: The screen-shot of the RF framework interface, used for performance evaluation, showing example for query *boat*.

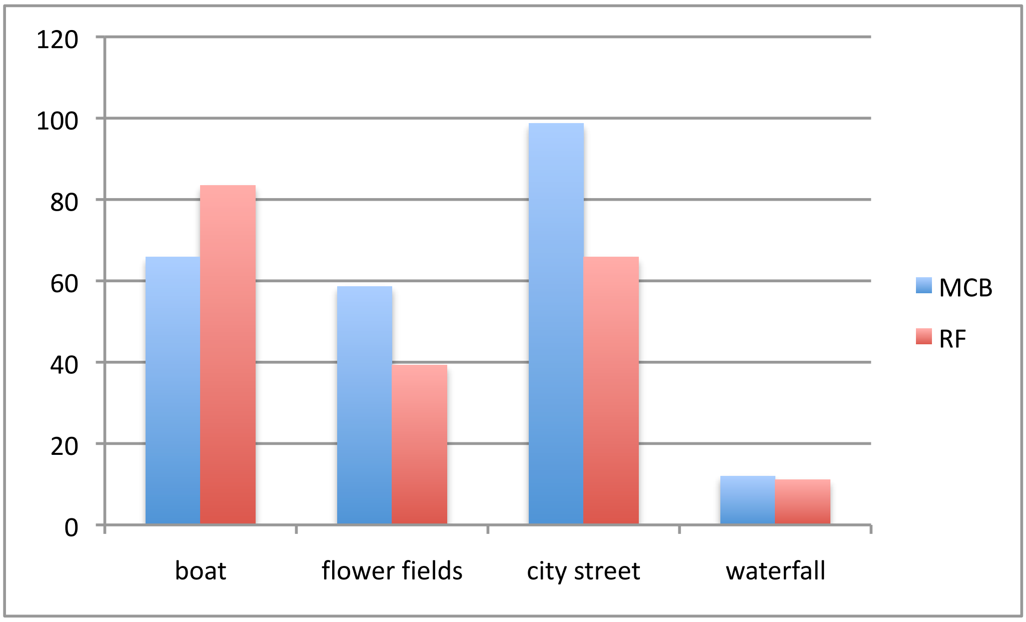

Figure 6.19: Comparison of average number of images retrieved per query between MC browser and RF system.

6.5.4 Usefulness of displaying concept relationships

At the beginning of a user session users were directed to examine the *Concept relationship display*, specified in Section 6.2.2 and shown in Figure 6.20. They were

User	Query	MC browser	RF
1	boat	63	81
	flower fields	**50**	39
	city street	**86**	30
	waterfall	12	12
2	boat	47	76
	flower fields	**43**	29
	city street	**87**	72
	waterfall	**13**	8
3	boat	62	86
	flower fields	**50**	43
	city street	**130**	77
	waterfall	**15**	13
4	boat	**60**	58
	flower fields	**117**	57
	city street	**97**	77
	waterfall	**15**	11
5	boat	59	106
	flower fields	**41**	35
	city street	**88**	73
	waterfall	11	14
6	boat	**104**	94
	flower fields	**51**	33
	city street	**101**	66
	waterfall	8	9

Table 6.12: Comparison of numbers of images retrieved by MC browser and RF framework within 10 minutes.

asked to comment on information they are able to infer from this visual representation, without having any previous knowledge on the content.

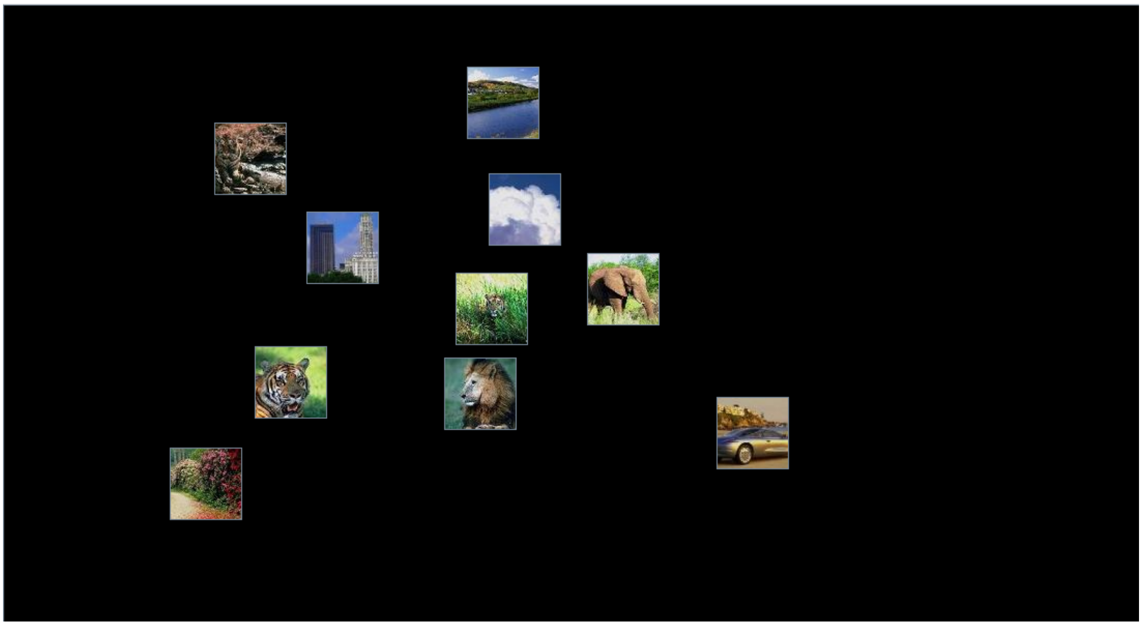

Figure 6.20: Concept relationship display was evaluated in terms of usability.

Related answers and comments are listed below:

- "Visualisation is helpful and presents relationships between concepts, e.g. images of grass is relevant to images of sky and images of animals";
- "A large number of high level concepts can be build straight from the combination of the given ones. Automatic classifier results might not be very accurate";
- "There are a large set of pictures of lions and elephants taken from long distance; pictures of cars will not show much of the landscape, they might just show the car; some pictures might have been taken at the zoo; some might show a park inside the city; short distance between *building* and *rock* might be due to the misclassification";
- "As it seems that there are more images combining buildings and tigers than buildings and cars for example, we can assume that there might be something wrong with image classification";
- "I did not use this visual representation too much to decide about the query refinement. However, at the begining, it was interesting to see what it means and why they are organised that way";

- "Although the concept relationships are useful in understanding the image content a bigger examples would be more beneficial".

Analysing given user comments the main conclusion is that users did not entirely grasp the idea of this visual representation in the given time. Giving more time to the users to use the retrieval tool would provide a clearer conclusion regarding this proposed visual element. In particular, it would show if the idea of presenting the concept relationships for the purpose of providing insight into the data-set domain is not very useful or the way it is visually presented is not an optimal way. This additional research is left for future work.

6.5.5 Scalability of high level query specification

One of the goals of the proposed framework is to enable the user to combine a limited set of simple concepts (or query classes) to formulate a wider range of queries. The limited set of concepts should cover the basic content elements of the images found in the data-set, such as: grass, sky/cloud, building, water and so on. In the test environment created for evaluation of the proposed solution there are, as previously stated 10 query classes: *flower*, *grass*, *rock*, *water*, *building*, *car*, *cloud*, *elephant*, *lion* and *tiger*. By displaying visual representations of these classes, the user is able to combine only existing image indices without making wrong query attempts.

In order to test this scalability feature, 3 users were asked to think of 10 high level queries which can be generated using the existing 10 concepts. High-level queries generated by users are given in the Table 6.13.

User 1	User 2	User 3
park with people	lion with tiger	man-made environment
sailing boat	water-flowers	natural environment
cruise ships	poppy flowers	panoramic views
fishing boats	city street with cars	sea shore
cliffs	beach with people	sky-scrapers
family of lions	elephant in the water	village
jungle waste land	rocks on the sea shore	swimming pool
mountains in the winter	tiger and waterfall	rural road
lakes	buildings in the skyline	street with people and cars
open sea	grass under rocks	nature in the spring

Table 6.13: High level queries generated by users based on 10 query classes.

As can be seen from the table, the scalability of the proposed framework enables specification of different types of queries. In other words, the specified query could be requesting more or less detail. For example, the query *boat* is considered as a broad type of high-level-query whereas *boat in the village marina* is a query with specific details. The advantage of the proposed framework is in the fact that both queries can be specified using proposed framework without developing two semantic detectors.

6.5.6 User satisfaction evaluation

As previously mentioned In order to examine the usability-related aspects of the MC browser user interface for image retrieval, a "cognitive walkthrough" method was used. As stated in [206] four things need to be defined before the cognitive walkthrough is performed: "who will be the users of the system, what task(s) will be analysed, what is the correct action sequence for each task and how is the interface defined". The target users of the system are persons completely unaware of the database content, without any knowledge on the way the database is indexed and with different levels of expertise in multimedia methods and tools. In other words, home users trying to retrieve images of interest from an unknown image database using the MC browser. Users with a technical background were asked to act as end users using the system for the first time and without knowledge of the collection. For this evaluation we defined the user task as retrieval of subject images (e.g. with waterfalls or boat). After a user task was defined, testers were asked to perform those tasks in order to assess the application usability. In order to retrieve the images using the MC browser, the user should perform several steps which are considered as evaluation sub-tasks. Following the specifications of the cognitive walkthrough approach, four sub-tasks were identified:

1. Subtask: the user should learn about the system and the content in order to be more efficient in terms of query;

2. Subtask: the user needs to understand which actions to perform to specify the query in this system;

3. Subtask: the user needs to find an image in the returned set of results;

4. Subtask: if the set of images returned by the system is not relevant or image is not there, the user should be able to reformulate the query;

In order to perform the first subtask the user should gather the first knowledge of the content through the use of three component of the first MC browser interface window: the concept grid, the concept list and the concept relationship display. Using the concept grid and/or the concept list, the user should understand which topics are to be found in the data-set. The list also provides information about the number of images per concept, since we wanted to see how useful it is for understanding the content and, later, in formulating the query. In order to understand the context of images, the user should also analyse the concept relationship display and learn about the content.

Very few people testing the application were not aware of the topics regarding concept detection and classification. Working in a multimedia lab even if on different topics, it was not difficult for them to understand the way queries can be composed and what the grid and the list interface components are useful for. However, the list view did not prove its usefulness for the query specification. According to the comments of the people testing the application, it would be more useful to have information about the number of images annotated with the set of the selected query concepts. In addition to this, the number of images per concept was not found useful as a tool for helping the process of query specification.

The concept relationship display was not easily understood by the testers before its idea was explained. After some explanation, we noticed a slightly increased influence this element made on the decisions on how to make a query. This was learnt from the users answers after analysing the user questionnaire. In addition to this, analysing their behaviour during the tests, it was noticed that they did not focus on the relationships between the concepts of interest for the query, but they analysed the entire relationship display. Consequently, the questions were raised regarding, for example, the proximity of concepts which should not be close according to human common sense. It is then assumed that this was an important reason for not using this interface element.

For performing the second subtask, the users were supposed to select a set of concepts and in this way to specify the query. This should be done by selecting some concept representative images in a grid. The concept selection and deselection was easily performed, since the interface components behind the concept images are buttons and, as such, their functionality was easily understood. The color encoding of the selected (red border) or deselected images (no border) was correctly interpreted and used without problems. During the test sessions, the users did not express any negative opinion regarding the method of query specification.

The third subtask concerns the part of the retrieval where the system returns the set of results. In order to accomplish this subtask, the user should browse through the visualised set of images retrieved by the system and either find relevant images or learn enough to be able to meaningfully reformulate the query Due to the large number of images which can be displayed, the user should interact with the content in order to examine details of the displayed image set. Distortion as an interaction tool was easy to use by all testers since its functionality was clear from the first mouse movement. Once the user puts the mouse pointer over the image set, the image dimension changes, enabling the user to zoom on image details.

The concept map as a global orientation method was not heavily used, based on the questionnaire and observations made during the sessions. It is certain that its functionality was clear after the explanation, but still no significant utilisation was noticed. It is left to analyse and test, if it is unusable in its current form or not necessary at all. The subtask four deal with the case where no relevant images were found in the visualised set, and concerns the fact that the user should be able to continue or change his query. The concept bar chart showing the image distribution and providing the interaction support for query reformulation within the selected concept set was utilised by few users. Most of them preferred going back to the main query window and starting a new query using the concept grid.

In order to gather users oppinions in somewhat quantitative way a set of questions was compiled as a part of the questionnaire given in the Appendix C. After using the proposed application users marked the statements using the scale given in Table 6.14. The results of this evaluation part are compiled and given here in terms of average users' mark per each question. The marking system used is shown in Table 6.14. Figure 6.21 shows these results in form of a bar chart.

Mark	Answer
Strongly disagree	1
Disagree	2
Neither agree nor disagree	3
Agree	4
Strongly Agree	5

Table 6.14: Marking system in Likert scale used for evaluating user satisfaction with MC browser.

As can be seen from the answer chart, the highest marks were given to statements 5 and 17, which are "Query formulation is easy since available terms for query are displayed" and "The interactive disappearing of images when selected is an interesting feature". The users highly appreciated the information on which semantic concepts are available for query specification. In addition to this, the novel interaction method "click-to-clear" was found interesting and effective method for image retrieval, which also introduces a game effect. On the other hand, the lowest mark was given for the statement 11: "Number of images per concept is an useful information during a retrieval process". Users did not consider the number of images per concept as an useful information. A more requested information was the number of images that would be displayed for exploration for the selected set of concepts. User

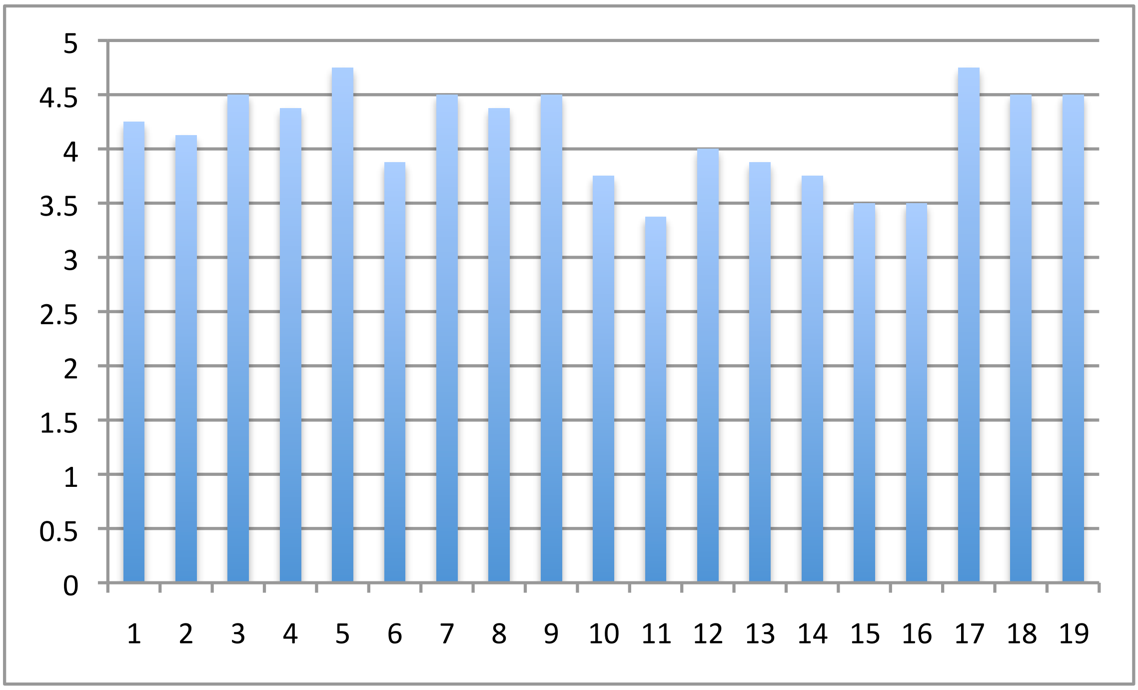

Figure 6.21: Average mark per question for user evaluation of the MC browser.

were also asked to provide positive and negative aspects of the application. Stated positive aspects of the proposed MC browser are:

- The retrieval method is novel;
- The proposed framework is interesting;
- It helps finding wanted images;
- The proposed graphical layout is much better than the grid layout;
- It provides intuitive interaction with the user;
- It has a good interface design;
- It is easy to use;
- It enables fast retrieval;
- The possibility of creation of new high-level query is interesting and novel concept;
- Supported small number of classes makes queries simple;
- The idea of grouping images in a 2D space according to their content is good for image retrieval;

- Interactive disappearing of images is an interesting interaction method;
- The interface has pleasant black background;
- It enables zooming on images pointed on by the user;
- The connection between concept and browsing window enable easy manipulation of query specification and image xploration.

Stated negative aspects about the MC browser are:

- Distortion is a bit dizzying;
- A wider set of basic concepts would be better;
- Some images are zoomed too much;
- It is difficult to focus for a long time (because of distortion);
- Some go back functionality would be useful, if user gets too many images and wants to reduce them;
- A number of images which will be displayed is missing which would guide the user towards inspection of smaller set of images;
- Browsing through large set of images application becomes slow;
- When browsing through a large set of images a lot of overlapping can cause relevant content to be missed;
- Displayed images are sometimes too small;
- Chart visualisation is difficult to understand;
- When zooming on images the resolution is poor.

As can be seen from the comments, the user in general appreciated the novelties incorporated within MC browser such as: interactive query specification, image disappearing and so on. The flexibility of the query method enabled users to generate various combination of concepts for retrieving images. Not once the users were uncertain how to continue with their query due to the intuitiveness of the solution.

Due to the length of the user tests users did not feel comfortable with the distortion as an interaction method. Better evaluation of this technique would be performed in shorter user sessions. The progressive filtering was somewhat an issue for users which were concerned of missing some relevant content.

While answering the question "Would you remove/add any functionality", the users stated the following:

- Inform the user if one concept combination was already selected;
- Annotations already made by previous users could be available with for example right click;
- A possibility of creating and storing new high-level concepts in order to be used in future searches;
- An option of large zooming on thumbnails could be added, with for example "CTRL+mouse click";
- Number of images already retrieved could be a useful information;
- Number of images which will be displayed after concept selection in the browsing interface would be appreciated by the user;
- Show preview of previously selected images;
- Undo functionality after clicking and disappearing one image would be good;

The general conclusion is that the users appreciated the novel browsing and image retrieval tool. Some of them observed it as a challenge, brought by the feeling that was emphasised by "disappearing" images when clicked upon.

Apart from that some general experiences were noted as a result of user comments. Most users found image overlap as an issue for retrieval. For providing a content overview overlap does not impose a big problem. However, when the user needs to find a specific image it can be an important drawback. Distortion was also found eye-tiring due to the fact that most users did all 40 minutes of user-test without any break. In order to properly test distortion as an interactive retrieval tool shorter user sessions should be organised. Image resolution when zoomed is poor due to the fact that images displayed are thumbnails obtained by re-scaling the original images for the visualisation purposes. Attempts to visualise original sized images will produce memory problems at standard computers.

Experience and results obtained during this evaluation procedure provide significant information which can determine further improvements and development directions.

More focused evaluation sessions can provide better conclusions regarding the particular tool elements.

This chapter presented the idea and details of the proposed novel visual-based method for browsing and retrieval of semantically indexed image collection. With the goal of increasing the involvement of human cognition into the query specifying process, the proposed solution aims at enabling the user to compose high-level semantic queries combining a limited set of concepts. Interactively combining such sets of concepts, the user is should be able to specify a wide range of queries without the assistance of specific image labels. The dynamics of users adapting his/her query due to learning-while-interacting is supported by the flexible nature of the system. Visual overview and query-related content filtering support visual exploration of limited image sets thus avoiding extensive examination of individual semantic classes.

The performed user evaluation aimed at testing the performance and usability of the proposed solution. Obtained results indicate that visual and user-oriented, image retrieval environment can be a productive solution in terms of user satisfaction and usefulness.

Chapter 7

Conclusion and future work

This chapter summarises the idea and the motivation for this thesis. It also includes a discussion regarding the work reported throughout the previous chapters.

After investigating approaches from different research domains, used for image processing and management, the presence of a huge amount of ideas and applications was evident. However, a particular observation was made regarding the fact that researchers and developers still focus (significantly) more on abilities of computers instead of humans. In other words, more efforts are invested in the development of machine algorithms (improving old and inventing new ones) than on smart ways to utilise cognitive abilities of the user. This is especially interesting considering the problem of the "semantic gap", which refers to the difference between the complex semantic human cognition and low-level image visual information used by the machine based techniques.

Following this, the work performed within this thesis approached this problem from a different perspective. Considering the power of graphical information presentation, the idea of using information visualisation techniques and tools for image management was born.

A number of existing information visualisation approaches were investigated in order to find suitable methods for image data-set presentation and visual exploration. The investigated information visualisation approaches are reported in Chapter 2 and their utilisation within the image management (retrieval) domain in Chapter 3. As a result of this analysis, issues which can be addressed by information visualisation methods and tools have been identified. Within those, three topics have been distinguishes and tackled in this thesis. The first problem addressed is the visualisation of the image collection, as explained in Chapter 4, for providing similarity-based content overview. Chapter 5 deals with the issues of hierarchically structured image collections in terms of adequate content visualisation, exploration and access. The third part, stated in Chapter 6 proposes a user-oriented solution for broad range query specification, exploration and retrieval of images.

In the **Chapter 4** novel visualisation solution is proposed for providing an overview of image data-set indexed with low-level visual image features. The particular problem addressed here, is the generation of an intuitive and low-complexity graphical layout for interactive exploration of such image sets. Starting from standard visualisation approaches used in such cases, which aim to represent the image collection as a faithful 2-dimensional embedding of high-dimensional features, this work proposed an alternative approach for similarity representation. Instead of applying dimension reduction techniques on the complete set of high-dimensional image indices, a combination of partitional clustering, dimensionality reduction and ranked random layout methods has been proposed for generating a visual overview of image collections. As a result of clustering, the complete information of image relationships is divided into two levels: global (inter-cluster) and local (intra-cluster). They are mapped using the proposed layout in two different ways. Global similarity information is embedded in the visual display using Multidimensional Scaling. On the other hand, local similarity information is only partially conveyed using a ranked Gaussian layout of images for each cluster. In other words, on the local similarity level, the proposed solution preserves image-to-centroid similarities while "sacrificing" image-to-image similarity in order to lower the layout complexity.

The proposed visualisation provides an intuitive overview of the image collection and assists the user in identifying image groups (or topics) within the entire collection.

Clarity of the information conveyed by the proposed layout as well as the preservation of similarity relationships between images is evaluated by comparison with the equivalent, commonly used visualisation approaches. In other words, since the proposed layout is a form of extension of the standard MDS method on the entire image collection, it global and local properties are compared with equivalent MDS approaches.

As a result of comparative evaluation, it was concluded that regarding the global similarity preservation, the proposed approach is equivalent to "full MDS" layout of the image collection. Considering the fact that both methods use low level visual information as similarity metric, the logic of image placement when observed from the semantic perspective is the same. As an extension to this, the proposed approach provides visually distinctive groups and enables easier exploration, group-by-group. *Visual clusters* break the entire image collection into smaller groups, which implies better information management by the user and more efficient use of the available (limited) visualisation space.

Comparison of the proposed random layout approach and MDS-per-cluster again shows no significant difference in the layout. Omitting the inter-image similarity information did not degrade the informative utility of the collection overview.

From the perspective of layout generation algorithm, there are two advantages: (1) reduced complexity and (2) increased scalability. Observing the complexity of the MDS algorithm (discussed in Chapter 4) it is easy to conclude that the proposed layout, with it's random component is faster. Performing the MDS only for a small number of cluster centroids requires significantly less time then when MDS is applied on the entire image collection. This makes this layout suitable even for fast on-line applications.

Even if the processing time of the proposed algorithm (clustering and MDS between clusters) were comparable to the time required by "full MDS" layout generation it would be comparable only for the first-time layout generation. In case a new element needs to be added to the set, the processing requirements differ significantly. The proposed approach needs to find an appropriate cluster to place new item(s) and take care where exactly to place them. On the other hand, "full MDS" needs to calculate distances to the new item(s) in the high-dimensional space and generate a completely new set of positions for graphical layout.

An additional gain of the proposed layout is that the requirement of specifying an initial number of clusters for partitional algorithms, such as k-means, can be addressed visually by this approach. In case the specified number is greater than optimal, due to the inter-cluster MDS distance preservation they will be located close to each other.

Image overlaps per cluster is an important visualisation issue, especially considering the importance of information carried by the image itself. This issue, from the layout perspective was not addressed in this work. However, an alternative interactive solution was implemented by means of spatial distortion of the visual space. It enables the dynamic "clearing" of the overlaps.

Grasping the content of an image database, organised by hierarchical structuring methods, can be a problem if dealing with total absence of semantic information. This case is addressed in **Chapter 5** by combining the visualisation and access methods for fast exploration and retrieval of images from the data-set organised into hierarchies, based on low level information.

In terms of visualisation, the proposed solution provides global and local content overviews of the image data-set, by visualising the hierarchical structure (using tree view) and images of hierarchical node in focus. Displaying the images belonging to the node compensates for the absence of semantic information. The user is thus able to grasp the node content by seeing the related content summary. Exploration and content access are supported by several interaction methods such as random and sequential access, random exploration and so on. The novelty of the visual exploration approach lies in modelling user navigational behaviour by a "biased-random" content exploration and access methods, useful for identifying new search directions in case of unknown database. Extending the idea of random user behaviour, several factors are included for improving the efficiency of the initial exploration stage. Influencing the "random" model by knowledge of the hierarchy, "biased random" approach aims at helping the user to find first relevant image. Content access methods are realised through interactivity of all the visual elements, tree view and images in the node preview. These novel approaches are combined with sequential hierarchical access for extended user-system interaction flexibility. An additional content access method is selection of one image in the node preview, which takes the user to the last level of hierarchy, the leaf, where this image is located. The "jump-to-leaf" functionality has been identified as an efficient navigation method which efficiently utilises the properties of hierarchically arranged content, in terms of parent-child relationships.

Evaluation of the proposed application was conducted in two stages. The first stage was performed in order to investigate the usability of the novel user model of exploration. The first set of tests was performed in the professional working environment where professional users were testing the proposed solution. The collected results were types of interaction strategies used during tests, which showed that random exploration was highly utilised and appreciated novel functionality. The initial efficiency test was, on the other hand, performed by non-professional users for testing the system efficiency.

The second evaluation stage was performed on the improved interface prototype, by larger set of users. Both home and professional users tested the application and gave their comments by answering the questionnaire. User tests and evaluation results showed that the proposed solution is a useful tool for assisting professional and home users in finding their way through the initially unknown content repository.

As a result of the performed evaluations it can be concluded that modelling users "random" behaviour in the initial exploration stage is a useful functionality for initialising image retrieval.

However, there are still issues which are considered as important for improving the performance of this visual exploration and retrieval system. First, a visual preview of images can be discussed. The way the node content preview is generated ensures fast image display but does not "pay attention" to different child nodes of the preview node. This implies the possibility that not all the child nodes will be "previewed" by corresponding images. This might influence a user decision about the continuing steps. It could also be helpful if the content of different child nodes is distinguishable, for example by a different border colour. Such a solution was proposed in [86]. In addition to this, there is no memory in the process, so the application extracts a random set from the entire node without distinguishing images which had already been displayed. The "memory" of the extraction process can be easily added and it is expected to reduce the time needed to find a relevant image or images. Focusing on the visualisation of hierarchical structure it is possible that some additional information can be presented in order to help the user to explore the content of the nodes. One such option is displaying image examples of node content. This however opens a new research question. Would one image be an adequate content summary? If not, how should we balance the size of representative set and limited display space? And so on.

It is important to state that the "biased random" exploration currently implements only one out of several considered factors. Quantity of information is the only information influencing the algorithm decision on the node selection. The incorporation of other discussed factors is part of the future work.

When talking about scalability of this visual system, it refers to its ability to use results of any type of hierarchical clustering algorithm. It also includes semantic hierarchical clustering. Although the hierarchical visualisation application for browsing and searching was developed for the *rushes* repository it can be easily expanded to any other data-set. Currently, a new test-set is being pre-processed in order to generate a hierarchy of movie key-frames to be explored using the proposed tool.

The general conclusion obtained based on observations of users' behaviour during the evaluation stage was that the diverse methods of interaction were highly appreciated. Not once a user was stuck at the point where he/she did not have an idea how to proceed while trying to accomplish the task of retrieval. Compared to existing retrieval systems this conclusion is seen as a promising sign in terms of effectiveness.

Future work in terms of evaluation will consider approaches of performing more quantitative analysis in order to answer the question: how does such a combination of visualisation and several interaction methods influence the retrieval process. Since the conclusion is that in terms of user satisfaction (or effectiveness), the influence is in general positive, tests will address the efficiency of the solution. One option will be to compare the retrieval time between two test-cases:

1. Using only sequential access for finding a given image;

2. Using the proposed combination of interaction methods for finding the same

image

Additional results regarding the efficiency of the proposed solution will contribute to future application improvements.

When dealing with image databases, indexed automatically with semantic labels, there is an issue of unreliable and incomplete image information. On one hand, machine algorithms can generate semantic interpretations incorrectly and on the other, they cover only limited information about the image content. Exploration of such image data-sets by visual means could provide more information to the user, thus disguising the faults of labelling algorithms and improving overall retrieval and exploration performances.

The proposed and implemented framework, Multi Concept (MC) browser, described in **Chapter 6** has been designed for visual and interactive query specification and image retrieval. It integrates the results of machine processing algorithms within the visual exploration environment. The developed prototype covers two levels of visual image retrieval: first related to query specification based on semantic labels, and second focused on visual exploration and retrieval of images. On the first level of visual framework, query classes are used as basic retrieval elements for query specification. Interactive support for combining these classes, provides high scalability of the proposed framework in terms of queries it covers. In other words, by combining query classes, a user can specify large numbers of different queries.

As a comparison it is useful to mention that, developing a new classifier for each new user query includes several stages:

1. Manually annotating set of images which would serve as a training set for classifier;

2. Training the classifier with set of images annotated as relevant and non-relevant for specific concept;

3. Performing the classification of the entire image database.

Steps two and three are usually performed very quickly, but the first stage requires significant time and human effort to manually annotate images. The advantage of proposed solution is evident.

Displaying sets of query classes does not provide complete information about the content. In order to provide additional knowledge about the images in the data-set a concept relation display is developed. This display provides information about the concept relationships which can the help user infer the contexts of image content.

After a query is specified, a user is presented with a set of images corresponding to his/her query in order to retrieve relevant content. This second level utilises several visualisation techniques for supporting users retrieval. Images found as relevant to the query are displayed in the *content space*. Here, the used layout and interaction techniques allow the user to navigate and select relevant images. *Concept map* and *concept chart* are information visualisation tools used for providing additional support to the user. They provide an overview of the selected query classes, navigational assistance and support for sub-query specification.

Two sets of evaluation have been performed in order to test the proposed solution. The first evaluation stage performed an initial testing of the proposed idea and implementation. The second evaluation stage aimed at testing the performance and user satisfaction with the proposed solution. Performance was tested by comparison with two standard retrieval approaches: direct use of classification results (Support Vector Machines (SVM)) and relevance feedback (RF). The goal was to examine how does extending the level of user involvement by visual means influences the performance of image retrieval.

One segment of evaluation, in terms of usefulness, addressed the proposed relational concept display. The conclusion was that users could not really grasp the purpose of this visual representation in the given short time. Giving more time to the users to use the retrieval tool would provide clearer conclusion regarding the proposed visualisation element.

Analysing the results of the user questionnaire valuable users' feedback was obtained. Some users found image overlapping an important drawback of the MC browser visualisation. Due to the fact that most users did all 40 minutes of user-test without any break, interactive distortion was found eye-tiring. In order to properly test distortion as interactive retrieval tool shorter user sessions should be organised.

The fact that users were comfortable with the tool after the short trial period might indicate simplicity of the proposed solution. However, the intuitiveness of the visual cluster approach might be improved if each cluster had different coloured backgrounds. This colour can be chosen randomly or the dominant colour of the concept can be extracted and displayed as a background.

There are still elements which can be considered in order to improve theMC browsing solution both in terms of retrieval support and information visualisation. Future work will use the results of the performed evaluation in order to ensure improvement of the framework.

One of the outcomes of this thesis, is the strong belief that visual means can provide meaningful assistance for the image retrieval tasks. Built upon the strengths of automatic and semiautomatic image processing and organisation algorithms, adequate visualisation and interaction techniques, can provide a smart, interesting and efficient system layer adapted to the user requirements. However, judging upon the current state in these two research domains, significant adaptations from both sides need to be performed in order to create a successful coalition. From the users' perspective, visualisation solutions have to adapt their complexity in terms of graphical layout and interaction methods. Excessively complex interfaces produce satisfaction regarding the design and artistic qualities, but fail in terms of their usability.

In the end, the conclusion of this work is that appropriate integration of information visualisation and machine processing algorithms used in image retrieval systems has good potential for reducing the *semantic gap*. The paradigm of "grid based interfaces with the strong support of machine algorithms" is a practic used for too long in image retrieval systems.

Chapter 8

Publications

Patent applications:

1. European Patent Application No. 09425254.1 concerning a method for browsing multimedia archives, in the name of **Tijana Janjusevic**, Ebroul Izquierdo (Queen Mary, University of London), Sergio Benini, Riccardo Leonardi (Università degli Studi di Brescia).

Journal papers:

1. **T. Janjusevic**, S. Benini, E. Izquierdo, R. Leonardi, "Random assisted browsing of rushes archives", Journal of Multimedia (JMM), Vol. 5, No. 2 (2010), p.p 142-150, April 2010;

2. **T. Janjusevic** et al,"RUSHES - An Annotation and Retrieval Engine for Multimedia Semantic Unit", ACM Multimedia Tools and Applications, Journal on Content-Based Multimedia Indexing, Volume 48, Issue 1, page 23, 2010.

Conference papers:

1. **T. Janjusevic**, Q. Zhang, K. Chandramouli, E. Izquierdo, "Concept based Interactive Retrieval for Social Environment", accepted for ACM Multimedia 2010 Workshop - Social, Adaptive and Personalized Multimedia Interaction and Access (SAPMIA 2010), Florence, Italy, 29th October 2010;

2. **T. Janjusevic**, S. Benini, E. Izquierdo, R. Leonardi, "Random methods for fast exploration of raw video material", in Proceedings of 3rd International Workshop on Multimedia Data Mining and Management (MDMM 2009), 20th International Workshop on Database and Expert Systems Application (DEXA 2009), Linz, Austria, August-September 2009;

3. **T. Janjusevic**, E.Izquierdo, "Visualising the Query Space of the Image Collection", in Proceedings of 13th International Conference on Information Visualisation (IV'09), Barcelona, Spain, July 2009;

4. **T. Janjusevic**, E. Izquierdo, "Layout Methods for Intuitive Partitioning of Visualization Space", in Proceedings of 12th International Conference on Information Visualisation (IV'08),London, UK, July 2008.

Appendix A

A.1 K-means clustering

Clustering is an unsupervised process which classifies different items into separate groups (which are called "cluster" or "classes"). This means that, starting from a collection of items (usually represented as vectors of measures, or points in a proper dimensional space) an attempt is made to organise them in separate groups, so as that items in each group are similar. The large variety of existing techniques to represent data, to measure similarities, and to finally cluster items together, has given birth to a rich literature on clustering methods [52]. In general the clustering process can be described by the following steps:

1. The representation of data in a feature space;
2. The definition of a similarity measure between objects in the specific data domain;
3. The process of data clustering.

k-means is a partitional algorithm, popular due to its simple implementation and low computational complexity, which is O(m), where m is the number of elements to be clustered. It starts with a initial random partition, and it reassigns the elements to clusters on the basis of the similarity between the element and the cluster's centroid, until a convergence criterion is met (i.e., no more elements to assign are left, or the squared error ceases to decrease after a number of iterations). One drawback of such an approach is that the number of the k clusters must be a-priori set, or automatically discovered, often without enough knowledge of the data set.

The first step of the algorithm is defining k centroids, one for each cluster. The choice of centroids influences the results of the clustering. One of the potential solution is to place them as far as possible away from each other. The next step is to take each point belonging to a given data set and associate it with the nearest centroid. After placing all points new k centroids are re-calculated as barycenters of the clusters resulting from the previous step. After finding centroids the clustering of points is repeated for all elements. As a result of these iterations, k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimising an objective function, in this case a squared error function given by

$$J_{MSE} = \sum_{i=1}^{C} \sum_{x} |x - \mu_i|^2 \ , \tag{A.1}$$

where C represents the total number of clusters (*i.e.,* the number of partitions which the space has been subdivided in), and μ_i represents each cluster's centroid, which is given by:

$$\mu_i = \frac{1}{N_i} \sum_{x} x \ . \tag{A.2}$$

One of the issues with this method is that the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The results depend upon the initially selected centroid positions. In order to reduce this effect, the k-means algorithm can be re-runned several times for obtaining better results.

A.2 Multi Dimensional Scaling (MDS)

Multidimensional Scaling is a technique for finding a configuration of points in low dimensional space that preserves, as far as possible relationships between the multivariate data. Low dimensional distances are approximations of dissimilarities between data objects and the obtained configuration can be used for graphical presentation of data objects and their relationships. As a result of the MDS algorithm unrelated objects will be mapped to mutually distant points, whereas related ones will be grouped closer together.

Often an MDS is defined as optimisation problem of minimising the error of the distance between the points x_i in some target space SP.

$$argmin_{x_i \in SP} f(d_{ij}(X, \delta_{ij})) \tag{A.3}$$

As stated in [15] the benefit of using Multidimensional Scaling is in its generality. It is the consequence of the fact that MDS can be applied for any type of dissimilarity measure between two objects, and it is independent of the type and quantity of data objects.

The input to MDS is a square, symmetric matrix containing values that quantify relationships among a set of items. By convention, such matrices are either similarity or dissimilarity matrices. A matrix is a similarity matrix if larger numbers indicate greater similarity between items, and vice versa. A matrix is a dissimilarity matrix if larger numbers indicate less similarity. However, the similarity is not the only relationship among items that can be measured and analysed using MDS. Hence, many input matrices are neither similarities nor dissimilarities.

The optimisation of distances is mostly achieved using an appropriate optimisation function or *stress* function. The basic form of the algorithm, a Classical MDS (CMDS) employs Euclidean distance to model dissimilarity. The distance between two points i and j in an p-dimensional Euclidean space is given by the formula:

$$d_{ij} = \sqrt{\sum_{a=1}^{p} (x_{ia} - x_{ja})^2} \tag{A.4}$$

where x_{ia} specifies the position (co-ordinate) of point i on dimension a (he distance can also be defined according to the more general Minkowski model).

The generated distances and the function f which allows transformation the proximities into distances are estimated by minimising the *stress* function. The most commonly used stress function is the one proposed by Kruskal [111] refereed to as *stress 1*:

$$S = \sqrt{\frac{\sum_{i=j}^{n}\sum_{j>i}^{n} (\delta_{ij} - d_{ij})^2}{\sum_{i=j}^{n}\sum_{j>i}^{n} d_{ij}^2}} \tag{A.5}$$

The d_{ij} values here are optimal approximations of the transformed proximities p_{ij} to the distances d_{ij} in the geometrical representation. They are obtained by applying the suitable transformation to the observed proximities, or: $d_{ij} \approx \delta_{ij} = f(p_{ij})$ The d_{ij} values are often referred to as the disparities in contrast to the non-transformed proximities p_{ij} on the one hand, and the modelled distances d_{ij} in the geometrical space on the other hand. To the extent that the disparities have a close fit to the distances, the f is a function relating the observed proximities to the modelled distances.

Metric MDS is a form of MDS where the matrix of original dissimilarities (or similarities) should be pre-processed in order to have a metric values. The elements of resulting matrix D should have two properties:

- *non-degeneracy* or $d_{ii} = 0$ for all $i \in N$, where N is number of data objects;
- *triangular inequality* or $d_{ij}^2 + d_{ik}^2 \leq d_{jk}^2$ for all combinations of i,j,k.

Non-metric MDS is an algorithm which produces the order of the proximities and not the proximities or a linear transformation of the proximities. In non-metric scaling, a optimisation parameter is a monotonic transformation of the input data that minimises the stress function. The monotonic transformation is computed via *monotonic regression*, also known as *isotonic regression*.

Based on the current configuration of point, the targeted distances or disparities are found by monotonic regression and represent the distances that are monotonically related to the original dissimilarities.

The iterative process of non-metric MDS algorithms consists of four steps:

1. Select the dimensions and determines the initial configuration and the resulting distances in matrix D;

2. Use monotone regression that relates original distances/dissimilarities δ_{ij} with distances d_{ij}, which results in generation of disparities δ_{ij} that are monotonically related to the d_{ij} (non-metric phase);

3. Revise the spatial configuration to obtain new distances, which are more closely related to the disparities generated in step 2 (metric phase);

4. Calculate the stress function according to Equation A.5 to determine the goodness of fit between the distances and the disparities.

Depending on the optimisation criteria steps 2 and 3 are repeated.

There are two types of monotonic regression used for fitting disparities: *weak monotonicity* and *strong monotonicity*.

Weak monotonicity allows the fitting of unequal data by equal or unequal pseudo-distances:

$$d_{ij} < d_{kl} \Rightarrow \delta_{ij} \leq \delta_{kl} \tag{A.6}$$

Strong monotonicity allows the fitting of unequal data by equal or unequal pseudo-distances:

$$d_{ij} < d_{kl} \Rightarrow \delta_{ij} < \delta_{kl} \tag{A.7}$$

Various MDS approaches have been developed addressing the drawbacks of the basic MDS algorithm [15], [208], [129], [131] and [132]. Within this thesis the algorithm described in [15] was implemented for analysis thus explained in following section.

A.2.1 Incremental MDS

The incremental Multidimensional Scaling method proposed in [15] is developed as a solution to the MDS problem of a convergence criterion when numerical optimisation method is used. The developed approach uses cluster analysis techniques to assess the structural significance of groups of data objects. This approach reduces the size of the input data since it ignores the dissimilarities between similar objects.

The incremental MDS method is based upon a least squares metric MDS. It emulates a spring-based system where data objects are anchor points connected by a spring. The dissimilarity between two data objects determines the length of the relaxed spring. Each spring has energy associated with it, and the total loss function is a sum of energies over all springs.

$$E = \sum \frac{(d_{rs} - \hat{d}_{rs})^2}{\hat{d}_{rs}^2} \tag{A.8}$$

$$d_{rs} = \sqrt{\sum (x_{ri} - x_{si})^2} \tag{A.9}$$

is the Euclidean distance between points x_r and x_s in the p-dimensional layout, and $\hat{d}_{rs}$ is the dissimilarity between data objects.

The gradient vector g and the Hessian matrix H of the loss function at the point x_k can be established analytically as:

$$g_a(x_k) = \frac{\partial E}{\partial x_{ka}} == 2 \sum \frac{d_{kt} - \hat{d}_{kt}}{d_{kt}\hat{d}_{kt}^2}(x_{ka} - x_{ta}) \tag{A.10}$$

$$H_{ab}(x_k) = \frac{\partial^2 E}{\partial x_{ka} \partial x_{kb}} = \begin{cases} a \neq b \;\; \Rightarrow \;\; 2 \sum \frac{(x_{ka}-x_{ta})(x_{kb}-x_{tb})}{\hat{d}_{kt} d_{kt}^3}; \\ a = b \;\; \Rightarrow \;\; 2 \sum\limits_{t \neq k} \frac{1}{\hat{d}_{kt}^2} - 2 \sum\limits_{t \neq k} \frac{d_{kt}^2 - (x_{ka} - x_{ta})^2}{\hat{d}_{kt} d_{kt}^3}. \end{cases} \tag{A.11}$$

An optimisation method is then used (such as Newton-Raphson) to solve the vector equation $g(x_k) = 0$ in order to find minimum of the loss function in the point x_k.

There are two versions of the specified algorithm: one where all N points undergo minimisation (full scaling), and another where the minimization is over one point and all other points are fixed previously (single scaling). Full scaling consists of iterative approximation of positions for all N points, one by one, with respect to the remaining ones. The sequence in which points are considered is randomised to ensure faster convergence. This has a side effect of randomising the output of the algorithm.

In the case of single scaling for detection of convergence a set of previous locations of the point is kept and compared. If the current location has already been considered scaling terminates. In order to avoid local minima the author employs the following heuristic: once the position of a point has converged it is reflected over its closest neighbour, and the whole process is repeated until a lower energy state is reached.

In order to determine the order of data objects for which proposed iterative algorithm is applied Minimum Spanning Tree (MST) algorithm is used as described in details in [15].

Appendix B

The first part of this appendix gives a description of the hierarchical clustering algorithm used for the implementation and testing of the hierarchical browsing prototype presented in Chapter 5. After a general algorithm description, an example of inter-mediate '.xml' file is given which represent the output of the hierarchical clustering (for this application) and input of the proposed visual tool. In the second part of this appendix, a questionnaire used for the tool evaluation is given. This questionnaire was compiled for the first stage of professional user evaluation and later used for the second and final user evaluation performed with the RUSHES project [61].

B.1 Hierarchical clustering

Hierarchical clustering algorithms used for visualisation described in Chapter 5 was originally developed for generation of video previews based on the extracted video key-frames as described in [20]. Here, the general description is given in order to explain the algorithm approach. For more details, reader is refereed to [20].

B.1.1 Hierarchical clustering algorithm

Let $\phi_f(S_i, S_j)$ be the measure of similarity between two shots S_i and S_j, based on the chosen feature f extracted from shot key-frames. According to the agglomerative approach, at the beginning of the clustering process each of the N_s shots is assigned to a different cluster. Then, iteratively on each *level-i* (where $i \in I = \{N_s, N_s - 1, \ldots, 1\}$ indicates the number of the remaining clusters at the current stage), the *Hierarchical Agglomerative Clustering (HAC)* merges the two most similar clusters, from *level-N_s* (one shot per cluster) up to *level-*1 (all shots in the last remaining cluster).

Similarity $\Phi(C_h, C_k)$ between two clusters is computed according to the average-linkage approach, *i.e.*, by averaging the similarities between all the shots of the two clusters, that is:

$$\Phi(C_h, C_k) = \frac{1}{N_h N_k} \sum_{S_i \in C_h} \sum_{S_j \in C_k} \phi_f(S_i, S_j) \ , \tag{B.1}$$

where N_h (respectively N_k) is the number of shots belonging to cluster C_h (respectively C_k).

To prevent irrelevant shots from being wrongly incorporated into a cluster, we propose a local measure of visual coherence by using the *Leading-Cluster-Analysis (LCA)*. *LCA* provides a robust solution to the problem of determining how many natural clusters are present at different levels of content representation, and reduces the computational time for the generation of summaries.

Traditional flat algorithms cannot easily produce nested partitions, unless by iterating the clustering on different levels. The proposed *LCA* instead, provides a natural solution to this issue by automatically building a hierarchy of partitions where the number of layers and the number of clusters in each layer depend only on the video-content, and are not assigned *a-priori*. By adopting such a scheme, the video content is progressively condensed, from bottom to top, at decreasing levels of granularity.

If no criterion is used to stop the clustering of the N_s key-frames, a classic hierarchical approach produces a tree which is $(N_s - 1)$ levels deep. The main objective of the *Leading-Cluster-Analysis* is to generate a hierarchical preview $\mathcal{P} = \{L_1, L_2, \ldots, L_w\}$ organised on a reduced number of w levels, where $w << N_s$.

As shown in Figure B.1, each level L_i of the hierarchy contains the whole set of N_s key-frames organised in a number of visually similar clusters. Such organisation enables structured exploration: once the user identifies an interesting key-frame, he/she can interactively request more similar content from the same cluster, or refine his search by descending into the hierarchy, thus restricting the scope of his quest.

Regarding the low-level features, the proposed procedure for creating the hierarchical summary is very flexible since it may use any type of real valued low-level feature to compute pairwise similarity between shots. In order to improve the performance of the clustering with respect to traditionally employed low-level features, we propose to represent the shot visual-content in terms of *Tree-Structured Vector Quantization (TSVQ)* code-books.

For each extracted key-frame, a *TSVQ* code-book is designed so as to reconstruct each frame within a certain distortion limit. After having been sub-sampled in both directions at *QCIF* resolution, every key-frame is divided into non-overlapping blocks of $N \times N$ pixels. Block color components in the *LUV* color space are then used as training vectors to a *TSVQ* algorithm [67] which uses the *Generalized Lloyd*

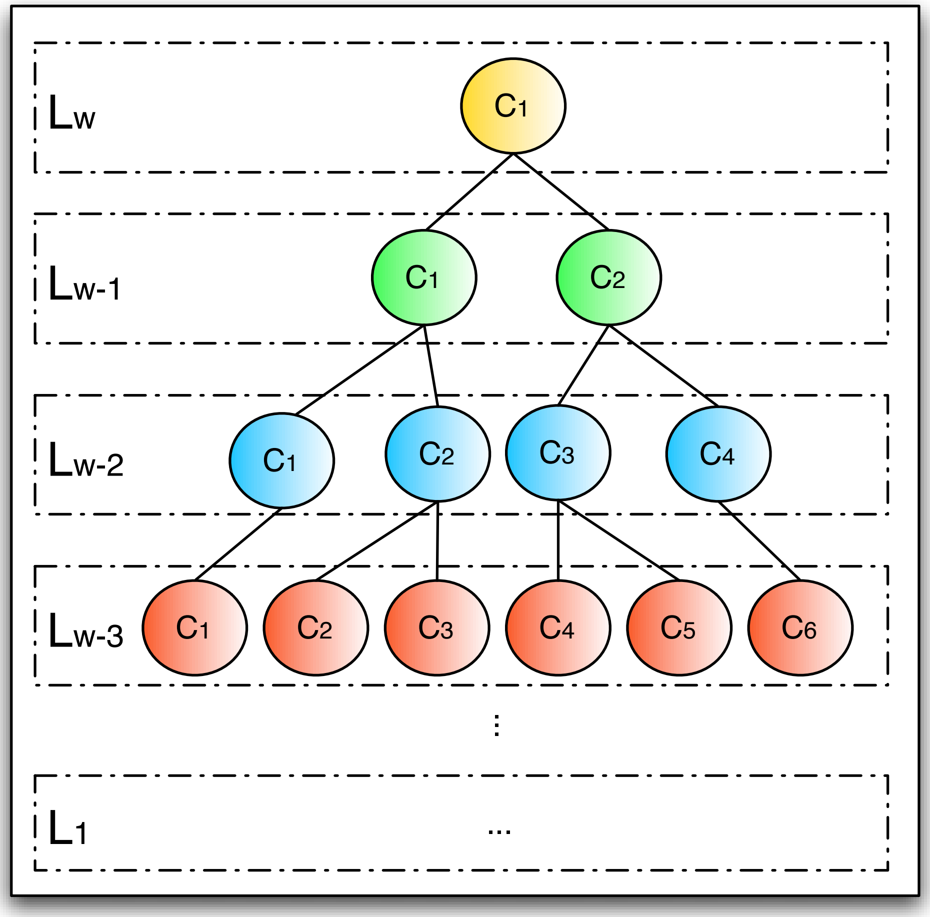

Figure B.1: Structure of the hierarchical preview.

Algorithm. Code-books of increasing size $2^n(n = 0, 1, 2, \dots)$ are then computed, until a pre-determined constraint on distortion is satisfied (or a maximum code-book size is reached). Then, an attempt is made to reduce the number of code-words in the interval $[2^{n-1}, 2^n]$ without exceeding the imposed distortion.

Finally the algorithm returns the code-words and the *TSVQ* code-book final dimension for each investigated shot. Note that the dimensions of each code-book could be different for each single shot. In fact the objective of this approach is to produce code-books with close distortion values, so that a code-book can represent the content of visually similar key-frames with a comparable distortion. This allows measuring the visual similarity between shot key-frames through a sound comparison between their related code-books.

The dissimilarity between two shots is then measured by crossing the code-books on the visual content of shot key-frames. Let S_i be a shot, and let K_j be a generic code-book; if V_i is the number of vectors of S_i, when a vector of S_i is quantized to a code-word of K_j, a quantization error occurs. This error may be measured by the average distortion $D_{K_j}(S_i)$:

$$D_{K_j}(S_i) = \frac{1}{V_i} \sum_{p=0}^{V_i - 1} \|s_{ip} - k_{jq}\|^2 \;, \tag{B.2}$$

where k_{jq} is the word of K_j with the smallest euclidean distance from s_{ip} ($q = \arg\min_z \|s_{ip} - k_{jz}\|^2$).

Furthermore, given two code-books K_i and K_j, the value:

$$\varphi_{i,j}(S_i) = |D_{K_i}(S_i) - D_{K_j}(S_i)| \tag{B.3}$$

can be interpreted as the dissimilarity between the two code-books, when applied to the same shot S_i.

The dissimilarity between two shots S_i and S_j is defined as a symmetric form of the measure used in [158], *i.e.*:

$$\phi_{vq}(S_i, S_j) = \varphi_{i,j}(S_i) + \varphi_{i,j}(S_j) \tag{B.4}$$

where $\varphi_{i,j}(S_j)$ is the dissimilarity between code-books K_i and K_j when are both applied to shot S_j. The smaller ϕ_{vq} is, the more similar the shots are. Note that the similarity is based on the cross-effect of the two code-books on the two considered shots. In fact, it may be possible that the majority of blocks of one shot (for example S_i), can be very well represented by a subset of code-words of code-book K_j representing the other shot. Therefore K_j can represent S_i with a small average distortion, even if the visual-content of the two shots is only partly similar. On the other hand, it is possible that code-book K_i does not lead to a small distortion when applied to S_j. So the cross-effect of code-books on the two shots is needed.

B.1.2 Output of the hierarchical algorithm used for visualisation

The hierarchical clustering algorithm generates a '.xml' file as an output further utilised by visual application described in Chapter 5. The information provided by the hierarchical clustering is the distribution of images through the hierarchy given in form of image identifications (ID's) in the appropriate tree branch (e.g. $< attributename = "frame"value = "Foot_301/kf_00171.jpg"/ >$). In order to reduce the size of such xml file, the image ID's are given only for the appropriate *leaf nodes* or in other words, the last levels of the tree containing these images. Content retrieval for nodes on the higher hierarchical level (non-leaves) are performed by the adequate query design implemented as part of the developed hierarchical browsing application explained in Chapter 5. For example, if user selects one tree node high in the hierarchy (e.g. level L_5), query parser examines the tree until it retrieves the image ID's stored in the leaves belonging to selected node/branch.

$< tree >$
$< declarations >$
$< attributeDeclname = "name" type = "String" / >$
$< attributeDeclname = "number" type = "Int" / >$
$< attributeDeclname = "id" type = "Int" / >$
$< attributeDeclname = "nChild" type = "Int" / >$
$< attributeDeclname = "nFrames" type = "Int" / >$
$< attributeDeclname = "frame" type = "String" / >$
$< attributeDeclname = "entropy" type = "Float" / >$
$< /declarations >$

$< branch >$
$< attributename = "name" value = "L16C0" / >$
$< attributename = "id" value = "17" / >$
$< attributename = "nFrames" value = "2" / >$
$< attributename = "entropy" value = "0.0551689" / >$
$< attributename = "frame" value = "Foot_3 01/kf_0 0171.jpg" / >$
$< attributename = "frame" value = "Foot_3 01/kf_0 1632.jpg" / >$
$< /branch >$
$< branch >$
$< attributename = "name" value = "L16C1" / >$
$< attributename = "id" value = "18" / >$
$< attributename = "nFrames" value = "2" / >$
$< attributename = "entropy" value = "32.5165" / >$
$< attributename = "frame" value = "Foot_3 01/kf_0 0429.jpg" / >$
$< attributename = "frame" value = "Foot_3 01/kf_0 1890.jpg" / >$
$< /branch >$
$< branch >$
$< attributename = "name" value = "L16C2" / >$
$< attributename = "id" value = "19" / >$
$< attributename = "nFrames" value = "1" / >$
$< attributename = "entropy" value = "0" / >$
$< attributename = "frame" value = "Foot_3 04/kf_0 0451.jpg" / >$
$< /branch >$
$< /branch >$

...

...

...

$< /tree >$

Since the visual application described in Chapter 5 uses this type of input (the given '.xml' file), it implies scalability of the approach and option of using any hierarchical clustering method which produces similar output information or can be adapted to do so.

B.2 RUSHES questionnaire

As part of the evaluation of hierarchical visualisation and interaction approaches, proposed in Chapter 5 and implemented as *Hierarchical browser* prototype, a questionnaire is compiled to retrieve users opinions and comments. Questions used within the questionnaire are given in Table B.1. This questionnaire was used in two evaluation stages during the prototype development. First stage was used for retrieving user's comments and understand the usability of the proposed solution in order to improve the application and it's functionalities. In the second evaluation stage users' answers to the questionnaire served as final evaluation of the proposed prototype.

No.	Statement
1.	It is easy to use the interface
2.	Interaction with the interface is comfortable
3.	It is easy to understand the functionality of the interface
4.	It is easy to understand the functionality of each button
5.	The tree in the upper window is helpful for understanding the database structure
6.	The interface is well organised
7.	The organisation of the visualisation display is clear
8.	The tree in the upper window is helpful for browsing and accessing content
9.	The key-frames displayed in the bottom window provide a good overview of the node content
10.	Random exploration is helpful for browsing when I don't know where to find the desired content
11.	The interaction method is intuitive
12.	The colour of the tree nodes helps me to understand my position in the database
13.	The interface is pleasant to use

Table B.1: Selected questions from user questionnaire.

Appendix C

C.1 Multi-feature based image retrieval

The original Multi-objective learning (MOL) [216] based retrieval consists of two main steps: pre-processing and multi-objective learning for retrieval.

Pre-processing steps are required before the MOL is performed in order to construct a suitable platform for MOL using the available information:

- Step 1: visual content decomposition,
- Step 2: low-level feature extraction,
- Step 3: training sample selection,
- Step 4: definition of the virtual centroid,
- Step 5: distances calculation,
- Step 6: distances normalisation,
- Step 7: construction of objective functions.

In Step 1, the visual content need to be decomposed into the basic elements that the proposed approach could handle. Usually, static images are decomposed into regions either by splitting them into blocks of regular sizes or segmenting them into regions of irregular shapes using some sophisticatedly designed algorithms. This is because user attention usually focuses on single objects rather than whole images. Regions are considered to be closer to object representations than whole images. This system uses block regions for their good performance and low computational cost.

In Step 2, a set of suitable low-level features are extracted from the basic elements of the visual content. These features are selected empirically for their capability of description and discrimination on different aspects of same visual content.

In step 3, a professional user is requested to select a group of training samples from which the proposed MOL mechanism can learn the optimal multi-feature merging metric. Naturally, the task of finding a single sample of element that can be regarded as an optimal representation for a concept is unachievable. To tackle this problem, we use a training sample group that usually consists of 10 to 20 samples selected by the professional user to approximate the optimal representation of the query concept.

Step 4 calculates a centroid for each considered visual feature by finding the training sample with the minimal sum of distances to all other training samples. Let l stand for the index of a feature in the feature set and L stand for the total number of visual features. Such a centroid $\bar{v_l}$ for a feature is denoted by the feature vector of the training sample. All the centroids across different feature spaces form a particular set of vectors $\bar{V} = \{\bar{v_1}, \bar{v_2}, ..., \bar{v_L}\}$. In general, $\bar{V}$ is referred to as the virtual centroid of the training group, since it does not represent a real training sample.

In Step 5, taking $\bar{V}$ as an anchor, for each feature vector extracted in Step 2 from one of the training samples in one feature space, a distance can be estimated.

In the following step, all these distances are normalised within each feature space to ensure the minimum comparability. The *Min-Max Normalisation* is utilised in this step to transform the distance values into the range [0, 1].

The last preliminary step constructs objective functions using the training sample distance values prepared in the previous steps. Such an objective function can be denoted as:

$$M^{(k)}(V^{(k)}, \bar{V}, A) = \sum_{l=1}^{L} \alpha_l m_l^{(k)}(v_l^{(k)}, \bar{v_l}) = \sum_{l=1}^{L} \alpha_l d_l^{(k)} \tag{C.1}$$

where $V^{(k)}$ is the set of vectors of the k^{th} training sample in all feature spaces. The letter K is used to represent the total number of training samples selected and k stands for the index. A is the collection of decision variables $alpha_l$, in other words the weighting factors, $m_l^{(k)}$ is the distance metric of the k^{th} training sample in the l^{th} feature space, and $d_l^{(k)}$ stands for the distance value between the k^{th} training sample and the centroid in the l^{th} feature space. The objective function set $M(\bar{V}, A)$ obtained from the preliminary steps is denoted as:

$$M(\bar{V}, A) = \left\{ \begin{array}{c} M^{(1)}(V^{(1)}, \bar{V}, A) \\ M^{(2)}(V^{(2)}, \bar{V}, A) \\ \vdots \\ M^{(K)}(V^{(K)}, \bar{V}, A) \end{array} \right\} \tag{C.2}$$

where $M^{(k)}$ is the merging metric function in the target multi-feature spaces of the k^{th} training sample. At this point, it is ready to perform the optimisation in the proposed MOL mechanism.

The reason why MOO strategy is necessarily required to solve the problems within the objective functions like in (1) is explained in our previous work [216], and is not repeated here due to space limitation. The MOL step is conducted based on coefficients $A = \{\alpha_l | l = 1, 2, \ldots, L\}$ subject to the constraint: $h(A) = \sum_{l \in [1,L]} \alpha_l = 1$. The target of optimisation is defined as minimisation of values of all the objective functions in (1).

However, by minimising $M(\bar{V}, A)$, the MOO algorithms usually do not generate a single solution, but rather a set of solutions, $\{A'_t | t = 1, 2, \ldots, T\}, t \in \Re$. They are referred to as *Pareto-optimal solutions*. Among these optimal solutions, the solution that produces the minimal value of the sum of distances in $M(\bar{V}, A')$ across all training samples is selected as the final optimal solution.

For a given query concept which represents a semantically meaningful object, an optimised multi-feauture metric $M(\bar{V}, A')$ is obtained for the multiple visual features. This combination metric is assumed to represent the symbolic visual pattern of the object within a multiple visual feature space. The multi-visual-feature similarities of each block regions are automatically calculated and the similarity of the most similar block in an image is considered as the similarity for this image.

C.2 Support Vector Machine

Support vector machines (SVMs) are a set of supervised learning methods used for classification and regression [202]. In simple words, given a set of training examples, each marked as "positive" or "negative", an SVM training algorithm builds a prediction model. This model is used to classify new example into one of the two categories. Generated SVM model creates a space where positive and negative examples are points in this space, divided by gap. New examples, which need to be classified are then mapped into that same space and model predicts to which category they belong to, based on the side of the gap they fall on.

The SVM algorithm was initially introduced in 1995 by Vladimir Vapnik [197] and later found application in various research domains (bioinformatics, text processing, image recognition, etc.).

In the first stage of using SVM or *training stage*, the SV algorith is given a training set of examples and related values. If a data D is input set of attribute vectors x_i

$$D = \{(\mathbf{x}_i, c_i) | \mathbf{x}_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^{n} \tag{C.3}$$

the value c_i determines the category of the data element. In other words, the values 1 and -1 of c_i imply if the data element x_i is a *positive* or *negative* example. We want to find the maximum-margin hyperplane that divides the points having $c_i = 1$ from those having $c_i = -1$. Any hyperplane can be written as the set of points $\mathbf{x}$ which satisfy

$$\mathbf{w} \cdot \mathbf{x} - b = 0, \tag{C.4}$$

The vector $\mathbf{w}$ is a vector perpendicular to the hyperplane. The parameter $\frac{b}{\|\mathbf{w}\|}$ determines the offset of the hyperplane from the origin along the normal vector $\mathbf{w}$.

We want to choose the $\mathbf{w}$ and b to maximize the margin, or distance between the parallel hyperplanes that are as far apart as possible while still separating the data.

After appropriate methods are applied in order to find the optimal margins, created model is used for data classification. The next data set given as an input to SVM will contain only attribute vectors without values. Values are then created by SVM algorithm.

Within this thesis a SVM algorithm implementation used is *SVMlight* [88]. As stated by the developer, SVMlight is an implementation of Vapnik's Support Vector Machine. More details about the algorithm can be found at [88].

C.3 Relevance Feedback framework

Relevance Feedback (RF) is regarded as an efficient method in Content Based Information Retrieval (CBIR) systems. It is particularly true considering the fact that they able to interpret the individuality of users and that are somewhat able to overcome the semantic gap between low-level visual features and high-level semantic concepts. By prompting the user for relevance feedback, the initial estimation of relevant documents could be improved to steer the results in the direction the user has in mind. This method employs the idea of exploiting the human computer interaction to refine high level queries for representations based on low-level features.

To simulate human visual perception, multiple low-level features are extracted from image content needs to be considered. The aim is to obtain information from different low-level visual cues at various levels of complexity and to jointly exploit that information to obtain higher levels of conceptual abstraction. Low-level descriptors are very useful to search for patterns of interest and similarities in image database. The proposed system [31], as shown in Figure C.1, consists of two main subsystems. The first subsystem runs offline and embraces two processing steps. The aim of this step is to extract the different low-level features from the image dataset. The extracted features are stored in the metadata repository. The metadata repository is then further indexed based on the image id's. The second subsystem involves online interaction with the user and comprises a number of processing steps. The second subsystem consists of two online search modules namely "visual search" and "RF system" which are discussed in detail in the following subsections. The reminder of this section will discuss the workflow of the framework.

The interaction is initialised by randomly presenting the user with equal distribution of the database. The user marks only the relevant images from the presented results. The first user interaction inputs are presented to the "visual search module". The visual search module implicitly generates a model for irrelevant model and performs the retrieval. The objective of this step is to infer and predict the user preferences. From the set of results presented from first iteration, the user selects both relevant and irrelevant images and the input is presented to "RF System". The aim of this step is to enhance the inference of the user preferences in order to improve the image retrieval. The user is then iteratively interacts with the system until the user has retrieved all relevant documents or satisfied with the retrieved results.

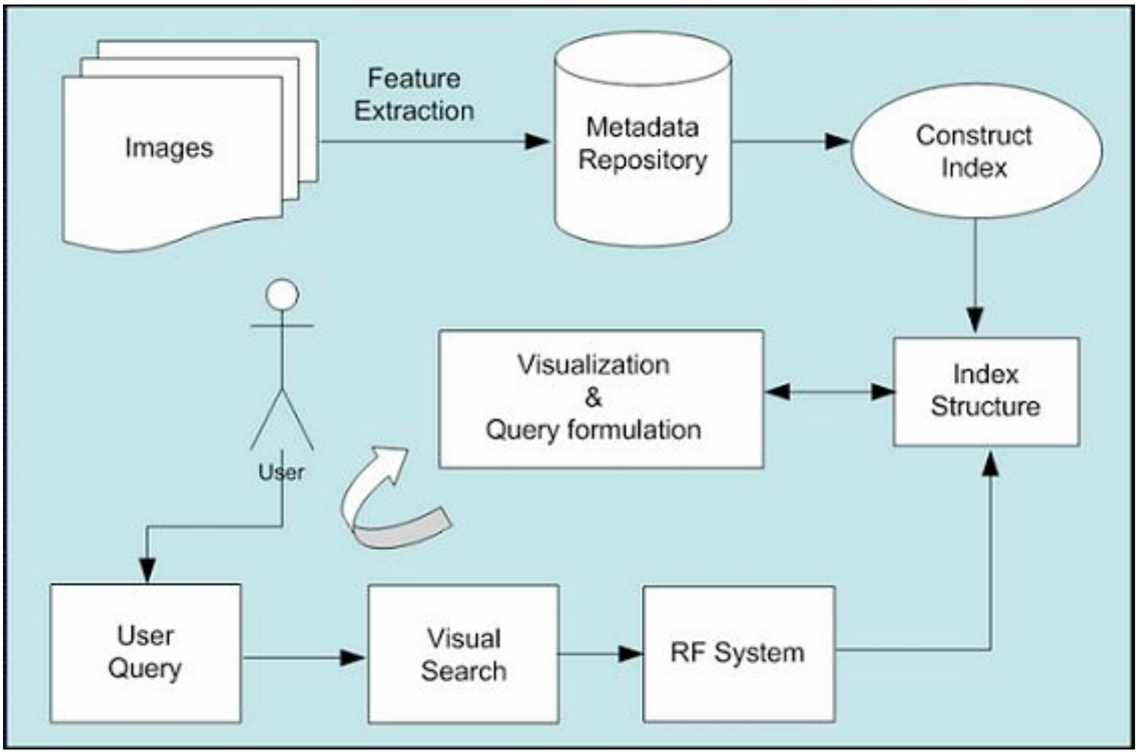

Figure C.1: RF Framework

The MPEG - 7 visual descriptors namely Colour Layout Descriptor (CLD) [121],

[120] and Edge Histogram Descriptor (EHD) are extracted for images in the following datasets. The CLD extracts colour histograms over 8 X 8 image layout. Its similarity measure is a weighted metric with nonlinearly quantised DCT coefficients. The EHD builds on histograms of edges in different directions and scales. Detected edges in a number of directions are used as localised input edge histogram of 80 bins. Its distance is a sum of distances over the original features, as well as global and semi-global histogram values generated by various grouping of local image parts.

The PSO model implemented is a combination of cognitive and social behaviour. The structure of the PSO is "fully connected" in which a change in a particle affects the velocity and position of other particles in the group as opposed to partial connectivity, where a change in a particle affects the limited number of neighbourhood in the group. Each dimension of the feature set is optimised with 50 particles. The size of the SOM network is pre-fixed with the maximum number of training samples to be used in the network. The stopping criteria threshold is set to 1.0. The value of the threshold indicated the closeness in solving the optimisation problem.

C.4 Questionnaire for user evaluation

C.4.1 Pre-evaluation questionnaire

P1 Gender:

P2 Age:

P3 Previous experience with image retrieval systems (Google images, yahoo image search, etc.):
0 - no previous experience 1 - very little experience 2 3 4 5 - extensive experience

P4 Familiarity with image retrieval tools and algorithms (image indexing, annotation, classification, relevance feedback, etc):
0 - no previous experience 1 - very little experience 2 3 4 5 - extensive experience

P5 Comment on expected image content based on concept relationships displayed:

C.4.2 Post-evaluation questionnaire

The purpose of the questionnaire is to evaluate the functionality of the interface. The questionnaire should be filled after having performed the specified user tasks.

Please, fill the questionnaire taking into account the meaning of each mark: 1 = "Strongly disagree", 2 = "Disagree", 3 = "Neither agree nor disagree", 4 ="Agree" and 5 = "Strongly agree"

If you may want to add some comment to the questions, use the table of the last section of the questionnaire.

Is there some important information about the content that is not shown and which one:

List three positive aspects of the tested tool:

-
-
-

List three negative aspects:

-
-
-

Would you add/remove/modify any functionality:

-
-
-

Comments:

-
-
-

		1	2	3	4	5
	General opinion					
1	In general, I am satisfied with how easy it is to use the interface					
2	I have found the interaction with the interface comfortable					
3	Interface provides interesting novel way to retrieve images					
	Comprehension of the tool					
4	It was easy to understand the functions of the first interface window for query selection					
5	Query formulation is easy since available terms for query are displayed					
6	It was easy to understand the purpose of concept relational display					
7	The interface is well organised					
8	The functionality of the interface is intuitive					
	Usability of the query window					
9	The organisation of information in the window display is clear					
10	The relations between the concepts is useful information to understand what kind of images are in the database					
11	Number of images per concept is useful information during retrieval process					
	Usability of the browsing window					
12	The image layout is easy to explore using distortion					
13	It is easy to find relevant images in the display					
14	The chart visualisation provides useful information about the distribution of images in the query concepts					
15	The chart visualisation is an effective way to perform sub-selection of the concepts					
16	The interface element in the upper right part of the window was useful for general orientation while browsing					
17	The interactive disappearing of images when selected is an interesting interaction method					
	Performance					
18	The interface is pleasant to use					
19	Finding relevant images can be successfully accomplished using this tool					

Bibliography

[1] J. Adcock, J. Pickens, M. Cooper, L. Anthony, F. Chen, and P. Qvarfordt. Fxpal interactive search experiments for TRECVID 2007. Proceedings of the TREC Video Retrieval Evaluation.

[2] C. Ahlberg and B. Shneiderman. The alphaslider: a compact and rapid selector. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '94*, pages 365–371, New York, NY, USA, 1994. ACM.

[3] C. Ahlberg and B. Shneiderman. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *Proceedings of ACM Conference on Human Factors in Computing Systems, CHI '94*, pages 313–317, New York, NY, USA, 1994. ACM.

[4] C. Ahlberg and B. Shneiderman. Visual information seeking using the filmfinder. In *CHI '94: Conference companion on Human factors in computing systems*, pages 433–434, New York, NY, USA, 1994. ACM.

[5] D. F. Andrews. Plots of high-dimensional data. *Biometrics*, 28(1):125–136, 1972.

[6] K. Andrews, W. Kienreich, V. Sabol, J. Becker, G. Droschl, F. Kappe, M. Granitzer, P. Auer, and K. Tochtermann. The infosky visual explorer: exploiting hierarchical structure and document similarities. *Information Visualization*, 1(3/4):166–181, 2002.

[7] P. Anick. Using terminological feedback for web search refinement: a log-based study. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 88–95, New York, NY, USA, 2003. ACM.

[8] D. A. Ankerst, M.and Keim and H.-P. Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *Visualization '96, Hot Topic Session, San Francisco, CA*, 1996.

[9] D. Archambault and D. Auber. Topolayout: Multilevel graph layout by topological features. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):305–317, 2007. Member-Munzner, T.

[10] L.F. et al. Ardeo. Requirement analysis and use-cases definition for professional content creators or providers and home-users, 2008. RUSHES Project, FP6-045189, Deliverable D5, WP1.

[11] D. Asimov. The grand tour: A tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143, 1985.

[12] BBC history timeline, 2009. *http://www.bbc.co.uk/history*.

[13] M. Balzer and O. Deussen. Voronoi treemaps. In *INFOVIS '05: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 7, Washington, DC, USA, 2005. IEEE Computer Society.

[14] T. Barecke, E. Kijak, A. Nurnberger, and M. Detyniecki. Videosom: A som-based interface for video browsing. *Lecture Notes in Computer Science, Image and Video Retrieval*, 4071:506–509, 2006.

[15] W. Basalaj. Incremental multidimensional scaling method for database visualization. In *In Proc. Visual Data Exploration and Analysis VI, SPIE*, pages 149–158, 1999.

[16] M. Bates. In *New Review of Information Behaviour Research, Fourth international Conference on Information Needs, Seeking and Use in Different Contexts*, pages 1–15, Lisbon, Portugal, 2002.

[17] B. B. Bederson. Photomesa: a zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of UIST '01*, pages 71–80, New York, NY, USA, 2001. ACM.

[18] B. B. Bederson, A. Clamage, M. P. Czerwinski, and G. G. Robertson. Datelens: A fisheye calendar interface for pdas. *ACM Transaction on Computer-Human Interaction*, 11(1):90–119, 2004.

[19] B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Transaction on Graphics*, 21(4):833–854, 2002.

[20] S. Benini, P. Migliorati, and R. Leonardi. Hierarchical structuring of video previews by leading-cluster-analysis. *Signal, Image and Video Processing*, 2010.

[21] R. Benmokhtar, E. Dumont, B. Mérialdo, and B. Huet. Eurecom in TrecVid 2006: high level features extractions and BBC rushes exploitation. In *TrecVid 2006, 10th International Workshop on Video Retrieval Evaluation, November 2006, Gaithersburg, USA*, Nov 2006.

[22] C. L. Borgman. Designing digital libraries for usability. *Digital library use: Social practice in design and evaluation*, pages 85–118, 2003.

[23] P. Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3), 2003.

[24] D. Borth, C. Schulze, A. Ulges, and T.M Breuel. Navidgator-Similarity Based Browsing for Image & Video Databases. In *Proceedings of 31st Annual German Conference on Artificial Intelligence, KI 2008*, volume II, pages 22–29, Kaiserslautern, Germany, September 2008.

[25] S. Brandt, J. Laaksonen, and E. Oja. Statistical shape features for content-based image retrieval. *Journal of Mathematical Imaging and Vision*, 17:187–198, 2002.

[26] P. Browne, S. Ruger, L. Q. Xu, and D. Heesch. ibase: Navigating digital library collections. *Lecture Notes in Computer Science, Image and Video Retrieval*, 4071:510–513, 2006.

[27] C.H. Busha and S.P. Harter. Research methods in librarianship: Techniques and interpretation, 1980. Academic Press, San Diego, CA.

[28] M. Campanella, R. Leonardi, and P. Migliorati. Interactive visualization of video content and associated description for semantic annotation. *Signal, Image and Video Processing*, August 2008. Springer.

[29] R. Capra, G. Marchionini, J. S. Oh, F. Stutzman, and Y. Zhang. Effects of structure and interaction style on distinct search tasks. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 442–451, New York, NY, USA, 2007. ACM.

[30] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *VIS '96: Proceedings of the 7th conference on Visualization '96*, pages 127–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.

[31] K. Chandramouli and E. Izquierdo. *Image Retrieval using Particle Swarm Optimization.* CRC Press, 2008.

[32] S.F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong. VideoQ: An automated content based video search system using visual cues. In *In Proceedings of ACM Multimedia*, pages 313–324, 1997.

[33] S.F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong. A fully automated content-based video search engine supporting spatiotemporal queries. *CirSysVideo*, 8(5):602–615, September 1998.

[34] S.F. Chang, W. Chen, and H. Sundaram. Semantic visual templates: linking visual features to semantics. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pages 531–535 vol.3, Oct 1998.

[35] C. Chen, G. Gagaudakis, and P. Rosin. Similarity-based image browsing. In *Proceedings of the 16th IFIP World Computer Congress, International Conference on Inteligent Information Processing*, pages 189–194, 2000.

[36] D.W.K Chen. The electrical engineering handbook, 2005. Academic Press.

[37] J.Y. Chen, C.A. Bouman, and J.C. Dalton. Hierarchical browsing and search of large image databases. *Image Processing, IEEE Transactions on*, 9(3):442–455, Mar 2000.

[38] H. Chernoff. $mathworld.wolfram.com/.../ChernoffFaces\_1000.gif$.

[39] H. Chernoff. $mapmaker.rutgers.edu/355/Chernoff\_face.gif$.

[40] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.

[41] M. G. Christel. Supporting video library exploratory search: when storyboards are not enough. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 447–456, New York, NY, USA, 2008. ACM.

[42] L. Cinque, S. Levialdi, A. Malizia, and K.A. Olsen. A multidimensional image browser. *Journal of Visual Languages and Computing*, 9(1):103 – 117, 1998.

[43] J. Clark, 2007. http://www.neoformix.com/2007/ATextExplorer.html.

[44] A. Cockburn, A. K. Karlson, and B. B. Bederson. A review of focus and context interfaces, 2006. HCIL Tech Report HCIL-2006-09.

[45] Canada Corel Corp., Ontario. Corel stock photo library, 1990. *www.corel.com*.

[46] J. M. Corridoni, A. Del Bimbo, and P. Pala. Image retrieval by color semantics. *Multimedia Syst.*, 7(3):175–183, 1999.

[47] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *Image Processing, IEEE Transactions on*, 9(1):20–37, Jan 2000.

[48] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.

[49] D. Deng, J. Zhang, and M. Purvis. Visualisation and comparison of image collections based on self-organised maps. In *ACSW Frontiers '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, pages 97–102, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[50] Digglabs. Bigspy, 2007. http://labs.digg.com/bigspy/.

[51] M. Dontcheva, M. Agrawala, and M. Cohen. Metadata visualization for image browsing, 2005. UIST 2005 Demonstration.

[52] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[53] Eclipse. Eclipse, 2010. http://www.eclipse.org/.

[54] EiTB. Euskal Irrati Telebista, 2009. http://www.eitb.com/.

[55] G. Ellis and A. Dix. An explorative analysis of user evaluation studies in information visualisation. In *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors*, pages 1–7, New York, NY, USA, 2006. ACM.

[56] N. Elmqvist and P. Tsigas. Causality visualization using animated growing polygons. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 189–196, Oct. 2003.

[57] J. Fan, A. K. Elmagarmid, X. Zhu, W. G. Aref, and L. Wu. Classview: hierarchical video shot classification, indexing, and accessing. *Multimedia, IEEE Transactions on*, 6(1):70–86, Feb. 2004.

[58] J. Fan, Y. Gao, H. Lu, and R. Jain. Mining multilevel image semantics via hierarchical classification. *Multimedia, IEEE Transactions on*, 10(2):167–187, Feb. 2008.

[59] S. E. Fienberg. Graphical methods in statistics. *The American Statistician*, 33(4):165–178, 1979.

[60] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the qbic system. *Computer*, 28(9):23–32, Sep 1995.

[61] RUSHES FP6-045189. Retrieval of multimedia semantic units for enhanced reusability. http://www.rushes-project.eu/.

[62] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266, 1987.

[63] M. Friendly. Milestones in the history of thematic cartography, statistical graphics, and data visualization, 2008.

[64] G. W. Furnas. Generalized fisheye views. In *Proceedings of ACM Conference on Human Factors in Computing Systems, CHI'86*, pages 16–23, 1986.

[65] G. W. Furnas and A. Buja. Prosection views: Dimensional inference through sections and projections. *Journal of Computational and Graphical Statistics*, 3:323–385, 1994.

[66] S. Card G. Robertson, J. Mackinlay. Cone trees: Animated 3d visualizations of hierarchical information. In *Proceedings of ACM SIGCHI'91*, pages 189–194, 1991.

[67] A. Gersho and R. M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[68] T.M. Green, W. Ribarsky, and B. Fisher. Visual analytics for complex concepts using a human cognition model. In *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on*, pages 91–98, Oct. 2008.

[69] K.A. Han and S.H. Myaeng. Image organization and retrieval with automatically constructed feature vectors. In *Proceedings of ACM SIGIR '96*, pages 157–165, New York, NY, USA, 1996. ACM.

[70] J. Harris. Word count, 2003. $http://wordcount.org/main.php$.

[71] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[72] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[73] X. He and P. Niyogi. Locality preserving projection. *Advances in Neural Information Processing Systems*, 16:153–159, 2004.

[74] M. A. Hearst. Tilebars: visualization of term distribution information in full text information access. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 59–66, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[75] D. Heesch. A survey of browsing models for content based image retrieval. *Multimedia Tools Appl.*, 40(2):261–284, 2008.

[76] D. Heesch and S. Ruger. Three interfaces for content-based access to image collections. In *CIVR04*, pages 491–499, 2004.

[77] G. Hinton and G. Roweis. Stochastic neighbor embedding. *Neural Information Processing Systems*, 15, 2002.

[78] D. F. Huynh. Timeline, 2005. http://www.simile-widgets.org/timeline/.

[79] D.F. Huynh, S.M. Drucker, P. Baudisch, and C. Wong. Time quilt: scaling up zoomable photo browsers for large, unstructured photo collections. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '05*, pages 1937–1940, New York, NY, USA, 2005. ACM.

[80] Infovis09, 2009. $http://vis.computer.org/VisWeek2009/vis/$.

[81] Infovis:Wiki, 2009. $http://www.infoviswiki.net/$.

[82] F. Idris and S. Panchanathan. Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation*, 8(2):146–166, June 1997.

[83] A. Inselberg. N-dimensional coordinates. In *IEEE Pattern Analysis and Machine Intelligence (PAMI), "Picture Data Description and Management"*, 1980.

[84] P. Irani and C. Ware. Diagramming information structures using 3d perceptual primitives. *ACM Trans. Comput.-Hum. Interact.*, 10(1):1–19, 2003.

[85] H. V. Jagadish. A retrieval technique for similar shapes. In *Proceedings of the ACM SIGMOD '91*, pages 208–217, New York, NY, USA, 1991. ACM.

[86] T. Janjusevic, S. Benini, E. Izquierdo, and R. Leonardi. Random methods for fast exploration of the raw video material. In *DEXA Workshops*, pages 226–230, 2009.

[87] T. Janjusevic and E. Izquierdo. Layout methods for intuitive partitioning of visualization space. In *Proceedings of 12th International Conference Information Visualisation,IV '08.*, pages 88–93, July 2008.

[88] T. Joachims. Svmlight, 2008. Cornell University, Department of Computer Science, $http://svmlight.joachims.org/$.

[89] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on Visualization, VIS '91*, pages 284–291, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[90] J. M. Jose, J. Furner, and D. J. Harper. Spatial querying for image retrieval: a user-oriented evaluation. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–240, New York, NY, USA, 1998. ACM.

[91] C.A. Julien, J. E. Leide, and F. Bouthillier. Controlled user evaluations of information visualization interfaces for text retrieval: Literature review and meta-analysis. *J. Am. Soc. Inf. Sci. Technol.*, 59(6):1012–1024, 2008.

[92] Keyword and search engines statistics. http://www.keyworddiscovery.com/.

[93] Mpeg-7 overview, 2004. $http://www.chiariglione.org/mpeg/standards/$.

[94] E. Kandogan. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *In Proceedings of the IEEE Information Visualization Symposium, Late Breaking Hot Topics*, pages 9–12, 2000.

[95] J. Karhunen and J. Joutsensalo. Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Netw.*, 8(4):549–562, 1995.

[96] S. Kaski, T. Honkela, Krista Lagus, and Teuvo Kohonen. Websom - self-organizing maps of document collections. *Neurocomputing*, 21(1-3):101 – 117, 1998.

[97] T. Kato, T. Kurita, N. Otsu, and K. Hirata. A sketch retrieval method for full color image database-query by visual example. In *Pattern Recognition,*

1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on, pages 530–533, Aug-3 Sep 1992.

[98] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[99] D. A. Keim, M. C. Hao, J. Ladisch, H. Meichun, and U. Dayal. Pixel bar charts: a new technique for visualizing large multi-attribute data sets without aggregation. In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, pages 113–120, 2001.

[100] D.A. Keim. Pixel-oriented visualization techniques for exploring very large databases. *Journal of Computational and Graphical Statistics*, 5:58–77, 1996.

[101] D.A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, 2000.

[102] D.A. Keim and H.-P. Kriegel. Visdb: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, 14:40–49, 1994.

[103] A. Keith. Visual exploration of large hierarchies with information pyramids. *Information Visualisation, International Conference on*, 0:793, 2002.

[104] L. S. Kennedy and S.-F Chang. A reranking approach for context-based concept fusion in video indexing and retrieval. In *Proc. of CIVR '07*, pages 333–340, 2007.

[105] A. Kobsa. An empirical comparison of three commercial information visualization systems. In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, pages 123–130, 2001.

[106] T. Kohonen. Exploration of very large databases by self-organizing maps. In *Proceedings of International Conference on Neural Networks*, volume 1, pages PL1–PL6 vol.1, Jun 1997.

[107] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, V. Paatero, and A. Saarela. Organization of a massive document collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3):574–585, May 2000.

[108] R. Kosara, C.G. Healey, V. Interrante, D.H. Laidlaw, and C. Ware. Thoughts on user studies: Why, how, and when. *IEEE Computer Graphics and Applications*, 23:2003, 2003.

[109] S. Krishnamachari and M. Abdel-Mottaleb. Image browsing using hierarchical clustering. In *Proceedings of IEEE International Symposium on Computers and Communications*, pages 301–307, 1999.

[110] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a non-metric hypothesis. *Psychometrika*, 29(1):1–27, March 1964.

[111] J.B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, 1964.

[112] A. Kuchinsky, C. Pering, M. L. Creech, D. Freeze, B. Serra, and J. Gwizdka. Fotofile: a consumer multimedia organization and retrieval system. In *ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '99*, pages 496–503, New York, NY, USA, 1999. ACM.

[113] R.L. Kullberg, W.J. Mitchell, and S.A. Benton. Dynamic timelines - visualizing historical information in three dimensions, 1995.

[114] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '95*, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[115] Y.K. Leung and M.D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput.-Hum. Interact.*, 1(2):126–160, 1994.

[116] X. Lin, D. Soergel, and G. Marchionini. A self-organizing semantic map for information retrieval. In *SIGIR '91: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 262–269, New York, NY, USA, 1991. ACM.

[117] X. Lin, D. Soergel, and G. Marchionini. A self-organizing semantic map for information retrieval. In *Proceedings of the ACM SIGIR '91*, pages 262–269, New York, NY, USA, 1991. ACM.

[118] H. Liu, X. Xie, X. Tang, Z.W. Li, and W.Y. Ma. Effective browsing of web image search results. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 84–90, New York, NY, USA, 2004. ACM.

[119] J. MacCuish, A. McPherson, J. E. Barros, and P.M Kelly. Interactive layout mechanisms for image database retrieval. In G.G. Grinstein and R.F Erbacher, editors, *Visual Data Exploration and Analysis III*, volume 2656-1, pages 104–115. SPIE, 1996.

[120] B. S. Manjunath, J.R. Ohm, V. V. Vinod, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

[121] B. S. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG - 7, Multimedia Content Description Interface.* New Cork, Wiley, 2003.

[122] G. Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, 2006.

[123] G. Marchionini and B. Shneiderman. Finding facts vs. browsing knowledge in hypertext systems. *Computer*, 21(1):70–80, 1988.

[124] A. Marcus, D. Comorski, and S. Sergeyev. Supporting the evolution of a software visualization tool through usability studies. *International Conference on Program Comprehension*, 0:307–316, 2005.

[125] T. Meiers, T. Sikora, and I. Keller. Hierarchical image database browsing environment with embedded relevance feedback. In *Proceedings of 2002 International Conference on Image Processing*, volume 2, pages II–593–II–596 vol.2, 2002.

[126] MIT, 2009. *http://web.mit.edu/16.399/www/course/notes/*.

[127] B. Moghaddam, Q. Tian, N. Lesh, C. Shen, and T.S. Huang. Visualization and user-modeling for browsing personal photo libraries. *International Journal of Computer Vision*, 56:2004, 2004.

[128] A. Morrison and M. Chalmers. Improving hybrid mds with pivot-based searching. *Information Visualization, IEEE Symposium on*, 0:11, 2003.

[129] A. Morrison and M. Chalmers. A pivot-based routine for improved parent-finding in hybrid mds. *Information Visualization*, 3(2):109–122, 2004.

[130] A. Morrison, G. Ross, and M. Chalmers. Combining and comparing clustering and layout algorithms, 2002.

[131] A. Morrison, G. Ross, and M. Chalmers. A hybrid layout algorithm for sub-quadratic multidimensional scaling. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 152, Washington, DC, USA, 2002. IEEE Computer Society.

[132] A. Morrison, G. Ross, and M. Chalmers. Fast multidimensional scaling through sampling, springs and interpolation. *Information Visualization*, 2(1):68–77, 2003.

[133] A. Mueen, R. Zainuddin, and Sapiyan. Miars: A medical image retrieval system. *Journal of Medical Systems*, 2009.

[134] T. Munzner. Drawing large graphs with h3viewer and site manager. In *GD '98: Proceedings of the 6th International Symposium on Graph Drawing*, pages 384–393, London, UK, 1998. Springer-Verlag.

[135] T. Munzner. Exploring large graphs in 3d hyperbolic space. *IEEE Comput. Graph. Appl.*, 18(4):18–23, 1998.

[136] T. Munzner and P. Burchard. Visualizing the structure of the world wide web in 3d hyperbolic space. In *VRML '95: Proceedings of the first symposium on Virtual reality modeling language*, pages 33–38, New York, NY, USA, 1995. ACM.

[137] G. P. Nguyen and M. Worring. Interactive access to large image collections using similarity-based visualization. *J. Vis. Lang. Comput.*, 19(2):203–224, 2008.

[138] G.P. Nguyen, M. Worring, and A.W.M. Smeulders. Interactive search by direct manipulation of dissimilarity space. *Multimedia, IEEE Transactions on*, 9(7):1404–1415, Nov. 2007.

[139] Q.V. Nguyen and M.L. Huang. Dualview: a focus + context technique for navigating large graphs. In *Computer Graphics, Imaging and Vision: New Trends, 2005. International Conference on*, pages 162–166, July 2005.

[140] N.F. Noy, R.W. Fergerson, and M.A. Musen. The knowledge model of protege 2000: Combining interoperability and flexibility. *Lecture Notes in Computer Science*, 1937:69–82, 2000.

[141] D. Petrelli. On the role of user-centred evaluation in the advancement of interactive information retrieval. *Inf. Process. Manage.*, 44(1):22–38, 2008.

[142] P. Picard, A. Pentl, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases, 1994.

[143] R.M. Pickett and G.G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Systems, Man, and Cybernetics, 1988. Proceedings of the 1988 IEEE International Conference on*, volume 1, pages 514–519, Aug 1988.

[144] E. Pietriga. http://www.w3.org/2001/11/isaviz/, 2004.

[145] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of Advanced Visual Interfaces, AVI '04*, pages 109–116, New York, NY, USA, 2004. ACM.

[146] C. Plaisant, J. Grosjean, and B. B. Bederson. Spacetree: supporting exploration in large node link tree, design evolution and empirical evaluation. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 57–64, 2002.

[147] Cambridge University Press. English dictionary, 2010.

[148] P. Pu and Z. Pecenovic. Dynamic overview techniques for image retrieval. In *In Data Visualization 2000: Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*, page pp.). Springer, 2000.

[149] G. Robertson, K. Cameron, M. Czerwinski, and D. Robbins. Polyarchy visualization: visualizing multiple intersecting hierarchies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '02*, pages 423–430, New York, NY, USA, 2002. ACM.

[150] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: animated 3d visualizations of hierarchical information. In *Proceedings of ACM Conference on Human Factors in Computing Systems, CHI'91*, pages 189–194, New York, NY, USA, 1991. ACM.

[151] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does organisation by similarity assist image browsing? In *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI '01*, pages 190–197, New York, NY, USA, 2001. ACM.

[152] S.T. Roweis and L.K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290:2323–2326, December 2000.

[153] Y. Rubner, L. J. Guibas, and C. Tomasi. The earth movers distance, multi-dimensional scaling, and color-based image retrieval. In *APRA Image Understanding Workshop*, pages 661–668, May 1997.

[154] Y Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, Sep 1998.

[155] RUSHES. D25: Report on usability and acceptability based on prototype evaluation, 2009. RUSHES Project, FP6-045189, Deliverable D25, WP1.

[156] P. Salembier and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface.* John Wiley & Sons, Inc., NY, USA, 2002.

[157] S. Santini and R. Jain. Integrated browsing and querying for image databases. *IEEE MultiMedia*, 7(3):26–39, 2000.

[158] C. Saraceno and R. Leonardi. Indexing audiovisual databases through joint audio and video processing. *International Journal of Imaging Systems and Technology*, 9(5):320–331, 1999.

[159] M. Sarkar and M. H. Brown. Graphical fisheye views of graphs. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '92*, pages 83–91, New York, NY, USA, 1992. ACM.

[160] M. Sarkar and M. H. Brown. Graphical fisheye views. *Commun. ACM*, 37(12):73–83, 1994.

[161] J. Sauro. Are the terms formative and summative helpful or harmful?, 2010. http://www.measuringusability.com/blog/formative-summative.php.

[162] S. Sav, G. J. F. Jones, H. Lee, N. E. O'Connor, and A. F. Smeaton. Interactive experiments in object-based retrieval. In *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin / Heidelberg, 2006.

[163] G. Schaefer. A next generation browsing environment for large image repositories. *Multimedia Tools and Applications*, 2009.

[164] G. Schaefer and S. Ruszala. Hierarchical image database navigation on a hue sphere. In *Proceedings of International Symposium on Visual Computing, ISVC06*, pages II: 814–823, 2006.

[165] B. Scholkopf, E. Smola, and K.R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[166] R. W. Schvaneveldt, editor. *Pathfinder associative networks: studies in knowledge organization*. Ablex Publishing Corp., Norwood, NJ, USA, 1990.

[167] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: A content-based image browser for the world wide web. *Content-Based Access of Image and Video Libraries, IEEE Workshop on*, 0:2, 1997.

[168] I. K. Sethi and I. Coman. Image retrieval using hierarchical self-organizing feature maps. *Pattern Recognition Letters*, 20(11-13):1337–1345, 1999.

[169] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors*, pages 1–7, New York, NY, USA, 2006. ACM.

[170] T. Sikora. The mpeg-7 visual standard for content description-an overview. *IEEE Transactions on circuits and systems for video technology*, 11(6):696–702, June 2001.

[171] M. Sintek. Ontoviz, 2006.

[172] A. W. M. Smeulders, J. C. van Gemert, B. Huurnink, D. C. Koelma, O. de Rooij, K. E. A. van de Sande, C. G. M. Snoek, C. J. Veenman, and M. Worring. Semantic video search. In *International Conference on Image Analysis and Processing*, Modena, Italy, September 2007.

[173] G. Smith, M. Czerwinski, B. Meyers, D. Robbins, G. Robertson, and D.S. Tan. Facetmap: A scalable search and browse visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):797–804, 2006.

[174] J. R. Smith and S.F. Chang. Visualseek: a fully automated content-based image query system. In *MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia*, pages 87–98, New York, NY, USA, 1996. ACM.

[175] C. G. M Snoek and et al. The MediaMill TRECVID 2008 semantic video search engine. In *Proceedings of the 6th TRECVID Workshop*, Gaithersburg, USA, November 2008.

[176] C. G. M. Snoek, M. Worring, D. C. Koelma, and A. W. M. Smeulders. A learned lexicon-driven paradigm for interactive video retrieval. *Multimedia, IEEE Transactions on*, 9(2):280–292, Feb. 2007.

[177] R. Spence. A framework for navigation. *Int. J. Hum.-Comput. Stud.*, 51(5):919–945, 1999.

[178] R. Spence and M. Apperley. *Data base navigation: an office environment for the professional.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[179] A. Spoerri. Infocrystal: A visual tool for information retrieval. *Proceedings of IEEE Conference on Visualization '93*, pages 150–157, Oct 1993.

[180] D. Stan. A new approach for exploration of image databases. In *In Proceedings of the Grace Hopper Celebration of Women in Computing 2002*, 2002.

[181] D. Stan and I. K. Sethi. eid: a system for exploration of image databases. *Information Processing and Management*, 39(3):335–361, 2003.

[182] D. Stone, C. Jarrett, M. Woodroffe, and S. Minocha. *User Interface Design and Evaluation, The Morgan Kaufmann Series in Interactive Technologies.* Morgan Kaufmann, March 2005.

[183] A. G. Sutcliffe, M. Ennis, and J. Hu. Evaluating the effectiveness of visual user interfaces for information retrieval. *International Journal of Human-Computer Studies*, 53(5):741–763, 2000.

[184] D. F. Swayne, D. Cook, and A. Buja. Xgobi: Interactive dynamic graphics in the x window system with a link to s, 1991.

[185] D. F. Swayne, D. Temple Lang, A. Buja, and D. Cook. Ggobi: evolving from xgobi into an extensible framework for interactive data visualization. *Computational Statistics and Data Analysis*, 43(4):423 – 444, 2003. Data Visualization.

[186] Trecvid. $http://www-nlpir.nist.gov/projects/trecvid/$.

[187] Trecvid 2008 evaluation. $http://www-nlpir.nist.gov/projects/tv2008/tv2008.html$.

[188] Treemap visualisation, 2010. $www.cs.umd.edu/.../million-treemap.gif$.

[189] Media Techtarget, 2010. $http://media.techtarget.com/digitalguide/images/$.

[190] J. B. Tenenbaum. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[191] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17:401Ð419, 1952.

[192] M. Tory and T. Moller. Evaluating visualizations: Do expert reviews work? *IEEE Comput. Graph. Appl.*, 25(5):8–11, 2005.

[193] J. G. Trafton, S. S. Kirschenbaum, T. L. Tsu, R. T. Miyamoto, J. A. Ballas, and P. D. Raymond. Turning pictures into numbers: extracting and generating information from complex visualizations. *Int. J. Hum.-Comput. Stud.*, 53(5):827–850, 2000.

[194] J. Urban, J. M. Jose, and C.J. van Rijsbergen. An adaptive approach towards content-based image retrieval. In *Proc. of the Third International Workshop on Content-Based Multimedia Indexing (CBMI'03)*, pages 119–126, Rennes, France, September 2003.

[195] J. J. van Wijk and R. van Liere. Hyperslice: visualization of scalar functions of many variables. In *VIS '93: Proceedings of the 4th conference on Visualization '93*, pages 119–125, Washington, DC, USA, 1993. IEEE Computer Society.

[196] J.J. van Wijk. The value of visualization. In *Visualization, 2005. VIS 05. IEEE*, pages 79–86, Oct. 2005.

[197] V.N Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[198] A. Veerasamy and N. J. Belkin. Evaluation of a tool for visualization of information retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 85–92, New York, NY, USA, 1996. ACM.

[199] R. C. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey, 2000.

[200] J. Vendrig, M. Worring, and A. W. M. Smeulders. Filter image browsing: Interactive image retrieval by using database overviews. *Multimedia Tools Appl.*, 15(1):83–103, 2001.

[201] Wikiviz, 2009. $http://www.wikiviz.org/wiki/$.

[202] Wikipedia, 2010. $http://en.wikipedia.org/wiki/Main\_Page$.

[203] J. Z. Wang and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:947–963, 2001.

[204] T.D. Wang and B. Parsia. Cropcircles: Topology sensitive visualization of owl class hierarchies. In *The Semantic Web - ISWC 2006*, pages 695–708. Springer Berlin / Heidelberg, 2003.

[205] M. Ward. Starglyphs visual example, 1994. *http : //davis.wpi.edu/ xmdv/*.

[206] C. Wharton, J. Rieman, C. Lewis, and P. Polson. The cognitive walkthrough method: a practitioner's guide. pages 105–140, 1994.

[207] R. W. White, B. Kules, S.M. Drucker, and M.C. Schraefel. Supporting exploratory search. *Communications of the ACM*, 49(4), April 2006.

[208] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization*, pages 57–64, Washington, DC, USA, 2004. IEEE Computer Society.

[209] J.A. Wise, J.J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Information Visualization, 1995. Proceedings.*, pages 51–58, Oct. 1995.

[210] M. Worring, C. G. M. Snoek, O. de Rooij, G.P. Nguyen, R. van Balen, and D.C. Koelma. MediaMill: Advanced browsing in news video archives. In H. Sundaram et al., editors, *Proceedings of the International Conference on Image and Video Retrieval, CIVR 2006, Tempe, Arizona, July 13-15, 2006*, volume 4071 of *LNCS*, pages 533–536, Heidelberg, Germany, July 2006. Springer-Verlag.

[211] C. C. Yang, H. Chen, and K. K. Hong. Visualization tools for self-organizing maps. In *DL '99: Proceedings of the fourth ACM conference on Digital libraries*, pages 258–259, New York, NY, USA, 1999. ACM.

[212] J. Yang, J Fan, D. Hubball, J. Gao, H. Luo, W. Ribarsky, and M. Ward. Semantic image browser: Bridging information visualization with automated intelligent image analysis. In *2006 IEEE Symposium On Visual Analytics Science And Technology*, pages 191–198, 31 2006-Nov. 2 2006.

[213] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. pages 401–408. ACM Press, 2003.

[214] E. Zavesky, S.F. Chang, and C.C. Yang. Visual islands: intuitive browsing of visual search results. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 617–626, New York, NY, USA, 2008. ACM.

[215] H. Zhang and D. Zhong. A scheme for visual feature based image indexing. In *Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Databases*, 1995.

[216] Q. Zhang and E. Izquierdo. Combining low-level features for semantic extraction in image retrieval. *Eurasip Journal on Advances in Signal Processing*, 2007, 2007.

[217] J. Zhou and A. Robles-Kelly. A quasi-random sampling approach to image retrieval. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

"It is a mistake to think of a picture as less sophisticated than mathematics ... the eye, the brain and human intuition are the best tools we have for finding patterns", Levine 1996.