Research Report
Similarity of Abstract Images

Benjamin Rosman
0407237H

Abstract

Abstract images are defined as images devoid of any recognisable content, but instead consist of a variety of geometric patterns and shapes of various colours. They have various applications, including use as passwords in some authentication systems. This is as a result of an abstract image being easy for the owner of the password to recognise but difficult for an intruder based on a verbal description of the image. Applications such as this require the use of an algorithm which is able to measure the similarity between abstract images.

This research proposes an approach to developing such an algorithm – by designing separate methods to match different aspects of the images. Previous work has not been specifically conducted into the matching of abstract images. However, the literature suggests that both colour and texture based methods can be used to compare images in a similar way to human vision. Thus the use of algorithms based on both colour and texture approaches is investigated in this research. The ultimate aim is for the development of an algorithm which can categorise image pairs as being similar or different, such that these results match the opinion of a human.

The colour algorithm developed in this research relies on the technique of histogram binning, which is a summary of the colour content of an image. The texture descriptor which is used is based on a Gabor filter bank, which extracts textures of various sizes and directions from an image. Both of these algorithms generate signatures of the images which contain the essential colour and texture information extracted by the algorithms. These signatures for two images are then compared directly by the Earth Mover's Distance metric, to determine the distance between them. This document describes the design, implementation and testing of these algorithms and ultimately shows that they are suitable for abstract image recognition.

The colour matching algorithm has an accuracy of 72% when compared to human results, and the texture algorithm is only 63% accurate. Consequently, the colour approach is seen to be intrinsically better suited to the task than the texture matching. However, a combination of these two algorithms is shown to classify image similarities with an 82% accuracy when compared to human vision. This confirms the hypotheses that images can be accurately compared using methods which are based on the formal elements of an image, and also that the inclusion of more of these factors will improve the algorithm's accuracy.

These results suggest that the accuracy of an algorithm could be further improved by considering the addition of other factors, such as the geometric shape content of the images.

Declaration

I declare that this research report is my own, unaided work. It is being submitted for the degree of Bachelor of Science with Honours at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

Benjamin Rosman, 25th November 2007

Acknowledgements

I would like to acknowledge and thank my supervisor, Professor Ian Sanders, for proposing this research topic, and also for the insightful and invaluable discussions, advice, comments and feedback he has provided during the time I have spent working on this research.

I would also like to express my sincere gratitude to my family for all the help, understanding and encouragement they have provided throughout the course of this research. Their support in everything from editing documents to discussions of the results is greatly appreciated.

Contents

Chapter 1

Introduction

Images have become an essential part of our daily interactions with computers. Developments and advances in digital cameras and high-speed Internet and networking links have resulted in a greater need for information content to contain images. As a result, where text was once sufficient for any inter-agent communication, the inclusion of images is now ubiquitous. The images originate from a variety of sources, including photographs, user drawn images or computer generated images.

As these images have become so prominent, it has become important that a sufficient range of operations can be performed on such images. Therefore a technique to determine the similarities between images should also be considered. There are a number of fields where this operation can be applied. An important application is in robotics and computer vision [Swain and Ballard 1991], where it is vital that a device can identify its surroundings and various objects. Other uses include extracting particular images from a large image database by means of a query image, or in medical imaging [Kelly *et al.* 1995].

Another application for image similarities is in alternative approaches to password systems. Together with the increase in global connectivity, comes an increase in the need for greater security. Computer users, particularly on the Internet, often sign up to a multitude of sites and services. These often require some form of identification for authorisation, which in turn requires the use of passwords. As computer processing power and thus speed improves, so does the likelihood of passwords being cracked by intruders within a relatively short space of time. As a result, users are forced to use and thus remember even longer and more complicated passwords, and often even resort to writing these down to avoid forgetting them. This in itself is a security flaw which could easily be exploited.

Consequently, different security methods are being examined and proposed by many researchers. Weinshall and Kirkpatrick [2004] suggest one solution would be the use of images as passwords. Such an image should preferably be abstract in nature. The reason for this approach of using abstract images in password systems would be that such an image may be very difficult to describe to another person, yet easy to remember and recognise. This would imply great difficulty in obtaining the password from a user. If a user was forced to describe his password, it would be almost impossible to describe it by anything other than vague phrases, such as "it has blue and yellow wavy squiggles with some rectangles in the background". From this example it is clear that the user could easily recognise their own password, but would have severe difficulties explaining it. If the distractors were similar enough to the original password, for example all having blue and yellow patterns and shapes, then an intruder would find difficulty entering the system. As a result, security would be greatly improved.

The implementation of this system would involve the user selecting a password from amongst a collection of similar images, or distractors, displayed on the screen. This process may, in certain circumstances, be repeated one or several times to enhance the security of the system. These distractor images are required to be similar enough to the actual password image in order to present a challenge to anyone who wishes to gain unauthorised entry into the system and who has only been given a high-level description of the password. On the other hand, the differences between the actual password and the distractors should not be too subtle to prevent the correct user from being able to identify their password.

The aforementioned password application is therefore heavily reliant on the ability of the system to accurately generate or select distractors similar to an original image. So an important subproblem of the entire system is the ability to recognise similar images. Recognising objects within a picture is an active topic of research with many potential applications in artificial intelligence. The techniques that are used for such tasks arise from a number of differing fields, including image processing and statistics. Machine learning techniques, such as neural networks, are also frequently applied to solve such problems. The difference is that in this application, there should be no recognisable objects in the images because these images are abstract.

It is a trivial task to determine whether or not two images are identical, by examining and comparing each pair of corresponding pixels. This is matching images on a discrete scale. A continuous scale would on the other hand involve a score, for example from 0 to 1, or a percentage, that would indicate the degree of similarity between two images based on some predefined criteria. This factor, or collection of factors, should preferably be chosen based on the manner in which this same task is performed by humans. While it would also be simple to give this statistic as the percentage of pixels that match, a more sophisticated approach is sought. If an object was slightly rotated, this may result in all the pixels changing somewhat, but the difference may be hardly noticeable to the human eye.

It remains unclear as to how humans distinguish images, and judge the similarities between scenes, yet this is done automatically. Human responses also differ greatly in various contexts, as different information may be taken into account and used as the basis for this judgment. An example of this is that a red shirt and a red box may be seen to be similar in terms of colour, yet different when the shapes are compared. Furthermore images may be judged differently by the same person at different times or by different people. Results by Squire and Pun [1997] also indicate that categorisation of images can vary greatly between individuals.

It can thus be seen that the human similarity metric is complicated and relies on a number of different factors. A definite advantage of this specific problem of judging image similarities for the password application is that the particular images to be considered are abstract images. By being abstract, these pictures are devoid of recognisable content, thus eliminating one of the major factors of similarity; one which is quite possibly the most difficult to model. Abstract images are also particularly likely to be computer generated, especially in a situation where a number of varieties of similar images are required as distractors.

The research discussed in this document aims at finding a reliable, robust method for ranking the similarity or difference between two images to be used in a context such as password distractor generation. This method is required to be fully autonomous and operate in a way that produces results congruent to those produced should a human perform the same task manually.

To do this, an in depth discussion of previous literature is required. This is presented in Chapter 2. A number of techniques are presented that may be used in the determination of this similarity metric. This literature review highlights different approaches that have been considered in the past for various applications of image matching, and include both the problems of extracting particular information from an image and comparing this information from two images by means of a suitable metric. They are also applied to different features of an image. This chapter furthermore provides a discussion on ways of incorporating these different methods into a single algorithm.

Chapter 3 then proceeds to outline the direction that was considered for this research. This is done by a formal expression of the problem that was considered by presenting the research question, and from this the specific research hypotheses are derived. This chapter continues to explain the methodology that was followed in this research, by fully describing the approach taken to developing an abstract image similarity metric. This chapter then proceeds to discuss the methods that were considered during the research, and the colour and texture algorithms that were eventually used. This was all done with reference to the background literature. An overview is given of the subsequent experimentation that was conducted on these algorithms. Thus the document explains how the research hypotheses were tested.

Following the design and implementation of the methods used for determining similarities, a testing phase was required to determine the extent to which the different methods could be said to be successful at examining abstract image similarities. For this reason, Chapter 4 presents the results of the experiments that were run using the implemented methods, to show particular successes and failures of these techniques. These include examples of images that were matched both correctly and incorrectly. These results indicate that the colour based algorithm provides more accurate results than its texture counterpart, however these results can be combined to give an improved solution.

The findings presented in Chapter 4 are then analysed and discussed in Chapter 5. This chapter highlights the strengths and weaknesses of the various methods that were used, and shows the results of using a combination of these methods. The evaluation that is given of these results shows that the hypotheses of the research can be accepted, by providing an estimation of the similarity of the results of these algorithms with actual human perception. This chapter also gives recommendations of further work that can be conducted into this topic.

This research in its entirety can be considered as highly valuable and unique. Although extensive work has been done into the field of image matching in image processing, the application of this knowledge to abstract images, and particularly with a potential application such as that of password systems, remains largely unexplored.

Chapter 2

Background

2.1 Introduction

In order to develop a method which could be used to determine similarities in abstract images, ultimately to be used in an application such as visual passwords, it is important to examine previous research that has provided various approaches to image matching.

Much research has been conducted in recent years into obtaining a robust metric for describing the similarities or differences between two images. These are primarily used in the context of obtaining and ranking all images in a database that are similar to the query image, with the aim of providing a method for locating all related images in that database.

There are many different approaches to measuring image similarities, ranging from the lowest level of working with individual pixels of an image to the highest level of a user or expert classifying an image using keywords or labels. This latter approach will not be considered in this research, primarily because abstract images are considered to be too difficult to classify by keywords. It is also inefficient for a human to classify every distractor used by an application, particularly if these images are computer-generated. Instead, a more thorough approach using image processing techniques will be investigated and, as a result, the developed methods will focus on information stored at the pixel level.

An ideal image matching algorithm will compare the content of an image, and rank other images based on how closely the objects represented in those images match the original query. A method such as this still evades researchers, and there is no doubt that huge advances are still required in artificial intelligence and image processing before such a thorough object recognition system could be designed. This does not, however, affect this research because an object recognition system would not provide any advantages in the domain of abstract images. Instead, this work will focus on a multitude of different image processing techniques that utilise a broad range of information derived from the pixel details in an image. The methods examined in the literature identify colours, lines, textures, or various combinations of these and use this information to match images.

Many of the methods described herein attempt to mimic various aspects of the human recognition system, and as such, have been derived in conjunction with advances in the psychophysical understanding of human perception. Although the aim of the research described in this document is to determine the similarities between images in a way that is familiar to humans, the similarities between the underlying method itself and the mechanisms of human vision and understanding are considered largely irrelevant.

A hybrid of different techniques would likely lead to the best results, and therefore a number of different approaches to image similarity techniques are presented in this chapter.

This section aims to provide an overview of work conducted into the field of image processing, with direct applications to image matching, and particularly abstract images. Since the colour and texture of an image are among the most important factors for describing an image, attention is primarily given to these factors. Section 2.2 first describes work that has been conducted into understanding the use of colours in images, as this is perhaps one of the most fundamental aspects of an image. Section 2.3 then considers some different approaches to classifying and comparing the textures present in a picture.

A fairly recently developed metric that has been used successfully in comparing images is the Earth Mover's Distance. This has been used in conjunction with both colour and texture classifiers, and so is discussed separately in Section 2.4. Finally, this chapter is summarised in Section 2.5.

2.2 Colour Based Methods

This section provides an overview of colour based comparison methods. Colour is an essential property of an image, particularly an abstract one. This is because, even though humans are able to identify grayscale images in most situations, the application of colour information simplifies and speeds up the identification process. It is also important for this particular application, since a lack of content in an image requires that the emphasis of recognition be placed on other formal elements of the image. Colour is also a property of individual pixels and is thus easy to access and manipulate.

Colour spaces refer to the method used for representing the colours in an image. A description of various colour spaces is provided as an introduction to the use of colour in a similarity method. This is an important consideration, as different colour spaces provide direct access to different properties of colour.

An overview of colour based methods is then provided, starting with the most common approach of using colour histograms, as well as a discussion of some of the techniques used to compare the histograms of different images. Alternate approaches to dealing with colour, signatures and segmentation, are then presented.

2.2.1 Colour Spaces

An important consideration when dealing with colours in an image is selecting the colour space to be used by a method. A colour space is the coordinate system used to describe different colours within an image. There are a large variety of colour spaces, and each emphasises different aspects of a colour. The results of an algorithm can vary significantly based on the choice of colour space.

Selecting a different colour space for an algorithm that operates by matching some aspect of the colour property of abstract images could result in very different distances being produced. Some may place an emphasis on the intensity of a colour, others on the existence of specific colour in the images, and yet others on aspects such as hues, contrast or brightness.

The standard, and conceptually most simple colour space is RGB, which represents intensities of red, green and blue in a particular colour. Each of these colours is represented on a separate colour channel, which are combined to produce the full colouring of the image. This is used by a large percentage of commercial applications. However, RGB is a perceptually non-uniform colour space, implying that perceptual colour changes do not relate well to the change in RGB values [Di Sciascio and Celentano 1997]. Where the RGB coordinate system is used, pixels are often normalised by dividing each colour component of each pixel by the total colour intensity at that point, or else converted to grayscale [Stevens 2001].

Over the years, a number of different colour spaces have been proposed. YIQ is one such colour space, which was standardised for television use [Katsumata and Matsuyama 2005]. A colour in RGB space can be easily transformed into YIQ space [Katsumata and Matsuyama 2005] by means of linear transforms.

Two commonly used colour spaces are HSV (Hue, Saturation and Value) and HVC (Hue, Value and Chroma). These are potentially more useful colour spaces for determining similarities in a manner reminiscent of human perception as they incorporate the hue of a colour (represented by the H component). They also provide saturation and luminance information respectively. These transformations from RGB are non-linear, and provided by Katsumata and Matsuyama [2005], and Di Sciascio and Celentano [1997] respectively. Applications using these spaces typically focus on the use of hue to identify colours. This is a very valuable approach, as the hue can describe whether a colour is blue or green, for example. It is also an angle and therefore periodic.

The CIE-XYZ space [Di Sciascio and Celentano 1997] is another that can be obtained by means of linear transforms from RGB space. This is taken to describe every colour that humans can visualise, and other colour spaces are in fact subsets of CIE-XYZ [Rubner 1999]. Therefore, many others are defined in terms of transformations from this space.

This is the case for $O_1O_2O_3$ space, which is used by Mirmehdi and Petrou [2000]. This is the opponent colour space, and describes the luminance of a colour, as well as the balance between red and green, and blue and yellow, both of which are pairs of colours that cannot exist together.

CIE Lab and CIE Luv are two other spaces that are derived from the CIE-XYZ space, albeit with non-linear transforms. These are considered as perceptually uniform [Mirmehdi and Petrou 2000], which implies that the Euclidean distance between two colours in the colour space is approximately equal to the distance between the two colours as perceived by humans [Rubner 1999]. The L corresponds to the luminance of a colour and is the same in the two systems. These systems differ in their use of a, b coordinates or u, v coordinates, which are used to uniquely identify a colour in each respective system.

Seaborn *et al.* [2005] propose a different colour space, the fuzzy colour category map (FCCM). This consists of a categorised map of regions in the hue-value plane that differ in size and shape, corresponding to colour categories laid out by similarity with respect to human visual perception. As would be expected, this colour space is shown to correlate well to human vision.

2.2.2 Histograms

Colour histograms are among the most commonly used methods for comparing the colours in multiple images. The principle behind colour histograms is to define a set of bins, and an equivalence relation that hashes different colours to different bins, based on similarities in quantities such as hue, intensity or any other property of colour. This property is dependent on the colour space used. The histogram is then represented by a set of weights, being the number of pixels in each bin.

As such, a histogram could be said to be a colour summary of an image, with each bin corresponding to a colour, or range of colours, and the weight of each bin which is the number of pixels that fall into that bin, or alternatively the percentage of the image that is of that particular colour.

This technique is, however, sensitive to changes in lighting conditions which may force colours into different bins [Swain and Ballard 1991]. Another major concern with histograms is the size of bin to use. Coarse binning, that is the use of large bins, implies a loss in detail of image information, whereas fine binning may not only be computationally inefficient, but may also result in similar colours not being grouped together [Rubner 1999]. A solution to this problem is given by Puzicha *et al.* [1999] as adaptive binning with flexible bin sizes, but that requires prior knowledge of the colour distribution of the image.

The most practical criticism of the histogram method is that it does not account for the layout and distribution of pixels of a certain colour in an image. Greenspan *et al.* [2001] provides a slightly different approach, which includes storing the coordinates of a pixel as well as its colour, and thus representing colours in a five-dimensional space.

Histograms are particularly useful for colour quantisation, which is the reduction in the number of colours in an image [Hsieh and Fan 2000]. This is dependent on the hashing method used, and requires a two-phase method for initially generating the reduced palette, and then assigning the pixels of an image to one of the palette colours. One simple way of performing this reduction, is by computing a

Singular Value Decomposition (SVD) of the histogram bins, and eliminating the smaller singular values, as described by Guillon *et al.* [2001].

The colour histogram is employed widely in the literature, and authors often use it as a base against which new methods can be tested, such as by Parker and Behm [2004]. It is also used by Berwick and Lee [2004] to generate image signatures, which are compared using other methods.

There is great potential in the use of histograms in comparing two abstract images, as they provide a fingerprint of all the colours present, together with the quantities thereof, in an image. As colour information could be construed as an important aspect of an abstract image, the use of a method such as this seems justified.

2.2.3 Histogram Comparisons

An obvious way of comparing the similarities between the colour content of two images, is to generate histograms of their respective colour distributions, and compare these histograms. Histogram comparisons are classified as either bin-by-bin measures which compare corresponding bins only, or cross-bin measures which compare different bins [Greenspan *et al.* 2001]. Although cross-bin measures have been shown to be more robust in certain instances [Rubner 1999], they tend to be more computationally intensive.

The most straightforward of these histogram comparisons involve using one of the distance metrics, the Mean Absolute Error or the Mean Square Error (Euclidean distance). These are commonly referred to as L_1 and L_2 respectively and are special cases of the Minkowski metric [Russell and Sinha 2001], given by

$$L_p(X,Y) = \Big(\sum_i |x_i - y_i|^p\Big)^{\frac{1}{p}}.$$

Russell and Sinha [2001] performed a comparison of the two methods (L_1 and L_2), by examining which of these produced results closest to those of human subjects. The findings showed that similarities determined by the L_1 metric were significantly closer to human perception. Ecker [2002] suggests that Pearson's correlation coefficient may provide an even better comparison.

Many other statistical methods, such as the χ^2 statistic, can be used for comparing bins [Greenspan *et al.* 2001]. In fact, any of the methods described in this section could be applied directly to two images and compare them pixel-wise, but this would obviously be much less robust as they do not take any transformations into account and these metrics are therefore extremely sensitive to rotational, translational, scale, colour and tone changes [Jacobs *et al.* 1995].

The histogram intersection operator, H, is a histogram-specific operator defined by Swain and Ballard [1991] that acts on two histograms of the same size, X and Y as the sum of the smallest of the two corresponding bins, normalised by the bin sizes of the model image, or

$$H(X,Y) = \frac{\sum_i \min(x_i, y_i)}{\sum_i x_i}.$$

A limitation of many histogram methods is that they generally require that the same number of bins be present in each histogram. This restricts the histograms, as these methods would be unable to do things such as compare histograms of different coarseness or from different colour spaces. Both of these cases would likely require histograms of different sizes.

A different approach to histogram comparisons is given by the Kullback-Leibler Divergence [Rubner 1999] and the Jeffrey Divergence [Rodden *et al.* 2000]. These measure how efficiently one of the histograms could be derived from the other. The Jeffrey Divergence is a modification of the former, to introduce symmetry into the method, and therefore improve the stability thereof. It is given by

$$d_J(X,Y) = \sum_i \left(x_i \log \frac{x_i}{m_i} + y_i \log \frac{y_i}{m_i} \right)$$

with $m_i = \frac{x_i+y_i}{2}$.

2.2.4 Colour Signatures

A colour histogram of an image is one example of a colour signature. A signature is a selection of information taken from an image to summarise one aspect of that image, such as colour. These signatures are usually developed in a way that allows them to be compared directly, by using a metric such as the histogram comparisons. In this way, information about a particular feature of an image is represented in a manner that allows for simpler and more meaningful comparisons between images.

Rubner [1999] uses signatures that are represented as a kind of generalisation of histograms, by certain dominant colours and the percentage of an image that is similar to each dominant colour. A difference is that the signature allows for a certain percentage of the image to be specified as unknown.

The same paper suggests another type of signature, where colour bins are sorted by position in the image, to allow for simple representation of spatial information within the signature.

The Gaussian Mixture Model is proposed by Greenspan *et al.* [2001] as another form of image signature. It involves labelling the pixels in an image as belonging to one of a number of Gaussian mixtures of an image, which are convex ellipses over a region with the average colour experienced over that region. In this way, each image is represented as a collection of these ellipses. These are compared using the Kullback-Leibler Divergence. A similar method is employed by CANDID [Kelly *et al.* 1995] that involves defining clusters of related colours by Gaussian distributions, each with a size or weight.

2.2.5 Segmentation

Segmentation refers to the process of dividing an image into regions of similar colour. This operation is strongly linked to colour quantisation, as the result is usually the same image presented in fewer colours than the original. The aim of this is usually a simplified representation of the original image and can aid in removing subtle detail and noise from images before comparing them.

As such, one approach to segmentation is to create a coarse histogram of an image, and then reassign a colour to each pixel in the image, based on its histogram bin.

Clustering algorithms can also be used for image segmentation, but these are more commonly used for texture clustering and so are discussed in Section 2.3.4. An example of clustering used primarily for colour segmentation is Chen *et al.* [2002], wherein the authors make use of the adaptive clustering algorithm (ACA). This is iterative, and forms clusters based on a dominant colour, which are refined as a window of decreasing size is repeatedly passed over the image, averaging the colours under the window. This algorithm was designed for smooth regions, and so does not perform well over highly detailed textural regions.

2.3 Texture Based Methods

In an image, colour is stored at the individual pixel level, but texture is defined over a region. Therefore it is incorrect to refer to the texture of a pixel, but rather the texture of different localised parts of an image. As a result, texture matching is a more difficult problem than that of colour matching.

The texture of an image describes repeated details in a portion of the image. Computer-generated abstract images are likely to consist of a variety of geometric shapes, filled with various textures. The use of texture matching would thus be appropriate.

The different approaches to detecting textures are often linked in the literature, and one method usually employs others. Some of the main approaches are presented here, being the use of Gabor filters, other wavelet based methods, and clustering algorithms.

A major drawback associated with many texture based methods, is that they require supervision [Fauzi and Lewis 2003]. This means that an algorithm may require some additional information from a human in order to complete its task. A typical example of this is the K-means clustering algorithm (discussed in more depth in Section 2.3.4) which requires a value for K, the number of textures to be found in an image. The algorithm developed in the research described in this document requires the use of an unsupervised, fully automatic method for texture matching.

Texture based methods will perform one of two main functions. Either, they match particular textures in one or more images, or else they can segment a single image into various regions of different texture. From that point, the individual textures can be handled separately. The methods described in this section can be used to perform either task.

2.3.1 Gabor Filters

Gabor filters rely on the fact that a textured region in an image will consist of sets of lines that are repeated, possibly with different scales (corresponding to frequencies) as well as directions. The combination of these different types of repeated lines describe a texture.

Gabor filters have been used extensively for texture detection and discrimination in the literature. They treat textures in images as energy or frequency patterns. These filters are the product of a two-dimensional Gaussian function, with a specific harmonic function [Fogel and Sagi 1989]. From this, variations of a family of Gabor wavelets are defined, similar to those given by Kruizinga and Petkov [1999] or Rubner [1999], as

$$g_{\xi,\eta,\lambda,\theta}(x,y) \quad = \quad \frac{1}{2\pi\sigma_x\sigma_y}\,\mathbf{e}^{-\frac{1}{2}\left((\frac{x'}{\sigma_x})^2+(\frac{y'}{\sigma_y})^2\right)}\,\mathbf{e}^{i\pi x'} \tag{2.1}$$

$$\begin{aligned} with \quad x' &= \lambda\Big((x-\xi)\cos\theta + (y-\eta)\sin\theta\Big) \\ y' &= \lambda\Big(-(x-\xi)\sin\theta + (y-\eta)\cos\theta\Big). \end{aligned}$$

A specific filter is designed to detect the presence of a texture of a certain orientation and scale [Rubner 1999]. Filters are applied by the process of convolution, and as a result of having a particular orientation and frequency, the filter will emphasise any region of the image with lines corresponding to the properties of this particular filter. Once a filter has been applied to a region, the total textural energy of that region can be computed to characterise the components of that texture in a specific direction and frequency [Kasparis *et al.* 2001].

Once Gabor filters have been applied to an image, the resulting Gabor features can be post-processed [Grigorescu *et al.* 2002]. There are a number of different ways of doing this, including the calculation of Gabor energy which is a combination of the results, or the grating cell operator, which is used in the presence of a oriented texture consisting of a set of regularly spaced bars [Kruizinga and Petkov 1999; Grigorescu *et al.* 2002].

Owing to the fact that each filter is designed to detect only a particular aspect of a texture, filter banks are created to apply the Gabor filter to multiple scenarios. These consist of an array of filters, with varying values for orientation and bandwidth, that are applied successively and independently to an image [Fogel and Sagi 1989]. As described by Rubner [1999], the filter bank should be designed so that the orientation change between successive filters is uniform, and the scale change is exponential.

The results of the application of the filters of a filter bank to an image are known as a feature vectors which can be regarded as a signature for a particular image or region thereof [Ma and Manjunath 1997], each with a defined frequency and orientation. A comparison of a set of these feature vectors for two images would result in a direct comparison of the textural information available in an image or part thereof.

2.3.2 Edge Detection

Edge detection is the traditional method for determining the structure of an image. There are a number of techniques that can be used to identify the edges present within an image, and some of them are presented in this section. An edge detection method is used to identify different regions in an image, demarcated by an edge – a change in shape or tone, represented by a hard line. This is a useful function in that it allows one to distinguish between different shapes in an image.

Another important application of edge detection methods, which is the use highlighted in this section, is for enhancing the edges in a texture before attempting to detect them, using a method such as Gabor filters. This results in the textural details being enhanced in an image, while reducing the background noise of the image. Edge detection could thus be a useful preprocessing step for a texture detection method.

The traditional class of edge detecting methods is to apply filters to an image by the process of convolution. An example of this is the Sobel edge detector [Parker and Behm 2004]. Two filters, S_X and S_Y, are defined as follows

$$S_X = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad\qquad S_Y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

These estimate the gradient of the image in the x and y directions, respectively. Therefore the application of both, successively, to an image will provide the overall gradient at each point. A sudden change in the image is therefore marked by a steep gradient which gives a high value and will thus appear as white in an image, whereas a relatively uniform area has a low gradient and thus appears dark. The result of this edge detection is therefore a predominantly black image with the edges between regions standing out as white.

There are many alternative ways of finding the edges in an image. Theoharatos *et al.* [2005] describe an algorithm for performing colour edge detection based on the use of graph theory, in particular using minimal spanning trees (MSTs). These trees are built using information on the distribution of the colour distribution of pixels in the image and considering neighbouring pixels of similar colour as being clustered together.

Another approach, is the use of Independent Component Analysis (ICA) and Principle Component Analysis (PCA) which are two methods that have been employed by Katsumata and Matsuyama [2005] to determine and distinguish shapes in an image, to be used for matching. A high-level description of this method is that it involves representing an image in terms of orthogonal ICA or PCA bases. These are then compared between images by finding the maximum inner product between pairs of corresponding basis sets.

One final edge detection method to consider is from the field of mathematical morphology. By taking the erosion of an image A with respect to some small structuring element B, one would obtain an image consisting of a slightly reduced A. This is due to the fact that the erosion of A by B, written $A \ominus B$ is defined [Haralick 1987] as

$$A \ominus B = \{x | x + b \in A \text{ for every } b \in B\}$$

where B would typically be a small symmetric shape, such as a round disk with a radius of only a few pixels. As a result, one could subtract the erosion of an image from the original image to obtain another image C consisting of the edges of the original image. Formally,

$$C = A - (A \ominus B)$$

2.3.3 Other Wavelet Based Methods

Although Gabor filters can be considered a wavelet based method [Manjunath and Ma 1996], there are a host of other methods that rely on the use of different wavelet functions. None of these, however, are as widely employed in the literature as the Gabor filter.

Wavelet based methods are usually applied successively to an image in order to obtain multiresolutional approximations to the image. Thus, the resolution of these versions of an image become repeatedly of higher quality. This simulates a human approaching an object, with more details becoming clear as the human moves closer. These images form either a pyramid, in that they are subsampled at each stage and so become smaller, or a tower, where they are just blurred and the number of pixels at each level is preserved [Mirmehdi and Petrou 2000].

Fauzi and Lewis [2003] present a method for segmenting textures that is completely unsupervised. The method uses a tower multiresolution approach based on a discrete wavelet frames decomposition to perform the segmentation. This segmentation method consists of two phases. The first is a decomposition phase, which involves multiple repeated decompositions using discrete wavelet frames. The decomposed images stay the same size as the original. The second phase is the segmentation phase. Starting with the most decomposed level, the texture cluster centres of each of the four images on that level are calculated using a mean shift algorithm. The respective data points are then associated with the nearest cluster centre, which represents a texture region. This method is shown to be accurate as well as not too computationally expensive.

Haar wavelets also provide a multiresolutional approach to working with an image. As described by Stollnitz *et al.* [1994], the Haar wavelet decomposition of a vector involves the pairwise averaging of the elements of a vector, to lower the resolution, with half the elements. In order to reconstruct the original vector, one requires the use of the detail coefficients. These describe the information lost in producing the averages.

Consider an example of this concept. Let the original vector be given as $(12\ 4\ 9\ 3)$. Now, a lower resolution version of this vector can be obtained by averaging the values. This decomposed vector would now be given as $(8\,6)$. The detail coefficients are the distance between the original elements of the vector, and the averages. In this case, they are $(4\ 3)$. Appending the detail coefficients to the lower resolution vector results in a vector of the same size as the original. This process can then be repeated, by just considering the first half of the vector (the averages from the previous resolution) at each step. The final result would be a single number representing an average of the whole of the original vector, followed by the detail coefficients required to restore the vector to any intermediate resolution. This same principle can be applied to an entire two-dimensional image [Stollnitz *et al.* 1994] by performing this operation on rows as well as columns of the image.

2.3.4 Clustering Algorithms

Clustering algorithms are used in a wide variety of different approaches for grouping elements with the same texture together. This is done in order to reduce the number of texture details and thus produce a compact signature. The concept of a clustering algorithm is to group together all the neighbouring pixels in an area that possess an approximately similar value of some property, and then in some cases refine these clusters progressively. As demonstrated by Chen *et al.* [2002], different clustering algorithms can be utilised to cluster pixels based on either texture or colour.

One of the most commonly used clustering techniques for the process of texture clustering is the K-means clustering algorithm. It operates by generating K centroids and then assigning all points to their nearest centroid. Centroids are then recomputed and the process repeats until centroids remain constant in position. As described by Rubner [1999], although this is a fast algorithm, it suffers the major drawback that the value for K, the number of clusters required from the algorithm, is required as input into the algorithm by the user. The downside of this is that the algorithm is thus supervised and requires human assistance. Furthermore, setting a low value for K in a highly detailed image will result in the loss of information, while too high a value may result in strange artifacts in the final clustering. Despite this, the K-means algorithm is sometimes used with a predefined value of K, as an approximated starting point for another algorithm. The number of clusters can also be calculated by the use of another algorithm.

The mean shift algorithm, as described by Fauzi and Lewis [2003], is a technique for determining the centres of clusters. The algorithm examines the density gradient of the feature data, and shifts points toward their cluster centres in the hope that they will converge on those locations. This shift is based on the positions and distributions of other data points within each point's immediate neighbourhood and so this shift is toward density centres. This method can therefore be used to calculate the number of clusters in a dataset.

The fuzzy c-means algorithm [Fauzi and Lewis 2003] is another clustering algorithm that functions in a similar manner to the K-means algorithm. It requires a predefined number of clusters, associates points with different clusters and calculates cluster centres. The process repeats until the distances between the data and cluster centres is minimised. The crucial factor here is that data can belong to more than one cluster, i.e. membership of clusters is fuzzy.

2.4 The Earth Mover's Distance

The Earth Mover's Distance (EMD) is a metric for comparing two signatures, but is derived from a type of transportation problem, and is also known as the Monge-Kantorovich mass transference problem [Rubner 1999].

This metric is used to compare the differences between signatures [Rubner *et al.* 2000]. It is considered to be superior to other metrics for this task, and bears resemblance to human similarity perception [Levina and Bickel 2001]. Another advantage of the EMD over other methods is that it allows for partial matching between signatures.

According to Rubner [1999], the EMD operates on two signatures, possibly of different sizes, defined respectively as $P = (p_1, w_{p_1}), ..., (p_m, w_{p_m})$ and $Q = (q_1, w_{q_1}), ..., (q_n, w_{q_n})$. The p_i and q_j refer to clusters within the signatures, and the w_{p_i} and w_{q_j} are their respective weights.

This metric relies on an underlying distance measure $d(p_i, q_j)$ which is defined as the ground distance between the clusters p_i and q_j. Conceptually, this ground distance defines a space into which the components of the signatures are placed. The Earth Mover's Distance then computes the minimum work required to transform the one signature into the other in this predefined space.

The exact problem is to determine the flow, f_{ij} from p_i to q_j to minimize

$$\text{WORK}(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} d(p_i, q_j) f_{ij}$$

subject to

$$\begin{aligned}
f_{ij} &\geq 0 & 1 \leq i \leq m, 1 \leq j \leq n \\
\sum_{j=1}^{n} f_{ij} &\leq w_{p_i} & 1 \leq i \leq m \\
\sum_{i=1}^{m} f_{ij} &\leq w_{q_j} & 1 \leq j \leq n \\
\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} &= \min(\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j})
\end{aligned}$$

with the Earth Mover's Distance defined as the normalised work,

$$\text{EMD}(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d(p_i, q_j) f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}.$$

This is a linear programming problem [Rubner *et al.* 1997] and can be applied to signatures of various natures, including colour and texture. The Earth Mover's Distance, as described above, computes the minimum flow required to transform one signature of weighted components into another.

The Earth Mover's Distance has also been shown to satisfy the mathematical conditions for a true metric, as long as the ground distance is itself a metric [Rubner 1999].

Puzicha *et al.* [1999] have shown that although the Earth Mover's Distance is a very computationally expensive method, it does not have high space requirements and produces significantly lower errors and higher precision in classifying images than other metrics.

2.5 Conclusion

This chapter examined a number of different ways for comparing images in the pursuit of similarities. The techniques covered were diverse in their respective approaches, yet all focused on the comparison of a particular property of the images.

Colour can be considered an important property of any image, particularly an abstract one. The way in which colour is represented by means of differing colour spaces can have an impact on which aspects of the colour are emphasised whilst performing comparisons. The result is that a suitable colour space must be selected for a particular application.

The colour based techniques presented here offer methods of generating signatures, which is a useful course of action because it allows for the colour information of an image to be summarised in a meaningful manner. Common successfully used methods to do this include histograms, which have the advantage that they may be compared in a number of different ways.

Texture is another aspect of an image that can be used as a distinguishing feature, and describes repeated details over regions of an image. The primary method for detecting and matching texture is through the use of wavelet methods, such as Gabor filters. Different wavelet methods consider textures as particular distributions of energy and rely on using this representation to extract texture information from an image. Again these result in signatures, which can be directly compared in a number of ways.

The Earth Mover's Distance is used as a metric to compare different signatures, regardless of the method used to generate them. It allows the combination of different signatures with different weights and performs an optimisation in calculating the minimum cost of transforming one signature into another. Furthermore, it has been used successfully to compare both colour and texture signatures.

The methods and algorithms discussed in this chapter form the basis for the algorithms that are developed during this research. The details of this developmental process are explained in the following chapter.

Chapter 3

Research Methodology

3.1 Introduction

The previous chapter provided an overview of some of the many strategies that have been employed, with varying levels of success, to define and identify the similarities between various kinds of images in differing contexts. These methods all used particular colour or texture information to compare images.

The aim of this research is to develop a technique that can be used for establishing a quantitative measure of the similarity or difference between two abstract images. The algorithm to do this is based largely on the approaches discussed in the previous chapter.

This chapter describes the manner in which this research was conducted, and the details thereof. Section 3.2 formulates a research question based on the background literature presented in the previous chapter, as well as the motivation of developing a password application wherein each password takes the form of an abstract image. From this research question, two research hypotheses are presented in Section 3.3.

The general methodology of the study is discussed in Section 3.4 with additional details provided on the design of the various aspects of the algorithm in Section 3.5. This section also presents some findings to demonstrate situational robustness of each algorithm. The details of the implementation as well as all testing of the algorithms appear in Section 3.6.

3.2 Research Question

One potential alternative to conventional password systems is the use of graphical passwords. This involves each user of a system having a particular abstract image, or series thereof, as a password. This could be from an image database or generated on-the-fly by the system. The prompt for a password would involve selecting the correct image from amongst a series of distractors.

The distractors that are displayed should not be too dissimilar to the actual password, so that no distractor stands out as being very different from the password. Similarly, there must be enough variation between the images for identification by a human to be possible. Such a password system should thus independently and automatically be able to make such judgments about sets of images.

With this motivation, the focus of this research is on the automatic quantification of similarities between abstract images. The specific research question that this work will attempt to answer is presented as

- Can two abstract images be effectively compared by considering the formal elements of the two images?

That the images can be compared effectively refers to the potential use in the application of visual passwords. This question could be answered by examining different methods of comparing images based on the formal elements of an image, which is the approach taken in this research.

3.3 Research Hypotheses

Human perception of visual similarities and dissimilarities is a complex skill and can be said to rely on a large number of factors. Perhaps the most important of these factors is the content, or arrangement of recognisable objects, that appear in a scene. However, by working in the domain of abstract images, this factor is eliminated from the concerns of this application.

Instead, comparisons can be made based on formal elements, of which there are many examples, such as colour, texture and shape. As discussed in Chapter 2, one of the more efficient techniques to do this, is to use a method to generate a signature, or fingerprint, for each image and then provide a metric to compare these signatures.

The hypothesis for this research can then be formalised as

- The similarity between two images can be measured in a manner that agrees with human observation by computing and comparing image signatures based on colour and texture information.

Although this hypothesis states the basic methods that were used for comparisons, it does not specify the techniques used for generation or contrasting of signatures. This was the focus of the research, and is discussed in the following sections.

Furthermore, it is required that any developed algorithms be practical for use in the aforementioned password application. This implies that the metric provided by these algorithms should in some way coincide with human similarity judgment. Thus, image similarities should be ranked by the algorithms as they would be by a human. This would imply that the results agree with human observation, as required by the aforementioned hypothesis.

This hypothesis could therefore be accepted by an algorithm being repeatedly able to distinguish different degrees of similarity between images, in a manner consistent with human judgment. The aim of this research was thus to develop such an approach and verify its reliability and robustness.

Additionally, this hypothesis advocates the idea that the discriminating power of such an algorithm would be improved due by two factors being used together, rather than each one considered individually. More formally,

- The inclusion of additional factors corresponding to the formal elements of an image provide a more reliable similarity measure between images.

The implications of this second hypothesis are that one could potentially examine even more factors and incorporate them into an algorithm with the effect of improving the accuracy of the similarity measure between images. This research aimed to provide insight into this matter by considering the similarities between images determined by both colour and texture, and comparing this to the results generated by each measure alone.

3.4 Methodology

The problem of comparing images based on any criteria is not a trivial one, and as such has been extensively investigated in the literature. Although a number of different methods have been proposed and used, the context of the question posed in this research is somewhat different. Therefore a new approach was considered, based on a combination of past successes.

The crux of this research is the development of algorithms which are to examine two abstract images, and measure their degree of similarity. In order to achieve this, several methods for determining colour or texture similarities were examined, modified where necessary and finally combined. The algorithms were completely specified by taking combinations of these methods found in the literature.

The various methods were implemented and incorporated into the different algorithms that consisted of a hybrid of some of these approaches. All the methods were designed to take an image and generate a signature based on some quality of the image. The algorithms accept two images as input, use appropriate methods to generate signatures for each of the two, and then apply a metric to compare the signatures of the two respective images.

Implementation leads to experimentation and evaluation. A diverse set of abstract images has been compiled, as described in Section 3.6, on which the different algorithms were run. The accuracy of similarity scoring of each algorithm was recorded. The algorithms were also run with variations in the parameters required by the methods.

An analysis of the results produced during this testing phase provided information on the relative accuracies of the algorithms, particularly regarding the similarity ranking as determined by a human observer, in this case the author. By examining these accuracies, it will be possible to determine which algorithm is best suited to the problem of quantifying abstract image similarities.

The development of these algorithms through a variety of techniques is intended to replicate human perception and recognition by use of a hybrid system in a very specific environment – that of abstract images.

3.5 Algorithm Design

Developing the algorithms required an understanding of the specifics of the problem at hand. Therefore this section contains a discussion of the specific domain of abstract images, before covering the major factors that were taken into consideration when developing the algorithms in this research.

The actual design of the algorithms is based on the generation of image signatures. These are generated by means of both a colour based method as well as a texture based method. These were implemented independently, each with a separate means of signature comparison. Finally, they were combined in an effort to improve the ability of the algorithms to identify similarities in images.

3.5.1 Abstract Images

An abstract image is one in which the emphasis is on the design, rather than representation of any recognisable form. As such, an abstract image would be typified by a nonrepresentational composition consisting of straight and organic lines, shapes, colours, patterns and textures. Such images are likely to be computer generated, and are devoid of subject matter. As a result, object identification is not a requirement of this problem.

Figure 3.1 shows some typical examples of the type of abstract images that may be considered in such an application. As can be seen, they consist of varying geometric shapes with different textures applied over regions of the images. The arrangement of these shapes and textures is diverse, so that no assumptions may be made about the structure of the images. There is also a diversity of colour and range of colour in the images.

It is assumed that the images being compared will not be variations of the same image that have been subjected to different lighting conditions, due to their abstract nature. It is however, fair to assume that there may be colour changes across the images.

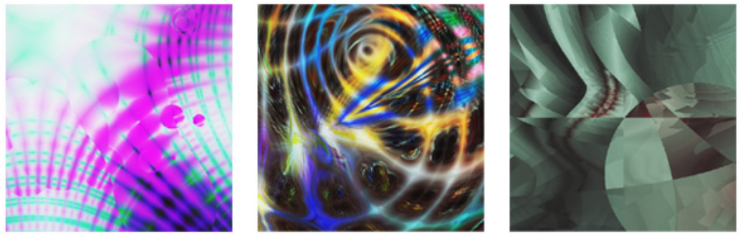

Figure 3.1: Examples of abstract images

Additionally, the metric used for comparison is not required to exhibit either translational invariance or rotational invariance. It is also assumed that the scale of the images is unimportant and that the images are the same size, as a result of possibly being computer generated. The focus of the algorithms should instead be on sensitivity to information provided by formal elements such as colour and texture.

The abstract images used in this research have all been obtained from *www.random-art.org*, which is continuously updated with new computer-generated images. All the images generated on this site and therefore used for this research are abstract, and were all 512×512 pixels in size.

3.5.2 Image Signatures

As has been previously described a signature, or alternatively fingerprint, of an image is a summary of some aspect of an image. These are typically smaller than the original image and have a strict predefined structure. Therefore it is a meaningful exercise to compare corresponding elements of a signature, as those elements refer to corresponding features of the original images, which may not have been easy to compare directly.

The generation and comparison of these signatures has been repeatedly shown in the literature to be an effective way of comparing images. This is useful because the signature is created in such a way so as to contain only specific information that is relevant for a particular context. The signatures of two images can also be compared directly with any of a number of different techniques.

Different signatures can be created by emphasising certain aspects of an image, through the use of techniques such as either colour or texture based methods (the element based methods), although provided the same method is used in each case, any methods can be used to generate the signatures.

Comparison of signatures can also be done by means of many different methods. These include method specific metrics, such as histogram intersection, and statistical techniques such as the χ^2 test. Another important metric that was considered is the Earth Mover's Distance.

An important consideration is how to combine signatures generated by different methods. A conceptually simple method which was used in this research, is to compare the signatures pairwise using one of the aforementioned techniques and then treat the combined result as a linear combination of the individual distances derived from the comparison metrics of these signatures generated by the element based methods. In order to combine the distances computed by the metrics as a linear combination, each distance would be multiplied by a weight. This was the method employed in this research. The weights (coefficients of the linear combination) were determined experimentally. These weights should all be nonnegative and sum to one.

3.5.3 Colour Matching

Many authors assume colour to be an irrelevant factor in comparing images. The algorithms to solve this abstract image similarity problem is intended to determine similarities based on the way in which a human would perform the same task. However, it seems logical that when humans compare objects, one of the primary factors to take into account is colour. It is for this reason that colour differences will be considered important in this research.

Although histograms are a relatively simple technique for storing the colour information of an image, as discussed in Section 2.2.2, they have several fundamental problems. In this application, the paramount of these is that they do not store spatial information. By working in the domain of abstract images, the potential exists for comparisons to be made between several monochromatic images. In this case, the situation could arise where several images contain the same colours in similar frequencies but differing distributions.

Thus, the use of spatial histograms for this algorithm seems necessary. As a result, colour information of an image will be contained within a signature that represents the distribution of the largest groupings of similar colours. The problem with this is that it increases the complexity of what is essentially a simple technique.

By considering examples of the images that could be evaluated for similarity, it was decided that colour information is one of the primary factors in determining similarity. As a result, in the unlikely case of a very different distribution of the identical colours in two different images, these two images should be considered similar and would likely act as effective distractors in the password application. Further distinction would also be made by the inclusion of other factors in the final similarity score of these two images.

Once colour signature generation has been performed on an image, that image can be treated as a grayscale picture for the remaining processes of textural signature generation and matching, and indeed any other factors which could be considered at a later stage.

The colour matching algorithm that was used for this research used a single colour channel for the histogram binning. As presented in the literature, it was decided from the onset that the RGB colour space was inappropriate for the task. An initial investigation was made into the use of the CIE Luv colour space, but ultimately the HSV (Hue, Saturation, Value) colour space was chosen. The reason for this was that the hue (H) effectively represents colours as a single polar angle. This does not take the saturation or intensity of the colour into account, but this was deemed to be satisfactory for this method.

The colour histogram generating algorithm is presented in Algorithm 1. It takes as input an image and the number of bins required, and returns a colour histogram. Each pixel in the image has a hue value between 0 and 1. For each pixel, the algorithm identifies the bin into which that pixel should be classified based on its hue, and the counter for that bin is incremented. The histogram is finally normalised to accommodate for different sized images.

Algorithm 1: Colour Histogram Generation

Input: An image, *image*, and number of histogram bins, *numBins*
Output: A histogram of the hues in the image
begin
  $image \longleftarrow getHue(convertToHSV(image))$
  $histogram[0..n-1] \longleftarrow 0$
  for *each pixel* $p_i \in image$ **do**
    $bin \longleftarrow floor(getValue(p_i) \times n)$
    $histogram[bin] \longleftarrow histogram[bin] + 1$
  end
  $histogram \longleftarrow histogram/norm(histogram)$
  return $histogram$
end

These colour histograms are the signatures that will be compared for any two given images to determine their level of similarity. Although a number of different techniques exist for comparing histograms (see Section 2.2.3), the Earth Mover's Distance was used in this case. This was primarily because it allows for complete cross-bin comparisons. It also allows for the signatures to be of different sizes, which is an aspect that could be examined at a later stage to improve the method.

The ground distance used between two bins numbered i and j for the Earth Mover's Distance metric was initially the difference between the bin numbers. This was then changed to incorporate the fact that the hue of a colour is a polar angle, and therefore periodic. As a result, the ground distance between bins i and j was defined as

$$d(i,j) = \min(|i-j|, n-|i-j|) \tag{3.1}$$

The result of combining the histograms of two images, using the Earth Mover's Distance with the ground distance in (3.1) is a single number α representing the similarity between the histograms. $\alpha = 0$ if the histograms are identical, otherwise $\alpha > 0$, with an increased value of α corresponding to an increased level of dissimilarity.

3.5.4 Colour Perturbations

In order to test the robustness of the colour based Algorithm 1, the implemented method was run on several different perturbation test images, which are presented in Figure 3.2.

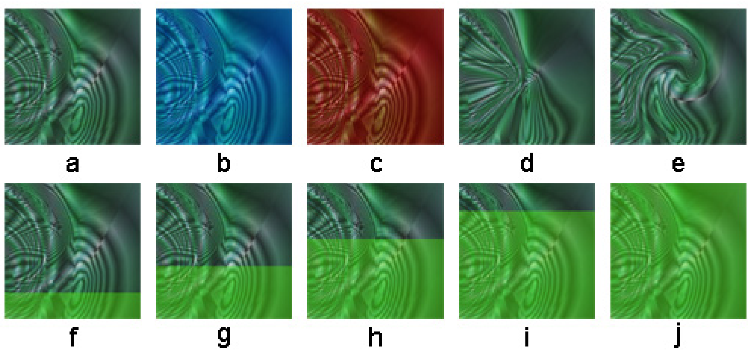

Figure 3.2: Colour Perturbations

This perturbation experiment was conducted as follows. A base image (shown in image a of Figure 3.2) was selected, predominantly because of its use of a narrow palette of different colours. Various changes were made to this base image, and the colour based algorithm was then asked to determine the distance between each perturbation image and the base image. These similarity scores are presented in Table 3.1.

Image	a	b	c	d	e	f	g	h	i	j
Similarity score	0	2.114	6.029	0.020	0.003	0.047	0.184	0.772	1.510	1.978

Table 3.1: Colour Perturbations

The first tests that were conducted were to ensure that colour was the primary factor that is measured by Algorithm 1. To do this, the colour palette of the base image was shifted, first to blue (image b in Figure 3.2) and then to red (image c in Figure 3.2).

Most human observers would agree that the colouration of the base image, which is dark green, is much closer to the blue colouring of image b than the red of image c. This is reflected in the results provided by the algorithm, which give a much larger distance of 6.029 from the base image to the red one than the distance of 2.114 from the base to the blue. This base-blue distance is large in itself, which represents the fact that the colouring of the whole image has shifted by a moderate amount.

The next tests were to ensure that other changes in the structure of the images would not affect the distance as measured by the colour algorithm. Two more perturbed images were created, images d and e in Figure 3.2, by taking various transforms and distortions of the base image. By doing so, the arrangement of pixels in the image was changed, thus changing the structure of the image but preserving the colour.

In terms of colour information, these images are almost identical to the original. This is in agreement with the results of the colour algorithm in determining the distances between these images and the base image – 0.02 and 0.003 respectively. The reason that there are these very slight variations in the colour, is that as a result of applied transforms, several pixels would have been altered slightly due to the compression and stretching of different parts of the image.

The last set of tests were to determine whether or not the distance between two images increases in proportion to the amount of changes in colour. For this test, a translucent bright green rectangle was placed over the base image, covering various percentages of the image. These ranged from 20% to 100% in steps of 20%, and these are shown in images f–j of Figure 3.2.

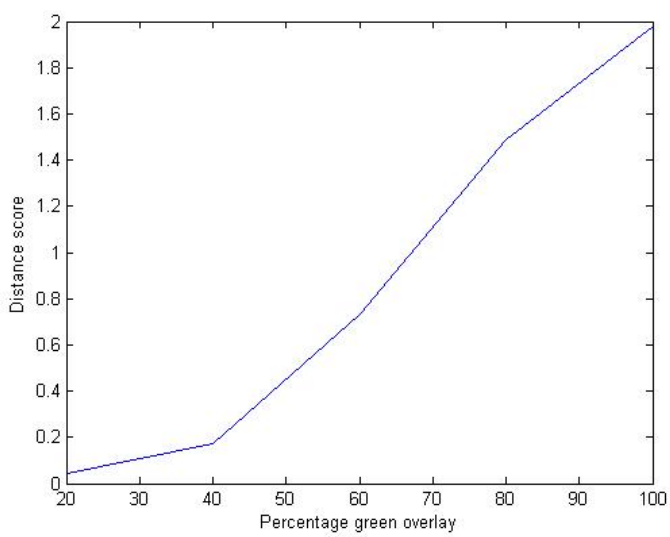

Figure 3.3: Colour distance with increased dissimilarity

The results corresponding to these five cases and also present in Table 3.1 and range from 0.047 for a 20% overlay to 1.978 for the entire image to have shifted in colour. These results increase fairly uniformly, as can be seen in Figure 3.3, which shows a near linear graph. This confirms that the change in colour score is proportional to the change in colour content of an image.

The results in this section show that the colour based algorithm operates exactly as specified in its design.

3.5.5 Texture Matching

Understanding the textures of an image provides insight into the fine, repeated details that exist within that image. Depending on how it was generated, the potential exists for an abstract image to have anything from no texture at all, to consisting entirely of a single, or multiple, textures. However, most of the abstract images considered had a large quantity of textured areas. This was therefore an important element to consider in abstract image comparisons.

As was demonstrated in Section 2.3.1, Gabor filters have been repeatedly shown to be successful at discriminating textures. For this reason, the use of banks of Gabor filters was examined as a method for identifying different textures and textured regions.

Gabor filter banks were developed based on specifications given by Rubner [1999]. In order to ensure a proportional increase in the frequencies of the filters, Rubner [1999] defines the minimum frequency as $U_l = 0.04$ and the maximum frequency as $U_h = 0.3$, and then proceeds to define

$$a = \left(\frac{U_h}{U_l}\right)^{\frac{1}{M-1}}$$

where M is the required number of different frequency scales for the Gabor filters.

Figure 3.4: (a) A Gabor filter (b) Example of a texture signature

The wavelengths for the filters are given as $\lambda = a^{-m}$ for $0 \leq m \leq M-1$ and the orientations are $\theta = \pi l/L$ for $0 \leq l \leq L-1$.

The filter bank was designed to use $M = 5$ frequencies and $L = 8$ orientations, resulting in a total of 40 different filters. An example of the real part of one such filter is shown in Figure 3.4a. This is the filter generated for $l = 7, m = 3$.

Each filter g_{ml} contributed a single value to the signature. This value is the magnitude of the result of the convolution of the filter g_{ml} with the image, and appears in position (m, l) of the signature. An example of such a signature is shown in Figure 3.4b. The two bright vertical bars indicate that the original image had a high content of lines in orientations corresponding to those values of l.

Algorithm 2 presents the algorithm used to generate the texture signatures. It operates on a single image, which is converted to grayscale. Then, for each (m, l) combination, the appropriate Gabor filter is generated, as described in (2.1). The convolution is the actual process of applying the filter to the image, and results in a filtered image. The norm of this filtered image is the amount of information highlighted by the filter, and this becomes the values of the (m, l) component of the signature.

Comparisons of two such signatures were performed using the Earth Mover's Distance, for the same reasons highlighted in Section 3.5.3. Another reason was that this metric can capture the spatial relationship between components of the signatures. Thus, the use of this metric requires that a ground distance between elements of the signatures be defined.

The ground distance for these signatures is based on the fact that the values of (λ, θ) define a cylindrical polar coordinate system, where λ is the cylindrical height, θ is the polar angle, and the radius is fixed.

Algorithm 2: Gabor Signature Generation

Input: An image *image*, the number of orientations *L*, the number of scales *M*
Output: A signature of size $M \times L$
begin
    $image \longleftarrow convertToGrayscale(image)$
    for $m = 0$ **to** $M - 1$ **do**
        $\lambda = a^{-m}$
        for $l = 0$ **to** $L - 1$ **do**
            $\theta = \pi l / L$
            $filter \longleftarrow generateGaborFilter(\lambda, \theta)$
            $signature(m, l) \longleftarrow norm(convolution(image, filter))$
        end
    end
    return $signature$
end

As a result, the ground distance can be calculated as [Rubner 1999]

$$d((m_1, l_1), (m_2, l_2)) = |m_1 - m_2| + \min(|l_1 - l_2|, L - |l_1 - l_2|) \quad (3.2)$$

which is the sum of the distance between scales and the shortest distance between the orientations of the filters used to generate the value at (m_1, l_1) of the first signature and (m_2, l_2) of the second.

Using the Earth Mover's Distance with (3.2) to find the distance between the texture signatures of two images provides a measure of the similarity β of the texture content of the two images. In the case where the textures are identical $\beta = 0$, but as the level of dissimilarity between them increases, so too does β to reflect this change.

3.5.6 Texture Perturbations

The discussion in this section is similar to that of Section 3.5.4 and presents some results to ensure that the properties of the images that are measured by Algorithms 1 and 2 are orthogonal, and also to ensure that the texture algorithm responds as expected to perturbations in the texture of an image.

As was the case for the colour perturbation experiments, a base image was selected. The same base image that was used in the colour case was used for the textures. Several changes were then made to the image. The set of images used is shown in Figure 3.5.

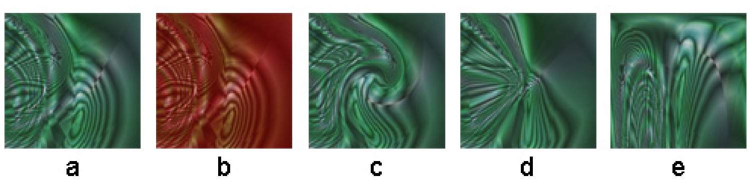

Figure 3.5: Texture Perturbations

The transformations that were applied to these images were intended to show different levels of distortion to the image. The texture algorithm was run on these images and the similarity scores between each image and the base image are reported in Table 3.2.

The first experiment was to determine whether the colour of an image has an effect on the texture. For this, the colour palette of the base image was shifted to red. This is shown in image b of Figure

Image	a	b	c	d	e
Similarity score	0	0.044	0.431	0.519	1.134

Table 3.2: Texture Perturbations

3.5. Changing various parts of the image to different colours would have changed the actual texture of the image and for this reason was not done. Table 3.2 shows that this colour change did have a minimal effect on the results of the algorithm, which reported a distance of 0.044. This is of a similar low order to the results of texture perturbations on the colour method, which indicates the factors they measure are largely unrelated. The fact that this score is not zero is probably due to the fact that a change in colour of an image will result in a slight change in brightness and contrast. This would be reflected on the texture detector, and suggests that other factors which may be incorporated into the image matching is brightness and contrast (although these could be expected to have lesser weighted contributions, as they are partially dependent on both colour and texture).

It was not possible to show gradual change in texture content and plot this as was the case for the colour perturbations, but the changes made to the base image are sorted in Figure 3.5 by the severity of the perturbation.

Images c–e of Figure 3.5 show various distortions applied to the base image, with the colouring held constant. These are sorted by the magnitude of distortion, although images c and d are fairly similar. Image e is definitely the most perturbed. The results in Table 3.2 show that changes to the structure of the image reflect in the texture distance. As can be seen, the results indicate that a change has occurred, and the distance measure correlates with the magnitude of the perturbation. As a result, the texture algorithm shows sensitivity to structural distortions in the images.

3.6 Implementation and Experiments

Once the different algorithms were designed, they were all implemented. All implementation was done in MATLAB. This was for several reasons. Firstly, MATLAB provides a robust platform that has been optimised for matrix operations. As images can be conceptualised as large matrices, it is a natural choice of platform for this application. Furthermore, it has many predefined functions for image handling and image processing, which are easily incorporated into other functions. There is also extensive support for MATLAB, and it is widely used in a number of fields.

Although MATLAB allows for easy implementation, calculations can take a significant amount of time to execute. This was found to be the case for several operations, particularly the convolutions required for the Gabor filters. The speeds of these operations would be greatly enhanced if they were implemented in a lower-level language such as C++.

The different algorithms were implemented as separate functions, with the parameters included as arguments to the functions. In this way, it was easy to change the values of parameters and use different combinations of methods in algorithms. The modular design also greatly simplified the process of combining results and attempting different techniques in tandem.

Following the implementation was the testing phase. The purpose of this was threefold. Firstly, this aimed to provide empirical evidence for the accuracy of the proposed algorithms. Secondly, this was used to determine whether or not a combination of techniques is better suited to determining abstract image similarities than a single technique. Finally, it was also necessary in order to determine weights for the results of the various algorithms when combined. The results of these tests are presented in Chapter 4.

The designed algorithms were tested on various sets of abstract images. The first important criterion was that there should be some images that are very similar to each other, and others that are very different, with varying degrees of difference in between. This degree of similarity was judged by the author, and the degree of similarity between every pair of images was measured with reference to other pairs. The aim was to be as objective as possible, bearing in mind that the judgment of this characteristic would vary between individuals.

An alternative approach would have been to compare the findings of the algorithms presented within this document to the opinions of a wide range of other people. This would involve surveying a large sample of various people to classify the similarities between a large set of image pairs, and compare this to the answers provided by these algorithms in the hope that this would provide a more objective measure of the success of these approaches.

There is however an inherent problem with averaging the responses of many people in an application such as this. It has already been established that the human eye relies on various aspects of images to make comparisons. It is for this reason that multiple factors have been examined in this reason. Different people under different circumstances would make image comparisons based on different contributions from these factors. Therefore one person may regard two images as similar based on colour, whereas another person would disagree and state that those same images are different as they contain different geometric images. By averaging these opinions, one is likely to obtain some results that disagree with the opinions of both of these people.

An application such as the graphical password system is required to produce results that agree with the perception of people as individuals, rather than the average of a group. Thus the system should not be permitted to produce incorrect results for a single individual. Consequently, the decision was taken that it would be sufficient to compare the results of the algorithms to those of the author, and not the average of a group of people.

The images selected also represented a broad spectrum of variability in the criteria measured by the various algorithms. Therefore the sample requires images with different amounts and distributions within a range of colours, assorted texture intensity and number of textured regions, as well as differing levels of structural complexity. Such diverse images were selected to assess the robustness of the algorithm with respect to varying conditions.

3.7 Conclusion

This chapter offered insights into the actual research that was conducted. The problem of determining the degree of similarity between two abstract images has been shown to be an important one, and has immediate applications in the development of novel password systems. It is also not a trivial problem, and requires an analysis into the way in which humans perceive pictorial differences in terms of the formal elements of an image.

This research required the development of algorithms to quantify image differences using colour and texture based matching techniques. These proposed solutions were then tested on different sets of sample images to ascertain their accuracy and determine which of them are more adept at solving the posed question.

The implemented algorithms relied on several different methods, which were combined to attempt to obtain an overall understanding of the composition of an image. The aim was to achieve a final algorithm which would be able to rank image similarities based on the properties of an image, and achieve similar results in this regard to ones produced by humans.

The following chapter presents the results of experiments that were run using the developed, implemented algorithms as well as the actual images that were used to test them. These results are used as the basis for evaluating the extent to which the algorithms implemented in this chapter could be considered successful. In addition, Chapter 4 presents evidence of the improvements that were obtained by combining these proposed solutions.

Chapter 5 then provides a full discussion and analysis of these results, as well as additional results that demonstrate strengths and flaws of the algorithms described in this chapter.

Chapter 4

Results

4.1 Introduction

The previous chapter discussed the development and implementation of the various algorithms that were used in this research. The algorithms based on a colour histogram method as well as a Gabor filter texture based algorithm were implemented in MATLAB. Both of these resulted in the generation of signatures, which were then compared using the Earth Mover's Distance metric.

Having implemented these algorithms, it was important to examine the results when they were tested on abstract images. This would enable an evaluation of their performance, in terms of their accuracy when compared to human similarity judgments.

This chapter illustrates the results that were obtained by running the aforementioned algorithms on various abstract images. Section 4.2 explains the approach that was taken when comparing the results to human perception. This also provides a basis against which the accuracy of the different algorithms can be compared. Section 4.3 then proceeds to present the results of the colour based algorithm when run on the test images. Similarly, Section 4.4 presents the results of the texture based algorithm. Finally, Section 4.5 gives the results of the combination of the two approaches, and also discusses issues such as the weighting of the two component results.

A full analysis of the results in this chapter appears in Chapter 5.

4.2 Comparison of Results with Human Perception

The results generated by the implemented solutions were required to be compared against human similarity perception for two reasons. Firstly, an aim of this research is to develop a technique that is capable of reproducing the human similarity metric to some extent, and so it is vital that these two approaches be directly compared. Another reason for doing so, is that the results of any two different techniques should be compared in some meaningful way. As the object of this exercise would be to find the one which produces the more accurate results, comparing the measure of success that one method has in simulating human judgment with that of another method seems a natural way of evaluating which is a better technique.

It should be stated from the onset that the method used to do this was somewhat subjective. However, it was done in a way that would hopefully minimise the effect of any bias from the author.

For each pair of images that were considered by any algorithm for similarities, that pair was also considered by the author. They were then given a rating on how similar they appeared to the author. Squire and Pun [1997] showed that the individual responses of humans varies greatly in determining the extent to which images could be considered similar. Therefore every person would be expected to present a unique bias to the task of determining similarities. However, to attempt to reduce the effect of this bias, the classification by the author was either a ‘Y’, ‘M’ or ‘N’. These correspond to ‘Yes’, ‘Maybe’ and ‘No’ and act as the possible answers to the question “Are these two images similar?”

By the mathematical properties of a metric, the distance between two signatures is always greater than or equal to zero. Additionally, the distance between identical signatures is zero (reflexivity) and the distance increases as the level of similarity between the images decreases. Thus, two images with a large distance between them are more dissimilar than two images with a smaller distance between them. It is thus logical to interpret a numerical distance score provided by a method as falling into one of the three categories: ‘Y’, ‘M’ or ‘N’. This was done by means of a simple thresholding function, where the category σ was determined for a similarity score x generated by one of the algorithms, as

$$\sigma(x) = \begin{cases} Y & \text{for } 0 \leq x < 0.4 \\ M & \text{for } 0.4 \leq x \leq 0.8 \\ N & \text{otherwise} \end{cases}$$

where the threshold values were determined experimentally.

The intention was that by utilising such a crude scale of similarity, the effect of the human bias of the author would be reduced to what could hopefully be considered reasonable approximations to the level of similarity between two images when judged by any other person. The fact that the scale provides a positive, negative and unsure result to the question of similarity, would still make it useful in determining some estimate of the level of success of one of these proposed solutions at replicating human judgment. Furthermore, it could be used to identify false positives and false negatives in the results of the algorithms.

To determine the success of an algorithm on a large data set, one could count the number of times that the algorithm classified a similarity in the same way a human would. Note that a ‘maybe’ was considered to match a ‘yes’ for purposes of this research. The number of times that a human classified the pair of images as not matching, with disagreement from the algorithm, was the number of false positives generated by the algorithm. Similarly, the number of times the algorithm claimed a pair of images were not similar and this was disputed by the human, was the number of false negatives of that algorithm on that data set.

4.3 Colour Results

The set of colour abstract images that were used to test the algorithms is shown in Figure 4.1. Most of the images were either pinks and purples, or greens. The most obvious exception is image 8, which is yellow. Although many of these images have similar colouring, it is apparent that this is not the only important factor, as many are still fairly dissimilar in overall appearance.

These images in Figure 4.1 formed the basis of the experiments that were performed. Each pair of these images were tested for similarities in the colour based method given in Algorithm 1. The number of histogram bins used in each case for the generation of the signatures was 16.

As would be expected, the results from every algorithm gave the distance between identical images as zero.

The first aspect of the colour matching results that was examined, was the closest match in the set. This is, for each image i, the image j in the set which measured the smallest distance from i (such that $i \neq j$, which is the trivial case). These closest matches are given in Table 4.1.

Most of these results behave as one might expect. For example, the closest match of the yellow image 8, is 6 which is the only other image to contain substantial amounts of yellow. Another intuitive result is that the closest match to image 5 is image 4.

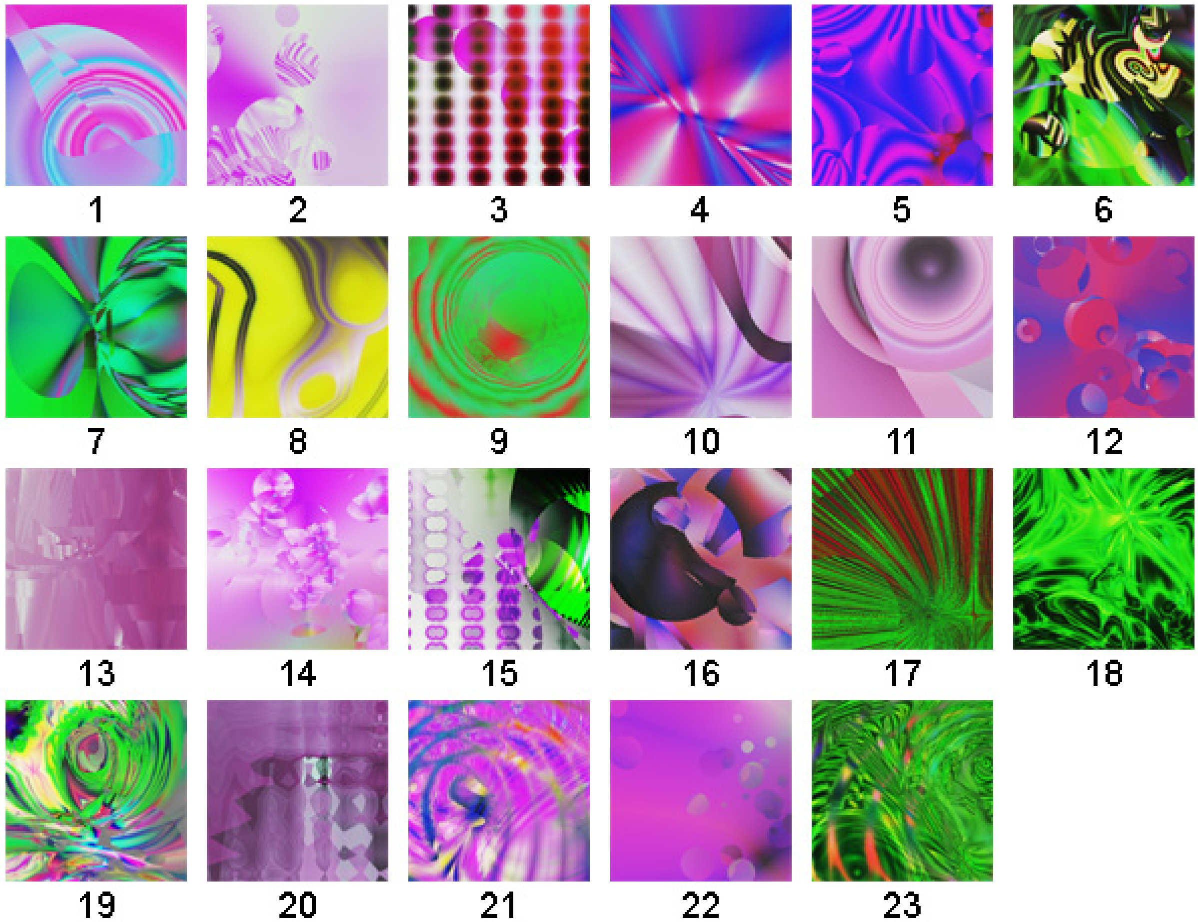

Figure 4.1: Set of images that were used to test the methods

The images 11 and 13 provide an example of a pair of images which have similar colours but these colours take on very different distributions in the images. This demonstrates the criticism of histogram methods that they do not incorporate spatial information (see Section 2.2.2).

Some image pairings do not seem to agree with human judgment. Examples of this are that the similarities between images 4 and 22, and images 12 and 11 are not apparent. The most likely reason for these is that the fact that the histograms only operate on the hue channel means that the intensities of the colours are not taken into account. The result of this, is that a very intense area of a colour may

Image	**1**	**2**	**3**	**4**	**5**	**6**
Closest Match	20	21	14	22	4	23
Image	**7**	**8**	**9**	**10**	**11**	**12**
Closest Match	19	6	18	20	13	11
Image	**13**	**14**	**15**	**16**	**17**	**18**
Closest Match	11	21	18	12	18	9
Image	**19**	**20**	**21**	**22**	**23**	
Closest Match	9	10	22	21	18	

Table 4.1: Closest colour matches

match identically with a washed-out version of the same colour in another image. Another important consideration, is that some of these distances are much further than others (for instance, the distance between images 12 and 11 is 0.3145, whereas the distance between images 11 and 13 is 0.1567).

An observation from Table 4.1 is that the results are not symmetrical. An example of this is that image 20 is shown to be the closest match for image 1, but image 10, rather than image 1, is the closest match for image 20. The reason for this is simple. The colour distance between images 1 and 20 is 0.246 and the distance between 10 and 20 is 0.049. Therefore, although the closest match to image 1 is image 20, image 20 is actually far closer to image 10 than image 1. This is confirmed by the fact that Table 4.1 shows that image 20 is the best match for image 10.

The complete set of colour distances for the images in Figure 4.1 is tabulated in Appendix A, for each pair of images.

A comparison of the converted table of 'Y', 'M' and 'N' values generated automatically by the algorithm was compared with a similar table generated manually by the author, as described in Section 4.2. The result of this comparison, was that the colour algorithm achieved a 72% successful matching rate. Additionally, 25.3% of the rankings given by the colour matching approach were false positives, meaning that the algorithm incorrectly classified a pair of images as being similar. A further 2.7% of image pairs were classed as false negatives, which means that the algorithm failed to detect their similarity.

4.4 Texture Results

The same set of images were used to test the texture based approach given in Algorithm 2 as was used for the colour algorithm, i.e. the images shown in Figure 4.1. The texture analysis was however performed on the grayscale versions of these images. For performance reasons each signature was generated once for each image, and then these signatures were repeatedly compared.

Again, for every pair of images, the distance was nonnegative, with a zero distance reserved only for the comparison of an image against itself. The distance increased as the difference in textures as detected by the Gabor filter bank increased.

Image	**1**	**2**	**3**	**4**	**5**	**6**
Closest Match	18	16	15	6	14	16
Image	**7**	**8**	**9**	**10**	**11**	**12**
Closest Match	12	6	5	11	10	21
Image	**13**	**14**	**15**	**16**	**17**	**18**
Closest Match	20	22	3	15	23	7
Image	**19**	**20**	**21**	**22**	**23**	
Closest Match	20	22	19	20	2	

Table 4.2: Closest texture matches

Table 4.2 presents each texture together with the texture that was deemed most similar to it by the texture algorithm (smallest distance). Far fewer of the results in this table seem intuitive. An example of a match that agrees with human perception is the pair of images 3 and 15. Both have a very characteristic grid of repeated circles, and this is detected by the algorithm. The full set of texture distances for each pair of images in Figure 4.1 is given in Appendix B.

As with the colour results in Section 4.3, the closest matches in Table 4.2 are not symmetric. The same justification applies. Although, for example, the closest match to image 13 is image 20, image 22 is actually more similar to image 20 than image 13, hence the unsymmetrical results.

As can be seen by examining the results in Table 4.2, the correlation of the results of the texture algorithm and human perception were not as accurate as they were for the colour algorithm. The reasons for this are discussed in Chapter 5. Again, the classification of image pairs by the algorithm as 'Y', 'M' or 'N' were compared against the same classification by the author. The result was that the algorithm only achieved a 63% success rate at classifying images. Of all the image pairs considered, 30.2% were false positive matches and the remaining 6.8% were false negatives.

4.5 Combination

In order to test the second hypothesis, it was necessary to obtain results for the combination of the two approaches. This was done as a linear combination of their independent results, and provided the overall match γ as

$$\gamma = w_1\alpha + w_2\beta \tag{4.1}$$

where α is the colour distance between two images as computed in Section 3.5.3, β is the texture distance between the two images computed in Section 3.5.5, and w_1 and w_2 are the weights assigned to each.

The aim of a combination of these results was to determine if improved results could be obtained – a distance measure that was more accurate than either of the constituent scores.

4.5.1 Weighting

The weights were chosen subject to additive and nonnegative constraints

$$\begin{aligned} w_1 + w_2 &= 1 \\ w_i &\geq 0, \qquad i = 1, 2 \end{aligned}$$

These weights were chosen experimentally by varying the contribution of the colour distance in 10% increments from 0% to 100%. These results are reproduced in Table 4.3 and are plotted in Figure 4.2.

w_1	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
w_2	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0
Percentage matches	62.9	74.3	78.1	80.3	80.7	81.9	80.7	75.8	73.9	71.6	72.0

Table 4.3: Effect of varying weights

The left endpoint corresponds to the situation where the sole contributing factor is the texture distance, and the rightmost endpoint is the distance between the images comprised by the colour distance exclusively. As can be seen in these figures, a combination of the two factors can greatly improve the discriminating power of the algorithms. The most effective solution comes when the weights w_1 and w_2 are approximately both equal to 0.5.

4.5.2 Results

As shown by Figure 4.2 in the previous section, an optimal result is achieved for $w_1 \approx w_2 \approx 0.5$. This situation obtained a higher successful match rate than either component individually when compared to human classification of the similarity of image pairs. The results for this scenario show that 81.9% of image pairs were successfully classified, with 13.2% false positives, and the remaining 4.9% of image pairs were incorrectly classified as false negatives.

These results show a very significant improvement over the texture algorithm by itself, and even a large improvement over the individual results of the colour algorithm.

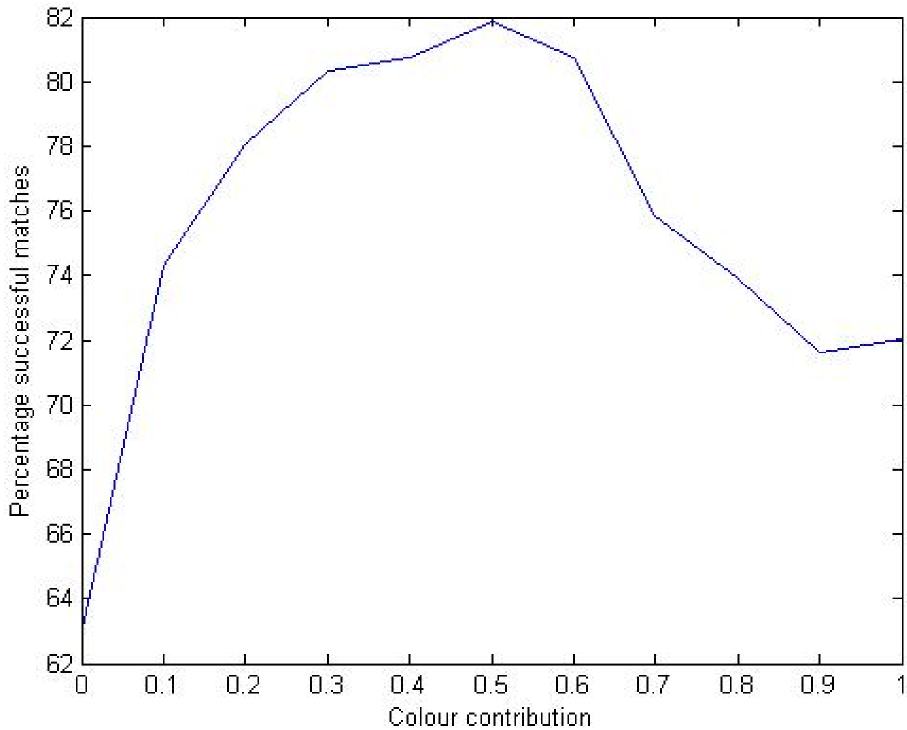

Figure 4.2: Effect of increasing contribution of colour on combined results

4.6 Conclusion

As has been shown in this section, both the colour and texture distances computed by Algorithms 1 and 2 respectively reflect some aspect of human similarity judgment, although the results of the colour approach were significantly more convincing.

Taking a linear combination of the two results produced an improved distance measure between the pair of images – one that obtained a higher success rate when compared to the classifications that had been performed by the author.

The following chapter presents an analysis of the results obtained in this chapter, as well as a discussion of the robustness or pitfalls of various techniques and the results they produced.

Chapter 5

Discussion and Analysis

5.1 Introduction

The previous chapter presents the results that were obtained when running the algorithms developed in Chapter 3 on the images in Figure 4.1. This chapter aims to provide some insight into the workings of the algorithms by examining different situations they encounter, and the responses they generate in each situation.

An investigation into both the colour and the texture solutions is presented in Sections 5.2 and 5.3 respectively. These discuss the successful and unsuccessful results produced by these approaches, and offer insight into the reasons for these varying degrees of success in the results. This discussion is continued into Section 5.4 which demonstrates how the matching ability of the algorithms improves when the results of the two are considered in conjunction.

Finally, Section 5.5 comments on the extent to which the hypotheses of this research could be accepted. Section 5.6 then looks at some of the inherent limitations of this research and Section 5.7 concludes by extrapolating from those points to pose some issues and approaches that could be considered in future work into this research topic.

5.2 Colour Findings

As was seen in the previous chapter, the colour based similarity approach achieved a success rate of approximately 70% at classifying images in a manner similar to a human. This high correlation with the classification of image pairs which was completed by the author suggests that colour information plays an important role in the identifying of images in human vision – particularly abstract images.

A likely reason for not having achieved an even higher success rate is the limitation of using just a single colour dimension in the colour matching algorithm: the hue channel. To further improve the results of this method, it may be worthwhile to investigate the use of the full three dimensions of colour information.

As was mentioned in Section 4.3, some of the results seem more perceptually correct than others. However, when comparing the classification of the similarity of each pair of images as “yes”, “no” or “maybe” none of the closest matched pairs are classified incorrectly. Therefore there may be some images that seem better choices for a closest match, but all the closest pairs that were computed by the algorithm are to an extent similar. As a result, the images can be considered to have been clustered by the algorithm into two main categories – the pink images and the green images.

There are several reasons for why a pair of images that are the closest match to the human eye may not have been selected by the algorithm. Firstly, the fact that only hue is considered is potentially an important factor, even though hue is arguably the most significant component of colour. An example is a situation where two images may contain a large amount of a very pale colour. To the human eye these could both look white and hence be very similar, whereas one may be a very pale red and the other a pale green which would result in a large difference.

Another factor that could influence the matching is the choice of metric. The Earth Mover's Distance is a very robust metric that, through the use of the ground distance, can take advantage of the fact that the signatures exist in a specific space and there are defined spatial relationships between them. However it could be the case that for some instances one would not want this information to be taken into account, or that it would be used in different ways. In this case another metric may be more appropriate. Alternatively, the ground distance for the Earth Mover's Distance could be changed so as to drastically increase the cost of matching nonadjacent histogram bins. This is all grounds for further research.

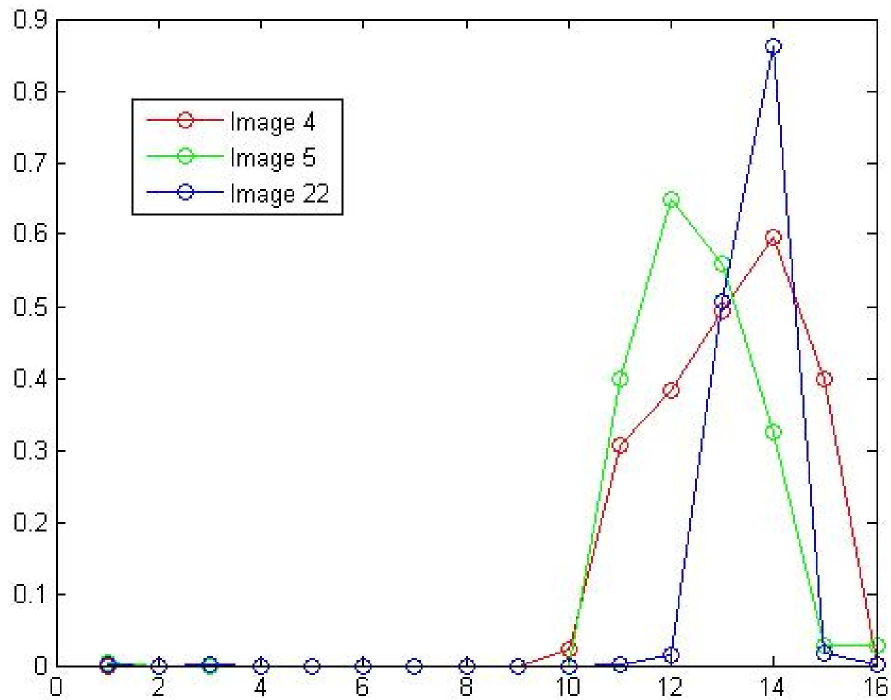

Figure 5.1: Comparison of colour histograms

Figure 5.1 shows the colour histograms for images 4, 5 and 22 in red, green and blue respectively. As was seen in Table 4.1, the closest matching image to image 4 was image 22, rather than image 5 which may seem more intuitive. The histogram for image 22 is a sharp spike over the bins 13 and 14. This corresponds to a high content of pink and purple, with very little else. On the other hand, image 5 additionally has much colour content with falls into bins 11 and 12, corresponding to a prevalence of blue in the image.

When viewing the histogram of image 4, it is seen to have a similar mound shape to that of image 5 and consists mainly of the colours that fall into bins 11 to 15. Thesefore this histogram is very similar in shape to that of image 5, as would be expected by their similar colour content. However, the peak in the histogram of image 4 coincides with that of image 5, as the largest percentage of the colour which comprises both of these images is the purple of bin 14. As a result, the algorithm regards this pair as more similar than images 4 and 5. Conversely, image 5 can be seen to be more similar to image 4 than image 22, resulting in image 4 being the closest match for image 5.

Although the choices of the closest matching pairs do not always seem intuitive, when compared to the human classification of the image pairs falling into one of the three categories, the colour algorithm performed well. This shows that the algorithm is capable of distinguishing a pair of similar images from a pair of dissimilar images. The results achieved by doing so bear strong resemblance to human visual results.

5.3 Texture Findings

The results from the previous chapter show that the texture algorithm was capable of a success rate of approximately 60%. Although this shows better than random performance, it is not significantly better. However, a combination of these results with that of another factor, colour, provided a significant improvement to the accuracies of both – increasing to over 80%. The fact that an improvement was shown was independent of the weightings used to combine them.

These findings suggest that the texture information may not be a primary factor to take into account when comparing abstract images. However, it certainly improves on the results provided by a more prominent factor such as colour. It could be argued that humans are not adept at comparing texture information in images. As the aim of this research is the development of a method for comparing images in a similar way to humans, this inability to match textures would imply that it is not as important a factor as colour.

Many of the images chosen by the algorithm as the closest matching pairs are far from the obvious choices, and similarly many that display similarities are considered dissimilar by the algorithm. The reasons for this are discussed in Section 5.3.1. It is however encouraging to note the similarities in images 3 and 15, which were recognised by the algorithm. Both these images display a similar repeated circular pattern over a majority of the image. This pair is considered very similar by the algorithm, in accordance with human perception.

Although images may be regarded as similar in terms of texture, this is not an easily visible aspect of the images (particularly in the small thumbnails of Figure 4.1) and as such would not be expected to contribute a large amount to the human visual comparison of these images. This observation provides some explanation for the fact that colour performed better than texture on an individual basis.

5.3.1 Difficulties of Gabor Method in Abstract Context

Although it initially seemed that the Gabor filter bank for texture matching of these images would be a successful technique, it proved to be inadequate for the task.

Gabor texture banks work particularly well for determining the existence of similar textures in two different images. For this reason it was used successfully by Rubner [1999] to determine if an image contained leopards, or alternatively cheetahs, and it provided very few false matches.

The difficulty of working within the context of abstract images is twofold. Firstly, and most importantly, the filter determines the existence of a particular texture, rather than positioning and other spatial and topological information. This downside to the technique can actually be observed in the same results of Rubner [1999]. Given a query texture (cheetah spots) the filter was able to extract images from a database that contained the matching texture. Although the filters may have been successful in this task, on closer inspection of the extracted images, none of them were compositionally alike in any way although they all contained a texture similar to the query.

This is the problem shared with this texture algorithm. Although the images considered by the algorithm as similar may contain matching textures, the images are not guaranteed to look alike. This problem is aggravated by the fact that a large number of varying textures appear in the images. As a result, many different textures may appear in a single image. Alternatively, several very similar textures may occur in a wide range of images that differ in every other aspect of their appearance.

As a result, the Gabor method works well in regions of homogeneous texture, or for searching for the occurrence of a particular texture, but its limitation is fundamentally a lack of texture positioning information. This is particularly pronounced in this application of image similarity, with potentially highly irregular structuring and texture in the images. To improve the results of the Gabor algorithm and overcome this fundamental flaw, it would therefore be critical to review the way in which it summarises information from an image and generates the corresponding texture signature. An ideal solution would be for the signatures to contain information about the positioning of the textures within the images.

5.4 Combination of Techniques

Despite weaker individual results from the Gabor texture matching technique, this research showed that an improvement was obtained by using a combination of the two factors of colour and texture. This can be seen in the results displayed in Table 4.3, where an accuracy of 81.9% could be achieved by combining results of the colour and texture methods, which scored 72% and 62.9% respectively.

This table suggests that the optimal results are achieved with a 50% contribution from each of these distance measures. This seems counter-intuitive considering that the colour algorithm not only performs better experimentally but, as discussed above, is also fundamentally better suited to the problem of abstract image matching. The answer to this is that there is extra implicit weighting in the colour results. This can be seen by considering the distances this algorithm generated, in Appendices A and B. For some of the more dissimilar images, the maximum of the distances reported by the colour algorithm were over 2.3 times larger than those produced by the texture algorithm. Thus the colour algorithm was actually more heavily weighted. Furthermore, by considering the success rates of the colour and texture algorithms independently, compared to the optimum combined result, it is easy to see that the texture algorithm benefitted far more from the inclusion of the colour results, than vice versa.

The graph in Figure 4.2 confirms this argument. As can be seen, the leftmost point on the graph ($x = 0$) corresponds to the case where the image distance consists purely of the texture component. Similarly, the rightmost point ($x = 1$) shows the accuracy for the colour results without any contribution from the texture factor. A greater value of x therefore corresponds to a heavier weighting of the colour component and thus a smaller contribution from texture. The fact that colour matching alone performs significantly better than pure texture matching is apparent by examining the endpoints of this graph.

By studying the curve of this graph starting at $x = 0$ as the colour content is gradually increased, the immediate sharp improvement in the performance of the combined algorithm is noticeable. A colour contribution of just 10% increases the success rate of the algorithm by 11.4%. On the other side of the graph, it can be seen that a small contribution of texture to the colour results do not significantly improve accuracy. Indeed, with the addition of a texture contribution of 30%, the accuracy of the method increased by only 3.8%, compared to the accuracy of the colour algorithm alone. It is only with a fairly large contribution of texture that the results are enhanced. Therefore, there is a sharp improvement to the texture scores by adding colour, but only gradual improvement to the colour when texture informtion is added.

These findings confirm the suggestion that the colour aspect of images is a more important discriminating feature of abstract images than texture. However, the fact that a combination of the two factors does improve the results of the algorithms show that the two methods consider factors that are largely unrelated and measure a different aspect of the images – both of which are responsible to some extent for human image identification.

It should be noted at this point that when considered by a human, the similarity of various images is a relative, rather than an absolute, measure. Two images that seem similar in a particular set of images may appear less so when placed in the context of another set of images. In this case, that same pair could appear quite dissimilar. This is in contrast to the results produced by this combined algorithm, which consider image similarities as an absolute measure – it is always the same. However, by incorporating the different aspects of the images as is done by the combined algorithm, it is hoped that the total score will accommodate any method used by a human for comparisons of the images.

5.5 Results of the Research

The hypotheses presented in Section 3.3 attempted to establish two things. The first was to determine if a system could be developed to rank similarities between images using colour or texture information to reproduce the results obtained had a human ranked the same images. The second hypothesis set out to determine whether or not the system would be improved by considering additional factors corresponding to the formal elements of an image.

The first hypothesis was tested by examining results of both the colour histogram method as well as the Gabor texture method when compared with the human-generated results. Even though the texture results were not entirely convincing by themselves, both methods provided positive correlations with the ranking obtained from a human. The combination of these techniques showed an even stronger correlation. Thus this first hypothesis can be accepted.

The second hypothesis considered how the results would change if multiple features of an image were analysed. This was done using a linear combination of the results produced by each algorithm, using nonnegative weightings that summed to one (Section 4.5). With the exception of one point, the case of a 90% contribution from colour and the remaining 10% from texture, all combined results showed an improvement over the individual colour and texture scores. This one exception was a minor one, and could be due to experimental error. The optimal success rate in the accuracy of these algorithms of 81.9% was obtained with such a combination. The fact that the results could be improved in such a way suggests that the second hypothesis should also be accepted.

5.6 Limitations of this Research

Although the major findings of this research described throughout this and the previous chapter seem conclusive, there were several limitations in the research process which may have ultimately affected the work and its results.

The first limitation is the subjectivity involved in the human rankings. As was discussed in Section 4.2 all the categorisations of the image pairs as ‘similar’, ‘dissimilar’ or ‘not easily decidable’ were done by the author. The fact that these decisions were made by the author leaves room for personal bias in the way in which the images were compared. The reason that such a crude scale was used was to attempt to compensate for this bias, and reduce subjectivity.

The discussion in Section 3.6 explained that conducting a survey of a number of people and averaging the results would not necessarily be an appropriate way to determine the accuracy of a similarity method. This is largely due to the fact that different people may use completely different criteria for performing comparisons and thereby arrive at very different results. Furthermore, a graphical password system is required to simulate the opinions of an individual rather than a group. This argument justifies the decision that image similarities were rated by the author to be compared to the ratings given by the algorithm because, to be valid, any algorithm must perform well when compared to the opinions of a single person before it can be evaluated against a group.

Another problem with this research is that only two factors, namely colour and texture information, were taken into account in this research. Unfortunately, the allotted time did not allow for further experimentation with different approaches. However, the trends that were shown in this work proved the concept that the addition of multiple factors improve the results. It would therefore be expected that the incorporation of other factors different to those already considered would further improve the success rate of the combined algorithm.

5.7 Future Work

In light of some of the limitations of this research that were highlighted in Section 5.6 this section provides some recommendations for further work that could attempt to determine the impact of these limitations. Furthermore, some additional ideas are proposed which could enhance the findings of this research and guide further work in this field.

The first recommendation would be to compare the findings of these algorithms with the opinions of various other people in order to gain a wider human comparison. This could be done by the distribution of a survey to a large number of randomly selected people, in order to reduce any bias that may be brought about by querying a group expert in a particular topic [Squire and Pun 1997]. This survey would require people to indicate the level of similarity that they perceive between various images. These results should not be averaged, but instead compared against the algorithm on an individual basis. The reason for this is that, in the application of a graphical password system, the image similarities should coincide with the opinion of an individual and not a group average.

The next avenue for continued research into this field is additional testing of the colour based method which could be performed by incorporating different aspects of the colour, such as saturation or intensity. This could be accomplished by the addition of extra colour components of the HSV space (S and V). Furthermore, the use of different colour spaces, such as CIE Lab or CIE Luv, could be tested in order to determine which colour space presents the closest match to human colour perception.

All these changes would involve extending the histogram method to operate on more than one colour layer. The resulting histograms would therefore be multidimensional – an extra dimension for each additional colour component considered. These histograms could still be compared using the same metrics, but the calculated distance would better represent the images. However, as very accurate results have already been achieved with the colour algorithm, further increases in accuracy by adding more colour factors are not expected to be significant.

To improve the ability of the texture based algorithm to identify similar images based on texture, it may be useful to consider the inclusion of spatial information into the algorithm, by examining the positioning of textures within the image. This could possibly be achieved by, instead of just storing the magnitude of the texture energy detected by each filter, dividing the image into regions and storing the amount of texture information in each of these regions in the signature.

Thus a grid of different sizes (for example 4×4) could be placed over each image, and the texture algorithm run on each of these subimages. The signatures generated on each of these could then be combined into a single three dimensional signature, which would then contain information about regional texture in the image. A comparison of these signatures is likely to better represent texture content.

Further experimentation with metrics other than the Earth Mover's Distance may prove insightful. This specific metric was chosen for several reasons, such as the ability to incorporate a ground distance between various elements of a signature and so take full advantage of the structure of the signature. Different metrics will however have different advantages. It would be interesting to note the results generated with algorithms implemented using a wide range of metrics as each would place a different emphasis on aspects of the signatures.

Another approach that should be taken in future research is considering algorithms to compare different factors in the images and determine if these improve the distance measure or not. Some factors would have a larger effect on the similarity than others but, as was demonstrated in this research, the incorporation of additional factors is likely to improve the accuracy of an algorithm. The extent of this improvement with the addition of an extra factor is dependent on how important that particular factor is in human perception, as well as how much additional information that factor provides. A factor that is orthogonal to the others considered so far (ie colour and texture) would be desired, as this would provide more information than a factor which overlaps information already used. For example, comparing the number of red pixels in an image provides no new information, as this is already considered by the colour algorithm.

The most obvious choice of factor would be shape matching. This could be considered as a comparison of the underlying geometric shapes in an image, without the presence of colour or texture. Shapes could be classified in different ways. An edge detection algorithm would be able to separate different shapes within an image. Another approach would be to perform a Fourier analysis and examine shapes in the frequency domain. A further possible expression of the shapes in an image would be to vectorise the image, as this is a representation which describes the lines of an image. The difficulty with all these methods would be the generation of a compact signature which could summarise this information effectively, and then be compared.

A final thought for future work into this field is that the generation of distractor images may actually benefit more from the separation rather than the combination of different factors. It has already been seen that humans may compare images based on numerous different factors, and each person may place more of an emphasis on a certain factor than other people. Consequently, attempting to incorporate as many factors as possible into an algorithm may result in a similarity measure that provides results which are too specific in their criteria and so do not agree with a large number of people.

A solution to this problem would be for certain percentages of the distractor images to match the actual password on various criteria. For example, it may be required that 80% of the distractors match the colour of the actual password, 40% match the texture, 55% match in geometric shapes, and so on. The advantage to this approach would be that different distractor images would be similar to the password for various reasons, and the probability that any are overspecified then decreases. This approach would hopefully overcome this potential problem, but this should be the subject of further research.

5.8 Conclusion

This chapter presented a discussion of the results obtained to Chapter 4, as well as an analysis of the performance of the techniques that were specified and implemented in Chapter 3.

It was shown that although the Gabor texture bank approach was largely inappropriate for solving the problem of abstract image similarities it did still provide positive results which correlated slightly with image classification by humans. On the other hand, the colour matching technique was shown to display results which were significantly more accurate.

Taking a combination of the results generated by the two algorithms was ultimately the best predictor of image similarities judged by a human. Thus, even with the relative weakness of the Gabor method, when added to the colour algorithm it allowed for an approximate 10% increase in the success rate of the method.

The results of this research provide enough evidence to allow for both hypotheses to be accepted, thus completing the initial goals described as the beginning of the research. This chapter also showed some limitations of this research, namely the potential for subjectivity of human results as well as the fact that only two elements of the images were used for the comparisons.

The future work outlined some interesting challenges that still remain to be considered on this issue of abstract image similarities. These recommendations were largely in response to the limitations of this work. Further research should examine the inclusion of various other factors into the image comparison, such as shape information, and should also obtain human-judged similarity scores from a variety of people for the image pairs.

Chapter 6

Conclusion

Quantifying the degree of similarity, or conversely dissimilarity, between different images is a task which humans handle on a daily basis with great proficiency. An algorithm capable of the same task would be extremely valuable in various fields including computer science, and thus this research aims at developing such an algorithm. The particular scope of this research involves restricting the domain of the problem to abstract images. This has specific applications in the development of alternative password and security systems which would use abstract images as passwords that are easily recognised by the correct user, but difficult for an eavesdropper to identify based on a verbal description. The lack of recognisable objects in an abstract image suggests the use of image processing techniques based on matching the formal elements of two images.

Chapter 2 presents a literature review which discusses some of the most successful approaches that have been developed in the field of image matching. These are broken down into colour based methods and texture based methods. These methods rely on information from various levels of an image: colour information is stored at the individual pixel level, whilst textures are defined over localised regions in the image. The value of such techniques is derived from the fact that these elements represent important formal aspects of an image. From the beginning, focus is placed on selecting methods which could generate signatures that summarise some aspect of the image. Advantages to this approach include the fact that comparing signatures allows for a direct comparison of some inherent components of the images.

The background literature also provided exposure to several metrics that are used specifically for comparing signatures generated by these colour and texture based methods. The collection of algorithms presented in this chapter provides a broad range of tools and techniques that can be combined in various ways to develop a solution to the problem posed in this document.

The research question that this work aimed to answer is presented in Chapter 3, along with the specific hypotheses which suggest that it is possible to develop a system for quantifying abstract image similarities by generating and comparing signatures of images based on the formal elements of those images, and further that the power of such a system would increase with the incorporation of additional factors. Initially, two algorithms were presented to solve this problem: one based on a histogram approach to determine the similarities in the colour content of images, and the other used a Gabor filter bank to characterise the texture content of different images.

These algorithms were implemented, and then run on a database of differing abstract images, to rank the similarities between those images. These results were presented in Chapter 4. A comparison of these results with an equivalent characterisation by a human observer showed that the colour based algorithm was more successful in providing results that conformed with human judgment.

Analysing these results in Chapter 5 showed that even though, when considering their relative accuracies, the colour approach vastly outperforms the texture method at the task of matching abstract images, a combination of the two algorithms produces a ranking that is superior to either of the individual approaches. This confirms both hypotheses of this research.

The value of this research lies in the fact that the specific domain of the problem has not been previously considered. Furthermore, the approach that was taken in testing the hypotheses and developing this algorithm, relied on forming a combination of different techniques that have all been proposed for different applications by different authors.

The final aim was to produce an algorithm that was capable of quantifying the similarities between two abstract images, with results that replicate a human perception of these similarities. Although this was not fully achieved, this research showed that this specific goal is ultimately realisable, and the incorporation of additional formal elements of the images, such as shape, structure or the position of a texture within the image, would aid in the final implementation of this desired system.

References

[Berwick and Lee 2004] D. Berwick and S.W. Lee. Spectral gradients for color-based object recognition and indexing. *Computer Vision and Image Understanding*, 94:28–43, 2004.

[Chen *et al.* 2002] J. Chen, T.N. Pappas, A. Mojsilovic, and B. Rogowitz. Adaptive image segmentation based on color and texture. *Proceedings of the International Conference on Image Processing*, 3:777–780, June 2002.

[Di Sciascio and Celentano 1997] E. Di Sciascio and A. Celentano. Similarity evaluation in image retrieval using simple features. *Proceedings of the Society of Photo-Optical Instrumentation Engineers*, 3022:467–477, 1997.

[Ecker 2002] A. Ecker. *Images Similarity Based on the Distributions of Similar Substructures*. Master's thesis, Weizmann Institute of Science, October 2002.

[Fauzi and Lewis 2003] M.F.A. Fauzi and P.H. Lewis. A fully unsupervised texture segmentation algorithm. *Proceedings of British Machine Vision Conference*, pages 519–528, 2003.

[Fogel and Sagi 1989] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61:103–113, 1989.

[Greenspan *et al.* 2001] H. Greenspan, J. Goldberger, and L. Ridel. A continuous probabilistic framework for image matching. *Computer Vision and Image Understanding*, 84:384–406, 2001.

[Grigorescu *et al.* 2002] S.E. Grigorescu, N. Petkov, and P. Kruizinga. Comparison of texture features based on Gabor filters. *IEEE Transactions on Image Processing*, 11(10):837–842, October 2002.

[Guillon *et al.* 2001] F. Guillon, D.J.C. Murray, and P. DesAutels. Singular value decomposition to simplify features recognition analysis in very large collection of images. *AIP Conference Proceedings*, 583:143–145, August 2001.

[Haralick 1987] R. Haralick. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4):532–550, July 1987.

[Hsieh and Fan 2000] I.-S. Hsieh and K.-C. Fan. An adaptive clustering algorithm for color quantization. *Pattern Recognition Letters*, 21:337–346, 2000.

[Jacobs *et al.* 1995] C.E. Jacobs, A. Finkelstein, and D.H. Salesin. Fast multiresolution image querying. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 277–286, 1995.

[Kasparis *et al.* 2001] T. Kasparis, D. Charalampidis, M. Georgiopoulos, and J. Rolland. Segmentation of textured images based on fractals and image filtering. *Pattern recognition*, 34:1963–1973, 2001.

[Katsumata and Matsuyama 2005] N. Katsumata and Y. Matsuyama. Database retrieval for similar images using ICA and PCA bases. *Engineering Applications of Artificial Intelligence*, 18:705–717, 2005.

[Kelly *et al.* 1995] P.M. Kelly, T.M. Cannon, and Hush D.R. Query by image example: the candid approach. *Proceedings of the SPIE: Storage and Retrieval for Image and Video Databases III*, 2420:238–248, 1995.

[Kruizinga and Petkov 1999] P. Kruizinga and N. Petkov. Nonlinear operator for oriented texture. *IEEE Transactions on Image Processing*, 8(10):1395–1407, October 1999.

[Levina and Bickel 2001] E. Levina and P. Bickel. The earth mover's distance is the Mallows distance: Some insights from statistics. *Proceedings of the Eight International Conference on Computer Vision*, 2:251–256, 2001.

[Ma and Manjunath 1997] W.Y. Ma and B.S. Manjunath. Edge flow: A framework of boundary detection and image segmentation. *Proc. Computer Vision and Pattern Recognition*, pages 744–749, 1997.

[Manjunath and Ma 1996] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, August 1996.

[Mirmehdi and Petrou 2000] M. Mirmehdi and M. Petrou. Segmentation of color textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):142–159, February 2000.

[Parker and Behm 2004] J.R. Parker and B. Behm. Use of multiple algorithms in image content searches. *Proceedings of the International Conference on Coding and Computing, 2004*, 2:246–250, April 2004.

[Puzicha *et al.* 1999] J. Puzicha, J.M. Buhmann, Y. Rubner, and C. Tomasi. Empirical evaluation of dissimilarity measures for color and texture. *Proceedings of the International Conference on Computer Vision*, pages 1165–1172, 1999.

[Rodden *et al.* 2000] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. A comparison of measures for visualising image similarity. *Proc. Challenges of Image Retrieval*, 2000.

[Rubner *et al.* 1997] Y. Rubner, L. Guibas, and C. Tomasi. The earth mover's distance, multidimensional scaling, and color-based image retrieval. *Proceedings of the ARPA Image Understanding Workshop*, 1997.

[Rubner *et al.* 2000] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth movers distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[Rubner 1999] Y. Rubner. *Perceptual Metrics for Image Database Navigation*. PhD thesis, Stanford University, May 1999.

[Russell and Sinha 2001] R. Russell and P. Sinha. Perceptually-based comparison of image similarity metrics. *Proceedings of the second annual symposium on Computational geometry*, pages 1–13, July 2001.

[Seaborn *et al.* 2005] M. Seaborn, L. Hepplewhite, and J. Stonham. Fuzzy colour category map for the measurement of colour similarity and dissimilarity. *Pattern Recognition*, 38:165–177, 2005.

[Squire and Pun 1997] D. Squire and T. Pun. A comparison of human and machine assessments of image similarity for the organization of image databases. *Scandinavian Conference on Image Analysis*, pages 1–8, June 1997.

[Stevens 2001] M.R. Stevens. Evaluating 2d image comparison metrics for 3d scene interpretation. *Computer Vision and Image Understanding*, 84:179–197, 2001.

[Stollnitz *et al*. 1994] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin. *Wavelets for Computer Graphics: A Primer*. Technical report, University of Washington, September 1994.

[Swain and Ballard 1991] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[Theoharatos *et al*. 2005] C.H. Theoharatos, G. Economou, and S. Fotopoulos. Color edge detection using the minimal spanning tree. *Pattern recognition*, 38:603–606, 2005.

[Weinshall and Kirkpatrick 2004] D. Weinshall and S. Kirkpatrick. Passwords you'll never forget, but can't recall. *CHI '04 extended abstracts on Human factors in computing systems*, pages 1399–1402, April 2004.

Appendix A

Colour Results

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0.673	2.009	0.851	0.56	4.895	4.579	4.832	4.985	0.349	0.666	1.312
2	0.673	0	0.901	0.784	0.728	4.784	4.638	4.908	5.365	0.684	1.309	0.9
3	2.009	0.901	0	1.498	1.907	4.034	4.634	1.885	4.338	0.935	0.535	0.907
4	0.851	0.784	1.498	0	0.499	5.226	5.158	4.392	5.487	0.202	0.248	0.437
5	0.56	0.728	1.907	0.499	0	5.223	4.898	5.344	5.512	0.614	1.069	1.154
6	4.895	4.784	4.034	5.226	5.223	0	1.386	0.757	0.695	5.222	4.817	5.011
7	4.579	4.638	4.634	5.158	4.898	1.386	0	3.286	1.341	5.67	5.821	5.461
8	4.832	4.908	1.885	4.392	5.344	0.757	3.286	0	2.231	4.826	4.353	3.926
9	4.985	5.365	4.338	5.487	5.512	0.695	1.341	2.231	0	6.015	6.208	5.765
10	0.349	0.684	0.935	0.202	0.614	5.222	5.67	4.826	6.015	0	0.449	0.373
11	0.666	1.309	0.535	0.248	1.069	4.817	5.821	4.353	6.208	0.449	0	0.315
12	1.312	0.9	0.907	0.437	1.154	5.011	5.461	3.926	5.765	0.373	0.315	0
13	0.804	1.227	0.492	0.262	1.234	4.711	5.881	4.215	6.134	0.636	0.157	0.333
14	0.423	0.72	0.471	0.41	0.846	4.583	5.563	4.47	6.011	0.397	0.225	0.46
15	2.853	1.365	2.502	3.158	2.87	2.168	1.773	1.857	1.626	2.503	2.24	3.02
16	1.136	0.933	1.041	0.604	0.9	4.414	4.405	2.851	4.604	0.706	0.777	0.477
17	4.874	4.757	3.663	5.162	5.205	0.594	2.123	1.34	0.892	5.198	4.754	5.016
18	4.994	6.08	4.604	5.924	5.711	0.442	0.575	2.935	0.113	6.752	7.422	6.337
19	3.11	2.665	2.997	3.453	3.36	0.528	0.465	1.12	0.139	3.2	2.813	3.245
20	0.246	0.818	0.886	0.233	0.563	5.185	5.542	4.956	5.927	0.049	0.601	0.42
21	0.788	0.6	0.678	0.398	1.073	4.849	5.584	4.146	5.895	0.223	0.274	0.489
22	0.254	0.951	0.634	0.199	0.721	4.979	5.729	4.79	6.141	0.176	0.392	0.348
23	5.322	6.174	4.414	5.801	5.947	0.332	1.138	2.626	0.404	6.525	6.862	5.922

Table A.1: Colour Similarities

	13	14	15	16	17	18	19	20	21	22	23
1	0.804	0.423	2.853	1.136	4.874	4.994	3.11	0.246	0.788	0.254	5.322
2	1.227	0.72	1.365	0.933	4.757	6.08	2.665	0.818	0.6	0.951	6.174
3	0.492	0.471	2.502	1.041	3.663	4.604	2.997	0.886	0.678	0.634	4.414
4	0.262	0.41	3.158	0.604	5.162	5.924	3.453	0.233	0.398	0.199	5.801
5	1.234	0.846	2.87	0.9	5.205	5.711	3.36	0.563	1.073	0.721	5.947
6	4.711	4.583	2.168	4.414	0.594	0.442	0.528	5.185	4.849	4.979	0.332
7	5.881	5.563	1.773	4.405	2.123	0.575	0.465	5.542	5.584	5.729	1.138
8	4.215	4.47	1.857	2.851	1.34	2.935	1.12	4.956	4.146	4.79	2.626
9	6.134	6.011	1.626	4.604	0.892	0.113	0.139	5.927	5.895	6.141	0.404
10	0.636	0.397	2.503	0.706	5.198	6.752	3.2	0.049	0.223	0.176	6.525
11	0.157	0.225	2.24	0.777	4.754	7.422	2.813	0.601	0.274	0.392	6.862
12	0.333	0.46	3.02	0.477	5.016	6.337	3.245	0.42	0.489	0.348	5.922
13	0	0.555	1.921	0.754	4.488	7.508	2.622	0.806	0.445	0.597	6.843
14	0.555	0	1.748	0.744	4.548	7.402	2.507	0.454	0.102	0.293	6.73
15	1.921	1.748	0	2.762	1.731	0.611	0.945	2.121	2.626	2.144	1.109
16	0.754	0.744	2.762	0	4.025	5.106	3.02	0.708	0.596	0.715	4.746
17	4.488	4.548	1.731	4.025	0	0.216	0.334	5.158	4.854	4.951	0.315
18	7.508	7.402	0.611	5.106	0.216	0	0.201	6.937	6.261	7.361	0.13
19	2.622	2.507	0.945	3.02	0.334	0.201	0	3.043	2.958	2.759	0.226
20	0.806	0.454	2.121	0.708	5.158	6.937	3.043	0	0.223	0.197	6.598
21	0.445	0.102	2.626	0.596	4.854	6.261	2.958	0.223	0	0.024	6.118
22	0.597	0.293	2.144	0.715	4.951	7.361	2.759	0.197	0.024	0	6.801
23	6.843	6.73	1.109	4.746	0.315	0.13	0.226	6.598	6.118	6.801	0

Table A.2: Colour Similarities Continued

Appendix B

Texture Results

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	2.129	1.603	1.983	1.518	1.745	1.137	2.205	1.616	2.639	2.475	1.261
2	2.129	0	0.759	1.319	0.9	0.581	1.18	1.416	0.759	1.109	1.056	1.347
3	1.603	0.759	0	1.462	0.52	0.868	0.75	1.464	0.78	1.357	1.208	0.632
4	1.983	1.319	1.462	0	1.748	0.776	1.965	1.505	1.482	1.154	0.99	2.05
5	1.518	0.9	0.52	1.748	0	1.154	0.516	1.68	0.58	1.637	1.492	0.53
6	1.745	0.581	0.868	0.776	1.154	0	1.371	1.161	0.888	0.893	0.761	1.457
7	1.137	1.18	0.75	1.965	0.516	1.371	0	1.791	0.653	1.825	1.628	0.471
8	2.205	1.416	1.464	1.505	1.68	1.161	1.791	0	1.251	1.375	1.439	1.795
9	1.616	0.759	0.78	1.482	0.58	0.888	0.653	1.251	0	1.326	1.258	0.783
10	2.639	1.109	1.357	1.154	1.637	0.893	1.825	1.375	1.326	0	0.256	1.748
11	2.475	1.056	1.208	0.99	1.492	0.761	1.628	1.439	1.258	0.256	0	1.607
12	1.261	1.347	0.632	2.05	0.53	1.457	0.471	1.795	0.783	1.748	1.607	0
13	1.651	1.016	0.472	1.869	0.508	1.275	0.641	1.584	0.709	1.553	1.397	0.503
14	1.263	1.032	0.504	1.779	0.325	1.254	0.494	1.9	0.787	1.852	1.712	0.614
15	1.641	0.715	0.338	1.538	0.358	0.944	0.718	1.523	0.588	1.481	1.335	0.673
16	1.885	0.412	0.494	1.282	0.56	0.704	0.965	1.419	0.611	1.25	1.137	0.987
17	3.24	1.777	1.968	2.182	2.013	2.032	2.499	3.193	2.215	2.4	2.26	2.457
18	0.889	1.321	0.761	1.624	0.638	1.04	0.487	1.557	0.727	1.749	1.585	0.688
19	1.656	1.275	0.795	2.257	0.654	1.663	0.64	2.07	0.917	2.022	1.883	0.508
20	1.687	1.142	0.565	2.027	0.476	1.433	0.641	1.827	0.777	1.779	1.639	0.533
21	1.519	1.436	0.967	2.429	0.767	1.835	0.575	2.125	1.062	2.077	1.937	0.451
22	1.515	1.042	0.564	1.992	0.363	1.398	0.505	1.919	0.754	1.871	1.731	0.553
23	2.68	0.88	1.262	1.903	1.295	1.261	1.781	2.229	1.359	1.858	1.727	1.739

Table B.1: Texture Similarities

	13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
1	1.651	1.263	1.641	1.885	3.24	0.889	1.656	1.687	1.519	1.515	2.68
2	1.016	1.032	0.715	0.412	1.777	1.321	1.275	1.142	1.436	1.042	0.88
3	0.472	0.504	0.338	0.494	1.968	0.761	0.795	0.565	0.967	0.564	1.262
4	1.869	1.779	1.538	1.282	2.182	1.624	2.257	2.027	2.429	1.992	1.903
5	0.508	0.325	0.358	0.56	2.013	0.638	0.654	0.476	0.767	0.363	1.295
6	1.275	1.254	0.944	0.704	2.032	1.04	1.663	1.433	1.835	1.398	1.261
7	0.641	0.494	0.718	0.965	2.499	0.487	0.64	0.641	0.575	0.505	1.781
8	1.584	1.9	1.523	1.419	3.193	1.557	2.07	1.827	2.125	1.919	2.229
9	0.709	0.787	0.588	0.611	2.215	0.727	0.917	0.777	1.062	0.754	1.359
10	1.553	1.852	1.481	1.25	2.4	1.749	2.022	1.779	2.077	1.871	1.858
11	1.397	1.712	1.335	1.137	2.26	1.585	1.883	1.639	1.937	1.731	1.727
12	0.503	0.614	0.673	0.987	2.457	0.688	0.508	0.533	0.451	0.553	1.739
13	0	0.563	0.402	0.708	2.161	0.802	0.498	0.382	0.642	0.466	1.378
14	0.563	0	0.435	0.65	2.079	0.562	0.693	0.492	0.761	0.322	1.417
15	0.402	0.435	0	0.394	1.915	0.805	0.719	0.489	0.891	0.457	1.118
16	0.708	0.65	0.394	0	1.774	1.099	1.022	0.792	1.201	0.772	0.925
17	2.161	2.079	1.915	1.774	0	2.64	1.993	1.975	2.248	2.02	0.964
18	0.802	0.562	0.805	1.099	2.64	0	0.923	0.812	0.99	0.651	1.922
19	0.498	0.693	0.719	1.022	1.993	0.923	0	0.267	0.4	0.397	1.275
20	0.382	0.492	0.489	0.792	1.975	0.812	0.267	0	0.47	0.258	1.257
21	0.642	0.761	0.891	1.201	2.248	0.99	0.4	0.47	0	0.515	1.53
22	0.466	0.322	0.457	0.772	2.02	0.651	0.397	0.258	0.515	0	1.302
23	1.378	1.417	1.118	0.925	0.964	1.922	1.275	1.257	1.53	1.302	0

Table B.2: Texture Similarities Continued