INVESTIGATING IMAGE PROCESSING ALGORITHMS FOR PROVISION OF INFORMATION IN ROCK ART SITES USING MOBILE DEVICES

AYODEJI OLOJEDE

A DISSERTATION SUBMITTED IN PARTIAL
FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCE
UNIVERSITY OF CAPE TOWN

SUPERVISED BY

HUSSEIN SULEMAN

JULY 2016

Plagiarism Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for which is properly acknowledged, is my own.

Ayodeji Olojede

Date

Acknowledgement

Gratitude to the Almighty God for his grace and mercy over me and seeing me through the completion of this interesting and challenging journey. I also want to thank God for his guidance and protection over me during my stay in South Africa.

Many thanks to my parents for their financial, moral and spiritual support in seeing to the completion of my Master degree. You have both been a blessing to me. I love you so much. Also many thanks to my siblings for all the advice and words of encouragement. I also want to thank my friends both in Nigeria and in South Africa. To all my lab colleagues, you guys have been awesome. I love you all.

I also want to thank the Department of Archaeology at the University of Cape Town especially Professor John Parkington for facilitating access to the rock art digital collection on CD and for the recommendation to the rock art site in Clan William.

I specially want to thank my supervisor, Associate Professor Hussein Suleman for giving me the opportunity to work with him. I appreciate the encouragement and support you gave to me throughout my graduate study. I really appreciate all your effort. God bless you.

ABSTRACT

The term cultural heritage spaces incorporates places, objects and practices of cultural and historical significance. Examples include the Southern African rock art heritage sites. Rock art is an archaeological term used to describe man-made markings on stones. Studies have revealed that visitors to rock art sites usually do not understand the meaning of the rock art artefacts they are looking at due to a lack of descriptive information necessary to frame the artefact in the proper cultural and historical context. Instead, rock art sites offer humans as tour guides. One problem observed with human tour guides is that they often do not provide enough information about the artefact and they also do not answer questions to the satisfaction of most visitors. Also, human guides are a limited and expensive resources and do not always provide a personalized experience for each visitor.

Therefore, in this research, an alternate interpretation mechanism that gives visitors a personalized interaction with rock art artefacts is proposed. We introduce Heritage Vision, a mobile guide application that enables visitors to take a picture of a rock art artefact of interest and automatically presents information about the artefact to the visitor. This is done via a content based image retrieval system with the aid of image processing. We investigate 3 image processing algorithms for digital recognition of rock art images on mobile devices. The ubiquitous nature and recent technological advances has made mobile devices the preferred medium. Image processing algorithms such as Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF) and Oriented Fast and Rotational Brief (ORB) have been incorporated in a mobile guide prototype and their performance has been evaluated. Performance evaluation has revealed that the ORB algorithm has a better and acceptable performance over the SIFT and SURF algorithms. A user experiment was performed to evaluate the usability of the application prototype using SUMI (software usability measurement inventory) and the result obtained shows a SUMI global scale (perceived quality of use) score of above average, suggesting that such a solution is feasible.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Cultural Heritage are physical artefacts and attributes from the past that are of great importance to a country and need to be passed on to current and future generations. Cultural heritage space are places of cultural or historical significance. Examples include the Southern African **rock art heritage sites**. Rock art is a term used to describe human-made engravings and paintings on stones. In rock art literature, rock art paintings and engravings are referred to as pictographs and petroglyphs respectively. Commonly associated with a nation's rich archaeological past, rock art represent an important asset for tourism and they provide a body of information on several different dimensions **[1]** beyond their aesthetic expression.

Studies have shown that visitors to **rock art heritage sites** usually do not understand the meaning of what they are looking at during physical visits. Ideally, such sites should include descriptive information about the artefacts in the rock art heritage sites but this is usually not the case. Instead, rock art heritage sites usually offer human tour guides for visitors. Recent observations **[191]** have revealed that these tour guides usually do not provide enough information about the exhibits and do not answer questions to the satisfaction of most visitors. Also, it has been observed that, often, the tour guide speaks too fast or communicates in an accent that is difficult for most visitors to comprehend. An alternate interpretation media could provide visitors with a more **personalized** interaction with the rock art artefact using **mobile devices**. Interpretation media refers to different means employed by museums for the comprehension and interpretation of museum exhibits by the public **[2]**.

The recent trends in cultural heritage contexts has seen the adoption of **mobile** multimedia guides for personalized presentation and interpretation of exhibits. In this approach, the cultural heritage site provides mobile handheld devices that can be used by visitors during visits. A recent and preferred approach is the use of the visitor’s own technology (mobile device), that they are comfortable with, to navigate heritage spaces. This is a preferred approach because it saves the heritage site the cost of providing mobile devices to visitors and it will be easier for a visitor to make use of a device he/she is familiar with.

Leveraging on the latter approach, we present Heritage Vision, an **image based mobile guide** application for providing information about artefacts in rock art heritage sites. The application serves the purpose of providing visitors with personalized interaction with rock art artefacts by using the application on their own handheld mobile device. This guide recognises artefacts in a rock art heritage site based on images taken by the visitor with their mobile device and instantaneously retrieves information about the rock art artefact of interest. Recognition of rock art and retrieval of information is done via a content based image retrieval system. In the Content Based Image Retrieval system, we have adopted certain image pre-processing techniques along with SIFT (scale invariant feature transform), SURF (speeded up robust features) and ORB (oriented fast and rotational brief) image processing algorithms for recognition of rock art artefact. These algorithms have been integrated in a prototype and their performance has been evaluated.

1.1 Problem Statement

Digital recognition of objects remains an open challenge in computer vision due to numerous reasons such as changes in viewpoint, illumination and deformations. In rock art literature, digital recognition of rock art is a major problem. The issue is that there is no single object recognition approach that generalizes to all rock art images. Researchers have argued that the complex structure of rock art, such as cluttered background and jagged edges, defies most image processing algorithms. The quality of the photographs taken by the visitor also plays an important role in object recognition. Mobile devices still tend to come with poor camera lenses that are only capable of producing poor quality and noisy photographs, especially in a poorly illuminated rock art site where artefacts are hidden in caves. Also, camera phones are not primarily designed for taking high quality pictures, hence they are more difficult to hold steady, which in turn increases the likelihood of shakes that will prevent the camera from setting the focus correctly. This is a challenge as this will produce blurred photographs that will have a negative effect on the object recognition process. Also, rock art heritage sites are usually located in remote regions with little or no network support. This means that the traditional client and server approach of processing may not work. This suggests that all processing will be done on the visitor's mobile device, however, the object recognition process is computationally expensive, which will be a problem for visitors who have mobile devices with low processing capacity.

We aim to overcome these problems by using a standalone image matching application in a real life rock art heritage site scenario. Most of the related work reviewed in this thesis has made use of the client-server approach, where the mobile device is only used for image capture and processing is done on a powerful remote server. In this work, we deployed the mobile guide and the processing entirely on a mobile device and the performance was evaluated.

1.2 Research Questions

RQ1. How do the SIFT (scale invariant feature transform), SURF (speeded-up robust features) and ORB (oriented fast and rotational brief) algorithms compare in terms of accuracy and speed of similarity search for rock art images on mobile devices?

RQ2. Can an image based mobile guide be used to provide information about artefacts in a rock art heritage site?

In order to answer research question 2, user satisfaction with the developed mobile application prototype was evaluated.

1.3 Research Scope

The focus was on recognition and matching of pictographs using mobile devices with the aid of different image processing algorithms. It is however not feasible to investigate the use of all possible algorithms, thus a selection of 3 image processing algorithms was investigated based on general performance as seen in literature. The methods used in completion of this research were:

- Literature Survey
- Formulation of research questions
- Gathering of research data
- Design
- Evaluation

1.4 Ethical Clearance

This research involves user evaluation and, as such, the appropriate ethical clearance was obtained from the Science Faculty ethics committee and the Department of Student Affairs. As the experiment was conducted offsite, the appropriate permission was obtained from the owners of the building used for user experiments. All software and tools used in development of the mobile application prototype are freely available under an open source licence.

1.5 Thesis Outline

- **Chapter 1** is the introduction to what this study is all about, the problem statement, the motivation and the research questions.
- **Chapter 2** is the background chapter that gives information about the relevant topics and all works related to this research. Topics such as cultural heritage spaces, image retrieval, content based image retrieval systems, mobile guides and the use of mobile guides in cultural heritage spaces are discussed.
- **Chapter 3** describes the digital recognition of rock art and all the problems that are associated with object recognition of rock art images. All related works are also discussed.
- **Chapter 4** describes the design of the system employed in the research. The design approach and the methods employed in designing the prototype are discussed. The experiment design and procedures are also discussed in this chapter.
- **Chapter 5** describes the experimental results to compare the performance of the image processing algorithms in terms of accuracy and speed. This chapter also presents and analyses the experimental results from the usability testing of the mobile guide using SUMI (software usability measurement inventory).
- **Chapter 6** highlights concluding remarks and recommendations for future work.

Chapter 2

BACKGROUND

2.0 Introduction

The appreciation of cultural heritage artefacts, such as the Southern African rock art paintings, can be enhanced when the relevant information about the cultural heritage object is provided to the site visitor. In the past decades, cultural heritage content delivery has been only achieved through traditional audio-guides **[3]**. These feature a very primitive form of interaction, mostly based on a keypad, where the user selects a number corresponding to the exhibit he/she wants to be informed about **[4]**. Later research in this field have shown that these limitations can be overcome by the adoption of mobile multimedia application guides. Such applications have made use of content based image retrieval systems with the aid of digital image processing algorithms like SIFT and SURF **[5]** to deliver rich media content about cultural heritage artefacts to the user on mobile devices. The current advances in mobile technology, such as the recent generation of mobile devices, gives, for the first time, the chance to exploit high processing capability, available on very light and portable devices, that a few years back was only available on personal computers. This section examines some topic of interest to this research such as the Southern African Rock Art, image retrieval, content based image retrieval system, mobile guides and adoption of mobile guides in cultural heritage context.

2.1 Cultural Heritage Spaces: Rock Art Sites

Cultural heritage spaces refers to public places of historical or cultural significance. The main focus will be archaeological site such as rock art sites in the Southern African region. Rock art is an archaeological term used to describe man-made markings on stone surfaces **[1]**. Rock art provide a rich body of information on several different dimensions, beyond their value as an aesthetic expression **[1]**. A Recent survey has shown that the earliest expression of abstract thinking appeared as far back as 77,000 years ago in Southern Africa **[4] [6]**. Given this long history, rock art is one of the most valuable sources of history that has persisted to the present time.

The Southern African rock art are archaeological artefacts that depict the daily lives of the Khoisan people who are the indigenous people of Southern Africa who lived in the region thousands of years before colonisation **[7]**. The Khoisan people are mainly hunter-gatherers of the Southern African region. Studies have shown that the Khoisan Rock art forms an integral part of the Khoisan culture and they hold a deep religious and spiritual significance to the Khoisan people **[8].** They are hence considered and viewed as representations of the sensations felt by the Khoisan people, usually in a state know as trance **[8]**. There are tens of thousands of works of rock art in Southern Africa. Some contains geometric shapes while others contain human and animal figures, usually scenes illustrating human figures engaging in hunting exercises, gathering food or dancing. Figures also include therianthropes that are considered half human and half animal **[9]**.

2.1.1 Types of Rock Art

According to Whitley **[10]**, rock art sites consist of paintings and engravings that are referred to as pictograph and petroglyph, respectively, in rock art literature.

2.1.1.1 Petroglyph

The word petroglyph comes from the Greek words *Petros,* which means "stone", and *glyphein,* which means "to carve" **[11]**. Petroglyphs (also called rock engravings) are art works created by removing part of a rock surface by carving, incising, abrading and pecking, which can produce a visible identification on the rock **[12]**. We can also simply refer to it as the scratching away of a weathered rock surface to reveal an unweathered material of a different colour. Petroglyphs are found worldwide and are commonly associated with prehistoric people.

2.1.1.2 Pictograph

Pictographs are paintings on rock surfaces created by applying a combination of different natural substances that have different colours to the rock. These are commonly referred to as pigments. A pigment can be a composition of ferric oxide and ochre, white from silica, china clay, gypsum, charcoal and specularite **[8]**. Other known pigments include blood from hunted or sacrificed animals, and mineral materials such as chalk, limonite **[13]** (an amorphous iron oxide used as an ore of iron and as a pigment for thousands of years) or hematite **[14]** (an iron oxide and a rock forming mineral found in sedimentary, metamorphic and igneous rocks). Pictographs tend to be less durable than petroglyphs because they are merely a surface coating, except for the ones found in rock shelters, caves and areas with dry climate, which protects them from natural conditions.

2.1.2 Rock art object classification

The Southern African rock art can be divided into geometric objects and representational objects.

2.1.2.1 Geometric objects

According to Lewis-Williams **[8]**, geometric objects are abstract forms such as shapes, lines, circles, zig-zags and grids. These abstract forms are deep rooted in the sacred rituals of the Khoisan, where shamans dance persistently till they enter a state of consciousness while women clap and sing special songs intensely **[8]**. This is what is known as the trance state; these geometric forms are observed in the early stages of trance **[15]**.

2.1.2.2 Representational objects

Representational objects consist of humans, animals, therianthropes, the trance dance and equipment. These objects actually depicts real life scenes. The animal objects represent images of some Southern African wildlife such as elands and elephants. The human objects are composed of paintings and drawings of both male and female human figures. The therianthropes **[9]** classification contains drawings and paintings of half human and half animal beings. The trance dance classification contains images of humans engaged in trance while they are surrounded by human figures singing and clapping. The equipment objects contain images of bows, arrows, quivers, bags and digging sticks.

2.1.3 Characteristics of rock art

2.1.3.1 Perspective and View points

The most common view in rock art is the side view. For example, animals such as eland and elephants are painted in a side view and are usually painted in several positions such as standing, leaping, head turned or running **[8]**. There are also human paintings that exhibit mainly side views and are painted in several positions. They are also painted in positions such as standing, walking, running, hunting and dancing. The most notable difference is that sometimes humans are painted in a front view and then the only distinguishable factor is the shape of the painting. Women figures are usually identified with a particular shape. In a side view, women are identified with big buttocks. There are also paintings that identify the state of trance where women are painted clapping and dancing while they surround another human figure in the center.

2.1.3.2 Deterioration

It has been observed that rock art in the Southern African region is gradually degenerating, especially those that are exposed to humans and natural forces. Literature has shown that human activities account for more rock art deterioration than any other factors **[16]**. Human interest in rock art appears to result in its destruction **[16]**. The authors stressed that human activities such as scratching on or surrounding of painted areas, repainting and addition of detail, water splashing and smoke from ill-considered fire places have all taken their toll in the destruction of rock art paintings. Climatic conditions surrounding rock art paintings are also known to be a great cause of rock art decline **[16]**. With the exception of rock art found in caves, others are constantly exposed to climatic forces such as rain, movement of water to and from the rock art surfaces, direct sunlight and wind, which have constantly caused irreplaceable damage to the rock art paintings. Movement of water to and from rock art surfaces has a negative effect on the paintings by gradually removing pigments, thereby causing a slow fading away of the paintings **[16]**.

2.2 Image Retrieval

Advances in digital technology has led to growth in digital media collections, which can be a combination of still images or videos. Nowadays, storage devices come in hundreds of terabytes in storage capacity, which allows for storing or archiving of millions of digital objects such as images, thereby making it harder to search and retrieve images of interest from such collections **[17]**. This therefore necessitates effective and efficient search capabilities. The first attempt towards image retrieval was text-based search. This method requires the manual annotation of images by making use of text that best describe each image in the collection **[18]**. But manual annotation of images have been shown to have some potential flaws. This is a manual process (user-based) and manually annotating images could become cumbersome especially when the image database is large. Also, in user-based annotation, keywords used usually fail to properly describe the image. The above reasons have motivated research into the use of content-based image retrieval (CBIR) in the early 1990s. In CBIR, images are indexed according to their own visual content, such as colour, texture, shape or any other feature or a combination of visual features **[19] [20]**.

This section presents literature on image retrieval that was reviewed during the course of this research. There is a vast amount of literature available in this research area; hence this section is not aimed at giving a complete description of the research in CBIR. Instead, a selection of important papers that are relevant to this study are presented.

2.2.1 Content Based Image Retrieval

In image retrieval systems, similarity search underlies much of what we associate with human intelligence, including reasoning, classification and prediction **[21]**. The text based approach provide two main disadvantages. One is about human labour; how many hours are needed to manually annotate every image? The other is about the accuracy; how can we be sure that the annotation is correct? It depends on human perception **[22]**. As opposed to text based approaches of image retrieval, CBIR systems have attracted attention in recent years due to the growth of multimedia information storage in various fields of life. As illustrated in Fig 2.1, the goal of CBIR systems is to retrieve content related images from databases by employing suitable image processing and matching algorithms.

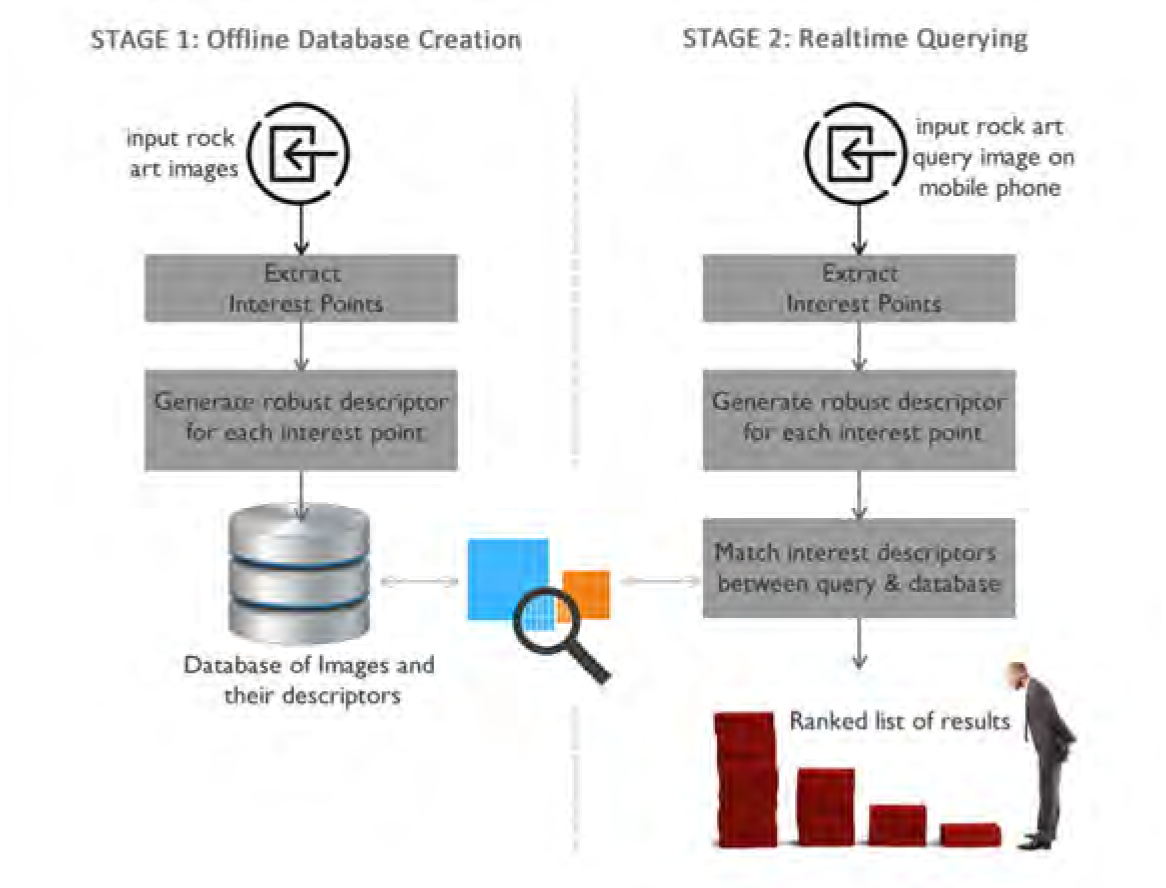


Fig 2.1 Content based image retrieval illustration

The word "Content-based" means that the search will analyse the actual contents of the image rather than the metadata associated with the image, which was the norm in text based image retrieval. The difference between a text-based and a content-based retrieval system is that the human interaction is an essential part of the former **[23]**. The term "content" is used to refer to those features, such as colours, shapes and textures that can be derived from the image itself. As mentioned earlier, the term "content-based image retrieval" seems to have originated in the 1990s **[24]** when it was used to describe the experiments of automatic retrieval of images based on some image features, such as colours and shapes **[25]**. CBIR has been used to describe the retrieval of images based on their visual features from a large database of training images, using techniques borrowed from fields such as computer vision, pattern recognition, statistics and signal processing **[26]**. Mostly, CBIR systems try to reproduce human discernment of image similarity **[27]**. In CBIR systems, extracted image descriptors are used to index the image database. In most CBIR systems, the aim is to find images that are visually similar to the query image from the database of training images.

Visual similarity search in CBIR systems are computationally expensive and also influenced by common image characteristics such as noise, changes in view or scale changes of the imaged content. To mitigate the sensitivity to different image characteristics, some image pre-processing is carried out. First, discriminant key points that serve as identifying feature vectors are extracted from each image in the repository. Second, the images are indexed on these precomputed feature vectors. Third, at query time, the feature vectors extracted from the query example are compared with these indices of the images in the database. This abstraction, while serving the purpose of adding robustness to the above-mentioned variations in imaged content, can potentially introduce a disparity (commonly referred to as gap) between the expected result and the computed result. Smeulders et al **[28]** gives a very comprehensive overview of gaps in CBIR. It describes common problems such as the semantic gap and the sensory gap associated with CBIR systems. In their article **[28]**, Smeulders et al. describe sensory gap as the difference between a real world scene and the derived computational interpretation. This description can be observed by the user either via displays and sensors or can be used by computers in the learning process. Sensory gap is caused by either the parameters of the scene (e.g., clutter, occlusion, illumination) or the parameters of the sensors (e.g., viewpoint, perceptual spectra) **[28]**. In Computer Vision, the data is recorded by cameras that perform similarly to the human vision system. Therefore, the sensory gap is narrow and can be attenuated by training the models on multiple images, representing various interpretations of an object **[28]**. As illustrated in **Fig 2.2**, Semantic gap on the other hand can be defined as the difference in meaning between constructs formed within different representation systems **[29]**.

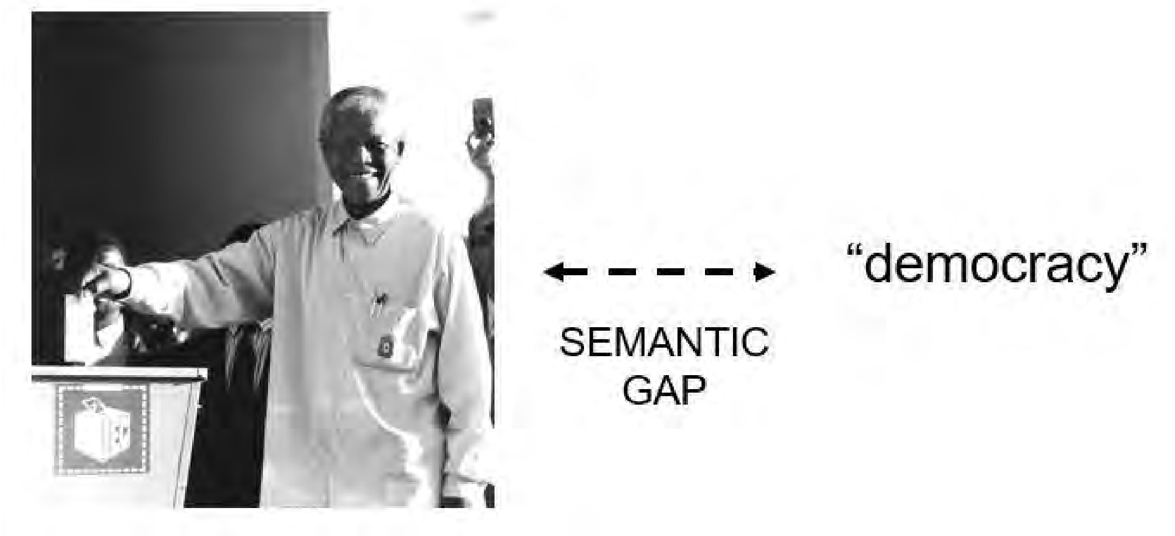


Fig 2.2 Illustration of Semantic disparity [30]

A Semantic gap arises when the user seeks high-level semantic concepts using a content-based retrieval or classification system **[31]**; however, the system perceives and processes images based on low-level visual features (e.g., colour, texture, and shape). Because multiple kinds of low-level visual features can contribute in a high-level semantic concept, the provided results usually suffer from confusion and misclassification.

In CBIR systems, feature descriptors are generated from query images and matched against feature descriptors of training images in the dictionary or database **[32]**. While they agree that extracted features are natural and objective, Chandrika and Jawahar **[33]** believed that there is a big gap between the high-level concepts and the low-level features (**see Fig 2.2**). Due to the semantic gap, CBIR systems that extract and compare primitive image features have no understanding of the image semantics and thus cannot meet the user's needs **[34]**. Much research has been carried out in this domain but bridging the semantic gap is still a active research area in CBIR **[35]**. In this section, literature on some key aspects of content-based image retrieval are discussed.

2.2.2 Image Features

An image feature is an interesting part of the image and is used to represent a valuable piece of information for image processing. Feature is the representation of any distinguishable characteristic of an image. Features can be applied in image processing and computer applications. In image processing, features automatically extracted from images are usually low level image features.

2.2.2.1 Low level image features

The most common image low level features used in the literature are: colour, texture and shape (spatial layout) **[37]**.

2.2.2.1.1 Colour

Colour feature is the most frequently used feature in image retrieval and indexing. Colour features are easy to obtain, often directly from the pixel intensities. One of the most common colour features is colour histograms **[38]**. Each bin in a colour histogram represents a range of colours and the value of the bin counts the number of image pixels that fall in the colour range. In colour histograms, images are transformed to the hue, saturation and value colour space (see Fig 2.3), and a 64-bin colour histogram is generated by uniformly quantizing H, S, and V components into 16, 2, and 2 regions, respectively **[39]**.

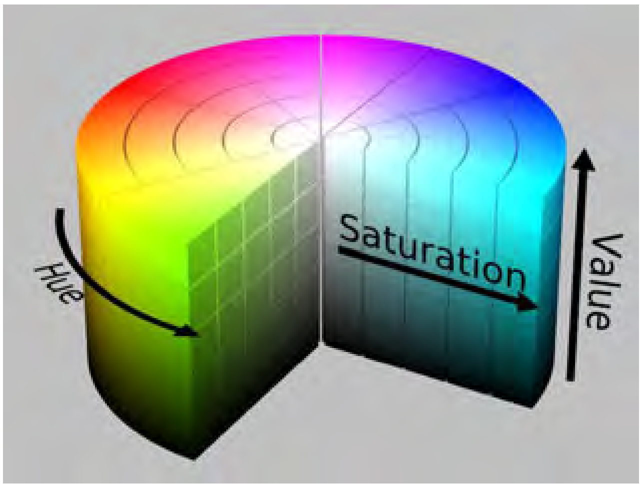


Fig 2.3 The HSV colour space

Colour histograms hence describe the overall colour content of the image. Colour histograms are easily computed and are invariant to image transformations but in order to combat problems related to sensitivity to scale and the lack of dimensional and structural information, various region-based histogram approaches have been proposed **[40]**. RGB is another colour space that is frequently used. RGB stands for Red, Green, and Blue. It is a well-known colour space and is it commonly used for visualization. See Fig 2.4 for a diagrammatic representation.

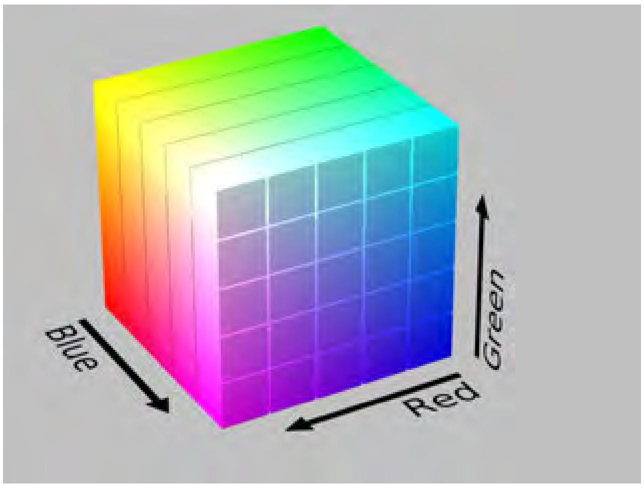


Fig 2.4 The RGB gamut arranged in a cube.

The RGB colour space is only rarely used for indexing and querying as it does not correspond well to human colour perception. Colour spaces such as HSV (Hue, Saturation, and Value) or the CIE Lab **[41]** and Luv **[42]** spaces are much better with respect to human perception and are more frequently used. This means that differences in the colour space are similar to the differences between colours that humans perceive.

2.2.2.1.2 Texture

Texture is the main feature utilized in image processing and computer vision to characterize the surface and structure of a given object or region **[43]**. There is no general definition of image texture but texture can be explained as an entity that consist of related pixels since pixels make up an image. Rosenfeld and Kak **[44]** describe texture as patterns that consist of sub-patterns. These sub-patterns exhibit characteristics such as colour, slope, brightness and size. Sub-pattern properties give rise to the perceived characteristics such as regularity, uniformity, coarseness, roughness, granularity, lightness, directionality, density, smoothness, randomness and frequency of the texture as a whole **[45]**. There are four methods of characterizing texture, which are the statistical, structural, model based and transform methods.

2.2.2.1.2.1 Statistical Method

In statistical method, non-deterministic properties are used to represent texture indirectly. The relationships between grey levels of an image are determined by these properties. Statistical methods given by pairs of pixels have been proven to have higher discrimination rates than structural and transform-based methods **[46]**.Accordingly, differences in second order moments result into spontaneous discrimination of textures in grey-level images. A deliberate cognitive effort is required in different third-order moments and equal second order-moments **[48]**. For texture analysis, co-occurrence matrix **[49]** is used to derive second order statistical features. In classification of texture, the co-occurrence matrix approach has been shown to do better than a tranform-based technique **[50]**

2.2.2.1.2.2 Structural Method

In this method, defined micro-texture and macro-texture are used to represent texture **[45] [49]**. Structural approach is more ideal for synthesis rather than analysis; however, one of the main advantage of this method is that, it gives a good description of the image. Because of the variability and lack of distinction between micro and macro texture structures, the descriptions can be ill defined for natural textures.

2.2.2.1.2.3 Transform Method

Transform method describe texture characteristics such as frequency or size that are closely related to the interpretation of the co-ordinate system of an image in a space. Examples of transform method include Fourier **[51]**, Gabor **[52]** and Wavelet transforms **[53]**. Fourier transform lack spatial localisation which result to their poor performance in practice. The usefulness of Gabor filters is limited in practice due to lack of single filter resolution to localise a spatial structure in natural textures. Wavelet transform represents textures at the most suitable scale due to flexibility in spatial resolution. This allows for a wide range of selection choices for wavelets deemed appropriate for texture analysis in a specific application. The main disadvantage with wavelet transform is that it is not translation-invariant **[54]**.

2.2.2.1.2.4 Model Based

In model based method, stochastic and fractal image models are employed to interpret an image texture [55]. The primary problem with stochastic model is in practice, the computational complexity arising in the estimation of it parameters. On the other hand, fractal model can be used for texture discrimination and analysis [56]; however, fractal models are not good at describing local image structures because it lacks orientation selectivity.

2.2.2.1.3 Shape

Shapes are geometric forms that help humans to identify real-world objects **[58]**. In shapes, the outline of the object is used to distinguish an object from its surroundings **[57]**. In CBIR systems based on shape features, images are segmented into regions and objects, which are then analysed for feature extraction. Geometric transformation invariant features are extracted from the segmented objects, which are then used for similarity search. Shapes can be broadly represented based on two categories **[59]**, namely region based and boundary or contour based. In the region based category, features are extracted from the entire region. It utilizes the entire shape region with its internal characteristics to describe shape i.e. the pixels contained in that region **[60]**. It also specifies the object's "body" within the closed boundary. Such a shape is represented with moment invariants, or a collection of primitives such as rectangles, disks, quadrics, etc., deformable templates, skeletons, or simply a set of points **[61]**. On the other hand, the boundary or contour based category calculates shape features only from the boundary of the shape **[60]**.

Boundary representation describes the closed curve surrounding the shape. As illustrated in Fig 2.5, the curve can be specified in numerous ways, e.g., by chain codes, polygons, circular arcs, splines, or boundary Fourier descriptors **[61]**:

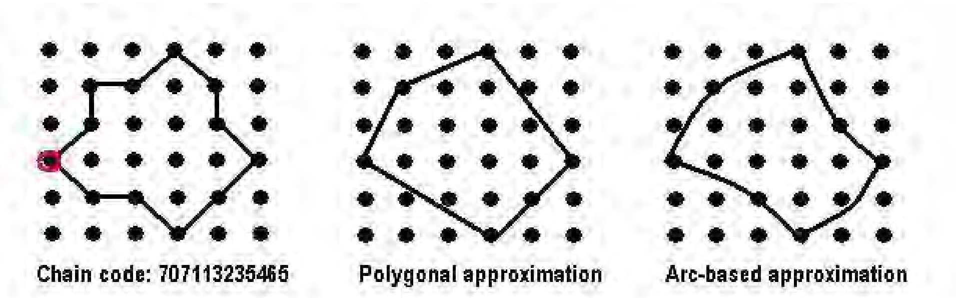


Fig 2.5 Boundary based representation of shape. [61]

The boundary features, such as an ordered polygonal approximation, formulates the user's query as a sketch **[61]**. Generally, the sketch is deformed to adjust to the shape of target models, and the amount of deformation, for example, the energy for an elastic deformation model. Such a shape descriptor (deformable template) represents shape variability by allowable transformations of a template **[61]**. However, in restricted environment, retrieval by shape provide a very good basis for segmentation **[62]**. Other descriptors include turning angle functions, algebraic moments, deformable templates, and Fourier coefficients **[63]**. Several studies have shown that shape is a powerful feature that has been extensively exploited in the past for constructing effective CBIR systems.

2.2.2.2 High level image features

In CBIR systems, studies have shown that low-level image features usually does not describe the high level human semantics **[66]**. With this in mind, it is clear that the performance of CBIR still fails to meet users' expectations. Higher-level features are intended to represent semantically meaningful concepts in the image, which are of more direct interest to humans **[67]**, for example, the activities taking place in an image. To automatically derive semantically meaningful features or concepts from images is still a challenge, however, there are various attempts and suggestions in the literature as described in a survey by Liu et al **[36]**, where a number of approaches are identified. Such approaches include:

1. To use object ontologies,
2. To use machine learning methods,
3. To use relevance feedback,
4. To use semantic templates for mapping high-level concepts to low-level visual features.

2.2.2.2.1 Object Ontology

An object ontology is used to allow for querying with semantically meaningful concepts without requiring manual annotation of images **[68]**. The user describes objects using intermediate-level descriptors such as intensity, position and size.

2.2.2.2.2 Machine Learning

Tsai, C-F and Hung **[70]** demonstrated that machine learning can be used to annotate images in an automatic fashion. High-level sematic features can be derived using tools such as supervised and unsupervised learning technique **[36]**. Predicting the value of an outcome measure based on a set of input measures is the main goal of supervised learning **[69]**. In machine learning, techniques such as Bayesian classifier **[71]** and support vector machine **[70]** are used to learn high level concepts from low level image features.

Support vector machine is a method that can be used to learn multiple concepts in image retrieval. A good example is the work by R. Shi et al. **[72]**, where a support vector machine was employed for annotating images. Bayesian classification is another widely used method. High level semantic concepts can also be extracted from low-level image features using Binary Bayesian classifiers **[73][74]**.

Neural networks is another technique used for concept learning. In order to learn high level semantics from low-level image features, Town and Sinclair **[75]** used 11 different concepts and then fed a large amount of training data into the neural network classifiers. One major problem with machine learning algorithms such as neural networks is that, they are computationally intensive and require lots of training data **[76].**

Semantic features can also be derived from images using a supervised learning technique called decision tree. In decision tree induction methods such as CART, ID3 and C4.5, input attribute space is recursively partitioned into non-overlapping spaces **[69]**. Following the paths from the root of the tree to the leaves, a set of decision rules can be obtained. C4.5 **[77]** is designed based on set of images that are of relevance to the query. Database images are classified into relevant and irrelevant classes using these images. Another useful decision tree method is the CART. It helps to formulate decision rules that are critical for mapping textual description to global colour distribution in a given image **[78]**. Decision trees can also be used in expert systems **[69]** for decision making but there is still an underlying problem of lack of modularity **[79]** when used in learning high level concept for image retrieval.

2.2.2.2.3 Relevance feedback

Relevance feedback is a query modification technique originally intended for text-based retrieval systems **[83]**. In the 90's, relevance feedback was introduced in content based image retrieval as an attempt to bridge the semantic gap. Relevance feedback captures the users' exact needs through iterative feedback and query refinement technique. Relevance feedback was developed with the goal of keeping the user in the retrieval loop. It has been shown that by continuous learning through interaction with users, relevance feedback has provided a performance boost in Content Based Image Retrieval Systems [84].The process of relevance feedback (see Fig 2.6) is performed in the following steps:

1. Initial user query,
2. The system provides a query result,
3. The user decides whether the returned results are relevant or irrelevant to the query,
4. Learning of user's feedback with machine learning technique. Then go back to (2).

From the above, step 2, 3 and 4 will be repeated until the user is satisfied with the results. In step 3, re-weighting **[36]** can be used to gradually adjust the low-level features till the user need is met.

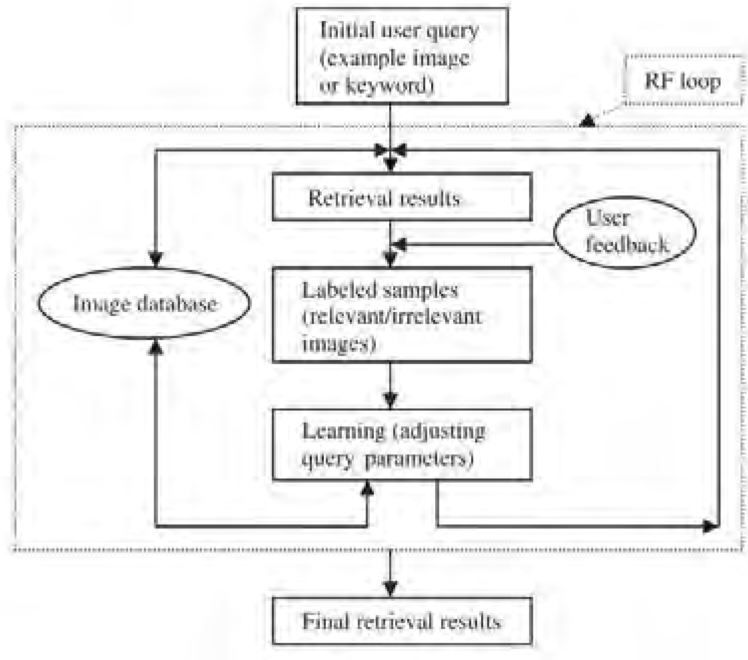


Fig 2.6 CBIR with Relevance feedback. [36]

In contrast to the conventional approach of relevance feedback, a generalized non-linear relevance feedback approach for image retrieval has been proposed **[85]**. The authors claim that this approach is a better alternative to conventional relevance feedback. In this approach, an online learning technique is used to estimate the similarity measurement based on a recurrent superlative estimation of a non-linear relation of known functional components instead of adjusting the degree of importance of each descriptors **[36]**.

2.2.2.2.4 Semantic template

Semantic templates are a promising technique but rarely used for semantic based image retrieval **[36]**. An example of how it can be used is that a user makes a search using query-by example, but where keywords/concepts are added to the search. After some iteration with relevance feedback, the system provides images that are judged as relevant to the query **[67]**. A potential problem with this approach is that much manual work is needed for the system to learn new concepts/keywords. Some sample work on semantic templates can be seen in the work by Chang et al. **[86]**, where they introduced a concept for linking low-level image features to high level semantics for video retrieval. The concept is known as semantic visual template. Another semantic template work is introduced by Smith and Li **[87]** where they introduced a concept called composite region template (CRTs) to decode image semantics. In their work, CRTs were generated from photographic images by segmenting the images into regions. They were able to demonstrate that CRTs performed well in classifying images and in searching for images in large collections.

2.2.2.2.5 Web Image retrieval

Existing Web image retrieval search engines make use of text to search for images, without considering the image content **[88]**. In web image retrieval, some of the HTML DOM elements contains some useful information that can be used as a basis for image search, however, the precision of the retrieved result is poor when the surrounding words are irrelevant to the image. The drawback of this approach is that it is a time consuming process as there are several results that are returned and users will have to go through the entire result list to find the desired images **[36]**. Recent research in Web image retrieval have proposed a fusion of textual context and visual features to provide better retrieval results **[89]**. In a recent approach, Web images are automatically annotated with a given set of concepts for retrieval purposes using a bootstrapping co-training framework **[90]**. In this approach, the system forms two independent classifiers by making use of the information from both visual features of the image and the HTML text. Their experiment results show improved performance of the system when used with a pre-defined set of 15 concepts, however, the performance of certain concepts were not satisfied due to inaccurate textual information that was extracted.

2.2.2.3 Global and Local Features

Most CBIR systems adopt either global or local features for object recognition. Global image features have the ability to describe an entire object with a single vector **[91]**. In image retrieval, colour histograms and variations are part of the global features that have been proposed to describe image content **[92]**. Since they focus on describing the image as a whole, global features cannot distinguish foreground from background, which limits this approach to only when the user is interested in the overall composition of the image as a whole rather than the foreground object. Global features have also been used in object recognition to describe an entire image. Most shape and texture descriptors fall into this category. Such features are attractive because they produce a very compact representations of images, where each image corresponds to a point in a high dimensional feature space **[91]**. Murase and Nayar **[93]** proposed a novel approach called PCA (principal component analysis) to compute a set of model images and to use the projections onto the first few principal components as descriptors. Their result shows that a whole new range of objects could be detected, however, the global nature of the descriptors makes it prone to problems such as image clutter and occlusions. This is because global features are sensitive to clutter and occlusion and, as a result, it is either assumed that the image only contains a single object or that a good segmentation of the object is available from the background. The literature on local feature recognition is vast and dates back to the 50's **[94]**. Local features are commonly associated with a change of a specific image property or several properties. Such properties can include colour, intensity and texture. Local features can be points, edges or small patches. In image retrieval systems that make use of local features, interest points are detected at multiple scales and are expected to capture the essence of the object's appearance **[91]**. Ideally, measurements are taken from a region for a local feature and converted into feature descriptors. The feature vectors can then be used for various purposes in several applications. An example is using feature vectors for finding image similarities by comparing feature descriptors generated from training image and query image. Usually, this requires a distance metric for performing a heuristic procedure in order to determine when a pair of feature descriptors is considered a match **[91]**. Local features can be used to detect objects despite significant clutter and occlusion. They also do not require a segmentation of the object from the background, unlike many texture features or shape features **[91]**.

2.2.2.4 Existing CBIR systems

Since the beginning of the image retrieval age, many commercial and research based image retrieval systems have been developed. Though each system was developed by different organizations and individuals, each one of these systems supports one or more of the following options **[95]**:

- Retrieval by example
- Retrieval by text
- Retrieval by sketch
- Retrieval by random browsing

Literature on existing content based image retrieval systems was reviewed. Venters and Cooper **[96]** reviewed 74 systems. A majority of those are however research prototypes. Here, we have selected a few CBIR systems and highlighted their distinct characteristics. Note that many of the systems listed have ceased to exist.

QBIC (Query by Image Content) is a product developed by IBM and it is one of the most well-known CBIR systems **[97]**. QBIC enables users to query by patterns, sketches or input image. QBIC uses multiple visual properties (colour, texture and shape) to enable users to graphically pose and refine queries **[98]**.

Virage supports texture, colour and spatial location matching **[98]**. There are many different Autonomy in Virage products for image processing, such as number plate recognition, face recognition and intelligent scene analysis **[67]**.

Octagon is a simple CBIR system developed in Java. In Octagon, image similarity is conducted based on visual contents such as colour and structure **[96]**. It has a very easy to use GUI as much attention was not paid to developing its interface. It has very limited functionality, which makes the usage quite straight forward. The usage of octagon is listed in the below sequential steps **[67]**:

- An image database is created when user selects images to import. Users can select a maximum of 10 images at a time.
- A user can then select a query image, whereupon the system returns a set of images most similar to the query image.
- Users are provided with a simple relevance feedback function where the user can indicate if returned images are relevant to the query or not.

IMatch [99] is a commercial CBIR system also found in the literature. IMATCH makes use of colour and shape features and each individual feature weight can be adjusted **[96]**. They ran tests with an image database consisting of 2 datasets from the Caltech image database, categorized into 1074 images of airplanes and 126 images of road vehicles. For the query image, they did a random selection of a car image. The query result returned by IMATCH was quite disappointing as it returned images that were dissimilar to the query image. It was observed that IMATCH produced a lot of airplane images as similarity results. The query search was reversed by using a picture of an airplane as query and IMATCH returned mostly images of cars. In IMATCH, you can also submit sketches as image query.

EMIR is another CBIR project described by lux and Savvas **[100]**. Emir is part of the software package Caliph and Emir, downloadable from Sourceforge **[67]**. Emir makes use of edge histograms, colour layout and scalable colour to search for image similarity. Emir also includes image annotation functionality powered by Caliph **[67]**. Caliph and Emir are released under a GPL license.

A more recent product is a feature of the Google search engine. **Google Image Search** utilizes the “Search by Image” technique. With the aid of computer vision techniques, Google image search matches a query image to the Google image index and other image collections. The precision of Search by Image's results is higher if the search image is more popular. Google Search by Image offers a "best guess for this image" based on the descriptive metadata of the results **[67]**. The search returns results containing the generated text description as well as related images. A test on Google image search confirms that the product has the ability to learn about an image after subsequent search for the same but then the image search only becomes a metadata search. Their test also confirms that colour information is very vital in Google image search and searching for a specific object in an image is not possible.

2.2.3 Content Based Image Retrieval Process

In content based image retrieval systems, there are four basic steps. They are discussed below in the follow section.

2.2.3.1 Image pre-processing

Image pre-processing is the initial step in CBIR, whose aim is to improve image data by suppressing undesired distortions or by enhancing some image features important for further image processing or analysis. This operation is performed at the lowest level of abstraction **[101]** and they do not increase image information content but decrease it if entropy is an information measure **[102]**.Below are the most common pre-processing steps.

- Image rescaling.
- Gray scale transformation.
- Contrast enhancement to increase the tonal distinction between various features in a scene. Examples are linear contrast stretch and a histogram-equalized stretch.
- Image filtering to remove sensor noise and speckles from the images in order to enhance the imagery for better interpretation and to extract features such as edges.

2.2.3.2 Feature Detection & Extraction

Feature detection is a low-level image processing operation, which examines every image pixel to see if the region around that pixel could be used as a feature **[103]**. Once features have been detected, the next step is feature extraction. During feature extraction, feature vectors are generated, which are commonly known as feature descriptors **[103]**. Feature detection and extraction are the most important aspect of a CBIR system. Using colour-based algorithms, such as colour histograms, was the first attempt towards digital image recognition using colour features **[104]**. Colour histograms were basically used to extract colour distribution features in any image and they were quite successful but the major setbacks were consistency and accuracy, usually when dealing with large image datasets **[105]**. Other attempts made use of algorithms that focused on corners, edges and image texture. A number of feature detectors have been developed. These detectors are based on computational complexity and repeatability. Some of the important detectors are edge, corner/interest point, Blobs/regions of interest and texture.

2.2.3.2.1 Feature Based Detectors (corners and edges)

In computer vision, recognition of corners/interest points and edges were the second attempt towards digital image recognition. This can be traced back to the work of Harris and Stephen in 1988, which is later called Harris Corner detector **[106]**. An interest point **[107]** can reduce the required computation time and, as such, these points are used to make up for some computer vision problems, such as object recognition, structure from motion, motion tracking, camera calibration, stereo matching, 3D reconstruction, image registration, image mosaic and mobile robot navigation **[108]**. As seen in several studies, there has been much research work on interest/corner point detectors. Some detectors locate corner points while others are programmed to find points of high local symmetry **[109]**. Corner points are derived at the meeting point of two or more edges. Tuytelaars and Mikolajczyk **[108]** describe certain characteristics of a good corner detector. These characteristics are described in the illustration in Fig 2.7.

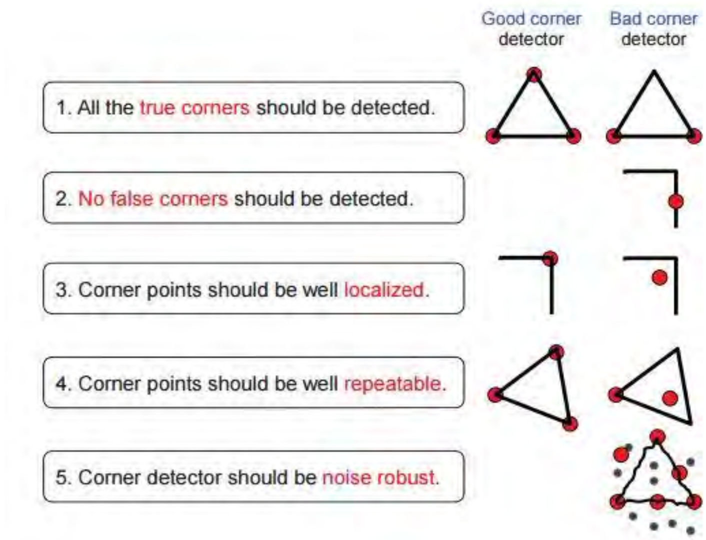


Fig 2.7 Characteristics of good corner detectors. [107]

A number of approaches have been proposed in order to achieve the characteristics illustrated above. These detectors have been categorized into several approaches but in this section we will only look into the intensity-based approach and the contour-based approach **[107]**. A very good example of intensity based approach is the Harris corner detector. The detector is the most successful algorithm in the intensity-based approach **[106]**. In this approach, corner points are derived if the auto-correlation matrix has two significant eigenvalues. The work by Harris was able to detect robust features in any given image but since it is limited to corner detection, his work suffered from a lack of feature-points connectivity which represented a major set-back for obtaining major level descriptors such as surfaces and objects **[106]**. Interest points detected with this approach (see Fig 2.8) did not have the level of invariance required to obtain reliable image matching due to this set-back.

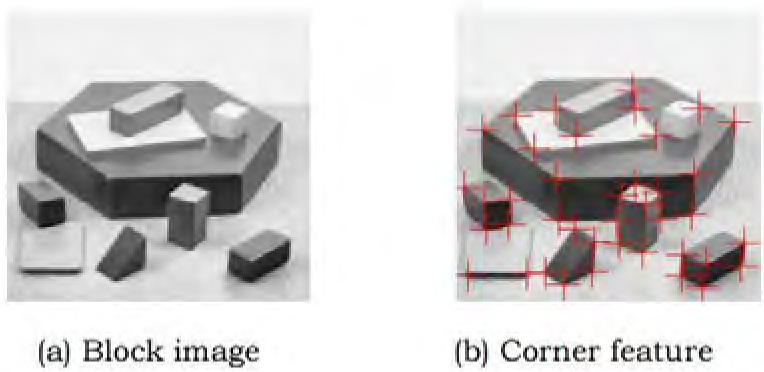


Fig 2.8 (a) a block image, (b) corner feature detection using Harris corner [106]

Harris then focused on ensuring consistency in the corners detected in order to overcome the set-back of his previous work **[106]**. He combined the isolated corners detected with a corresponding connection edge; this way, the corners randomly detected by the Harris detector where assigned to a specific space and geometry that could be more robustly matched **[110]**. The original work of Harris was later improved by Schmid et al **[111]**. In the contour-based approach, the focus is on line drawings rather than natural scenes **[107]**. The Curvature Scale Space (CSS) algorithm is a popular contour based method **[112]**. He and Yung **[113]** introduced a dynamic region of support and the adaptive curvature threshold to improve the original CSS corner detector.

Edges are the points where there is a sharp change in pixel values or gradient **[114]**. Edge detection can be achieved in an image by various approaches such as Laplacian operators, Sobel operators, Prewitt operators, gradient operators, smallest univalue Segment assimilating nucleus (SUSAN), Roberts operators, canny edge detectors, Krish operators, Isotropic edge operators, Harris and Stephens / Plessey operators **[114]**.

Robert, Prewitt, Sobel and Laplacian operators are based on the difference of gray levels but they are very sensitive to noise **[115]**. The presence of noise will most likely degrade edge points. The Canny edge detector was invented in 1986 by John Canny **[115]**. It makes use of a multi-scale algorithm to detect a wide range of edges in an image. Gradients operator are used in canny edge detectors to detect edge points. A Gaussian smoothing function is used to remove the effect of the noise in the edge detection. After the smoothing process, the signal is applied to the gradient operator. Figure 2.9 below illustrates edge detection using canny edge detectors.

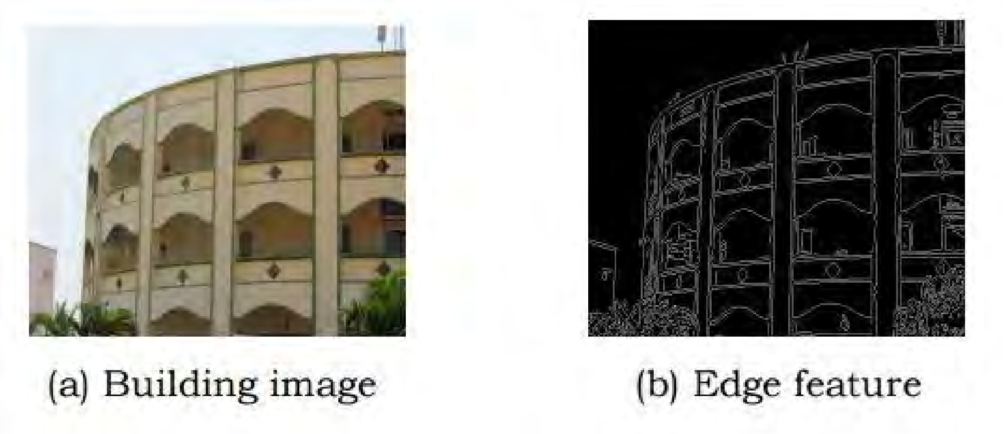


Fig 2.9. (a) a building image (b) edge detection using canny edge detectors [115]

Many other corner and edge detectors were published and made available to the public by the late 90s. Algorithms such as the SUSAN **[116]** corner detector and the FAST method are good examples of these algorithms. One of the most notable corner detectors is the FAST (Features from accelerated segment test) algorithm. This was presented almost a decade after the Harris Detector was published.

The **FAST (Features from accelerated segment test)** corner detector was presented in 1998 by Trajkovic and Hedley **[117]**. They argued that corners are one of the most intuitive types of features that show a strong two dimensional intensity change, and are therefore well distinguished from the neighbouring points, therefore the author's prioritized detection of corners over edges. Trajkovic and Hedley also corroborated the work of Tuytelaars and Mikolajczyk **[108]** on the characteristics of good corner detectors and added speed as part of the criteria. Trajkovic and Hedley **[117]** did a comparative study of the four most successful corner detectors, namely Harris, Modified Harris, SUSAN and Wang. The authors compared their performance to the FAST algorithm and their results show that FAST did better in accuracy than most of the four, except the Harris algorithm, but FAST was proved to be significantly faster than any other algorithm, which is important for real time computer vision application. The main contribution of FAST is the improved computational speed. The FAST algorithm make use of a corner response function (CRF) that gives a numerical value for the corner strength based on the image intensity in the local neighbourhood **[117]**. Also, a multi-grid technique is employed, which was responsible for the improvement in the computational speed of the algorithm and also for the suppression of false corners being detected **[110]**. Many researchers took the fundamentals of the FAST method to improve on it or to implement it in new applications after its undisputed success in speeding up corner detection. An example is the work by Rosten and Drummond **[118]**. In their work, titled *Machine learning for high speed corner detection*, the authors detected interest points by using a circle of 16 pixels (see Fig 2.10) around a point p to verify whether the point is a corner. P is detected as a corner when at least 12 of these 16 surrounding pixels are below or above the intensity of p by some threshold t.

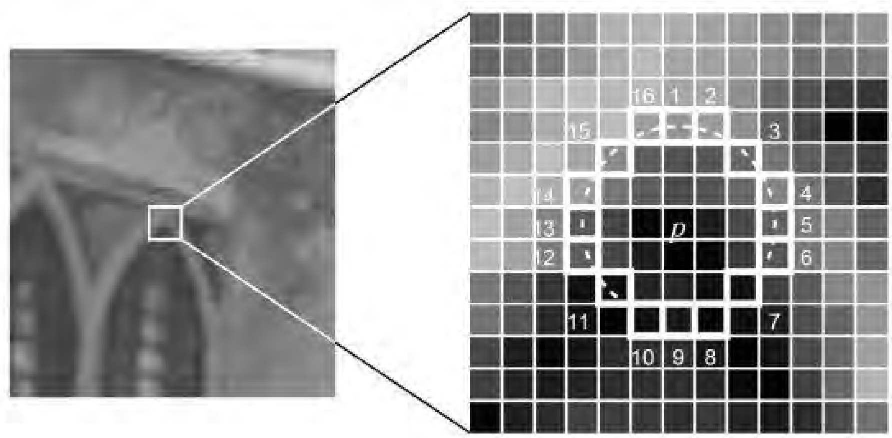


Fig 2.10. Image showing the interest point under test and the 16 pixels on the circle [118]

To be able to quickly reject points that are not corners, pixels 1, 5, 9 and 13 (see Fig 2.10) can be verified. The intensity of at least three of these pixels should be below or above the intensity of p, since a total of at least 12 pixels has to be reached to classify p as a corner. If this is the case, the other pixels will be verified as well before classifying p as a corner. If less than three pixels have intensities that are above or below the intensity of p, then p will be rejected as a corner point. Another improvement of the FAST algorithm is Oriented FAST, which is part of the ORB algorithm developed by Ethan Rublee et al. to be computationally efficient and invariant to orientation and rotation **[119]**.

ORB (Oriented Fast and Rotated Brief) was developed as a better alternative to SIFT (scale invariant feature transform) and SURF (speeded-up robust features) in terms of computational cost, matching performance and mainly dealing with the patent issue **[119]**. SIFT and SURF are texture based algorithms in computer vision. SIFT and SURF will be discussed in the texture based algorithm sections. ORB builds on the well-known FAST (Features from Accelerated Segment Test) algorithm to detect key points. One problem is that FAST doesn't compute the orientation so authors of ORB came up with some modifications to the FAST algorithm. In their first modification to the FAST algorithm, Ethan Rublee et al. argued that FAST does not produce a measure of cornerness and that they found out that it has a large response along edges **[119]**. To tackle this, they applied the Harris corner measure to find the top N points among the interest points. Another modification they did was to employ a scale pyramid of the image in order to produce FAST features at each level in the pyramid. This was for the purpose of producing multiscale features **[119]**. The authors also added an orientation layer to the modifications. They used an orientation by intensity centroid approach. This approach make use of intensity centroids **[120]**, which are simple but effective measures of corner orientation **[119]**. Intensity centroids are image properties, which are found via image moments. Image moments are weighted averages of the image pixels' intensities. They are very useful in describing objects after segmentation. For rotation invariance of this measure, Ethan Rublee et al **[119]** computed moments with x and y which should be in a circular region of radius r, where r is the size of the patch. ORB uses rBRIEF (Rotational Binary Robust Independent Elementary Features) descriptors, which are invariant to rotation because the ordinary BRIEF descriptor cannot handle rotation invariance. More information about the rBRIEF can be found in the work by Ethan Rublee et al **[119]**.

In 2010, Fraser et al **[121]** presented another improvement of the FAST algorithm and succeeded in achieving very distinctive matching features. In their paper, they were able to uncover weaknesses of corner based detectors. According to them, corner detectors perform very well with smooth or plain background images but detect false or no features at all on images with cluttered backgrounds. Guerrero Maridalia **[110]** argued that the possible cause of this is that corner detectors are based on the analysis of a pixel and its neighbours with no additional filtering **[110]**. This has a potential of resulting in erroneous detection. Although Trajkovic and Hedley **[117]** were able to overcome this problem by using a linear inter-pixel approximation approach to reduce the sensitivity of the algorithm to false corners in textured regions of an image, this still wasn't enough when dealing with images with very busy backgrounds.

2.2.3.2.2 Texture Based Detectors

The limitations of feature based detectors such as corner and edge detectors, when it comes to dealing with textured images, gave rise to the third attempt towards digital image recognition **[110]**. El-gayar et al **[104]** argued that the performance of feature based detectors were quite disappointing, especially when images are subjected to variations in colour distribution, rotation, illumination changes, affine transform and scale changes. Several researchers came up with different algorithms that focused more on image texture. Their aim was to achieve reliable image feature matching from textured images with the presence of cluttered backgrounds and lack of planar and well-defined edges. These algorithms are known as texture-based algorithms. In 1999, David Lowe **[122]** made one of the first approaches towards developing textured based detectors. In his publication **[122]**, **Lowe** established the possibility of extracting invariant features from images. Lowe demonstrated that by using a stage filtering approach to efficiently detect features. To ensure accurate detection of key points within very busy backgrounds, Lowe used image keys for local geometric deformations by representing blurred image gradients at multiple scales and in multiple orientation planes **[110]**. The experiment from his work shows the feasibility of object recognition in cluttered and partially occluded images.

SIFT (Scale Invariant Feature Transform): Lowe developed an improved version of his previous work a few years after his first publication. He presented SIFT in his publication *distinctive image features from scale-invariant features* in 2004 **[123]**. SIFT detects and extract image features that are highly distinctive. These features are invariant to scale, rotation and slightly invariant to illumination changes. The localization of SIFT features in frequency and spatial domains helps to reduce disruption by clutter or noise. A single SIFT feature can be correctly matched against a large database of features due to the high distinctive nature of the features **[123]**. The ground breaking work of this publication is the cascade filtering mechanism, which is used to reduce the cost of feature extraction **[110]**. Lowe **[123]** broke down the computation of features into four stages executed in descending order. The four stages are summarized below:

- **Scale-space extrema detection:** In this stage, stable interest points that are invariant to scale and orientation are detected in scale space using the difference-of-Gaussian function. SIFT approximates the Laplacian of Gaussian (LoG) with difference-of-Gaussian (DoG) function to identify potential interest points (see Fig 2.11).

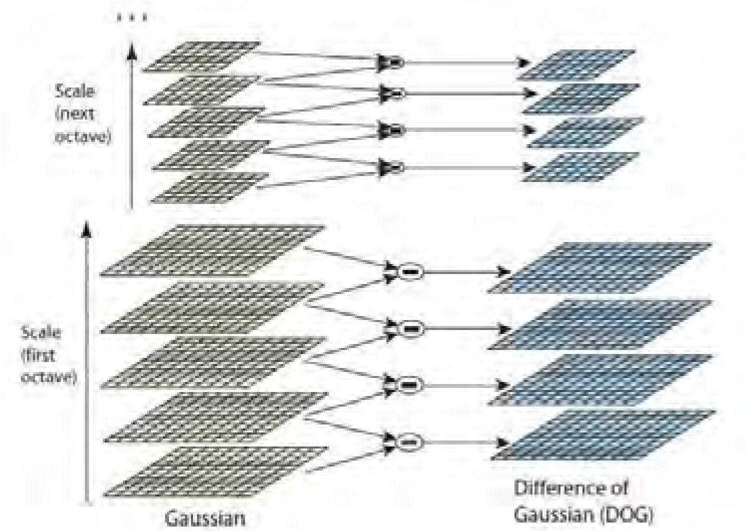


Fig 2.11 Approximation of the Gaussian Blob detector with difference-of-Gaussian function [123]

- **Keypoint localization:** At this stage, potential keypoints are further refined to get more accurate results.
- **Orientation assignment:** At this stage, an orientation histogram with 36 bins covering 360 degrees is created. The main purpose of this stage is to apply an orientation to each refined keypoint to achieve invariance to image rotation.
- **Keypoint descriptor:** At this point a 128 bin vector is formed to produce the keypoint descriptor. Other measures are also taken at this stage to ensure robustness against illumination changes.

The above mentioned cascade filtering approach has made SIFT very successfully in detecting stable keypoints invariant to scale and rotation changes. However, researchers and developers who tested the SIFT algorithm showered praise on its ability to produce accurate results but many had reservations about its 128-dimension feature vector descriptor, which made feature detection computationally expensive **[110]**.

PCA (Principal Component Analysis-SIFT): Ke and Sukthankar **[124]** were the first to make an attempt to develop an improved version of the SIFT algorithm. Their intention was to reduce dimensionality and to eliminate the computational cost of the SIFT algorithm. They employed the Principal component analysis technique to normalize gradient patch instead of the smooth weighted histograms used by SIFT **[124]**. PCA is a technique for dimensionality reduction **[110]**. The authors argue that in PCA-SIFT **[124]**, faster matching can be achieved since it requires less storage due to fewer components. Ke and Sukthankar have successfully been able to prove that PCA-SIFT can speed-up the SIFT algorithm matching process by an order of magnitude but it was proved to be less accurate than SIFT **[110]**.

SURF (Speeded-Up Robust Features): A few years after the work of Ke and Sukthankar **[124]**, Bay et al. **[125]** came up with an algorithm conceived to ensure high speed in three of the feature detection steps: detection, description and matching, without sacrificing the quality of the detected key points. SURF (Speeded Up Robust Features) was proposed at the ECCV 2006 conference in Graz, Austria **[125]**. SURF is a local feature detector and descriptor that was partly inspired by SIFT. SURF uses integral images to reduce computation time. For key point detection, SURF uses a Hessian blob detector, which can be computed with 3 integer operations using a pre-computed integral image **[125]**. The authors argued that Hessian matrix is used as a key point detector because of its good performance in computation speed and accuracy. For feature description, SURF uses the sum of the Haar wavelet responses around the point of interest in horizontal and vertical directions, which are also computed using integral images **[125]**. The authors claimed SURF is several times faster than SIFT and more robust against different transformations. In conclusion, SIFT, PCA-SIFT and SURF are now the most employed texture based digital image recognition techniques due to their robustness in invariant feature localization **[110]**.

2.2.3.3 Image Feature Matching

Image matching aims at establishing the relation between two images using different image features such as corners, edges and interest points, as explained in the earlier sections of this review. In image matching, a good measure of work has been demonstrated, which has proved to be valuable for various applications such as fingerprint recognition, robotics, face recognition and in various medical applications. In image matching applications, features from training images and the source image are compared using several image matching techniques in order to establish potential matches. Most times, image matching techniques output matches based on percentage of similarity. The image with highest percentage of match success is retrieved as output result **[126]**. Image matching techniques are used to find the existence of a pattern between query image and training images **[127]**. Many image matching techniques have been developed. One of the approaches is the minimum distance approach. Most CBIR systems make use of the minimum distance measures such as Euclidean distance, weighted sums, Hausdorff distance, or hamming distance to calculate the distance between the training images in the database and the query image **[67]**. In this approach, the image is initially divided into 3 x 3 equal sub blocks and features of selected sub blocks are used for similarity computation and comparison **[128]**. Colour or texture features are computed at each region of the query image, which is then compared with that of the training image, as illustrated in Fig 2.12 below.

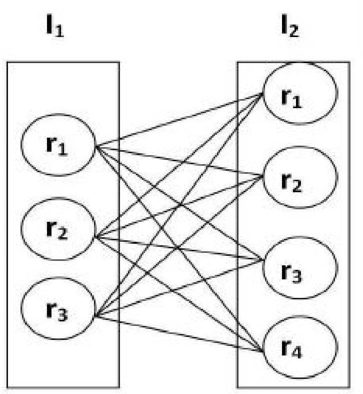


Fig 2.12 Each identified sub-block of image I1 is compared with that of I2. [128]

The Nearest Neighbour technique is one of the most commonly used techniques that makes use of the minimum distance approach. In this approach, images are first represented in n-dimensional feature vectors **[127]**. Euclidean distance is then calculated to find similarity among the objects. The lesser the vector distance, the more the percentage similarity **[127]**.

Other well-known techniques are the K-Nearest Neighbour technique **[129]** and the Approximate K Nearest Neighbour technique **[130]**. The major drawbacks of distance measure approaches is their low response time when there are many features to be compared. Many researchers have proposed indexing techniques to help minimize the response time in high-dimensional feature space. Examples are KD tree, Randomized KD Tree, Locality sensitive Hashing, LPC file indexing and so on. Many of these techniques have been tested and proved to improve matching speed. KD Tree is an algorithm that is efficient in low dimensional features but performance degrades gradually as feature dimension increases **[131]**. SIFT and SURF algorithms are example of algorithms that produce high dimensional features. Silpa-Anan and Hartley **[132]** proposed the randomized KD-tree algorithm. In their approach, data is split at each level of the tree into half. The split dimensions are then randomly selected from the first D dimensions on which the data has greatest variance **[132]**. Silpa-Anan and Hartley made use of a queue mechanism for all randomized trees. This mechanism allows an ordered search of each randomized tree according to the distance to each bin boundary. Another approach is the hashing method. Hashing transforms a data item to a low dimensional representation, or equivalently a short code consisting of a sequence of bits **[133]**. The application of hashing to approximate nearest neighbour search includes indexing data items using hash tables that is formed by storing the items with the same code in a hash bucket, and approximating the distance using the one computed with short codes **[133]**. In 1998, Indyk and Motwani **[134]** proposed the locality sensitive hashing algorithm that makes use of the hashing method. They developed a randomized hashing framework for efficient approximate nearest neighbour (ANN) search in high dimensional space. In 2009, Marius Muja and David Lowe **[135]** presented the FLANN (Fast Library for Approximate Nearest Neighbour) library for indexing multi-dimensional feature vectors. The intention of FLANN is to solve the exhaustive search problem presented by the K Nearest Neighbour algorithm by carrying out search based on approximations rather than exhaustive search. The FLANN library include implementations for several algorithms optimized for fast search in high dimensional feature spaces. FLANN contains an index class which is the base class used as a template with the type of elements for which the index is built **[136]**. This index class has specified parameters that act as a reference to one of the following:

- LinearIndexParam : This performs a linear brute-force search
- KDTreeIndexParam: This constructs an index that consist of set of randomized kd-trees that are then searched in parallel
- KMeansIndexParam: This constructs a hierarchical K-Means tree
- LshIndexParam: This creates an index that uses multi probe locality sensitive hashing

2.3 CBIR Systems with Mobile Devices

In Content Based Image retrieval systems, object recognition is the most important task. Object recognition and identification on mobile devices such as PDA's and mobile phones is nearly unexplored despite the wide adoption of mobile devices which has given rise to expansion of computer vision applications on mobile devices **[137]**. Ideally, consumer devices utilize several vision applications, such as image filtering but complex vision applications, such as real-time object detection, require much more processing power than what some mobile devices can provide. This may have been the reason for the poor exploration of real time object recognition applications on mobile devices. In view of this, most object recognition approaches make use of client server architectures such that, mobile devices are only used for image capture and simple pre-computation while the output is sent to a powerful remote server that performs intensive object recognition. Such application include the work by Fritz et al **[138]** in their paper 'Mobile vision for ambient learning in urban environments'.

In 2004, **Fritz et al [138]** proposed a Mobile Vision system for navigating urban spaces. The proposed system enables automated object recognition of images captured from a PDA or a mobile phone camera in the visitor's line of sight. After capture of image, the image is sent to a remote server via wireless connectivity. Object recognition is performed at the server. The server also receives a GPS position estimate and concludes a selection of relevant sights that potentially might appear in the tourist's field of view. The server sends a response back to the mobile device with multimedia type data about the history, architecture or any other related cultural context of historic or artistic relevance about the image.

In 2000, **Corrs et al. [139]** made use of a wearable tablet computer to determine position and line of sight of a user in an outdoor environment using a Geographical information system. The aim was to develop a system accompanying a tourist while he visits a city. Whenever the user requires information, the system provides such information based on the location of the user. Initially, the reference images are taken with known external camera parameters determined via GPS. This images are taken in different angles and views. Secondly, a relative transformation between the two cameras is calculated by matching the new image pair-wise to the reference images. With this transformation, the user position and line of sight is calculated.

Seifert et al. [140] introduced a system that detects and classify road signs using images captured with a PDA. The capture image is sent to a remote server for carrying out all computations. The captured image from the PDA is analysed using local features that are detected via pixel classification using trained colour filters. Ellipse filter and Hough transform are applied to retrieve the shape of the sign. The detected sign is classified by matching the object's pattern to reference patterns in a road sign database.

Hare et al. [141] introduced the Map Snapper project. In the Map Snapper project, users could query a remote database for information, facilities, events and geographical information about a particular region of the picture of a paper map that is taken by the user's mobile camera. This approach also make use of the traditional client and server approach where the mobile phone is used as the capturing and receiving device. Processing is done on a remote server.

Jean et al [142] proposed the SnapToTell project. This was a joint project by the French-Singapore IPAL Lab. In the application scenario, the tourist make use of his phone camera to take a picture of a tourist scene. This picture is sent to a server along with location information from the GPS to a service provider via multimedia messaging service (MMS). A little later, the system responds back with an audio clip (MMS) or text (SMS) that provides the tourist with more information about the scene. The application is focused on recognizing tourist scenes by distinguishing local discriminative patches described by colour and edge information and providing rich media information about the scene. The system database contains a number of images per object. Support Vector Machines were also used as discriminative classifiers.

2.4 Guidance systems with Mobile Devices

The use of guidance systems in cultural heritage spaces such as museums and archaeological sites (rock art sites) are one potential application of **content based image retrieval system** with **mobile devices**. This section examines the use of mobile guides in cultural heritage spaces as well as a background information on the topic.

2.4.1 Mobile Guides in Cultural Heritage Spaces

In the cultural heritage context, the use of mobile guides is increasing. They provide either guided tours or free-choice tours **[3]**. In a guided tour, the visitor are guided through a particular path, usually in a logical order to experience artefacts in the cultural heritage space. On the other hand, the free-choice tour allows users to select in any order which artefact they will like to learn about. This allows visitors to move freely around the cultural heritage space. The concept of free-choice-tour was developed by **Falk and Dierking [143]** as part of their contextual model used to describe how cultural heritage site visitors are now able to experience artefacts in heritage spaces according to their own needs. Many research have proposed mobile applications for guidance in the cultural heritage spaces such as museums but little work has been done in the context of cultural heritage spaces such as rock art sites. **Kenteris et al. [144]** classified mobile guides for cultural heritage spaces into 4 different categories, namely Web to mobile applications, mobile Web-based applications, mobile phone navigational assistant and mobile guide application. **Kenteris et al [144]** describe mobile guide applications as stand-alone or network applications developed to run on one or many mobile platforms to offer tourist/visitors information or services. This category is of particular interest in this study.

2.4.2 Background on Mobile Guides in Heritage Spaces

The emergence of new technologies, particularly mobile guides has changed the way cultural heritage site visitors interact with artefacts. In 1952, the Short-Wave Ambulatory Lectures was launched in Stedelijk museum in Amsterdam **[3]**. This was able to deliver information about individual art work in each gallery to visitors using a closed-circuit short-wave radio broadcasting in a pre-defined order. Visitors were able to listen to discussions via headphones. Shortly after the short wave ambulatory lectures was launched, the **Guide-a-Phone** was released by the American Museum of National History in 1954. In 1961, the American Museum of National History also adopted the Sound-Trek audio guide **[3]**. These earlier approaches followed a pre-defined path such that, the visitors are only allowed to experience and listen to audio playbacks about artefacts in a particular order. In 1993, the **INFORM** project was introduced at the Louvre museum in Paris. It was the first free choice mobile guide. The INFORM allows visitors to access more information about exhibit of their choice.

Proctor **[145]** did a survey on research and projects on mobile guides. From this survey, he discovered more than 100 applications and projects either still being researched or successfully implemented. One of such project is the **iGO**. In 1994, the first mobile tour in cultural heritage spaces using a PDA was introduced at the Minneapolis Institute of Art **[146]**. The project was called **iGO**. The **iGO** was deployed on a PDA and is regarded as the first free choice interactive guide on a PDA device. The iGO project was able to offer more than just audio guides to the visitors. Visitors were able to access cultural heritage content in audio and text format throughout site visits and they were also able to experience only cultural heritage artefacts that they are interested in. This was known to have enhanced the visitor’s experience **[146]**. In 1995, the iGO project was enhanced to provide a more personalized interactive tour to the cultural heritage site visitor **[146]**. The project was known as **Pocket Curator.**

Over the years, improvements were made to mobile guides in cultural heritage spaces and many museums and archaeological sites have been able to deliver multimedia contents of artefacts in their cultural heritage sites using various technology. For example, in 2002, **Vlahakis et al [147]** proposed the ARCHEOGUIDE (Augmented Reality-based Cultural Heritage On-site GUIDE). The intention of the project was to stress particularly the virtual reconstruction of ruined archaeological sites, while providing information on the archaeological site being visited. The ARCHEOGUIDE backbone is made of a central server, which stores the database with all information of the site. The user is required to carry a portable unit made of a lightweight portable computer, a camera, a microphone and headsets. The portable unit is then connected to the main server through a wireless LAN.

In 2003, **Bombara et al. [148]** made use of infra-red beamers for determining the position of a user to provide additional information in museums. If the user enters the small zone of an IR beamer and his beam can be detected by a PDA or cell phone, the location information is transferred to the mobile device.

In 2005, **Bay et al [5]** proposed an interactive system capable of recognizing objects in the Swiss National Museum. The object recognition system was implemented on a Tablet PC using a conventional USB webcam for image acquisition. This mobile device allows the museum visitor to take a picture of an object of interest from any position and this is processed and information about the object is communicated back to the device interface. In order to reduce the search space, Bluetooth emitters were installed in the museum premises. They compared the SIFT and SURF algorithms for object recognition of paintings in an art gallery. They concluded SIFT was better in recognizing paintings but SURF was better in processing speed.

Also in 2005, **Adriano Alberti et al [149]** presented an augmented reality museum mobile guide for delivering information based on the recognition of drawing/paintings detail selected by the user through a mobile device camera. Synthetically generated images were used to train the system. The recognition process utilizes a set of multidimensional receptive field histograms that represent features such as hue, edginess and luminance. In order to get information about a drawing or painting, the visitor points a PDA camera to the painting of interest. The camera view is provided on the PDA screen annotated with short text labels as links. By clicking the text labels associated to an object, the visitor may retrieve a multimedia presentation. The system architecture makes use of client-server technology. The PDA device communicates with a remote server via a wireless LAN. All object recognition is done at the server.

Ancona et al [150] presented the AGAMEMNON project. The project make use of mobile devices with embedded cameras to enhance visits of cultural heritage spaces like museums and archaeological sites. Their experiment demonstrated a 95% recognition rate with unrefined images of artefacts and archaeological scenes. The system architecture adopted is the classic client-server architecture. All object recognition computation is done remotely on the AGAMEMNON server. They made use of an edge based algorithm for object recognition. A more recent approach is the work of **Deufemia et al [192]** in 2014. They presented petro Advisor which is a volunteer based system for supporting the digital preservation of rock art sites. PetroAdvisor provide information that typically consists of Petro glyph pictures, descriptions, and several useful metadata, such as geo-referenced information and petroglyph contours.

Finally, **Fockler et al [137]** introduced the PhoneGuide. It was the first work we found that attempted to carry out object recognition and all computation on the mobile device in the museum context. It was first presented in Bauhaus University in 2005. In contrast to all other work reviewed above, the PhoneGuide enables users to take a picture of an object in the museum and object recognition is done directly on the mobile device rather than sending it to a remote server to perform computationally expensive image processing. Bluetooth emitters were used in combination with two layered neural networks. They were trained directly on the mobile device. Their main achievement was a simple and light weight object recognition technique that can run conveniently on a mobile device.

2.4.3 Mobile Guides in Rock art sites

The use of mobile guidance systems in outdoor cultural heritage space like the rock art sites is nearly unexplored. To the best of our knowledge, most of the early works and recent research in this domain only seems to focus on navigation of heritage spaces like indoor museums or at most open air museums situated close to civilization. We believe that this is because most rock art sites are often left in their historical context found many miles away from civilization. **Constance [151]** also stated that the ambience of the site context is lost if the cultural heritage artefact is removed from its original location. With this in mind, it is clear that most approaches that have been used in previous research and projects may not work for rock art heritage sites. For example, the use of the traditional client-server architecture via Internet or GPRS adopted by most of the earlier and recent approaches may not be ideal as rock art sites are located in places where there is little or no network connection. The work by **Jean et al [142]** proposed the use of multimedia messaging service (MMS) as a means to send query images to the server and receive a response back in text message (SMS) or Multimedia message (MMS). This approach also may not work well for rock art sites located where there is low or no mobile network signal, which will certainly hinder sending MMS to a remote server. **Fockler et al [137]**, **Bay et al [5]** and **Corrs et al. [139]** have made use of location awareness techniques like Infra-red beamers, Bluetooth emitters and GPS to determine either the user's location or location of the artefact of interest. In most cases, these involve the installation of this equipment on site. An approach like this may not be ideal for a rock art site as most government around the world prohibit all form of human activities that may damage the rock art. With all the above mentioned, an approach that can run independently without any form of network or location support is desirable in a rock art heritage site.

In this research, our approach is tailored towards the development of a standalone **image-based** mobile guide for navigating rock art heritage spaces by investigating object recognition algorithms for identifying rock art paintings. The use of **image** input modality for information access in tourism applications is gradually gaining momentum. A recent field study by **Davies et al [152]** concludes that a significant number of museum and heritage site visitors (37%) embraced the use of image-based object identification of artefacts. Although the use of images to navigate cultural heritage spaces is not a new technique, digital recognition of rock art images is far from easy. According to **Zhu et al [1]**, the diverse and extraordinary nature of rock art images defies most image processing algorithms, which make object recognition of rock art a problem. **Fockler et al [137]** made use of normalized colour features for recognizing artefacts in a museum to enable object recognition computations on mobile device. They claimed normalized colour features are easily computed and recognition computation was done in just seconds on a mobile device. Most rock art are constantly exposed to factors that causes irreplaceable damage. We believe the use of colour features to recognize decaying and faded rock art paintings may not produce accurate recognition results. Our approach is a little similar to the museum mobile guide presented by **Bay et al [5]**, where they made use of pictures of artefacts taken by the camera of a PDA in a museum. Their approach compared the performance of SIFT and SURF algorithms for object recognition of museum artefacts on a remote server. Our approach differs in the processing mode (processing done completely on the mobile device) and the images to be recognized (rock art images), which is a more complex image in object recognition literature. Our approach also compared the performance of the SIFT, SURF and ORB algorithms for object recognition of rock art images.

2.5 Summary

This chapter has discussed all the background information that forms the basis of this research. Section 2.0 gave brief information about the various topics and how each of this topic fits into the research context.

Section 2.1 gave a definition of cultural heritage spaces and various examples of cultural heritage spaces were also mentioned with a focus on rock art heritage spaces. We then focused on the rock art of Southern African, which are the works of the Khoisan people, the indigenous people of Southern Africa. To navigate a rock art heritage site with mobile guides, we found it important to know the type of artefacts that exist in a rock art site. So, different types of rock arts such as petroglyphs and pictographs were discussed. Various characteristics that influence the appearance of rock art were also discussed. This is important because in object recognition, it is paramount to know in advance the type of image one will be dealing with. In section 2.2, the concept of image retrieval was discussed. We also discuss content based image retrieval which was born out of the necessity to find solutions to manual annotation of images through indexing of images based on their content.

Section 2.3 discussed the use of CBIR systems in mobile guides. In this section we were able to use various examples to identify that most mobile guides that have used CBIR systems rely heavily on the classic client and server architecture as a medium for processing. We believe this is because much more processing power is required for computation of complex computer vision applications, which were only available on personal computers.

In section 2.4, we discussed the use of guidance systems with mobile devices in cultural heritage spaces such as museums and archaeological sites. We have shown through various works how technological advances in mobile technology has shaped the development of mobile guides for navigating cultural heritage spaces. We then concluded this chapter with introducing mobile guides for rock art heritage spaces. To the best our knowledge, there hasn’t been any record of any image based mobile guide for navigating rock art heritage sites.

Chapter 3

DIGITAL RECOGNITION OF ROCK ART

3.0 Introduction

Searching for accurate and reliable techniques for digitizing and analysing rock art has been the hallmark of rock art studies **[154]**. The traditional approach is usually the manual tracing of pictographs and/or the rubbing of petroglyphs is a complicated and time consuming task that does not guarantee reliable results **[155]**. This method is also considered very intrusive and has the potential ability to cause damage to the rock art. Later research adopted the use of photography for digitizing rock art **[155]**. Technological advances such as digital image processing are now available to make sense of real world photos by detecting and extracting information from these photos. This approach has great potential for the study and analysis of rock art images **[154]**. Despite these advances, research in digital recognition of rock art is still lacking. Researchers have come up with algorithms and techniques for object recognition but little work have been done in recognition and segmentation of rock art **[155]**. In this chapter, we summarize some of the methods and techniques that have been adopted for rock art object recognition. This section also presents our approach for digital recognition of rock art paintings to produce good features for matching. This is done by comparing the performance of the SIFT, SURF and ORB image processing algorithms.

3.1 Supervised Learning Technique

Supervised learning techniques can be used for object recognition and classification. Detailed information about supervised learning is available in the literature review chapter. In 2002, **Diaz and Castro [156]** made use of supervised learning techniques to classify rock art images. They used neural networks to recognize and learn three rock art image patterns. However, their network suffered setbacks from oscillation with the last pattern trained always emerging as the output label when presented with new examples.

3.2 Unsupervised Learning Technique

Unsupervised learning is used to draw inferences from datasets consisting of input data without labelled responses. **Deufemia et al. [157]** presented an approach that made use of a combination of an unsupervised recognizer such Self-organizing maps (SOM) with Fuzzy Visual Language Parser for classifying petroglyphs. Self-organizing maps (SOM) make use of the clustering technique with an unsupervised learning process. In SOM, clusters are organised in a two-dimensional topologically ordered feature map where common features are positioned geometrically close to each other. Fuzzy Visual Language Parser makes use of grammatical specifications to embed information about patterns and structures with their relationships. The former classifies rock art engravings extracted from a scene by using a Radon transform **[158]** as shape descriptor. The latter exploits the archaeological knowledge about recurring patterns within scenes to solve ambiguous interpretations. Their work has been evaluated on a set of 50 petroglyph scenes containing about 500 carved symbols from the Mount Bego rock art site. They claimed their experiment produced very positive results.

In 2013, **Deufemia et al. [159]** made an improvement over their previous work. They also made use of SOM with an Image Deformation Model (IDM). SOM clusters were developed with the aid of shape context descriptors **[160]**. This method connects each of n points identified on a shape contour to all other points through n – 1 vectors. According to Daniel et al **[161]**, Image Deformation Model computes the distances of each pixel of a query image and the corresponding pixels of the target image located within a certain twisted range by taking into account the surrounding pixels. In their experiment, they used a dataset containing about 51 binarised images grouped into 17 classes. The classes were manipulated to achieve rotation and skewed images. The final dataset contains about 1530 manipulated images. In the evaluation of their experiment, they were able to achieve an accuracy of about 68.6% with the combination of SOM and IDM. The flaw of this approach was that the process is very time consuming.

Fig 3.1: An example of a digitised and binarised petroglyph that **Deufemia et al. [159]** attempted to classify.

3.3 Segmentation Technique

In 2014, **Deufemia et al. [162]** made another advancement on their previous work in 2013. They especially focused on maintaining the accuracy while trying to improve the speed of computation. Their approach focused on the segmentation of petroglyphs scenes and recognition of petroglyph symbols. Their recognition procedure adopts an Enhanced version of Generic Fourier descriptors (EGFD). The generic Fourier descriptors have been improved by **Takaki et al [163]** through a shape normalization before extraction of the features. EGFD is a descriptor algorithm that is invariant to rotation, and scaling and robust to deformations. The algorithm examines each isolated part of the scene extracted from a rock panel by combining them with the closer ones in order to obtain the more likely combination **[162]**. This approach was evaluated on a dataset of 53 complex scenes with each containing an average of 22 petroglyphs. Their approach is broken down into four steps and also illustrated in Fig 3.2 below:

- Elements that are not part of the target petroglyph symbol or object are first eliminated. At this stage the image is binarized and a median filter is applied.
- The image is then segmented by extracting the connected components.
- The components are then grouped into clusters with the aid of Euclidean distance among them. This results in a set L clusters.
- A greedy algorithm is used to finally used to select the elements from L based on the EGFD distance among them.

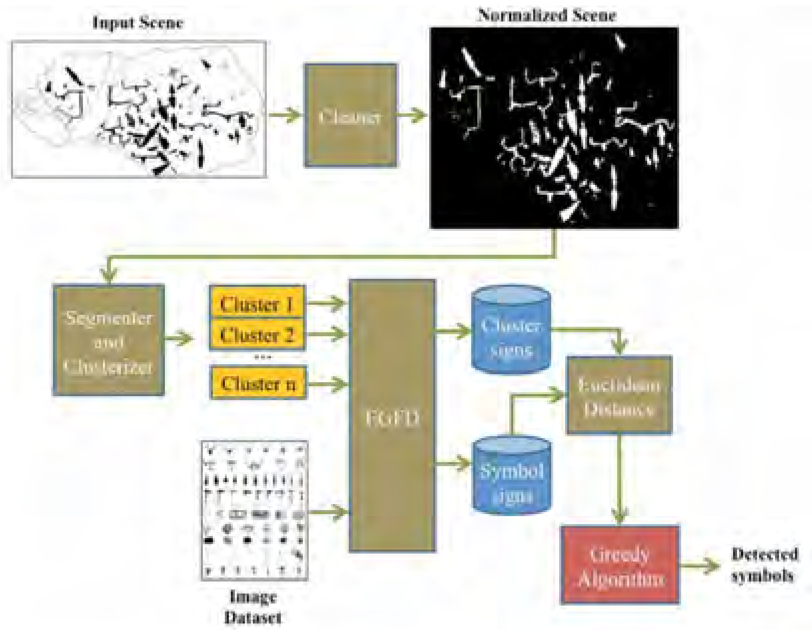


Fig 3.2: Illustration of the segmentation and recognition process described by Deufemia et al. **[162]**

They were able to demonstrate improved speed and accuracy over recognition based on Image Deformation Model (IDM) through their experiment **[162].**

Seidl and Breiteneder [164] proposed a pixel classification technique for segmenting petroglyphs. Their approach was based on a support vector machine classifier. They made use of SIFT (scale invariant feature transform), Local Binary Patterns **[165]** and gray level co-occurrence matrices **[166]** to train three separate support vector machines and the performance of each was tested. This was used as they argue that colour, edges, and texture features are best for segmentation. They were also able to experiment with a combination of the above mentioned features and their performance was also measured. The combination of output labels of the best performing classifier from above was also experimented with and they employed a majority voting method with these classifiers. See Fig 3.3 for an illustration of the latter approach. The latter approach was found to be better than others as performance measurement using standard precision and recall yielded 88% and 41% respectively.

Fig 3.3: Result of the latter classifier with majority voting method [164]

Amrit et al **[153]** attempted to segment images of some pictographs from the SARADA (South African Rock Art Digital Archive) collection. This exercise was particularly intended to clean up the cluttered background of the pictographs so that algorithms can focus mainly on the object of interest. They applied standard image processing algorithms for segmentation and smoothening. Amrit et al. made use of Sobel, Prewitt, Roberts, Laplacian of Gaussian, and Canny methods to perform image segmentation. However, the result of their experiment demonstrated that most of the methods were unsuccessful in segmenting the pictograph. The ones that were successful require fine tuning. They claimed Sobel edge filters achieved a far better result than other image processing algorithms. But this didn't scale particularly well as a substantial part of the object of interest was removed by the segmentation algorithm and the result differs for each image. This goes to show that the algorithm does not generalize well for all rock art images. The result of the Sobel edge filter from their experiment is illustrated below in Fig 3.4.

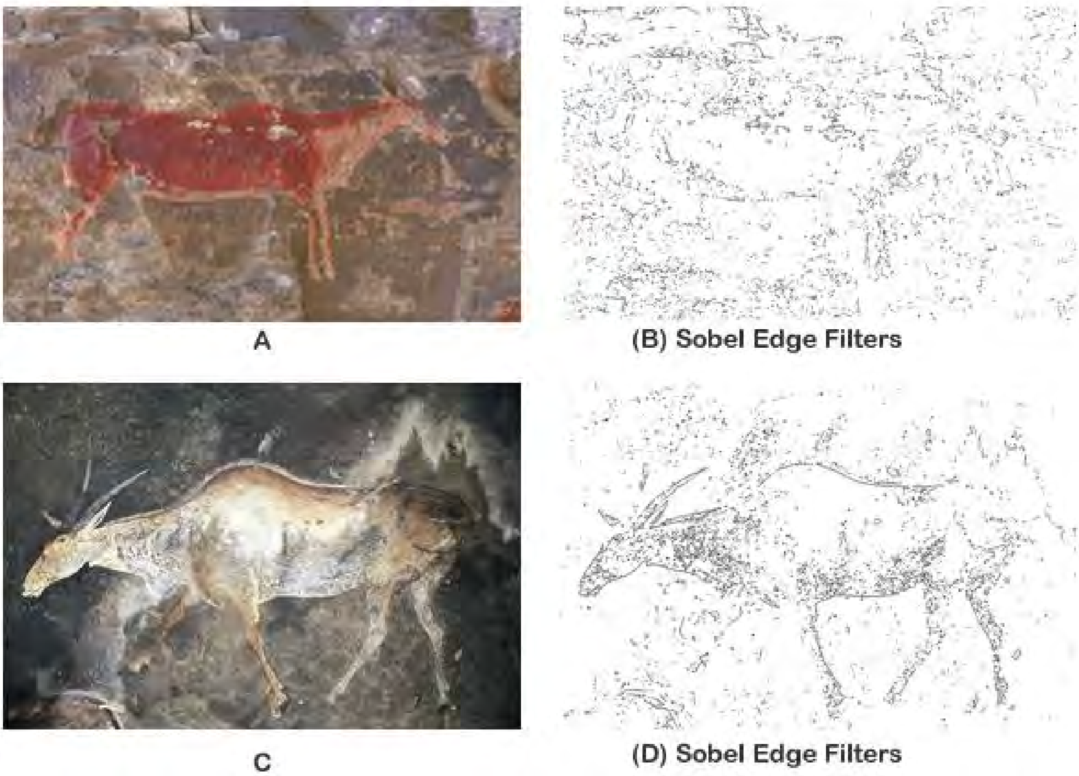


Fig 3.4 (A) Low contrast pictograph, (B) result of A after applying Sobel edge filters, (c) High contrast pictograph, (D) result of C after applying Sobel edge filters. **[153]**

Amrit et al [153] also made attempts to reduce the background noise of a pictograph sample from SARADA by applying an edge preserving anisotropic diffusion filter **[167]** and Gaussian smoothing. This was in an attempt to simplify the image features and remove as much background noise as possible. Their result shows that the Gaussian smoothing blurred the image but the background did not fuse into a greyish-brown colour. The horse painting also did not blend into a solid red colour (**Fig 3.5B**).

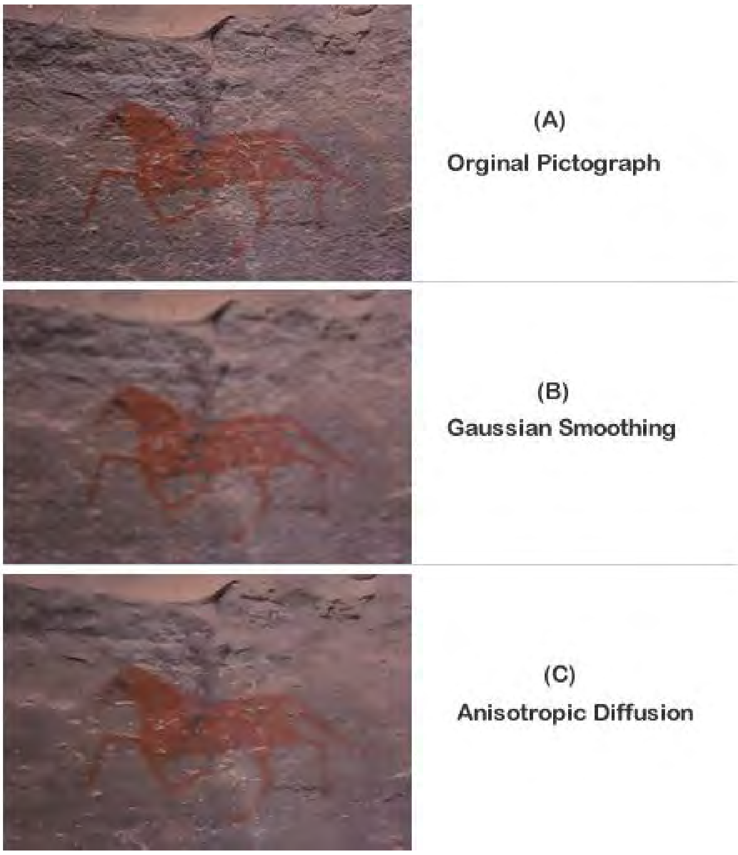


Fig 3.5: Result of the application of Gaussian smoothing (B), Anisotropic diffusion filter(C) on the pictograph image (A). **[153]**

In the result of the anisotropic diffusion filter (**Fig 3.5C**), the spots on the horse painting seem to have been over emphasized by the filter though it appears better than the Gaussian smoothing. With the above representation, it is evident that these methods do not generalize well to all pictograph images since their parameters have to be fine-tuned for each image to achieve a good result. A more robust approach is required for recognizing objects in pictographs even at their cluttered state.

3.4 Human Computation Technique

Commonly known as crowdsourcing, human computation involves giving certain critical steps to humans that are difficult for computers and easy for humans. In their publication*, an efficient and effective similarity measure to enable data mining of petroglyphs,* **Zhu et al. [1]** propose the usage of a slight modification of the generalised Hough transform (GHT) for the mining of large petroglyph datasets. They made use of datasets with more complex petroglyphs. They were able to show existence of petroglyph images where a single petroglyph consists of several parts and the possibility of merged parts of petroglyphs that drastically change the topology of the petroglyphs. They argue that no image processing algorithm is fit enough to produce good recognition for such complex image structures. They created the CAPTCHA –ROCK system where humans are used to help computer algorithms to segment and annotate petroglyphs (see Fig 3.6). They extensively evaluated their approach and achieved good results. However, they mostly evaluated synthetic petroglyph shapes drawn by human volunteers.

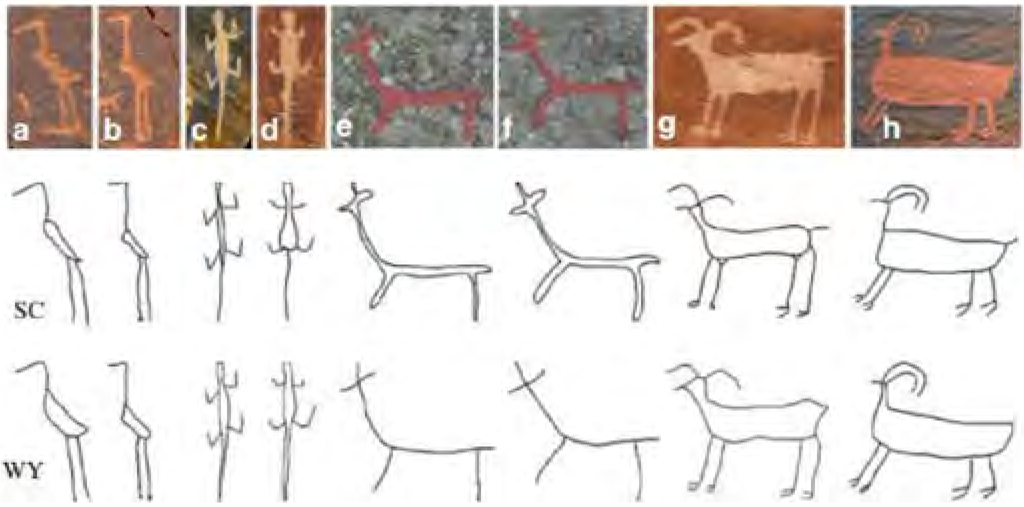


Fig 3.6: Extracts from **Zhu et al [1]** showing result of a crowdsourcing exercise by two different volunteers.

3.5 Shape Similarity Technique

Takaki et al [163] presented methods to characterize shapes of the petroglyphs in Central Asia. The first method was to characterize shapes of each rock art by the use of a software program for image analysis, and the other was to use statistics of quantities that are considered to characterize the properties of the groups of petroglyphs. Shapes of binarized petroglyphs were treated by image analysis to obtain simplified shapes called skeletons (**see Fig 3.7**).

Fig 3.7 (Left) A binarized petroglyph, (Right) the skeleton of the binarized petroglyph **[163]**

The skeleton **[168]** is a group of centre lines within a silhouette. A skeleton is made up of a given shape made of a closed contour. Many contact circles are drawn inside the shape. Trace of the centres of these circles produce a combination of curved lines, which is called a skeleton. Structures of skeletons are expressed by the use of some symbols, from which quantitative comparison among petroglyphs is shown to be possible **[163]**. The authors experimented on petroglyphs in central Asia and they claimed that their results were very promising.

Seidl et al [169] introduced an approach for petroglyph shape descriptors based on the skeletal graph topology **[170]** and propose matching with graph edit distance (GED) and graph embedding (GE). The authors used 21 different scalar topological features for graph embedding. Graph embedding was used to create feature vectors for each graph and the distances between graphs are calculated with Euclidean distances. They then evaluated the descriptor and the matching on 100 digitized and binarized tracings of petroglyphs classified into 10 classes with 10 examples each. The K Nearest Neighbour algorithm was further used for classification. They then compared the performance with other shape descriptors used in petroglyph classification (See fig 3.8).
In their experiment, matching of the skeletal graphs with GE and with GED delivers comparable results. They claimed both matching methods achieve 57% accuracy. GED is of high computational complexity, whereas GE has low computational demand due to low feature vector dimensionality. The two best performing combinations of graph embedding features have only 4 and 5 feature dimensions.

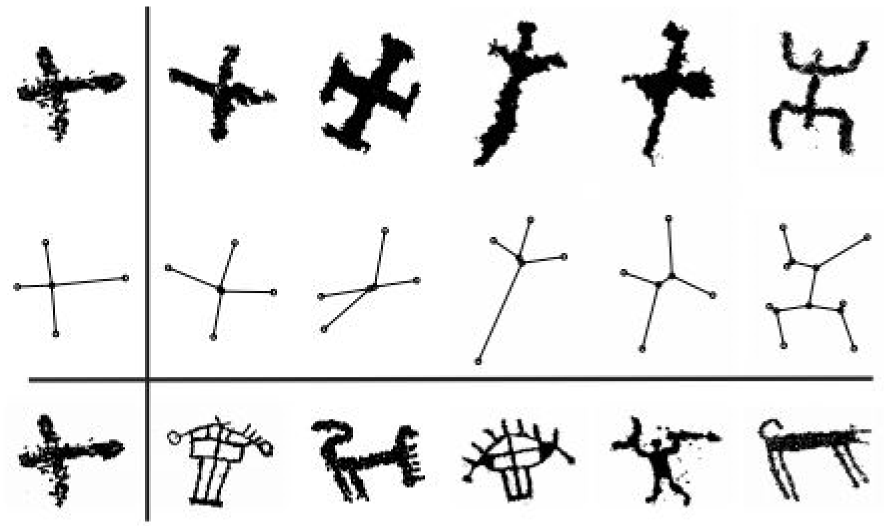


Fig 3.8 Illustration of correctly classified binarized petroglyphs using graph embedding and mis-classified binarized petroglyphs using GHT. The query image is to the left. The first five results are to the right. The first two row is the classification using GE. The last row is the classification by GHT.

Seidl et al [169]

They claimed the combination of skeletal descriptor with other descriptors such as the GHT (generalised Hough transform) **[1],** Inner distance shape context **[171]** and Shape Context **[172]** yields a classification performance of 88%, which is not achieved without their proposed skeletal descriptor. They concluded that descriptors derived from skeletons are valuable for petroglyph classification.

Wieser et al [173] present a study on skeletonization of real-world shape data that represents contact tracings of prehistoric petroglyphs. In their study, they discovered that skeletonization of petroglyphs is challenging since their shapes are complex, contain numerous holes and are often incomplete or disconnected. They proposed an adaptive automated shape pre-processing step that enables the computation of robust skeletons for complex and diverse petroglyph shapes. We will only focus on the pre-processing as this is relevant to our research. Their pre-processing technique consist of a combination of 4 methods executed in a cascade mode repeatedly **(See fig 3.9)** till certain conditions are met. These methods are listed below:

1. Close small holes,
2. Smooth the contour,
3. Connect nearby parts, and
4. Avoid the unwanted decomposition of the shapes

The authors defined a stopping criterion for the pre-processing steps that makes use of a robust indicator function that is suitable for the different complex shapes in the dataset. For each petroglyph image, the pre-processing steps are ran repeatedly till the criterion is met. They evaluated different indicator functions such as solidity and circularity of the shape, the number and size of foreground and background blobs, and the number of endpoints in the thinning skeleton.

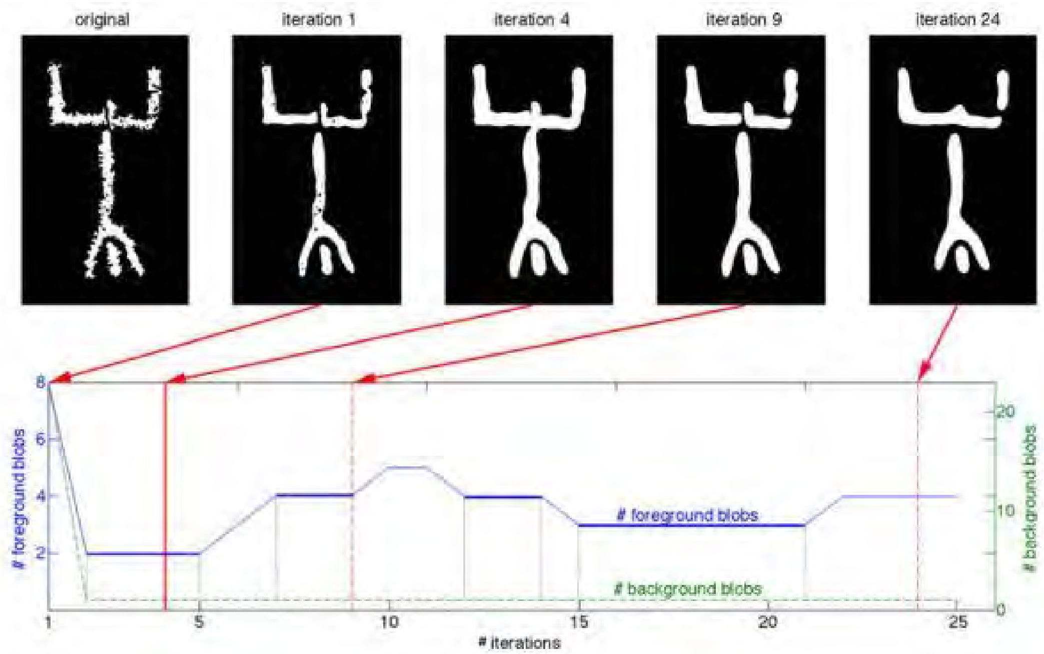


Fig 3.9 Sequence of the automated pre-processing of a shape with the proposed method. The numbers of foreground and background blobs decrease rapidly and the stopping criterion aborts the loop at iteration 3. Further iterations would decompose the shape again **[173]**

They performed their pre-processing experiment on a large heterogeneous database of petroglyph shapes that exhibit numerous challenges to skeletonization algorithms. Evaluations on their dataset show that 79.8 % of all petroglyph shapes can be improved by the proposed pre-processing techniques and are thus better suited for subsequent skeletonization. They concluded that proper pre-processing is crucial for successful sketelonization of petroglyphs

3.6 Applicability to this Research

While some of the techniques and algorithms described in sections above are well tested and most of them were successful within the confines of their chosen dataset, many of the works concentrated more on well refined images that have been segmented from the rock surface and then binarised to produce clean images of rock art with no cluttered backgrounds. In this research, the approach is more focused on recognition of features in pictograph images and retrieving images with similar features from an archive using standard image similarity search algorithms. In contrast to the approaches reviewed in this chapter, our dataset contains raw and unrefined photographs of pictographs with problems such as cluttered background and jagged edges. Other noticeable problems of our dataset include pictographs that are partially occluded, cracked and faded. According to **Phil and Margarita [154],** each rock art is unique with its own unique problems. With this in mind, it is evident that no single image processing technique can solve these problems. Our approach made use of a combination of image pre-processing techniques (**see section 4.7.1**) in combination with object recognition algorithms (SIFT, SURF, ORB). These two methods work on different area of the problems and these two methods complement each other to produce a solution for recognizing rock arts features. This approach may not necessarily work for rock art with other challenges different from the ones observed in our dataset (see Fig 3.10). The approach in this research is focused on the comparison of 3 standard image processing algorithms, such as SIFT (Scale Invariant Feature Transform), SURF (Speeded-up Robust Features) and ORB (Oriented FAST and Rotational Brief) for recognizing rock art paintings.

Fig 3.10 Sample photographs of pictographs illustrating cluttered surface and noisy background. Photograph taken in Clan-William at the Cederberg Region Western Cape, South Africa. Pictures taken by A. Olojede

These algorithms will be used as a tool for detecting features on rock paintings and using such features to retrieve similar rock paintings from an archive. These algorithms were specifically selected based on their general performance, especially with their ability to identify highly distinctive local image features, which is beneficial in image similarity search. More details about each of these algorithms is found in the previous chapter. Computation will be carried out completely on a mobile device, as rock art heritage sites are often left in their historical context located many miles away from civilization, hence the use of mobile devices becomes necessary for capturing pictographs and finding features of the same to perform image similarity computation. The use of image processing algorithms is computationally demanding and many believe that they may be unsuitable on mobile devices, especially when dealing with images with complex structures such as the Southern African pictographs.

3.7 Summary

This chapter has provided some insight into the various techniques and approaches that have been used in the digital recognition of rock art images for classification and presentation purposes. This chapter also shows that some of the techniques were successful within the confines of the image datasets used. Most of the techniques used refined image datasets that have been cleaned using several image segmentation techniques to produce high quality features, unlike a dataset(combination of rock art paintings from SARADA and pictures taken at a rock art site in Clan William at the Cederberg region) that are raw photographs of rock art paintings with rough and cluttered backgrounds. Several works reviewed in this chapter have also shown that most of the approaches were focused more on recognition and classification of pre-segmented rock art engravings (petroglyphs). This chapter then concluded by discussing the applicability of the previous approaches to this research and gave an insight into the main focus of this research.

Chapter 4

DESIGN

4.1 Introduction

In this chapter, we describe the design approach and we also describe all the tools and techniques that were employed in implementing the system. The source and selection of the dataset is discussed, and the algorithms selected (mentioned in the background chapter) and how they were implemented to extract features from the dataset is discussed as well. The open source framework and the technologies used in the implementation of these algorithms is discussed. The image matching techniques to compare the extracted features are also discussed. We also discuss the design approach, the system design and experiment design.

4.2 Research Aim

This research aim is two-fold.

- The first is to investigate the performance of 3 standard image processing algorithms (SIFT, SURF, ORB) for digital recognition of the Southern African rock art paintings using mobile devices. Recognized features will be extracted and used for rock art image similarity search.
- The second aims to investigate the feasibility of an image based mobile guide for provision of information about artefacts in a rock art heritage site.

In view of the research aims, 2 experiments were conducted. The first is a **comparative experiment (see Section 4.8.1)** that was used to evaluate the performance of the 3 selected algorithms. This acts as the primary indicator of quality. The first experiment will also help in answering research question 1.

The second is a user experience experiment that involves **testing with users (see Section 4.8.2)**. The experiment assessed the usability of the mobile guide application in the cultural heritage context. As part of the second experiment, an **interactive experiment** (**Section 4.8.2.1**) was first conducted using the mobile guide prototype, running the algorithm with the preferred performance from the comparative experiment. This is then followed by a usability evaluation. This will help to answer research question 2.

4.3 Resources and Tools

This project required certain software and hardware. The prototype and final system can be installed on a mobile device running Android OS. The required resources are discussed below:

4.3.1 OpenCV

OpenCV (Open Source Computer Vision) is a software library mainly aimed at real time computer vision applications. It was developed at the Intel Russia research center in Nizhny Novgorod, and is now supported by Willow Garage and Itseez **[174]**. It is free for use under the open source BSD license. The library is cross-platform. It focuses mainly on real-time image processing. OpenCV contains implementations of the SIFT, SURF and ORB image processing algorithms being investigated in this research. OpenCV also contains various implementations that we adopted for matching image features detected by these algorithms. One of such is the FLANN (Fast Library for Approximate Nearest Neighbour) library **[135]** that contains methods for indexing image features for fast retrieval.

4.3.2 Android Powered Mobile Device

This research is focused on developing the application on the Android platform. A mobile device with processing speed of at least 500MB RAM is considered the minimum hardware requirement. The minimum Android version is 4.0. For this research, we have made use of Samsung Galaxy S6 with a memory capacity of 3GB for evaluating the performance of the algorithms. A Samsung Galaxy Pocket was used as the device to conduct the user experience experiment *for* usability evaluation. This was because we want to ascertain the algorithm performance on a low resource device.

4.3.3 Android Developer Tools (ADT)

The ADT is an Eclipse IDE with built in android development tools to streamline the development of Android applications. With a single download, the ADT includes everything needed to start developing Android apps, such as;

- Eclipse IDE
- Android Development Tool Plugin
- Android Platform-tools
- Android system image for the emulator

4.4 Data Source and collection

As mentioned in Section 4.3, collection of data was carried out in three phases with one of the phases involving a physical site visit. All the images were representative of the characteristics of rock art explained in the background chapter. Images have a cluttered background with rough and jagged edges. Some were also faded. Generally, data can be categorized into the training set and query set.

4.4.1 Training set

In the first phase, we collaborated with the Department of Archaeology at the University of Cape Town headed by Professor John Parkington. The Department of Archaeology provided an archive of some pictographs from the Western Cape region of South Africa. The archive was used for our initial testing and implementation of the algorithms using their default settings. The second phase was extracting some pictographs from the SARADA archive, but it was realised this will not be sufficient to carry out experiments based on a real life scenario. In a real life scenario, a user will most likely take pictures at different angles and light conditions during a site visit. Most images of pictographs in the SARADA archive are carefully taken and in most cases are front views. The images from the Department of Archaeology at the University of Cape Town and SARADA archive have similar characteristics and are hence used as a training image set. With this in mind, a third phase collection focused on taking photographs of pictographs from a Clan William rock art heritage site located in the Cederberg region. This rock art site was preferred due to its proximity to Cape Town and because of a referral from Professor John Parkington at the Department of Archaeology.

Images were taken with a digital camera and most were taken in day time under good light condition and taken at different angles. There were also cases where the paintings are in caves. The light condition in the cave is minimal and this was particularly good for testing illumination changes in pictograph images. See Fig 4.1 below for sample data from the Clan William rock art site.

Fig 4.1: Sample rock art images taken from the two sites in the Cederberg Region, Western Cape, South Africa Photographs taken by A. Olojede, H. Suleman

4.4.2 Query image set

There were also images taken by 2 students at the rock art site with their mobile phone camera. These students played the role of site visitors and were allowed to take pictures in whichever way they liked. These images represent the query images. This is because it was necessary to make sure that experiment is conducted using a real life scenario. The query set was used for the first experiment that served the purpose of evaluating the performance of the 3 selected algorithms for recognition of rock art pictographs. A total of **32 images** was extracted from the student's mobile devices. See Fig 4.2 for sample query images taken by students.

Fig 4.2: Sample rock art images taken by Students at the Cederberg Region, Western Cape, South Africa
Photograph taken by S. Olaleye and Y. Olaleye

4.5 Application Prototype

The mobile application was developed using the OpenCV framework. Implementing these algorithms required a mobile application framework that can conveniently run programs written in C++. As mentioned earlier, the Android developer tools were utilized. Android provides a rich application framework that allows building innovative apps and games for mobile devices in a Java language environment **[175]**. The implementation of the algorithms is written in c++ language. To effective run the algorithm in the Android environment, the Java Native Interface (JNI) is employed. JNI will allow us to run OpenCV in the Android Framework. Three identical application prototypes were developed in OpenCV under the Android Application Framework. Prototype 1 has a SIFT implementation. Prototype 2 has a SURF implementation and Prototype 3 has an ORB implementation. The 3 prototypes were used in the comparative experiment for performance evaluation.

4.6 Design Approach

An Iterative design methodology was adopted in the design process. The iterative design method utilizes an incremental development approach and refines designs based on feedback from users. The main purpose and benefit of iterative design is that it helps to easily identify problems earlier through getting quick user feedback and it also facilitates more focus on designing and less focus on documenting **[176]**. To facilitate the iterative design process, we adopted the operational prototyping method **[177]**. Operational prototype is a combination of throwaway prototype and evolutionary prototype. The throwaway prototype involves sketches of the intended system made on paper and then evaluated by the user. This is usually useful for user interface design. Evolutionary prototyping focused mainly on developing the actual application program with a clear set of requirements. In this case, we already know what type of functionality is expected of the system so it was safe to develop a prototype to accept a query picture and find matches in an embedded database of training images.

4.7 System Design

In this section, the system design in terms of logical and physical design are described. Functionality and all additional components that were included in the prototype are also described. Additionally, the pre-processing functionality that further improves the condition of the images is also described.

Unlike most guidance systems in the cultural heritage context that make use of a client-server architecture where the feature detection and extraction process is outsourced to a powerful remote server, our approach is based on a system architecture that does not rely on the traditional client-server architecture. It is a single layer approach where all processing is done at the client side, which in this case is the mobile device.

4.7.1 Logical Design

In the logical design **[178]**, we describe a symbolic representation of the system data flow, inputs and output. The logical design follows the content based image retrieval system design (**see Section 2.2.1**) with the inclusion of the image pre-processing component. The logical representation **(see Fig 4.3)** is divided into 5 stages, each with their specific function. These stages are explained below:

INPUT: The first stage is the input stage. This stage serves as the point of entry into the system. Entry can be made through submission of a query image through the phone gallery or through the phone camera. At this stage the raw images are submitted into the system for processing.

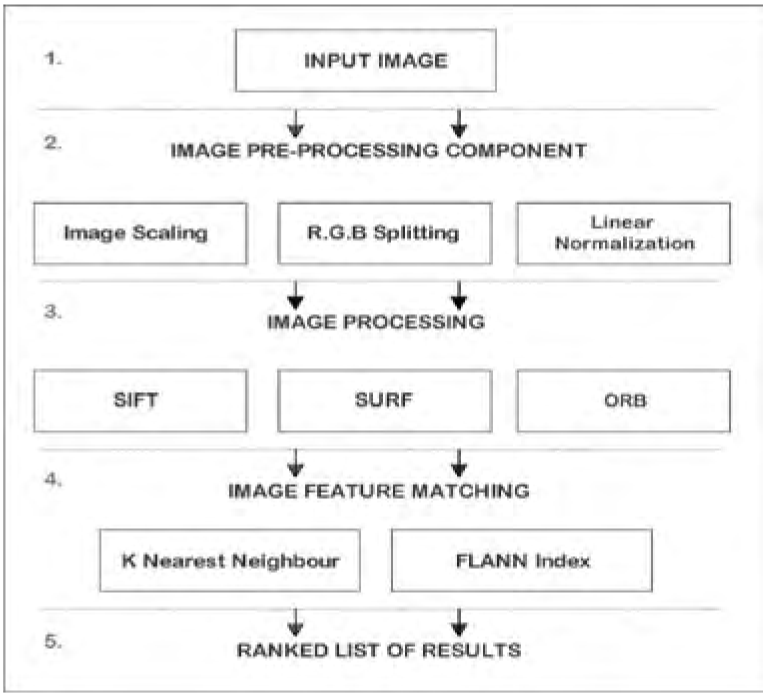


Fig 4.3: Logical representation of the system design

IMAGE PRE-PROCESSING: Image pre-processing is a set of steps carried out on the input image, usually to make the image easier to process before it is finally delivered to image processing algorithms. The output of image pre-processing is usually an enhanced or compressed image. The image pre-processing was carried out in the following steps:

1. The application normalizes the size of each image such that the final size is 125 x 93 pixels.
2. The image was split into R, G, and B channels. The Green channel was preferred as rock art paintings were more visible under this channel. For a visual reference, see **Fig 4.4**. In the image below, the Green channel output is the circled section.

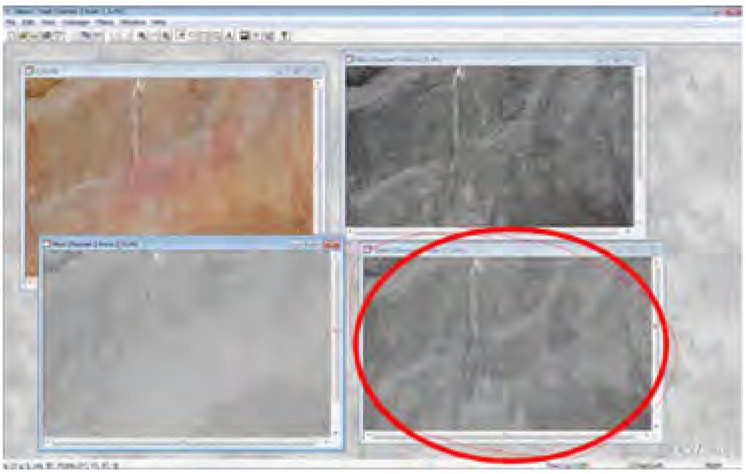


Fig 4.4. A view of the Original Image, R Channel, G Channel and B channel. The G channel is circled.

3. The application performs a contrast enhancement (linear normalization) to [0, 255] as rock art images have low contrast. The final output of the image pre-processing step on one of the rock art pictographs is illustrated below in **Fig 4.5**.

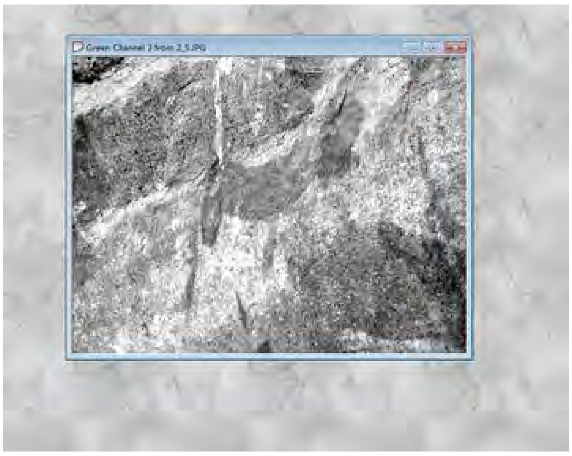


Fig 4.5. Result of linear normalization (contrast enhancement) of the green channel output

The image pre-processing step does not attempt to separate rock art objects from the background. The background also forms part of the features for the image matching.

IMAGE PROCESSING: In this stage, the enhanced image from the image pre-processing stage is passed to the object recognition algorithms (SIFT, SURF, ORB). These algorithms will extract low level features from the image. The features are converted to descriptors that are used by image matching algorithms to establish similarities between query and training images in the database.

IMAGE MATCHING: In this stage, K Nearest Neighbour was used to establish similarities between query image features and training image features in the database. For fast matching, we employed the indexing technique for fast retrieval. Indexing is a data structure technique for retrieving records from the database based on some attributes on which the indexing has been done **[179]**. The classical KD-tree was first considered for indexing these features but it was observed that KD Trees are very good for low dimensional features but performance degrades gradually as feature dimension increases **[180]**. Indexing these features requires a high-dimensional indexing technique. High-dimensional indexing techniques like Randomized KD-Tree **[181]** and Locality Sensitive Hashing **[182]** were employed. OpenCV contains the FLANN **[136]** library, which includes implementations for the indexing techniques.

OUTPUT: This displays a ranked list of matches found by the image matching algorithms. The output is ranked according to their confidence level as determined by the image matching algorithm. Users can click any item in the ranked list to see more information about the items.

4.7.2 Physical Design

In this section, the user interface design and data design is described.

4.7.2.1 User Interface Design

Georgieva et al **[47]** defined user interface design as the design of applications that focus specifically on user's interaction. The main aim of the user interface design is to facilitate the user application interactivity and to increase the effectiveness of the user's work. The user interface provides two basic options **[47]**:

- Input, through which users can control and interact with a device
- Output, which reflects the users' actions

Georgieva **[47]** identified some iterative steps necessary in the development of a mobile application user interface, which are:

- Identifying and analysing the users;
- Identifying and defining the functional and non-functional requirements;
- Development of a navigation scheme of the application;
- Prototype design in the form of simple screens shots or sketches, which include basic information – text and graphics;
- Testing the prototype with real users;
- Designing the final version of the user interface. The results of testing with real users may be taken into account. If it is necessary the architecture of the application can be changed. In some cases it may be necessary to develop different graphic templates for different screen resolutions.

4.7.2.1.1 Design Principles

From the 80's, there has been much research about designing user interfaces for computing systems **[64]**. Most of the design principles derived are mostly applicable to stationary applications but much of it can be applied to mobile applications **[64]**.

1. Simplicity: User interfaces should be simple and intuitive. The usage and the design should be optimized in a way that the application can do most things with least interference. Also designing easy to use navigation that are understandable and aesthetically pleasing is important. It is imperative that users can easily navigate or switch between screens with ease. A user should be able to easily navigate through from the first time he/she uses the application without much hassles. It is also important to optimize the number of steps that are required to complete a task.

2. Consistency: The operating environment should be consistent with the user interface and each user interface should be consistent with each other. Consistency in terms of design theme and navigation appearance is desirable **[65]**. The author also stressed about consistency with the operating environment of the application. For example, if designing on windows platform, the design should be consistent with the Windows environment.

3. Learnability: In a mobile context, learnability is the ability for the user to be able to learn the user interface within few usage times. Any form of complexity should be avoided in user interface design so that it's easy for users to learn and master the user interface screen flow and most importantly where to find information within the interface.

4. Helpfulness: A good user interface should be able to provide help to the user in a way that it does not become an impediment to the application efficiency. Help could be provided in form of hints or pop-ups.

5. Efficiency: Efficiency in this context can be defined as the amount of related information that is made available to the user when needed. Only a good user interface provides information to the users in ideal time and precision **[81] [82]**.

6. Trustable: A good user interface design should be trustable. It should be easy for a user to understand and predict the outcome of an action on the user interface.

7. Error Handling: In case of the occurrence of an error, a good user interface should display the proper pop-up messages to guide the user **[80]**. This principle goes in hand with helpfulness.

In the user interface design, we adapted some of the design principles mentioned in **[64].** We describe and illustrate how users input data to the system and how the system processes the information. We also describe how the system displays output to the user. The user interface design employs the throwaway prototype **[177]** method in an iterative process as described by Georgieva **[47]**. Sketches of the interface design were produced using Adobe Photoshop CS5. This was produced in several copies. The interface design is illustrated in **Fig 4.6** and **Fig 4.7**. It shows the application home screen, with a welcome message and 2 buttons; one to select picture from gallery and the other to take a picture. The next screen shows the algorithms to process the intended request. The fourth screen triggers either the gallery or the phone camera depending on the options chosen in screen two.

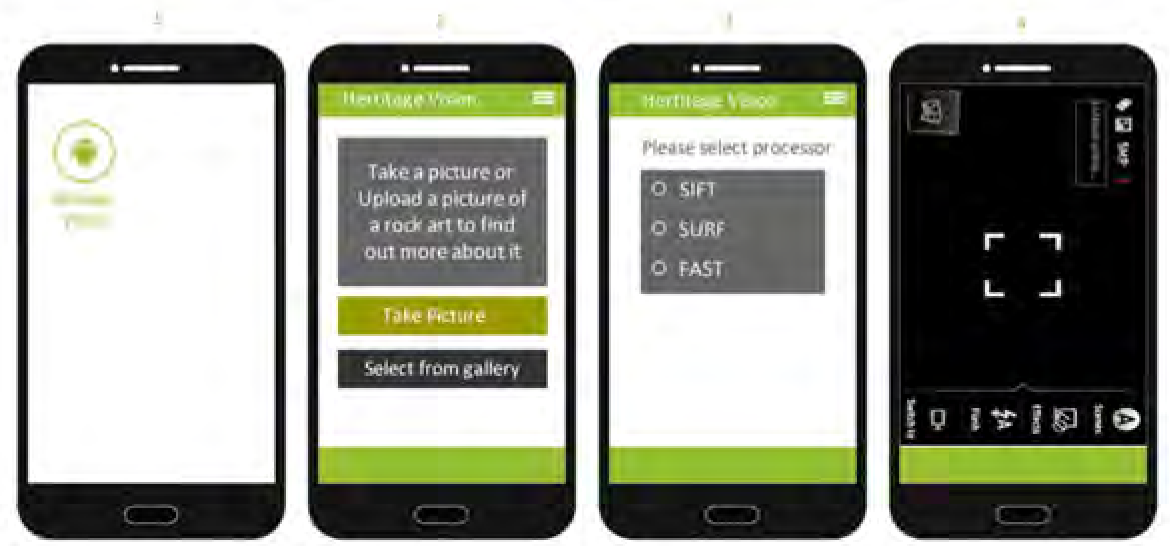


Fig 4.6: The First prototype Design (1)

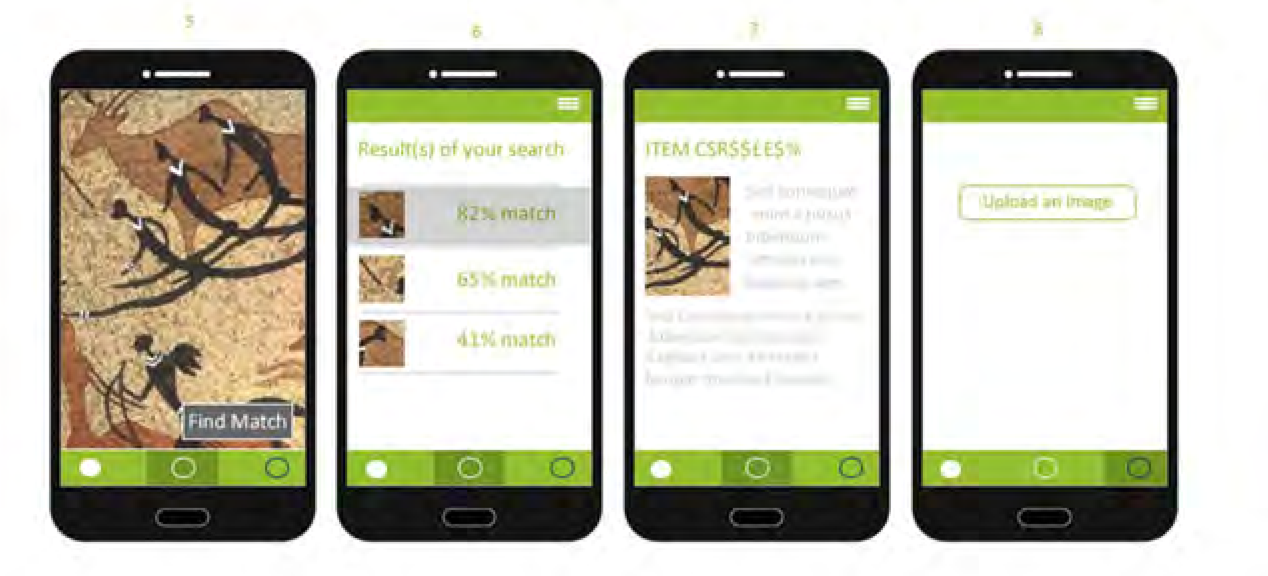
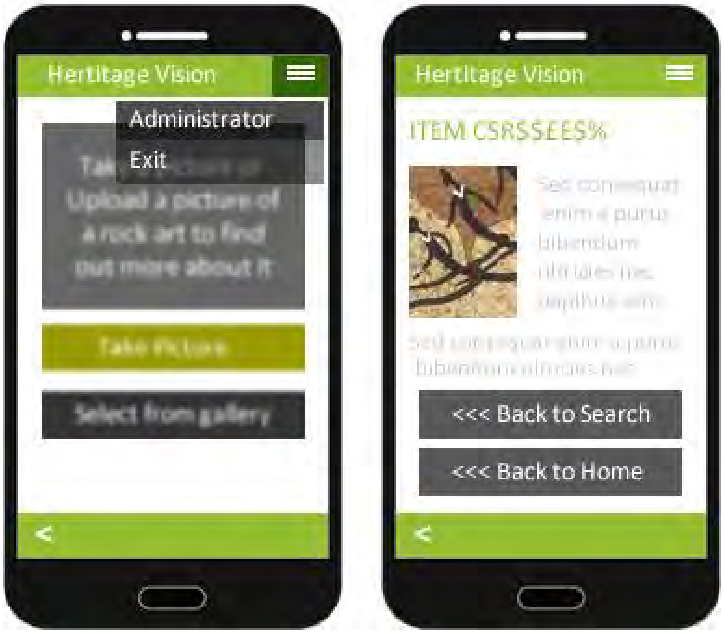


Fig 4.7: The First prototype Design (2)

In the above illustration (**Fig 4.7**), the image captured from phone camera or gallery is transferred into the application and the user is expected to click the '**Find Match**' button. The **Find Match** triggers the pre-processing stage, processing (object recognition) and image matching function described in the logical design section. These functions are executed in a sequential order. Screen 6 displays a list of possible matches ranked based on their percentage confidence level. Screen 7 displays detailed information about any of the matches in screen 6. Screen 8 has image upload features. 10 students were recruited to evaluate the interface design. Each of the students were postgraduate students from the computer science department at the University of Cape Town. The students are between the age range of 25 and 35 years. They were first informed about the idea of the application and they gave feedback based on what they felt could be improved and what they felt wasn't necessary. In general, they were all happy with the design but many of them agreed on some changes that were necessary. Some of the main points are highlighted below

1. The application icon on screen 1 could use a picture of a rock art or something depicting heritage.
2. More navigation options were needed, especially in the footer of the screen.
3. Screen 3 is not necessary as users do not understand the meaning of SIFT, SURF or ORB
4. Screen 7 can have two buttons at the bottom of the result details; a back button to go back to search result screen and a match another image button to start the process all over again.
5. An admin area can be created to facilitate upload of images to the image archive.

The feedback from the throwaway prototype was used to construct the actual user interface. Point 1 was not really relevant at this point so we kept the android app logo as the application icon. In response to point 2, the need for better navigation, various improvements were made to make user navigation easier. At the start-up screen, we added an embedded menu (see Fig 4.8). This has an admin section for uploading pictures and an exit menu for quitting the application. Also at the screen footer, we added a back button for easy exit when on the home screen and for navigating to previous screens when not on the home screen.



Screen 1 Screen 7

Fig 4.8: (1) Home screen of the second prototype Design, (7) Details screen of the second prototype

In response to point 3, it was decided to have three prototype designs running the 3 different image processing algorithms. All the three prototypes have the same user interface and screen flow logic but they just differ in the feature detection and matching algorithms. This eliminate the need for physical selection of processing algorithm.

In response to point 4, Screen 7 was modified (see Fig 4.8) to accommodate two buttons at the bottom of the screen. The first button goes back to the search result screen where users can easily select other search results to view details. The second button clears out the system log and redirects user back to the home screen where they can initiate another search.

The amended user interface was evaluated by the same group of postgraduate students. From the data gathered from the students, they were all happy with the amendments but 6 out of the students suggested that during the search process, displaying an indicator that informs users of the search progress is desirable. This was incorporated into the design by adding a progress bar layer on the screen five (see Fig 4.9) which immediately activates when users click on the "Find Match" button.

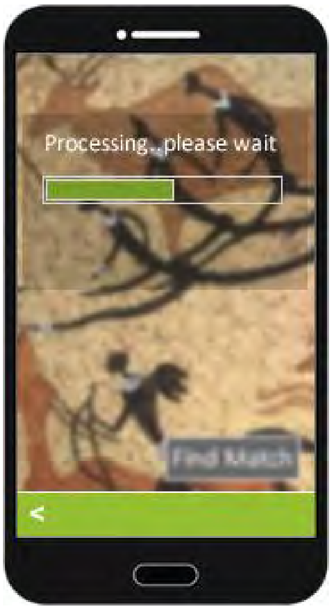


Fig 4.9: Amended screen 5

4.7.2.2 Data Design

In this section, we describe how data is represented and stored within the system. In content based image retrieval system, extracted training image features are usually stored in a database along with their corresponding image. In our approach, training images and their extracted features are stored separately. The extracted feature vectors or descriptors are stored in a database embedded (local/client storage) in the application. The corresponding images are stored in a separate folder that is also embedded in the application. The text that describes each of the images is stored in the database with reference to each corresponding image. The input images have been reduced to a size not exceeding 30 kilobytes during the image pre-processing step. This was to ensure the entire application does not require too much of physical memory space. We utilize the SQLite database.

4.7.2.2.1 SQLite

SQLite is a popular choice relational database management system as embedded database software for local/client storage in software such as mobile application. Due to the server-less design, SQLite applications require less configuration than client-server databases. The embedded nature has made SQLite the preferred database system for our mobile guide application over other RDMS such as MySQL.

4.8 Experiment Design

In the experiment design, we describe the two different experiment that were conducted. The first experiment is a **comparative experiment** that compares the performance of the 3 image processing algorithms (SIFT, SURF and ORB) in terms of precision and speed, using the same image queries, image database and mobile device. This experiment aims to confirm which of the algorithms performs better on extracting and matching rock art image features on a mobile device. The first experiment will attempt to answer research question 1 described in Section 1.3.

The second experiment is a user experience experiment that assessed usability of the mobile guide application. This experiment involves **testing with users**. This experiment is however not a primary indicator of quality but the main purpose is to evaluate the usability of the mobile guide application in the cultural heritage context. Real users were recruited for this experiment to interact with rock art artefacts and their feedback was used for usability evaluation. The second experiment will attempt to answer research question 2 described in Section 1.3.

4.8.1 First Experiment: Comparative

The comparative experiment does not require recruiting users. In order to get accurate results, the experiment was conducted using an identical apparatus for the 3 prototypes running the SIFT, SURF and ORB implementations.

4.8.1.1 Experiment Apparatus

Mobile Device: The experiment was conducted using the same mobile device. This was to ensure a fair comparison ground for all the 3 algorithms in terms of accuracy of match. Speed of matching may differ if using a different device with different configuration. For this experiment, we made use of one Samsung Galaxy s6 Android mobile device with the below configuration.

- Android OS, v5.0.2 (Lollipop)
- Quad-core 2.1 GHz Cortex-A57
- 5.7 inches TFT capacitive touchscreen
- 3GB RAM
- 32GB

Image Database: The image database was preloaded with a total of 460 pictograph images. The images consist of a total of 235 different pictographs that consist of humans(187), hand prints(5), elands(34), sheep(2), bow & stick(5), giraffe(1) and elephant(1). Each image is scaled down to a size of 125 by 93 pixels. 112 of those images have different views. As at the time of this experiment, this was the dataset we had access to. Other rock art site were too far away and were not easily accessible.

Query Images: A total of 26 images was used as query images. These were selected from the pictograph pictures taken by 2 students (**see Section 4.4.2**) who accompanied us to the rock art site. The 26 images were unique (**see Appendix D**). We used only 26 images out of the 32 images because the remaining 6 query images are duplicates. Each of the selected query images has at least one reference image in the image database but differs in quality, viewpoint and light conditions.

Application prototype: In this experiment, 3 nearly identical prototypes were used with each running SIFT, SURF and ORB respectively. Because we are limited to just one mobile device, the prototypes were installed and used consecutively.

4.8.1.2 Experiment Mode

The experiment was conducted consecutively since we had only one mobile device to conduct the experiment. In order to prevent confusion and conflicts, one prototype was installed on the mobile device at a time. After running each of the 26 queries and results have been obtained, the prototype was uninstalled and the next prototype was installed.

4.8.1.3 Collection of Results

The application has been programmed to collect information about the matching output and the time it takes to find a match. On completion of each query, the system generates a text file that contains information about time taken to successfully execute a query and the outputs of the query. The query results are sorted according to the percentage confidence in descending order.

4.8.1.4 Analysis: Performance

Performance in terms of accuracy of retrieval was evaluated based on precision and recall (**see Section 5.2.1**). Precision is the fraction of the documents retrieved that are relevant to the user's information need while Recall defines the fraction of the relevant documents that were found **[184]**. In information retrieval, precision and recall are standard metrics for measuring search efficiency **[184].**

Average precision (AP) for a set of queries will also be calculated. **Average precision** is the un-interpolated average of precision values at all ranks where relevant documents are found. The average precision is calculated as:

$$\text{AveP} = \frac{\sum_{k=1}^{n}(P(k) \times \text{rel}(k))}{\text{number of relevant documents}}$$

[15]

Where **n** is the total number of ranked images, P (k) is the precision at cut-off k in the list, and rel (k) is a binary relevance indicator (0 for an incorrectly ranked image and 1 for a correctly ranked image) **[185]**. To evaluate the speed, the average time in milliseconds for the application to complete the feature detection and extraction process of each algorithm was logged in a text file. The application generates a result file in text format each time a we run a query. The result text file contains a summary of the query and the results. The total time in milliseconds is also recorded in this file. Analysis of speed is tabulated in **Table 5.2**.

4.8.2 Second Experiment: Tests With Users

As part of the evaluation exercise of the mobile guide application, this section describes the experiment that was conducted to evaluate user experience in terms of usability. According to ISO 9241, usability is the ability of a software product to be used to achieve specific task with efficiency and satisfaction usually within its operational environment **[187] [188]**. This can also be referred to as 'quality-in-use' **[187]**. A usability evaluation technique that is reliable and involves users but still requires limited resources is desirable. SUMI (Software Usability Measurement Inventory) was employed as the usability evaluation technique.

4.8.2.1 Interactive Experiment

As part of the usability evaluation exercise, an **interactive experiment** was first conducted. This is a user oriented experiment. Users (**Section 4.8.2.4**) were given mobile devices (**Section 4.8.2.3**) to interact with the rock art artefacts in a rock art simulated environment (**Section 4.8.2.2**). The mobile devices were pre-installed with the mobile guide prototype running the algorithm with the preferred performance from the comparative experiment (**Section 4.8.1**). After the **interactive experiment**, users were given SUMI questionnaire to evaluate the mobile guide application. The purpose is to determine if an image based mobile guide application can be used to support the provision of information about artefacts in a rock art site.

4.8.2.2 Experiment Venue

Rock art sites are located many miles away from civilization, which poses a transportation/distance problem. In view of this, the usability experiment was conducted in a rock art simulated environment. The experiment was conducted in December 2015 in Cape Town at Green Elephant Backpackers. The building is a 2 storey. The building has about 20 rooms. Appropriate authorization was obtained from the owners. We were given authorization to use 5 rooms. The rooms have dark and light illuminated areas.

4.8.2.3 Experiment Apparatus

Mobile Device: The experiment was conducted using 10 mobile devices of the same specification. For this experiment, we made use of the Samsung Galaxy pocket Android mobile device with the below configuration.

- OS: Android OS, v2.3 (Gingerbread)
- CPU: 832 MHz ARM 11
- SIZE: 2.8 inches screen
- TYPE: TFT capacitive touchscreen

The appropriate permission was obtained from the Department of Computer Science for using the phones for experiment.

Query Images: A total of 10 images was used as query images. These were selected from the pictograph pictures taken by 2 students (see Section 4.4.2) who accompanied us to the rock art site. Due to limitation of 5 rooms in the experiment venue, we only used 10 query images. Each of the image was printed on A4 size paper and placed on the walls of 5 different rooms in the experiment venue. The 10 images (see Appendix E) were unique and selected at random. Each of the query image has at least one corresponding image in the image database. The same image database used in the first experiment was also used in the second experiment

Application prototype: 10 Android mobile phones were provided for the purpose of usability evaluation. These mobile phones had the final mobile guide application prototype pre-installed. The size of the application file is 35 megabytes and requires at least 80 megabytes of memory space to install the application successfully. It was ensured that the mobile devices had at least 100MB of free disk space.

4.8.2.4 Participants

A total of 30 people were recruited to participate in the usability experiment. This includes a combination of students, researchers and tourists. A voluntary request for participation was sent to every occupant in Green Elephant. There were more than 30 participants who indicated interest in participating in the experiment. Participants were selected randomly based on their availability at the time the experiment was conducted. Each participants was given R40 as an incentive to participate in the study.

4.8.2.5 Experiment mode

Before the experiments took place, a short tutorial session for participants was conducted to describe the purpose of the intended study and how the application works. All participants were given a task description and its relevance to the purpose of study. A research assistant was also hired for the experiment. The experiment was conducted in 3 phases. The first phase was the tutorial phase, which ran for about 15 to 20mins. The second phase was the interactive experiment, where participants were given mobile phones (**Section 4.8.2.3**) to interact with the rock art pictures placed on the walls of different rooms of the experiment venue. Experiment was conducted during day time. Participants were divided into 3 groups of 10 participants each. The third phase is the evaluation exercise where participants were given the SUMI questionnaire to evaluate the usability of the application.

4.8.2.6 Application Prototype flow

Like most Android applications, the mobile application prototype does not involve any form of authentication for access. It was designed with simplicity in mind. The matching task can be completed in 5 steps; select input, find match, view search results, view details and back to match results/find another match. The application is designed to run independently without Internet support.

4.8.2.7 Collection of Results

Collection of experiment results was done in the following ways:

4.8.2.7.1 Analysis: Performance

During the **interactive experiment**, the participants were observed on how well they interact with the application. The observation paid more attention to the time and output. The application has been programmed to collect information about time and matching output. On completion of each query, the system generates a text file that contains information about time taken to successfully execute a query and the outputs of the query. The query results are sorted according to the percentage confidence in descending order. The performance in terms of accuracy was evaluated using precision and recall (**see Section 5.2.2**)

4.8.2.7.2 Analysis: Usability

The evaluation questionnaire (see Appendix C) was given to each participant to gain their feedback about the functionality and usability of the application (**see Section 5.3**). The Software Usability Measurement Inventory is the usability evaluation (SUMI) framework employed. SUMI is a questionnaire based method developed by the European ESPIRIT project called MUSiC **[183]**. SUMI evaluates the quality of a software product from the user perspective. Experimenting with SUMI requires at least 10 - 12 users for an analysis with tolerable precision **[183**. Based on the answers the users provided via the questionnaire, the usability scores are calculated for each of the SUMI subscales **(see table 5.4)**. SUMI can be used to evaluate any kind of application as long as the application provides user input by means of keyboard, mouse or pointing devices **[186]**. Also, SUMI requires either a prototype or working version of a particular program. SUMI is thus suitable to evaluate the usability of the mobile software. There are over 2000 different kinds of application usability profiles that are embedded in the SUMI reference database that makes SUMI an effective analysis and report generation tool. SUMI is a questionnaire based method derived using psychometric practice. SUMI questionnaire consist of 50 items that are answered with "agree", "undecided" or "disagree". SUMI incorporates five empirically defined sub-scales and a global usability figure **[186]**. Sub scales include the following:

- Efficiency: measures the product use in a cost effective and timely manner
- Affect: measures the users instinctive or intuitive feeling about the product
- Helpfulness: measures how well the product assist the user
- Control: measures how well does the user feel in control of the product
- Learnability: measures if the user is able to easily learn how to use the product features

4.9 Summary

In this chapter, the system design and experiment design is discussed. In the system design, we describe the physical and logical design. In the physical design, we described principles that guided the design of the user interface. The user interface design was also human centered. We showed how each stage of the interface design was evaluated using users' feedback to arrive at the final design. In the logical design, we describe the logical flow of processing in the application. The experiment design was also described. We described the two experiments that was conducted and we also explained the evaluation methods employed for performance and usability.

Chapter 5

EVALUATION

5.1 Introduction

Evaluation methods of image processing algorithms are usually based on comparison of the performance of one image processing algorithm over the other using the same image datasets. This has led to different inferences where, in one publication, an algorithm is presented with very good performance and, in other publications, the same algorithm is said to have underperformed **[189]**. It is believed that some algorithms are best suited to a particular type of image and that they will perform better when tested on these images. In this chapter, the results of the experiments are presented and discussed. The image retrieval results are tabulated in Table 5.1 with precision and recall values. The usability evaluation results are also discussed. The five SUMI (Software Usability Measurement Inventory) subscales are discussed. Table 5.2 provides a summarized output of the average scores of the SUMI scales. The chapter is then concluded with suggestions as to possible improvements for the mobile application by the users via the free-form statements.

5.2 Evaluation of Accuracy and Speed

5.2.1 Comparative Evaluation

The performance of the algorithms in terms of accuracy (see Table 5.1) was evaluated based on Precision (how many returned documents are relevant) and Recall (what fraction of the relevant documents were found). The query images used can be found in **Appendix D**

Table 5.1. Shows the precision and recall of the query set for each algorithm at cut-off k = 10. It also shows the Average precision.

	SIFT		SURF		ORB	
Query	Precision	Recall	Precision	Recall	Precision	Recall
Image 1	0.7	0.87	0.3	0.37	0.1	0.12
Image 2	0.5	0.5	0.3	0.3	0.1	0.1
Image 3	0.6	0.35	0.3	0.17	0.3	0.17
Image 4	0.9	0.81	0.8	0.72	0.4	0.36
Image 5	0.9	0.25	0.5	0.14	0.8	0.23

Image 6	0.8	0.8	0.4	0.4	0.2	0.2
Image 7	0.7	1	0.4	0.57	0.3	0.42
Image 8	0.6	0.26	0.5	0.21	0.6	0.26
Image 9	0.7	0.63	0.6	0.54	0.7	0.63
Image 10	0.5	0.83	0.3	0.5	0.2	0.33
Image 11	0.4	0.8	0.2	0.4	0.3	0.6
Image 12	0.3	0.75	0.3	0.75	0.2	0.5
Image 13	0.2	1	0.1	0.5	0.2	1
Image 14	0.7	0.7	0.4	0.4	0.6	0.6
Image 15	0.2	1	0.2	1	0.2	1
Image 16	0.4	0.8	0.1	0.2	0.3	0.6
Image 17	0.5	1	0.3	0.6	0.4	0.8
Image 18	0.5	1	0.4	0.8	0.4	0.8
Image 19	0.2	1	0.2	1	0.1	0.5
Image 20	0.1	1	0	0	0.1	1
Image 21	0.3	1	0.2	0.6	0.2	0.6
Image 22	0.3	1	0.2	0.6	0.1	0.6
Image 23	0.2	1	0.1	0.5	0.2	1
Image 24	0.1	1	0	0	0.1	1
Image 25	0.1	1	0.1	1	0.1	1
Image 26	0.2	1	0	0	0.1	0.5
A.PRECISION	**0.47**		**0.27**		**0.28**	

From **Table 5.1** above, it is clear that SIFT (47%) outperforms SURF (27%) and ORB (28%) in terms of accuracy. The SIFT algorithm has demonstrated competence in all the queries when used with Fast Approximate nearest neighbour search (FLANN). In terms of speed, an indexing technique was employed. With the time constraints, only KDTreeIndexParam and Multi-probe LshIndexParam of the FLANN library were implemented. For high dimensional features like SIFT and SURF, FLANN's KDTreeIndexParam was used with Euclidean distance. Locality Sensitive Hashing is particularly good with binary descriptors so this was employed for indexing ORB feature descriptors with Hamming distance **[182]**. Hamming distance was used because several issues and errors were encountered with using the Euclidean distance for matching ORB binary descriptors. Table 5.2 shows the application performance in terms of retrieval speed.

Table 5.2. Total time taken for application to perform search

	SIFT	*SURF*	*ORB*
Query	Time to Match (ms)	Time to Match (ms)	Time to Match (ms)
Image 1	1622	945	112
Image 2	1613	1232	87
Image 3	1532	1022	93
Image 4	1554	832	192
Image 5	1366	922	67
Image 6	1701	924	122
Image 7	1201	965	153
Image 8	1211	1023	164
Image 9	996	943	165
Image 10	974	966	183
Image 11	983	503	31
Image 12	1875	779	57
Image 13	796	488	30
Image 14	1645	688	57
Image 15	1610	698	77
Image 16	915	463	38
Image 17	910	494	28
Image 18	2736	1165	110
Image 19	1002	402	24
Image 20	1090	476	47
Image 21	1002	1001	76
Image 22	1241	969	161
Image 23	1112	941	164
Image 24	989	764	59
Image 25	1044	981	121
Image 26	2022	1021	211
Average Time (ms)	**1336**	**831**	**101**

From the result in **Table 5.2**, The FLANN matcher spent more time matching features from SIFT and this may be because of SIFT 128 dimension feature vector. This is higher than SURF's 64 dimension feature vector and ORB's binary features. Matching of ORB features performed better in terms of speed than SIFT and SURF but it has the lowest accuracy. We also believe that the processing speed of the mobile device used to conduct the experiment may have influenced the matching speed. This suggest that the speed might differ when used with a device with low processing capacity. We selected the **ORB implementation** as the prototype for interactive and usability evaluation (Tests with Users) because of it retrieval speed which is a major requirement for mobile based applications.

5.2.2 Interactive Evaluation (Tests with Users)

The performance of the **ORB implementation** used for the usability evaluation was analysed using standard precision and recall. Precision and recall were taken across 10 different queries submitted by each participant. The average precision values were calculated as well. The average speed of retrieval was also analysed. The precision and recall analysis is tabulated in **Table 5.3** below.

Table 5.3 Precision and recall taken across 10 queries for all participants (ORB Algorithm).

QUERY	*WHEN K = 10*		*WHEN K = 5*	
	PRECISON	**RECALL**	**PRECISION**	**RECALL**
1	0.312	0.294	0.544	0.2596
2	0.288	0.2628	0.504	0.234
3	0.324	0.3172	0.592	0.29
4	0.304	0.3444	0.584	0.3288
5	0.264	0.2508	0.52	0.2508
6	0.328	0.304	0.568	0.2688
7	0.276	0.2948	0.504	0.268
8	0.32	0.338	0.64	0.338
9	0.372	0.3932	0.688	0.3552
10	0.304	0.316	0.592	0.3092
Avg. Precision	**0.3092**		**0.5736**	

Table 5.3 shows the precision and recall taken across an average of 10 queries. At cut off K = 10, the average precision is approximately 31% and this is because the majority of the queries only returned good matches at the top of the result list. The poor results could be attributed to poor query images taken by the participants, which resulted in detection of few correct matches. **Fig 5.1** and **5.2** shows

some of the query images that were generated by the user's mobile devices. The query images in Fig 5.1 were poorly taken and this may be because of poor light conditions in certain areas in the experiment venue. There were also some query images that appeared blurred. At cut off k = 5, there was better precision (57%) due to high concentration of true positive matches at the top. This shows that an increase in the value of K will have a negative impact on the precision. A further decrease in the value of K will probably give a much higher average precision. In terms of speed, the average speed across 10 queries was about **11 seconds**. The result might vary depending on the mobile device processing capacity, as seen during the prototype development stage when a high power mobile device was used to carry out testing on the same dataset and an average speed of one second was recorded.

Fig 5.1. Poor quality query image

Fig 5.2. Acceptable quality query image

5.3 Usability Evaluation using SUMI (Tests with Users)

As mentioned in the experiment design, SUMI combines a global usability figure in conjunction with five different subscales on which the participant's questionnaire responses were evaluated. The strength rating from the strength and weakness analysis (**see Appendix A**) is a probability rating and is measured as a percentage value. The strength statistic shows the probability that the participants' responses to a specific SUMI question differed overall from chance. The strength statistics is based on the chi squared distribution. In contrast, SUMI Global scale and the other 5 subscale metric scores represent points on a scale and therefore aren't represented as percentages. Table 5.4 represents SUMI subscale score for each participant extracted from the SUMI questionnaire based on the mobile application usability.

Table 5.4: SUMI subscale scores for each participants measured in scale of 0 to 100

Participant	Global	Efficiency	Affect	Helpfulness	Control	Learnability
1	72	72	67	68	67	66
2	71	69	66	65	67	69
3	70	68	72	70	68	71
4	70	72	66	66	56	64
5	69	72	57	65	66	66
6	69	72	67	66	60	69
7	68	64	53	68	65	71
8	67	66	72	65	60	56
9	67	58	66	68	60	71
10	67	68	72	61	69	69
11	66	69	69	64	48	64
12	66	64	59	62	61	68
13	66	68	64	64	59	60
14	65	68	61	58	64	64
15	64	66	61	65	58	52
16	64	63	48	70	60	67
17	64	66	61	68	47	66
18	63	69	56	64	53	62
19	62	65	58	67	57	65
20	61	60	63	56	64	62
21	61	58	63	60	45	71
22	60	63	46	64	62	69
23	60	72	65	52	56	58
24	60	51	61	61	47	63
25	59	72	45	53	66	59
26	58	60	57	58	60	61
27	56	51	42	60	51	58
28	51	56	48	54	47	63
29	47	50	47	49	40	45
30	45	36	53	47	43	42

5.3.1 Analysis of SUMI Scales

The results from the SUMI evaluations of the mobile application for each participant are presented in Table 5.4 in terms of the global, efficiency, affect, helpfulness, control and learnability subscales. Table 5.5 represents the summary subscale statistics presented in terms of mean, standard deviation, median, inter-quartile range, minimum and maximum levels. These levels are derived from the global usability scale and each of the five usability sub-scales.

Table 5.5: SUMI sub scale summary statistics

	Mean	St Dev	Median	IQR	Minimum	Maximum
Global	62.93	6.62	64.0	7.0	45	72
Efficiency	63.60	8.34	66.0	9.0	36	72
Affect	59.50	8.54	61.0	13.0	42	72
Helpfulness	61.93	6.15	64.0	8.0	47	70
Controllability	57.53	8.17	60.0	13.0	40	69
Learnability	63.03	7.17	64.0	9.0	42	71

Each subscale measure is discussed in the sub sections below

5.3.1.1 Global Scale

The Global scale focuses on a general usability factor. The Global scale represent 25 questionnaire item out of the SUMI 50 questionnaire items. The Global scale represent a single construct of perceived quality of use **[190]**. In the standard normal distribution, the properties indicate that the score of most software product on the SUMI scales will fall within one standard deviation of the mean, which lies between 40 and 60 **[190]**. This implies that software products ranked below these points are below average and those ranked above these points are above average. The mobile application scored above the global average for SUMI. With the global scale score of 62.93, it was safe to conclude that the mobile application shows an acceptable degree of quality of use.

5.3.1.2 Efficiency

The SUMI efficiency sub scale represents the software use in a cost effective and timely manner. The efficiency subscale has a score of 63.60, which is above the SUMI required average. The SUMI strength and weakness analysis in Appendix A shows certain SUMI questions that address efficiency of a system. Some of the questions for evaluation of efficiency of the mobile application include:

#28: "The software has helped me overcome any problems I have had in using it". This question has an overall score of 84.9%

#39: "It is easy to make the software do exactly what you want". A score of 99.9 % was attained. These responses reflect acceptable scores for efficiency in using the mobile application.

5.3.1.3 Affect

Affect relates to the extent to which the product captures the user's instinctive or intuitive feeling. The affect metric score is compiled via the questionnaire responses on certain questions related to emotional response. The questions pose a variety of emotions ranging from satisfaction to mental stimulation, frustration, tension and awkwardness. A few examples of the emotional response questions include the following:

#12. Working with this software is satisfying

#17. Working with this software is mentally stimulating

#27. Using this software is frustrating

#32. There have been times in using this software when I have felt quite tense

#47. This software is really very awkward

#31. It is obvious that users' need has been taken into consideration

All the above receive positive responses with the exception of question #17, which had a result of slightly agree (19.4%). This may be due to the fact that about 45% of the participants found the software unimportant or irrelevant (**see Table 5.6**). The SUMI question "*How important for you is the kind of software you have just been rating*" was used to measure the level of relevance the application has to the participants.

Table 5.6: How important for you is the kind of software you have just been rating?

Scale	n
Extremely important	3
Important	17
Not very important	8
Not important at all	2

In **Table 5.6**, 3 participants found the application extremely important. 17 participants found the software important. 8 participants found the software not very important. 2 participants found the software irrelevant to them.

5.3.1.4 Helpfulness

The Helpfulness subscale is measures how well the product is able to assist the user. With regards to this study, helpfulness can be of two types. First, it can refer to the program's ability to assist the user by providing adequate help on a certain program feature. Secondly, it can refer to the ability of the mobile application to help the user learn about the functionalities.

Help features can include either context sensitive help (also known as the popup message, which appears or gives direction at every step of application usage) or the program's built-in help function, which lists all available features the program supports and how to use each feature.

Question #15: "*The software documentation is very informative*" of the strength and weakness analysis (Appendix A) shows a strong rating (95.8%) of the quality of documentation and information regarding how to use the program and complete tasks. Figure 5.3 shows the home screen of application with the required information to get started.

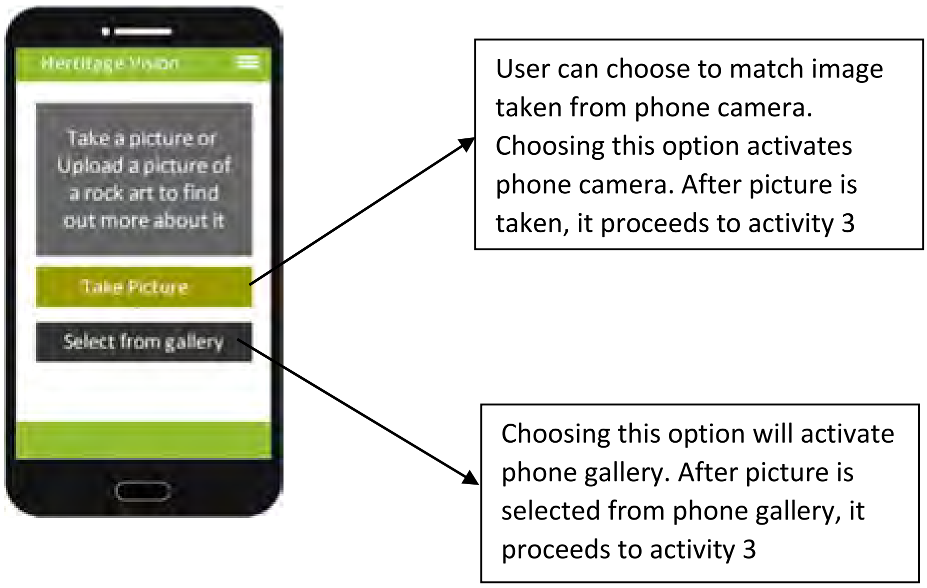


Fig 5.3. Home screen of App

Furthermore, participants' response to question # 8: "*I find that the help information given by the software is not very useful*" of the strength and weakness analysis (see Appendix A) further supports how helpful the system was with a verdict of Disagree. The verdict has a score of 86.4%.

5.3.1.5 Controllability

Controllability in the SUMI subscale measures the extent to which a user feels in control of the software product. With regard to question #19: “I feel in command of this software when I am using it” in the strength and weakness analysis (see Appendix A), the results indicate that the participants felt in complete control of the mobile application with a score of 88.1% agreeing.

5.3.1.6 Learnability

Learnability measures the extent to which the user is able to easily learn how to use features of the software product. With regard to question #5:“Learning to operate this software initially is full of problems”, the users disagreed with this and the score was about 99.3. Other related questions in the strength and weakness analysis (see Appendix A) about learnability of the mobile application follow:

#10:“It takes too long to learn the software functions” (Disagree = 85.9%)

#35:“Learning how to use functions is difficult” (Disagree = 96.7)

#40:“I will never learn to use all that is offered in this software” (Disagree = 94.3)

This goes to show the degree to which users were able to easily get started with the application and learn its features. During the experiment, most users didn’t have problems using the application as there was nobody who came back during the experiment to complain about functionality of the mobile application itself. This is because the application is self-explanatory and give users options at every point; this improvement was made after the initial analysis of the first prototype. The summary shows that the application has an average acceptability in terms of quality of use. Also, there are outliers in efficiency, helpfulness and learnability subscales. This can be traced to two specific participants. **Table 5.7** shows the outliers with the subscales and their relative scores.

Table 5.7: Outliers

PARTICIPANT	SCALE	SCORE
29	Global	47
	Learnability	45
30	Efficiency	36
	Learnability	42
	Global	45

5.3.2 Differences in behaviour

There were factors that may influence participant behaviour. The SUMI question "***How would you rate your software skills and knowledge***" helped to analyse participant behaviour based on their level of experience with software usage. **Table 5.8** shows the participants' software experience categorized into 3 levels. The first category is the category of participants that are very experienced and technical and the category represents 23% of the total participants. The global subscale of this category is 62.4 which is well above the required SUMI average for quality of use of a software system. The second category are participants with good software experience but not technical. This category represent 36% of the total participants. The global SUMI sub scale for this category is 62.7 which is also well above the SUMI average for quality of use. Other subscales for this category of participants were also above average. The third category are participants who can cope with most software. This category contains the highest number of participants. This category represents 40% of the total participants. The SUMI global scale score for this category was 63.4. This is also above the SUMI average requirement for quality of use.

Table 5.8: How would you rate your software skills and knowledge?

	n	G	E	A	H	C	L
Very experienced and technical	7	62.4	65.9	58.3	62.0	59.7	65.3
I'm good but not very technical	11	62.7	62.4	60.4	61.5	56.6	64.0
I can cope with most software	12	63.4	63.4	59.4	62.2	57.1	60.8

n: number of users, **G**: global, **E**: efficiency, **A**: Affect, **H**: Helpfulness, **C**: Control, **L:** Learnability

With the above analysis, we could determine participant background in software usage and participant background relevance to the context of use of the mobile application they evaluated. Interestingly, their experience level with using software didn't have any negative impact on their behaviour with the use of the mobile application which they evaluated. All three categories of participants had a global scale score above the required SUMI average for quality of use of a software application. With these results and analysis, we could conclude that the level of experience of the participants with softwares didn't have any negative effect on their behaviour during the evaluation of the mobile application.

5.3.3 Summary of best aspects of the program

Table 5.9 is a thematic breakdown of the free-form question on the best aspects of the mobile guide application as identified by the users. We broke down the user responses to the free form question (see Appendix B: Question 2) into 4 categories and each user response is mapped to a category. The thematic breakdown is further analysed in Fig 5.4. Please see below:

Table 5.9: Best aspect of program category

Partici pant	Ease of Use	Image Matching	Informati ve	General
1			1	
2	1			
3	1	1		
4				
5	1			
6	1			
7	1		1	
8			1	
9		1		
10	1			
11	1			
12		1		
13	1		1	
14			1	
15	1			
16				
17	1			
18	1	1	1	
19				
20	1		1	
21				
22	1			
23	1			
24		1		
25				1
26			1	
27	1			1
28				
29		1	1	
30				
TOTAL	15	6	9	2

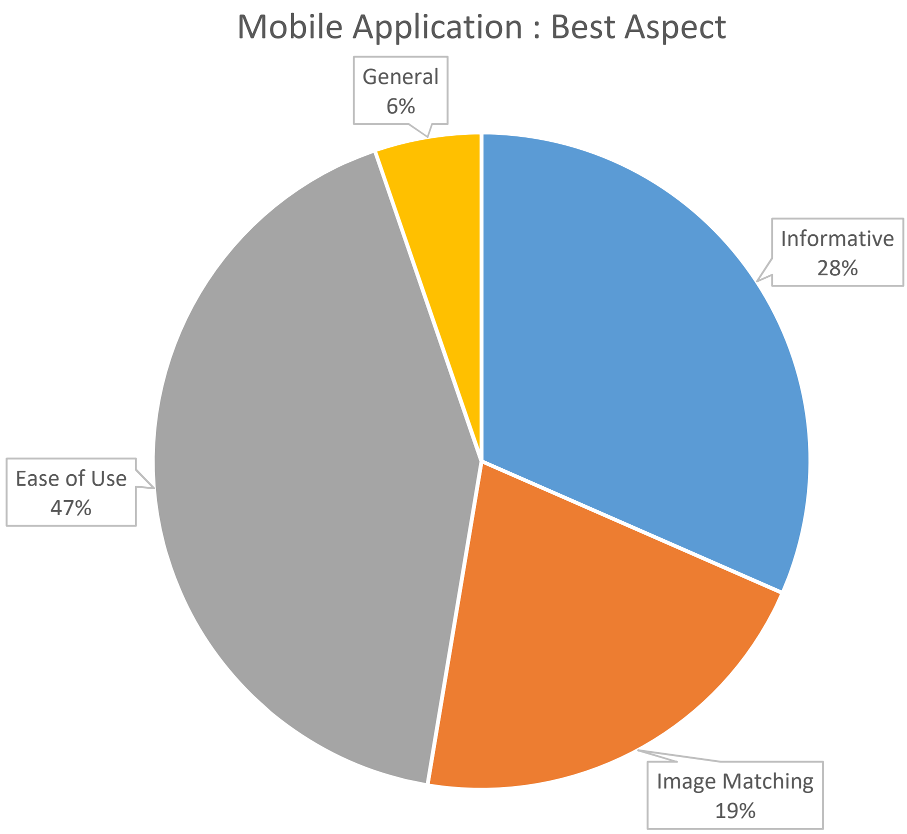


Fig. 5.4 *Best aspect of the mobile application as identified by users*

From Fig 5.4, 47% of the responses indicated that the best part of the heritage vision mobile application is the ease of usage. 28% of the participants were of the opinion that the informative aspect of the application, including information about rock art pictures taken, is the best aspect. 19% were of the opinion that the best aspect of the application is image matching such that they can take rock art pictures and the application can produce a ranked list of similar matches. 6% of the users are of the opinion that the application was good all round.

5.3.4 Suggested improvements and recommendations

The users listed possible improvements to the mobile application. These improvements form part of the free-form statements the users answered on the SUMI questionnaire. The question: "What do you think needs improvement and why?"(See Appendix B: Question 3) was issued to the users and they responded with suggestions. All responses were compiled for this question and illustrated to highlight the key differences between the suggested improvements.

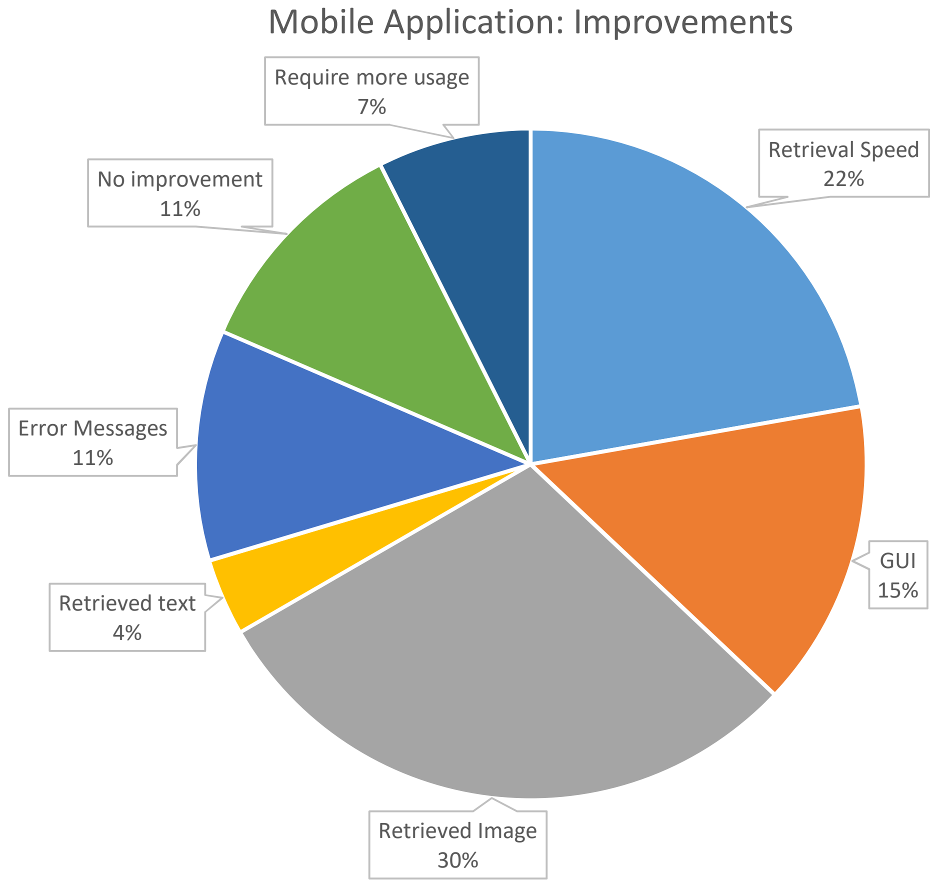


Fig 5.5. Improvements and recommendations suggested by the users

From Fig 5.5, 11% percent of the users prefer the application the way it is. About 7% of the users were of the opinion that a one time usage of the application wasn't enough to enable them to give recommendations and improvements.22% of the users made comments on the speed at which the application retrieves results. This could be because of the quality of the mobile phone used for the interactive experiment. 15% of the users suggested the GUI can be improved upon. They didn't state in particular what they want improved in the GUI. 30% of the users believe there could be improvement in the presentation of the retrieved images. Most of the participants complained that the size of the retrieved images was so small and it was sometimes difficult to tell if a retrieved image is a match or not. This could be because of the small screen resolution of the mobile devices used to carry out the evaluation. 4% of the users also didn't like the presentation of the descriptive text accompanying the retrieved results. This can also be because of the screen resolution of the mobile phone used to conduct the experiment.

In terms of error messages, 11% of the users weren't comfortable with the way errors are handled. For instance, one of the users complained that when a query is submitted for search and it couldn't find any similar result, an appropriate message should be displayed.

Finally, two users gave new ***recommendations*** that could be possible add-ons to the application. The first is support for people with disabilities to be able to use the application. Another user recommended including voice communication to make it more interactive.

5.4 Summary

In this chapter, we presented the results of the comparative experiment. The comparative experiment is the primary indicator of quality. The precision and recall values at cut-off k =10, shows that SIFT is better in accuracy. In terms of retrieval speed, ORB performed better, having an average speed of less than one second. In the interactive experiment using the ORB implementation on a low resource mobile device, results shows a precision of approximately 31% when the value of K=10, but when the value of K = 5 the precision value increased to approximately 57%. This shows that an increase in the value of cut-off K will have a negative effect on the precision. This is because ORB returned relevant matches at the top of the list.

SUMI (software usability measurement inventory) was the standard adopted for usability evaluation of the mobile application. Next, the SUMI global scale and sub scales scores were presented and this was used to generate the SUMI summary statistics table. Looking at the summary statistics table (Table 5.5), one can see that the users rated the application above the standardization average score in the SUMI database in both the global scale (general perception about the quality of use) and the other SUMI 5 subscales (Learnability, Controllability, Helpfulness, Affect and Efficiency). This shows that the application was perceived to be acceptable in terms of the quality of use. Also, we were able to conclude that the difference in experience of software usage of the participants didn't have any negative impact on their behaviour during the mobile application evaluation.

Chapter 6

CONCLUSION

The goal of this research was to determine whether an image based mobile guide can be used to provide information in a rock art heritage site. With the aid of a content based image retrieval system, we investigated the performance of the SIFT, SURF and ORB image processing algorithms for digital recognition of rock art. Highly distinctive local features of rock art were identified and extracted for image matching. An image pre-processing technique was adopted to enhance the rock art image before processing. This technique involves scale reduction, splitting of the image into RGB channels and applying a linear normalization to increase contrast. This process generalized to most of the images in our training set. FLANN (fast library for approximate nearest neighbour) was used in place of the classic nearest neighbour image matching technique for fast retrieval of images. The randomized KD Tree technique in the FLANN library was used for indexing the SIFT and SURF high dimensional features. The locality sensitive hashing technique (LSH) in the FLANN library was used for indexing the binary features of the ORB algorithm. LSH is known to be particularly good with binary feature descriptors. LSH was used in conjunction with Hamming distance because several errors were encountered when used with Euclidean distance. A comparative experiment was performed to compare the performance of the 3 image processing algorithms using mobile devices with the same configuration and same training set and query set. Also a user experiment was conducted to evaluate the usability of the software application. The nearest rock art site to Cape Town is about 3 hours drive away, so experiments were conducted in a rock art simulated environment. In the rock art simulated environment, pictures of rock art were placed on the wall at different locations in the experiment venue. With this completed, we are now able to answer the research questions.

6.1 Research Questions

In summary, the research questions were resolved as follows:

1. **How do the SIFT, SURF and ORB algorithms compare in terms of accuracy and speed of similarity search for rock art images on mobile devices?**

   The comparative experiment confirm that the SIFT algorithm performed better than the SURF and ORB algorithms in terms of accuracy. However, performance of the SIFT algorithm in terms of speed was lower than SURF and ORB, especially on low resource mobile devices. The SIFT algorithm appears to be the preferred image processing algorithm for recognizing rock art images as its key points and descriptors are good to produce acceptable results that are relevant to the user query. However, the SIFT algorithm has the lowest retrieval speed but was still acceptable when used with a high resource mobile device. SURF's performance in terms of accuracy was below the accuracy of SIFT and ORB but has a better retrieval speed than the SIFT algorithm. The ORB algorithm has an acceptable accuracy and a very acceptable speed. Although the accuracy of the ORB algorithm is not as good as the accuracy of the SIFT algorithm, ORB was still able to return one to two accurate matches at the top of the ranked list of results, which is acceptable. The ORB algorithm has also demonstrated acceptable speed in feature extraction and matching. This implies that the ORB algorithm has a better average performance over the SIFT and SURF algorithms both on low resource and high resource mobile devices and thus is more suited for a mobile guide application in providing information about artefacts in a rock art site.

2. **Can image based mobile guides be used to provide information about artefacts in a rock art heritage site?**

   In the usability experiment using the SUMI (Software Usability Measurement Inventory) scales for evaluation, the results shows that the **global** usability scale of the application is over 60% **(see Table 5.5)**, which shows the users are happy with the application in the context of use. The SUMI global scale represents the single construct of perceived quality of use and it focused heavily on general usability. The SUMI **efficiency** subscale also had a score above average, which shows that users were able to complete matching tasks in a timely fashion. They also agreed that the application was easy to use and helpful **(helpfulness subscale**). Most importantly, users were able to learn about the rock art pictographs through the mobile application and this shows that the application could be used to support educational field trips to rock art sites.

Generally, we found the results promising, considering the difficulties that are evident in the domain of digital recognition of rock art images and the use of mobile guides in cultural heritage spaces. A significant contribution of this research revolves around a generalized approach for digital recognition of rock art pictographs for image matching applications. This approach provides the advantage of extracting local and highly distinctive features of rock art despite the cluttered background and jagged edge nature of rock art. Furthermore, the design approach employed in this research could be practically applied to other related applications such as applications in the domain of rock art classification.

6.2 Future Work

For future work, it will be interesting to see how the accuracy of the SIFT algorithm can be maintained while speed is improved, especially on low resource mobile devices. It will be interesting to investigate using a scalable indexing method like PCA-SIFT for SIFT features to improve the matching speed. PCA-SIFT is known to be successful in reducing the dimensionality of SIFT features from 128 to 36.

Also, it will be worthwhile investigating more advanced image pre-processing techniques tailored specifically for rock art paintings with more complex structures. This will help algorithms like ORB detect more features, which improves the algorithm precision. The image pre-processing technique employed in this research focused heavily on rock art paintings; the approach can be improved to generalize to all rock arts.

The application could also be extended to accommodate multiple languages. This could help to provide support for users who are more comfortable with their native language.

Several segmentation techniques exist for rock art engravings hence it will be worthwhile to investigate tailor made segmentation technique for rock art paintings. A good segmentation technique could help to boost feature detection and extraction speed.

The application GUI can also be improved on so that the steps required to complete a matching task can be reduced. It will also be interesting to add a navigation layer to the mobile guide without depending on any form of location assistance program like GPS. This will automatically guide users to the exact location of artefacts in a rock art site based on pictures taken.

REFERENCES

1. Zhu, Q., Wang, X., Keogh, E. and Lee, S.H., 2011. An efficient and effective similarity measure to enable data mining of petroglyphs. Data Mining and Knowledge Discovery, 23(1), pp.91-127.

2. Hooper-Greenhill, E., 2004. Museums and the interpretation of visual culture. London and New York, Routledge.

3. Tallon, L., 2008. Introduction: Mobile, digital and personal. In L. Tallon & K.E. Walker (eds), Digital. Technologies and the Museum Experience: Handheld Guides and Other Media. Lanham, MD: Altamira Press, pp. 13-25.

4. Bruns, E., Brombach, B., Zeidler, T. and Bimber, O., 2007. Enabling mobile phones to support large-scale museum guidance. IEEE multimedia, *14*(2), pp.16-25.

5. Bay, H., Fasel, B. and Gool, L.V., 2006, May. Interactive museum guide: Fast and robust recognition of museum objects. In *Proceedings of the first international workshop on mobile vision*.

6. Henshilwood, C.S., d'Errico, F., Yates, R., Jacobs, Z., Tribolo, C., Duller, G.A., Mercier, N., Sealy, J.C., Valladas, H., Watts, I. and Wintle, A.G., 2002. Emergence of modern human behavior: Middle Stone Age engravings from South Africa. *Science*, *295*(5558), pp.1278-1280.

7. Vergnani, L. (1999), `In south Africa, researchers encounter spirit paintings by a lost people.' Chronicle of Higher Education 45(22), p A44. [Online] Available at http://chronicle.com/article/In-South-Africa-Researchers/23353 [Accessed 19 July 2015].

8. Lewis-Williams, J.D., 1990. Discovering southern African rock art. David Philip Publishers (Pty) Ltd, 208 Werdmuller Centre, Claremont, South Africa.

9. Khoisan Rock Art in the Cederberg Conservancy (2015). [Online] Available at http://www.cederberg.co.za/rockart.html, [Accessed 12 June 2015].

10. Whitley, D.S. 2005 Introduction to Rock Art Research. Walnut Creek: Left Coast Press. P. 215.

11. Douglas Harper, "Petroglyph", Etymology Dictionary. [Accessed 12 December 2015].

12. Petroglyph. New World Encyclopaedia, http://www.newworldencyclopedia.org/p/index.php?title=Petroglyph&oldid=987360, [Accessed 15 May 2016].

13. Limonite. (2015, April 18). Geology.com. http://geology.com/minerals/limonite.shtml, [Accessed 19 December 2015].

14. Hematite. (2015, April 18). Geology.com. http://geology.com/minerals/hematite.shtml, [Accessed 19 December 2015].

15. Lewis-Williams, J.D., Dowson, T.A., Bahn, P.G., Bednarik, R.G., Clegg, J., Consens, M., Davis, W., Delluc, B., Delluc, G., Faulstich, P. and Halverson, J., 1988. The signs of all times: entoptic phenomena in Upper Palaeolithic art [and comments and reply]. *Current Anthropology*, *29*(2), pp.201-245.

16. Wilcox, A.R., 1956. A scheme for the preservation and recording of the prehistoric art of South Africa. *South African Museums Association. Bulletin*, *6*, pp.194-197.

17. Adistambha, K., Davis, S.J., Ritz, C.H. and Burnett, I.S., 2010. Efficient multimedia query-by-content from mobile devices. *Computers & Electrical Engineering*, *36*(4), pp.626-642.

18. Chang, N.S. and Fu, K.S., 1980. Query-by-pictorial example. *IEEE Transactions on Software Engineering*, *6*(6), p.519.

19. Gudivada, V.N. and Raghavan, V.V., 1995. Content based image retrieval systems. *Computer*, *28*(9), pp.18-22.

20. Datta, R., Joshi, D., Li, J. and Wang, J.Z., 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, *40*(2), p.5.

21. Wolfram, S., 2002. *A new kind of science* (Vol. 5). Champaign: Wolfram media, pp. 235.

22. Eakins, J.P. and Graham, M.E., 1999. Content based image retrieval: A report to the JISC technology applications programme. 85, pp. 1-15.

23. Liu, Y., Zhang, D., Lu, G. and Ma, W.Y., 2007. A survey of content-based image retrieval with high-level semantics. *Pattern recognition*, *40*(1), pp.262-282.

24. Eakins, John; Graham, Margaret. "Content-based Image Retrieval". University of Northumbria at Newcastle. [Accessed 10 March 2015].

25. Lew, M.S., Sebe, N., Djeraba, C. and Jain, R., 2006. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *2*(1), pp.1-19.

26. Content based image retrieval. http://en.wikipedia.org/wiki/Contentbased_image_retrieval. [Accessed 12 June 2015].

27. Viitaniemi, V., 2002. *Image segmentation in content-based image retrieval*, Helsinki University OF Technology, Department of Electrical and Communications Engineering, Master Thesis, Finland 24th May 2002.

28. Smeulders, A.W., Worring, M., Santini, S., Gupta, A. and Jain, R., 2000. Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(12), pp.1349-1380.

29. Hein, A.M., 2010. Identification and Bridging of Semantic Gaps in the Context of Multi-Domain Engineering. In *Abstracts of the 2010 Forum on Philosophy, Engineering & Technology. Colorado*.

30. Oxfam Canada, At the polling station with nelson Mandela 1994, http://oxfam.ca/blogs/at-the-polling-station-with-nelson-mandela [Accessed 20 June 2015].

31. Zhao, R. and Grosky, W.I., 2002. Negotiating the semantic gap: from feature maps to semantic landscapes. *Pattern Recognition*, *35*(3), pp.593-600.

32. Zheng, Q.F. and Gao, W., 2008. Constructing visual phrases for effective and efficient object-based image retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *5*(1), p.7.

33. Chandrika, P. and Jawahar, C.V., 2010, July. Multi modal semantic indexing for image retrieval. In *Proceedings of the ACM International Conference on Image and Video Retrieval* (pp. 342-349). ACM.

34. Van Nguyen, N., Boucher, A., Ogier, J.M. and Tabbone, S., 2009, September. Region-based semi-automatic annotation using the Bag of Words representation of the keywords. In *Image and Graphics, 2009. ICIG'09. Fifth International Conference on* (pp. 422-427). IEEE.

35. Lin, W.C., Oakes, M. and Tait, J., 2010. Improving image annotation via representative feature vector selection. *Neurocomputing*, *73*(10), pp.1774-1782.

36. Liu, Y., Zhang, D., Lu, G. and Ma, W.Y., 2007. A survey of content-based image retrieval with high-level semantics. *Pattern recognition*, *40*(1), pp.262-282.

37. Ojala, T., Pietikainen, M. and Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(7), pp.971-987.

38. Heczko, M., Hinneburg, A., Keim, D. and Wawryniuk, M., 2004. Multiresolution similarity search in image databases. *Multimedia systems*, *10*(1), pp.28-40.

39. Toldin, P.P., 2010. A survey on content based image retrieval/browsing systems exploiting semantic, pp. 09-13.

40. Wang, W., Zhang, D., Zhang, Y. and Li, J., 2010, March. Fast and robust spatial matching for object retrieval. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 1238-1241). IEEE.

41. Niblack, C.W., Barber, R., Equitz, W., Flickner, M.D., Glasman, E.H., Petkovic, D., Yanker, P., Faloutsos, C. and Taubin, G., 1993, April. QBIC project: querying images by content, using color, texture, and shape. In*IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology* (pp. 173-187). International Society for Optics and Photonics.

42. Sclaro, S., Taycher, L. and La Cascia, M., 1997, June. Image Rover: a content-based browser for the World Wide Web. In *IEEE Workshop on Content-Based Access of Image and Video Libraries, (San Juan, Puerto Rico)*. Pp. 2-9

43. Analysis and Comparison of Texture Features for Content Based Image Retrieval. https://www.researchgate.net/publication/261288301_Analysis_and_Comparison_of_Texture_Features_for_Content_Based_Image_Retrieval [Accessed Apr 13, 2016].

44. Rosenfeld, A. and Kak, A.C., 2014. *Digital picture processing* (Vol. 1). Elsevier, Orlando Academic Press.

45. Levine, M.D., 1985. *Vision in man and machine*. McGraw-Hill College.

46. Weszka, J.S., Dyer, C.R. and Rosenfeld, A., 1976. A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *4*(SMC-6), pp.269-285.

47. Georgiev, T., Georgieva, E., 2009: User Interface Design for Mobile Learning Applications. e-Learning' 09. pp. 145–150.

48. Niemann, H., 1981: Pattern Analysis, Springer-Verlag, Berlin.

49. Haralick, R.M., 1979. Statistical and structural approaches to texture. *Proceedings of the IEEE*, *67*(5), pp.786-804.

50. Valkealahti, K. and Oja, E., 1998. Reduced multidimensional co-occurrence histograms in texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(1), pp.90-94.

51. Rosenfeld, A. and Weszka, J.S., 1976. Picture recognition. In *Digital pattern recognition* (pp. 135-166). Springer Berlin Heidelberg.

52. Bovik, A.C., Clark, M. and Geisler, W.S., 1990. Multichannel texture analysis using localized spatial filters. *IEEE transactions on pattern analysis and machine intelligence*, *12*(1), pp.55-73.

53. Laine, A. and Fan, J., 1993. Texture classification by wavelet packet signatures. *IEEE Transactions on pattern analysis and machine intelligence*, *15*(11), pp.1186-1191.

54. Brady, M. and Xie, Z.Y., 1996. Feature Selection for Texture Segmentation, in Advances in Image Understanding, K. Bowyer and N. Ahuja (Eds.), IEEE Computer Society Press. Pp.29-44.

55. Manjunath, B.S. and Chellappa, R., 1991. Unsupervised texture segmentation using Markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(5), pp.478-482.

56. Chaudhuri, B.B. and Sarkar, N., 1995. Texture segmentation using fractal dimension. *IEEE transactions on pattern analysis and machine intelligence*, *17*(1), pp.72-77.

57. Danish, M., Rawat, R. and Sharma, R., 2013. A Survey: Content Based Image Retrieval Based On Color, Texture, Shape & Neuro Fuzzy, Int. Journal Of Engineering Research And Applications, ISSN: 2248-9622, Vol. 3, Issue 5, Pp.839-844

58. Ping Tian, D., 2013. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*,*8*(4), pp.385-396.

59. Zhang, D. and Lu, G., 2004. Review of shape representation and description techniques. *Pattern recognition*, *37*(1), pp.1-19.

60. Sifuzzaman, M., Islam, M.R. and Ali, M.Z., 2009. Application of wavelet transform and its advantages compared to Fourier transform, Journal of Physical Sciences, Vol. 13

61. CBIR: Indexing and Retrieval, https://www.cs.auckland.ac.nz/courses/compsci708s1c/lectures/Glect-html/topic5c708FSC.htm. [Accessed 16th June 2015].

62. Gupta, N. and Athavale, V.A., 2011. Comparative Study of Different Low Level Feature Extraction Techniques for Content based Image Retrieval, International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 1, Issue 1

63. Gupta, N., Bhargava, N., Khan, M.I., Kumar, S. and Verma, B., 2009. Coefficient of Correlation Based CBIR. *Digital Image Processing*, *1*(4), pp.149-154.

64. B'far, R., 2004. Mobile computing principles: designing and developing mobile applications with UML and XML. Cambridge University Press, pp. 235-236

65. Bodker, S., 1989. A human activity approach to user interfaces. *Human-Computer Interaction*, *4*(3), pp.171-195.

66. Zhou, X.S. and Huang, T.S., 2000, April. CBIR: from low-level features to high-level semantics. In *Electronic Imaging* (pp. 426-431). International Society for Optics and Photonics.

67. Jörgen, A., Fredrik, J., Ronnie, J., Magnus, J., Anna, L., Pontus, S., Gustav, T., "Content Based Image Retrieval", Swedish Defence Research Agency Information and Aeronautic Systems SE-164 90 stockholm.

68. Mezaris, V., Kompatsiaris, I. and Strintzis, M.G., 2004. Region-based image retrieval using an object ontology and relevance feedback. *Eurasip Journal on applied signal processing*, *2004*, pp.886-901.

69. Friedman, J., Hastie, T. and Tibshirani, R., 2001. The elements of statistical learning (Vol. 1). Springer, Berlin: Springer series in statistics.

70. Tsai, C.F. and Hung, C., 2008. Automatically annotating images with keywords: A review of image annotation systems. *Recent Patents on Computer Science*, *1*(1), pp.55-68.

71. Jin, W., Shi, R. and Chua, T.S., 2004, October. A semi-naive bayesian method incorporating clustering with pair-wise constraints for auto image annotation. In *Proceedings of the 12th annual ACM international conference on Multimedia* (pp. 336-339). ACM.

72. Shi, R., Feng, H., Chua, T.S. and Lee, C.H., 2004, July. An adaptive image content representation and segmentation approach to automatic image annotation. In *International conference on image and video retrieval* (pp. 545-554). Springer Berlin Heidelberg.

73. Vailaya, A., Figueiredo, M.A., Jain, A.K. and Zhang, H.J., 2001. Image classification for content-based indexing. *IEEE transactions on image processing*, *10*(1), pp.117-130.

74. Luo, J. and Savakis, A., 2001, October. Indoor vs outdoor classification of consumer photographs using low-level and semantic features. In *Image Processing, 2001. Proceedings. 2001 International Conference on* (Vol. 2, pp. 745-748). IEEE.

75. Town, C. and Sinclair, D., 2000. Content based image retrieval using semantic visual categories. *TR2000-14, AT&T Labs Cambridge*.

76. Feng, H. and Chua, T.S., 2003, November. A bootstrapping approach to annotating large image collection. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval* (pp. 55-62). ACM.

77. MacArthur, S.D., Brodley, C.E. and Shyu, C.R., 2000. Relevance feedback decision trees in content-based image retrieval. In *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on* (pp. 68-72). IEEE.

78. Sethi, I.K., Coman, I.L. and Stan, D., 2001, March. Mining association rules between low-level image features and high-level concepts. In*Aerospace/Defense Sensing, Simulation, and Controls* (pp. 279-290). International Society for Optics and Photonics.

79. Hall, L.O., Chawla, N. and Bowyer, K.W., 1998, October. Decision tree learning on very large data sets. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on* (Vol. 3, pp. 2579-2584). IEEE.

80. Urgen Herczeg, J., Hohl, H. and Schwab, T., 1991. XIT-A Multi-Layered Tool for User Interface Design. In *Human Aspects in Computing, Proceedings of the Fourth International Conference on Human-Computer Interaction* (Vol. 1, pp. 678-683).

81. Mulligan, R.M., Altom, M.W. and Simkin, D.K., 1991, April. User interface design in the trenches: some tips on shooting from the hip. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 232-236). ACM.

82. Rumpradit, C. and Donnell, M.L., 1999, January. Navigational cues on user interface design to produce better information seeking on the World Wide Web. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on* (pp. 10-pp). IEEE.

83. Salton, G., 1989. Automatic text processing-the analysis, transformation and retrieval of information by computer.

84. Rui, Y., Huang, T.S., Ortega, M. and Mehrotra, S., 1998. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, *8*(5), pp.644-655.

85. Doulamis, A.D. and Doulamis, N.D., 2004. Generalized nonlinear relevance feedback for interactive content-based retrieval and organization. *IEEE transactions on circuits and systems for video technology*, *14*(5), pp.656-671.

86. Cheng, S.F., Chen, W. and Sundaram, H., 1998, October. Semantic visual templates: linking visual features to semantics. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on* (pp. 531-535). IEEE.

87. Smith, J.R., 1998, June. Decoding image semantics using composite region templates. In *Content-Based Access of Image and Video Libraries, 1998. Proceedings. IEEE Workshop on* (pp. 9-13). IEEE.

88. Kherfi, M.L., Ziou, D. and Bernardi, A., 2004. Image retrieval from the World Wide Web: Issues, techniques, and systems. *ACM Computing Surveys (CSUR)*, *36*(1), pp.35-67.

89. He, R., Xiong, N., Yang, L.T. and Park, J.H., 2011. Using multi-modal semantic association rules to fuse keywords and visual features automatically for web image retrieval. *Information Fusion*, *12*(3), pp.223-230.

90. Feng, H., Shi, R. and Chua, T.S., 2004, October. A bootstrapping framework for annotating and retrieving WWW images. In *Proceedings of the 12th annual ACM international conference on Multimedia* (pp. 960-967). ACM.

91. Lisin, D.A., Mattar, M.A., Blaschko, M.B., Learned-Miller, E.G. and Benfield, M.C., 2005, June. Combining local and global image features for object class recognition. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops* (pp. 47-47). IEEE.

92. Tuytelaars, T. and Mikolajczyk, K., 2008. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, *3*(3), pp.177-280.

93. Murase, H. and Nayar, S.K., 1995. Visual learning and recognition of 3-D objects from appearance. *International journal of computer vision*, *14*(1), pp.5-24.

94. Attneave, F., 1954. Some informational aspects of visual perception. *Psychological review*, *61*(3), p.183.

95. Omhover, J.F. and Detyniecki, M., 2004. Combining text and image retrieval. In *Proceedings of the EUROFUSE Workshop on Data and Knowledge Engineering* (pp. 388-398).

96. Venters, CC., and Cooper M., 2000. “Content-based image retrieval”, Tech. Rep. JTAP-054, JISC Technology Application Program.

97. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D. and Steele, D., 1995. Query by image and video content: The QBIC system. *Computer*, *28*(9), pp.23-32.

98. Danish, M., Rawat, R. and Sharma, R., 2013. A Survey: Content Based Image Retrieval Based On Color, Texture, Shape & Neuro Fuzzy, Int. Journal Of Engineering Research And Applications, ISSN: 2248-9622, Vol. 3, Issue 5, Pp.839-844.

99. Talbar, S.N. and Varma, S.L., 2009, December. iMATCH: image matching and retrieval for digital image libraries. In *2009 Second International Conference on Emerging Trends in Engineering & Technology* (pp. 196-201). IEEE.

100. Lux, M., and Savvas, C. 2008. “Lire: Lucene Image Retrieval – An Extensible Java CBIR Library”. Proceedings of the 16th ACM International Conference on Multimedia, pp. 1085-1088.

101. Miljkovic, O., 2009. Image pre-processing tool. *Kragujevac J Math*, *32*, pp.97-107p.

102. Sonka, M., Hlavac, V. and Boyle, R., 1998. *Image processing, analysis, and machine vision*. PWS Publishing.

103. An introduction to image feature extraction, http://blog.iconfinder.com/finding-more-duplicate-icons-introduction-image-feature-extraction/, [Accessed 18th July 2015].

104. El-gayar, M.M. and Soliman, H., 2013. A comparative study of image low level feature extraction algorithms. *Egyptian Informatics Journal*, *14*(2), pp.175-181.

105. Jinxia, L. and Yuehong, Q., 2011, August. Application of SIFT feature extraction algorithm on the image registration. In *Electronic Measurement & Instruments (ICEMI), 2011 10th International Conference on* (Vol. 3, pp. 177-180). IEEE.

106. Harris, C. and Stephens, M., 1988, August. A combined corner and edge detector. In *Alvey vision conference* (Vol. 15, p. 50).

107. Kim, S., 2012. Robust Corner Detection by Image-Based Direct Curvature Field Estimation for Mobile Robot Navigation. *International Journal of Advanced Robotic Systems* (9)

108. Tuytelaars, T. and Mikolajczyk, K., 2008. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, *3*(3), pp.177-280.

109. Loy, G. and Zelinsky, A., 2002, May. A fast radial symmetry transform for detecting points of interest. In *European Conference on Computer Vision* (pp. 358-368). Springer Berlin Heidelberg.

110. Guerrero Pena, M., 2011. A comparative study of three image matching algorithms*: SIFT, SURF, and FAST*. Utah State University.

111. Schmid, C., Mohr, R. and Bauckhage, C., 2000. Evaluation of interest point detectors. *International Journal of computer vision*, *37*(2), pp.151-172.

112. Mokhtarian, F. and Suomela, R., 1998. Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(12), pp.1376-1381.

113. He, X. C., Yung, N. H. C., 2008. Corner detector based on global and local curvature properties. Optical Engineering 47 (5), 057008.

114. Concept of edge detection. http://www.tutorialspoint.com/dip/concept_of_edge_detection.htm, Accessed 14th July 2015.

115. Canny, J., 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), pp.679-698.

116. Smith, S.M. and Brady, J.M., 1997. SUSAN—a new approach to low level image processing. *International journal of computer vision*, *23*(1), pp.45-78.

117. Trajković, M. and Hedley, M., 1998. Fast corner detection. *Image and vision computing*, *16*(2), pp.75-87.

118. Rosten, E. and Drummond, T., 2006, May. Machine learning for high-speed corner detection. In *European conference on computer vision* (pp. 430-443). Springer Berlin Heidelberg.

119. Rublee, E., Rabaud, V., Konolige, K. and Bradski, G., 2011, November. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision* (pp. 2564-2571). IEEE.

120. Rosin, P.L., 1999. Measuring corner properties. *Computer Vision and Image Understanding*, *73*(2), pp.291-307.

121. Fraser, C., Jazayeri, I. and Cronk, S., 2010, April. A feature-based matching strategy for automated 3D model reconstruction in multi-Image Close-Range Photogrammetry. In *ASPRS 2010 Annual Conference*.

122. Lowe, D.G., 1999. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157). IEEE.

123. Lowe, D.G., 2004. Distinctive image features from scale-invariant key points. *International journal of computer vision*, *60*(2), pp.91-110.

124. Ke, Y. and Sukthankar, R., 2004, June. PCA-SIFT: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (Vol. 2, pp. II-506). IEEE.

125. Bay, H., Tuytelaars, T. and Van Gool, L., 2006, May. Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Springer Berlin Heidelberg.

126. Nagar, A., Saxena, A., Bucak, S., Fernandes, F. and Bhat, K.P., 2013, November. Low complexity image matching using color based SIFT. In *Visual Communications and Image Processing (VCIP), 2013* (pp. 1-6). IEEE.

127. Walia, E. and Suneja, A., 2010. A conceptual study on image matching techniques. *Global Journal of Computer Science and Technology*, *10*(12).

128. Vimina, E.R. and Jacob, K.P., 2014. An Evaluation of Image Matching Algorithms for Region Based Image Retrieval. *International Journal of Advancements in Computing Technology*, *6*(6), p.75.

129. Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3), pp.175-185.

130. Approximate Nearest Neighbour. https://www.cs.umd.edu/~mount/ANN, [Accessed 9th June 2015].

131. Friedman, J.H., Bentley, J.L. and Finkel, R.A., 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, *3*(3), pp.209-226.

132. Silpa-Anan, C. and Hartley, R., 2008, June. Optimised KD-trees for fast image descriptor matching. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1-8). IEEE.

133. Wang, J., Shen, H.T., Song, J. and Ji, J., 2014. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*.

134. Indyk, P. and Motwani, R., 1998, May. Approximate nearest neighbours: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing* (pp. 604-613). ACM.

135. Muja, M. and Lowe, D.G., 2009. Fast Approximate Nearest Neighbours with Automatic Algorithm Configuration. *VISAPP (1)*, *2*(331-340), p.2.

136. Flann Index, http://docs.opencv.org/2.4/modules/flann/doc/flann_fast_approximate _nearest_neighbor_search.html, [Accessed 15 May 2015]

137. Föckler, P., Zeidler, T., Brombach, B., Bruns, E. and Bimber, O., 2005, December. PhoneGuide: museum guidance supported by on-device object recognition on mobile phones. In *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia* (pp. 3-10). ACM.

138. Fritz, G., Seifert, C., Luley, P., Paletta, L. and Almer, A., 2005. Mobile vision for ambient learning in urban environments. *Mobile Learning Anytime Everywhere*, pp.77-80.

139. Coors, V., Huch, T. and Kretschmer, U., 2000. Matching buildings: Pose estimation in an urban environment. In *Augmented Reality, 2000.(ISAR 2000). Proceedings. IEEE and ACM International Symposium on* (pp. 89-92). IEEE.

140. Seifert, C., Paletta, L., Jeitler, A., Hödl, E., Andreu, J.P., Luley, P. and Almer, A., 2004, September. Visual object detection for mobile road sign inventory. In *International Conference on Mobile Human-Computer Interaction* (pp. 491-495). Springer Berlin Heidelberg.

141. Hare, J.S., Lewis, P.H., Gordon, L. and Hart, G., 2008, January. MapSnapper: engineering an efficient algorithm for matching images of maps from mobile phones. In *Electronic Imaging 2008* (pp. 68200L-68200L). International Society for Optics and Photonics.

142. Lim, J.H., Li, Y., You, Y. and Chevallet, J.P., 2007, July. Scene recognition with camera phones for tourist information access. In *2007 IEEE International Conference on Multimedia and Expo* (pp. 100-103). IEEE.

143. Falk, J.H. and Dierking, L.D., 2002. Lessons without limit: How free-choice learning is transforming education. Rowman Altamira.

144. Kenteris, M., Gavalas, D. and Economou, D., 2011. Electronic mobile guides: a survey. *Personal and ubiquitous computing*, *15*(1), pp.97-111.

145. Proctor, N., 2005. Off base or on target? Pros and cons of wireless and location-aware applications in the museum. *ICHIM, Paris, France*.

146. Kuehner, T., Schmidt, M., Hinze, C. & Glemme, S. (2003). Information systems: Information system task. 1-19. (http://www.grin.com/en/e-book/18074/information-systems-in-museums, Compiler) Munich: GRIN Publishing GmbH.

147. Almeida, L., 2002. Archeoguide: an augmented reality guide for archaeological sites. *IEEE Computer Graphics and Applications*. Pp. 52-60.

148. Bombara, M., Cali, D. and Santoro, C., 2003, September. Kore: A multi-agent system to assist museum visitors. Joint Workshop "From Objects to Agents": Intelligent Systems and Pervasive Computing, pp. 175-178, 2003.

149. Albertini, A., Brunelli, R., Stock, O. and Zancanaro, M., 2005, January. Communicating user's focus of attention by image processing as input for a mobile museum guide. In *Proceedings of the 10th international conference on Intelligent user interfaces* (pp. 299-301). ACM.

150. Ancona, M., Capello, M., Casamassima, M., Cazola, W., Conte, D., Pittore, M., 2006. Mobile Vision and Cultural Heritage: the AGAMEMNON project. In Proceedings of the 1st International Workshop on Mobile Vision, Austria

151. Silver, C.S., 1985: The Rock Art of Seminole Canyon State Historic Park: Deterioration and Prospects for Conservation. Report #4000-430 may be obtained from Texas Parks and Wildlife, 4200 Smith School Road, Austin, TX 78744.

152. Davies, N., Cheverst, K., Dix, A. and Hesse, A., 2005, September. Understanding the role of image recognition in mobile tour guides. In Proceedings of the 7th international conference on Human computer interaction with mobile devices & services (pp. 191-198). ACM.

153. Amrit P., Turgay C., David, P., Azizo f., Hima, V., 2014. Automatic Indexing of South African Rock Art, Southern African Institute for Computer Scientists and Information Technologists (SAICSIT): Empowered by Technology

154. Clogg, P., Díaz-Andreu, M. and Larkman, B., 2000. Digital image processing and the recording of rock art. Journal of Archaeological Science, 27(9), pp.837-843.

155. Zeppelzauer, M., Poier, G., Seidl, M., Reinbacher, C., Breiteneder, C., Bischof, H., 2015, "Interactive segmentation of rock-art in high-resolution 3D reconstructions," *International Digital Heritage conference*, Granada, pp. 37-44.

156. Díaz, D. and Castro, D., 2002. Pattern recognition applied to rock art. *BAR International Series*, *1016*, pp.463-468.
157. Deufemia, V., Paolino, L. and de Lumley, H., 2012, November. Petroglyph recognition using self-organizing maps and fuzzy visual language parsing. In Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference, volume 1, pp. 852-859.

158. Dean, S. R., 1985. The Radon Transform, In Mathematical Analysis of Physical Systems, Ed. R. E. Mickens, pp. 81-133. Van Nostrand Reinhold, New York.

159. Deufemia, V. and Paolino, L., 2013, July. Combining unsupervised clustering with a non-linear deformation model for efficient petroglyph recognition. In International Symposium on Visual Computing (pp. 128-137). Springer Berlin Heidelberg.

160. Belongie, S., Malik, J. and Puzicha, J., 2002. Shape matching and object recognition using shape contexts. IEEE transactions on pattern analysis and machine intelligence, 24(4), pp.509-522.

161. Keysers, D., Gollan, C. and Ney, H., 2004, August. Local context in non-linear deformation models for handwritten character recognition. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on (Vol. 4, pp. 511-514). IEEE.

162. Deufemia, V. and Paolino, L., 2014, June. Segmentation and Recognition of Petroglyphs Using Generic Fourier Descriptors. In International Conference on Image and Signal Processing (pp. 487-494). Springer International Publishing.

163. Takaki, R., Toriwaki, J., Mizuno, S., Izuhara, R., Khudjanazarov, M. and Reutova, M., 2006. Shape analysis of petroglyphs in central Asia. Forma, 21(3), pp.243-258.

164. Seidl, M. and Breiteneder, C., 2011. Detection and classification of petroglyphs in gigapixel images-preliminary results. In VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage-Short and Project Papers, pp. 45-48.

165. Ojala, T., Pietikainen, M. and Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence, 24(7), pp.971-987.

166. Haralick, R.M. and Shanmugam, K., 1973. Textural features for image classification. IEEE Transactions on systems, man, and cybernetics, (6), pp.610-621.

167. Perona, P. and Malik, J., 1990. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on pattern analysis and machine intelligence, 12(7), pp.629-639.

168. SONKA, A., HLAVAC, V. and BOYLE, R. (1999) Image Processing, Analysis, and Machine Vision, Brooks/Cole Publishing.

169. Seidl, M., Wieser, E., Zeppelzauer, M., Pinz, A. and Breiteneder, C., 2014. Graph-based similarity of petroglyphs. In VISART \Where Computer Vision Meets Art". ECCV 2014, Springer, 2014.

170. Bai, X., Latecki, L.J. and Liu, W.Y., 2007. Skeleton pruning by contour partitioning with discrete curve evolution. IEEE transactions on pattern analysis and machine intelligence, 29(3), pp.449-462.

171. Ling, H. and Jacobs, D.W., 2007. Shape classification using the inner-distance. IEEE transactions on pattern analysis and machine intelligence, 29(2), pp.286-299.

172. Belongie, S., Malik, J., Puzicha, J., 2002. Shape matching and object recognition using shape contexts. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(4):509-522, 2002.

173. Wieser, E., Seidl, M. and Zeppelzauer, M., 2016. A study on skeletonization of complex petroglyph shapes, Multimedia Tools and Applications, 2016, DOI: 10.1007/s11042-016-3395-1

174. About OpenCV, https://en.wikipedia.org/wiki/OpenCV, [Accessed 11 January 2015]

175). Sheng, L., (1999). Java Native Interface: Programmer's Guide and Specification. Prentice Hall. p. 320. ISBN 0-201-32577-2.

176. Ramsay, A. (2009), `Three reasons to start designing iteratively'. [Online] Available at http: //www.andersramsay.com/2009/03/01/three-reasons-to-start-designing-iteratively, [Accessed 21-October-2012].

177 Davis, A.M., 1992. "Operational prototyping: a new development approach", IEEE Software - SOFTWARE, vol. 9, no. 5, pp. 70-78, 1992.

178. Systems Design, https://en.wikipedia.org/wiki/Systems_design, [Accesses 25 February 2015].

179. Böhm, C., Berchtold, S. and Keim, D.A., 2001. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases, ACM Computing Surveys (CSUR), v.33 n.3, pp.322-373.

180. Freidman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Softw. 3:209–226.

181. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A., 2007. Object retrieval with large vocabularies and fast spatial matching. In Proc. IEEE Conference on Computer Vision and Pattern Recognition. 1, pp.3.

182. Gionis, A., Indyk, P., Motwani., R., 1999. Similarity search in high dimensions via hashing. In Proc. Int'l Conf. on Very Large Data Bases (VLDB). 1, 2, 4

183. European Commission. CORDIS (1999), "Music Report Summary", http://cordis.europa.eu/result/rcn/23029_en.html. [Accessed 28 November 2015].

184. Powers, D.M., 2011. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation" (PDF). Journal of Machine Learning Technologies **2** (1), pp.37–63.

185. Turpin, A. and Scholer, F., 2006, August. User performance versus precision measures for simple search tasks. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 11-18). ACM.

186. Van Veenendaal, E.P., 1999. "Questionnaire based usability testing", Conference Proceedings European Software Quality Week, Brussels.

187. ISO 9241-11 (1995), "Ergonomic Requirements for office work with visual display terminals (VDT's) - Part 11: Guidance on usability", International Organization of Standardization.

188. ISO/IEC FCD 9126-1 (1998), "Information technology - Software product quality - Part 1: Quality model", International Organization of Standardization.

189. Babbar, G., Bajaj, P., Chawla, A. and Gogna, M., 2010. A comparative study of image matching algorithms. International Journal of Information, Technology and Knowledge Management. July-December. 2(2): 337-339.

190. Kirakowski, J., 1994. The use of questionnaire methods for usability assessment. Available at: http://sumi.ucc.ie/sumipapp.html. [Accessed 19 December 2015].

191. Ap, J. and Wong, K.K., 2001. Case study on tour guiding: Professionalism, issues and problems. Tourism Management, 22(5), pp.551-563.

192. Deufemia, V., Paolino, L., Mascardi, V. and De Lumley, H., 2014. PetroAdvisor: A Volunteer-based Information System for Collecting and Rating Petroglyph Data. J. Vis. Lang. Comput. 25(6): 963-972(2014).

APPENDIX A: STRENGTH AND WEAKNESS ANALYSIS

Item: 41 The software hasn't always done what I was expecting.
Verdict: Disagree! Strength: 99.9

Item: 4 This software has at some time stopped unexpectedly.
Verdict: Disagree! Strength: 99.9

Item: 46 This software occasionally behaves in a way which can't be understood.
Verdict: Disagree! Strength: 99.9

Item: 3 The instructions and prompts are helpful.
Verdict: Agree! Strength: 99.9

Item: 11 I sometimes wonder if I am using the right function.
Verdict: Disagree! Strength: 99.9

Item: 39 It is easy to make the software do exactly what you want.
Verdict: Agree! Strength: 99.9

Item: 45 It is easy to forget how to do things with this software.
Verdict: Disagree! Strength: 99.9

Item: 38 Error messages are not adequate.
Verdict: Disagree! Strength: 99.8

Item: 13 The way that system information is presented is clear and understandable.
Verdict: Agree! Strength: 99.7

Item: 30 I keep having to go back to look at the guides.
Verdict: Disagree! Strength: 99.7

Item: 32 There have been times in using this software when I have felt quite tense.
Verdict: Disagree! Strength: 99.7

Item: 22 I would not like to use this software every day.
Verdict: Undecided! Strength: 99.7

Item: 37 I think this software has sometimes given me a headache.
Verdict: Disagree! Strength: 99.6

Item: 27 Using this software is frustrating.
Verdict: Disagree! Strength: 99.6

Item: 48 It is easy to see at a glance what the options are at each stage.
Verdict: Agree! Strength: 99.3

Item: 5 Learning to operate this software initially is full of problems.
Verdict: Disagree! Strength: 99.3

Item: 24 This software is awkward when I want to do something which is not standard.
Verdict: Disagree! Strength: 99.1

Item: 49 Getting data files in and out of the system is not easy.
Verdict: Undecided. Strength: 98.7

Item: 26 Tasks can be performed in a straight forward manner using this software.
Verdict: Agree. Strength: 98.3

Item: 25 There is too much to read before you can use the software.
Verdict: Disagree. Strength: 98.2

Item: 44 It is relatively easy to move from one part of a task to another.
Verdict: Agree. Strength: 97.6

Item: 6 I sometimes don't know what to do next with this software.
Verdict: Disagree. Strength: 97.5

Item: 35 Learning how to use new functions is difficult.
Verdict: Disagree. Strength: 96.7

Item: 23 I can understand and act on the information provided by this software.
Verdict: Agree. Strength: 96.4

Item: 36 There are too many steps required to get something to work.
Verdict: Disagree. Strength: 96.4

Item: 9 If this software stops it is not easy to restart it.
Verdict: Disagree. Strength: 95.9

Item: 15 The software documentation is very informative.
Verdict: Agree. Strength: 95.8

Item: 50 I have to look for assistance most times when I use this software.
Verdict: Disagree. Strength: 95.3

Item: 21 I think this software is inconsistent.
Verdict: Disagree? Strength: 94.6

Item: 40 I will never learn to use all that is offered in this software.
Verdict: Disagree? Strength: 94.3

Item: 31 It is obvious that user needs have been fully taken into consideration.
Verdict: Agree? Strength: 94.3

Item: 12 Working with this software is satisfying.
Verdict: Agree? Strength: 89.6

Item: 2 I would recommend this software to my colleagues.
Verdict: Tend to agree? Strength: 88.6

Item: 19 I feel in command of this software when I am using it.
Verdict: Slightly agree? Strength: 88.1

Item: 20 I prefer to stick to the functions that I know best.
Verdict: Undecided? Strength: 87.8

Item: 8 I find that the help information given by this software is not very useful.
Verdict: Disagree? Strength: 86.4

Item: 10 It takes too long to learn the software functions.
Verdict: Disagree? Strength: 85.9

Item: 28 The software has helped me overcome any problems I have had in using it.
Verdict: Agree? Strength: 84.9

Item: 14 I feel safer if I use only a few familiar functions.
Verdict: Agree? Strength: 81.8

Item: 29 The speed of this software is fast enough.
Verdict: Undecided? Strength: 79.2

Item: 47 This software is really very awkward.
Verdict: Disagree? Strength: 75.2

Item: 42 The software presents itself in a very attractive way.
Verdict: Disagree? Strength: 63.1

Item: 7 I enjoy the time I spend using this software.
Verdict: Undecided? Strength: 59.1

Item: 1 This software responds too slowly to inputs.
Verdict: Disagree? Strength: 56.6

Item: 43 Either the amount or quality of the help information varies across the system.
Verdict: Disagree? Strength: 52.8

Item: 33 The organisation of the menus seems quite logical.
Verdict: Agree~ Strength: 48.9

Item: 18 There is never enough information on the screen when it's needed.
Verdict: Tend to disagree~ Strength: 44

Item: 16 This software seems to disrupt the way I normally like to arrange my work.
Verdict: Disagree~ Strength: 43.6

Item: 34 The software allows the user to be economic of keystrokes.
Verdict: Agree~ Strength: 30.6

Item: 17 Working with this software is mentally stimulating.
Verdict: Slightly agree~ Strength: 19.4

APPENDIX B: FREE FROM QUESTIONS

Question 1: What in General do you use this Software for?

USER ID	COMMENT
1	To find paintings
2	Finding art pictures
3	I would use this software for navigation at art gallery/museum
4	To get familiar with cultural art
5	locating and matching art paintings
6	finding matches to generate information
7	Finding and obtaining information about cultural art
8	To find matches of art paintings
9	Matching of Paintings
10	image matching
11	To find information about pictures
12	locating and matching art paintings
13	finding heritage paintings
14	Getting information about paintings
15	navigation
16	nil
17	to find meaning of things
18	locating and matching images
19	image matching
20	matching pictures to a database to gain information about the pictures
21	To get information about visual art. As a tourist, it's a new way of getting more information
22	locating art pictures
23	image matching
24	Picture Capture
25	Navigating spaces
26	locating and matching art paintings
27	art location and matching for description
28	Asides this experiment, it could be useful in various types of sight seeing
29	image matching
30	i think it's easy to use

Question 2: What do you think is the best aspect of this software and why?

USER ID	COMMENT
1	A direction to where paintings are located
2	Ease of Use
3	Front End GUI is user friendly. Simple to Use
4	it can be used in a wide variety of images
5	Ease of Use
6	It is very simple and easy to use
7	It is easy to use and straight forward plus it helps provide information
8	The instructions are clear and to the point
9	Image Identification
10	Very simple and easy to use
11	The simplicity, it's really easy to handle
12	Searching the pictures and knowing the meaning
13	Ease of you...the instructions were very helpful
14	Getting of information about the paintings
15	easy to use
16	nil
17	Ease of Use
18	Matching concept makes it easier to locate images and get information about them
19	matching of images with their percentage of match
20	The idea of being able to learn about something that the name isn't known makes this application very interesting
21	The originality and flexibility of it
22	Clear and simple.
23	Its informative and time saving and that can be very important in a world where we are always in a rush
24	the image comparison
25	It's interesting
26	Description to where the art pictures are located
27	the software is great
28	Simplicity...It pretty straight forward
29	Helps to learn more about culture
30	Matching of pictures

Question 3: What do you think needs most improvement and why?

USER-ID	COMMENT
1	nil
2	Response time
3	Make retrieved images larger
4	nil
5	speed
6	i would need to use it more to answer this fully
7	The graphics user interface could have some design input that could make the picture taken look more nicer and friendly
8	Not really sure
9	When the software does not find match, an option should be available that user can select
10	Nothing..i think it’s great
11	The searching speed
12	When it shows the pictures, they need to be more clear
13	a voice communication can be added to make it more fun
14	The interface text is too small
15	speed
16	Interface can be better
17	nil
18	The images need more description
19	Pictures need to be large
20	The pictures matches were very small and even the text are too tiny. There are multiple matches in result and it’s difficult to tell which one is the exact one.
21	The inclusion of sign which can carter for blind people
22	The images should be larger...
23	Error messages should me more detailed
24	It restarted...
25	GUI
26	speed
27	make images larger
28	The user interface could be more attractive. Saving images can be eliminated so one can just match straight away
29	presentation can be improved
30	There should be faster loading process

APPENDIX C: SUMI QUESTIONAAIRE

NB The information you provide is kept completely confidential, and no information is stored on computer media that could identify you as a person.

This questionnaire has 50 statements. Please answer them all. After each statement there are three boxes.

- Check the first box if you generally AGREE with the statement.
- Check the middle box if you are UNDECIDED, or if the statement has no relevance to your software or to your situation.
- Check the right box if you generally DISAGREE with the statement.

In checking the left or right box you are not necessarily indicating strong agreement or disagreement but just your general feeling most of the time.

There are also five general questions at the end.

Statements 1 - 10 of 50.	Agree	Undecided	Disagree
This software responds too slowly to inputs.	○	○	○
I would recommend this software to my colleagues.	○	○	○
The instructions and prompts are helpful.	○	○	○
This software has at some time stopped unexpectedly.	○	○	○
Learning to operate this software initially is full of problems.	○	○	○
I sometimes don't know what to do next with this software.	○	○	○
I enjoy the time I spend using this software.	○	○	○
I find that the help information given by this software is not very useful.	○	○	○
If this software stops it is not easy to restart it.	○	○	○

	Agree	Undecided	Disagree
It takes too long to learn the software functions.			
Statements 11 - 20 of 50.	Agree	Undecided	Disagree
I sometimes wonder if I am using the right function.			
Working with this software is satisfying.			
The way that system information is presented is clear and understandable.			
I feel safer if I use only a few familiar functions.			
The software documentation is very informative.			
This software seems to disrupt the way I normally like to arrange my work.			
Working with this software is mentally stimulating.			
There is never enough information on the screen when it's needed.			
I feel in command of this software when I am using it.			
I prefer to stick to the functions that I know best.			
Statements 21 - 30 of 50.	Agree	Undecided	Disagree
I think this software is inconsistent.			
I would not like to use this software every day.			
I can understand and act on the information provided by this software.			
This software is awkward when I want to do something which is not standard.			

Statement			
There is too much to read before you can use the software.	○	○	○
Tasks can be performed in a straight forward manner using this software.	○	○	○
Using this software is frustrating.	○	○	○
The software has helped me overcome any problems I have had in using it.	○	○	○
The speed of this software is fast enough.	○	○	○
I keep having to go back to look at the guides.	○	○	○

Statements 31 - 40 of 50.	Agree	Undecided	Disagree
It is obvious that user needs have been fully taken into consideration.	○	○	○
There have been times in using this software when I have felt quite tense.	○	○	○
The organisation of the menus seems quite logical.	○	○	○
The software allows the user to be economic of keystrokes.	○	○	○
Learning how to use new functions is difficult.	○	○	○
There are too many steps required to get something to work.	○	○	○
I think this software has sometimes given me a headache.	○	○	○
Error messages are not adequate.	○	○	○
It is easy to make the software do exactly what you want.	○	○	○
I will never learn to use all that is offered in this software.	○	○	○

Statements 41 - 50 of 50.	Agree	Undecided	Disagree
The software hasn't always done what I was expecting.	○	○	○
The software presents itself in a very attractive way.	○	○	○
Either the amount or quality of the help information varies across the system.	○	○	○
It is relatively easy to move from one part of a task to another.	○	○	○
It is easy to forget how to do things with this software.	○	○	○
This software occasionally behaves in a way which can't be understood.	○	○	○
This software is really very awkward.	○	○	○
It is easy to see at a glance what the options are at each stage.	○	○	○
Getting data files in and out of the system is not easy.	○	○	○
I have to look for assistance most times when I use this software.	○	○	○

What, in general, do you use this software for?

How important for you is the kind of software you have just been rating?

- ○ Extremely important
- ○ Important
- ○ Not very important
- ○ Not important at all

How would you rate your software skills and knowledge?

- Very experienced and technical
- I'm good but not very technical
- I can cope with most software
- I find most software difficult to use

What do you think is the best aspect of this software, and why?

What do you think needs most improvement, and why?

APPENDIX D: QUERY IMAGES FOR COMPARATIVE EXPERIMENT

APPENDIX E: QUERY IMAGES FOR INTERACTIVE EXPERIMENT