Automated Searching of Illicit Drug Logos and Imprints

by

Ping Hei Ronnie Ng

A thesis submitted for the Degree of Doctor of Philosophy (Science)

2015

School of Chemistry and Forensic Science

Faculty of Science

University of Technology, Sydney

Certificate of Original Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Ping Hei Ronnie Ng

Acknowledgements

I would like to thank the University of Technology, Sydney for providing me this opportunity to carry out my own research in the forensic science discipline. In particular, the following people have given me a lot of guidance and help throughout the project. This project would not have been so successful without them.

1. Dr. Brian Reedy: He has given me a lot of assistance and help throughout this research. I enjoyed interpreting and discussing this project with him. His insights have inspired me with new ideas in numerous occasions. His patience and encouragement throughout this period means a lot to me. He has also dedicated a lot of his personal time for my project, especially the time spent in reading and giving me valuable comments on my literature review and my thesis.
2. Dr. Mark Tahtouh: He offered this interesting project to me. He provided me access to the AFP instruments (macroscopes, twin swan lights and ring lights etc), a genuine pill press and different kinds of dies for pressing and photographing my own tablets. The images that were processed in this project were therefore collected as they would have been in real forensic examination work.
3. Martin Lopatka: Thank you for your idea about matching side profiles of tablets and your suggestions for my MATLAB scripts.
4. The Australian Federal Police: Thank you for providing funding for my project.
5. Joshua Abraham: Thank you for your help in MATLAB programming and suggestions for various image processing methods, transformations and operations.
6. Mum, Dad and sis: I have unlimited support from all of you. I would not be here without you.

Table of Contents

List of Figures

List of Tables

Abstract

This thesis describes solutions for the automation of illicit tablet image searching based on tablet logos, colour and edge profiles. An original aim was to develop or exploit algorithms for searching images in an existing Australian Federal Police (AFP) drug tablet database. However, the pre-existing images were found to be inadequate for use with accepted logo extraction methods, due to poor illumination and insufficient contrast between the tablet surface and the logos. Extracting a complete logo from them was difficult. New “high contrast” images were taken in a way such that the contrast between the tablet surface and the logo was maximised. This was achieved by lowering the ring light that was attached to the macroscope very close to the tablet, resulting in a very shallow angle of illumination.

An in-house algorithm was designed to select the most appropriate threshold for each of these “high contrast” images to be converted to black-and-white (BW) images. Morphological and logical operations were used to segment the logos from the tablets. The final images contained only the silhouettes of the logos.

The logo silhouettes were encoded by calculating the Zernike moment invariants with the Kintner method, followed by the normalisation approach. The best chance of finding matching logos in the database was by ranking these invariants using the Euclidean distance.

Matching illicit tablets by colour was also investigated. The best method found involved changing the RGB colour values of the tablet photos to the CIE L*a*b* coordinates and then comparing these coordinates using the Normalised Inner Product.

The side profiles of tablets were suggested as an extra feature for comparison. Although there were limitations that remained to be solved (namely the symmetry problems for the positioning of tablets with asymmetric side profiles), results were promising when comparing the derivatives of symmetric side profiles using the Euclidean distance.

All of the algorithms developed or used in this project have been compiled into executable files such that any tablet database computer can run the scripts for searching

logos and side profiles. In addition, the scripts can provide measurements of a tablet's diameter and thickness in millimetres.

Chapter 1: INTRODUCTION

Chapter 1: Introduction

1.1 Background

The use of drugs, including tobacco, alcohol, prescription drugs and illicit drugs, cost Australia $23.5 billion in 2004-05 (1). Alcohol and Illicit drugs in particular cause a lot of problems to human society, economically and socially. In 2010, more than one in three Australians aged 12 years and older (38.6%) were drinkers who had put themselves at risk of injuries at least once in the past 12 months and 19.5% drank at levels which risk lifetime harm. Meanwhile, 14.7% of Australians aged 14 and over had used illicit drugs. Illegal drug trading and consumption have led to criminal activities and anti-social behaviour. The cost of drug use is incurred by the Australian health and welfare system in helping and treating drug abusers, by law enforcement agencies, who need additional resources to enforce law and order within the community, by the individual users and by the general public, since Australian economic productivity suffers. Law enforcement agencies attempt to limit smuggling and trading in the illicit drug market, as well as the consumption of such drugs.

According to an Australian report, 19.6 tonnes of illicit drugs were seized in 2012-13 (2). While cannabis was the most common illicit drug in Australia during that period, the number of illicit drug seizures and the amount seized has increased gradually over the last ten years. New technologies have enabled both the suppliers and buyers of illicit drugs to extend their reach, often transnational, with greater anonymity. Supply routes and transhipment hubs around the world are always changing to avoid law enforcement detection. Therefore, in order to have a better understanding of the global marketplace and give better responses to the threat illicit drugs pose globally, law enforcement agencies have been trying to link drug batches, seizures, drug processing equipment and drug trafficking from an intelligence perspective so as to effectively target their efforts and work across their jurisdictional boundaries. For example, being the second most widely-used illicit drug globally, amphetamine-type simulants (ATS), which induce intense euphoria to users, are often administered orally in tablets or powder form (2). The tablets, especially 3, 4-methylenedioxymethamphetamine (MDMA) tablets,

commonly exhibit logos or impressions on their surfaces. Linking different ATS batches and seizures can be achieved by obtaining chemical profiles and recording basic physical features such as the shapes and colours of tablets, as well as any logos that are stamped on tablets or blocks. Currently, the Australian Federal Police (AFP) Australian Illicit Drug Data Centre manages a drug profiling program which records these physical features by capturing digital images and manually entering characteristic details into a database. Subsequent searching of the recorded tablet images in the database is based on tagged metadata.

Retrieving tablet images using such text-based methods is difficult and inefficient, especially if the number of images in a database is large. The recording of physical features of a large number of different drug tablets is labour-intensive and time-consuming. More importantly, the annotation of colour and shape of tablets and logos is very subjective. This inconsistency of human assessment is caused by the different perception of the rich content of each image by each assessor. Mismatches are common in this kind of inefficient text-based retrieval process.

1.2 Content-based Image Retrieval (CBIR)

Content-based image retrieval (CBIR) is proposed to overcome the above-mentioned difficulties. Images are indexed by their contents, such as colour and texture, instead of manually annotating them with text. Unique features are extracted from the contents of each image in the form of vectors. A vector is a row of numbers, one number for each member of a list of variables. The number of variables determines the dimensionality of the vector. Image retrieval is carried out by comparing feature vectors from a query image and images from the database. In this way, there will not be any subjective human annotation. The amount of labour and time involved are also greatly reduced.

Content-based image retrieval (CBIR) is mainly used commercially or for research purposes to organise, manage and retrieve the exponentially increasing number of images in all sorts of image databases and in the World Wide Web. General literature surveys about CBIR are available in (3-8). It has been applied in different domains such as remote sensing, architecture and art and cultural imaging. The scientific domain that

attracts major research attention is in the medical field (9). Image retrieval techniques are reported in forensic practices for identifying handwriting, examining photographs from forensic cases to find case-specific information, and matching tattoos and graffiti (10-14). Very few articles are dedicated to the retrieval of images with similar illicit drug tablet logos in a database (15-18). The details of these are discussed later in the chapter.

In this project, it is envisaged that a tablet database might be searched using the following low-level features (colour, texture, shape etc). Feature vectors are created that can then be searched and compared in a manner analogous to spectral searching (i.e. using "distance" measures to determine similarity). Therefore the different stages might be:

1. pre-processing
2. feature vector extraction
3. similarity measurement using searching algorithms
4. retrieval of images that meet the search criteria
5. ranking of images based on their similarity to the query image.

The remainder of this chapter is a review on the current knowledge about the techniques and algorithms that would be necessary to carry out the above steps and retrieve tablet images from a database using CBIR. Section 1.3 discusses image pre-processing and various low-level image feature extraction techniques in CBIR systems. Section 1.4 outlines the different types of image similarity measures. Section 1.5 discusses CBIR system performance evaluation. Section 1.6 reviews the current literature on forensic drug tablet searching techniques. Section 1.7 introduces methods of narrowing down the "semantic gap", which will be useful for achieving the ambition of automated annotation of the physical features of drug tablets and the logos on them instead of manual tagging. Searching without a query image is also possible.

1.3 Low-level Feature Extraction

Low-level feature extraction is the basis of most CBIR systems (3, 7). These features can be either local or global (19). A local feature describes contents of a portion of an image while a global feature represents the whole image. Portions of images can be obtained

by partitioning images of equal size and shape, or by segmentation, which "divides the image into homogenous regions according to some criterion using region segmentation algorithms" (19). The latter method is superior to the former because segmentation produces perceptually meaningful regions. Segmenting objects from images is possible but the process involved is more complex and therefore it is unlikely to yield good results.

Extraction methods that are consistent, robust and efficient are desirable for the accurate retrieval of relevant images from a database. The features extracted have to be invariant to changes caused during the imaging process. The ideal extraction algorithm should produce the same unique feature vector regardless of a tablet's position within an image (translation), orientation (rotation), or magnification (scaling), as well as illumination and field of view. A thorough understanding of the different options available is necessary to choose the most suitable algorithm for this task.

Application-dependent features (e.g. human faces and fingerprints) are not included in this review as they are domain-specific. These features require domain knowledge and, more importantly, are largely irrelevant to physical feature comparisons for tablets.

1.3.1 Image Pre-processing

In order to achieve good feature extractions, some form of image pre-processing is necessary, apart from choosing the right feature extraction technique. Image pre-processing is the preparation and treatment of images to enhance image properties or aspects in the image data. Some of the feature extraction techniques and algorithms surveyed here may require special image pre-processing. These steps will be described with the relevant methods in the appropriate sections below. Only pre-processing methods unrelated to feature extractions are mentioned here. They are normally applicable for processing other sorts of multichannel signals as well.

Pre-processing is often achieved using filtering operations (20). The main idea is to use a sliding window to pass over an image pixel by pixel. Information is captured in a localised area to determine the output value of the pixel at the centre of the window. The window can have different shapes and sizes and can be designed to preserve data in specific directions.

A lot of different filtering techniques are available in the literature. They can be classified as either linear or nonlinear techniques (21). The former are mathematically simple and a unifying theory exists such that they are easy to design and implement. They yield satisfactory results operating under the assumption of a Gaussian model. However, in image processing, these linear algorithms fail because they "cannot cope with the nonlinearities of the image formation model and do not take into account the nonlinear nature of the human visual system" (21). Nonlinear techniques have taken advantage of increasing computing power and being able to make more realistic assumptions (21). Image details are better preserved. Due to the lack of a unifying theory, different nonlinear techniques have their own mathematical foundations.

1.3.1.1 Noise Removal

One of the most common reasons for carrying out pre-processing is to remove noise (20). Noise can come from a lot of different sources and is present in all image processing systems. They can be introduced during image formation (sensor noise) or when images are transmitted (channel noise). The perceptual quality of an image will be reduced under the naked eye. In addition, the presence of noise will reduce the performance of feature extraction algorithms and techniques.

Noise characteristics are usually non-Gaussian, unknown in nature or vary with time. Images consist of areas of homogeneous features and regions that change abruptly (e.g. at edges). Image details may not be satisfactorily enhanced if linear filtering techniques are used. On the other hand, nonlinear techniques are based on more realistic assumptions and are therefore better at suppressing non-Gaussian or signal-dependent noise.

Homomorphic filters are among the earliest-used nonlinear filters and have been used in image processing for removing multiplicative and signal-dependent noise (22). The filters are able to transform nonlinear signals to additive signals using logarithms. The additive signals are then suitable for processing using linear filters. The output is then converted back by the inverse nonlinearity to give the final result.

The Vector Median Filter (VMF) and the Basic Vector Directional Filter (BVDF) are examples of order statistic-based techniques that can be implemented to remove noise

(20). The output of the former filter is the colour value inside a sliding window that gives the minimum sum of distances to the other colour values. Similarly, the output of BVDF is the colour value that has the smallest total angular differences to the other colour values. Both methods may lead to excessive smoothing and affect image contents.

Neural networks can also achieve noise removal effects (23, 24). Neural filters can be implemented on existing hardware in neural networks. Different kinds of noise such as Gaussian noise and impulsive noise can be removed. In (23), two adaptive neural filtering algorithms in neural networks were used to form two optimal neural filters that satisfy different criteria.

Peer group filtering is a nonlinear algorithm to remove impulse noise (25). The peer group of a pixel is a set of neighbouring pixels within a window of variable sizes such that the surrounding pixels have a similar colour to itself. Each image pixel is replaced by the weighted average of its peer group members. If the pixel contains an impulse noise, the colour at that location is estimated by the rest of the pixels in the window. Edge blurring is avoided.

Fuzzy techniques have been used in multichannel signal processing (21). They provide a bridge between linear and nonlinear techniques. An adaptive fuzzy filter is one where the basis of local signal context adaptively determines and controls the parameters of the filters. This leads to data-dependent filters that are able to adapt to local image data.

Data Adaptive Filtering is an example of vector filtering that applies fuzzy rules to remove noise (20). The general formula is taken in the form of a fuzzy weighted average of the input colour values inside the sliding window. The fuzzy weights and the nonlinear function within can be changed to alter the filter characteristics.

1.3.2 Colour

Tablets often are coloured or contain dyes. The colour of tablets is one of the useful physical characteristics that can be linked to the tablet source, or other seized tablets. Colour features are commonly used in CBIR. The colour of each pixel or the dominant colour in a homogenous region of an image can be represented by a 1x3 vector in a 3D colour space. Examples of different colour spaces used for image retrieval purposes

include RGB (Red, Green, Blue), Munsell (hue, value, chroma), CIE L*a*b* (International Commission on Illumination, or, in French, Commission internationale de l'éclairage; Lightness, position between red and green axis, position between yellow and blue axis), CIE L*u*v* (Lightness, chroma coordinates u*, v* in the Uniform Chromaticity Scale Diagram, as derived by CIE in 1976) and HSV (Hue, Saturation, Value) (19). Different 1x3 vectors in a 3D space are much better differentiated for image retrieval than comparing scalar, single intensity values in gray-scale images. Therefore comparing and retrieval using single intensity values directly is seldom carried out.

There is no definitive choice of which colour space to use. However, according to Mathias (26), desirable characteristics for any colour space are completeness, uniformity, compactness and being user-oriented. A complete colour space has to include all perceptible colours. Uniformity refers to similarity between the measured proximity (using some distance algorithms) and the perceptive resemblance of two colours. A colour space is compact if each colour from the colour space is perceptually different. User-oriented colour spaces are easy to understand and must contain three basic attributes of the colours. Munsell, HSV, CIE L*a*b* and CIE L*u*v* are perceptually uniform colour spaces (19, 27, 28). It is known that RGB colour spaces are not perceptually uniform and are not efficient for image matching and retrieval tasks (27).

Although an ideal colour space includes all perceptible colours, colour clustering or colour quantisation is usually executed before extracting colour features from images. Images contain hundreds or thousands of different colours. This pre-processing step can significantly reduce the number of colours in an image. It effectively reduces the dimension of colour feature vectors extracted in most of the colour extraction methods mentioned below, especially histograms. Examples of such procedures are given by Deng et al. and Wang et al. (25, 29).

It should be noted that the variety of colours in an image affects the matching and retrieval results if colour features are used. Images with more perceptually different colours are more likely to achieve successful matching. Normal images may contain multicoloured objects on backgrounds of different colours. Identification or matching of such objects is more decisive. Most image retrieval discussions in the literature belong

to this type. However, tablets are usually mono-coloured. Their images mainly consist of the dominant colour and black backgrounds. Searching tablets using colour features only may not be a wise idea as there may not be enough discriminating power for effective tablet matching.

1.3.2.1 Colour Histograms

Colour histograms are graphical presentations of the distribution of pixels having the same colours in images. This basic technique is easy and fast to compute (3). They are also robust to translation along the image plane and rotation along the axis perpendicular to the plane. Changing the scale or the viewing angle will only change the histogram slightly (19). A colour histogram with more quantised colours or bins will be more discriminating as there are extra dimensions in the feature vector. The downside of this will be increasing computational cost and a loss of efficiency for building indexes in databases because of the high dimensionality. Retrieval performance does not necessarily improve (19).

Swain and Ballard used colour histograms to index multicoloured objects (30). They showed the method was robust and efficient in the presence of occlusion and at different fields of view. A three-dimensional object representation is obtained by imaging canonical views of the object before extracting a set of histograms for each view. Image histograms are compared to a database of model histograms to identify objects. However, changes in lighting may cause a lot of problems. In addition, the ability to identify multicoloured objects may not be useful for matching tablets with a single colour.

One part of the work by Stricker and Orengo involved improving on Swain and Ballard's method by proposing cumulative colour histograms (31). These were defined in the same way as statistical cumulative histograms. In the RGB colour space, colour A was less than colour B if all three colour values from A were less than those from B. The cumulative colour histogram counted the cumulative number of pixels in all the bins less than and equal to a specific bin colour. They suggested that the normal colour histograms were unstable, such that false positives were retrieved. On the other hand,

cumulative histograms were more robust with respect to colour distributions and therefore more stable and suitable.

The scalable colour descriptor (SCD) is histogram-based. It is included as one of the descriptors in the MPEG-7 standard (27). It first maps an image onto the HSV colour space. This space is uniformly quantised into 16 levels of hue, four levels of saturation and four levels of value to form 256 bins. This 256-bin histogram is eventually mapped onto a "non-linear" 4-bit representation for efficient encoding. Each histogram occupies 1024 bits and is encoded using a Haar transform to reduce the bits and make it scalable. Matching can be carried out directly using the histogram or by using the Harr coefficients.

The colour structure descriptor (CSD) is another histogram-type descriptor for colour features in the MPEG-7 standard (27). It expresses colour in a HMMD (hue-min-max-difference) colour space with an 8x8-structuring element in an image. The colour space is quantised to 184, 120, 64 or 32 bins. The frequency of each colour inside the structuring element is counted. In each histogram, the value in each bin corresponds to the number of structuring elements which contain at least one of such colour. The spatial extent of the 8x8-structuring element within an image can be scaled according to the image size. The L- metrics (see below) can be used to rank similarities between two CSDs.

In order to address the downside of high computational costs and low efficiency because of the high dimensionality of colour histograms, Kotoulas and Andreadis proposed the composite colour image histogram for images with high colour resolutions (32). This histogram has 2048 bins resulting from six bits in hue, three bits in saturation and two bits in value in the HSV colour space. Histograms are then processed using a FIR filter with Gaussian response before comparison.

1.3.2.2 Colour Moments

Colour moments have been successfully used in retrieval systems when images contain objects only. The first, second and third order moments are mean, variance and skewness respectively. They are calculated for each of the colour components to give nine colour moment numbers that represent the colour content of an image. Defining those moments on L*a*b* and CIE L*u*v* spaces gives better retrieval results than on

the HSV colour space (19). Although they are more compact than other colour feature representations, they lack discrimination power.

In the paper describing cumulative colour histograms, Sticker and Orengo also proposed a totally different technique to index colour contents by using colour moments (31). Quantisation problems and effects could be avoided by not using histogram methods. The colour distribution of an image was characterised by colour moments instead. Only the first three order moments were used. This method was mathematically valid and comparisons were easy using weighted Euclidean distance (section 1.4.1).

Mindru et al. proposed using colour moments of individual colour bands or in combinations for recognising different colour patterns (33). Invariants were generated by moments up to the first order and the second degree so that patterns could be matched under different viewpoints and illuminations. The shapes, intensities and colour distributions of patterns were encoded in these invariants. However, information about colour distribution may not be useful in our project as tablets mostly have one uniform colour.

1.3.2.3 Colour Invariants

Gevers and Smeulders introduced new colour models which were robust to a substantial change in viewpoint, object geometry, illumination and highlights (34, 35). These were designed for object recognition. The best recognition accuracy was achieved by the $l_1l_2l_3$ model under white illumination and the $m_1m_2m_3$ model under various illuminations. A system that was robust to a substantial change in viewpoint was also introduced. Colour pattern-cards were constructed using hue, hue-hue pairs and three values from the second order colour invariant. Images were retrieved using the colour pattern-cards by statistic-based or geometry-based matching functions.

1.3.3 Texture

Texture is “a visual pattern that has properties of homogeneity that do not result from the presence of only a single colour or intensity” (36). This feature provides important information about the structural arrangement of surfaces as texture is an inherent property of all surfaces. It can also reveal their relationship to the surrounding environment (37). Extracting texture information was first done in areas researching

pattern recognition and computer vision. Now it has been applied for retrieving images. A lot of different methods have been proposed. Old review papers comparing the different algorithms of the time showed the co-occurrence matrix method achieved the best performance (38, 39).

1.3.3.1 Co-occurrence Matrix

One of the first published articles dedicated to extracting textural features was from Haralick and his colleagues (37). They computed features in the spatial domain to give a set of gray level spatial-dependence probability-distribution matrices. Fourteen textural features were extracted from these matrices. A subset or a linear combination of the mean and the range of these textural features could be used for matching. As a follow-up, Gotlieb and Kreyszig conducted experiments and concluded that combinations of less than five features performed better than using a single one (40). Groups of four features were optimal. Three out of the 14 textural features had the best discriminatory power. The names of the three features were contrast, inverse deference moment and entropy. This co-occurrence matrix method may not provide unique identifications. Nonetheless, it was useful for discrimination.

1.3.3.2 Perceptually Corresponding Features

A paper by Tamura et al. described a method to extract texture features approximating human visual perception (41). From the work about human perception in the psychological field, Tamura et al. applied six visual texture features, namely coarseness, contrast, directionality, linelikeness, regularity and roughness. Between them, coarseness, directionality and regularity were more significant than the others. These sets of features were superior to co-occurrence matrices because the former were visually meaningful, while the statistics-based co-occurrence matrices were not easily understood visually. Using perceptually meaningful features allowed image retrieval system designers to provide a user-friendly interface.

1.3.3.3 Wold Features

Liu and Picard introduced a new way to describe texture using the Wold features (42). The Wold features were formed by using three mutually orthogonal subfields, periodicity, directionality and randomness, from the two-dimensional Wold decomposition of homogeneous random fields. The authors quoted a previous study

that these three properties were similar to human texture perception. The Wold decomposition would decompose homogeneous random fields to an indeterministic component (the randomness dimension) and a deterministic component which was further split into the periodicity and directionality dimensions. Then the Maximum Likelihood Estimation (MLE) was used to give more accurate estimation of the components from each subfield. The authors improved the method by using harmonic peak extraction and MRSAR modelling without decomposing images. Higher tolerance to a variety of local inhomogeneities in natural textures and to its invariance to rotation was achieved.

1.3.3.4 Wavelet-based Texture Extraction

Wavelet transforms are very frequently used for extracting, analysing and classifying texture. The main idea is to decompose signals, in the form of images, into a set of basis functions by recursive filtering and sub-sampling. These basis functions are obtained from translation and dilation of a mother wavelet (19). A lot of different wavelet-based transforms are available in the literature. Some are discussed below.

Chang and Kuo described a multiresolution approach for texture analysis and classification (43). The approach was based on a modified wavelet transform called the tree-structure wavelet transform (TWT). It addressed the problem that the conventional pyramid-structured wavelet transform (PWT) performed decomposition in low frequency channels when the quasi-periodic signals of most natural textures had dominant frequencies in the middle frequency channels. The transform could be easily computed and performed excellently. The authors demonstrated that TWT provided better results than other available methods.

The wavelet filter bank was used by Smith and Chang to extract the mean and variance as an image's texture representation (36). The idea of the extraction procedure was to make use of the spatial-frequency decompositions provided by the wavelet filter banks. This texture feature set was a binary vector at the pixel level. It was composed of binary combinations of basis vectors which were dependent on the threshold output of the wavelet filter. Texture points were grouped into smaller areas using non-linear filtering

techniques. Another binary texture vector represented the new region and was used for matching.

Jain and Farroknia applied a bank of Gabor filters to create a multi-channel filtering function for finding homogeneous texture regions (44). The filter set formed the basis for the Gabor wavelet transform. Texture information was obtained as a measure of "energy" in a window around each pixel after a nonlinear transformation of each filtered image.

A paper by Ma and Manjunath illustrated that the Gabor wavelet performed better than others such as the TWT (45). The authors claimed the Gabor method was easy to interpret and could offer flexibility in controlling orientation and scale information, despite a higher computational cost. Such controllability could lead to features that were invariant to those factors.

Do and Vetterli improved the traditional approaches by accurately modelling the marginal distribution of wavelet coefficients using the generalised Gaussian density (46). Their method was more accurate and flexible for obtaining texture information while achieving similar computational complexity as the previous methods. The authors suggested using their approach for retrievals when the classes were not defined a priori.

1.3.3.5 Texture Browsing Descriptor

The MPEG-7 standard adopts the texture browsing descriptor as one of its texture descriptors (27). It characterises perceptual attributes similar to the features suggested by Tamura et al. in Section 1.3.3.2. It requires a maximum of 12 bits. A bank of scale and orientation selective band-pass filters is used to filter an image. The descriptor components can then be calculated.

1.3.3.6 Homogeneous Texture Descriptor (HTD)

The Homogeneous Texture Descriptor (HTD) is another method for indexing texture in MPEG-7 (27). It characterises quantitatively a homogeneous texture region by computing the local-spatial-frequency statistics of the texture.

1.3.3.7 Edge Histogram Descriptor (EHD)

The Edge Histogram Descriptor (EHD) is the third method for texture extraction in MPEG-7 (27). It works best for regions which are not homogeneous in texture properties.

The spatial distribution of edges is recorded by computing 16 edge histograms on an image partitioned into 4x4 sub-images. Five edge categories are considered for each histogram to give an 80-bin representation of an image. The bins are non-uniformly quantised to form an EHD of 240 bits. A limitation of EHD is its sensitivity to object and scene distortions.

1.3.3.8 Others

Two groups of researchers have applied both the wavelet transform and the co-occurrence matrix to extract texture (47, 48). The Quadrature Mirror Filter (QMF) or a wavelet transform was used to show the textural images in multiresolution. The matrix was calculated at different frequency bands for extracting textural features. The features chosen were contrast, angular second moment, entropy, image pixel correlation and inverse different moment as defined by Haralick et al. (37). This method obviously took advantage of both the transform-based wavelet method and the statistics-based co-occurrence method.

1.3.4 Shape

Shapes are very important to human perception. Retrieving images using shape features will require the shape representations be invariant to translation, rotation and scaling. Such representations can be divided into two main categories. The first type is a boundary-based feature which describes the outer boundary of the shape. Region-based representation, taking into account the whole shape region, is the other type. Mehtre et al. suggested in their paper that a combined representation of both types was better than using only boundary-based or region-based representation (49).

1.3.4.1 Fourier Descriptor (FD)

Fourier Descriptors (FD) are one of the boundary-based methods for describing shapes. The contours of an object can be defined by shape signatures to represent two-dimensional areas or boundaries using a one-dimensional function. The four most common and practical shape signatures in FD are the centroid distance, complex coordinates, curvature and cumulative angular function (50). The representations undergo Fourier transforms to form three sets of complex coefficients. These coefficients, representing a shape in the frequency domain, can be further processed or

adjusted to be scale and rotation invariant (19). Advantages of FD include strong discrimination power, excellent noise insensitivity, easy normalisation and information preservation.

A Fourier Descriptor (FD) was used by Zahn and Roskies in 1972 (51). They provided a method to analyse and reconstruct closed curves in a plane. A closed curve (shape) was represented as a function of the accumulated change in direction of the curve versus the arc length from a starting point. The descriptive coefficients were obtained without redundant information by expanding the function in a Fourier series. The axial symmetry of the shape could also be analysed. However, according to Persoon and Fu, the coefficients obtained from the linear interpolation of two descriptors from two closed curves did not necessarily represent another closed curve (52). In addition, reconstruction was inefficient (53).

Hand print character recognition was investigated by Granlund (54). Complex coordinates were used as the shape signature. While shape reconstruction was easy and all his descriptors represented closed curves, unlike Zahn and Roskies' counterpart, the descriptors had some strict constraints (52). A formula to compute a polygonal boundary using Granlund's definition was detailed by Persoon and Fu (52).

Rui et al. addressed the effects of spatial discretisation of shapes by modifying the FD method (53). Complex coordinates were also used as the shape signature in which a shape was redefined as a uniform length four-connected boundary. Although this Modified Fourier Descriptor (MFD) took more time to extract features than other methods, the authors argued that it was acceptable because extractions would be done offline. Their newly-defined distance metric specifically for MFD performed faster and was intrinsically invariant to translation, rotation, scale and starting point. The slower offline extraction process was justified by the faster online matching.

Zhang and Lu compared the four shape signatures used to derive the FD in their paper (55). They found that using different signatures could have significantly different effects on the retrieval results. The outcome showed that using the centroid distance (representing the distance of the boundary points from the centroid of a shape) to derive the FD gave the best retrieval results. It was invariant to translation. The complex

coordinates and the cumulative angular function signatures gave similar performances. Both were superior to the curvature function method.

Zhang and Lu also proposed a generic FD in another paper (56). Their method was not application dependent: it could be applied in cases without shape boundary knowledge or when the shape signature was unknown. This new generic FD was derived from a two-dimensional polar Fourier transform in a polar space. The polar Fourier coefficients obtained were used to derive a normalised FD for representing shape. Shape features were better captured in both radial and circular directions. The new method was more robust than the shape signature-based ones.

1.3.4.2 Wavelet Descriptor

Chuang and Kuo developed a new boundary representation with the wavelet transform (57). This hierarchical planar curve descriptor provided multiresolution representation, invariance, uniqueness, stability and spatial localisation. A curve would be decomposed into components of different scales. The coarse scale components contained global approximation information while local detailed information dictated the finer scale components. Features extracted could be normalised to be translation, rotation or scaling invariant. The authors stated that wavelet descriptors were “more efficient in representing and detecting local features of curves due to the spatial and frequency localisation property of wavelet bases” (57). They were also computationally efficient.

Kunttu et al. combined both wavelet transforms and Fourier transforms to describe shapes (58). The boundary of an object was first subjected to a complex wavelet transform. This allowed the Fourier transform to be applied on the resulting wavelet coefficients at multiple scales. The advantage of computing this Fourier descriptor at multiple scales was to achieve better shape retrieval rates. This descriptor would be rotation invariant.

1.3.4.3 Turning Function

A method to represent boundaries of polygons was illustrated in a paper by Arkin and his colleagues (59). A turning function measured the angle of the anticlockwise tangent as a function of the arc-length at an arbitrary point on a polygon. This was equivalent to recording the cumulative exterior angle of the polygon as a function of the length of

each side. The difference between the initial angle at the starting point and the final angle after summing the angles along the perimeter anticlockwise back to the starting point would always be 2π. The function was itself invariant to translation. It could set to be scale invariant by setting the total perimeter length to be one. Differences in the point of origin could be offset by horizontally shifting the turning function (phase). Comparing functions from polygons of different orientation and at different phases would be possible by calculating the L_p distances over all shifts and rotations and choosing the minimum distance for ranking.

1.3.4.4 Moment Invariants

Moments are numbers that represent the statistical properties of images (60). They are region-based representations of objects and shapes. Mathematically, moments are projections of a function onto a polynomial basis. In CBIR, the image pixel intensities form the continuous function. The sum of p and q in the general moment $M_{p,q}$ is called the order of the moment. Simple properties from moment calculations include the "mass" or area of the image, the centroid or the centre of gravity of an object, and the "distribution of mass" or the moments of inertia of an object. Invariants are defined as a set of measurable quantities that are insensitive to certain deformations. These quantities do not change when these deformation operations are performed on an image.

Moment invariants are invariant to translation, rotation and scaling. Different schemes can be employed to make them invariant to other aspects as well, such as contrast (61), illumination (62) and photometric changes (63). A study by Li and Ma demonstrated that the moments and the Fourier descriptor (FD) were related by a simple linear transformation (64). A huge number of different forms of moments are available in the literature, especially in the book by Flusser et al (60). Only some representative ones are mentioned below.

The first pioneer in setting up the fundamental theorem of image moments was Hu (65). He used the theory of algebraic invariants to derive seven famous moment invariants from the second and third order moments with a standard polynomial basis $x^p y^q$. Six of them were absolute orthogonal invariants and the last one was a skew orthogonal

invariant for identifying mirror shapes. However, the derivation method for these moments was complicated and Hu did not show how to derive more invariants from higher-order moments (60). Also, the mutual dependence of the seven moments and the incompleteness of the system limited the discriminating power. Nonetheless, his work became a classic and was successfully applied to different areas of research.

A lot of research has been published since the appearance of the Hu moment invariants. A total of fifty-two invariants from higher-order moments (up to the ninth) were later derived by Li using the Fourier-Mellin transform (66). Wong and his colleagues published an automated approach to generate a complete set of moment invariants up to the fifth order (67). Another way to quickly generate large numbers of moment invariants at higher orders was presented by Jin and Tianxu through the decomposition of the trigonometric functions (68).

Complex moments were introduced by Abu-Mostafa and Psaltis to derive moment invariants (69). The polynomial basis was simply $(x+iy)^p(x-iy)^q$. Using the properties of complex moments, they evaluated moment invariants for their discrimination power and noise tolerance. Information loss, suppression and redundancy were also tested.

A new set of orthogonal complex moments called Zernike moments were introduced by Khotanzad and Hong (70). These were calculated using an orthogonal basis. This change in the basis was crucial for making the Zernike moments mutually independent for improved discrimination power. The basis employed had a useful rotation invariant property. Image reconstruction was easy. Zernike moments were also insensitive to noise and could be calculated quickly.

Belkasim et al. presented a new way to derive moment invariants using Zernike moments (71). A new normalisation approach was included to reduce the large dynamic range and implicit redundancies.

Yang and Albregtsen improved on the already well-researched way of calculating moments using the discrete Green's theorem (72). Previous methods were either not accurate using Green's theorem or inefficient using the discrete version. The new version using the discrete Green's theorem was accurate and fast because a double sum

over an isolated object was used instead of integration. Methods for computing moments for regions or objects in grey level and binary images were given.

A new way to find useful invariants was investigated by Kapur and his colleagues (73). Previous searching for these invariants was by extensive experience and from geometric invariant properties studied in algebra. They discovered a way to systemically generate and search invariants. An algebraic invariant relation between object and image parameters was established first before extracting the invariant from the algebraic relation.

Sluzek proposed the idea of object recognition using only moment invariants of order two (74). He argued that any moments of higher orders were "very sensitive to digitalisation errors, minor shape deformations, camera non-linearity, and non-ideal positioning of a camera". In his paper, an object was occluded by circles of different sizes at its centre. All these circle-occluded shapes, and their moment invariants of order two, would represent the object.

Lei et al. published a method to recognise complex objects (75). Objects could be represented by a single polynomial of third or fourth degree or by partitioning them into patches where each patch was described by an implicit polynomial. Pairs of invariants were computed from the coefficients of pairs of polynomials if an object was partitioned. Bayesian object recognition of the invariants could be used as a robust way to identify objects at a low computational cost. Mutual invariant recognisers were also tested.

Shen and Ip introduced a set of wavelet moment invariants for classifying objects, especially objects that were very similar but with subtle differences (76). Instead of using a radial variable function to define moments, they were calculated using wavelet basis functions. Invariants were selected automatically based on the discrimination measures.

Recently, Flusser addressed the problem of the mutual dependence and incompleteness in the Hu moment invariants (77). He first showed how to generate moment invariants of any order using complex moments. Then he demonstrated that a small basis existed so that all these invariants could be expressed. These invariants were shown to be independent of each other and the set was complete. This was crucial because the same discriminative power could be achieved using the minimum number of moment

invariants at the least computational cost. In a follow up paper, Flusser showed this invariant basis could be easily converted to any other rotational invariants such as the Zernike moment invariants (78).

1.3.4.5 Edge Detection

Edge features are recognised as an important aspect of human visual perception and are thus commonly used in shape analysis. Edge detection in the context of image processing is "the localisation of significant variations of the grey level image and the identification of the physical phenomena that originated them" (79). The most common edge type is steps. Less commonly encountered types are lines and junctions. Images are usually pre-processed by smoothing, and then differentiated to detect changes in intensity gradients across the images (edges). A general edge detection algorithm suitable for all purposes does not exist. A lot of different edge detectors are proposed in the literature and surveys. Examples of these can be found in papers by Ziou and Tabbone (79), and by Basu (80).

Smoothing can reduce noise and enhance edge detection by removing false positives. On the other hand, it can lead to information loss. The optimal solution is to find a filter of appropriate scale. Some examples of such filters are the Green function and the Gaussian function (79). The Gaussian function is a better choice because it is invariant to rotation. The gradient (first-order derivative), the Laplacian or the second-order derivative can be used for differentiation (79). The non-linear and variant to rotation properties of the first-order and second-order derivatives are less desirable. The rotation-invariant Laplacian operator can be applied before or after smoothing due to its linearity. The outputs of edge detectors may contain a single output or a combination of information about the position and strength of edges, their orientation and the scale.

One approach to optimise the scaling of edge detection is to use multi-scale edge detection, as reviewed in work by Ziou and Tabbone (79), and Basu (80). A lot of different examples and implementations of such scale optimisation methods have been listed and discussed by these authors. Edges can be obtained by applying a single edge detector at different scales and then combining the edge information to form a more complete picture of the actual edge representation. The user decides the number of scales, which

scales to use and how to combine the outputs from each scale. Bergholm suggested a scale difference of 0.5 in the gradient of a Gaussian filter (81). The Laplacian of Gaussian was suggested by Ziou and Tabbone to combine edge information at different scales by first identifying edges from the same origin (79). The similar ones were combined to form a single image. Readers interested in multi-scale edge detection are directed to the reviews by Ziou and Tabbone (79), and by Basu (80).

Edge detectors are classified into two classes: autonomous detectors and contextual detectors (79). The former do not need a priori knowledge and are used flexibly for general images. The latter are for specific applications and few are proposed.

Prewitt (82) and Sobel (83) edge detectors are two famous edge detectors that have existed for nearly 40 years. They are based on gradient and Laplacian operators. Their performance deteriorates with increasing noise. The Canny edge detector is another benchmark Gaussian-based operator (84). It is optimised for good detection, good localisation and only one response to a single edge. However, although the Canny edge detector is sensitive to weak edges, it is prone to unstable boundaries or insignificant changes in intensity (80).

Ziou and Koukam suggested a solution to the difficulty of choosing the appropriate edge detector (85). Their system was called Selection of Edge Detectors (SED) which could automatically select edge detectors and their scales to extract a given edge. Selection of Edge Detectors (SED) stored different edge detectors and several sources of information such as edge characteristics, detector properties, the mutual influence between edges and detectors and the effective results of the run detectors. The authors claimed that such a method was not the combinatorial approach that previous researchers had implemented. However, more inputs were necessary apart from the image itself. For example, a set of constraints could be entered to specify the quality of edge results. The quality could be expressed by a set of constraints related to the delocalisation error and computational time. This would defeat the aim of the current project: automatic extraction and comparison of information from drug tablet images without human involvement.

Cheng opted for another way to extract edge features by applying the moment-preserving principle to images (86). Images were first partitioned into equal sections of 4x4 pixels. Each section was classified into one of the 36 predefined edge patterns and coded with the pattern category information for matching. The edge directions of those predefined edge patterns were multiples of 45°. The edge directions of all sections in an image were also quantised to multiples of 45° to match those edge patterns. Hence the matching process could be faster.

Phase congruency was described by Kovesi (87). The main difference from other methods was that it uses frequency domain data from images. Phase congruency was defined as a dimensionless measure ranging from zero to one, or from a feature with no significance to a highly significant feature. This allowed a user to specify a threshold using phase congruency to define features before displaying results. This was hugely different from other methods in which a threshold was defined beforehand. Although it was sensitive to noise, Kovesi provided solutions to estimate the energy function of noise and compensate such noise from phase congruency. This new measure was robust to changes in scale, illumination and contrast. Kovesi suggested using high-pass filtering instead of the usual low-pass filters.

1.3.5 Segmentation

Segmentation is a very important part of feature extraction. Segmenting visually meaningful regions allows the image contents extracted in each area to be more semantic. A lot of segmentation techniques have been proposed. However, images in the tablet image databases contain a single tablet and black background only. Segmentation is therefore not likely to be used in the current project, so only a few examples are listed here.

The wavelet-based texture extraction method proposed by Jain and Farrokhnia in Section 1.3.3 was extended to segment images based on texture (44). The “energy” measured contained attributes of the nonlinear transformation. Different attributes were associated with different texture. An unsupervised square-error clustering algorithm was used to identify textural categories and produce segmentation.

Lybanon et al. used the morphological operation of opening and closing to segment simple scientific images (88). This mathematical morphology processed images based on shapes. The operation was very similar to a convolution. While it was not efficient for complex general images, it may be suitable for tablet images because it was first designed for finding star boundaries in optical astronomical images, which had uniformly bright stars and a black background. Tablet images have similar properties. The application of such algorithms in this project, if deemed necessary, could be promising.

Hansen and Higgins used a combination of watershed analysis and relaxation labelling to segment images (89). Watershed analysis was applied with great speed to divide images into catchment basins. Refining and updating the classification of those catchment basins was achieved using relaxation labelling. The algorithm was very robust to noise.

A segmentation algorithm derived via a fuzzy logic approach was presented by Li and his colleagues (90). All the local entropy maxima were located by the fuzzy entropy function, which was obtained based on the fuzzy region width and the Shannon's function of the image. Each of these local entropy maxima corresponded to uncertainties among various regions in an image. It was suitable for segmenting images without any dominant colours or without knowing the number of the segmentation classes.

Ma and Manjunath presented Edge Flow to detect boundaries and segment images (91). This could identify and integrate different types of image boundaries. This predictive Edge Flow model identified the direction of change in colour and texture at different scales of an image. An edge flow vector could then be constructed. Boundaries were at locations where two opposite directions of flow were in the stable state. Image scale was the only parameter in the algorithm that a user had to adjust, to suit the user's own requirement.

Feng et al. adopted a curve evolution approach for segmenting images using adaptive flows (92). An active contour model was derived by minimising a simplified form of the Mumford-Shah function. Under the assumption that images were segmented to contain piecewise constant regions, local mean intensities in separate regions of a contour were

estimated for computing the adaptive flows to obtain a new contour. The process was repeated until the contour converged. The main advantage was that no prior knowledge about the number of regions or the values of their mean intensities was needed in advance.

Principles of feature space analysis were demonstrated by Comaniciu and Meer to segment images (93). Feature space analysis was used to recover the centres of high density regions and areas in a feature space so as to locate the corresponding significant features. This was based on the mean shift algorithm, a simple non-parametric procedure. Images to be segmented were processed in the CIE L*u*v* colour space. Users could choose three segmentation resolutions: undersegmentation, oversegmentation and quantisation. The method was efficient for images with homogeneous colour regions. In a follow-up paper, Comaniciu and Meer presented a general non-parametric technique to segment images using the mean shift (94). No parameters were required from users except the resolution of the analysis.

A two-step approach called JSEG tested the homogeneity of a given colour-texture pattern to achieve unsupervised segmentation (95). The first step was to quantise colours in an image to several representative classes. Image pixels were replaced by their corresponding colour class labels and a class-map of the image was formed. The second step was to perform spatial segmentation based on “good” segmentation criterion suggested by the authors. The resulting regions had homogeneous colour-texture properties. This method, unlike other texture segmentation formulas, did not require the difficult task of estimating the texture model parameters.

Blobworld segmentation is widely used for image segmentation (96). Pixels are clustered in a joint colour-texture-position feature space. The joint distribution of colour, texture and position features is modelled with a mixture of Gaussians. The model parameters are estimated using the expectation maximisation (EM) algorithm. Image segments are formed by the resulting pixel-cluster membership. They correspond well with objects or parts of objects. The colour and texture features of each region are also generated for retrieval purposes.

1.3.6 Colour Layout

Colour features are computationally inexpensive and are able to provide reasonable discriminating power in image retrieval. However, they often return false positives especially in large databases. Colour layout is another type of content-based feature which integrates spatial relations into the colour contents. Partition or segmentation is sometimes necessary before extracting colour features in each region for spatial information. It would probably not be as useful for retrieving tablet images as other images because most images in the AFP database consist of a single colour tablet at the centre of a black background. The relative position of a particular colour to other colours is not important in most cases.

1.3.6.1 Colour Sets

Smith and Chang used colour sets for retrieving images and even regions within images (97, 98). Colours in each region were transformed to the HSV colour space. They were then quantised by mapping onto carefully chosen colours and bins (colour sets). Multiple binary feature vectors for each image were formed by determining whether each region exceeded a particular threshold of each colour in the set. Searching was simple and fast through binary data. Spatial information could also be specified in a query. Spatial locations of regions containing colours from the colour set were recorded by using the minimum bounding rectangle (MBR) during the extraction along with the binary feature vectors.

1.3.6.2 Colour Histograms using Gauss Mixture Vector Quantisation (GMVQ)

Instead of following the traditional way of counting the number of pixels of the same colour to produce colour histograms, Jeong et al. applied Gauss Mixture Vector Quantisation (GMVQ) (99). Normally, uniform scalar quantisation is used to quantise a range of colours into uniformly spaced levels and therefore reduce the colour histogram dimensionality. However, this histogram contained no spatial information. The GMVQ approach took the actual colour distributions of images from the database into consideration for quantisation. Histograms were generated from the labels representing the Gaussian components instead of from the quantised pixel colours. Each label held pieces of information about the spatial variation of a colour channel within a group of

pixels. Image retrieval was better using histograms from GMVQ with a penalised log-likelihood distortion than using the conventional ones.

1.3.6.3 Multiresolution Histograms

Hadjidemetriou and his colleagues demonstrated the encoding of spatial information directly into multiresolution histograms (100). These were a set of intensity histograms of an image at different image resolutions. A multiresolution histogram could be presented as a vector in which each of its elements was itself a histogram in the order of decreasing image resolutions. They pointed out that the spatial information was strongly related to the weighted averages of the rates of change of histogram intensities. Those weighted averages could be obtained by using the Fisher information measures. This method retained the advantages of histogram matching in that these histograms were fast to compute and robust to translation.

1.3.6.4 Colour Coherence Vector (CCV)

Pass and Zabih incorporated spatial information into colour histograms to form colour coherence vectors (CCV) (101). Quantised colours from the histogram were differentiated into two categories based on spatial coherence. Pixels were coherent if they originated from a sizable adjacent region. The CCV expressed the numbers of coherent and incoherent pixels for all quantised colours as vectors. In that way, spatial information was added to the histogram data. CCV achieved better retrieval results than the colour histogram. The authors noted that perceptually uniform colour spaces such as the CIE L*a*b* colour space were preferred over the RGB colour space.

1.3.6.5 Colour Correlogram

The colour correlogram was defined by Huang et al. to include spatial information into colour features (102). It was able to describe the distribution of local spatial correlation of colours of an image as a whole. It could tolerate large changes in scale, viewing positions or background. The colour correlogram was presented as a three-dimensional matrix which listed the probability of finding a pixel of colour *j* at different distances *k* from a pixel of colour *i*. A simpler version called the colour autocorrelogram was also available. Only the spatial relationships between the same colours were considered (i.e. $i = j$). This approach gave the best retrieval results over colour histograms or CCV, but was the most computationally expensive and complex given the high dimensionality.

The authors suggested the technique be tested in the perceptually uniform colour spaces other than RGB that they had used.

1.3.6.6 Dominant Colour

Dominant colour is a descriptor in MPEG-7 used to represent a set of colours in a region of interest in an image (27). Representative colours are computed and used for each region or image. The feature descriptor contains the representative colours, their percentages in the region, the spatial coherency of the dominant colours and the colour variances for each dominant colour. Similarity is measured using algorithms similar to the quadratic colour histogram distance formula. Each and every representative colour in the query image is independently searched against regions of images for that colour. The matches are combined to form the final retrievals. This descriptor has an advantage over colour histograms because the dominant colour can be indexed in the 3D colour space, while high-dimensional indexing problems have to be tackled when using colour histograms. This makes the dominant colour descriptor more compact than colour histogram descriptors.

Deng and his colleagues applied the method to retrieve images from databases (103). The Edge Flow algorithm was selected to segment images (91) and representative colours from the CIE L*u*v* colour space were chosen using colour clustering (25). The system was considered fast and effective.

An article by Babu et al. described a retrieval method on databases containing trademark images and flag images instead of normal generic images (104). Their representative colours were chosen by a LEADER colour clustering algorithm (105). The dominant colours were partitioned in a three-dimensional colour space. The groups were used as spatial objects for indexing. This dominant colour descriptor was suitable for matching trademark images and flag images because the majority of the regions were homogeneous.

1.3.6.7 Colour Layout Descriptor (CLD)

The colour layout descriptor (CLD) in MPEG-7 includes the spatial distribution or the layout of colours in an image (27). The representative or average colour in the YCrCb colour space is first extracted from an arbitrary region. Then it undergoes the Discrete

Cosine Transform (DCT) with quantisation, and the resulting coefficients are encoded. Weighted Euclidean distances are used to compare the coefficients.

1.3.6.8 Signature-based Colour-Spatial Retrieval

Work by Chua et al. provided a novel way to retrieve images by first partitioning images into equal sizes (106). One hundred and fifty-nine perceptually dissimilar colours from the CIE L*u*v* colour space were chosen. A colour could represent a particular region if the percentage area of the colour exceeded a threshold. The signatures were compared taking into account the colours that were perceptually similar and the weighting on object colours. This approach was superior to another similar method published previously and more effective than histogram-based methods. However, this retrieval system performed poorly on images that had no dominant colours or those in which the dominant colours were not close together.

1.3.6.9 Sampling-based Matching (SamMatch)

Hua et al. extracted colour features at space intervals of images (107). The average colour was taken as the feature vector. They argued that this method provided users with control over the relevance of the results by varying different factors. Factors such as scaling of the space interval, contribution of different regions and colours could be adjusted manually or automatically. Partial retrieval was possible. Retrieval could be performed with query images and the potential candidate images taken at different scales or magnification. SamMatch could achieve better retrieval results with less storage space and time.

1.3.7 Others

1.3.7.1 Combination of Low-level Features

Extracting the colour and texture information at particular locations (salient points) instead of collecting those data globally or locally was investigated by Tian and his colleagues (108). These specific locations could be decided using coefficients from wavelet transforms. These wavelet-based salient points were places with high global variations. Colour moments and the Gabor wavelet transform were used to extract colour and texture features respectively. Retrieval accuracy was significantly improved with fewer data and, as a result, less computational complexity.

1.3.7.2 Discrete Wavelet Transform (DWT) with Principal Component Analysis (PCA)

The Discrete Wavelet Transform (DWT) was shown by de Bianchi and his colleagues to compress black-and-white images (109). Signature matrices were generated as a result. Principal Component Analysis (PCA) was applied on those matrices along with a signature matrix from a query image to extract the mean data, calculate the covariance matrix and generate eigenvalues and eigenmatrices. Similarity was based on the Euclidean distance between the query and data signature matrices that were projected onto a subspace by the longest eigenmatrix. The 2nd-level biorthogonal wavelet with support-size 4 worked best.

1.3.7.3 Scale-invariant Feature Transform (SIFT)

Lowe introduced features that were invariant to image scale and rotation and robust to affine distortion, noise and change in illumination (110). The Scale-Invariant Feature Transform (SIFT) converted images into scale-invariant coordinates relative to local features. Keypoints were appointed based on local image properties. Image locations, scale and orientation information were assigned at each keypoint.

1.3.7.4 Motif Co-occurrence Matrix (MCM)

Another way to tackle feature extraction was demonstrated by Jhanwar et al. using six Peano scan motifs (111). These six scan motifs were used to replace all of the 2x2 grids partitioned from an image such that each chosen scan motif minimised the local gradient of a particular grid. The MCM was defined as a three-dimensional matrix, in that it returned the probability of finding a motif *j* at different distances *k* from another motif *i*. The authors stated that the MCM combined data related to both colour and texture by capturing the third order neighbourhood statistics of the image. Although MCM was sensitive to translation, Jhanwar et al. provided solutions in their article to make it invariant. This method was very similar to the Colour Correlogram for extracting the colour layout defined by Huang and his colleagues that was discussed in the previous section (102).

1.4 Searching algorithms

Searching algorithms are formulas for comparing the extracted features from image contents. They give an objective measurement of similarity between features extracted from a query image and images from a database. As most extracted features are expressed in the form of numbers, vectors and matrices, searching algorithms are mathematically a form of distance calculation between two or more of these expressions.

Most distance metrics in mathematics can be applied directly as searching algorithms. The degree of similarity of vectors is defined by the distance calculated. Comparable vectors have shorter distances and are closer to each other in a feature space than dissimilar ones. Vectors along with their corresponding images are ranked by their decreasing similarity or increasing distance from the query vector and its corresponding query image. The number one ranking (shortest distance) image is said to be the most similar image to the query.

Image searching by comparing feature vectors is a very similar process to spectral searching. In spectral searching, a sample is identified by comparing its spectrum with spectra of different reference materials. These spectra are treated as vectors, or a row of numbers (intensities) at different wavelengths/frequencies. Searching algorithms are applied to the vectors to find the reference spectrum/vector that is closest (shortest distance) to the query spectrum/vector. The identity of the sample is then confirmed after visual comparison.

The choice of similarity/distance measurements will significantly affect the quality of the image retrieval system (19). Some commonly used algorithms are mentioned here together with a few specific algorithms to be used with specific feature vectors introduced above.

1.4.1 Minkowski-type Metric

Many workers have used the Minkowski-type metrics (L-metrics) for distance calculation (19). The distance between two vectors $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ is defined by:

$$L_r = \left(\sum_{i=1}^{n} |x_i - y_i|^r \right)^{\frac{1}{r}}$$

When r = 1, this becomes the city-block distance or the Manhattan distance (L_1 distance). For example, SCD histograms (Section 1.3.2.1) have been shown to achieve good retrieval accuracy using the L_1 distance (27). When r = 2, it is equivalent to the famous Euclidean distance (L_2 distance). The majority of papers surveyed here used this algorithm directly. A few authors made adjustments to it to suit their extraction methods (e.g. (52)). When r approaches infinity, it is the chess-board distance or the Chebyshev distance (L_∞ distance).

The general formula gives equal weights (w_i) to all components of a vector. The proportion of each element in a vector contributing to the similarity measurement can be changed by (7):

$$L_r = \left(\sum_{i=1}^{n} w_i |x_i - y_i|^r \right)^{\frac{1}{r}} \; where \sum w_i = 1$$

Swain and Ballard first proposed Histogram Intersection, a form of the L_1 metric, as the similarity measure for the colour histogram (30). Given two histograms with n bins, A = (a_1, a_2, …, a_n) and B = (b_1, b_2, …, b_n), the Histogram Intersection is defined by:

$$H = \frac{\sum_{i=1}^{n} \min(a_i, b_i)}{\sum_{i=1}^{n} b_i}$$

The Histogram Intersection will be equal to one if histograms A and B are identical.

Mathias and Conci tested the influence of colour spaces and metrics on the performance of image retrieval using colour histograms (26). They found out that the city-block distance (L_1 distance) and the Histogram Intersection produced the best recovery and retrieval efficiency in HSV colour space or in the Opponent colour space.

Mehtre and his colleagues argued that detailed comparisons of colour histograms using the Minkowski-type metrics were not necessary and could even lead to inaccurate retrieval due to noise (112). They introduced two new distance methods loosely based on the Minkowski-type metrics and both algorithms performed better than the

Histogram Intersection. The first one was the Distance Method. Three one-dimensional histograms were constructed for each of the three colour components from an image. These three mean values of the histograms (or the normalised values, considering the fraction of pixels in each bin of the histogram), were used to form the feature vector. The normal Manhattan distance or Euclidean distance was applied for distance calculation.

The other method was the Reference Colour Table Method (112). A set of n reference colours was chosen beforehand. Pixels from images were assigned the nearest colour within the set and histograms were computed. The relative pixel frequency with respect to the total number of pixels was used for each colour. The distance formula was very similar to the weighed Euclidean distance but the weights were defined differently. The Reference Colour Table Method achieved better retrieval than the Distance Method above.

1.4.2 Normalised Inner Product

The similarity of vectors can also be quantified by calculating the angular distance between them. The inner product is the sum of products of components from two normalised vectors of lengths equal to one. Geometrically it gives the cosine value of the angle between the two vectors X = (x_1, x_2, …, x_n) and Y = (y_1, y_2, …, y_n). The formula is:

$$s(x,y) = \frac{(X \cdot Y^T)}{\|X\|\|Y\|} = cos\theta_{X,Y}$$

The angular distance, or the angle, between two vectors is just the arccosine of the inner product:

$$\theta_{X,Y} = \cos^{-1}\left(\frac{X \cdot Y^T}{\|X\|\|Y\|}\right)$$

1.4.3 Quadratic Form Distance

The Minkowski distance calculation is easy to perform and execute. However, it assumes all components to be compared in feature vectors are independent of each other when

this does not always hold (19). Some components are perceptually dependent on each other. The quadratic form distance can be used:

$$D = \sqrt{(X-Y)^T Q (X-Y)}$$

The matrix Q is a similarity matrix whose its entries q_{ij} represent the similarity between components *i* and *j*. To measure colour histogram similarities using this quadratic form distance, the entries of the similarity matrix Q are calculated differently depending on the colour space used. In other words, the similarity calculation between two colour components in matrix Q is different for different colour spaces. Formulas applicable for RGB and HSV are listed by Jeong and his colleagues (99). For any two colours $I(R_i, G_i, B_i) = (h_i, s_i, v_i)$ and $J(R_j, G_j, B_j) = (h_j, s_j, v_j)$, the entries q_{ij} are given by:

$$q_{ij} = 1 - \frac{L_2(I,J)}{\max[L_2(I,J)]} \; where \; L_2 \; is \; the \; Euclidean \; distance$$

$$q_{ij} = 1 - \sqrt{\frac{\left(v_i - v_j\right)^2 + \left(s_i \cos h_i - s_j \cos h_j\right)^2 + \left(s_i \sin h_i - s_j \sin h_j\right)^2}{5}}$$

Mathias and Conci found that this technique was computationally more demanding than the Minkowski-type counterparts (26).

1.4.4 Mahalanobis Distance

$$D = \sqrt{(X-Y)^T C^{-1} (X-Y)}$$

The Mahalanobis distance given above is suitable for comparing vectors whose components are dependent on each other (19). The distance function factors in a covariance matrix C. The element c_{ij} in the covariance matrix is the measure of change, or the covariance, between components *i* and *j*. If the feature components are independent of each other or if C is diagonally dominant, the variance of each component c_i can be used to simplify the equation to:

$$D = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{c_i}}$$

Both forms of the Mahalanobis distance were used by Chang and Kuo to compare the performance of their texture extraction and classification algorithm in (43).

1.4.5 Kullback-Leibler Divergence/Distance

$$D = \sum_{i=1}^{n} x_i \log \frac{x_i}{y_i}$$

This is an equation to measure "how compact one feature distribution can be coded using the other one as the codebook" (19). This was the algorithm chosen by Do and Vetterli to compare the texture features extracted in (46). The similarity measurement step was asymptotically optimal in terms of retrieval error probability.

1.4.6 Others

Instead of solving a computationally expensive optimisation equation originating from the modified Euclidean distance, Rui et al. used the weighted sums of two components as the distance measure (53).The two components were the variance of the scale magnitude (D_m) and the variance of phase shifts (D_p) at different boundary points.

$$D = w_m D_m + w_p D_p$$

The weights were arbitrarily set as $w_m = 1$ and $w_p = 0.1$ for image retrieval.

Borgefors proposed an edge-matching technique called the Hierarchical Chamfer Matching Algorithm (HCMA) (113). Edge points or other low-level features could be matched. Edges were first extracted by methods discussed earlier. The matching was carried out from a coarse resolution to finer resolutions. This approach raised the computation speed considerably. Parallel computations were possible in appropriate hardware with such capabilities. It could be used to determine the best match, if any, and the corresponding position. The measure of the quality of the match provided was easy to interpret. It could also be used to match specific objects in images for recognition and identification work.

The Earth Mover's Distance (EMD) was introduced by Rubner et al. as a new, general and flexible distance between two distributions (114). It allowed a "many-many match" in which a region in a query image could match more than one region in the target image

and vice versa. Common distance algorithms allowed a “one-one match” only. This new algorithm was based on the solution to the old transportation problem. The solution was to apply linear optimisation to calculate the minimal cost to “transform one distribution into the other by moving ‘distribution mass’ around” (114). The main advantage was that it matched perceptual similarity well and more importantly, was applicable to compare feature representations of variable lengths by calculating the minimum cost to transform one representation to another.

Chang and Yeung avoided manually choosing a distance function by implementing a method where image data could choose a suitable function automatically (115). A kernel approach was used to learn a distance metric based on pairwise similarity and dissimilarity constraints between images. This kernel-induced feature space was nonlinearly related to the image space. The input feature points were mapped to the feature space before linear metric learning was performed in this space. A small sample size was sufficient for this process. It demonstrated better retrieval performance than other distance measurements due to its higher flexibility. The authors pointed out that performing linear transformation with the Minkowski-type or the Mahalanobis metrics was not suitable in a nonlinear image space that contained high variability of image contents.

Gondra and Heisterkamp attempted to perform image retrieval without feature extraction or comparing feature vectors (116). Images were searched based on their raw data in the form of raw pixel values alone. This simple technique removed the time and need to select, extract and apply weights to the low-level features. Image dissimilarity was based on a recent algorithm called Normalised Information Distance (NID). Approximations of the Kolmogorov complexity by different compression methods were used to calculate NID distances between images. They created statistically significant dissimilarity against random retrieval. However, this promising approach needed further research as NID was based on an incomputable notion of Kolmogorov complexity which was hard to accurately approximate with the current compression methods.

1.5 Database for CBIR and Performance Evaluation

Standard test data and performance evaluation models do not exist in the study of CBIR (7). Different papers and publications work on different image databases. There is no ideal vocabulary for representing the rich semantics of a wide range of images. The number of ways of evaluating the performance of a particular retrieval system is also limited. A standard way of evaluating a technique may never be known because different researchers design their CBIR systems based on their own needs. The requirements, performance and the final output are very goal-oriented. Hence one can only measure the true performance of a particular content retrieval technique relative to others in his or her own image database.

There are different image databases available commercially and on the World Wide Web. There exist databases with generic images such as portraits, landscapes and crowds. Databases for specific images such as a collection of different texture images are also present for a more specific purpose. Most published work uses normal generic images for testing their algorithms. In the current quest to identify or group the different physical features of illicit drug tablets, it is intended that only the AFP database containing images of all different kinds of illicit drug tablets seized by the law enforcement agency will be used. Comparison of the different feature extractions, matching algorithms or a combination of both will be based on the AFP tablet image database.

While finding an ideal list of vocabulary representing the rich semantics of normal generic images is not an easy task due to the variety of categories and complexity, words to describe the physical features of illicit drug tablets are relatively easier to define. These words have to be carefully chosen so that they do not represent the same or similar categories (7). Nor should they be too abstract such that images within the same category have varying contents. They should be well-defined and yet general enough to associate semantic meaning. The number of categories should not be too high either, because learning and classifying different categories is not easy.

Precision (Pr) and Recall (Re) are commonly used in CBIR systems as a way to reveal retrieval performances (4, 7, 9). Precision (Pr) is defined as the ratio of the number of retrieved relevant images (N_r) to the number of total retrieved images (K). Recall (Re) is defined similarly such that it is a comparison between the number of retrieved relevant images (N_r) and the total number of relevant images available in a database (N_t). Retrievals with high Pr and Re are preferred. Another way to measure performance is by using the Rank (Ra). It is simply the average rank of the retrieved images. In this case, a smaller rank number represents better performance.

$$Pr = \frac{N_r}{K}, Re = \frac{N_r}{N_t}$$

1.6 Drug Tablet Searching

Geradts et al. evaluated the ability of two commercially available CBIR systems, QBIC and Imatch, to search for images of illicit tablets and compared searching results using the MPEG-7 shape descriptors (15). They found that the commercial systems were able to give better results because their colour descriptors were more effective. However, the main disadvantage of using commercial systems was that no source code was available for the algorithms used. The MPEG-7 shape descriptors performed better only after pre-processing and segmentation were done on tablet images.

Chen and his colleagues designed an automated drug image identification system (ADIIS) using multiple features (17). Colour (in the HSV colour space), shape (using the Canny edge detection) and texture (using the Gabor filter) were extracted along with different physical dimensions. The feature vectors were compared using weighted Euclidean distances between features, with different weights based on fuzzy rules.

Lee et al. also designed a retrieval system called Pill-ID for tablet image searching (18). After images were smoothed using a Gaussian filter, images were treated with the Sobel operator to produce gradient magnitude images. From there, the logo shapes were encoded using SIFT descriptors. Colour was represented using three-dimensional 64-bin histograms. The tablet shapes were expressed using the Hu moment invariants. Rotational normalisation was achieved using multiple templates representing the

dominant orientations for each tablet. Searching was carried out using Euclidean distance. As each tablet generated multiple templates, a lot more matching had to be carried out and hence there was a greatly increased computational time.

Illicit drug tablet profiling was achieved using CBIR by Camargo and his colleagues such that tablets with similar contents could be identified (117). Two histograms were constructed for an image in the grey scale and in the RGB colour space. The Sobel edge detection was used to detect salient and smooth edges. Tamura texture was also extracted.

A method to distinguish counterfeit tablets from genuine ones was proposed by Jung et al. (118). Tablets were first imaged in a controlled and repeatable condition to avoid colour variations by different instruments or examiners. After segmentation, statistical analyses were done on the RGB colour components of the tablets. Classification was achieved by checking the adherence of the query to the statistical distribution.

Lopatka and Vallat introduced a new feature called the surface granularity for profiling illicit drug tablets (119). They referred to the inconsistencies at the tablet surfaces as speckles and quantified these in granularity profiles for similarity measurements. These profiles included speckle count, the mean and standard deviation of speckle areas, distances between speckles, speckle eccentricity and relative chromaticity, plus “the proportion of speckles that exhibit a higher chromaticity than the tablet median value” (119).

In another paper, Lopatka and van Houten developed a method for automating illicit tablet (external) shape classification (120). Nineteen shape categories were defined and tablets were classified into one of these categories in two steps. The shape feature could easily be integrated into any database. This classification could be useful to filter the results obtained when searching illicit tablets using other features, including colour. Possible candidates that shared the same source or the same production batch as the query could be narrowed down further.

1.7 The Semantic Gap

Discussions in the previous section show the variety of procedures available to extract contents such as colour, texture, shape and colour layouts from images. These low-level contents are obtained from computer vision and mathematical pattern recognition algorithms. Image retrieval by comparing different sets of image features in a database (i.e. query by example (19)) is straightforward. However, human beings interpret images using high-level features and concepts including keywords and text descriptors with logical inference or reasoning (3, 7). There is no direct link between these high-level semantic concepts and low-level numerical features. The latter is neither capable of modelling image semantics nor able to describe the former (121, 122).

The missing link or the gap between a rich high-level semantic concept from a user and a limited descriptive low-level image feature is called the “semantic gap”. The ability to bridge this gap is crucial for the automatic annotation of images of illicit drug tablets and the searching of these images by keywords or by text without any query images (i.e. query by concept (19)).

The major state-of-the-art methods for bridging this “semantic gap” can be classified into five categories and each will be discussed here. The last section will introduce the idea of classifying different types of images.

1.7.1 Object Ontology

Semantics are derived from our daily language in this category (7). Several low-level features are divided into different intervals. Each interval represents an intermediate-level descriptor. Different combinations of these intermediate-level descriptors define simple vocabularies or the “object ontology” based on our knowledge. Database images are classified into the different categories or vocabularies based on the intermediate-level descriptors present in the images. For example, a combination of intermediate-level descriptors “light blue” (colour), “uniform” (texture) and “upper” (spatial location) can be used to represent the semantic keyword “sky”.

Such an approach was adopted by Mezaris et al. for example (123). Images were represented in the CIE L*a*b* colour space. The intermediate-level features they chose

to classify the segmented regions were the three colour components, vertical and horizontal positions, size and shape.

Liu et al. quantised colours from the HSV colour space for colour naming (124). A set of 35 colours were preselected and individually given a semantic colour name. Each name was related to objects in natural scene images for image retrieval. Stanchev et al. applied the concept in painting retrieval (125). One hundred and eighty reference colours were chosen to relate the emotional expression and visual impression of paintings. Contrasts such as light and dark, warm and cold were associated with the reference colours and used as a query in their retrieval system.

Wang and his colleagues generated codebooks to support semantic retrieval (126). The semantic codebooks were composed by different classes of colour and texture. Images were segmented according to the codebooks. Each region was described by its content and context. The content included the colour and texture information. The context showed the relationship with other regions of an image. Both the content and context allowed semantic retrieval. Query using keywords (query by concept) was also possible.

1.7.2 Machine Learning

High-level semantic features can be derived from machine learning techniques (7). These techniques can be divided into two groups depending on their way of "learning". Supervised learning predicts the value of an outcome measure from a set of input measures. Unsupervised learning tries to find the logic behind how the input data are organised or clustered. Both learning processes can be performed offline. More research may be needed in the field of machine learning techniques because visual object learning and recognition is not easy. Learning different visual object categories takes thousands of training images. According to Liu et al., the techniques in the current literature have tested less than 20 object categories (7).

1.7.2.1 Supervised Learning

Support Vector Machine (SVM) is one of the supervised learning techniques available (7). It was originally developed for binary classification. Its aim is to find an optimal separating hyper-plane to separate two dissimilar groups of data in a vector space. This hyper-plane has the furthest distance between the nearest data point of each group and

itself. Groups of data on one side of the hyper-plane are labelled as +1 and the other vectors on the other side are labelled as -1. Different trainings are required for different concepts.

Chang and Tong implemented Support Vector Machine Active Learning (SVMActive) for image retrieval (127). Multiresolution image representation was employed for learning. Images were classified by colour using a colour mask of 12 bins and by texture using wavelet transforms. One relevant and one irrelevant image were needed to start the learning.

Shi and his colleagues trained 23 semantic keywords using SVM for annotating images (128). A binary SVM model was generated for each category using manually tagged segmented regions. An unlabelled region could be fed into the 23 models and the keyword that achieved the highest positive result was tagged for that region.

An Artificial Neural Network (ANN) is another example of a supervised learning technique (7). Town and Sinclair applied an ANN for learning 11 concepts commonly found in natural scene landscape images (129). Training data consisting of low-level features from segmented regions were fed into the neural network classifiers to establish the link over the “semantic gap”. The disadvantage was the large amount of training data required and the high cost of computing.

1.7.2.2 Unsupervised Learning

Here, the traditional k-means clustering and its variations are used for image clustering and unsupervised learning (7). Image clustering tries to maximise the similarity within clusters of image data and, at the same time, the dissimilarity between different clusters.

The k-means method was chosen by Stan and Sethi to map low-level colour features from a set of training images to semantic keywords (130). The statistical measure of the variation between clusters was considered to derive the required set of mappings. New untagged images could be indexed by those mapping definitions.

Zheng and his colleagues proposed the Locality Preserving Clustering (LPC) for image clustering (131). The clustering was done in the spectral domain. A mapping function was also included such that no clustering was needed for each new database entry,

similar to the traditional k-means technique. The authors demonstrated that LPC provided better retrieval than the k-means method and was computationally efficient.

Real-time annotation of images was achieved by Li and Wang (132). They designed a clustering algorithm for objects represented by discrete distributions called D2-clustering. Its underlying idea was similar to the k-means algorithm. It was a generalisation of the k-means method from using data in the form of vectors to sets of weighted vectors. However, the optimisation techniques were more sophisticated.

1.7.3 Relevance Feedback (RF)

While offline learning algorithms are discussed in the previous section, Relevance Feedback (RF) is an online processing technique (7). It was first used in text-based information retrieval systems (7, 19). The basic idea of RF is to let the user choose positive and negative examples from the retrieved results to improve CBIR performance. Database images are first retrieved and ranked by a similarity metric based on their similarity to a query image. A user then chooses relevant and irrelevant images out of the retrieved images and a new, refined set of images are retrieved based on the selection. The process is repeated until the user is satisfied with the search. A survey of different RF technologies can be found in (133).

One of the ways to refine the retrieved images is by the Query-Point Movement (QPM) (7). It improves the estimation of the query point in each feedback by moving it closer to the positive examples and away from the negative ones. It was first proposed by Jing and his colleagues (134). The initial query was set as a positive example. The positive examples from each feedback were assembled into a composite image by clustering similar regions of those examples. This composite image replaced the original query image for further user feedback. The importance of the prior positive examples decreased over iterations, while the weights of the newly added examples increased.

Another approach is to adjust the weights of low-level features in the comparison process (7). This was introduced by Jing et al. from the inspiration of feature re-weighting methods (135). The region importance of all relevant images was estimated from the user feedback under the assumption that important regions appeared more in

those relevant images and less in the irrelevant ones. The weight of a region changed for each feedback.

Relevance Feedback has been used with other machine learning algorithms (7). Examples include (123, 127).

1.7.4 Semantic Template

Semantic template is not a widely used technique (7), yet it is a promising way to bridge the "semantic gap". It maps high-level concepts and low-level visual features. A semantic template is a "representative" feature of a concept calculated from a collection of sample images. It gives rise to a system with two abilities. A user can enter a concept keyword and the retrieval system relates this to the corresponding feature vector for image retrieval even though images are not annotated. In reverse, the system can extract low-level features from an image and deduce the semantic meaning of the image from its feature vector.

Zhuang et al. demonstrated a way to generate semantic templates automatically (136). It was achieved using Relevance Feedback. A network of semantic templates was constructed to help users retrieve images on the semantic level.

1.7.5 Web Image Retrieval

Web image retrieval uses additional information from the World Wide Web to reduce the "semantic gap" (7). The URL of an image contains a clear hierarchical structure and information about the image category is sometimes included in this. The HTML document often describes and labels the image as well. These descriptive texts can be utilised to annotate images. It is effective and accurate to a certain extent (136).

A similar method was proposed by Zhou and Huang (122). A hybrid image retrieval system was presented that used text annotations and low-level features for image retrieval. This flexibility allowed a user to search images with high-level semantic keywords or with low-level feature contents. A statistical algorithm clustered groups in the concept space by relevance feedback in the image space. Eventually, it could learn the user's search preference along with semantic relations among concepts.

The ideas employed in web image retrieval may be useful to annotate illicit drug tablet images. The database from the AFP already contains annotated images from previous seizures and during data management procedures. Those texts, even though subjectively annotated, hold abundant information about the physical features and facts for bridging the "semantic gap". Careful use of those texts is necessary. At the same time, more illicit drug tablet images may be acquired from the World Wide Web. The descriptions can also be used to annotate the new images.

1.7.6 Image Classification

There are a number of research papers exploring the possibility of classifying different low-level feature categories and images. Being capable of differentiating different classes of features and images is a vital step to try to minimise the "semantic gap". This ability allows the retrieval system to recognise the class of feature present in an image and group the images accordingly. Semantic keywords of colours, textures and shapes may be associated with a group.

1.7.6.1 Texture Classification

Classifying different types of texture has been an active research area. Texture feature vectors are extracted from training images first using different extraction methods. Kundu and Chen applied the Fisher linear-discriminant classifier (47). It reduced a multi-dimension feature vector to a more manageable one dimensional vector by projecting it onto a certain direction determined by the mean and covariance matrices of two groups of training samples. A feature vector belonged to one of the classes if its projected vector was greater than a particular threshold set by the mean values and variances of the projected vectors from two classes. Classification of a given test sample in the presence of more than two classes was possible by implementing a voting scheme.

A similar classifier was used by Thyagarajan and colleagues to categorise textures (48). It was the maximum likelihood estimator. The probability density function of the feature vectors was assumed to have a known parametric form and to be determined by the value of the parameter vector. The maximum likelihood estimator was designed to find the parameter vector that maximised the likelihood function. If the feature vectors were Gaussian distributed, the parameter vector was just the mean and the covariance of the

probability density function of the feature vectors. The log likelihood function for a particular class was calculated using the parameter vector from the training feature vectors. Classification was done by comparing the likelihood functions from a particular class and from the feature vector in question.

Gross et al. employed a Kohonen-based learning vector quantisation (LVQ) classifier for texture classification (137). Local texture features represented by a set of respective wavelet coefficients from a wavelet transform process were treated by a local traversal of the wavelet pyramid to form a feature vector. De-correlation, normalisation and dimension reduction were performed on those vectors to optimise feature representation by Principal Component Analysis. The vectors were then classified. A network that was organised to preserve the data topology acted as a platform for such classification and learning.

1.7.6.2 Object Classification

Higashijima et al. classified images of objects using a lifting wavelet and Principal Component Analysis (PCA) (138). Training images of the objects were decomposed into low-pass and high-pass images by wavelet transform. Principal component vectors were obtained from the low-pass images by PCA. Feature vectors were generated from the low-pass images expanding along the principal component vectors. The average feature vectors were calculated and a lifting parameter was learnt to lift those vectors closer to the average. The process was repeated until different object categories were separated adequately.

Murtagh and Starck applied image classification for industrial purposes by classifying images of mixtures of aggregate for grading (139). The second, third and fourth order moments of the multiresolution transform coefficients were used for classification. The authors suggested that these moments enhanced discrimination at multiple resolution scales. The first order moment served no purpose in this case. Each image therefore produced 72 moments from five scales of wavelet and 19 bands of curvelet transform. Correspondence analysis and k-nearest neighbours supervised classification categorised the sizes and shapes of the granules.

1.8 Scope and Aims

The method currently used by the AFP to search its illicit drug tablet database is based entirely on operator-generated text-based metadata to identify colour, shape, logo etc. This approach is inefficient and inaccurate because it is (i) subjective and prone to inconsistencies; (ii) labour-intensive; and (iii) unable to be automated.

Therefore, the aims of the current project are multi-fold. The most current and up-to-date CBIR technologies will be examined and their potential assessed for application to the interrogation of drug tablet physical features. Then a CBIR system based on promising techniques will be developed and tested to identify and group images of illicit drugs based on physical features such as colour, shape and logo. The next step is to test the CBIR system in a real world scenario to match and retrieve illicit drug tablet images.

With a huge variety of different feature extraction techniques and methods mentioned above in the survey to choose from, automated image retrieval based on contents is a promising solution to the labour and time-intensive job involving subjective human annotation of illicit tablet images. Efforts will be focussed on tablet logo searching using moment invariants and the tablet colour searching performed in different colour spaces. The results are detailed in the following chapters.

The emphasis of this thesis is to solve a forensic problem through novel application of image processing tools, devising new algorithms where required, and to formulate an image collection and processing protocol that can be used in the construction of law enforcement databases.

Chapter 2: LOGO EXTRACTION FROM EXISTING AFP ILLICIT DRUG TABLET IMAGES

Chapter 2: Logo extraction from existing AFP illicit drug tablet images

Since 2003, the Australian Federal Police (AFP) has been taking photos of all illicit drug tablets seized during casework examinations in a consistent and standardised way. In the quest for automated tablet logo searching, the ideal scenario is that the designed algorithm for this process would be suitable to be used on any existing data available such that all the old and new images could be searched together. This chapter focuses on our efforts to extract logos from the old image database so as to encode them as feature vectors for searching and comparison, the main objective of our work.

2.1 Method

Access was granted to a representative portion of the photos in the AFP's database that were taken between 2003 and 2005 (Figure 2.1). The scripts and computational processing were developed and run using MATLAB installed on a standard UTS-issued laptop with a 2.54 GHz CPU and 4 GB RAM.

Figure 2.1: Some examples of the photos obtained.

2.2 Edge Detection

An article by Lee et al. describes attempts to tackle the problem of searching tablet logos (140). The authors used edge detection as the initial image processing step. The current work started with the application of different edge detection techniques to the tablet images. The built-in MATLAB function *edge* enabled edges in grayscale images to be found quickly and easily with a choice of different filter masks and parameters. The conversion of tablet images to greyscale ones was carried out using the built-in MATLAB function *rgb2gray*.

The following three tablet images (shown as greyscale images) were randomly chosen from the database for illustrating our results in this section.

Figure 2.2: The three tablet greyscale images used for illustration.

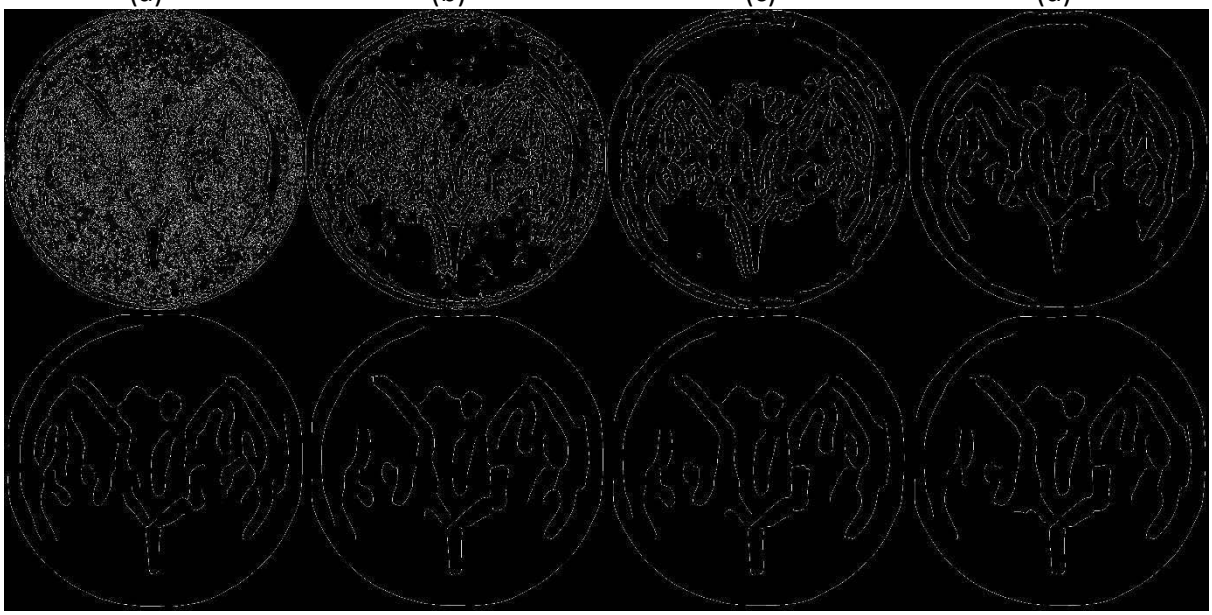
2.2.1 The "Canny" edge detection

To use the "Canny" method, function *edge* requires an input value for sigma to specify the standard deviation of the Gaussian filter and another value for threshold to determine the sensitivity of edge detection. The default sigma value is one (1). The threshold value, if not specified, can be chosen automatically by the function itself.

Setting the sigma value to either 16 or 32 was found to be the best compromise between the complete detection of logo edges and false positives (edges originating from anything other than the logo itself). An example showing the difference in edge images when the sigma value is changed is given in Figure 2.3. Lower sigma values led to false positives dominating the edge image due to insufficient smoothing (Figure 2.3 (a) – (d)). As the sigma value increased, the false positives disappeared along with the weak logo

edges. Above a sigma value of 64, the edge image no longer changed significantly as only edges detected at positions with significant contrast differences remained (Figure 2.3 (g), (h)). It was decided that instead of manually choosing a sigma value for each image, all images would be subjected to the same value of 32. It was believed that it was more favourable to have fewer false positives in a logo edge image than trying to accommodate those weak logo edges and compromising the logo searching results with the presence of extra false positives.

(a) (b) (c) (d)



(e) (f) (g) (h)

Figure 2.3 (a)-(h): "Canny" edge detection with different sigma values. Row 1 (a)-(d): Sigma = 1, 2, 4, 8; Row 2 (e)-(h): Sigma = 16, 32, 64, 128. Refer to the Appendix for a larger, clearer version.

After settling on the sigma value, an attempt to adjust the threshold was also carried out so as to extract more logo edges from the image (see the example in Figure 2.4). Emphasis was put on threshold values that were close to that determined by the function itself (Figure 2.4 (a)). Lower threshold values would include more edges and false positives (Figure 2.4 (b) – (d)) while higher values would remove the false positives, at the expense of losing some genuine logo edges (Figure 2.4 (e) – (g)). The automated threshold actually did a good job dealing with the dilemma that an edge image either had an appropriate number of genuine edges along with some false positives, or very few genuine edges without any false positives.

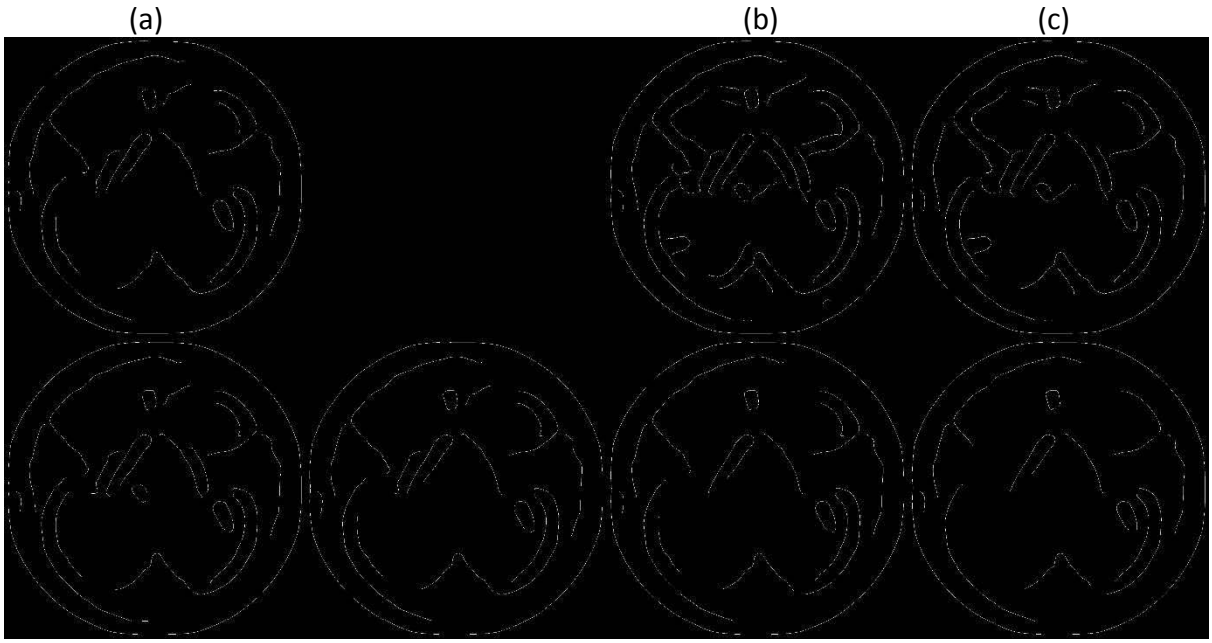


Figure 2.4 (a)-(g): "Canny" edge detection with sigma = 32 at different thresholds. Row 1 (a)-(c): Threshold = 0.1563 (auto), 0.1, 0.12; Row 2 (d)-(g): 0.14, 0.16, 0.18, 0.2. Refer to the Appendix for a larger, clearer version.

Based on the above observations, the methodology for applying the “Canny” method on these AFP images would be to let the function choose an appropriate threshold value, but for the user to specify a fixed sigma value of 32. This means that users do not need to determine a threshold value for each image, a time-consuming task.

2.2.2 Other edge detection filters

MATLAB also allows users to choose other filters for edge detection. These include the “Laplacian of Gaussian” filter, the “Sobel” filter, the “Prewitt” filter and the “Roberts” filter. A threshold can be specified when any of these filters are applied, or the algorithms can determine a threshold automatically. In addition to the threshold value, a sigma value can be assigned by users in the “Laplacian of Gaussian” method (otherwise a sigma value of two (2) is used as default). The edges detected in the AFP images using these filters were similar and they will be discussed together here.

When the threshold was not specified, the results were poor. These filters produced a lot more false positives and far fewer genuine edges (Figure 2.5) than the “Canny” method (Figure 2.3). In addition, the edges appeared as individual dots. The edges detected by the “Canny” method appeared as lines or contours of the logo (albeit broken sometimes). The reason for the difference originates from the way the edges are

constructed in different methods. Trying to adjust the threshold (and the sigma value in the "Laplacian of Gaussian" method) did not help to 'join the dots'. As the edges did not appear as lines or contours, it was hard to tell if the edges gained or lost by adjusting the threshold downwards or upwards respectively were genuine or not.

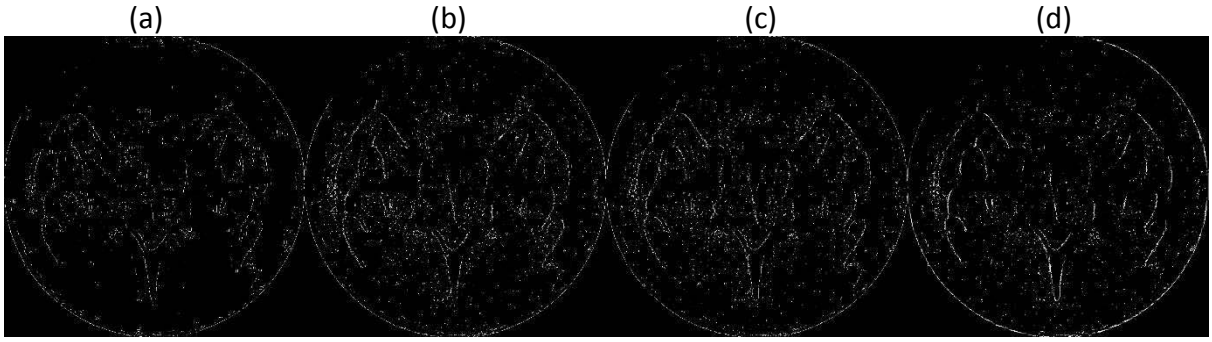


Figure 2.5 (a)-(d): An example of edge images using different filters. (a): The "Laplacian of Gaussian" filter; (b): The "Sobel" filter; (c): The "Prewitt" filter; (d): The "Roberts" filter. Refer to the Appendix for a larger, clearer version.

It was decided that having users determine the optimal threshold for each and every image was not the best way to carry out edge detection. There should be as little human intervention as possible to save time and labour, and so these methods were rejected as inferior to the "Canny" method.

2.2.3 Smoothing images before edge detection by resizing

To a certain extent, resizing images to smaller sizes is similar to smoothing them, as each output pixel value in the new image can be the weighted average of the neighbouring pixels. By averaging pixels nearby, smoothing is achieved. Computational speed improves as well as each image to be processed will be smaller. The MATLAB code *imresize* does exactly this.

The AFP images were reduced in size using *imresize* and all the above edge detection methods were tested again using automatically determined thresholds. Sigma values of two (2) and four (4) for the "Canny" method and the "Laplacian of Gaussian" method respectively were found to be suitable for the process. Improvements were obvious for the "Laplacian of Gaussian" method, the "Sobel" method, the "Prewitt" method and the "Roberts" method (Figure 2.6). The edges were not isolated pixel by pixel anymore but were forming lines and contours. Fewer false positives were seen, although they still occurred. However, as a consequence of this smoothing, only edges with high contrast

remained and the weaker edges disappeared (for example, compare Figure 2.4 (a) and the first image from the left in Figure 2.6 (a), both generated using the "Canny" method). Some details were lost as a result of resizing the photos. Although the edge images are still far from perfect as far as false positives are concerned, resizing the images removes a lot of these nonetheless, which is crucial. The "Sobel" method, the "Prewitt" method and the "Roberts" method can also be reconsidered for detecting edges in the smaller images as they now generate better edge images.

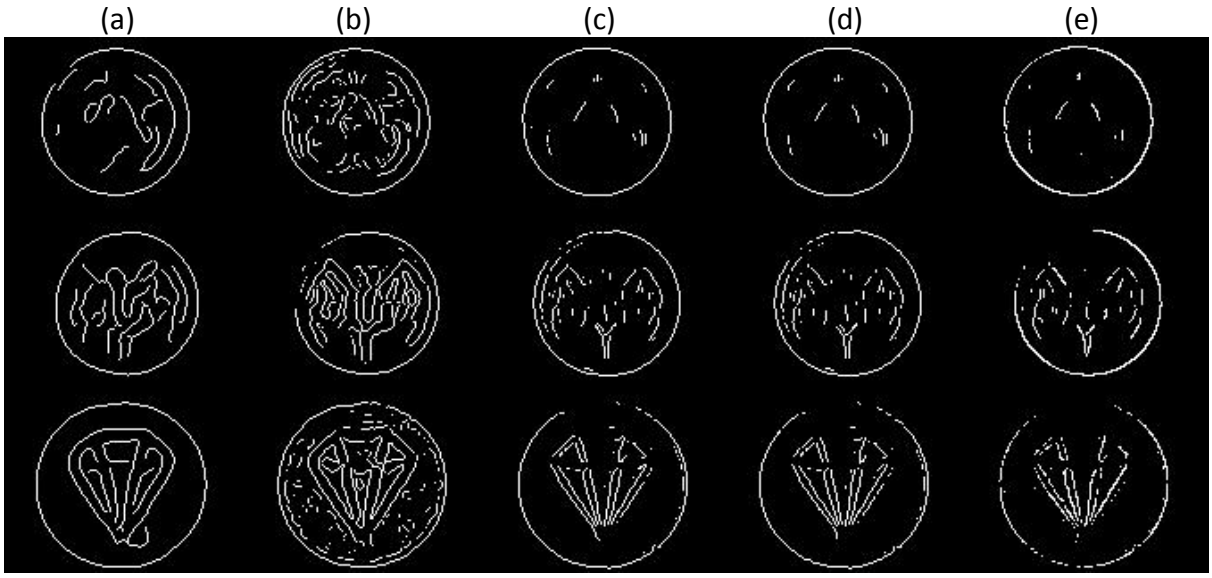


Figure 2.6 (a)-(e): Each row lists the edges detected using (a) the "Canny" method, (b) the "Laplacian of Gaussian" method, (c) the "Sobel" method, (d) the "Prewitt" method and (e) the "Roberts" method from the same resized photo.

2.3 Phase Congruency

It was observed that the illumination conditions of the images provided by the AFP varied greatly (see Figure 2.1). In the literature, *phase congruency* is said to be able to detect edges regardless of illumination and contrast (87). A MATLAB script by Kovesi, *phasecong3*, was used to generate phase congruency images for each AFP photo (87). This script has many user-specified arguments, but a decision was made to concentrate on the noise threshold point. Default values were used for the other arguments. The output images indicating the edge strength were subject to hysteresis thresholding for pinpointing the location of significant edges using *hysthresh*, another script by Kovesi (87).

A problem was encountered when the script was first used. The resources of the laptop used were not sufficient to handle the demanding computation of phase congruency of the images, and the photos had to be resized smaller so that the script could be run. This was a major limitation to the application of this technique for logo searching as computers had to be powerful enough to handle the calculations involved.

An attempt to determine the noise threshold point was first carried out. Again, the higher the threshold, the greater the number of genuine edges, and false positives were better suppressed. However, a general threshold that would give reasonable edge images for all tablet photos could not be easily determined. Some images could give good edge images at a low threshold (e.g. the threshold for Figure 2.7 (a) would be below 10). These were likely to be images of tablets which had homogeneous surfaces. The logo edges became more and more difficult to detect when the threshold value was increased because the phase congruency values dropped. Other photos needed a high threshold to prevent false positives that may originate from the colour granules/patches on the tablet surface or from poor illumination (e.g. the threshold for Figure 2.7 (b) would be close to 30). In order to pick up as many logo edges in as many AFP photos as possible, a high threshold (25) was chosen even though this often made it more difficult to detect the edges in the hysteresis thresholding step.

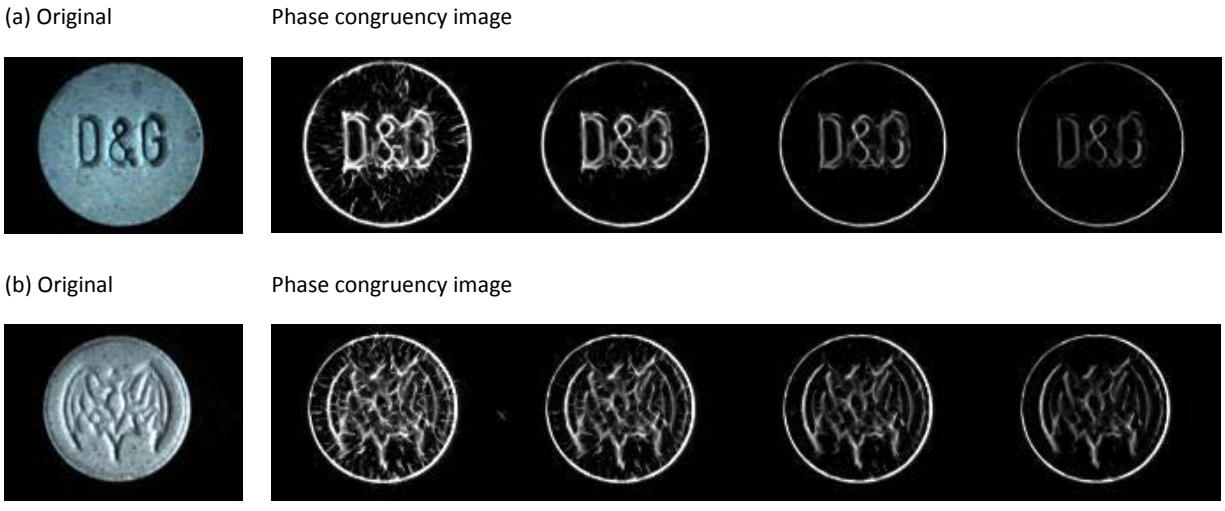


Figure 2.7: Scaled images showing the relative strength of phase congruencies as the noise threshold point increased from 2 to 10, 20 and 30. The true threshold for (a) would be below 10 while that for (b) would be above 20.

Hysteresis thresholding allows users to mark a pixel on a phase congruency image as an edge depending on a threshold value. From that pixel, any connecting pixels having

values greater than a second (smaller) threshold are marked as edges as well. Here, the solution to maximise the chances of detecting significant edges, regardless of their phase congruency values, is as follows. Instead of using any absolute numbers as the threshold values, they were dependent on the maximum value of the phase congruency image. Therefore different pairs of thresholds were applied in each image. Specifically, one-twentieth (1/20) and one-twenty fifth (1/25) of the maximum phase congruency value were used as the two thresholds (Figure 2.8). A higher upper limit would minimise the genuine edges identified but caused fewer false positives, while a smaller lower limit would reduce the resolution of the logos from clear strokes to clumps of pixels (Figure 2.8 (c)). Resolutions of edges became important because each edge was now defined by an area of connecting pixels instead of a line of pixels or a contour. A lower second threshold would cause more pixels to be connected to an area representing the same edge.

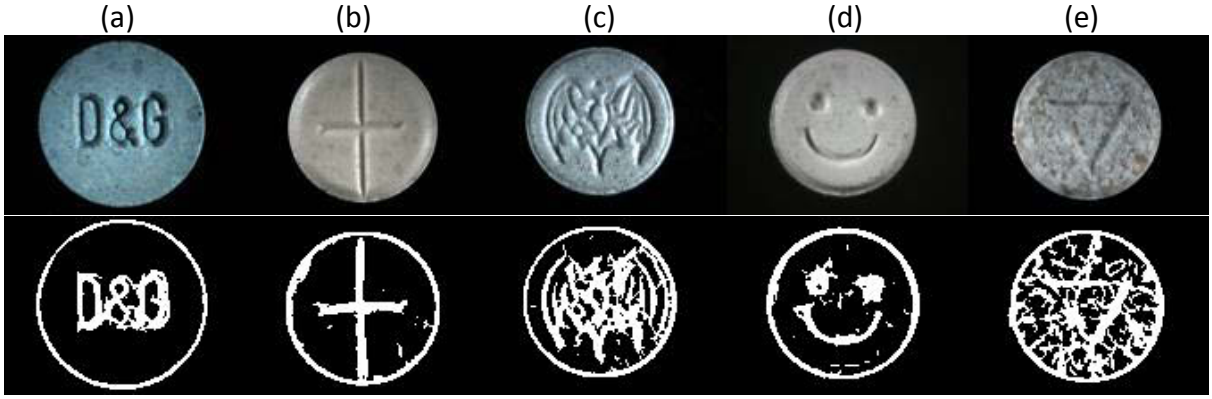


Figure 2.8 (a) – (e): Five final edges extracted from their respective original image using phase congruency.

If phase congruency is to be used in real casework, there are a number of problems to tackle. Apart from the fact that a more powerful computer is necessary for computing phase congruencies without resizing tablet images, the time necessary for processing each image is notably longer than any edge detection techniques. The average speed for performing edge detections using the “Canny” method is more than 10 times faster than the phase congruencies calculation speed. This method also does not suit tablets like Figure 2.8 (e) that are not homogeneous in colour. False positives were also difficult to prevent. However, some of the output images were quite satisfactory. The edges that were found filled parts of the logo area as silhouettes, instead of surrounding the logos as lines or contours with the thickness of one pixel (Figure 2.8 (a), (b) & (d)).

2.4 Conversion to BW images

With the results of the previous section in mind, it seemed desirable to be able to completely fill a tablet logo with pixels with a value of ones (1), like a silhouette, while suppressing all others as zeros (0). The problem at hand was similar to segmentation: in this case, we were trying to identify ways to separate a logo from the tablet surface.

The logical way to start was to nominate a threshold to be applied on the AFP images. The threshold could then be an intensity value between the lower average luminance value of the logo and the higher average luminance value of the surface. It would convert the tablet photos to black-and-white (BW) images, in which all pixels of the tablet surface were ones (1), because of their higher than threshold luminance values, and all of the logo pixels were zeros (0). From this point of view, a good BW image would be one which the logo pixels of zeros (0) were surrounded wholly by the surface pixels of ones (1). This criterion allowed the logo pixels be isolated from the background pixels, which were also zeros (0). A logo silhouette could be obtained by simple logical operations.

2.4.1 Determination of threshold by user input

A preliminary test was done on ten of the AFP photos by manually picking ten random points that were part of the tablet surface, but not the logo, for each photo. Locations with hotspots were also carefully avoided. The mean (μ) and standard deviation (s) of the intensity values of these ten points were calculated. Some values below the mean by multiples of the standard deviation were tested to find the optimal threshold for that image. The median and the first quartile of the ten intensities were compared as well. Two examples are shown in Figure 2.9. Any intensity above the mean would convert the surface pixels to zeros (0) as well, therefore those intensities were not chosen as thresholds for the comparison.

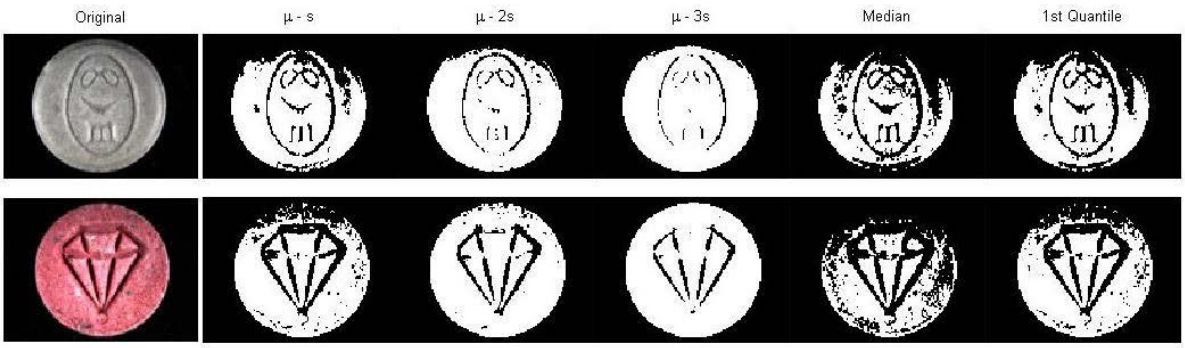


Figure 2.9: Two examples of converting two tablet photos into BW images using different thresholds.

Results showed that based on the criterion set at the beginning of this section, the best threshold would be somewhere between one and two standard deviations below the mean (μ - s and μ - 2s). Sometimes the first quartile of the ten intensities was also giving good segmentations but this seemed to be random. Possibly the first quartile was coincidentally close to the optimal threshold in those cases because these ten points were randomly chosen. The BW images created using the median as the threshold were the worst. A lot of the tablet surface pixels appeared as black (zeros) instead of white (ones). The threshold equal to μ - 3s converted most of the tablet surface to white pixels but the logo began disappearing and was not as complete as for other thresholds.

In order to pinpoint the optimal threshold, more values between one and two standard deviations below the mean were used for comparison (Figure 2.10). Unfortunately, no general rules to produce ideal BW images most of the time could be easily identified. It was decided that it was best to set the threshold at the lower end of the range under testing (i.e. μ - 2s). This would ensure the logo was isolated from the background although some logo details might be lost. A lower threshold would also minimise the number and the area of false positives (surface pixels that were not set as ones). Smaller false positives were easier to remove by morphological operations.

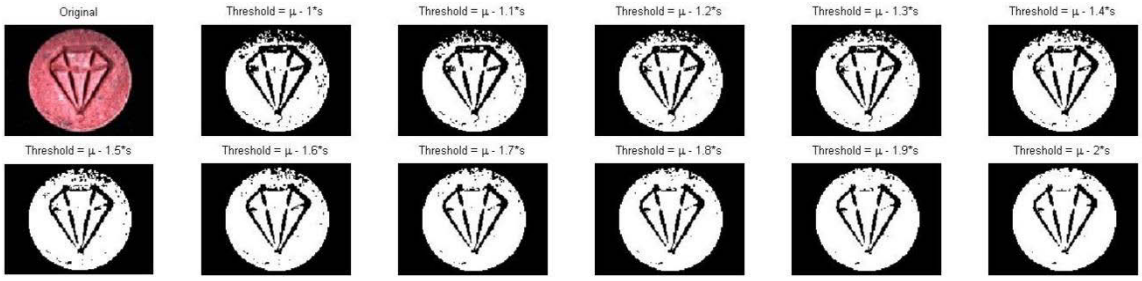


Figure 2.10: More testing on the optimal threshold.

As far as this project is concerned, there are two major disadvantages of this method. Firstly, user intervention is necessary. Examiners had to carefully pick ten points on every image they process. Time and labour costs are huge. Secondly, it is very difficult to reproduce the same results for a given image, mainly due to the fact that the points are chosen randomly. Different users will also choose these points differently. There is no guarantee that the same output will be regenerated a second time. How this will affect logo searching is unknown, because even repeated attempts by one user often fail to detect all of a given logo.

2.4.2 A tablet-specific BW conversion

To avoid user intervention, the MATLAB built-in function *graythresh* was tested for its ability to automatically determine a threshold for converting images to black and white. Based on a visual inspection of these black and white images, determining thresholds automatically using *graythresh* actually performed worse than the μ - 2s threshold tried in the previous section.

The only remaining option was to utilise another property of the AFP images: the colour values or intensities of pixels. From observations, the tablet surface occupied the second biggest area in an image after the black background. This can be seen by checking the image histograms of the greyscale images (Figure 2.11). In a typical histogram, there were two peaks (A and C). The major peak (A) was near zero (0) intensity, and it represented the background. Another lower and broader peak (C) was observed. This peak, at a higher intensity, had to be due to the illumination of the tablet surface. It was at different intensities in different photos/histograms because of the inconsistent lighting conditions and/or the colour of the tablet.

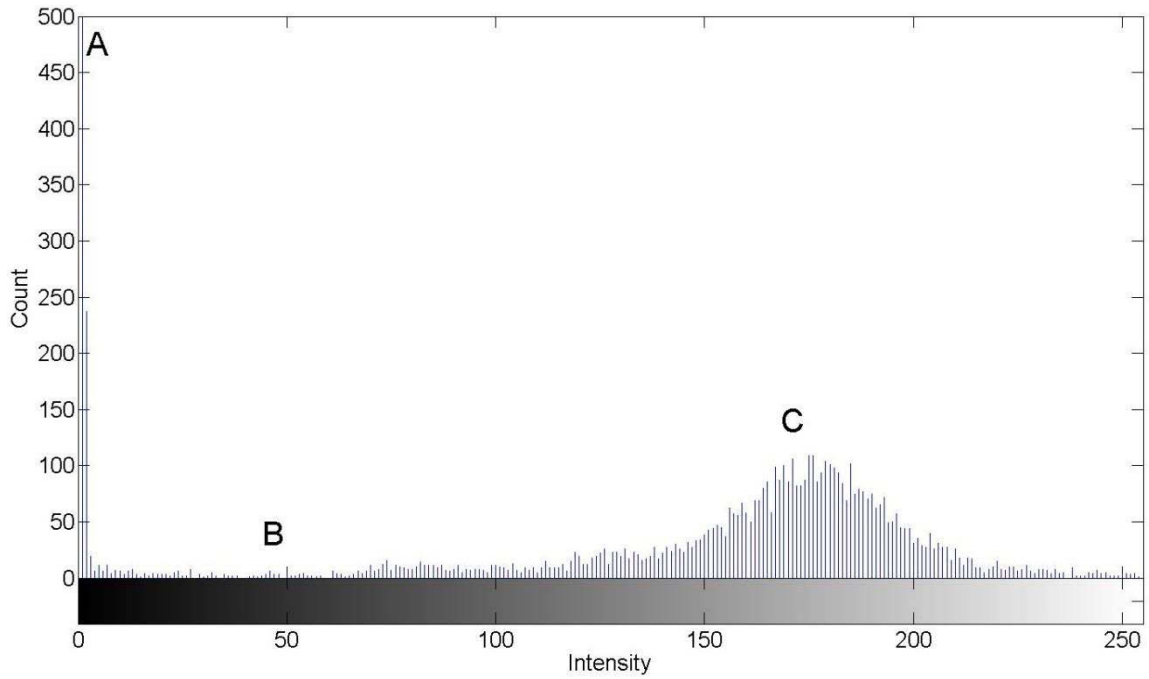


Figure 2.11: A typical image histogram (intensities at uint 8) with two peaks.

From the histogram (Figure 2.11), it could be concluded that the optimal threshold had to be some intensity values between point B and C. Point B was arbitrarily set to 50 (in unassigned-integer 8 (uint8) precision), where the trough between the two peaks occurred for the majority of the histograms. Our data showed that a threshold chosen at a point three-quarters of the distance from point B to point C was the best spot to achieve the ideal BW image for most photos. An example of different thresholds at different distances from point B to point C is shown below (Figure 2.12). A threshold closer to point B would set more logo pixels to ones (1). On the other hand, more tablet surface pixels would be converted to zeros (0) when a value closer to point C was chosen.

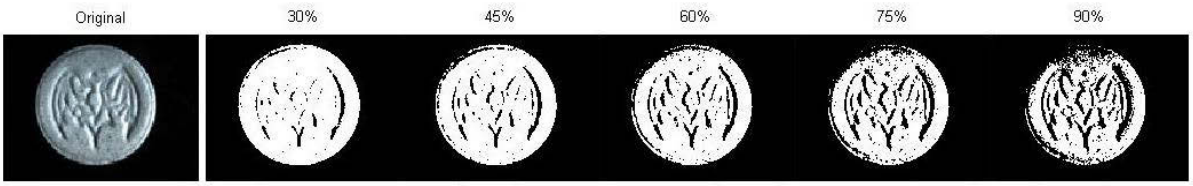


Figure 2.12: Thresholds at 30%, 45%, 60%, 75% and 90% away from point B to point C.

Figure 2.13: Some examples of converting BW images using our own threshold method.

2.5 Preliminary logo searching

To conclude the work on the existing AFP photos, a preliminary logo search was performed on the edge images generated from the different edge detection filters, and on the logo silhouettes, which had been segmented and binarised from the photos using the in-house thresholding technique described in Section 2.4.2. The edges and the silhouettes were encoded with the Hu moment invariants, complex moment invariants and Zernike moment invariants. The invariants were ranked against a query using the Euclidean distance. However misidentifications were common place and logos were ranked in random order. This was not surprising given the poor quality of the edges and silhouettes used to compute the feature vectors.

2.6 Summary

These logo edge detection and segmentation results were far from satisfactory. It was observed that some unwanted edges would always be detected using any of the methods discussed in this chapter. These often occurred when shadows introduced by the lighting source interfered with the logo boundaries. Other edges did not have enough contrast for the algorithms to detect their presence. Some edges were nearly impossible to detect even with the naked eye because the colours around those edges were so similar. Poor results were also obtained in trying to extract the whole logo silhouettes by segmentation techniques. Only parts of the logos were segmented most of the time. It was concluded that the majority of the images did not have the

appropriate lighting conditions for logo extraction, and that therefore a different photographic method was required, particularly in terms of tablet illumination.

Chapter 3: NEW METHODS FOR CAPTURING TABLET IMAGES FOR IMPROVED LOGO EXTRACTION

Chapter 3: New methods for capturing tablet images for improved logo searching

After applying different image processing techniques to try to utilise the existing AFP illicit drug tablet photos for logo edge detection (without huge success), the focus was turned to a new imaging protocol such that the logo information can be captured and extracted easily.

A pixel in a digital image, in whatever format, is made up of numbers that convey the colour at that particular position. All image processing techniques are basically mathematical operations on a pixel and its surroundings. For example, the main idea of edge detection algorithms is to determine whether the change in the colour vector at any particular pixel relative to its neighbours is significant, given a particular threshold. Some segmentation techniques group pixels that have similar colour vectors.

The main reason that the existing AFP tablet photos did not permit the extraction of pure logo edges and shapes was that the contrast between the logo and the tablet surface on these photos was often quite low. At the same time, due to the poor and inconsistent lighting conditions, various colour differences were present on the tablet surfaces that were picked up as edges and logos (false positives). Conventional image processing failed because the edges of a shaded area may be easier to detect than logo/surface boundaries. Current AFP protocols only demand an image such that the details of the tablet are perceptible by visual inspection. This highlights the sophistication of the human brain: examiners are able to distinguish a logo from the surface with little contrast and in the presence of background noise, but the task is impossible for conventional image processing algorithms. Therefore, an alternative technique for capturing tablet images is necessary to address this problem. A simple, yet efficient solution is to manually enhance the contrast between the logo area and the tablet surface while suppressing any colour differences on the tablet when an image is taken.

3.1 Method

A real pill press and eleven genuine dies (nine different logos, a “score” and a blank) were made available by the AFP to press new tablets for these tests (Figure 3.1 - Figure 3.3). A total of 300 tablets were pressed: five pills for each combination of the ten dies (excluding the “score” die, which is always on the opposite side of the logo for each pill) and the six colours. The logos, except the blank, are shown in Figure 3.4 while the colours are shown in Figure 3.5 (more details about the colours are given in Section 4.1). Excluding the coloured dyes, each tablet was made of 96% cellulose and 4% magnesium stearate. Some other pharmaceutical tablets were used in other tests (Section 3.2). Although 300 tablets were pressed, this number was excess to the requirements and so the unused pills were kept as backup replicates.

Figure 3.1: A genuine pill press.

Figure 3.2: A close-up of the pill press.

Figure 3.3: The dies used to press pills.

Figure 3.4: An example of the tablets pressed by the dies. A blank tablet pressed using a blank die is not shown here. Row 1: “Yin and Yang”, “8-ball”, “apple”, “three-pointed star 1” (“Mercedes”) and “three-pointed star 2” (“peace”); Row 2: “Euro”, “butterfly”, “kangaroo”, “e” and “score”.

Figure 3.5: The six colours used to dye the tablets.

A subset of the pills and tablets were imaged under different lighting conditions and protocols as described in Sections 3.2 and 3.3. A Leica DFC295 camera mounted on the Leica MZ6 Macroscope was used to collect these new images, which were processed using MATLAB installed on the same standard-UTS issued laptop described in Section 2.1. The new images used in this chapter were subjected to the same logo extraction methods that were previously applied to the existing AFP images for comparison. The same four edge detection techniques described in the previous chapter were applied on this batch of new images. The tablet-specific BW-conversion devised in Section 2.4.2 was also tested.

The database used in Section 3.4 below for logo searching was a collection of images taken from the tablets pressed for our project. The table below is a summary of the contents of this collection (Table 3.1).

Table 3.1: A break-down of the number of tablet logo images in the database for logo matching.

Logos/Markings	Colour of tablet	Number of photos
Yin and Yang	Blue	5
Yin and Yang	Green	1
8-ball	Blue	3
8-ball	Green	1
Three-pointed star 1 (“Mercedes”)	Blue	4
Three-pointed star 1 (“Mercedes”)	Green	1
Three-pointed star 2 (“peace”)	Blue	4
Three-pointed star 2 (“peace”)	Green	1
Euro	Blue	4
Euro	Green	1
Kangaroo	Blue	14
“e”	Blue	4
“e”	Green	1
Apple	Blue	12
Butterfly	Blue	12
“Score”	Blue	10
Blank	Blue	3
Blank	Green	1
	Total	82

3.2 Extracting the logo from multiple images

The first proposed solution to the problem described above was to take multiple images of the same tablet. Keeping the focus and magnification constant without changing the positions of either the camera or the tablet, several images were taken with the lighting source being placed in different orientations. The aim of illuminating the tablets from different directions was to introduce shadow (contrast) between the logo and the tablet surface that would not occur if the source was directly above the tablet (uniform lighting). During testing, four different lighting positions (two pairs of twin swan lights) were used and thus four images per tablet were captured. A ring light with independent switches to four groups of LEDs was also tested. An example is shown in Figure 3.6. New images were generated, pixel by pixel, by first extracting the greyscale values of the four images at each pixel, then comparing these four values and finally using either the minimum value, the 2nd minimum value, the median value or the average value of the four numbers as the greyscale value of the new images at that pixel (Figure 3.7). Four sets of these combined images were thus produced. The new images were then processed using different edge detection techniques and the tablet-specific BW-conversion (refer to previous description in Section 2.4.2) to isolate the logo area. Commercially available pharmaceutical tablets were used for the photos as real illicit drug tablets were not accessible at the time. The results are described below.

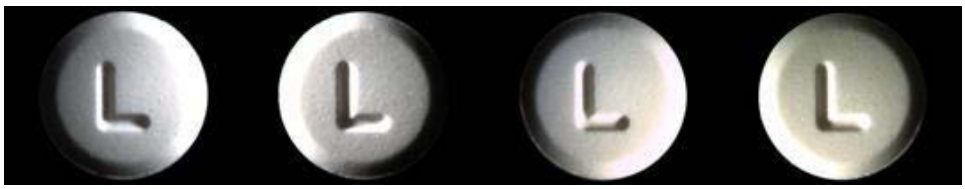


Figure 3.6: Four images of the same tablet with four different light locations.

Figure 3.7: Left four images: Images generated by the four images as shown in Figure 3.6 using minimum, 2nd minimum, median and average pixel greyscale values respectively; Right image: Same tablet under uniform lighting.

3.2.1 Logo extraction by edge detection

The “Canny”, “Laplacian of Gaussian”, “Sobel”, “Prewitt” and “Roberts” methods (as implemented in MATLAB) were tested. Examples of the edge images are shown in Figure 3.8. When these methods were applied on the new combined images, the quality of the edges was better than those from images using uniform lighting in general (column five of Figure 3.8). Under uniform lighting (as in the AFP database, using a pair of Twin Swan lights), no logo edges were detected on tablets with “shallow” logos. However, by combining images, those logo edges would be detected. Edges were readily detected by all of the methods on tablets with deep logos under uniform lighting. Edges were detected successfully in the combined images except when the “Laplacian of Gaussian” filter was applied. Fewer edges were detected with this filter than others.

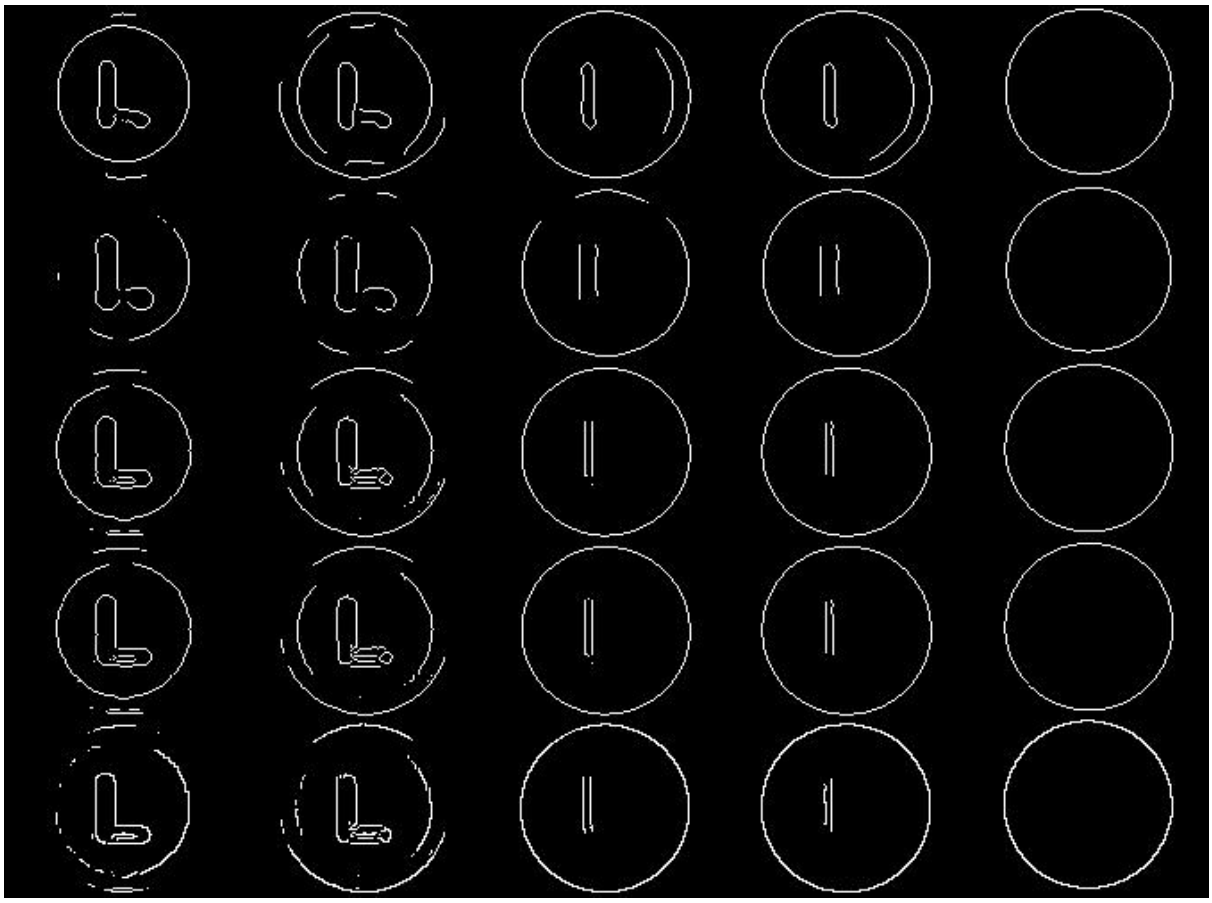


Figure 3.8: An example of edges detected using the five images in the order same as Figure 3.7. Row 1: the “Canny” method; Row 2: the “Laplacian of Gaussian” method; Row 3: the “Sobel” method; Row 4: the “Prewitt” method; Row 5: the “Roberts” method.

New images that were generated from the minimum intensities of the four differently-illuminated images produced similar edge images using the “Sobel”, “Prewitt” and

"Roberts" methods (column one of Figure 3.8). All of the edge filters detected more complete logo contours on these images than other new images. However, these edge images contained the most false positives no matter which edge detection technique was used.

Selecting the 2nd minimum intensity of the four "constituent" images to create a new image, we found the edges detected were also visually similar irrespective of whether the "Sobel", "Prewitt" or "Roberts" method was used (column two of Figure 3.8). These edge images contained fewer false positives but the logo edges were not as complete as those generated from the minimum intensity images.

The edge images detected using the median and average values were nearly identical except that the latter had fewer false positives. This similarity implied that the median and average image themselves were fairly similar. Once again, the "Sobel", "Prewitt" and "Roberts" methods gave similar edge results, and none of the five edge filters could detect edges from "shallow" logos. The "Laplacian of Gaussian" filter failed to detect logo edges in either of these two sets of images.

These results were considered a slight improvement because detecting "shallow" logos was possible from images generated using the 2nd minimum intensities, but was rarely successful when uniform lighting on tablets was used. However, the combination of the new types of images and the edge detection techniques could not provide a true and complete contour of a logo without false positives.

3.2.2 Logo extraction by converting to BW images

As well as edge detection algorithms from MATLAB, the tablet-specific BW-conversion (as described in Section 2.4.2) was also applied to these new images. This conversion returned a blank image for most of the trial tablet images taken using uniform lighting. When it was applied to the four sets of combined images, results were generally better: the logo area and the tablet surface had enough contrast to be segmented if the tablets did not have "shallow" logos (see Figure 3.9 as an example). This limited its use as a generic logo extraction algorithm. Moreover, there was no clear choice of which operation would produce the best contrast. Different operations produced better results on some of the images but not other ones. The two sets of combined images

using the median and the average values were again thought to be similar as they produced similar results.

Figure 3.9: An example of converted BW images by our own tablet-specific BW conversion method using the five images in the same order as Figure 3.7.

Applying intensity transformation algorithms from Gonzalez et al. on the original images did not improve the generated images or the segmentation results. When the ring light was used, its ability to individually turn on and off groups of LEDs was exploited. Attempts were made to take four images by turning *off* the four groups of light bulbs one at a time on the same tablet. This was the opposite of the above method in which the light bulbs were turned *on* a group at a time. However, based on visual inspections, the segmentation results generated by images using either light settings (turning *on* or *off* a group of light bulbs one at a time) were similar.

In addition to the drawbacks specific to the image processing techniques used, there were a few more in general. Firstly, in addition to a separate image for colour analysis, more than one image was necessary for logo extraction, as those raw images were not suitable individually due to the purposely induced shades. This would necessitate more storage space and the logo searching would be more computationally expensive. Extra time would also be required for the labour-intensive work of taking multiple photos. Secondly, it was evident from the extraction results that the directional lighting produced bright and dim patches that could introduce false contrasts and hence false edge detection and logo segmentation (especially if two pairs of twin swan lights were used). It did not meet the objective of suppressing colour differences on the tablet surface in the new images. In the generated images, one could see that the light source in some of them was not equidistant from the tablet or was angled inconsistently towards the tablet. The four light sources would have to be placed perpendicular to each other around a tablet for generating maximum contrast around the tablet. A procedure to standardise these settings to give reproducible results in future tablet examination was impossible with the equipment used. Although taking more images from more

directions may reduce the effect of inconsistent lighting conditions, more images would need to be processed (which would be even more demanding on the examiner and the computer). Increasing the number of images taken from all directions would eventually generate images not dissimilar to an image that was illuminated uniformly on the top, making all the raw images redundant. Using a ring light in this experiment instead of the two pairs of twin swan lights was the solution for more consistent lighting and it succeeded in addressing some of the issues above. However it is still more favourable to only have one image for logo extractions to reduce the examination time and computation resources. Furthermore, a method which would accommodate tablets with “shallow” logos is definitely preferred.

3.3 Adjusting the height of the ring light relative to the tablet

Another solution is to move the light source much closer to the tablet (Figure 3.10). It was discovered accidentally that this could be very effective in inducing contrast between the logo area and the tablet surface. With this contrast, different methods could be used to successfully segment the two regions. The general principle is explained in the diagram below (Figure 3.11). When the ring light was sufficiently lowered such that it was just above the tablet, the light rays that were hitting the tablet surface would be reflected upwards and be picked up by the camera. However, due to the acute angle of the rays towards the tablet, the step between the tablet surface and the logo became a barrier in which the rays were not able to directly shine on the logo region. Having much less light per unit area in the logo region than on the surface meant that a large difference in the light intensities was observed. However, as only a small proportion of the light rays from the light source were being reflected and captured on the camera, tablet colours in these images looked darker than what were perceived with the naked eye. Therefore these images could only be used for logo extraction, and not as a record of the general appearance/colour of the tablet.

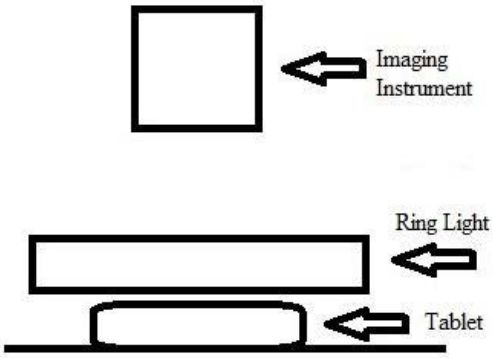


Figure 3.10: The new proposed imaging setup.

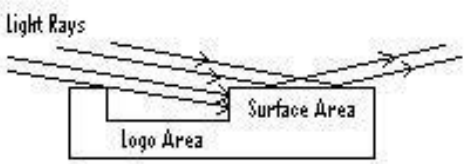


Figure 3.11: Only the light rays hitting the surface area are reflected.

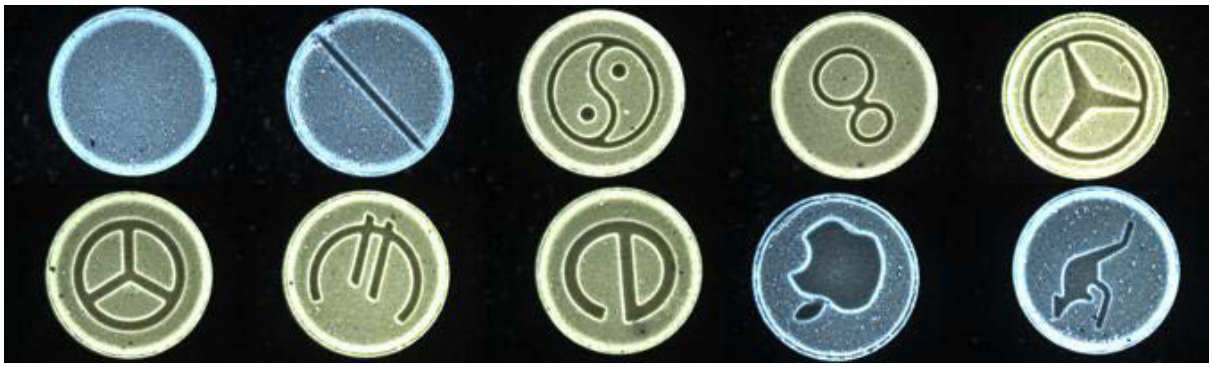


Figure 3.12: Examples of images taken with the ring light lowered towards the tablets. Row 1: “blank”, “score”, “Yin and Yang”, “8-ball” and “three-pointed star 1”; Row 2: “three-pointed star 2”, “Euro”, “e”, “apple” and “kangaroo”.

3.3.1 Logo Extraction by edge detection

The quality of the edges detected was very high in general using the “Canny”, “Laplacian of Gaussian”, “Sobel”, “Prewitt” or “Roberts” methods (Figure 3.13). Most edges corresponded to the location of the actual logo contours. It was encouraging that some edge images had no false positives. Apart from the “Laplacian of Gaussian” method, the filters detected even fewer false positives than before. For these methods, no one out-performed the others and any method would produce satisfactory results. It was common that one of the four methods generated excellent edge images from some of the tablet images but did not perform as well on the others.

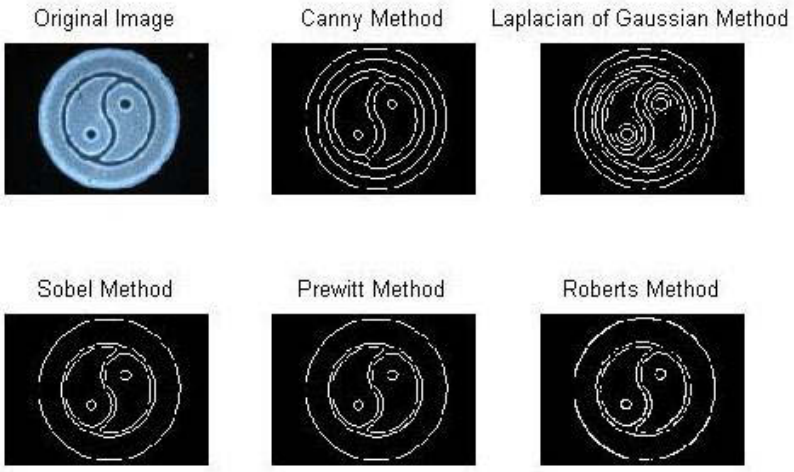


Figure 3.13: An example of edge detection results using different filters

An explanation for the observation that all of these edge detection techniques (including the "Laplacian of Gaussian" method) worked well was that this imaging procedure was able to introduce enough contrast between the tablet surface and the logo such that any filters on these image processing algorithms would be able to determine a change in shade. With enough contrast, any of the "tried-and-tested" methods would perform. This proved that this new imaging method could achieve the aim of enhancing contrast in tablet images.

Most of the false positives were caused by granules of dye present on the tablet surface (Figure 3.14). They may not have been mixed well with the ingredients but illicit drug tablets generally do not have such homogeneity of colour anyway. Any logo extraction method has to be able to adapt to this and identify whether contrast results from genuine logo shades or from randomly located colour granules. Unfortunately, edge detection techniques failed to differentiate these two phenomena. The false positives may also have come from the fact that the tablets were imperfect. It is not uncommon for the AFP to seize illicit tablets whose surfaces are not smooth or are partly chipped off. Edge detection treated these imperfections as edges of the logo.

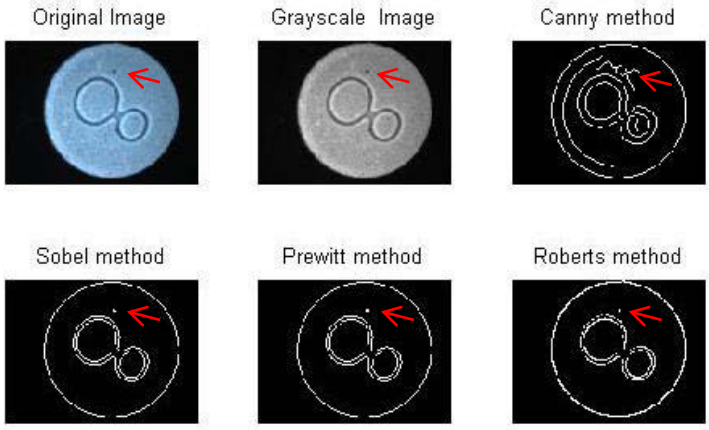


Figure 3.14: An example of false positives generated from a granule of dye

3.3.2 Logo extraction by converting to BW images

The in-house tablet-specific BW-conversion (Section 2.4.2) was also performing satisfactorily (Figure 3.15). It had no problem classifying the tablet surfaces as whites (1s) and logos with narrow strokes as blacks (0s). In these BW images, the false negatives (black pixels wrongly classified as logo regions that should have been white, i.e. the surface) occupied small areas, with a few pixels that could be easily removed by techniques such as morphological operations.

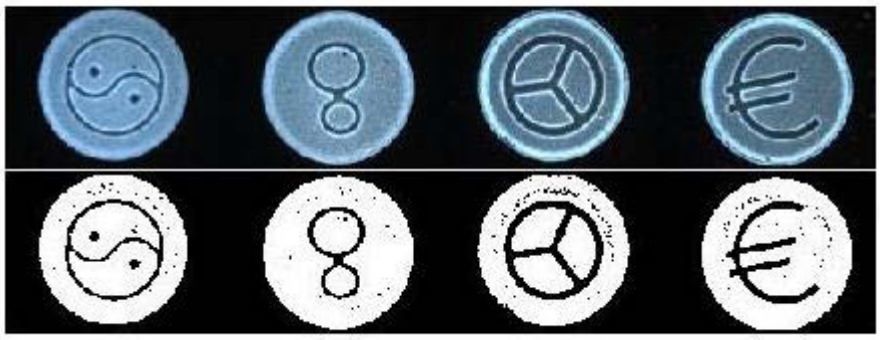


Figure 3.15: Some examples of BW images segmenting the logos out of the tablet surfaces.

For tablets with logos that had a large area, the segmentation results were initially of poorer quality. Large areas of false negatives were seen on the BW images. Two major contributing factors were blamed. The first was that initially the operator had to manually support the light and perform fine adjustment by hand. The optimum light position could not be achieved and hot spots were present in the middle of the logo in

some of the tablet images. A larger logo needed longer shadows from the steps between the logo and the tablet to cover its area. This meant that the light rays had to be directed at the tablet at a more acute angle. Setting the ring light too high would produce no shadows as the light rays did not hit the logo region at an acute enough angle. On the other hand, little or no light would hit the tablet surface if the ring light was positioned too low. When a purpose-designed ring light holder became available to image tablets, the logo extraction results improved vastly. The conversion was able to correctly segment the logo area from the surface in most cases.

The second factor was that some images were found to have a large number of pixels having saturated intensities (255 in greyscale uint-8 images; see Figure 3.16(b)). Using the BW-conversion, the threshold for all those images would be three-quarters of the distance between 25 and 255, i.e. 198 (Figure 3.16(c)). In designing the script, it was assumed that the intensity with the highest occurrence was the average value for the tablet surface. However this assumption was wrong in these images as the pixels with saturated intensities were from both the tablet surface and from the middle of the logo region. To fix this, a condition had to be added to the script such that the intensity taken as the average value for the tablet surface would not be the saturated intensity. More specifically, the intensity range from which the value was chosen was set to be between 50 and 204 (20-80% of the whole uint-8 range; see Figure 3.16(d)). Improvements were now seen on images which previously had thresholds of 198. Now their thresholds were determined by image intensities and would not be affected by the presence of hot spots in images.

The tablet-specific BW-conversion was used to set the threshold for converting images to black and white. The MATLAB built-in algorithm for determining thresholds, *graythresh*, was also used as an alternative to our threshold values for comparisons based on these new images. The thresholds determined by the in-house method were superior to those from the MATLAB function. They were closer to the real threshold necessary for optimum results (Compare Figure 3.16(d)-(f)). The quality of the logo silhouettes was better: complete and not fragmented. Therefore the in-house algorithm was preferred over *graythresh*, despite its added complexity.

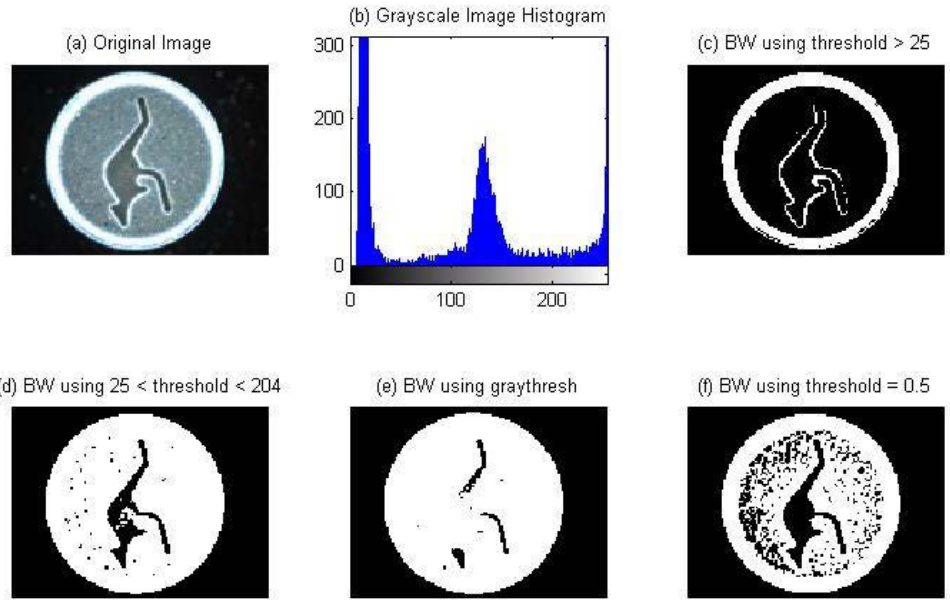


Figure 3.16: Another example of logo extraction by converting to BW.

From these logo extraction results, two drawbacks are apparent for taking tablet photos with the ring light at the lowered position. Firstly, special care has to be taken to image tablets which have logos with a large area. Fine and precise adjustments are necessary to achieve the right height and therefore the optimum angle of illumination. Secondly, another tablet image is definitely needed for colour analysis and searching, as the colour of the tablet in the high contrast image deviates from what an examiner would normally perceive. However, of two different imaging methods (shining the light onto the tablet at different angles or at a lower height), the latter method is preferred. While both methods require a separate image for colour matching, the latter allows logo extractions to be carried out using only one image. This has the huge advantage of saving time and computer resources.

3.3.3 The final logo image for searching

The results above show that using the BW-conversion method developed in this work on the new high-contrast images is an improvement on conventional edge detection. The final logo BW image consists of a solid white logo shape in which all pixels have a value of “1”. Moment invariants of these pixels would describe the logo better than those derived from an image with only a few pixels representing an outline (an edge image). It is believed that it is easier to clean up the false positives (pixels from the

surface that were classified as being from the logo region) and false negatives (pixels from the logo that were classified as being from the surface) using morphological operations than to remove the edges of a tablet from an edge image without removing the edge of the logo itself.

The method for generating the final logo BW image for moment invariant calculations is illustrated in Figure 3.17. First, the original tablet photo is converted to a black and white image using a threshold determined by the in-house tablet-specific BW-conversion method (Figure 3.17(a)). Then logical operations are used to keep the background as zero, then change the tablet surface to zero and the logo area to one (Figure 3.17(b)). A script in MATLAB called *bwlabel* is used to identify all "objects" present. Only those that exceed a minimum number of pixels are assumed to be part of the logo and are included in the final BW image (Figure 3.17(c)). This ensures that clusters of white pixels detected from blank tablet images (false positives) are not regarded as logos. Choosing only the largest single "object" as the logo is not favourable, as some logos are made up of different parts (e.g. "Apple"). Finally a condition is put in to test whether filling all pixels inside the "object" should be carried out. From the experience with these images, it was decided that filling the whole object would be carried out if the area to be filled in is less than 50% of the total logo area. Filling the zero pixels inside the "object" is necessary if the logo area is large and the photo taken is not ideal. For example, a photo with hot spots in the middle of the logo will produce a black hole (zero-valued pixels within the logo) after the processing described above (Figure 3.17(c)). Such a hole can be filled up at this stage (Figure 3.17 (d)). However, there may be times when filling the black holes is not desirable (see the example in Figure 3.18), and so the condition described above will prevent this from occurring.

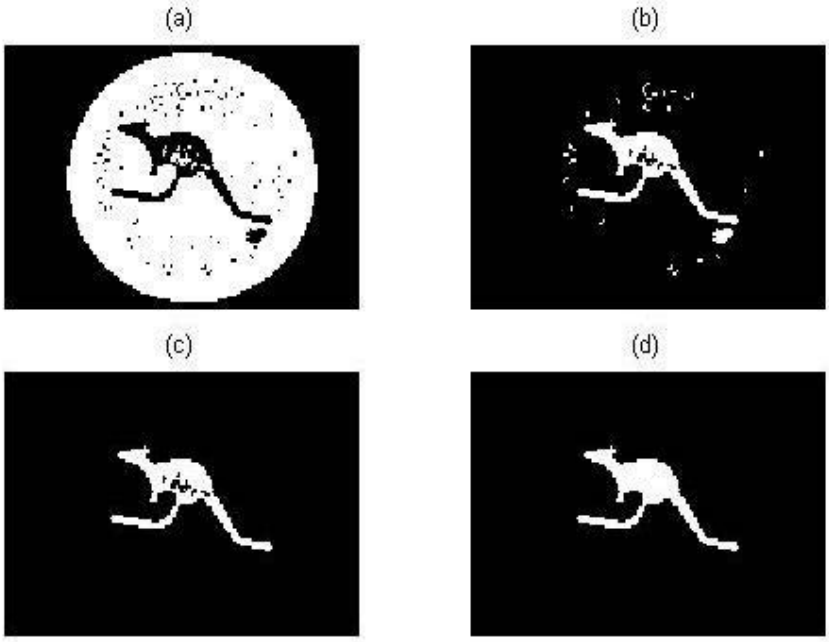


Figure 3.17: The flow to the final logo BW image from (a) to (d).

Figure 3.18: A logo where filling the black, zero-valued pixels inside is undesirable (left); the result of such filling (right).

3.4 Logo searching

The focus in the logo searching part of this work is on moment invariants, which are commonly used for pattern recognition (60). They are translation-, rotation- and scale-independent shape descriptors encoded as a row of numbers (a vector). Computation is quick and easy as pre-existing MATLAB scripts are readily available (60). Computers with powerful processing abilities are not necessary.

In particular, the complex moment invariants and the Zernike moment invariants as discussed in the book by Flusser et al. were tested (60). The Hu moment invariants are deliberately left out in our choices of moment invariants for tests as they are known to be incomplete and mutually dependent (60). Some other moment invariants are also

left out as they are invariant to projective deformation (e.g. a non-perpendicular view of the tablet from the camera), elastic transformations (e.g. a logo imprinted on a curved surface) or radiometric degradations (e.g. poor focus or camera vibrations when capturing the images). These scenarios will not arise in the logo images if the standard operating procedure is adhered to.

In a database containing 82 logo images (see Table 3.1 in Section 3.1), a query image was used to search against the remaining 81 images. The aim was to produce searching results which rank identical or similar logos before any others, by comparing their moment invariants. An ideal search result is shown below in Figure 3.19. A suitable searching algorithm for this purpose was determined and will be outlined in the following sections.

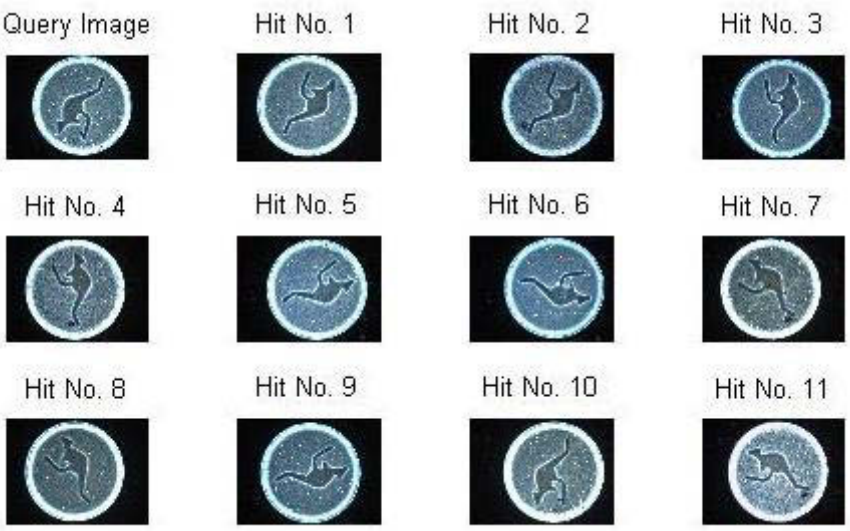


Figure 3.19: An ideal search result generated by comparing the complex moment invariants using Euclidean distance.

3.4.1 Complex moment invariants

The script for calculating complex moment invariants by Flusser et al., rotmi.*m*, includes magnitude normalisations to both order and degree (60). The third-order invariants ($r = 3$; $p_0 = 2$; $q_0 = 1$) with and without these normalisations were extracted from all logo images and these rows of six numbers were compared. The six numbers represented these invariants: c_{11}, the real and imaginary parts of $c_{20}c_{12}^2$, $c_{21}c_{12}$ and the real and

imaginary parts of $c_{30}c_{12}^3$. Different searching algorithms were also tested, in order to find the most suitable combination.

3.4.1.1 Normalisation experiments

Without the normalisations to both order and degree specified by Flusser et al., the 82 rows of invariant values were spread across a large dynamic range (10^0 – 10^{-11}) as anticipated. When logo images were ranked by their similarity to a query image using the Euclidean distance, the results were not ideal. Using a "Yin and Yang" logo as the query, for example, the five other "Yin and Yang" logos in the database were ranked first, sixth, eighth, 10^{th} and 15^{th}. Those logos that were ranked above 15^{th} were false positives and should have been placed behind these five logos. It was observed that in general, logos with a similar first value to the query logo would be ranked higher than the rest. This is not unexpected as the first invariant variable had a high range that dominated the features expressed in the remaining values. It was clearly more desirable for all six invariant values to have equal weights in any similarity ranking.

The next step was to try normalising these invariants by dividing them by their standard deviations. The result was that the dynamic range was still significant but it was narrowed, and the rankings improved as well. With the same "Yin and Yang" logo as the query, the other five "positives" were ranked first, second, third, fifth and ninth by the Euclidean distance. This showed that range normalisation is important for logo matching.

Normalisation to the range could also be achieved by adjusting the weights of the six invariant values in Euclidean distance calculations. This has the same effect as using the weighted Euclidean distances in searching. An add-in in Excel called *Solver* was used to optimise the weights for each invariant. A few attempts were made but ideal results could not be achieved. Initially, it was attempted to set the weights under a constraint such that the distances between invariants of the same logo were the shortest. It was later realised that such a condition would lead to multiple sets of weights for different logos, when the weights were supposed to apply to all the invariants irrespective of which logos they represent. In another attempt, the averages of the moment invariant vectors from tablets with the same logo were found and then the distances between the averages were calculated. Weights were computed under the constraint that the sum

of these distances was the greatest. Again, it was realised that this would not work, as constraining a real number to a maximum was meaningless. Therefore, the condition was changed slightly such that the *longest* distance calculated was set to 1 arbitrarily. Using the "Yin and Yang" logo as the query, the rankings for the five "positives" were worse than before at first, third, fourth, seventh and 25th. The condition was changed yet again this time, such that the *shortest* distance calculated was set to 1 arbitrarily. It was hoped that by separating the two closest average invariants by a fixed distance, invariants of different logos would be placed further apart. However, the rankings achieved using the same query were no better at second, eighth, ninth, 19th and 26th.

3.4.1.2 Normalisation to both order and degree

After normalisation to both order and degree, the invariants extracted from the logo images had a narrow dynamic range (10^0 – 10^{-1}). When the logo images were ranked again against the same "Yin and Yang" query logo using Euclidean distance, the outcome was much better. The rankings of the five logos achieved were first, second, third, sixth and 22nd. Better searching results were seen when a "kangaroo" logo was used as the query. Out of the thirteen logos present within the database, twelve of them were ranked top of the list. The last one was ranked much lower (45th) because some parts of the tablet surface were extracted to the final logo BW image as well and therefore the invariants were different from the others. Another logo search which produced good results was when a "Euro" logo was used as a query. The four matching logos in the database were the top four hits.

In the database, there were two groups of five logo images that had very similar but not identical "three-pointed star" logo designs (see Figure 3.12). Using one of those ten logos as the query, all other remaining nine logos were ranked randomly within the top hits by the Euclidean distance. As Figure 3.20 shows, the four logo images that exactly matched the query were not ranked as the first four but were ranked first, second, third and eighth. The other five logo images were ranked fourth, fifth, sixth, seven and ninth. This was considered an excellent result as the complex moment invariants and the Euclidean distance could rank logo similarity in a manner very similar to user perception.

Figure 3.20: Searching a "three-pointed star" logo using Euclidean distance.

Invariants normalised to both order and degree were clearly the data of choice for logo searching and for testing and comparing other searching algorithms (discussed below). The narrow dynamic range prevented any one variable overriding the others.

3.4.1.3 Comparison of different searching algorithms

The City Block distance, Normalised Inner Product and standardised Euclidean distance were compared against the Euclidean distance. The built-in function *pdist2* in MATLAB allows these distances to be computed easily. Recall that the Normalised Inner Product is the cosine value of the angle between two sets of invariants if they are treated as two vectors. This MATLAB function does not calculate the Normalised Inner Product directly. The final output value is one minus the Normalised Inner Product. Therefore if two invariants are similar, the angle between them will approach zero and the cosine value will approach one. The final output, after the subtraction from one, will approach zero.

It was found that if the logo extractions were executed perfectly (the whole logo was present in the image without any parts from the tablet surface), any distance algorithm could, in general, produce the ideal search results. For example, searching using the “kangaroo” logo as the query (Figure 3.19), all the twelve perfect logos were ranked as the top hits no matter which distance calculations were used. Only the order which these twelve logos appeared was different but this was not important. The last “kangaroo” logo image, which contained parts of the tablet surface (not perfect), was ranked low on the list. The extraction of “Euro” logos, the “e” logos and the two similar “three-pointed star” logos was also executed perfectly. All four “Euro” and “e” logos in

the database were ranked as the top four by all of the search algorithms when their respective fifth logo image was used as the query.

However, the Euclidean distance and the City Block distance did have an advantage over the other algorithms. Although the “three-pointed star” logos were extracted perfectly, the search for them in the database using the Normalised Inner Product failed to even rank all four logos identical to the query logo in the top 11. The other five very similar logos were missing from the top 11 hits as well. Therefore the Normalised Inner Product was inferior in this sample search. The Euclidean distance and the City Block distance performed better than the standardised Euclidean distance because in searching the “e” logos, they could rank the “three-pointed star” logos after the positive hits and vice versa (Compare Figure 3.21 and Figure 3.22). When the “three-pointed star” logos were searched through the database, the “e” logos followed after the positive hits. These “three-pointed star” logos are, in a sense, similar to the “e” logos in that both are round and have narrow strokes. They have empty pixels inside the logo and have a stroke across the centre. It was a good outcome for the two distance algorithms to be able to rank similar logos high on the list but below the positive hits.

Figure 3.21: Logo searching using Euclidean distance.

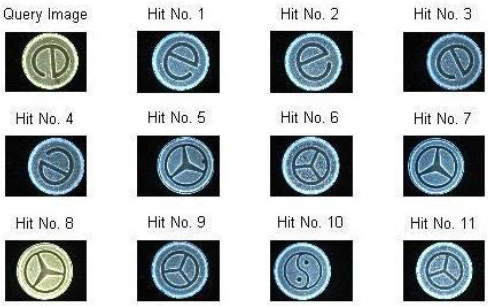


Figure 3.22: Same logo searching as in Figure 3.21 but with standardised Euclidean distance used instead.

For logos that were not extracted perfectly, there were a lot of false positives (logos that were ranked higher than the actual matching logos in the list) in the search results. A comparison of the performance of searching algorithms was based on the rankings of the matching logos and the number of these appearing in the first 11 hits. Table 3.2 summarises the rankings of the matching logos within the first 11 when imperfectly extracted logos were searched. The Precision (Pr) and Recall (Re) values as described in Section 1.5 (with K = 11) were also calculated as a reference.

Table 3.2: Rankings of the matching logos within the first 11 using complex moment invariants by different algorithms.

Logo (number of matching logos, N_t)	Euclidean distance	City Block distance	Standardi-sed Euclidean distance	Normalised Inner Product	Chebychev distance	Mahala-nobis distance
"Yin and Yang" (5)	1^{st}, 2^{nd}, 3^{rd}, 6^{th} Pr: 0.36 Re: 0.80	1^{st}, 2^{nd}, 3^{rd} Pr: 0.27 Re: 0.60	1^{st}, 2^{nd}, 4^{th} Pr: 0.27 Re: 0.60	5^{th} Pr: 0.09 Re: 0.20	1^{st}, 2^{nd}, 3^{rd}, 9^{th} Pr: 0.36 Re: 0.80	1^{st}, 2^{nd} Pr: 0.18 Re: 0.40
"8-ball" (3)	1^{st}, 2^{nd}, 8^{th} Pr: 0.27 Re: 1.00	1^{st}, 2^{nd}, 5^{th} Pr: 0.27 Re: 1.00	1^{st}, 2^{nd} Pr: 0.18 Re: 0.67	1^{st}, 2^{nd} Pr: 0.18 Re: 0.67	1^{st}, 2^{nd}, 7^{th} Pr: 0.27 Re: 1.00	1^{st}, 2^{nd} Pr: 0.18 Re: 0.67
"Apple" (11)	$1^{st} – 5^{th}$, $7^{th} – 10^{th}$ Pr: 0.82 Re: 0.82	$1^{st} – 5^{th}$, $7^{th} – 8^{th}$ Pr: 0.64 Re: 0.64	$1^{st} – 5^{th}$, $7^{th} – 8^{th}$ Pr: 0.64 Re: 0.64	$1^{st} – 4^{th}$ Pr: 0.36 Re: 0.36	$1^{st} – 5^{th}$, $7^{th} – 10^{th}$ Pr: 0.82 Re: 0.82	$1^{st} – 5^{th}$, 10^{th} Pr: 0.55 Re: 0.55

As concluded for the searches of "perfect" logos, the Normalised Inner Product and the Standardised Euclidean distance did not perform as well as the Euclidean distance and the City Block distance. There were fewer positive hits in the first 11 and those that were present were ranked lower. Out of the Euclidean distance and the City Block distance, the former may be the better choice because it included more positive hits when searching "Yin and Yang" logos and the "Apple" logo, at the small expense of having a lower ranking hit when the "8-ball" logo was searched. Therefore the Euclidean distance would be the method of choice for ranking complex moment invariants.

The built-in MATLAB function *pdist2* contains other choices for distance calculations. One of these is the other important Minkowski-type distance calculation, the Chebychev distance. A search using the Chebychev distance returned very similar results to those using the Euclidean distance with the same query logo images. While the Chebychev distance provided poorer rankings for searching "Yin and Yang" logos (the last positive hit at 9^{th} instead of 6^{th}), it ranked the positive hits of the "8-ball" logo slightly higher (the last positive hit at 7^{th} instead of 8^{th}). All other logo searching returned the same rankings.

The Mahalanobis distance gave worse rankings for positive hits than the Euclidean distance in general. The reason was that this measure is best performed when the vector components are dependent on each other (19). However, the components of complex moment invariants derived by Flusser et al. are independent of each other (60). Ranking logo images by the correlation between two invariant vectors was unsuccessful as well, based on the relative order of the true positives. None of these algorithms was good enough to replace the Euclidean distance as the preferred method of searching complex moment invariants.

3.4.2 Zernike Moment Invariants

Like the complex moment invariants, the third-order Zernike moment invariants consist of a row of six numbers. The MATLAB script for calculating Zernike moments by Flusser et al., *zermi.m*, provides two ways to perform these calculations (60). One way is using the efficient Kintner method as recommended by the authors (60). This method is more stable, quicker to implement and produces values in a narrow interval, thus avoiding the dynamic range problem for the invariants. The other one is by direct calculation from central moments normalised to scaling. This was carried out nonetheless because the highest order of invariants to be calculated was only three. It could serve as a comparison between the two calculation methods.

The script also provides two ways of constructing rotation-independent invariants from the Zernike moments. One is the normalisation approach and the other one is phase cancellation by multiplication. Although the demonstration by Flusser et al. showed that the former approach was slightly superior to the latter (60), both methods were used for comparison. The threshold for choosing the appropriate nonzero normalising moment for generating rotational invariants in both methods was 0.1.

3.4.2.1 Comparison between the computation methods using Euclidean distance

With two ways of calculating the Zernike moments and two methods for constructing invariants, there are four sets of Zernike moment invariants for comparison. This led to two general impressions. It was surprising to see that the dynamic range of these four sets was quite large but nonetheless smaller than the range seen in un-normalised complex moment invariants. Moreover, even with the large dynamic range, the

Euclidean distance surprisingly ranked far fewer false positives (other logos ranked higher than the positive hits) into the first 11 hit using any of the four sets of Zernike invariants (e.g. Figure 3.23) than using complex moment invariants (refer to the relevant cell in Table 3.2).

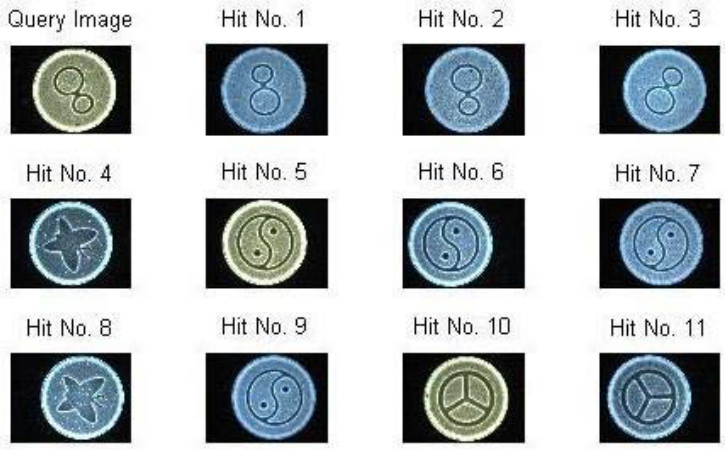


Figure 3.23: An example of a search for "8-ball" logos by Euclidean distance using Zernike moment invariants.

As in the previous section, the comparison of the performance of those four sets of Zernike invariants was based on the rankings of the positive hits and the number of the positive hits appearing in the first 11 hits. Table 3.3 summarises the rankings of the positive hits within the first 11, along with the respective Precision (Pr) and Recall (Re) values (with K = 11), when the 81 database invariants were ranked by Euclidean distance measurements relative to the query. Note that the two "three-pointed star" logos were treated as one group due to their high similarity. A query using one of these logos would return the four identical logo entries and five of the other similar logo entries in the database. Therefore the number of positive hits was 4+5.

Table 3.3: Rankings of the positive hits within the first 11 using different Zernike moment invariants searched by Euclidean distance.

Logo (number of positive hits, N_t)	Kintner method w/ normalisation	Kintner method w/ phase cancellation	Central moments w/ normalisation	Central moments w/ phase cancellation
"Yin and Yang" (5)	$1^{st} – 4^{th}$ Pr: 0.36 Re: 0.80	1^{st}, 2^{nd}, 3^{rd}, 9^{th} Pr: 0.36 Re: 0.80	1^{st}, 2^{nd}, 3^{rd}, 5^{th} Pr: 0.36 Re: 0.80	$1^{st} – 3^{rd}$ Pr: 0.27 Re: 0.60
"8-ball" (3)	$1^{st} – 3^{rd}$ Pr: 0.27 Re: 1.00	1^{st}, 4^{th} Pr: 0.18 Re: 0.67	$1^{st} – 3^{rd}$ Pr: 0.27 Re: 0.60	1^{st}, 2^{nd} Pr: 0.18 Re: 0.67
"Three-pointed Star -1, 2" (4 + 5)	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	1^{st}, 3^{rd}, 4^{th} Pr: 0.27 Re: 0.33	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00
	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 4^{th}$, 6^{th}, 9^{th} Pr: 0.55 Re: 0.67	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00
"Euro" (4)	$1^{st} – 4^{th}$ Pr: 0.36 Re: 1.00	$1^{st} – 4^{th}$ Pr: 0.36 Re: 1.00	$1^{st} – 4^{th}$ Pr: 0.36 Re: 1.00	$1^{st} – 4^{th}$ Pr: 0.36 Re: 1.00
"e" (4)	$1^{st} – 4^{th}$ Pr: 0.36 Re: 1.00	$1^{st} – 4^{th}$ Pr: 0.36 Re: 1.00	$1^{st} – 4^{th}$ Pr: 0.36 Re: 1.00	1^{st}, 3^{rd}, 4^{th}, 6^{th} Pr: 0.36 Re: 1.00
"Apple" (11)	$1^{st} – 5^{th}$, $7^{th} – 10^{th}$ Pr: 0.82 Re: 0.82	$1^{st} – 8^{th}$ Pr: 0.73 Re: 0.73	$1^{st} – 5^{th}$, $7^{th} – 10^{th}$ Pr: 0.82 Re: 0.82	$1^{st} – 5^{th}$, 7^{th}, 8^{th} Pr: 0.64 Re: 0.64
"Kangaroo" (13)	$1^{st} – 11^{th}$ Pr: 1.00 Re: 0.85	$1^{st} – 11^{th}$ Pr: 1.00 Re: 0.85	$1^{st} – 11^{th}$ Pr: 1.00 Re: 0.85	$1^{st} – 11^{th}$ Pr: 1.00 Re: 0.85

Out of these four sets of results, it was obvious that the result using the Kintner method with normalisation to calculate Zernike moment invariants was the best. The rankings

of the positive hits were the highest for their respective queries. The number of the positive hits appearing in the top 11 out of the maximum possible was the greatest as well. Therefore this procedure should be the method of choice for calculating Zernike moment invariants of logo images.

Using either of the Zernike moment calculation methods, the normalisation approach for constructing the rotational invariants performed better than the phase cancellation by multiplication method. It showed its superiority with the Kintner method. At the same time, the Zernike moments that were converted from scale-normalised central moments generated better rotational invariants with the normalisation approach than with phase cancellation by multiplication. This was in agreement with the observations by Flusser et al. in their demonstration (60).

The comparison between the two Zernike moment computation methods was not as clear cut. As described above, the Kintner method was better when the normalisation approach was followed for constructing the rotational invariants. However, mixed results were observed when phase cancellation by multiplication was used. The Kintner method produced better search results for the “Yin and Yang” logos, “e” logos and “apple” logos. The other method ranked the “8-ball” logos and the two “three-pointed star” logos more accurately. As the invariants were only third order here, most likely the order had to be higher before the invariants calculated by central moments would become more unstable and occupy a greater dynamic range and thus adversely affect the searching performance.

3.4.2.2 Comparison of different searching algorithms

The City Block distance, Normalised Inner Product and standardised Euclidean distance were again compared against the Euclidean distance. The Zernike moment invariants were calculated using the Kintner method with normalisation approach.

Near identical conclusions could be drawn in this comparison as in the searching algorithm comparison using complex moment invariants. Once more, all of the distance algorithms produced ideal search results if the logo extractions were executed perfectly (the whole logo was present in the image without any parts from the tablet surface). The perfect “kangaroo” logos were ranked top when one of them was used as the query.

The same thing happened when a “Euro” logo or an “e” logo was used as the query. All searching algorithms would rank the remaining four logos in the database as the top four.

As in the case of complex moment invariants, the Euclidean distance and the City Block distance performed better than the other algorithms with the Zernike moment invariants. The search for “three-pointed star” logos in the database using the Normalised Inner Product again failed to even rank all four logos identical to the query logo or the other five very similar logos in the top 11. The standardised Euclidean distance was inferior for the same reason. In a search for the “e” logos, not all of the “three-pointed star” logos were ranked immediately after the positive hits (Figure 3.24). Two “apple” logos were mixed within the “three-pointed star” logo rankings. The other two algorithms, however, ranked the “three-pointed star” logos together.

Figure 3.24: Searching "e" logos using Zernike moment invariants and the Standardised Euclidean distance.

The remaining logo searching results are summarised in the table below. The comparison of the performance of searching algorithms was also based on the rankings of the positive hits and the number of the positive hits appearing in the first 11 hits. The Precision (Pr) and Recall (Re) values (with K = 11) are also included.

Table 3.4: Rankings of the positive hits within the first 11 using Zernike moment invariants with different algorithms.

Logo (number of positive hits, N_t)	Euclidean distance	City Block distance	Standardi-sed Euclidean distance	Normalised Inner Product	Chebychev distance	Mahala-nobis distance
"Yin and Yang" (5)	$1^{st} – 4^{th}$ Pr: 0.36 Re: 0.80	$1^{st} – 4^{th}$ Pr: 0.36 Re: 0.80	$1^{st} – 4^{th}$ Pr: 0.36 Re: 0.80	$1^{st} – 3^{rd}$, 6^{th} Pr: 0.36 Re: 0.80	$1^{st} – 3^{rd}$, 9^{th} Pr: 0.36 Re: 0.80	$1^{st} – 4^{th}$, 10^{th} Pr: 0.45 Re: 1.00
"8-ball" (3)	$1^{st} – 3^{rd}$ Pr: 0.27 Re: 1.00	$1^{st} – 3^{rd}$ Pr: 0.27 Re: 1.00	$1^{st} – 3^{rd}$ Pr: 0.27 Re: 1.00	$1^{st} – 3^{rd}$ Pr: 0.27 Re: 1.00	$1^{st} – 3^{rd}$ Pr: 0.27 Re: 1.00	$1^{st} – 3^{rd}$ Pr: 0.27 Re: 1.00
"Apple" (11)	$1^{st} – 5^{th}$, $7^{th} – 10^{th}$ Pr: 0.82 Re: 0.82	$1^{st} – 5^{th}$, $7^{th} – 9^{th}$ Pr: 0.73 Re: 0.73	$1^{st} – 6^{th}$, 8^{th}, 9^{th} Pr: 0.73 Re: 0.73	$1^{st} – 5^{th}$ Pr: 0.45 Re: 0.45	$1^{st} – 5^{th}$, $7^{th} – 10^{th}$ Pr: 0.82 Re: 0.82	$1^{st} – 5^{th}$, $7^{th} – 9^{th}$ Pr: 0.73 Re: 0.73

As in the case of complex moment invariants, the Normalised Inner Product performed the worst logo searching again than the other algorithms. Hence this algorithm was the least favourable. The Standardised Euclidean distance performed slightly better than the City Block distance by ranking an "apple" logo one rank higher. Combining the searching results for the other logos, the abilities of the two searching algorithms were comparable. The Euclidean distance was obviously the better choice here as it gave the best search results for all logos. Therefore the Euclidean distance would also be the method of choice for ranking Zernike moment invariants.

As with the complex moment invariants, the Chebychev distance returned results that were very similar but inferior to the Euclidean distance searches. It gave poorer rankings for searching "Yin and Yang" logos (the last positive hit at 9^{th} instead of 4^{th}) and placed a false positive in the first 11 hits when a "three-pointed star" logo was searched. All other logo searching returned the same rankings. The Mahalanobis distance and the correlation were also unsuitable for this task as well. The Euclidean distance was still the best choice.

3.4.3 Complex vs Zernike Moment Invariants

Both types of moment invariants ranked database logos better using the Euclidean distance, so the two sets of search results using the said algorithm were compared. The table below summarises the rankings reported in the previous sections (Table 3.2 and Table 3.4).

Table 3.5: Rankings of logos searched by Euclidean distance using different moment invariants.

Logo (number of positive hits, N_t)	Complex Moment Invariants	Zernike Moment Invariants
"Yin and Yang" (5)	1^{st}, 2^{nd}, 3^{rd}, 6^{th} Pr: 0.36 Re: 0.80	$1^{st} – 4^{th}$ Pr: 0.36 Re: 0.80
"8-ball" (3)	1^{st}, 2^{nd}, 8^{th} Pr: 0.27 Re: 1.00	$1^{st} – 3^{rd}$ Pr: 0.27 Re: 1.00
"Apple" (11)	$1^{st} – 5^{th}$, $7^{th} – 10^{th}$ Pr: 0.82 Re: 0.82	$1^{st} – 5^{th}$, $7^{th} – 10^{th}$ Pr: 0.82 Re: 0.82

The other logos, as discussed in the relevant sections, were all ranked as the top hits when a matching logo was searched as the query. These results were therefore not included in this table. The Zernike moment invariants definitely had the edge in logo searching over the complex moment invariants. Although the number of positive hits appearing in the first 11 was the same for both invariants, the actual rankings achieved by the Zernike moment invariants were better.

3.5 Summary

By taking a new approach in imaging the tablets, specifically by lowering the light source closer to the tablet, better quality logos can easily be segmented from the photos. A new database was set up using these high contrast photos to generate binarised logo images. Based on this database of photos of freshly-prepared tablets, calculating Zernike moment invariants with the Kintner method, followed by the normalisation approach,

and then ranking these invariants using the Euclidean distance gave the best chance of finding logos in the database that were identical to the query. Ideally all possible combinations of moment invariants, normalisation methods and searching algorithms would be tested. Although it was not possible to achieve this within the time constraint of this project, the above protocol is considered to be a reasonable and effective compromise.

Chapter 4: SEARCHING ILLICIT DRUG TABLET IMAGES USING COLOUR

Chapter 4: Searching illicit drug tablet images using colour

In the previous chapter, the idea of lowering the light source close to the tablet was proposed in order to maximise the chances of extracting complete logo silhouettes. However, as these images do not truly represent how examiners perceive the tablet under normal lighting conditions, another set of images under these "normal" conditions is necessary. Apart from being a reference for operators, these images can also be used for extracting other features such as colour (Figure 4.1).

Figure 4.1: An example of a pair of images of the same tablet under different lighting conditions. The left-handed image is used for logo extraction, while the right hand image is used to extract other features.

Apart from the logo, another important feature of a tablet is colour. Colour extraction from these images is simple due to the common black background, which allows for easy segmentation. In addition, tablets usually contain only one colour and therefore how different colours are spatially distributed within the tablet does not need to be taken into consideration. Furthermore, as these photos are taken under normal lighting conditions, the logo contrast is low, so the colour is quite consistent across each tablet and can be represented by the average of colour values in the tablet. As a result, colours can be treated as points in a vector space and normal distance measurements are enough for sorting similar colours from dissimilar ones.

Searching by colour is made relatively simple in this project because all photos will be taken using the same hardware and software under strict instructions and guidelines that are outlined in the Australian Federal Police (AFP) standard operating procedures. Histogram equalisation and other normalisation techniques used to remove variations in lighting conditions will therefore not be necessary.

This chapter describes efforts to determine best colour comparison method for the new images collected under the consistent lighting conditions. Different colour space and distance measurement combinations are tested, using both the new images and the existing AFP database photos. Finally, the possibility of searching tablets using both logo and colour is discussed.

4.1 Method

A new set of photos of the new, custom-pressed tablets (refer to Section 3.1) was taken using the standard ring light positions using a Leica DFC295 camera mounted on the Leica MZ6 Macroscope and processed using the same standard UTS-issued laptop with MABTLAB described in Section 2.1. In a database containing 60 tablet photos, there were ten photos for each of six different colours. The six colours were Brilliant Blue, Green, Tartrazine Yellow, Amaranth Purple, Sunset Yellow and Ponceau Red (as shown, respectively, in Figure 4.2). A query image was randomly chosen from the set to search against the remaining 59 images. With these ideal photos (carefully taken by the same operator to avoid any illumination problems and variations), the aim is to produce searching results which rank identical colours first, followed by colours that are perceptually similar before any others, by comparing their coordinates in different colour spaces using the Euclidean distance, standardised Euclidean distance, City Block distance and Normalised Inner Product.

Figure 4.2: Sample colours of the new tablets we pressed.

A subset of the existing AFP photos was also chosen for testing. In this subset, the photos are picked based on their illumination conditions. Some of the AFP photos were taken

with very poor illuminations, such that strong reflections are seen in the background or the tablet surface is not evenly lit. These images were rejected for the purposes of this testing, leaving 63 images in this subset of existing AFP images. Some of the sample photos are shown in Figure 4.3. Again, a query photo was randomly chosen within the group and searched against the remaining 62. The main objective here was to test how well the best performed algorithm for the ideal photos works with the existing AFP tablet photos.

Figure 4.3: Sample photos chosen within the AFP database.

From the literature (refer to Section 1.3.2), it is well-known that comparing colours in RGB (Red, Green, Blue) colour space is difficult because the RGB colour space is not perceptually uniform. This colour space was therefore left out in the tests and RGB values were not used for direct comparisons. Other colour spaces were tested instead. Attention was focussed on the HSV (Hue, Saturation, Value) and the CIE L*a*b* colour spaces as they could be readily converted by built-in MATLAB scripts.

4.2 Colour searching in the HSV colour space

The conversion of RGB values in tablet images to their HSV counterparts was simply achieved by applying the built-in MATLAB function *rgb2hsv*. Distance measurements were also computed to return sorted distance results using the built-in MATLAB function *pdist2*.

4.2.1 New photos

Good searching results were immediately obtained when HSV values were compared using the Euclidean distance. When a Brilliant Blue, Green, Amaranth Purple or Ponceau Red-coloured tablet was searched, all remaining nine correct hits were ranked at the top of the list (e.g. Figure 4.4). Only eight (out of nine) positive hits were ranked in the top 11 hits when a Tartrazine yellow tablet was used as the query (Figure 4.5). Some false positives (tablets that did not have the same colour as the query) were ranked before the correct ones in the search for Sunset Yellow tablets, where only seven out of nine correct hits appeared in the top 11 results (Figure 4.6).

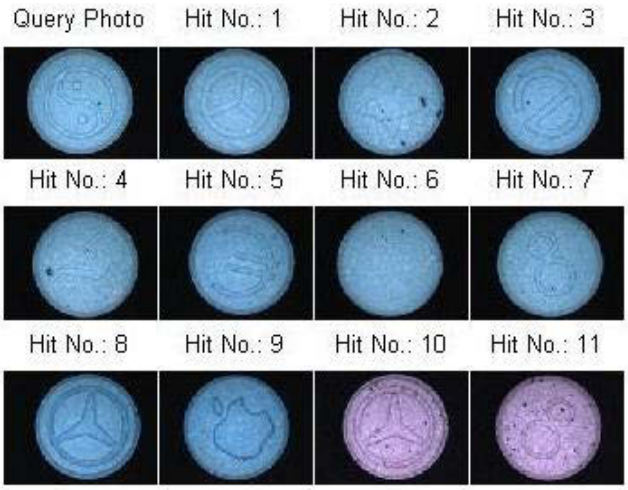


Figure 4.4: An example of all nine positive hits ranking top of the list using HSV values.

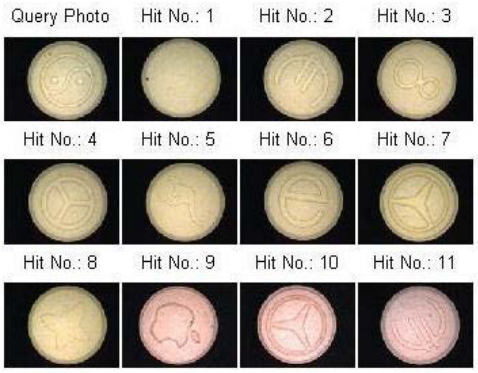


Figure 4.5: Searching Tartrazine Yellow tablets using Euclidean distance.

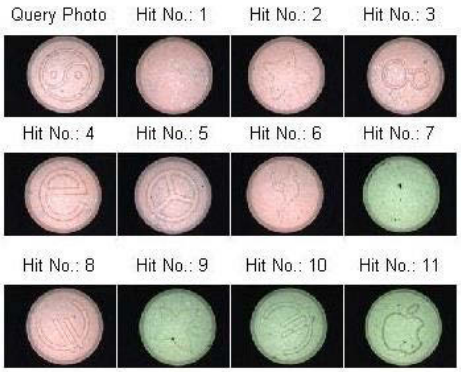


Figure 4.6: Searching Sunset Yellow tablets using Euclidean distance.

When the colour values were ranked using the *standardised* Euclidean distance, similar results were obtained. The searching algorithm was able to differentiate Brilliant Blue,

Green, Amaranth Purple and Ponceau Red from the others and it correctly ranked these tablets as the top nine hits, when the tenth tablet was used as the query image. The search result for Tartrazine Yellow was not quite so successful. It ranked a false positive at the eighth position, which was higher than one of the correct hits at ninth (Compare Figure 4.5 and Figure 4.7). However, the search for Sunset Yellow was slightly better, with all of the positives appearing in the top 11 (Compare Figure 4.6 and Figure 4.8).

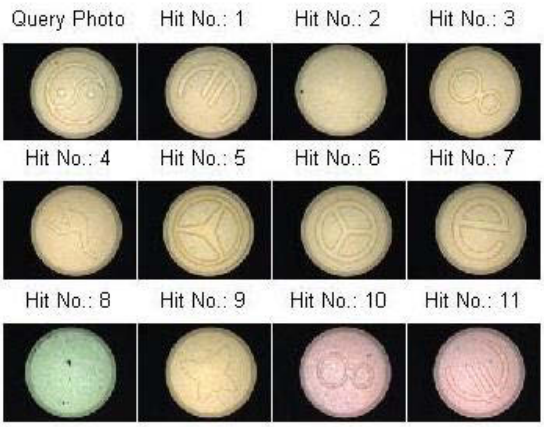


Figure 4.7: Searching Tartrazine Yellow tablets using Standardised Euclidean distance.

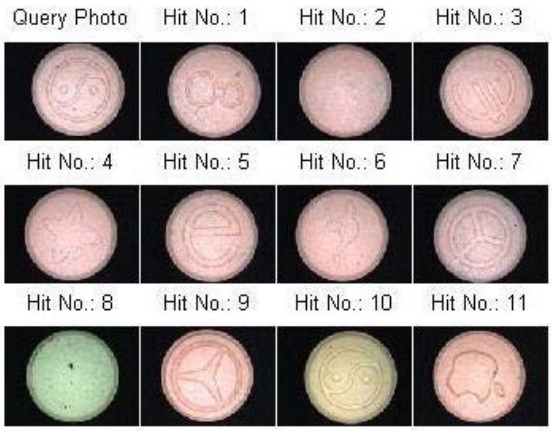


Figure 4.8: Searching Sunset Yellow tablets using Standardised Euclidean distance.

The City Block distance achieved the best performance out of the four distance measurements. It produced slightly better results than the Euclidean distance. This searching algorithm, like the Euclidean distance, was also able to rank the nine Brilliant Blue, Green, Amaranth Purple and Ponceau Red-coloured tablets as the top nine hits (with the tenth as the query). Although some “false positives” were identified in the search for Tartrazine Yellow tablets, nine correct hits made it into the top 11 hits (Figure 4.9) as opposed to only eight when the Euclidean distance was used (Figure 4.5). The Sunset Yellow tablets were ranked slightly more accurately using the City Block distance as well (Compare Figure 4.6 and Figure 4.10).

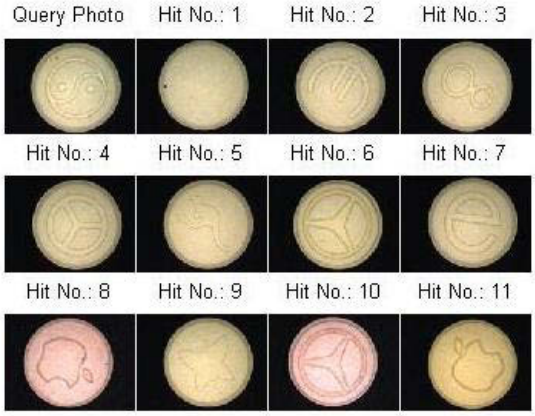


Figure 4.9: Searching Tartrazine Yellow tablets using City Block distance.

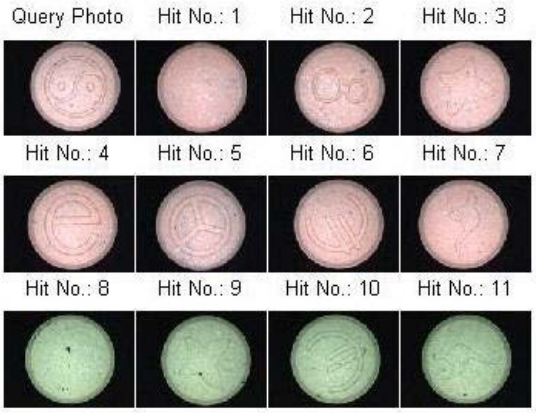


Figure 4.10: Searching Sunset Yellow tablets using City Block distance.

The Normalised Inner Product (which calculates one minus the cosine value of the angle between any two colour points being treated as vectors) performed the worst out of the four algorithms. It could rank all nine positives as the top hits for Brilliant Blue, Amaranth Purple and Ponceau Red but (in contrast to the other algorithms tested) not for Green (Figure 4.11). All searches for other colours included false positives.

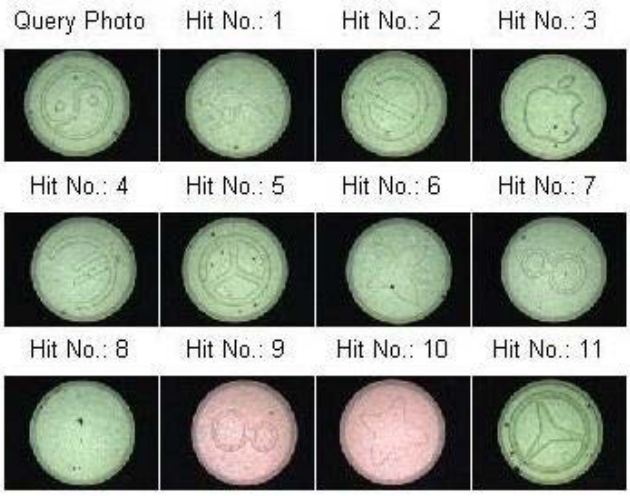


Figure 4.11: Searching Green tablets using Normalised Inner Product.

This led to the realisation that there could be a flaw with the idea of using any of the above distance measurements. The Hue in the HSV colour space represents the angular difference between the colour in question and red (0°). Trigonometric operations may be therefore necessary to compare the Hue values. The algorithm by Jeong et al. was therefore tested after some research into which output in the MATLAB built-in function *rgb2hsv* represented the Hue angle (99). It was established that the angle is converted

to a ratio relative to 360° or 2π, instead of being output as an angle in degrees or radians (i.e. 0.5 instead of 180° or π).

The new algorithm gave mixed results compared with using the City Block distance. Both algorithms could rank the nine Brilliant Blue, Green, Amaranth Purple or Ponceau Red-coloured tablets top positions when the corresponding tenth tablet were used as a query image. This algorithm ranked the nine Tartrazine Yellow tablets at the 1st to 8th positions and at the 11th position (Figure 4.12), performing better for this colour than the City Block distance (1st – 7th, 9th and 11th; Figure 4.9). However, it could only place seven Sunset Yellow tablets out of the possible nine positive hits at the 1st to 5th, 7th and 8th positions as shown in Figure 4.13 (the City Block distance correctly ranked seven tablets in the top 11 also but at the 1st to 7th positions in the same query; Figure 4.10).

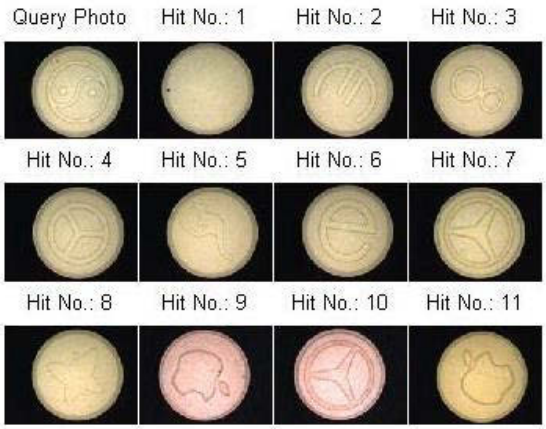


Figure 4.12: Searching Tartrazine Yellow tablets using the algorithm from Jeong et al. (99).

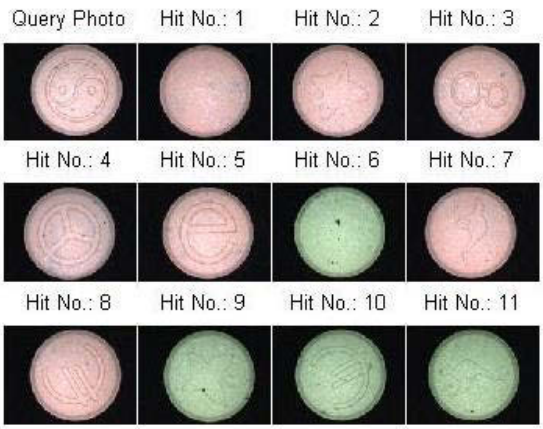


Figure 4.13: Searching Sunset Yellow tablets using the algorithm from Jeong et al. (99).

A summary of the rankings achieved by different algorithms as discussed in this section is shown in Table 4.1 below, along with the Precision (Pr) and Recall (Re) values (using K = 11).

Table 4.1: Summary of the rankings of the positive hits within the top 11 using HSV colour space with different search algorithms.

Colour (nine possible hits; $N_t = 9$)	Euclidean distance	Standardised Euclidean distance	City Block distance	Normalised Inner Product	Algorithm from Jeong et al. (99)
Brilliant Blue	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00
Green	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 8^{th}$, 11^{th} Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00
Tartrazine Yellow	$1^{st} – 8^{th}$ Pr: 0.73 Re: 0.89	$1^{st} – 7^{th}$, 9^{th} Pr: 0.73 Re: 0.89	$1^{st} – 7^{th}$, 9^{th}, 11^{th} Pr: 0.82 Re: 1.00	1^{st}, 2^{nd}, $4^{th} – 8^{th}$, 10^{th}, 11^{th} Pr: 0.82 Re: 1.00	$1^{st} – 8^{th}$, 11^{th} Pr: 0.82 Re: 1.00
Amaranth Purple	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00
Sunset Yellow	$1^{st} – 6^{th}$, 8^{th} Pr: 0.64 Re: 0.78	$1^{st} – 7^{th}$, 9^{th}, 11^{th} Pr: 0.82 Re: 1.00	$1^{st} – 7^{th}$ Pr: 0.64 Re: 0.78	1^{st}, 3^{rd}, 4^{th}, 10^{th}, 11^{th} Pr: 0.45 Re: 0.56	$1^{st} – 5^{th}$, 7^{th}, 8^{th} Pr: 0.64 Re: 0.78
Ponceau Red	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00

None of the algorithms had a clear advantage over the others. Given these results on the ideal images, none of the above distance measurements were expected to give favourable searching results on the existing AFP images but this was attempted anyway.

4.2.2 Existing AFP photos

Searches of the existing AFP images using HSV colour and Euclidean distance were less successful than the ideal image searches. Although there were examples where similarly coloured tablets were ranked at the top (Figure 4.14), the rankings of other searches

were quite inaccurate. Examples where similar colours were not ranked together in sequence were often seen (e.g. hits 2 and 6, 3 and 5 etc. in Figure 4.15).

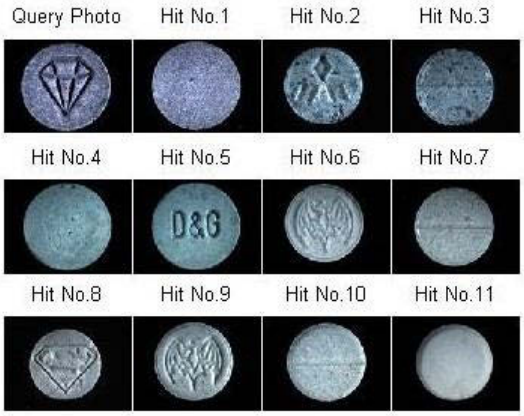


Figure 4.14: An example of colour searching in the HSV colour space using Euclidean distance.

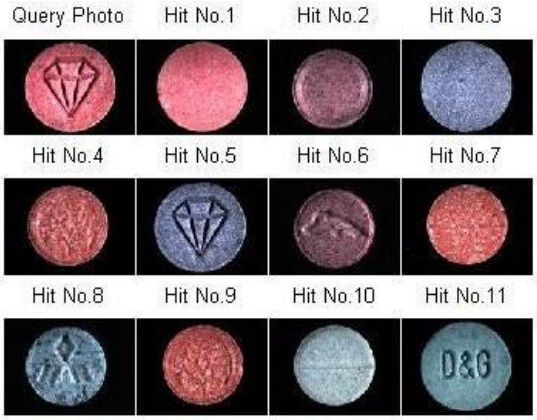


Figure 4.15: Another example of colour searching in HSV colour space using Euclidean distance.

Things did not improve when the HSV values were ranked using the standardised Euclidean distance, City Block distance or Normalised Inner Product. The distance measurements could not rank all of the tablets with identical colours consecutively at the top of the hit list. There were some slight improvements in the rankings of some of the searches, but most of the others were no better or were worse (e.g. Figure 4.14 vs. Figure 4.16 and Figure 4.15 vs. Figure 4.17). Differently coloured tablets were commonly seen in the first 11 hits of most searches.

Figure 4.16: An example of colour searching in the HSV colour space using Normalised Inner Product.

Figure 4.17: Another example of colour searching in HSV colour space using Normalised Inner Product.

Unfortunately, the algorithm by Jeong et al. did not improve this situation (99). There may have been some improvements over the other distance measurements but the

same problem occurred, and false positives were commonly observed (Figure 4.18 and Figure 4.19).

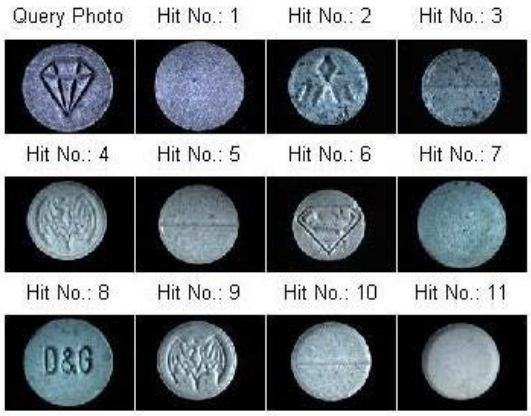


Figure 4.18: An example of colour searching in the HSV colour space using the algorithm from Jeong et al. (99).

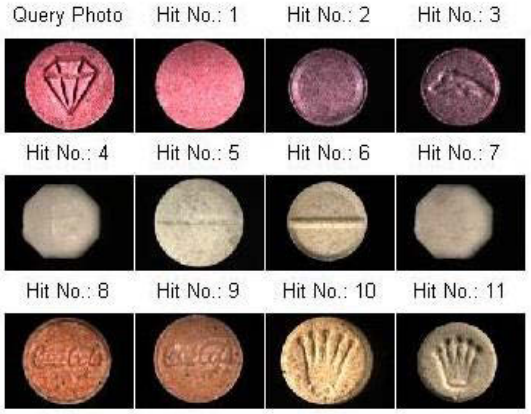


Figure 4.19: Another example of colour searching in HSV colour space using the algorithm from Jeong et al. (99).

These results led to the same conclusion reached in the previous section using the new, ideal images: none of these algorithms were suitable for ranking colour similarities of drug tablets in the HSV colour space.

4.3 Colour searching in the CIE L*a*b* colour space

The conversion of RGB values in tablet images to L*a*b* values was easily achieved by applying a colour transformation structure generated using the built-in MATLAB function *makecform*, although the computation time for this conversion was longer than to HSV values. Distances between two points in the colour space were again computed using the built-in MATLAB function *pdist2* and sorted to give search rankings.

4.3.1 New photos

Satisfactory results were immediately achieved for the ideal images using the Euclidean distance in the L*a*b* space. Improvements over the searches using the same distance algorithm with HSV values were obvious. All nine identical colours were ranked top hits when the corresponding tenth image was used as the query for each of the six colours (three are shown in Figure 4.20 – Figure 4.22). In addition, the colour of the tablets that

were ranked immediately after the nine positive hits was perceptually closer to the query colour in these searches. For example, when a Sunset Yellow tablet was searched (Figure 4.22), two Ponceau Red tablets were ranked after the nine positives instead of the Green ones identified using the HSV colour space (Figure 4.6).

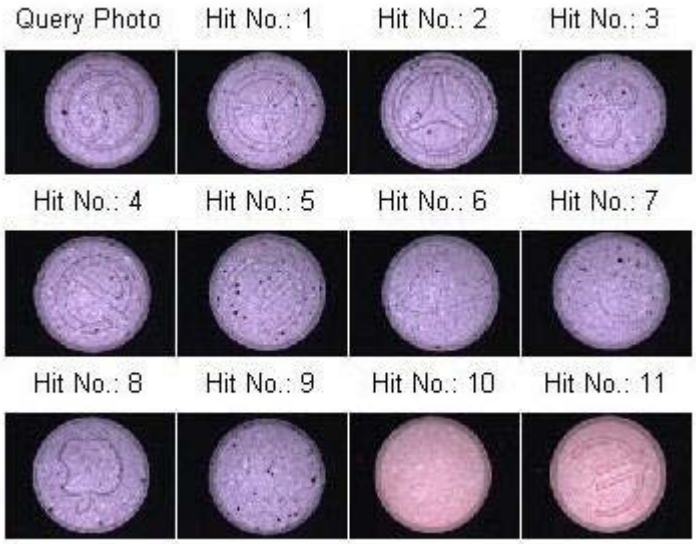


Figure 4.20: An example of all nine positive hits ranking at the top of the list using L*a*b* values.

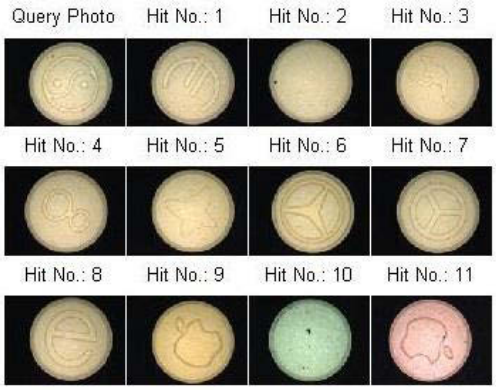


Figure 4.21: Searching Tartrazine Yellow tablets using Euclidean distance.

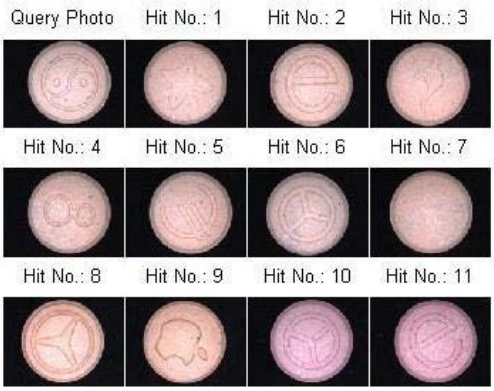


Figure 4.22: Searching Sunset Yellow tablets using Euclidean distance.

We noted that out of the six searches using each of the six colours as queries, only the search for Tartrazine Yellow tablets ranked tablets randomly after the nine positive hits (Figure 4.21). An investigation into this revealed that it was due to the nature of the colour values. The L*a*b* values compared here were uint8 integers. However, a* and b* values can be positive or negative, depending on the colour's position between red/magenta and green, and between yellow and blue respectively. The photos were therefore changed to double precision to accommodate the change from RGB

coordinates to their L*a*b* counterparts, before the searches for all six coloured tablets were repeated. Now the search results for Tartrazine Yellow tablets were better in that they ranked the green tablets immediately after the positive hits (Figure 4.23). The searches for other colours also ranked the nine positive hits followed by tablets with perceptually similar colours in the same way as before (e.g. compare Figure 4.22 and Figure 4.24).

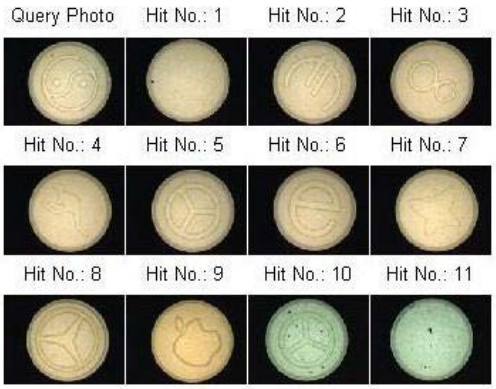


Figure 4.23: Searching Tartrazine Yellow tablets using Euclidean distance in double precision.

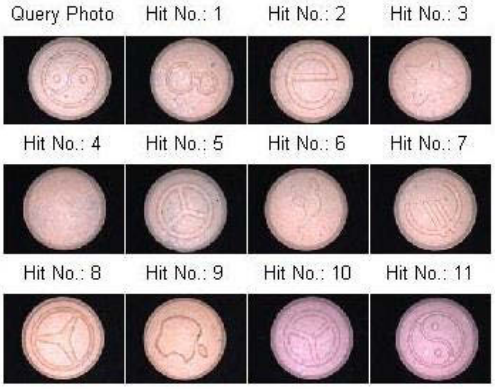


Figure 4.24: Searching Sunset Yellow tablets using Euclidean distance in double precision.

Ranking CIE L*a*b* colour values using the Standardised Euclidean distance was inferior to using the Euclidean distance. Although the algorithm could rank all the positive hits first for each query, some of the rankings afterwards seemed to be randomly ordered (Figure 4.25 – Figure 4.26).

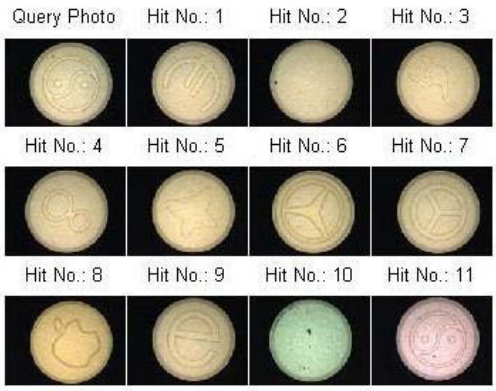


Figure 4.25: Searching Tartrazine Yellow tablets using Standardised Euclidean distance in double precision.

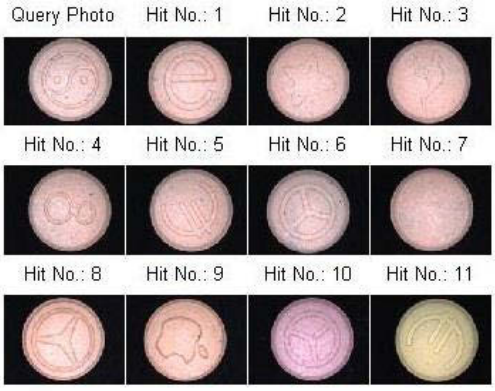


Figure 4.26: Searching Sunset Yellow tablets using Standardised Euclidean distance in double precision.

When the L*a*b* colour values were compared using the City Block distance or Normalised Inner Product, the outcomes were similar to those using the Euclidean distance. Both algorithms were able to rank all the positive hits first before similarly-

coloured tablets in each query (Figure 4.27 – Figure 4.30). Overall, they ranked colours in this colour space in the same way.

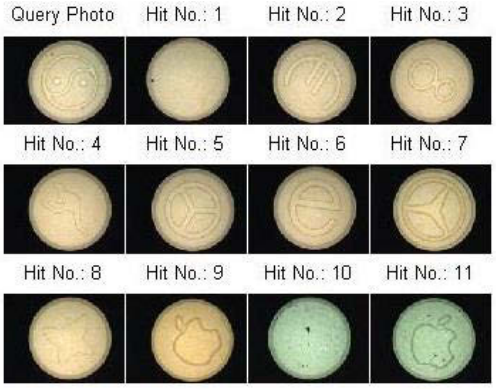


Figure 4.27: Searching Tartrazine Yellow tablets using City Block distance in double precision.

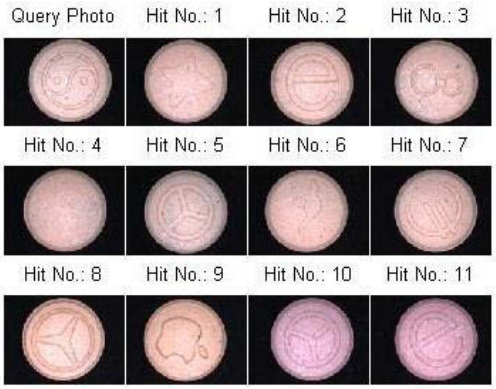


Figure 4.28: Searching Sunset Yellow tablets using City Block distance in double precision.

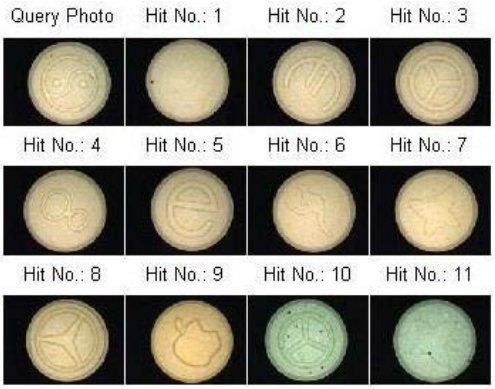


Figure 4.29: Searching Tartrazine Yellow tablets using Normalised Inner Product in double precision.

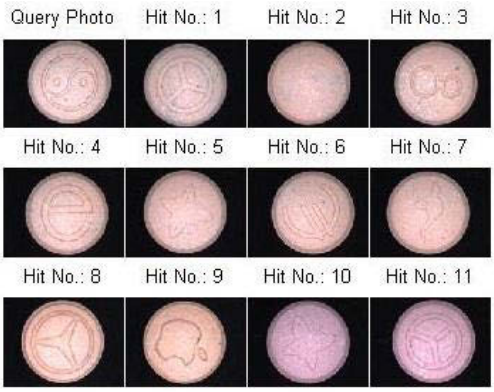


Figure 4.30: Searching Sunset Yellow tablets using Normalised Inner Product in double precision.

A summary of the rankings achieved by different algorithms, together with the respective Precision (Pr) and Recall (Re) values (using K = 11), is shown in Table 4.2 below.

Table 4.2: Summary of the rankings of the positive hits within the first 11 using L*a*b* colour space by different algorithms.

Colour (nine possible hits; N_t = 9)	Euclidean distance	Standardised Euclidean distance	City Block distance	Normalised Inner Product
Brilliant Blue	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00
Green	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00
Tartrazine Yellow	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00
Amaranth Purple	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00
Sunset Yellow	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00
Ponceau Red	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82 Re: 1.00

It is obvious that the CIE L*a*b* colour space is more suitable for the comparison and searching of tablet colours in the photo database we have in general. Amongst the four distance measurements tested here, all but the Standardised Euclidean distance would most likely give very satisfactory searching outcomes in the real applications because of their abilities to rank perceptually similar colours after the positive hits. These three measures were tested next using the existing AFP images.

4.3.2 Existing AFP photos

As for the new photos, searching the AFP photos improved in the L*a*b* colour space. Colours that were perceptually close were more likely to be ranked together, and it should be noted that this database did contain tablets with only subtly different colours. Using the Euclidean distance, some satisfactory results were achieved, although there was room to improve (Figure 4.31). In some cases, the colours of tablets that appeared after the positive hits were not quite perceptually similar to the query colour (Figure 4.32).

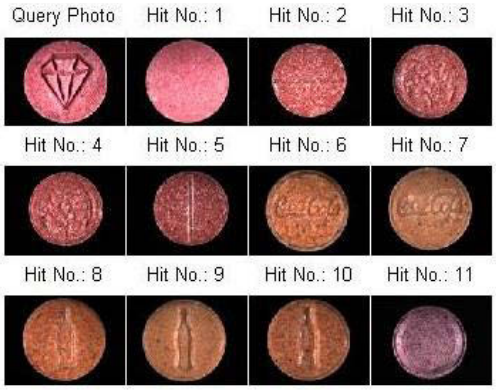


Figure 4.31: An example of colour searching in the L*a*b* colour space using Euclidean distance.

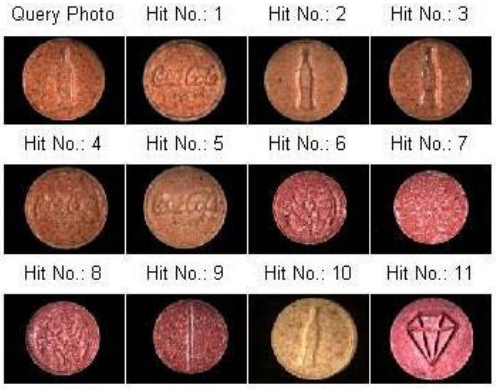


Figure 4.32: Another example of colour searching in the L*a*b* colour space using Euclidean distance.

When the City Block distances were computed to quantify the difference between the query colour and the database colours in L*a*b* values, the outcomes were slightly worse than using Euclidean distance. Results were better than when HSV values were used, but false positives were still seen.

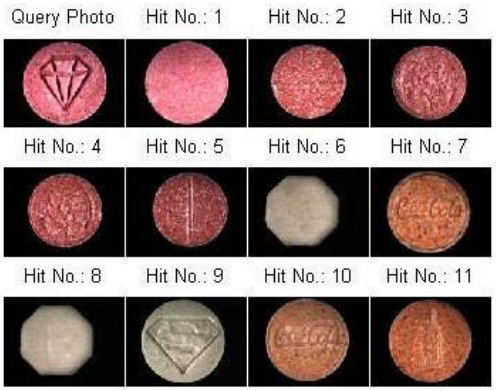


Figure 4.33: An example of colour searching in the L*a*b* colour space using City Block distance.

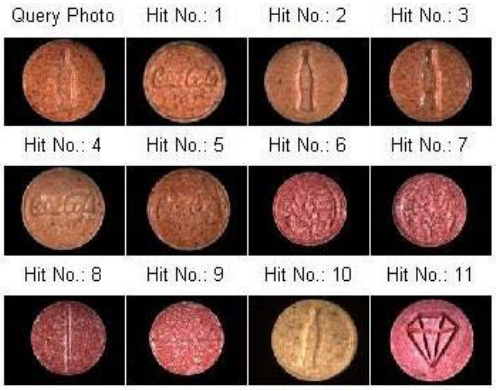


Figure 4.34: Another example of colour searching in the L*a*b* colour space using City Block distance.

Searching using the Normalised Inner Product produced the best rankings so far. Tablets with similar colours were ranked immediately after the positive hits. The major advantage of this algorithm was that it seemed to be able to rank together pairs of photos that showed both sides of the same tablet (hence the images should have shown the same colour even though sometimes the lighting conditions obscured this similarity). Some examples are shown below. In Figure 4.35, starting with the query image and Hit No. 1, each consecutive pair of images is from the same tablet but different sides. Similar observations were made for the search depicted in Figure 4.36 (6^{th} – 11^{th} hits). However, it was not as perfect as hoped because there were some rankings that seemed different to what humans would perceive. For example, in Figure 4.37, it would be much better if the 4^{th} – 7^{th} positions were Hit Numbers 5 and 7, then 4 and 6. Another example was in Figure 4.38 where there was a false positive (5^{th} rank) with a very different colour than its "neighbours" in the list.

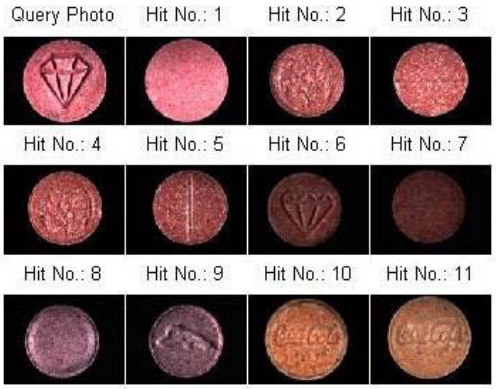


Figure 4.35: An example of colour searching in the L*a*b* colour space using Normalised Inner Product.

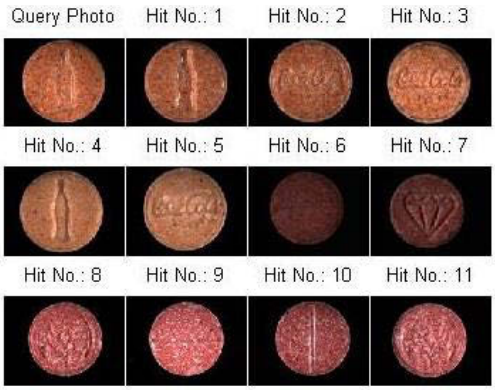


Figure 4.36: Another example of colour searching in the L*a*b* colour space using Normalised Inner Product.

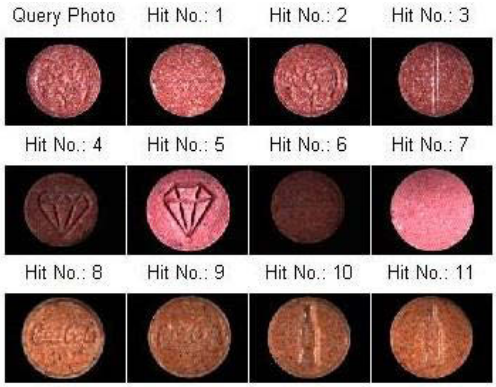


Figure 4.37: Third example of colour searching in the L*a*b* colour space using Normalised Inner Product.

Figure 4.38: Fourth example of colour searching in the L*a*b* colour space using Normalised Inner Product.

It was found that to maximise our chances of matching the query tablet colour, changing the RGB colour values of the photos to the CIE L*a*b* coordinates and then comparing them using the Normalised Inner Product was the best method.

The CIE L*u*v* colour space was not used in this project as the basis for searching images because the algorithm required to convert RGB colour values to the CIE L*u*v* counterparts was not readily available. Furthermore, the CIE L*a*b* search results were sufficiently accurate; the work on tablet colour searching was suspended during the project when the Australian Federal Police (AFP) indicated that an algorithm developed elsewhere would probably be used to search their databases.

4.4 A preliminary test for searching using both logo shapes and colours

Another set of photos of the new custom-pressed tablets was collected. In this set, a pair of photos was taken together for each tablet such that the logo and colour features extracted respectively could be traced back to the same tablet. Ninety-five pairs of images (entries) were included in this test set (see Table 4.3), which contained different combinations of some of the ten logos (as in previous chapter) and the six colours (Figure 4.2). The logos would be compared using the Zernike moment invariants and ranked using the Euclidean distance as discussed in the last chapter. The tablet colours would be compared in the CIE L*a*b* colour space using the Normalised Inner Product as above.

Table 4.3: A break-down of the number of tablet logo images in the database for logo and colour matching.

Logos/Markings	Colour of tablet	Number of photos
Yin and Yang	Green	5
	Tartrazine Yellow	5
Three-pointed star 1 (“Mercedes”)	Amaranth Purple	5
	Ponceau Red	5
Euro	Tartrazine Yellow	10
	Sunset Yellow	5
Kangaroo	Brilliant Blue	10
	Green	10
“e”	Ponceau Red	5
	Sunset Yellow	5
“Apple”	Amaranth Purple	20
	Brilliant Blue	10
	Total	95

The simplest way of combining both searches was to rank the sum of the “distances” from both the logo searching and the colour matching. Satisfactory results were achieved immediately. For example, when an Amaranth Purple “three-pointed star” tablet was searched against the other 94 entries (Figure 4.39), all of the other nine “three-pointed star” logos appeared in the top nine hits. This meant that all of the four tablets with the same colour and logo as the query were among those top nine. The same could be said for searching a Tartrazine yellow “Yin and Yang” tablet (Figure 4.40). All remaining nine “Yin and Yang” logos appeared at the top of the list. Again the four tablets having the same colour *and* logo were among the nine top hits.

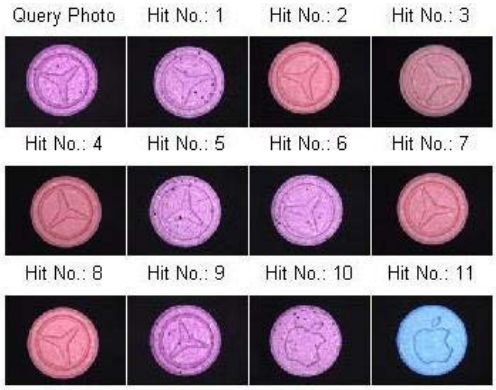


Figure 4.39: Searching using both logo and colour features.

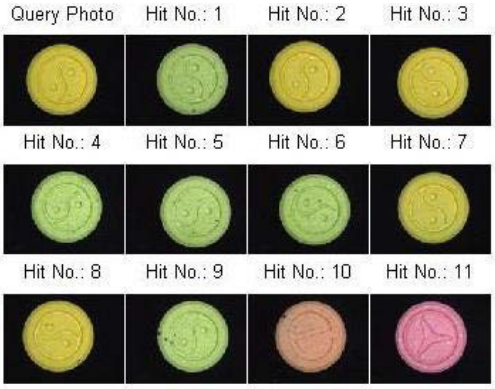


Figure 4.40: Another search using both logo and colour features.

However, it was noted that the colours perceptually similar to the query would not appear after the positive logo hits. Instead, those tablets seemed to be ranked randomly (e.g. Hit No. 11 in Figure 4.39 and Hit No. 10 and 11 in Figure 4.40). The main reason for this was that the logo "distances" were usually large numbers at magnitudes of 10^{-1} or above. If colours were similar, the distance between the vectors could be as low as 10^{-6}. The difference in magnitudes was very large, so the rankings shown here were dominantly influenced by the similarity of logos towards the query logo, especially when the colours were similar to that of the query.

One easy way to normalise the magnitudes was to divide all of the logo distances calculated using the Euclidean formula by their maximum value such that the distances were scaled between zero and one. As the Normalised Inner Product had the same range, neither feature distance would dominate. This led to some improvements in the searching results. Some cases demonstrated an ideal searching scenario. For example in Figure 4.41, all entries which had the same colour and logo were ranked first, before tablets with the same logo but different colours. In other cases, one or two tablets with the same logo but different colours ranked before some of the entries which had the exact match (e.g. Figure 4.42).

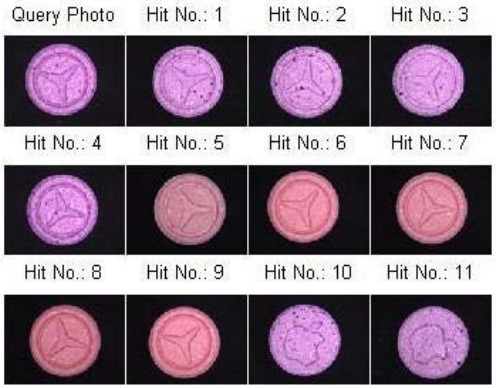


Figure 4.41: Searching using both logo (normalised between 0 and 1) and colour features.

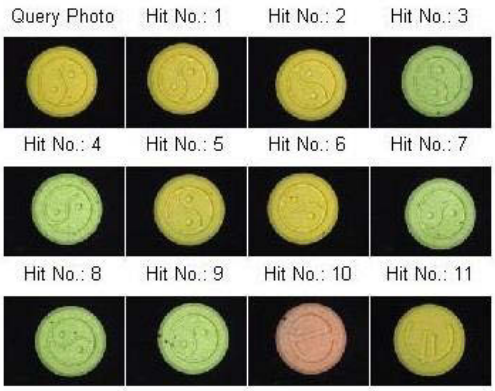


Figure 4.42: Another search using both logo (normalised between 0 and 1) and colour features.

The above normalisation step of dividing the Euclidean distances effectively put much more weight on colour matching than the initial test and hence ranked similar colours more favourably than before. The adjustment gave equal weighting to the logo and colour aspects of the search. This explained the 10th and 11th rankings in Figure 4.42. In this example, this "e" logo had an invariant vector that was very similar to that of the query (consistent with the findings in the previous chapter), such that despite its dissimilar colour, it was ranked right behind the "Yin and Yang" logos. For the next rank, the colour similarity of the 11th hit to the query prevailed.

Another way to normalise the magnitudes was to divide both the Euclidean distances for logo comparisons and the Normalised Inner Product for colour comparisons by their respective standard deviations. However, very similar results were obtained using this normalisation method. In fact, the query images that had been used in Figure 4.41 and Figure 4.42 produced the same rankings that are shown in those figures. However, for a third query image, the change in normalisation method did produce a different ordering of the search results (Figure 4.43 and Figure 4.44). Here, dividing both of the "distances" by their respective standard deviations seemed to improve the ranking because the positive hits (matching colours and logos) occurred in the first four positions (Figure 4.44). Normalisation of just logo Euclidean distances saw the same hits occur at first, second, third and fifth (Figure 4.43).

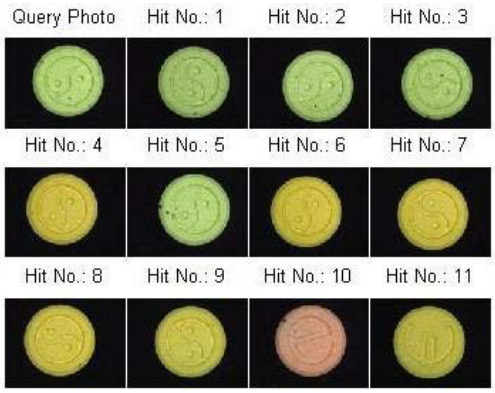


Figure 4.43: Searching using both logo (normalised between 0 and 1) and colour features.

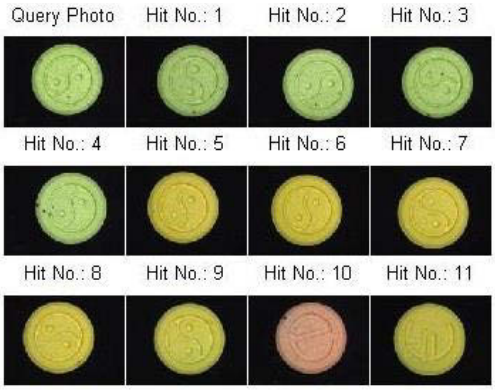


Figure 4.44: Same query search as Figure 4.43 using both logo and colour features normalised by their standard deviations.

All database tablets should be ideally ranked in the following sequence: tablets that had matching colours and logos as the query, and then tablets that had matching logos or colours, followed by tablets that had similar colours and/or logos before the rest. However, ranking tablets with matching or similar colours, but with dissimilar logos into such sequence was not easy, even though magnitude normalisations were implemented. After the "distances" were normalised, the distances representing two similar logos were of magnitude 10^{-1}. However, the inner product representing two similar colours could still be of magnitude 10^{-3}. The difference was still significant.

One solution was to rank the entries separately with respect to their logo and colour similarities. To find entries that matched both the logo and colour features of the query tablet, those that appeared above a certain rank (arbitrarily set at 12^{th}) in both searches were identified (same as Intersection in Set Theory). This led to some success in finding tablets with matching logos and colours (Figure 4.45 and Figure 4.46).

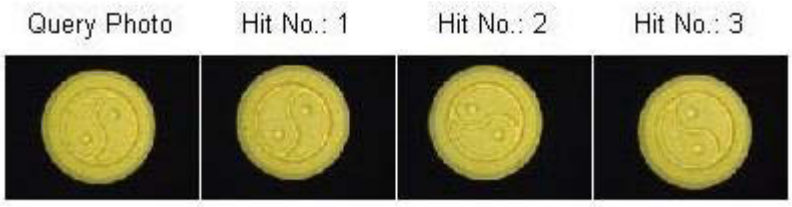


Figure 4.45: Combined logo and colour searching using intersection (only three out of the four tablets with matching colour and logo appeared). All four would appear if the ranking cut-off was set to 15.

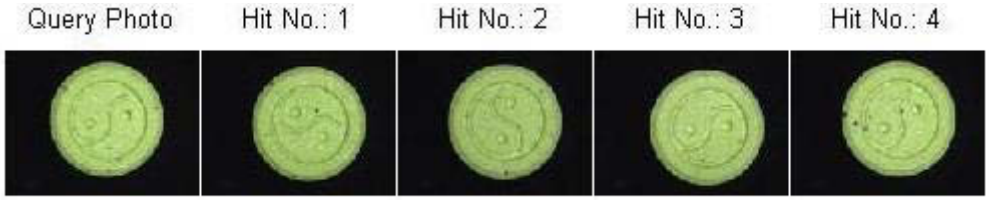


Figure 4.46: Another combined logo and colour search using intersection (all four tablets with matching colour and logo appeared).

Finding all entries that had matching colours *or* logos in a database could theoretically be accomplished by listing all candidates that appeared above a certain rank (again arbitrarily set at 12th) in both searches. This had the same effect as performing a union operation in Set Theory. Examples are shown below in Figure 4.47 and Figure 4.48.

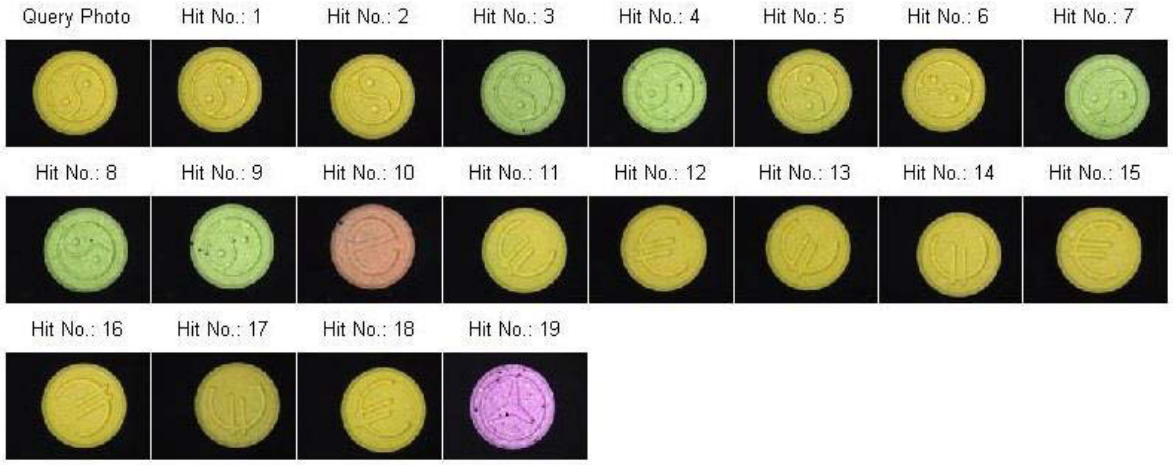


Figure 4.47: Combined logo and colour searching using union. Hit No. 10 and 19 appeared in the top 12 from the logo search, after all of the nine matching logos that were present in the database.

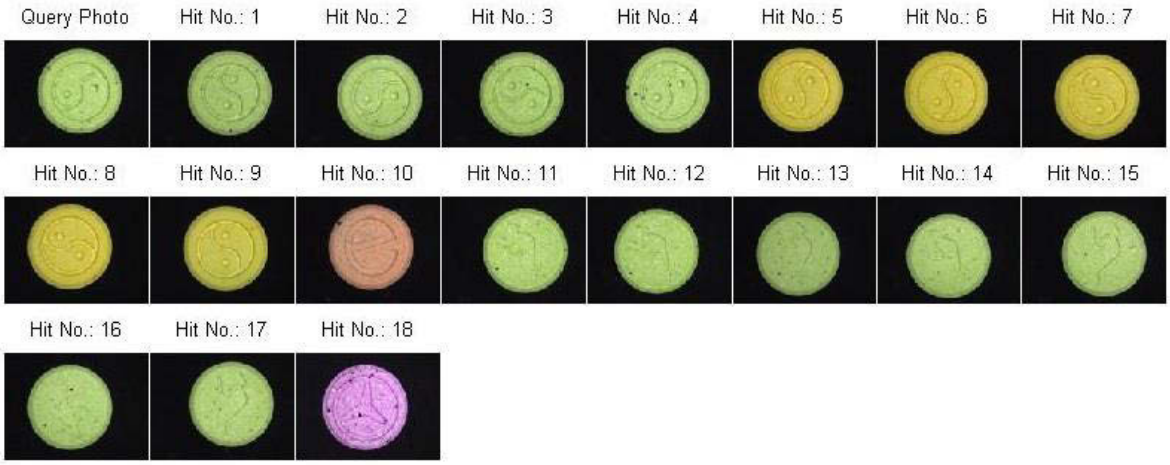


Figure 4.48: Another combined logo and colour searching using union. No. 10 and 18 were included for the same reason as in Figure 4.47.

4.5 Summary

This chapter has demonstrated that using fairly conventional colour comparison techniques, the most appropriate strategy to match the colours of typical illicit drug tablets is to first change the tablet images from the RGB colour space to the CIE L*a*b* colour space, and then rank colour similarities using the Normalised Inner Product. It is also possible to search tablets in a database by both logo and colour. Tablets with matching logo *and* colour can be identified by searching the two features individually before choosing the hits that appear in both searches (i.e. Intersection in Set Theory). Tablets with matching logo *or* colour can be found using Union (as in Set Theory) after searching the two features individually.

Chapter 5: IMPLEMENTATION INTO THE AFP ENIPID DATABASE

Chapter 5: Implementing CBIR into the AFP ENIPID database

The National Illicit Tablet Logo Database (NITLD) was developed in 2003 to capture and compile photos from seized illicit tablets across Australia into a national database. It sought to provide information that would contribute technical intelligence based on tablet logos and other characteristics, identify links between seizures and provide a strategic picture of tablets seized across the nation. The Enhanced National Intelligence Picture of Illicit Drugs (ENIPID) database currently houses drug profiling information on heroin, methylamphetamine, MDMA and cocaine seized by Australian State and Territory jurisdictions.

In 2013, UTS and AFP representatives attended a meeting hosted by the Western Australia Chemistry Centre in Perth, Australia, to discuss how to integrate the NITLD into the ENIPID database and add automated tablet searching functionality similar to the European Drug Profiling System (EDPS), on which the ENIPID system is based. It was decided that five photos will be stored for each tablet in the new ENIPID database. For each tablet surface (each tablet has two faces), two photos are taken one with the ring light at the normal position (for colour analysis) and one at a lowered position (for logo extraction). The fifth image will show the side edges of the tablet (see Section 5.3). Metadata stored in the image files will include case and sample numbers, and seizure details, as well as information that determines the feature extraction algorithm that each image will be subject to. Tablet searching functionality can be carried out using features such as the colour, shape, granularity, logo or combinations of these. Physical properties such as the diameter and thickness will also be automatically calculated instead of relying on user measurements. This project will be responsible for some of the features of the new tablet searching protocol.

5.1 Changes to the logo extraction script

One of the attendees of the Perth meeting, Martin Lopatka from the NFI (Netherlands Forensic Institute), recommended that we simplify the script whose development is described in Section 3.3.3. The suggestion was to replace the built-in MATLAB function *bwlabel* with another built-in function, *bwareaopen*, as the method for removing possible false positives (areas that are not part of the logo) in the BW images. By using the latter function, the false positives are assumed to be much smaller than the logos. Lopatka also suggested that truncating the images to only the size of the logos would make resizing of images unnecessary; this was advantageous since there was a chance that the rescaling process could destabilise the invariants.

These ideas were tested after the meeting. It was determined that preserving objects (regions of 1s) having an area greater than six per cent of the total tablet area was sufficient to remove false positives, while retaining the logo shapes. A summary of the rankings achieved using *bwareaopen* compared to what we had originally found (using *bwlabel* on the images described in Table 3.1) is listed in the table below. Precision (Pr) and Recall (Re) values are also included (using K = 11). As explained in the previous chapter, queries using one of the “three-pointed star” logos would yield four identical logo entries and the five other similar logo entries in the database. Therefore the number of positive hits for those searches was 4+5.

Table 5.1: Rankings by Euclidean distance of the positive hits within the first 11 using different morphological operations.

Logo (number of positive hits N_t in the first 11 (K))	Logo Extraction using *bwlabel* (Column 2 of Table 3.3)	Logo Extraction using *bwareaopen*
"Yin and Yang" (5)	$1^{st} – 4^{th}$ Pr: 0.36; Re: 0.80	$1^{st} – 3^{rd}$, 6^{th} Pr: 0.36; Re: 0.80
"8-ball" (3)	$1^{st} – 3^{rd}$ Pr: 0.27; Re: 1.00	$1^{st} – 3^{rd}$ Pr: 0.27; Re: 1.00
"Three-pointed Star -1, 2" (4 + 5)	$1^{st} – 9^{th}$ Pr: 0.82; Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82; Re: 1.00
	$1^{st} – 9^{th}$ Pr: 0.82; Re: 1.00	$1^{st} – 9^{th}$ Pr: 0.82; Re: 1.00
"Euro" (4)	$1^{st} – 4^{th}$ Pr: 0.36; Re: 1.00	$1^{st} – 4^{th}$ Pr: 0.36; Re: 1.00
"e" (4)	$1^{st} – 4^{th}$ Pr: 0.36; Re: 1.00	$1^{st} – 4^{th}$ Pr: 0.36; Re: 1.00
"Apple" (11)	$1^{st} – 5^{th}$, $7^{th} – 10^{th}$ Pr: 0.82; Re: 0.82	- Pr: 0.00; Re: 0.00
"Kangaroo" (13)	$1^{st} – 11^{th}$ Pr: 1.00; Re: 0.85	$1^{st} – 11^{th}$ Pr: 1.00; Re: 0.85

Again the comparison between the different methods was based on how many of the matching logos in the database appeared in the first 11 positions and whether there were any other logos ranked before them (false positives). It was obvious that using the morphological function *bwareaopen* with the six percent parameter did not improve the results. In fact, the new function returned a blank image for the query "Apple" logo in Figure 5.2 (see Table 5.1). These new sets of results were no better, and even slightly worse than those obtained using *bwlabel*. Not surprisingly, computational time was also found to be significantly longer to generate the six Zernike moment invariants from the truncated logo images than from the smaller, resized images.

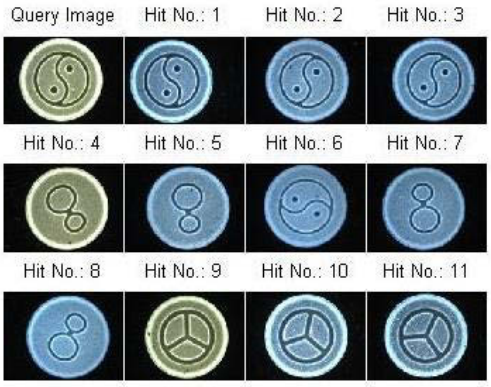


Figure 5.1: Logo searching using *bwareaopen* to extract logos and ranking using the Zermike moment invariants and Euclidean distance.

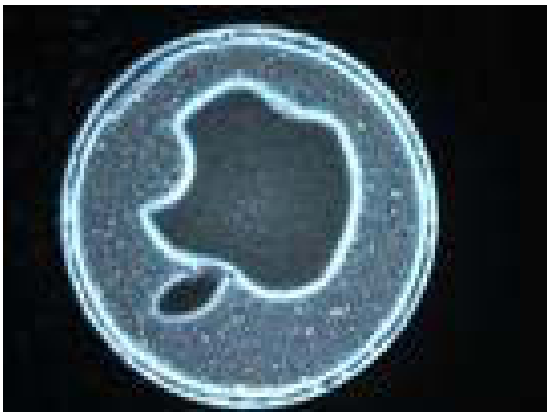


Figure 5.2: The "Apple" logo which could be extracted using *bwlabel* but not by *bwareaopen*.

These results did not suggest any reason to replace *bwlabel* with *bwareaopen* as this did not improve the searching algorithm.

5.2 Invariants for Symmetric Objects

Section 3.4 describes how a search for either of the "three-pointed star" logos would rank the four identical logo entries and the other similar logo entries together in random order. In an attempt to improve upon this, different methods were also investigated to try to prioritise the identical entries before the similar ones. It was realised that the common symmetry of the logos may be the reason why they were not able to be distinguished. An object has a *N-fold* symmetry if it appears identical after rotating around its centroid by $2\pi j/N$ where $j = 1, 2, \ldots, N$. Circles have an *infinite-fold* symmetry ($N = \infty$). By definition, both "three-pointed star" logos have a *3-fold* symmetry ($N = 3$).

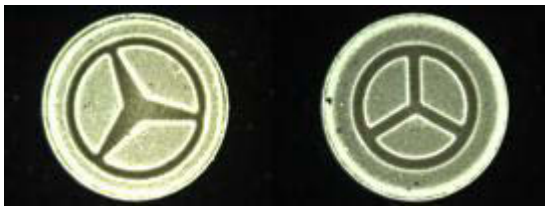


Figure 5.3: Both "three-pointed star" logos will appear identical when rotated 120° or $2\pi/3$ ($N = 3$).

Flusser et al. point out that "all odd-order moments of a centrosymmetric object are identically equal to zero" (60). Therefore its second and third order complex moment invariants (except c_{11}) are all zero, which provides little discrimination power. To counter

this, the authors formulated a new group of complex moment invariants (without using the zero moments) to recognise symmetric objects (logos).

An example in their book demonstrated the discrimination power of their proposed set of complex moment invariants (60). Simple shapes with various folds of symmetry (N = 2 – 6 and ∞) were successfully classified using this new set of invariants. This set contained only three complex moment invariants $c_{60}c_{06}$, $c_{50}c_{05}$ and $c_{40}c_{04}$, which were still invariant to translation, rotation and scaling. It was therefore decided that, as part of the current work, the same set of invariants would be tested on their ability to classify tablet logos. Logos with more than *6-fold* symmetry are not common and therefore these three invariants were thought sufficient to deal with the symmetric logos that the ENIPID database would encounter.

The MATLAB script (*rotmi*) by Flusser et al. was used to generate the specific complex moment invariants necessary, taking advantage of the magnitude normalisation methods provided by the authors (60). However, the dynamic range of these three invariants was still huge (from 10^{-5} to as high as 10^{12}) as they were at higher orders compared to the ones used for testing in Section 3.4.1 (6^{th} vs 3^{rd}), and a wide gap still existed using their normalisation methods. Euclidean distances were calculated to rank the distances between the query and the database invariants. Results were unsatisfactory (second column of Table 5.2). A lot of false positives were seen in the query searches. The major objective of this test was not met either: the invariants were not able to distinguish the two different “three-pointed star” logos (Figure 5.4 and Figure 5.5). An attempt was made to further minimise the dynamic gap by replacing the Euclidean distance algorithm with the standardised Euclidean distance (third column of Table 5.2), but no success was achieved (Figure 5.6 and Figure 5.7).

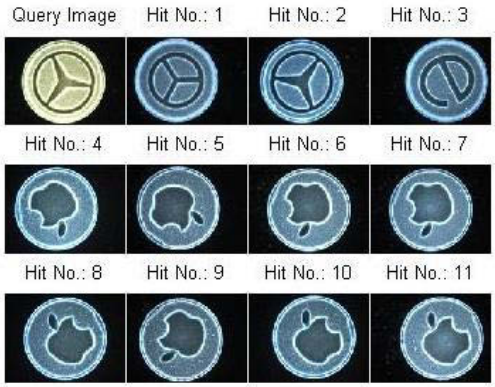


Figure 5.4: Searching one of the "three-pointed star" logos using complex moment invariants for symmetric objects and Euclidean distance.

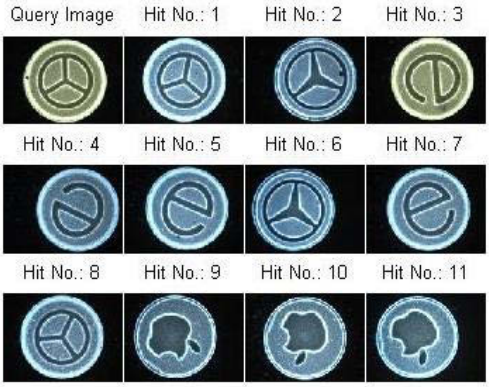


Figure 5.5: Searching the other "three-pointed star" logo using complex moment invariants for symmetric objects and Euclidean distance.

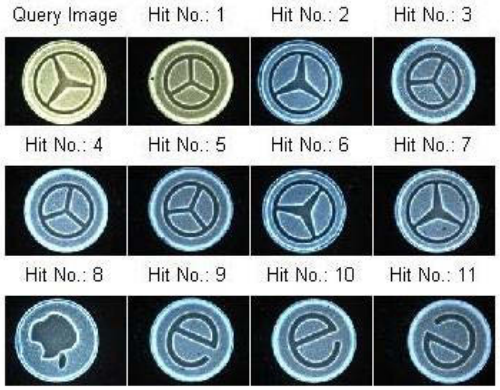


Figure 5.6: Searching one of the "three-pointed star" logos using complex moment invariants for symmetric objects and standardised Euclidean distance.

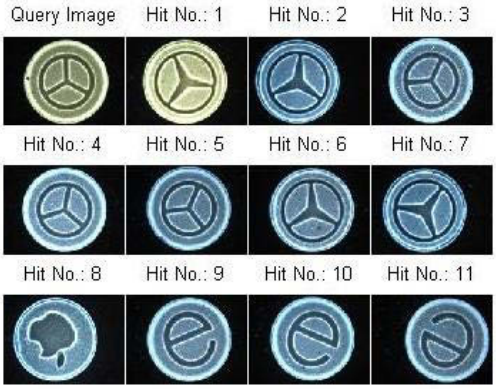


Figure 5.7: Searching the other "three-pointed star" logo using complex moment invariants for symmetric objects and standardised Euclidean distance.

These results were surprising, and so were further investigated. Initially, the suspicion was that Flusser et al.'s code for magnitude normalisation within their script *rotmi* was not effective above the 3rd order (60). However, an inspection of the code yielded no answers. In fact, it was shown in Section 3.4.1 that their script was able to generate a set of complex moment invariants from which reasonable searching results could be obtained. The clue actually came from the list of the values of the three new invariants generated. Negative numbers were found in the list, which should have been impossible. The three complex moment invariants, $c_{60}c_{06}$, $c_{50}c_{05}$ and $c_{40}c_{04}$, were each generated from the multiplication of a complex number ($a + ib$) by its complex conjugate ($a - ib$). Therefore, for any *real* numbers a and b, the product of the two complex numbers is $a^2 + b^2$, which should be a real number greater than or equal to zero.

Unable to explain why negative numbers were present in the list of invariants, it was decided to extract the necessary complex moments ourselves and perform the multiplication, which now yielded only positive values. The same normalisation code was used as in Flusser et al.'s script *rotmi* (60). Their normalisation involved taking the roots of the raw invariants, and the roots to be taken were dependent on the order of the invariants. For our three complex moment invariants, the square root was enough. After all these calculations, logo searching results immediately improved dramatically using Euclidean distance (Figure 5.8 and Figure 5.9) and standardised Euclidean distance (Figure 5.10 and Figure 5.11). Again, the invariants were not able to distinguish the two "three-pointed star" logos using either distance calculation. In addition, the search results for other logos were still inferior to those using Zernike moment invariants at the 3rd order that had been achieved previously (4th and 5th column of Table 5.2 respectively).

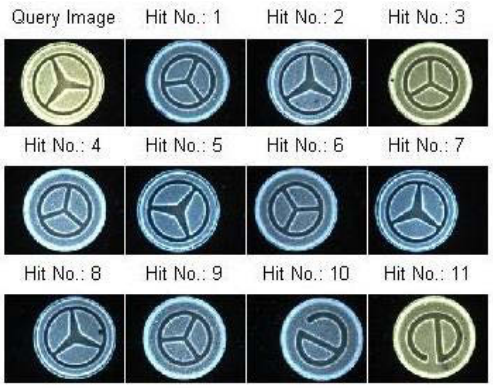


Figure 5.8: Searching one of the "three-pointed star" logos using complex moment invariants for symmetric objects (not from *rotmi*) and Euclidean distance.

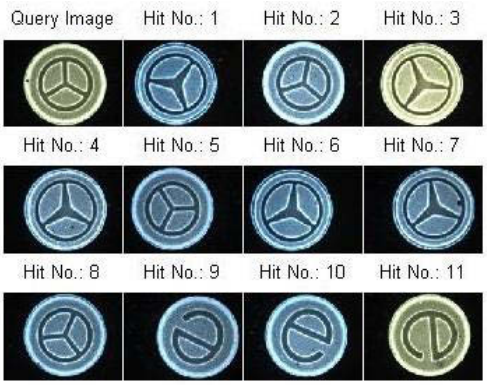


Figure 5.9: Searching the other "three-pointed star" logo using complex moment invariants for symmetric objects (not from *rotmi*) and Euclidean distance.

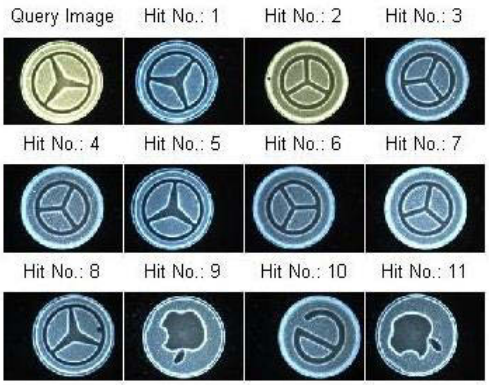


Figure 5.10: Searching one of the "three-pointed star" logos using complex moment invariants for symmetric objects (not from *rotmi*) and standardised Euclidean distance.

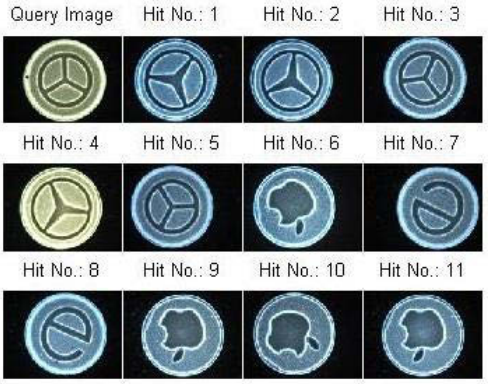


Figure 5.11: Searching the other "three-pointed star" logo using complex moment invariants for symmetric objects (not from *rotmi*) and standardised Euclidean distance.

The failure of the new complex moment invariants to recognise the two *3-fold* symmetric logos in the dataset tested here was probably due to the fact that the logos were very similar, even to the naked eye. In addition, this set of moment invariants suggested by Flusser et al. was for simple shapes, which was not always the case for the logos tested here (60). Their database contained photos of the *same* objects with random rotations, which made classification easier. In contrary, all logos in the database tested here were extracted individually from *different* tablets and had slight variations. Moreover, the other symmetric logo in our dataset was the "Yin and Yang" logo (*2-fold*). The database did not contain enough symmetric logos of different folds for a comprehensive study to determine a suitable set of moment invariants for classification, given the limited number of dies that were available. More logos would need to be made available, probably from future AFP seizures, for further studies.

Table 5.2: Rankings of the positive hits within the first 11 using moment invariants for recognising symmetric logos normalised by different methods.

Logo (number of positive hits N_t in the first 11 (K))	Ranking invariants from *rotmi* using Euclidean distance	Ranking invariants from *rotmi* using standardised Euclidean distance	Ranking invariants (not from *rotmi*) by the Euclidean distance	Ranking invariants (not from *rotmi*) by the standardised Euclidean distance
"Yin and Yang" (5)	Nil Pr: 0.00 Re: 0.00	Nil Pr: 0.00 Re: 0.00	$1^{st} - 4^{th}$ Pr: 0.36 Re: 0.80	1^{st}, 2^{nd} Pr: 0.36 Re: 0.80
"8-ball" (3)	Nil Pr: 0.00 Re: 0.00	Nil Pr: 0.00 Re: 0.00	2^{nd}, 4^{th}, 11^{th} Pr: 0.27 Re: 1.00	2^{nd}, 4^{th}, 5^{th} Pr: 0.27 Re: 1.00
"Three-pointed Star -1" (4)	2^{nd} (Figure 5.4) Pr: 0.09 Re: 0.25	2^{nd}, 6^{th}, 7^{th} (Figure 5.6) Pr: 0.27 Re: 0.75	2^{nd}, 5^{th}, 7^{th}, 8^{th} (Figure 5.8) Pr: 0.36 Re: 1.00	1^{st}, 5^{th}, 8^{th} (Figure 5.10) Pr: 0.27 Re: 0.75
"Three-pointed Star -2" (4)	1^{st}, 8^{th} (Figure 5.5) Pr: 0.18 Re: 0.50	$3^{rd} - 5^{th}$ (Figure 5.7) Pr: 0.27 Re: 0.75	2^{nd}, 5^{th}, 8^{th} (Figure 5.9) Pr: 0.27 Re: 0.75	3^{rd}, 5^{th} (Figure 5.11) Pr: 0.18 Re: 0.50
"Euro" (4)	1^{st} Pr: 0.09 Re: 0.25	$1^{st} - 3^{rd}$ Pr: 0.27 Re: 0.75	$1^{st} - 4^{th}$ Pr: 0.36 Re: 1.00	$1^{st} - 4^{th}$ Pr: 0.36 Re: 1.00
"e" (4)	1^{st}, 2^{nd}, 9^{th}, 10^{th} Pr: 0.36 Re: 1.00	2^{nd}, 3^{rd}, 5^{th} Pr: 0.27 Re: 0.75	$1^{st} - 4^{th}$ Pr: 0.36 Re: 1.00	$1^{st} - 4^{th}$ Pr: 0.36 Re: 1.00
"Apple" (11)	$1^{st} - 4^{th}$, 9^{th} Pr: 0.45 Re: 0.45	$1^{st} - 3^{rd}$, 5^{th} Pr: 0.36 Re: 0.36	$1^{st} - 8^{th}$ Pr: 0.73 Re: 0.73	$1^{st} - 7^{th}$ Pr: 0.64 Re: 0.64
"Kangaroo" (13)	1^{st}, 2^{nd}, 4^{th}, 5^{th} Pr: 0.36 Re: 0.31	$1^{st} - 5^{th}$, 7^{th} Pr: 0.55 Re: 0.46	$1^{st} - 11^{th}$ Pr: 1.00 Re: 0.85	$1^{st} - 8^{th}$, 10^{th} Pr: 0.82 Re: 0.69

5.3 Supplementary feature: side profiles

During the Perth meeting, it was decided that comparing the side profiles of the tablets would be a useful ability for a new database. Knowledge of whether a tablet's surface is flat or domed would increase the discrimination power of tablet searching. Therefore side profiles were introduced as part of the features available for searching. A fifth image would be used to extract two profiles for each tablet, one profile for each surface (Figure 5.12).

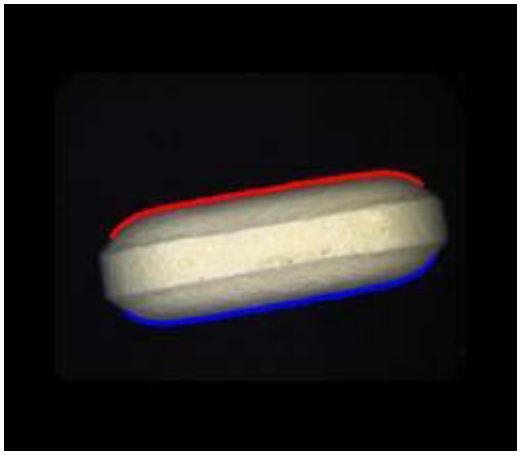


Figure 5.12: The fifth image for a tablet. Each tablet has two side profiles, illustrated by the red line and the blue line.

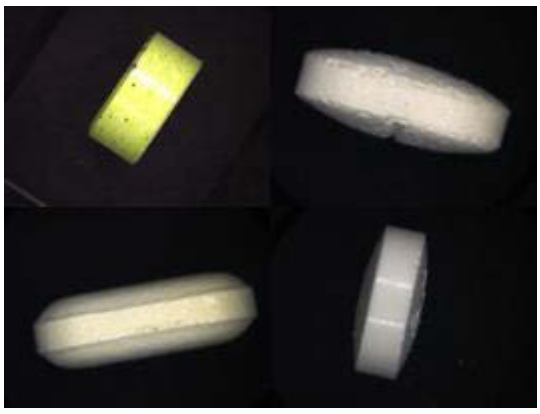


Figure 5.13: Some examples of the side profile images in our database (total 30).

In the discussions below, assume that the *xy*-plane is the plane of the tablet image, the major axis of a side profile is parallel to the *y*-axis and the *z*-axis is perpendicular to the *xy*-plane. The origin is at the centre of the tablet (Figure 5.14).

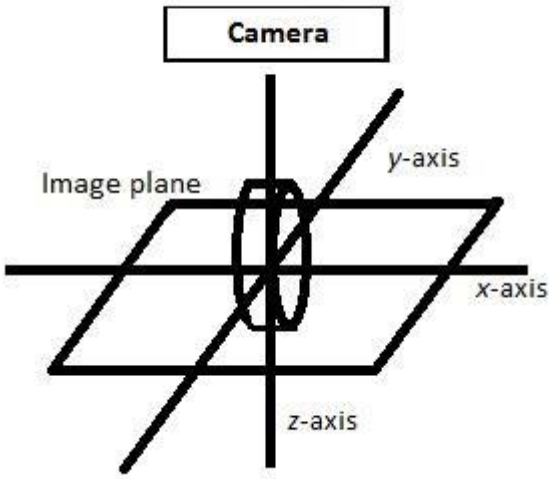


Figure 5.14: Defining orientations for discussions in this section.

5.3.1 Method

The images from which the side profiles were extracted were also taken using the Leica DFC295 camera mounted on the MZ6 macroscope. The pre-processing step was simple: they were converted to a binary image simply by using a threshold of 0.1 (Figure 5.15). This threshold was shown to be low enough to segment the whole tablet from the black background effectively. The MATLAB morphological operation *imfill* then filled any holes present within the tablet sides and another operation *bwareaopen* removed all the small false positives (white dots that were not part of the tablet). The resulting images were then rotated such that the major axis of the tablet would be parallel to the *y*-axis of the image. This rotated the tablets into a standard, vertical position for comparison (Figure 5.16).

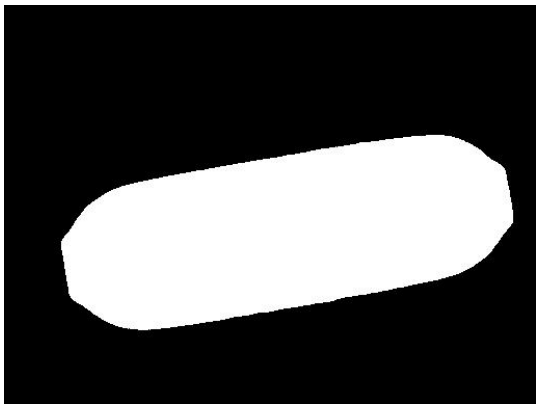


Figure 5.15: The binary image of Figure 5.12 using the threshold of 0.1.

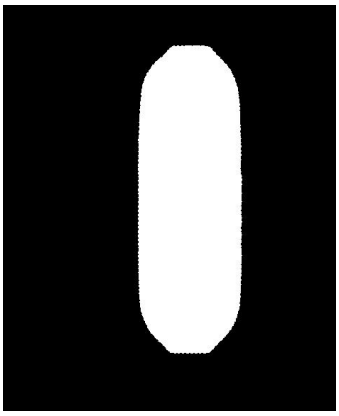


Figure 5.16: Figure 5.15 was rotated such that the major axis was parallel to the *y*-axis.

5.3.2 Derivatives

The first proposal was to take the slopes of the tangent (*dx/dy*) at multiple points along a tablet surface and average the numbers to represent the side profile; however, it was eventually realised that this approach was flawed. If the starting and ending points of a side profile are at the same *x*-coordinate, the average of the slopes has to be equal to zero. If these points had different *x*-values, the average slope only represented the change of *x*-coordinate precisely at two ends of the profile. Therefore it did not carry any meaningful information about the changes in slope along the two side if the tablet. Representing a side profile with only one number did not seem to be sufficient either. A decision was made to use two rows of slope values, one for each side of the major axis; i.e. a separate vector representing each side profile for comparison purposes.

In order to find the slopes of tangents at different points, the first derivatives at those locations could be used. The Savitzky-Golay filter was initially proposed for estimating the first derivatives along the tablet surface. The MATLAB built-in function *sgolay* returns coefficients necessary to calculate derivatives of any order. However, one important factor was not considered. The intervals between points on the side profiles of different tablets would not be in equal length because tablet sizes could be different. Hence comparing the first derivatives between different side profiles were invalid as the vector lengths (numbers of elements/variables) would be different.

A proposed solution to this problem is as follows. Viewed from the side, the tablet will look like a rectangle if both of its surfaces are flat or it will look like an oval if the surfaces

are domed. The side profile can be made invariant to camera magnifications or tablet sizes (and hence the problem solved) by dividing the major axis of the rectangle or the oval by the same number of intervals. These intervals would therefore have different pixel lengths for different images (compare the distances between dots along the side profiles in Figure 5.17 and Figure 5.18). So, slopes of the tangents were taken along tablet surfaces at these intervals. With the image already rotated such that all tablets were in a vertical position, the slopes at the intervals could be obtained without using the more complicated Savitzky-Golay filter (for coefficients to estimate derivatives). Derivatives at each interval could be calculated simply by finding the number of pixels displaced relative to its preceding interval in the horizontal direction (*dx*) and dividing it by the interval length in pixels (*dy*).

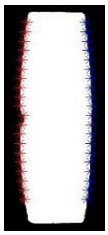


Figure 5.17: The location of the 20 points where the slopes were calculated.

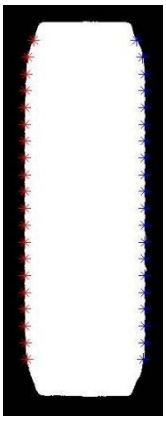


Figure 5.18: The location of the 20 points in another tablet.

This database contained 30 photos of the sides of different tablets. As each image generated two side profiles, 60 rows of derivatives were available. Promising results from these side profiles showed that it was possible to search for tablets with similar side profiles using the vectors derived above. Most side profiles that matched the queries were ranked at the top when the number of intervals was set to 20 (Figure 5.19 - Figure 5.22). Increasing the number of intervals did not improve the results (see Figure 5.21, Figure 5.23 and Figure 5.24), and the matching profile(s) were still ranked lower than other profiles.

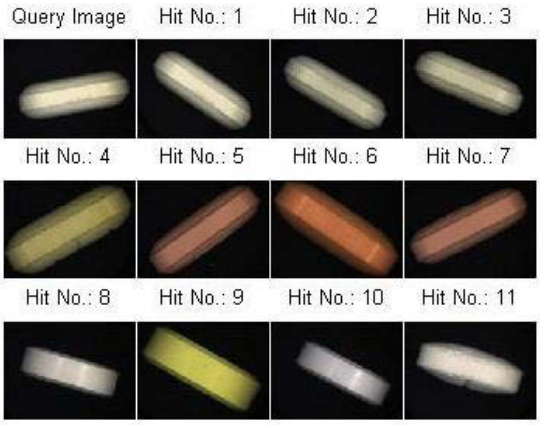


Figure 5.19: Searching one of the two side profiles of the query image. All other seven tablets with the same profiles appeared in the top 11 hits.

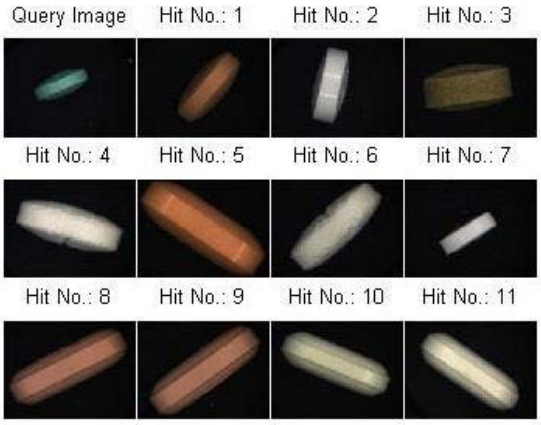


Figure 5.20: Searching one of the two side profiles of another query image. All other three tablets with the same profiles appeared in the top 11 hits.

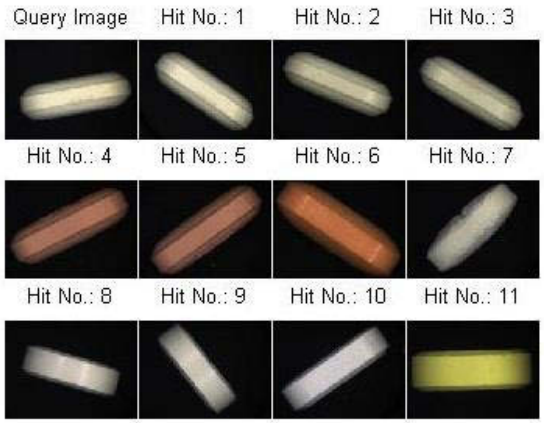


Figure 5.21: Searching the other side profile of the query image in Figure 5.19. Only six out of the seven tablets with the same profiles appeared in the top 11 hits.

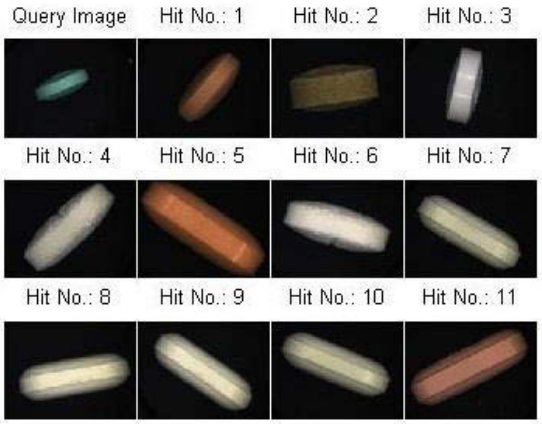


Figure 5.22: Searching the other side profile of the query image in Figure 5.20. All other three tablets with the same profiles appeared in the top 11 hits.

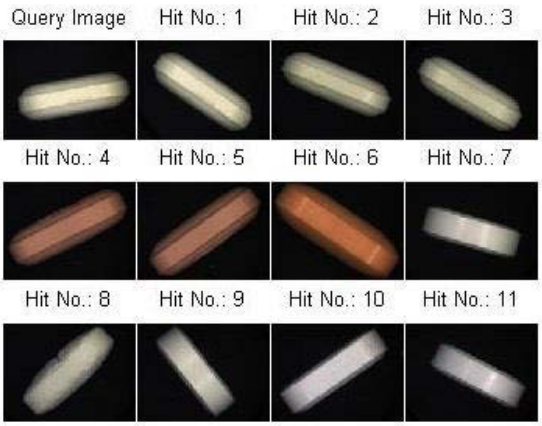


Figure 5.23: The same search as in Figure 5.21 but the number of intervals was set to 25 instead of 20. Only six out of the seven tablets with the same profiles appeared in the top 11 hits.

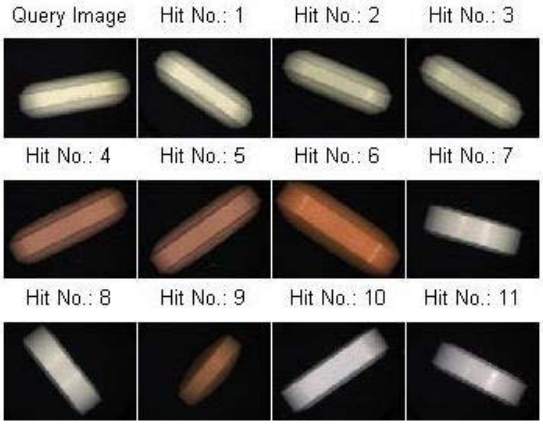


Figure 5.24: The same search as in Figure 5.21 again but the number of intervals was set to 50 instead of 20. Only six out of the seven tablets with the same profiles appeared in the top 11 hits.

However, using derivatives to describe side profiles for comparison could bring some problems: so far this approach does not account for side profiles that are not symmetric. Here, problems occur because there are no reference points to refer to on the side of a tablet so as to determine a standard position for photography. The major problem is that tablets that do not have symmetric surfaces will present different side profiles if they are rotated around the *x*-axis. In practise, the orientation in which tablets are placed vertically under the camera with respect to the *x*-axis is random. In Figure 5.25, the left-hand side profile generated will be different at different orientations. In this case, there is no solution to the problem, no matter what kind of feature extraction or comparisons are used. This will be a major limitation to the application of this idea in tablet searching. Fortunately, the AFP database does not contain many tablets with convex, asymmetric surfaces (at present), though this may change.

Figure 5.25: Asymmetric side profiles could be very different depending on how the tablet was placed under the camera.

A minor issue is that the vector containing the slopes along one side of a tablet will change if the tablet is rotated 180° around the *z*-axis. In order to extract the same vector regardless of this orientation, the slopes of tangents have to be consistently sampled in either a clockwise or an anti-clockwise direction. This will be crucial if the tablet surfaces are asymmetric (side profiles A and B in Figure 5.26). The derivatives cannot be collected just from the top to the bottom (or vice versa), since even a tablet with side profiles that are both symmetric but are different from each other (side profiles C and D in Figure 5.27) will yield slopes of opposite signs if the tablet is rotated 180° (compare side profiles C and C' in Figure 5.27).

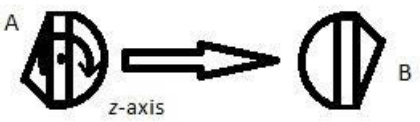


Figure 5.26: Asymmetric side profiles A and B match if only their derivatives are collected clockwise or anti-clockwise.

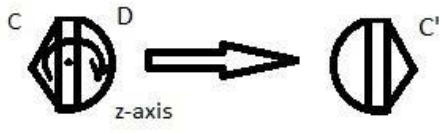


Figure 5.27: Symmetric side profiles C and D are vectors of opposite signs if the derivatives are collected just from the top to the bottom (or vice-versa).

Another concern is that there are no reference points on tablets to define one side as either “left” or “right”. Therefore, as examiners randomly place an asymmetric tablet under the camera, side profiles that are mirror images of each other can be detected (achieved by rotating an asymmetric tablet 180° around the *y*-axis). Collecting the derivatives in a clockwise (or anti-clockwise) direction or from the top to the bottom (or vice versa) cannot solve this problem. The side profiles of the two mirror images are the negatives of each other (side profiles E and F in Figure 5.28), and vector comparisons will not be straight forward and would be made irrelevant by the issue of rotation about the *x*-axis (above).

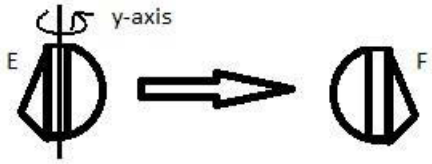


Figure 5.28: Asymmetric side profiles E and F have the same derivatives but different signs.

5.3.3 Moment Invariants

With these limitations of using derivatives for asymmetric side profile comparisons, a suggestion was made in which the outline of a tablet was compared as a whole. Moment invariants would be suitable for this comparison as they are rotation and scale invariant. How derivatives were collected (clockwise or anti-clockwise; top to the bottom or vice versa) were irrelevant as well. Some invariants are insensitive to objects that were mirror images of each other according to Flusser et al. (60). These characteristics of moment invariants address the problem of matching an object (i.e. the side of the tablet) rotated along the *y*- and *z*-axis and itself.

Despite these advantages, there are issues with using moment invariants to compare tablet sides. This was because the two side profiles on each tablet could not be searched

separately. Using a commercial machine, tablets are manufactured by pressing ingredients using two dies, one on top and one at the bottom. By searching the two profiles separately, both dies could be matched and classified separately. However, moment invariants can only match a pair of dies but not individual dies. It may be advantageous to search only a side profile for that particular shape of die.

Nevertheless, the Zernike moment invariants and two sets of complex moment invariants were computed and compared using the Euclidean distance as a trial. The first set of complex moment invariants (CMI1) was the original third order set with $r = 3$, $p_0 = 2$ and $q_0 = 1$. The output was a row of six numbers. The other set (CMI2) took into account of the fact that most of the tablet outlines had a *2-fold* symmetry. The parameters used to generate this set were $r = 4$, $p_0 = 2$ and $q_0 = 0$. This set of invariants contained a row of seven numbers for each shape. The numbers represented the following invariants: c_{11}, c_{22}, $c_{20}c_{02}$ and the real and imaginary parts of $c_{31}c_{02}$ and $c_{40}c_{02}^2$.

The search results were poor (see Figure 5.29 - Figure 5.34). Photos were ranked randomly against the query images. The best search achieved was using the first set of complex moment invariants (Figure 5.31 and Figure 5.32). In these two query examples, some of the matching profiles were ranked at the top.

Considering that moment invariants could only compare tablet outlines as a whole but not individual side profiles, these results showed that moment invariants, as a feature, were not the best technique for this purpose. Searching a side profile using a row of its derivative was a better alternative and could yield better results.

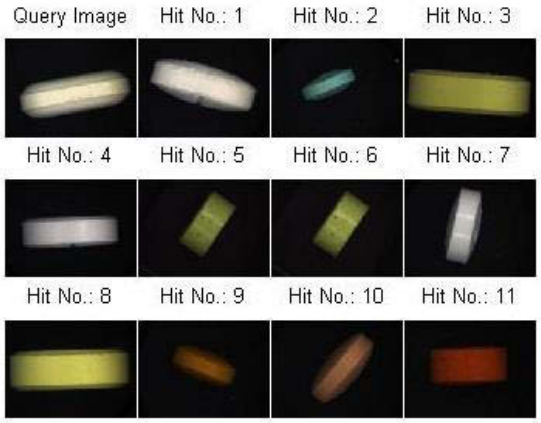


Figure 5.29: Searching the side of the query tablet using Zernike moment invariants. None of the seven tablets with the same profiles appeared in the top 11 hits.

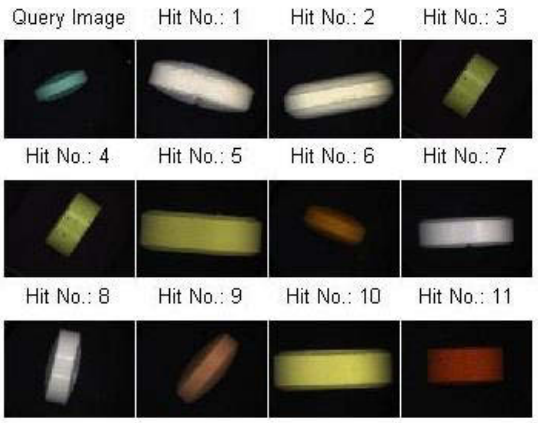


Figure 5.30: Searching the side of another query tablet using Zernike moment invariants. Only two out of the three tablets with the same profiles appeared at 8th and 9th.

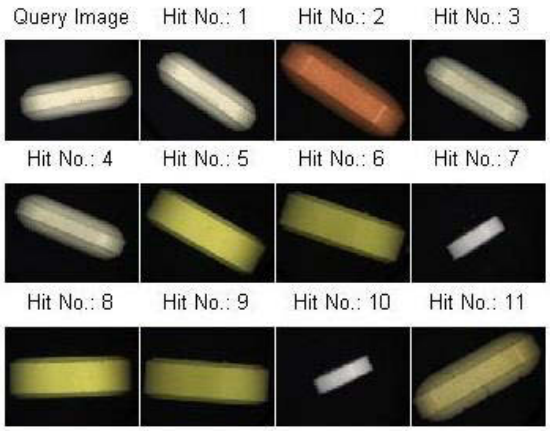


Figure 5.31: Searching the side of the query tablet using complex moment invariants (CMI1). Five of the seven tablets with the same profiles appeared at 1st-4th and 11th.

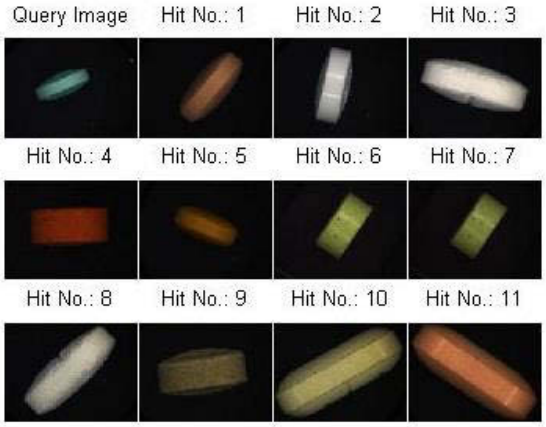


Figure 5.32: Searching the side of another query tablet using complex moment invariants (CMI1). All other three tablets with the same profiles appeared at 1st, 2nd and 9th.

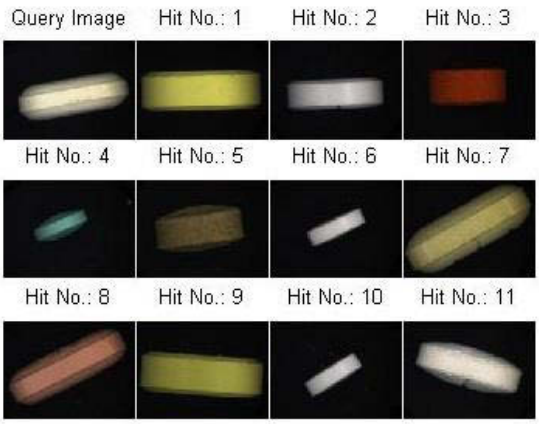


Figure 5.33: Searching the side of the query tablet using complex moment invariants (CMI2). Two of the seven tablets with the same profiles appeared at 7th and 8th.

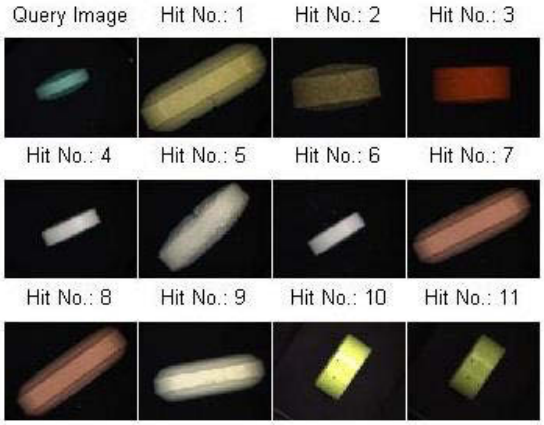


Figure 5.34: Searching the side of another query tablet using complex moment invariants (CMI2). Only one out of the three tablets with the same profiles appeared at 2nd.

5.4 Compiling MATLAB scripts to executable files

The scripts developed in MATLAB have to be translated to some other languages in order for them to be utilised in an external database system (such as the ENIPID system) that does not run the MATLAB software. An add-in toolbox from MATLAB called the MATLAB Compiler is available to convert any script to a number of different executable files so that any database computers can run the algorithms on the appropriate photos. These executable files are designed to extract logos, colours and side profiles as described in this thesis, and generate the respective feature vectors.

More information than only logos and side profiles can be extracted from the photos taken. The measurement of certain physical properties can be automated using these images. Specifically, measuring and recording the diameter and the thickness of tablets automatically are desirable and will further reduce the labour and time needed for tablet examination. These physical properties are characteristics of the tablets and are valuable information for intelligence investigations (141, 142).

The area of the surface of a tablet will give its radius and therefore the diameter in pixels if it is assumed to be a circle. Therefore, the image used for logo extraction can also be used to determine the diameter. Even if a tablet is not circular, this “diameter” can be used as a feature to denote a tablet’s relative size in a search. An executable file has been compiled for logo image processing which includes extracting and calculating the moment invariants of logos, as well as providing the diameter of the tablet as another output. A separate executable file has been compiled to extract tablet colours and convert them from RGB colour coordinates to CIE L*a*b* coordinates for matching.

For tablets that are not circular, the “diameter” determined above has no meaning to a human operator and so if geometric parameters/dimensions appropriate to the real shape need to be measured, another strategy is required. Tablet shape classification such as the method published by Lopatka and van Houten can be used to classify the tablets into different shape categories (120), after which the appropriate real dimensions can be calculated.

The measurement of the thickness of a tablet in pixels is straight forward using the image of its side. Therefore, the image used for extracting side profiles can also be used to find the thickness. Another executable file has been compiled to extract the side profiles, with an extra output for the thickness of the tablet.

The measurements of these physical properties are in pixels. They can be converted to lengths in millimetres (mm) with ease using a ratio which is available as a part of the internal AFP validation process.

5.5 Summary

Complex moment invariants that were designed to describe symmetric logos were used in an attempt to differentiate the two “three-pointed star” logos. However, this approach was unsuccessful. A lack of other symmetric logos in our data set prevented further investigations. The side profiles of tablets were examined as a new feature for tablet searching. Promising results were achieved, although there remain problems that need to be solved. Specifically, answers have to be found for the difficult question of how to encode asymmetric side profiles for searching.

Chapter 6: CONCLUSION

Chapter 6: Conclusion

The aim of this project was to automate illicit tablet logo searching using image retrieval algorithms instead of manually tagging those logos in tablet photos and carrying out the search using keywords and texts. It was initially attempted to use the existing tablet images that the Australian Federal Police (AFP) had collected previously. However, segmenting the logos from these images proved to be difficult. False positives (regions incorrectly identified by image processing techniques as logos) and false negatives (logo regions that were not classified as such) were commonplace. The deduction from these failures was that the images were poorly illuminated and did not have sufficient contrast at the logo boundaries to segment the logos from the tablets. Therefore, a new imaging methodology was required.

New imaging methods were investigated. It was found that the most effective way of introducing maximum logo contrast was by lowering the ring light attached to the macroscope so that it was very close to a tablet. Logo extraction and similarity searching was done using these new “high contrast” images.

An in-house algorithm for converting those colour images to black-and-white (BW) images was developed. It was based on the assumption that the most common colour value, other than the background, is the colour value of the tablet surface. Hence the threshold for this conversion was dependent on this value. Morphological and logical operations were then used to process these BW images such that they contained only the silhouettes of the logos.

Logo feature extraction and comparisons were carried out using moment invariants. Search results using the complex moment invariants and the Zernike moment invariants were compared. The Hu moment invariants were not tested as they are known to be incomplete and mutually dependent. It was discovered that encoding the logos using the Zernike moment invariants with the Kintner method, followed by the normalisation approach, then searching by Euclidean distance gave the best chance of correctly matching logos in the database.

Testing was also done for matching illicit tablets by colour. Another new set of images was used for colour matching because these “high contrast” images did not represent what examiners perceive under normal lighting conditions. In this project, colour values were expressed and compared as vectors in the HSV and CIE L*a*b* colour spaces. The RGB colour values were not used as this colour space is known not to be perceptually uniform. The best results were achieved using the Normalised Inner Product to compare the CIE L*a*b* colour coordinates in the database against the query.

The potential of side profiles of tablets to be used as another characteristic feature was explored. The side profiles were found to be represented best by rows of derivatives (vectors) along the profile. Matching side profiles appeared at the top positions in the searches using the Euclidean distance to compare the rows of numbers. However, this feature had its limitations. The limitations included rotational issues around different axes, and the effect of asymmetric side profiles on the vector comparison when tablets with the same asymmetric profiles were placed in different orientations relative to each other.

In order to generate the images required (logos, colour and side profiles) for the ENIPID database to carry out the searching procedures described in this work, photographic guidelines were drafted with the AFP as the basis for their Standard Operating Procedure. A summary of these guidelines is as follows:

1. Make sure the ring light is at its highest position;
2. Carry out off-centre white balance;
3. Place tablet flat on a platform which is then placed on the stage of the macroscope;
4. Focus on the surface of the tablet and take an image for colour searching (label it “Side 1 – Upper Light Position”);
5. Turn tablet over and without changing any settings, take another image for colour searching (label it “Side 2 – Upper Light Position”);
6. Lower the ring light closer to the tablet such that the logo area on the tablet is darker than the surface (change to higher exposure if necessary);

7. Take the third image for logo searching (label it “Side 2 – Lower Light Position”);
8. Turn the tablet back to Side 1 and take the fourth image for logo searching (label it “Side 1 – Lower Light Position”);
9. Return the ring light back to the Upper position and reset the camera configuration if necessary; and
10. Position the tablet on its edge and take the fifth image for side profile searching (label it “Edge – Upper Light Position”).

Unlike other published work, this project has focussed on tablet logo and side profile searching. Most researchers working on this forensic application of image retrieval have reported drug tablet searching results using colour, shape, texture and granularity of the tablets but (for the most part) tablet logos have not been addressed. The work by Lee et al. was similar to ours in that logos, as a feature, were being searched/compared (18), but the work described in this thesis has aimed to use lower-dimensioned descriptors to describe the logos, and to avoid the use of the Hu moment invariants, for the reasons discussed in Section 1.3.4.4 of this thesis.

A follow-up to this work can include exploring the most suitable texture descriptor for tablet searching and possibly implementing it into the AFP drug tablet database, although the granularity work by Lopatka and Vallat may achieve this aim (119). Another area to investigate will be the problems associated with unambiguously encoding images of high-symmetry logos, and possibly the symmetry problems encountered with searching tablet side profiles (although it is possible some of these issues will have no solution).

In conclusion, the methods described in this thesis allow for the automated extraction of logo, colour and side profile information from an illicit drug tablet image database. They remove the need for subjective or inconsistent data entry by operators and reduce the operational time required for the extraction of information. The algorithms used/described here can easily be combined with others devised for searching based on (say) tablet shape or texture/granularity.

REFERENCES

References

[1] (2011) Drugs in Australia 2010, Australian Institute of Health and Welfare, Canberra.

[2] (2014) Illicit Drug Data Report 2012-13, 11 ed., Australian Crime Commission, Canberra.

[3] Rui, Y., Huang, T. S., and Chang, S.-F. (1999) Image Retrieval: Current Techniques, Promising Directions, and Open Issues, *Journal of Visual Communication and Image Representation 10*, 39-62.

[4] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000) Content-based image retrieval at the end of the early years, *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 1349-1380.

[5] Gevers, T., and Smeulders, A. (2004) Image search engines: An overview, *Emerging Topics in Computer Vision*, 1-54.

[6] Datta, R., Li, J., and Wang, J. Z. (2005) Content-based image retrieval: approaches and trends of the new age, ACM Press.

[7] Liu, Y., Zhang, D., Lu, G., and Ma, W.-Y. (2007) A survey of content-based image retrieval with high-level semantics, *Pattern Recognition 40*, 262-282.

[8] Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008) Image retrieval: Ideas, influences, and trends of the new age, *ACM Computing Surveys 40*, 1-60.

[9] Müller, H., Michoux, N., Bandon, D., and Geissbuhler, A. (2004) A review of content-based image retrieval systems in medical applications—clinical benefits and future directions, *International journal of medical informatics 73*, 1-23.

[10] Chen, Y., Roussev, V., Richard, G., and Gao, Y. (2005) Content-Based Image Retrieval for Digital Forensics, In *Advances in Digital Forensics* (Pollitt, M., and Shenoi, S., Eds.), pp 271-282, Springer Boston.

[11] Srihari, S. N., and Shi, Z. (2004) Forensic handwritten document retrieval system, In *Proceedings of the First International Workshop on Document Image Analysis for Libraries*, pp 188-194.

[12] Jain, A. K., Lee, J.-E., and Jin, R. (2007) Tattoo-ID: automatic tattoo image retrieval for suspect and victim identification, In *Proceedings of the multimedia 8th Pacific Rim conference on Advances in multimedia information processing*, pp 256-265, Springer-Verlag, Hong Kong, China.

[13] Jain, A. K., Lee, J.-E., and Jin, R. (2009) Graffiti-ID: matching and retrieval of graffiti images, In *Proceedings of the First ACM workshop on Multimedia in forensics*, pp 1-6, ACM, Beijing, China.

[14] Jain, A. K., Lee, J.-E., Jin, R., and Gregg, N. (2009) Content-based image retrieval: An application to tattoo images, In *16th IEEE International Conference on Image Processing*, pp 2745-2748.

[15] Geradts, Z. J., Hardy, H., Poortman, A., and Bijhold, J. (2001) Evaluation of contents-based image retrieval methods for a database of logos on drug tablets, In *Enabling Technologies for Law Enforcement*, pp 553-562, International Society for Optics and Photonics.

[16] Geradts, Z., and Bijhold, J. (2002) *Content-based information retrieval from forensic image databases*, Universiteit Utrecht.

[17] Chen, R.-C., Pao, C.-T., Chen, Y.-H., and Jian, J.-C. (2010) Automatic Drug Image Identification System Based on Multiple Image Features, In *Computational Collective Intelligence. Technologies and Applications* (Pan, J.-S., Chen, S.-M., and Nguyen, N., Eds.), pp 249-257, Springer Berlin Heidelberg.

[18] Lee, Y.-B., Park, U., Jain, A. K., and Lee, S.-W. (2012) Pill-ID: Matching and retrieval of drug pill images, *Pattern Recognition Letters 33*, 904-910.

[19] Long, F., Zhang, H. J., and Feng, D. (2003) Fundamentals of Content-based Image Retrieval, In *Multimedia information retrieval and management : technological fundamentals and applications* (Feng, D., Siu, W. C., and Zhang, H. J., Eds.), Springer, Berlin ; New York.

[20] Lukac, R., Smolka, B., Martin, K., Plataniotis, K. N., and Venetsanopoulos, A. N. (2005) Vector filtering for color imaging, *IEEE Signal Processing Magazine 22*, 74-86.

[21] Plataniotis, K. N., Androutsos, D., and Venetsanopoulos, A. N. (1999) Adaptive fuzzy systems for multichannel signal processing, *Proceedings of the IEEE 87*, 1601-1622.

[22] Oppenheim, A. V., Schafer, R. W., and Stockham, T. G., Jr. (1968) Nonlinear filtering of multiplied and convolved signals, *Proceedings of the IEEE 56*, 1264-1291.

[23] Yin, L., Astola, J., and Neuvo, Y. (1993) A new class of nonlinear filters-neural filters, *IEEE Transactions on Signal Processing 41*, 1201-1222.

[24] Haykin, S., Yee, P., and Derbez, E. (1997) Optimum nonlinear filtering, *IEEE Transactions on Signal Processing 45*, 2774-2786.

[25] Deng, Y., Kenney, C., Moore, M. S., and Manjunath, B. S. (1999) Peer group filtering and perceptual color image quantization, In *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, 1999*, pp 21-24.

[26] Mathias, E., and Conci, A. (1998) Comparing the Influence of Color Spaces and Metrics in Content-based Image Retrieval, In *Proceedings of the International Symposium on Computer Graphics, Image Processing, and Vision*, p 371, IEEE Computer Society.

[27] Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., and Yamada, A. (2001) Color and texture descriptors, *Circuits and Systems for Video Technology, IEEE Transactions on 11*, 703-715.

[28] Wyszecki, G., and Stiles, W. S. (1982) *Color science : concepts and methods, quantitative data and formulae*, 2nd ed., Wiley, New York.

[29] Wang, J., Yang, W.-J., and Acharya, R. (1997) Color clustering techniques for color-content-based image retrieval from image databases, In *IEEE International Conference on Multimedia Computing and Systems*, pp 442-449.

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

[31] Stricker, M. A., and Orengo, M. (1995) Similarity of color images, In *Storage and Retrieval for Image and Video Databases III* (Wayne, N., and Ramesh, C. J., Eds.), pp 381-392, SPIE, San Jose, CA, USA

[32] Kotoulas, L., and Andreadis, I. (2003) Colour histogram content-based image retrieval and hardware implementation, *Circuits, Devices and Systems, IEE Proceedings - 150*, 387-393.

[33] Mindru, F., Moons, T., and Van Gool, L. (1999) Recognizing color patterns irrespective of viewpoint and illumination, In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, p 373 Vol. 371.

[34] Gevers, T., and Smeulders, A. (1997) Color based object recognition, In *Image Analysis and Processing* (Del Bimbo, A., Ed.), pp 319-326, Springer Berlin / Heidelberg.

[35] Gevers, T., and Smeulders, A. (1999) Content-based image retrieval by viewpoint-invariant color indexing, *Image and Vision Computing 17*, 475-488.

[36] Smith, J. R., and Shih-Fu, C. (1996) Automated binary texture feature sets for image retrieval, In *Conference Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp 2239-2242.

[37] Haralick, R. M., Shanmugam, K., and Dinstein, I. H. (1973) Textural Features for Image Classification, *IEEE Transactions on Systems, Man and Cybernetics 3*, 610-621.

[38] Weszka, J. S., Dyer, C. R., and Rosenfeld, A. (1976) A Comparative Study of Texture Measures for Terrain Classification, *IEEE Transactions on Systems, Man and Cybernetics SMC-6*, 269-285.

[39] Ohanian, P. P., and Dubes, R. C. (1992) Performance evaluation for four classes of textural features, *Pattern Recognition 25*, 819-833.

[40] Gotlieb, C. C., and Kreyszig, H. E. (1990) Texture descriptors based on co-occurrence matrices, *Computer Vision, Graphics, and Image Processing 51*, 70-86.

[41] Tamura, H., Mori, S., and Yamawaki, T. (1978) Textural Features Corresponding to Visual Perception, *IEEE Transactions on Systems, Man and Cybernetics 8*, 460-473.

[42] Liu, F., and Picard, R. W. (1996) Periodicity, Directionality, and Randomness: Wold Features for Image Modeling and Retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*, 722-733.

[43] Chang, T., and Kuo, C. C. J. (1993) Texture analysis and classification with tree-structured wavelet transform, *IEEE Transactions on Image Processing 2*, 429-441.

[44] Jain, A. K., and Farrokhnia, F. (1991) Unsupervised texture segmentation using Gabor filters, *Pattern Recognition 24*, 1167-1186.

[45] Ma, W. Y., and Manjunath, B. S. (1995) A comparison of wavelet transform features for texture image annotation, In *Proceedings of the 1995 International Conference on Image Processing*, p 2256, IEEE Computer Society.

[46] Do, M. N., and Vetterli, M. (2002) Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance, *IEEE Transactions on Image Processing 11*, 146-158.

[47] Kundu, A., and Chen, J.-L. (1992) Texture classification using QMF bank-based subband decomposition, *CVGIP: Graphical Models and Image Processing 54*, 369-384.

[48] Thyagarajan, K. S., Nguyen, T., and Persons, C. E. (1994) A maximum likelihood approach to texture classification using wavelet transform, In *Proceedings of the IEEE International Conference on Image Processing*, pp 640-644.

[49] Mehtre, B. M., Kankanhalli, M. S., and Wing Foon, L. (1997) Shape measures for content based image retrieval: A comparison, *Information Processing & Management 33*, 319-337.

[50] Otterloo, P. J. V. (1991) *A contour-oriented approach to shape analysis*, Prentice Hall International (UK) Ltd.

[51] Zahn, C. T., and Roskies, R. Z. (1972) Fourier Descriptors for Plane Closed Curves, *IEEE Transactions on Computers 21*, 269-281.

[52] Persoon, E., and Fu, K.-S. (1977) Shape Discrimination Using Fourier Descriptors, *IEEE Transactions on Systems, Man and Cybernetics 7*, 170-179.

[53] Rui, Y., She, A. C., and Huang, T. S. (1996) Modified Fourier Descriptors for Shape Representation - A Practical Approach, In *Proc of First International Workshop on Image Databases and Multi Media Ssearch*.

[54] Granlund, G. H. (1972) Fourier Preprocessing for Hand Print Character Recognition, *IEEE Transactions on Computers C-21*, 195-201.

[55] Zhang, D., and Lu, G. (2003) A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures, *Journal of Visual Communication and Image Representation*, 41-60.

[56] Zhang, D., and Lu, G. (2002) Shape-based image retrieval using generic Fourier descriptor, *Signal Processing: Image Communication 17*, 825-848.

[57] Chuang, G. C. H., and Kuo, C. C. J. (1996) Wavelet descriptor of planar curves: theory and applications, *IEEE Transactions on Image Processing 5*, 56-70.

[58] Kunttu, I., Lepisto, L., Rauhamaa, J., and Visa, A. (2004) Multiscale Fourier descriptor for shape-based image retrieval, In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, pp 765-768 Vol.762.

[59] Arkin, E. M., Chew, L. P., Huttenlocher, D. P., Kedem, K., and Mitchell, J. S. B. (1990) An efficiently computable metric for comparing polygonal shapes, In *Proceedings of the first*

annual ACM-SIAM symposium on Discrete algorithms, pp 129-137, Society for Industrial and Applied Mathematics, San Francisco, California, United States.

[60] Flusser, J., Suk, T., and Zitov, B. (2009) *Moments and moment invariants in pattern recognition*, J. Wiley.

[61] Hupkens, T. M., and De Clippeleir, J. (1995) Noise and intensity invariant moments, *Pattern Recognition Letters 16*, 371-376.

[62] Wang, L., and Healey, G. (1998) Using Zernike moments for the illumination and geometry invariant classification of multispectral texture, *Image Processing, IEEE Transactions on 7*, 196-203.

[63] Van Gool, L., Moons, T., and Ungureanu, D. (1996) Affine/photometric invariants for planar intensity patterns, In *Computer Vision—ECCV'96*, pp 642-651, Springer.

[64] Li, B., and Ma, S. D. (1994) On the relation between region and contour representation, In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1994*, pp 352-355.

[65] Hu, M.-K. (1962) Visual pattern recognition by moment invariants, *IRE Transactions on Information Theory 8*, 179-187.

[66] Li, Y. (1992) Reforming the theory of invariant moments for pattern recognition, *Pattern Recognition 25*, 723-730.

[67] Wong, W.-H., Siu, W.-C., and Lam, K.-M. (1995) Generation of moment invariants and their uses for character recognition, *Pattern Recognition Letters 16*, 115-123.

[68] Jin, L., and Tianxu, Z. (2004) Fast algorithm for generation of moment invariants, *Pattern Recognition 37*, 1745-1756.

[69] Abu-Mostafa, Y. S., and Psaltis, D. (1984) Recognitive Aspects of Moment Invariants, *Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-6*, 698-706.

[70] Khotanzad, A., and Hong, Y. H. (1990) Invariant image recognition by Zernike moments, *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 489-497.

[71] Belkasim, S. O., Shridhar, M., and Ahmadi, M. (1991) Pattern recognition with moment invariants: A comparative study and new results, *Pattern Recognition 24*, 1117-1138.

[72] Yang, L., and Albregtsen, F. (1996) Fast and exact computation of Cartesian geometric moments using discrete Green's theorem, *Pattern Recognition 29*, 1061-1073.

[73] Kapur, D., Lakshman, Y. N., and Saxena, T. (1995) Computing invariants using elimination methods, In *IEEE International Symposium on Computer Vision*, pp 97-102.

[74] Sluzek, A. (1995) Identification and inspection of 2-D objects using new moment-based shape descriptors, *Pattern Recognition Letters 16*, 687-697.

[75] Lei, Z., Keren, D., and Cooper, D. (1995) Computationally fast Bayesian recognition of complex objects based on mutual algebraic invariants, In *International Conference on Image Processing, 1995*, pp 635-638.

[76] Shen, D., and Ip, H. H. S. (1999) Discriminative wavelet shape descriptors for recognition of 2-D patterns, *Pattern Recognition 32*, 151-165.

[77] Flusser, J. (2000) On the independence of rotation moment invariants, *Pattern Recognition 33*, 1405-1410.

[78] Flusser, J. (2002) On the inverse problem of rotation moment invariants, *Pattern Recognition 35*, 3015-3017.

[79] Ziou, D., and Tabbone, S. (1998) Edge detection techniques - an overview, *International Journal of Pattern Recognition and Image Analysis 8*, 537-559.

[80] Basu, M. (2002) Gaussian-based edge-detection methods - A survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 32*, 252-260.

[81] Bergholm, F. (1987) Edge Focusing, *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9*, 726-741.

[82] Prewitt, J. M. S. (1970) Object Enhancement and Extraction, In *Picture Processing and Psychopictorics* (Bernice Sacks, L., and Azriel, R., Eds.), p 526, Academic Press, Inc.

[83] Lyvers, E. P., and Mitchell, O. R. (1988) Precision edge contrast and orientation estimation, *IEEE Transactions on Pattern Analysis and Machine Intelligence 10*, 927-937.

[84] Canny, J. (1986) A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*, 679-698.

[85] Ziou, D., and Koukam, A. (1995) The selection of edge detectors using local image structure, In *Proceedings of the Seventh International Conference on Tools with Artificial Intelligence*, pp 366-370.

[86] Cheng, S.-C. (2003) Content-based image retrieval using moment-preserving edge detection, *Image and Vision Computing 21*, 809-826.

[87] Kovesi, P. (2000) Phase congruency: A low-level image invariant, *Psychological Research 64*, 136-148.

[88] Lybanon, M., Lea, S. M., and Himes, S. M. (1994) Segmentation of diverse image types using opening and closing, In *Proceedings of the 12th IAPR International Conference on Pattern Recognition - Conference A: Computer Vision & Image Processing*, pp 347-351.

[89] Hansen, M. W., and Higgins, W. E. (1994) Watershed-driven relaxation labeling for image segmentation, In *Proceedings of the IEEE International Conference Image Processing*, pp 460-464.

[90] Li, X. Q., Zhao, Z. W., Cheng, H. D., Huang, C. M., and Harris, R. W. (1994) A fuzzy logic approach to image segmentation, In *Proceedings of the 12th IAPR International Conference on Pattern Recognition - Conference A: Computer Vision & Image Processing*, pp 337-341.

[91] Ma, W. Y., and Manjunath, B. S. (1997) Edge flow: A framework of boundary detection and image segmentation, In *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Proceedings*, pp 744-749.

[92] Feng, H., Castanon, D. A., and Karl, W. C. (2001) A curve evolution approach for image segmentation using adaptive flows, In *Proceedings. Eighth IEEE International Conference on Computer Vision*, pp 494-499.

[93] Comaniciu, D., and Meer, P. (1997) Robust analysis of feature spaces: color image segmentation, In *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp 750-755.

[94] Comaniciu, D., and Meer, P. (2002) Mean shift: a robust approach toward feature space analysis, *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24*, 603-619.

[95] Deng, Y., and Manjunath, B. S. (2001) Unsupervised segmentation of color-texture regions in images and video, *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 800-810.

[96] Carson, C., Belongie, S., Greenspan, H., and Malik, J. (2002) Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying, *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 1026-1038.

[97] Smith, J. R., and Chang, S.-F. (1995) Single color extraction and image query, In *Proceedings of the 1995 International Conference on Image Processing*, p 3528, IEEE Computer Society.

[98] Smith, J., and Chang, S.-F. (1996) Tools and techniques for color image retrieval, In *Storage and Retrieval for Still Image and Video Databases* (Sethi, I. K., and Jain, R., Eds.), pp 426-437.

[99] Jeong, S., Won, C. S., and Gray, R. M. (2004) Image retrieval using color histograms generated by Gauss mixture vector quantization, *Computer Vision and Image Understanding 94*, 44-66.

[100] Hadjidemetriou, E., Grossberg, M. D., and Nayar, S. K. (2004) Multiresolution histograms and their use for recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 831-847.

[101] Pass, G., and Zabih, R. (1996) Histogram refinement for content-based image retrieval, In *Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96)*, p 96, IEEE Computer Society.

[102] Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., and Zabih, R. (1997) Image Indexing Using Color Correlograms, In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, p 762, IEEE Computer Society.

[103] Deng, Y., Manjunath, B. S., Kenney, C., Moore, M. S., and Shin, H. (2001) An efficient color representation for image retrieval, *IEEE Transactions on Image Processing 10*, 140-147.

[104] Babu, G. P., Mehtre, B. M., and Kankanhalli, M. S. (1995) Color indexing for efficient image retrieval, *Multimedia Tools and Applications 1*, 327-348.

[105] Hartigan, J. A. (1975) *Clustering algorithms*, Wiley, N.Y.

[106] Chua, T. S., Tan, K.-L., and Ooi, B. C. (1997) Fast signature-based color-spatial image retrieval, In *IEEE International Conference on Multimedia Computing and Systems*, pp 362-369.

[107] Hua, K. A., Vu, K., and Oh, J.-H. (1999) SamMatch: a flexible and efficient sampling-based image retrieval technique for large image databases, In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pp 225-234, ACM, Orlando, Florida, United States.

[108] Tian, Q., Sebe, N., Lew, M. S., Loupias, E., and Huang, T. S. (2001) Image retrieval using wavelet-based salient points, *Journal of Electronic Imaging 10*, 835-849.

[109] de Bianchi, M. F., Guido, R. C., Nogueira, A. L., and Padovan, P. (2006) A wavelet-PCA approach for content-based image retrieval, In *Proceeding of the Thirty-Eighth Southeastern Symposium on System Theory (SSST '06)*, pp 439-442.

[110] Lowe, D. G. (2004) Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision 60*, 91-110.

[111] Jhanwar, N., Chaudhuri, S., Seetharaman, G., and Zavidovique, B. (2007) Content Based Image Retrieval Using Motif Cooccurence Matrix, *Image and Vision Computing 22*, 1211-1220.

[112] Mehtre, B. M., Kankanhalli, M. S., Narasimhalu, A. D., and Man, G. C. (1995) Color matching for image retrieval, *Pattern Recognition Letters 16*, 325-331.

[113] Borgefors, G. (1988) Hierarchical chamfer matching: a parametric edge matching algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence 10*, 849-865.

[114] Rubner, Y., Tomasi, C., and Guibas, L. J. (1998) A metric for distributions with applications to image databases, In *Sixth International Conference on Computer Vision*, pp 59-66.

[115] Chang, H., and Yeung, D.-Y. (2007) Kernel-based distance metric learning for content-based image retrieval, *Image and Vision Computing 25*, 695-703.

[116] Gondra, I., and Heisterkamp, D. R. (2008) Content-based image retrieval with the normalized information distance, *Computer Vision and Image Understanding 111*, 219-228.

[117] Camargo, J., Esseiva, P., Fabio, G., Wist, J., and Patiny, L. (2012) Monitoring of illicit pill distribution networks using an image collection exploration framework, *Forensic Science International 223*, 298-305.

[118] Jung, C. R., Ortiz, R. S., Limberger, R., and Mayorga, P. (2012) A new methodology for detection of counterfeit Viagra® and Cialis® tablets by image processing and statistical analysis, *Forensic Science International 216*, 92-96.

[119] Lopatka, M., and Vallat, M. (2011) Surface granularity as a discriminating feature of illicit tablets, *Forensic Science International 210*, 188-194.

[120] Lopatka, M., and van Houten, W. (2013) Automated shape annotation for illicit tablet preparations: A contour angle based classification from digital images, *Science & Justice 53*, 60-66.

[121] Mojsilovic, A., and Rogowitz, B. (2001) Capturing image semantics with low-level descriptors, In *Proceedings of the International Conference on Image Processing*, pp 18-21.

[122] Zhou, X. S., and Huang, T. S. (2000) CBIR: from low-level features to high-level semantics, In *SPIE, Image and Video Communication and Processing* (Vasudev, B., Hsing, T. R., Tescher, A. G., and Stevenson, R. L., Eds.) 1 ed., pp 426-431, SPIE, San Jose, CA, USA.

[123] Mezaris, V., Kompatsiaris, I., and Strintzis, M. G. (2003) An ontology approach to object-based image retrieval, In *Proceedings of the 2003 International Conference on Image Processing*, pp 511-514.

[124] Liu, Y., Zhang, D., Lu, G., and Ma, W.-Y. (2005) Region-Based Image Retrieval with Perceptual Colors, In *Advances in Multimedia Information Processing - PCM 2004* (Aizawa, K., Nakamura, Y., and Satoh, S. i., Eds.), pp 931-938, Springer Berlin / Heidelberg.

[125] Stanchev, P. L., Green, D., Jr., and Dimitrov, B. (2003) High level color similarity retrieval, *International Journal "Information Theories and Applications" 10*, 363-369.

[126] Wang, W., Song, Y., and Zhang, A. (2002) Semantics Retrieval by Content and Context of Image Regions, In *Proceedings of the 15th International Conference on Vision Interface*, pp 17-24.

[127] Chang, E., and Tong, S. (2001) SVM_{Active} – Support Vector Machine Active Learning for Image Retrieval, In *Proceedings of the ACM International Multimedia Conference*, pp 107-118.

[128] Shi, R., Feng, H., Chua, T.-S., and Lee, C.-H. (2004) An Adaptive Image Content Representation and Segmentation Approach to Automatic Image Annotation, In *Image and Video Retrieval* (Enser, P., Kompatsiaris, Y., O'Connor, N. E., Smeaton, A. F., and Smeulders, A. W. M., Eds.), pp 1951-1951, Springer Berlin / Heidelberg.

[129] Town, C. P., and Sinclair, D. (2001) Content based image retrieval using semantic visual categories, In *Technical Report*, Society for Manufacturing Engineers

[130] Stan, D., and Sethi, I. K. (2001) Mapping low-level image features to semantic concepts, In *Proceedings of the SPIE: Storage and Retrieval For Media Databases* (Yeung, M. M., Li, C.-S., and Lienhart, R. W., Eds.) 1 ed., pp 172-179, SPIE, San Jose, CA, USA.

[131] Zheng, X., Cai, D., He, X., Ma, W.-Y., and Lin, X. (2004) Locality preserving clustering for image database, In *Proceedings of the 12th annual ACM international conference on Multimedia*, pp 885-891, ACM, New York, NY, USA.

[132] Li, J., and Wang, J. Z. (2008) Real-Time Computerized Annotation of Pictures, *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 985-1002.

[133] Zhang, H. J. (2003) Relevance Feedback in Content-Based Image Retrieval, In *Multimedia information retrieval and management : technological fundamentals and applications* (Feng, D., Siu, W. C., and Zhang, H. J., Eds.), Springer, Berlin ; New York.

[134] Jing, F., Li, M., Zhang, H.-J., and Zhang, B. (2002) Region-based relevance feedback in image retrieval, In *IEEE International Symposium on Circuits and Systems*, pp 145-148.

[135] Jing, F., Li, M., Zhang, L., Zhang, H.-J., and Zhang, B. (2003) Learning in Region-Based Image Retrieval, In *Image and Video Retrieval* (Bakker, E., Lew, M., Huang, T., Sebe, N., and Zhou, X., Eds.), pp 199-204, Springer Berlin / Heidelberg.

[136] Zhuang, Y., Liu, X., and Pan, Y. (1999) Apply semantic template to support content-based image retrieval, In *Proceedings of the SPIE: Storage and Retrieval For Media Databases* (Yeung, M. M., Yeo, B.-L., and Bouman, C. A., Eds.) 1 ed., pp 442-449, SPIE, San Jose, CA, USA.

[137] Gross, M. H., Koch, R., Lippert, L., and Dreger, A. (1994) Multiscale image texture analysis in wavelet spaces, In *Proceedings of the IEEE International Conference on Image Processing*, pp 412-416.

[138] Higashijima, Y., Takano, S., and Niijima, K. (2008) Image Classification by Lifting Wavelet PCA, Department of Informatics, Kyushu University

[139] Murtagh, F., and Starck, J.-L. (2008) Wavelet and curvelet moments for image classification: Application to aggregate mixture grading, *Pattern Recognition Letters 29*, 1557-1564.

[140] Lee, Y.-B., Park, U., and Jain, A. K. (2010) PILL-ID: Matching and Retrieval of Drug Pill Imprint Images, In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp 2632-2635.

[141] Marquis, R., Weyermann, C., Delaporte, C., Esseiva, P., Aalberg, L., Besacier, F., Bozenko Jr, J. S., Dahlenburg, R., Kopper, C., and Zrcek, F. (2008) Drug intelligence based on MDMA tablets data: 2. Physical characteristics profiling, *Forensic Science International 178*, 34-39.

[142] Milliet, Q., Weyermann, C., and Esseiva, P. (2009) The profiling of MDMA tablets: A study of the combination of physical characteristics and organic impurities as sources of information, *Forensic Science International 187*, 58-65.

APPENDIX

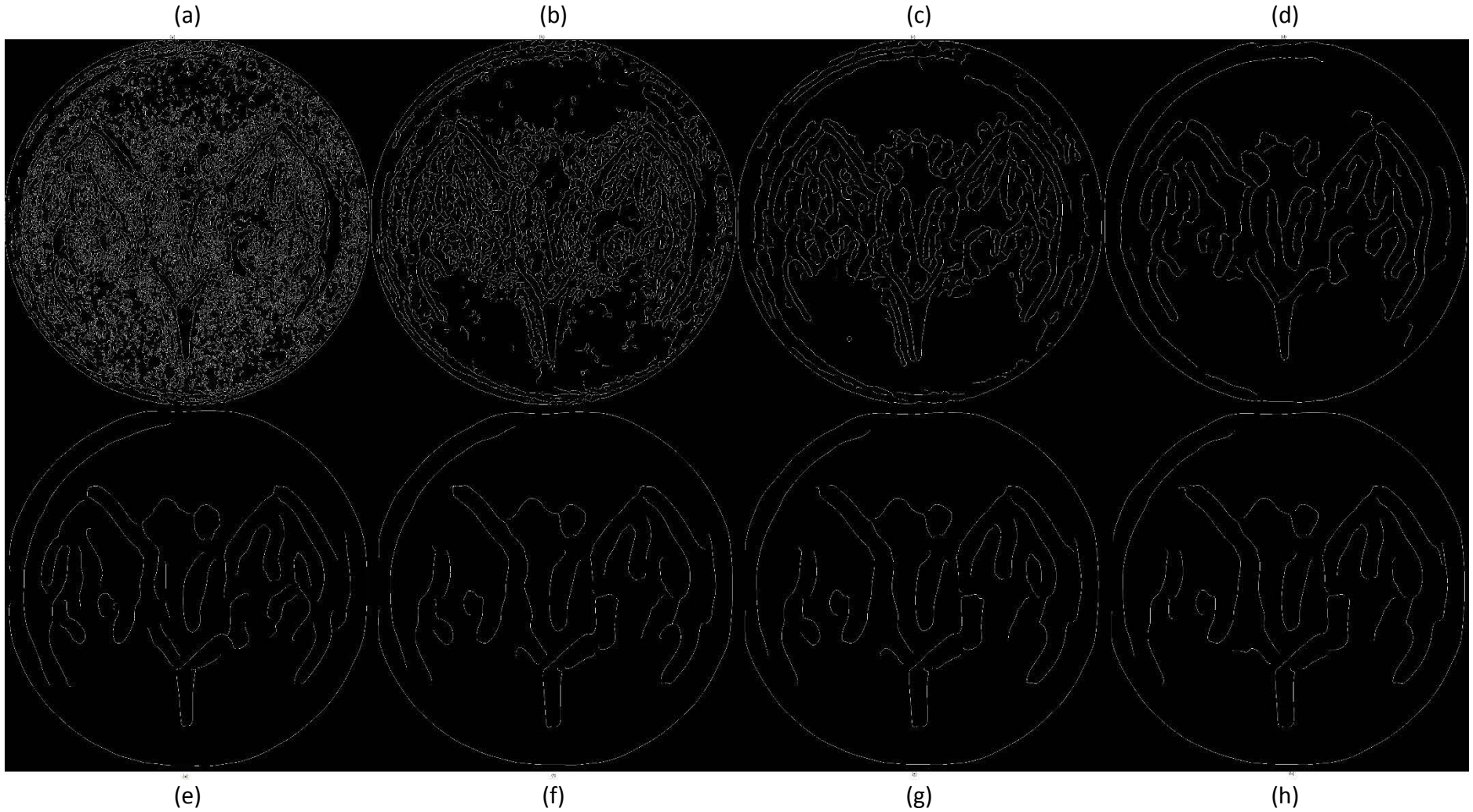


Figure 2.3 (a)-(h): "Canny" edge detection with different sigma values. Row 1 (a)-(d): Sigma = 1, 2, 4, 8; Row 2 (e)-(h): Sigma = 16, 32, 64, 128.

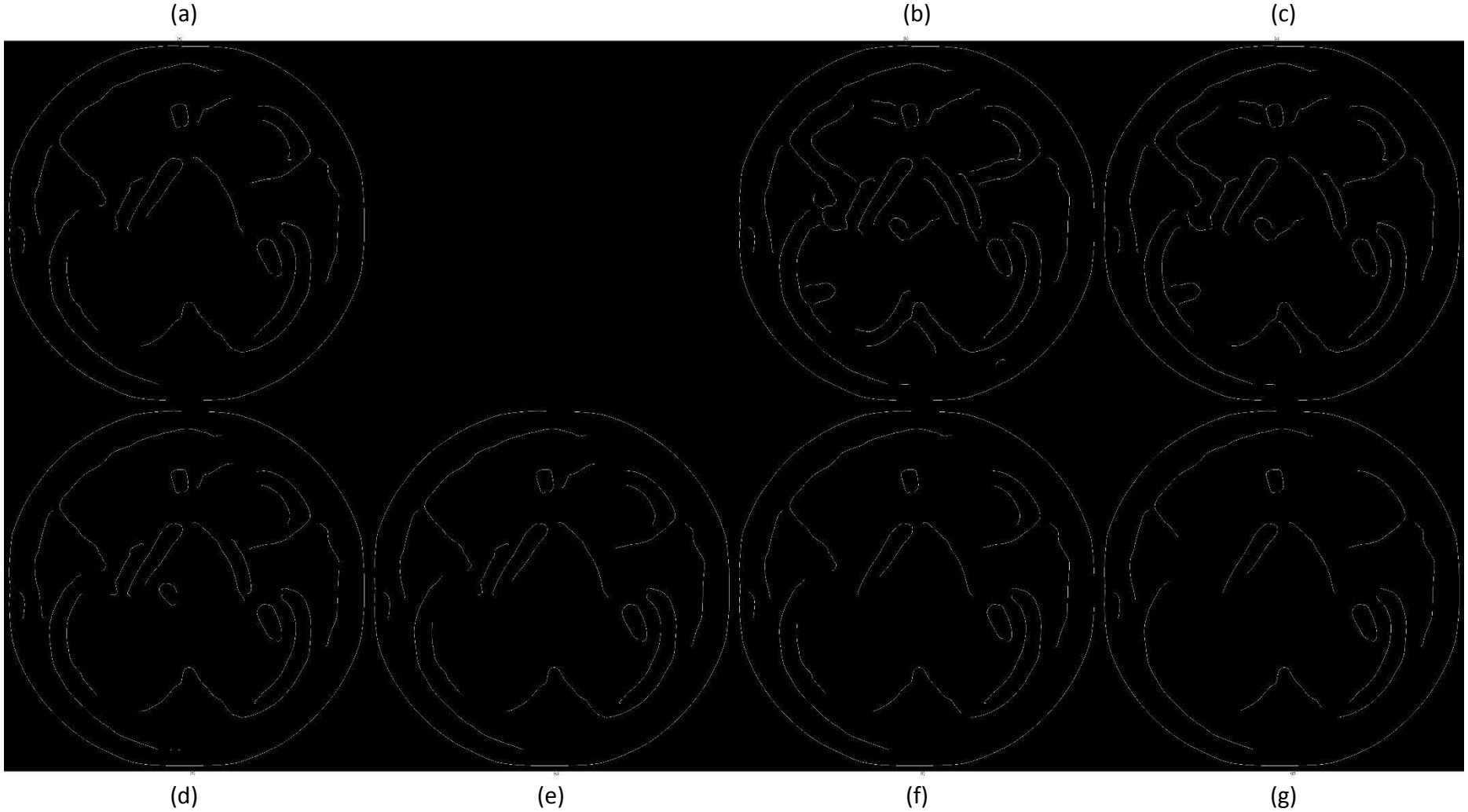


Figure 2.4 (a)-(g): "Canny" edge detection with sigma = 32 at different thresholds. Row 1 (a)-(c): Threshold = 0.1563 (auto), 0.1, 0.12; Row 2 (d)-(g): 0.14, 0.16, 0.18, 0.2.

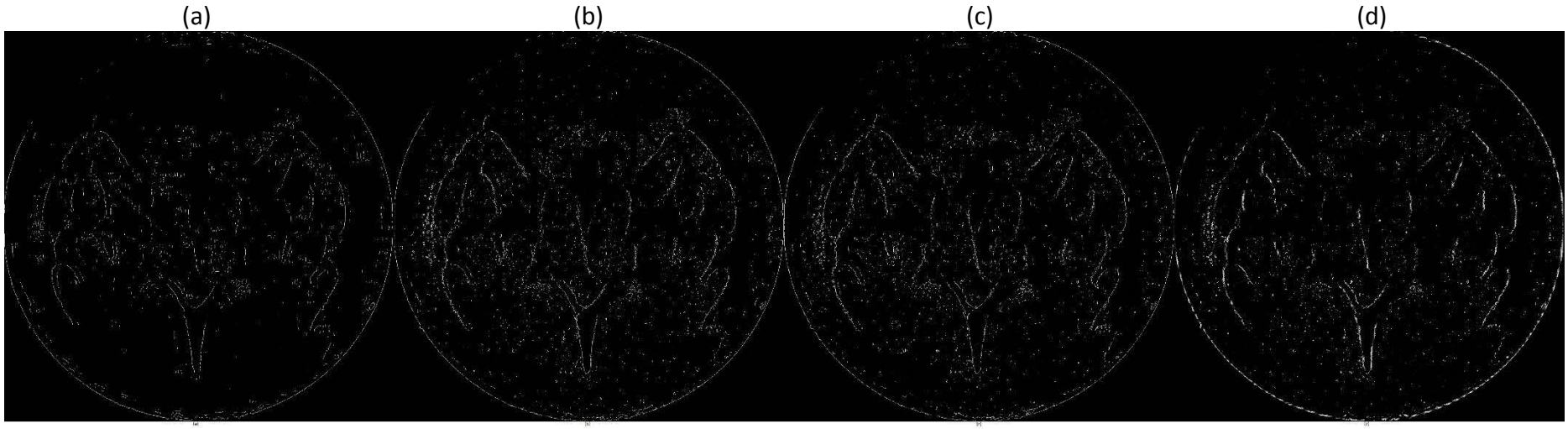


Figure 2.5 (a)-(d): An example of edge images using different filters. (a): The “Laplacian of Gaussian” filter; (b): The “Sobel” filter; (c): The “Prewitt” filter; (d): The “Roberts” filter.