Content-Based Image Retrieval Using Histograms

Georgios Gkagkaoudakis

Department of Computer Science
University of Wales College of Cardiff

May 2003

A thesis submitted for the partial fulfilment
of the requirement for the degree of Doctor of Philosophy.

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed . . . . (candidate)
Date . . . .

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by explicit references. A bibliography is appended.

Signed . . . . (candidate)
Date . . . .

Signed . . . . (supervisor)
Date . . . .

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed . . . . (candidate)
Date . . . .

Abstract

Recent technological advances in hardware have resulted in a rapid increase in the use of pictorial data. Successful utilisation of these vast volumes of information requires effective ways of retrieval. Content-Based Image Retrieval (CBIR) was proposed as a solution to this, utilising different image aspects directly for the retrieval of relevant images from a database without the need of human involvement. These aspects are typically low-level features (colour, texture, shape *etc.*) extracted automatically from images.

The major part of this thesis is related to methods for extracting image features that result in histograms. A number of novel histogram methods based on texture, shape and spatial information (and some hybrid) of images are presented. The motivation of this approach is due to the level of robustness and effectiveness of histogram methods proposed during the early stages of CBIR.

One of the main problems of histogram methods is the limitation to capture of the semantic content from images. In this thesis BEKAS is proposed as an approach to this limitation. In BEKAS histograms are utilised on local image regions and associated to semantic terms. These terms are the link to a structure that consists of a concept hierarchy and spatial templates, allowing active use of semantic (and spatial) information extracted from images to be used to query an image database.

Performance evaluation is another major issue in CBIR research, as initial attempts were tested on small scale. A framework is presented in this thesis that allowed evaluation of the presented histogram methods and their combinations. Multiple datasets are used to study the methods in isolation (e.g. tolerance to transformations), and within the context of CBIR (e.g. recall and precision of retrieval). The results are encouraging indicating significant increase in performance when multiple histograms are combined.

Acknowledgements

I wish to thank my supervisor Paul L. Rosin, for his invaluable advice, constant encouragement and support.

I wish to thank my colleagues in the vision laboratories and also those who participated the V-Lunch sessions, for the exchange of ideas throughout the duration of the course.

I am greatful to the Computer Science Department of Cardiff University for providing resources and EPSRC for funding my work.

I wish to thank my parents, relatives and friends for their support.

Contents

Chapter 1

Introduction

1.1 Motivation

Advances in electronics and information technologies progressed dramatically over the last decade. Availability of vast storage capacities and processing power is now a fact in our everyday life. This has enabled generation and storage of enormous amounts of image data in a broad range of activities. In medicine, radiologists generate millions of images every year. In remote sensing, satellites constantly photograph the earth. News photographers generate equally large quantities of photographs over the time. These vast quantities of images need to be stored and retrieved on demand. Other general applications exist that apply to a huge range of concerns. Stock control is one such application in which its users (art galleries, museums, shops, manufacturers *etc.*) might like to keep pictorial records of their products.

The benefits of keeping all this information digitally are obvious but of little use unless it is accessed in an intelligent and effective manner. Manual search is possible but becomes impractical as the number of images gets beyond a few hundred.

In early image retrieval, images are annotated by keywords. These are stored and used as retrieval keys in a traditional Relational Data Base Management System (RDBMS). Manual annotation of images suffers from two problems that render such a solution ineffective, specially as the number of images increases. The first is the effort required for the annotation of the images. The second, and more severe, is the difficulty to express the image content with a restricted number of terms. Even if the selection of annotation terms is broadened the process suffers from human subjectivity, misinterpretation is possible especially when dealing with complex images. This intensified the need to automate the process of image retrieval.

A solution to this is to quantify image similarity. A straight forward method is to measure the pixel differences between two images. Although simple to grasp and implement, this method is only useful on very specific applications (e.g. medical imaging, remote sensing *etc.*). Because this approach is very sensitive to subtle differences (e.g. due to noise or object occlusion) it is impractical for general image retrieval. Also, the computational effort required for comparing images is such that extension of the method to large image collections is impractical. This led to the search for alternative methods.

In the early 1990s Content Based Image Retrieval (CBIR) was proposed as a solution. Initial CBIR techniques are focused on the use of low level image features and queries by example. An overview of this methodology is illustrated in Figure 1.1. While new images are added to the collection their features are extracted and stored to a repository. Later, a user can do a search by providing an example image (as a sketch or a file location). Features are extracted from the example image and then compared to the stored ones. A list of the top matches is then passed on to the retrieval module, to load and display the results of the search. Multiple areas of research interest exist in the above system, such as the *query methodology*, *feature comparison* and *retrieval techniques.* Early research on CBIR was mainly focused on the extraction of features and comparison methods (distance measures) to quantify image similarity.

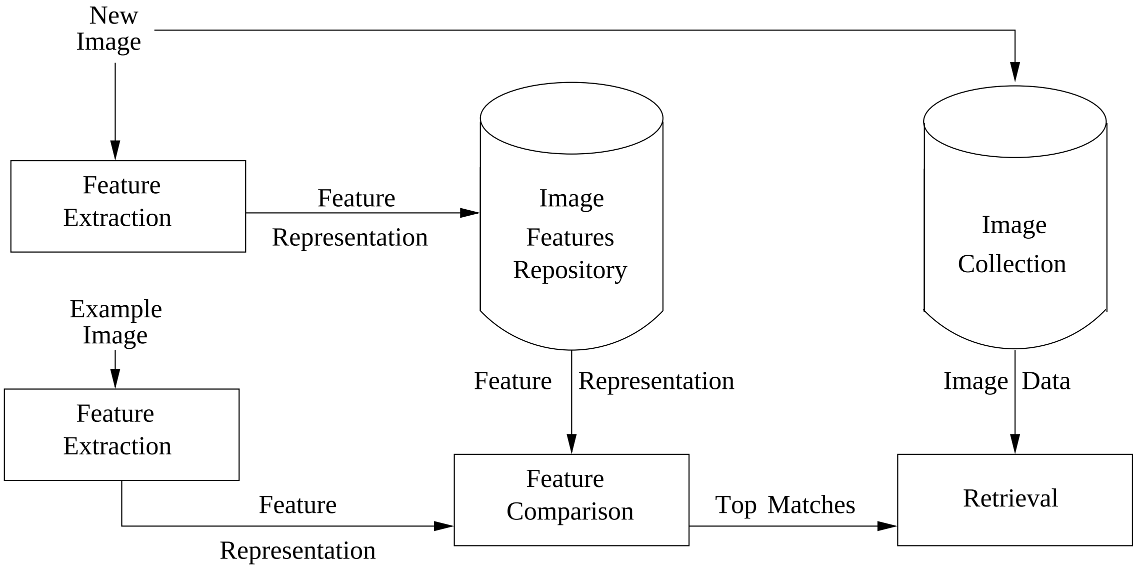


Figure 1.1: Overview of automatic CBIR by example.

Low level image features can be *domain specific*, such as biometrical features (like fingerprints, faces *etc.*), or *general* such as colour, shape, texture etc. The latter attracted attention in CBIR because of their robust behaviour over a broader range of images. One of the first features attempted in CBIR is colour and in particular the colour histogram representation [114]. The colour histogram captures the overall colour proportions of images, expected to be similar for images with similar content.

By using low level features a lot of information is ignored while enough is kept to potentially summarise the image content. This approach tends to be less sensitive to noise and transformations, while it is computationally cheaper than pixel based comparisons. For the same reason, as the size of image databases increases the discriminatory power of low level representation weakens, resulting in false retrievals. Thus, browsing and searching through semantic image categories is a more desirable and useful approach [13].

An alternative solution would be that where the system could automatically extract and identify all the objects that appear in images. Labels could then be attached and comparing two images would be a matter of comparing the attached labels, similar to the traditional keyword approach. Since the amount of information used for the comparison is minimal (compared to the size of the image) this solution can easily be extended to searching through larger collections. Such a system would require a mechanism capable of meaningful segmentation that would provide recognisable regions. Unfortunately, segmentation is as yet an unsolved problem for still images therefore such a direct approach would be unrealistic.

Although there is no single histogram feature capable to providing a desired level of CBIR functionality, we are attempting a compromise. We insist on using histogram features, mainly due to computational conveniency and robustness of these methods. A range of feature histogram methods (colour, shape, texture and hybrid) are implemented and their use in CBIR applications is studied when used independently or in combinations. Statistical analysis of the results showed a significant performance improvement over the colour histogram.

As low level histogram representations contain no explicit semantic content, it is difficult for the user of a CBIR system, based on such methods, to manage the search process. Trying to avoid this alienation of the user from the system we identified a second line of approach. We localised histogram features in an attempt to use this information to classify areas of images and attach semantic labels.

In the following sections of this chapter a brief background summary is given, related to CBIR and the context of this project. Then the project goals are introduced along with essential assumptions and limitations, followed by the contributions made by this study. The chapter concludes with the outline of the structure of the remaining thesis.

1.2 Background

The research work of this thesis falls into three different categories:

- Low level feature extraction
- The semantic gap between low-level and high-level features, and
- Performance evaluation in CBIR

1.2.1 Low level features

In [114], Swain and Ballard calculated global histograms describing the colour distribution of images. A similarity-based framework for retrieval is established by utilising histograms to describe image content. The result of this is that retrieval can be performed directly by a computer without human involvement. This approach became very popular due to its advantages:

- *Robustness.* The colour histogram is invariant to rotation of the image on the view axis, and changes in small steps when rotated otherwise or scaled [114]. It is also insensitive to changes in image and histogram resolution and occlusion.
- *Effectiveness.* There is high percentage of relevance between the query image and the extracted matching images.
- *Implementation simplicity.* The construction of the colour histogram requires a simple scan of the image, to get the colour values, and discretisation of the colour values so they fit into the resolution of the histogram.
- *Computational simplicity.* The histogram computation is only bound to the size of the image and not its actual content complexity while the computational complexity of histogram comparison depends on the resolution of the histogram.

- *Low storage requirements.* The colour histogram size can be significantly smaller than the image itself, assuming colour quantisation.

There do remain some problems with the colour histogram:

- Different images may have similar or identical colour histograms. For instance, as an extreme case, randomly scrambling the positions of pixels in an image leaves its histogram unaffected despite massive changes in the image content.
- Images taken under different ambient lighting may produce different histograms. This has been partly addressed by applying colour constancy normalisation [27].

In the intervening years many more sophisticated methods have been developed. Forsyth *et al* [34] attempted to drive the matching process by trying to identify objects (e.g. people, horses, trees). However, this is extremely difficult, requiring large systems that have specialised algorithms for identifying each type of object (e.g. tree identifiers with sub-algorithms necessary for different types of trees). Thus with the current state of the art such an approach is not general purpose or easily extensible without significant scaling problems.

More practical approaches are still rooted in low level feature extraction and description. Shape is potentially an extremely useful and powerful feature, but shape-based image systems generally run into two problems. First, they mostly require the image to be partitioned into regions from which shape descriptors can then be extracted. Unless the segmentation is directed by the user (e.g. [72]) segmentation algorithms are prone to fail, especially when new situations due to different imaging modality or object type are present [45]. Second, determining effective shape descriptors for complex natural objects is still an active area of research [29][81].

Given the inherent difficulties of methods requiring segmentation some authors have built in a spatial component into CBIR by splitting the image up based on a fixed grid. Standard analysis techniques (e.g. colour histograms) can then be performed in each cell. However, even if the cells are not regular (e.g. Stricker and Dimai's fuzzy oval [112]) this approach is still static and does not adapt to the image content.

1.2.2 Semantic Gap

In Figure 1.1, a common model used in CBIR is illustrated. Images in collections are processed and described by a set of feature representations. These are computed off-line and typically represent one or more aspects of the image e.g. colour [114][2], texture [47], shape [1][64] *etc.* When a query is submitted, distances between the example image and images in the collection are computed, using some similarity measure, applied to the feature vectors, to retrieve the top-N images.

Such a scheme is too simplified to represent human perception. This makes CBIR systems potentially unreliable for real life use - for instance two dogs may look rather dissimilar despite the fact that they belong to the same high level group. To overcome this several approaches have been proposed, roughly categorised as follows.

More feature vectors. Multiple feature representations are combined together to construct the image signature, which was proved to be beneficial [106]. In some cases more active use of spatial information in the images [112] showed further improvement.

Relevance feedback and iterative image querying. In such systems, the user directs the system and effectively the system is used as a navigation tool. Typically the user starts with a simple query or a random selection of images. The relevance of retrieved images is given to the system over a series of queries. Navigation eventually stops when the user is satisfied with the retrieved images.

Perceptual feature vectors. Two different trends have been noticed in this group. Systems using transformations of the images to a perceptually acceptable space, like in [101] and [78]. Alternatively, the feature space is partitioned with each partition representing some semantic property or attribute [21].

Regardless such improvements, such approaches typically target only one side of the problem. CBIR systems are therefore limited by the restrictions posed by the used features or similarity measures. Such systems could be improved by more active use of semantic content.

Semantic image analysis has always been attractive to CBIR community. Ultimately, semantic level features would replace the features based only on mathematical or computational convenience rather than their perceptual plausibility, and constitute a tool for efficient and perceptually accurate image retrieval.

In [57] Leow and Miikkulainen presented VISOR, a visual schema based system for object recognition and scene analysis. The emphasis is on methods for representing and learning visual schemas using neural networks and its application was simplified by simulated input. Clinque *et al* [19] extract regions of semantic importance based on the size/position of the region, while no actual association with semantic level info is used. Similarly Martinez and Serra [62] associated semantic labels to objects and experimented with hierarchical structures to access the images using semantic labels, without the need of low level features.

Tao and Grosky [118] proposed the point feature map representation and attached semantic labels to objects for retrieval. In that work the spatial relationships between objects in an image is recognised. More recently Grosky and Zhao [44] expanded the feature maps method using semantic level information. Lexical data is integrated with local and global colour features, improving system performance.

Tansley *et al,* in [115], discuss one of they key points of their system (MAVIS 2) on automatic linking of content with concepts following a two phase approach. First, meta-data available for some media object are associated with concept class descriptions. That knowledge is then used to classify media objects without meta-data. This is done using low-level feature similarities to media objects classified during the first phase.

Santini *et al*, in [99], followed a different direction based on the meaning of the image and organisation of the database. The user starts by organising the images returned by the system (initially random) and the system is trying to figure out the meaning of the image in question, which evolves over time as the user proceeds.

Activity towards semantic image analysis for CBIR is obvious, but none of the proposed approaches is balanced. Although, there are attempts to bridge the semantic gap (between high and low level features), focus is typically biased toward one side (semantic or low-level).

1.2.3 Performance Evaluation

Early performance evaluation in CBIR was often minimal and limited to show results of one or more example queries [33]. As systems evolved, and more feature methods were proposed, the CBIR community started to get more serious about performance evaluation. In [125] a survey on performance measures is conducted and in [73] guidelines for constructing performance measures are given.

The realisation of the common ground between CBIR and Information Retrieval (IR), and the advances of performance evaluation in that area (IR) was the main reason that automatic performance evaluation of CBIR systems became a requirement. Such a task in CBIR is difficult due to high subjectivity involved in evaluating similarity between images. Researchers tried to approach a solution to this problem in two different ways.

One was to exclude human factors by assigning a ground-truth to the test dataset(s). The COIL-100 [76] dataset provides instances of rotated objects and has been used extensively for performance evaluation in CBIR. In [65], an alternative automatic ground-truth method is proposed, that provides more application oriented test datasets, that is sampling images from video sequences. In both cases enough change in the image content is evident, resulting in disjoint image classes. These are used as the ground-truth to quantify the performance of feature methods, without any human intervention.

The other approach was to use human subjects to provide the ground-truth of the test dataset. Human subjects either describe images (e.g. [5]) or separate them into groups. This way human subjectivity is accounted, through the different descriptions or groupings of the same images but the process is generally slow and impractical for large datasets.

Although convenient when it comes to calculating numbers, in reality classes sometimes overlap in content making the process of automatic performance evaluation unreliable. In an attempt to overcome this, the method of subjective grouping was coupled with the grouping provided by the machine using the image features, and using learning techniques reorganise the image database, [66][67] and later in [111].

Recent work on automatic performance evaluation includes [68] where a positive and negative feedback loop is involved. The automation of the process is achieved by the relevance information extracted by the image descriptions. This approach provides a powerful tool for evaluation and comparison of CBIR systems. Though we believe that the basis of the system, the relevance of images and eventually their grouping, can be ill defined (subjective), making the process potentially unreliable.

Extensive subjective evaluation would be the ideal solution to this problem. But due to difficulties (even when it comes to narrow down the application area, subjectivity is a major difficulty [59]), time consumption and inability to be reliably repeatable, subjective evaluation should be left for the last stage of the CBIR system development.

In testing low level computer vision algorithms it often appears that an author jumps to a higher level, using the algorithm in some application context. This application essentially provides the means to quantify the performance of a proposed method. An example of such an approach appears in [89] for the assessment of polygonal approximation of curves. In CBIR this kind of approach is not really feasible. This is because CBIR comprises an application by itself. Trying to use applications for CBIR targeting evaluation promotes simplification (e.g. the scene cut detection problem can be attacked with simpler than multi-feature extraction methods) and possibly divergence from the main problem, compromising the evaluation process.

In [98], Santini mentions the three different modalities of evaluation of a perceptual and interactive system *physical performance*, *contextual* and *de-contextual evaluation.* While the focus is on the decontextual evaluation, the complexity of the evaluation process is discussed.

1.3 Project Objectives

In the pursuit to completion of this project a number of objectives were set and categorised as follows.

1.3.1 Feature Extraction

Implementation of a basic CBIR system. This step is essential so we could gain valuable knowledge on basic CBIR issues (feature extraction, distance measures *etc.*) and system architecture. This was to be achieved by implementation of a histogram based CBIR system following the architecture of Figure 1.1. The first version involved implementation of existing methods for extracting and comparing histograms, and perform queries-by-example using single histograms. The focus of the implementation was on scalability, in terms of adding more feature extraction methods, and easy acquisition of results.

Implementation of new histogram methods. Expansion of the basic system was necessary to understand and contribute to the research area. The target was to implement more histogram methods and modify the basic system to allow combinations of histograms to be used for image retrieval. This involved implementation of novel histogram methods and simple methods to combine multiple histograms for querying.

Issues regarding computational and storage efficiency (retrieval speed, histogram size) may be of great interest but are not discussed in this study.

1.3.2 Concept Oriented Image Retrieval

Implementation of a concept oriented CBIR system. Active use of semantic information is a desired direction in CBIR. In achieving this, we propose BEKAS. Low-level features are extracted in the form of histograms, and associated to semantic labels through a learning process. This way semantic labels can be attached to localised areas of new, unknown to the system, images. Using the knowledge on the spatial constraints between semantic labels in images we can achieve a link from low-level features to a concept-level hierarchy. This way the hierarchical knowledge can be extend, *e.g.* a thesaurus, to support more complex terms (like describing events or moods).

CBIR methodologies exist using content descriptors in forms other than histograms. In the scope of this thesis, however, we are using only histogram methods.

1.3.3 Testing

Establishment of a histogram based CBIR evaluation framework. Methodical and easily repeatable testing methods are a necessity to test and assess the potential of histogram methods for use in CBIR. We need to established an evaluation framework that is based on both qualitative and quantitative methods. The basis of the framework is the collection of images. These are separated into two sets *i*) controlled environment, and *ii*) case studies. Images in (*i*) are used to test individual histogram methods under basic transformations (rotation, translation and scale), and so to assess the tolerance of the methods. Case study image sets are used to test the histogram methods in the context of CBIR application, and the acquisition of quantitative performance values.

The quantitative testing methods are automatic and require no supervision. Qualitative methods are mainly based on the generation of graphs and plots, and require minimal attention, mainly for assessment.

Some of these methods may have been used by researchers but we have found no records of their use in a framework for systematic testing in the context of CBIR.

1.4 Contributions to Knowledge

This section describes the contributions to learning made by this thesis, roughly categorised into most and least important. This classification is, of course, subjective and merely represents the author's current opinion.

The author has chosen to use the pronoun "we" to represent himself in order to avoid both the jarring effect of "I" and the awkwardness of the passive tense. Nevertheless, all the original research presented here is the work of the sole author.

1.4.1 Most Important

1. We developed a number of new histograms for use in CBIR (see Chapter 3). This demonstrates the capability of histograms to be generated using a variety of image aspects, other than just colour. Therefore, enabling querying to be more responsive to the shape, texture or spatial content of images [38][39].

2. We proposed BEKAS, a concept based CBIR system (see Chapter 4). A three-level knowledge structure is proposed to allow use of information of different levels (low-level features, semantic labels, spatial arrangements, concepts *etc.*) to perform queries to image databases. At the top level of the knowledge structure a concept hierarchy is utilised, which can potentially outperform a thesaurus, as higher level terms (like events, moods *etc.*) can be used.

3. We developed a framework for assessing histogram for use in CBIR systems (see Chapter 5). The framework is composed of a series of qualitative and quantitative methods allowing easily repeatable tests to be performed.

4. We experimentally compared histogram features and their combinations (see Chapter 7). Analysis is not limited to average performance values but on more elaborate assessment, including test of statistical significance of the performance differences [38][39]. We used a variety of datasets, allowing testing of histogram methods (individual or combinations) within and outside the context of CBIR.

5. We demonstrated that combining histograms can improve CBIR performance (see Chapter 7). A peaking phenomenon starts to appear as we add more features, which

means that the performance improvement rate is reduced as more methods are combined. The peak point depends on the test dataset, and after it is reached performance reduces.

1.4.2 Least Important

1. We experimentally used histogram features to generate image database visualisation (see Chapter 5). This may be extended for use as an image database navigation tool [40][18][16][17]

2. We experimentally compared different distance measures (see Chapter 6). Histograms can be used to summarise, categorical, ordinal or angular information. A distance measure that takes that information into account may result to better performance.

3. We propose a method for automatic association of semantic labels to images (see Chapter 4). In BEKAS we propose localised histograms to associate semantic labels to images. Spatial information is extracted and can be used to improve the labelling process of new images.

4. We proposed the spatial dependency matrix (see Chapter 4). This constitutes a representation of spatial relationships of semantic labels within images. The spatial dependency matrix can be used for single images or as a template to a group of images.

1.5 Thesis Outline

The remainder of this thesis is organised as follows:

In Chapter 2, background theory related to CBIR that is useful for the subsequent chapters, is exposed in more detail. The histogram is introduced in the context of CBIR and discussed along with issues related to combining and comparing histograms. The discussion follows with a brief survey on the research literature related to CBIR. Emphasis is on principles and methods used for CBIR, establishing the state of the art. Comments about related problems or specific benefits are also stated there.

The histogram methods implemented for the CBIR prototype system is the focus of

Chapter 3. The methods are grouped by the type of feature they represent. Three main types of features are considered Colour, Shape and Texture. An attempt to use spatial information into histograms and hybrid methods, that combine more than one different types of features, are discussed.

BEKAS is introduced in Chapter 4. BEKAS is a proposed system that utilises a three-level knowledge structure to allow use of different levels of information to perform image retrieval queries. The architecture of the system is discussed with respect to training and how queries are executed. Details about implementation decisions and issues are also discussed.

In Chapter 5, a framework used to evaluate the implemented histogram methods is presented. The discussion is separated into two main parts discussing *i*) qualitative, and *ii*) quantitative evaluation methods. The exploitation of histograms for image database visualisation is also discussed in Chapter 5.

The results of experiments are presented and discussed in Chapter 6, followed by the conclusions of this thesis and discussion for future research, in Chapter 7.

Chapter 2

About Content Based Image Retrieval

In this section we list ideas, algorithms and formulae in order to establish a framework for content based image retrieval. Furthermore we establish the state of the art in areas related to the objectives of this thesis (Section 1.3). None of these are new, and where no citation is given, it indicates that the idea is either well-known or obvious.

2.1 Preliminaries

2.1.1 What is an Image

In its simplest form an image is a 2-dimensional signal of light intensity. It can be represented as a function $f(x, y)$, with the (x, y) coordinates representing the spatial location of a point. The value of $f(x, y)$ represent the the light intensity (also called grey level) of that point. Images can be classified to continuous and discrete. A continuous image is a function that takes values in a continuum, e.g. a photograph. A discrete image is a function of two variables that take values over a discrete set, e.g. an $M \times N$ integer grid. A discrete image can therefore be represented as 2-dimensional matrix $I[x, y]$. The range of the x and y values define the resolution of the image, e.g. $[0..M]$ and $[0..N]$, or just $M \times N$. A point of a discrete image is called pixel. This discrete form is preferred in the digital world to store pictures.

Images can also be multi-dimensional. A typical example is the RGB images, where each pixel is represented by a three dimensional vector where each component is mapped to a colour (*red*, *green* and *blue*). Effectively, each pixel represents a point in a 3-dimensional space which is known as the *colour space.* Other colour spaces do exist, like the *Hue-Saturation-Value* (HSV), with properties useful to particular applications.

2.1.2 What is a Histogram

A histogram is a graphic depiction of a sample of values for an attribute, usually as frequency of occurrence. In its most common form the independent variable can attain only a finite number of discrete values (bins) along the horizontal axis, while the dependent can span a continuous range along the vertical axis. A histogram H is defined formally as:

$$H = \{(h_1, h_2, \ldots, h_n) | h_i \geq 0, \; (1 \leq i \leq n), \; \sum_{i=1}^{n} h_i = N\} \tag{2.1}$$

Histograms can also be normalised so each bin represents a probability of occurrence,

$$\begin{aligned} &H' = \{(h'_1, h'_2, \ldots, h'_n) | h'_i \geq 0, \; (1 \leq i \leq n), \; \sum_{i=1}^{n} h'_i = 1\} \\ &\text{where}\, h' = \tfrac{h_i}{N}, \text{ and}\, N = \sum_{i=1}^{n} h_i \end{aligned} \tag{2.2}$$

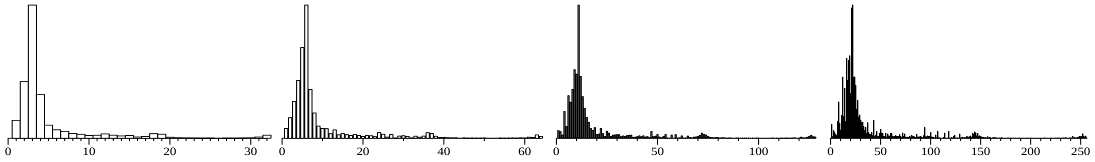
Histograms have found application in various areas such as segmentation, remote sensing and image enhancement to name a few. In the majority of these problems raw pixel values are histogrammed and then decisions are based on the shape of the histogram. The vector like nature of the histograms allows comparisons between extracted features.

2.1.3 Images, Histograms and CBIR

Content-based image retrieval can be considered as an image similarity problem. A straight forward approach would be to compare the two images directly at pixel level. Having two images A and B, of $M \times N$ dimensions, their similarity can be quantified by the summation of the Euclidean distance of their corresponding pixels.

$$D_{AB} = \sqrt{\sum_{i=1}^{M \times N} (A_i - B_i)^2} \tag{2.3}$$

This direct approach is problematic since it can be easily affected by *i*) different dimensions of images, or *ii*) slight changes at pixel level (illumination change, noise, viewpoint change etc.). Also, the computational effort involved is prohibitive, even when considering



(a) 32 bins (b) 64 bins (c) 128 bin (d) 256 bins

Figure 2.1: Histograms of different resolution.

a handful of images. In [114], Swain and Ballard proposed the use of the histograms to describe the distribution of colour in images, and use these colour histograms as the means to perform image similarity comparisons for CBIR.

In the CBIR context the histogram can be redefined as the graphical depiction of the distribution of values of a particular characteristic in an image. In [114], the colour characteristic is used with success, but it is suggested that alternative characteristics may also be used (e.g. local texture, gradient magnitude *etc.*). Therefore, we need to extend the definition of the image to accommodate these potential characteristics, to $I(x, y, f)$, with $f \in S$. The parameters x and y are the coordinates of a pixel and f is a function, from a set of functions S, that returns a value corresponding to an image characteristic at a given pixel location (x, y).

We also need to redefine the histogram for use in CBIR, as

$$\begin{aligned} &H' = \{(h'_1, h'_2, \ldots, h'_n) | h'_i \geq 0,\ (1 \leq i \leq n),\ \sum_{i=1}^{n} h'_i = 1\} \\ &\text{where}\, h' = \tfrac{h_i}{N},\ N = \sum_{i=1}^{n} h_i \\ &\text{and}\, h_i(I|O_i) = \#(I(x, y, f) | I(x, y, f) \in O_i) \end{aligned} \tag{2.4}$$

As it is not expected to always handle images of the same size (width and height), it is logical to use the probability of occurrence form of the histogram. The difference is that h_i is now a function that counts the number of pixels where the value of f is within a range of values defined by O_i.

Essentially the range of possible values of f is quantised to allow histogramming. The coarseness of this quantisation affects the resolution of the histogram (see Figure 2.1). In [122], the choice of histogram parameters, with emphasis on the resolution (bin size), is discussed by Wand.

Histograms can be classified, with respect to the histogramming function, into three different types, *i*) *nominal type* *ii*) *ordinal type*, and *iii*) *modulo type*. The bins in nominal type histograms represent different classes of values (e.g. colour references), that are not necessarily in any particular order. However, bins in ordinal type histograms are typically ordered (e.g. pixel intensities) have a start (1^{st} bin) and an end, so a distance between two bins can be defined by $d_{ij} = |i - j|$. The bins in modulo type histograms are distributed on a circle, and they used for densities of angular data (e.g. hue angles). Because of this the distance between two bins is always less or equal to π, that is $\frac{n}{2}$, (with n number of bins). For the case of modulo type histograms the distance between two bins can be defined by the function

$$d_{ij} = \begin{cases} |i-j| & \text{if} |i-j| \leq \frac{n}{2} \\ n - |i-j| & \text{otherwise} \end{cases} \tag{2.5}$$

This distinction of the histogram types is important otherwise measuring histogram similarity may return misleading results.

2.1.4 Comparing Histograms

Similarity of histograms can be quantified using *statistical tests*, as well as *vector norms* (also referred to as similarity, dissimilarity or distance measures). Note however that for statistical test based methods, non-parametric tests are required, this is due to non-availability of *a-priori* knowledge about the shape of the densities. Furthermore, within the CBIR context the significance associated to the test is not required, the statistic value itself is used. In all cases of similarity measures the comparison results are normalised to a convenient range (typically $[0 \ldots 1]$), with low values representing similar histograms and high values different ones.

Following is a non-comprehensive list of distance measures, assuming compatible histograms (of the same resolution):

L_1 *Norm.* This method is also known as histogram intersection, and is defined as:

$$D_{xy} = \sum_{i=1}^{n} |x_i - y_i| \tag{2.6}$$

L_2 *Norm.* Also known as the Euclidean distance.

$$D_{xy} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.7}$$

The above methods are not for direct use on modulo type histograms. As a counter measure Jain and Vailaya proposed to perform multiple rotations of one of the two histograms until the best match is found and used a similarity value [52].

A variant of the Euclidean distance uses the cumulative distributions. That is to reduce the effects of noise due to sparse population of the histograms. We have not found any evidence that this method could be used for modulo type histograms. It is defined as:

$$D_{xy} = \sqrt{\sum_{i=1}^{n}(\sum_{k=1}^{i} x_k - \sum_{k=1}^{i} y_k)^2} \tag{2.8}$$

Edit Distance. The edit distance is effectively the same as the earth mover's distance ([49] and [93]) when applied on 1D histograms. The similarity of two histograms is related to the work required in moving "amounts of earth", from the first histogram, to fill "holes", located in the second histogram. The edit distance is defined as

$$D_{xy} = \sum_{i=1}^{n-1} \left| \sum_{j=1}^{i}(x_j - y_j) \right| \tag{2.9}$$

For modulo type histograms, Cha *et al* proposed a different version of the edit distance in [11] and later in [12].

Statistical tests have also been used for histogram comparisons in CBIR, a short list includes:

Chi-square Test.

$$D_{xy} = \chi^2 = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i} \tag{2.10}$$

Kolmogorov-Smirnov Test.

$$D_{xy} = d_{KS}(x, y) = \max_{1 \leq i \leq n}(|\widehat{x_i} - \widehat{y_i}|) \tag{2.11}$$

Kuiper Test. This is an improvement of the Kolmogorov-Smirnov test at the tail of the histograms. This makes this method usable with modulo type histograms.

$$D_{xy} = \quad D^{-}_{xy} + D^{+}_{xy} \tag{2.12}$$

$$D^{+}_{xy=} \quad \max_{1 \leq i \leq n}(|\widehat{x_i} - \widehat{y_i}|) \tag{2.13}$$

$$D^{-}_{xy} = \quad \max_{1 \leq i \leq n}(|\widehat{y_i} - \widehat{x_i}|) \tag{2.14}$$

Pearson's Product Moment Correlation.

$$\begin{array}{l} D_{xy} = \frac{\frac{1}{n}\sum\limits_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{S_{xx}S_{yy}}} \\ \text{where}\, S_{xx} = \frac{1}{n}\sum\limits_{i=1}^{n}(x_i - \bar{x})^2 \\ \text{and}\, S_{yy} = \frac{1}{n}\sum\limits_{i=1}^{n}(y_i - \bar{y})^2 \end{array} \tag{2.15}$$

The result of the Pearson's product moment correlation test need to be mapped to the $[0 \ldots 1]$ range.

Similarity measures is an ongoing research area, and more sophisticated techniques are also considered. In [84], Puzicha *et al* non-parametric statistical tests are studied for texture classification and texture based segmentation.

Aksoy and Haralick performed comparative tests between ***probabilistic*** and *geometric* similarity measures, in [4]. They concluded that the probabilistic methods performed consistently better than the geometric methods (for normalised and non-normalised distributions).

In [110], human subjects provided information about the similarity / dissimilarity of pairs of images. The frequencies of these judgements were used to cluster the images used by the subjects, and measure an agreement factor against the default ground truth. Then they attempted to generalise the agreement using learning techniques, and use it as a similarity measure for CBIR, with promising results.

2.1.5 Combining Histograms

In [114], limitations of the colour histogram were identified. In this thesis we attempt to use multiple histograms for CBIR as a measure to limit weaknesses of individual methods. In order to achieve this we need a method to fuse the distances of individual histogram methods to a single similarity value. We have identified two widely used methods:

Weighted Sum. Each method is assigned a weight, that denotes the contribution of the individual method to the final similarity value.

$$D = \frac{\sum_{1}^{k} w_n d_n}{n} \tag{2.16}$$

The existence of the weights allows dynamic changes of method contributions, which makes the weighted sum and ideal method for interactive CBIR systems, allowing the user to modify the behaviour of the system.

Geometric Mean. An alternative method to the weighted sum, is the geometric mean. The advantage is no requirement for the discovery or tuning of any weights.

$$D = \sqrt[n]{d_1 d_2 \ldots d_n} \tag{2.17}$$

2.2 Image Features in CBIR

A wide variety of low level features have been considered for image retrieval. The range covers from general purpose CBIR, to domain specific image retrieval applications. These features can roughly be classified as follows:

- *Colour.* Including colour features with or without spatial constraints.
- *Texture.* Features used in texture analysis have also been successfully applied for CBIR.
- *Shape.* As more difficult to extract from images, shape features are used mainly for domain specific CBIR.
- *Combinations.* Multiple features combined together to form a single image index.
- *Spatial.* Use of spatial information to drive the extraction of features.

The remainder of this section follows the above classification, where a selection of research work is briefly presented. The scope of this is to help the reader form a picture about the advances on feature extraction in the area of CBIR during the last decade of its active life.

2.2.1 Colour in CBIR

Colour was first used for CBIR by [114] and ever since has been a very popular feature within the CBIR community. As a result, a plethora of CBIR methods have been developed utilising colour in various configurations. These can roughly be categorised into i) *global*, and ii) *spatial* methods. Before we move to the discussion of the methods themselves it is important to introduce (briefly) the two main parameters related to colour based methods for CBIR, the choice and quantisation of the colour space.

Issues

The common targets of CBIR methods using colour features are:

- *Perceptual similarity.* The distance between the colour values of different pixels approximates their perceptual difference.
- *Computational and storage efficiency.* The colour spaces are typically quantised to reduce the size of the feature vector. This also means lower computational effort while searching.
- *Robustness.* Changes in the image conditions (e.g. illumination variance, transformations *etc.*) should not have any effect on the accuracy of the retrieval.
- *Scalability.* The size of the image database should have little (if not at all) effect on the retrieval accuracy.

Choice of a suitable colour space. A variety of colour spaces exists, that can be classified into *machine dependent* and *human oriented*. Machine dependent colour spaces are based on the trichromatic theory (proposed by Young and Helmholtz[1]), and include the RGB, CMY *etc.* Human oriented colour spaces are based on the three human percepts of colour, that is *hue*, *saturation* and *brightness* and include the HLS, HSV, HSB, MTM, CIE*LAB, CIE*LUV *etc.* More details about colour and colour spaces can be found in [82] and [41]. Regardless of this variety of colour spaces, we have not yet found a single perceptual colour space that yields the best retrieval results.

[1] The Young-Helmholtz trichromatic theory suggests that, in our retina we have three different types of cones. Each one of these is maximally sensitive to a particular range of wavelengths, corresponding to Red, Green or Blue.

Human oriented colour spaces attracted attention as they can provide the means for perceptual similarity. A short list of studies include, Flickner *et al* and the Munsell space for QBIC [33], Pass and Zabih used the CIE*LAB space in [78], Tao and Grosky used the HSI in [118], HVC was used by Rao *et al* in [85].

The choice of colour space is also important in terms of achieving a reasonable level of robustness. As most of the methods typically use the distribution of colour to describe, they are not very sensitive to transformations and occlusion. Illumination variance remains as the main problem for robustness. In the early stages of CBIR, Swain and Ballard suggested the possibility of preprocessing the images with colour constancy algorithms [114]. In [35], Funt *et al* showed that current machine colour constancy algorithms are not really capable of increasing the robustness of the colour histogram method.

Alternative methods using illumination-independent colour representations (e.g., relative colour instead of absolute colour [32] or moment-based representations of colour histograms [48]) have been used to reduce the illumination problem and increase robustness.

Colour space quantisation. Making a successful choice of a colour space may provide the means for improved robustness but we also need to account for computational and storage efficiency. This way the usability of the CBIR system is improved. This can be achieved by quantisation of the colour space, which is reflected by a shorter feature representation.

The choice of quantisation method and resolution depends heavily on the target of the application the information is used. A short list of methods and the quantisation method used, includes:

- Keep a low number of the most significant bits for each of the colour components, in [113] and [50] they kept only 2 bits for each component of the RGB colour space.
- Fixed partitioning of the colour space, in [107] the HSV colour space is split into 166 partitions, emphasising the hue more by assigning 16 partitions, while the saturation and value had only three partitions each.
- Reference colours, in [101], Seaborn *et al* partitioned the Munsell colour space using fuzzy logic, so each partition could map colour labels as defined in [6] by Berlin and

Kay.

- Colour sets were introduced by Smith and Chang, in [107], where after partitioning of the colour space, every partition is represented by a single bit in a binary string.

Colour features

In the pioneering work of Swain and Ballard [114], the RGB colour space is quantised into $16 \times 16 \times 16$ blocks (16 levels for each colour channel) and then mapped to a histogram. The similarity of two images is quantified by comparing their corresponding histograms using the L_1 metric. The use of alternative colour spaces is studied further by Ma and Zhang in [58]. Other approaches, diverting little from the original colour histogram, were developed towards improved retrieval performance.

In [33], Faloutsos *et al* presented QBIC where the k-element colour histogram is introduced. The variable k is user defined and related to the number of bins of the colour histogram. The RGB space is partitioned into 4096 blocks (16 levels for each channel). Then, the coordinates of the centre of each block are mapped onto the Munsell colour space. Greedy minimum sum of square clustering is performed to obtain the final k-super cells, mapped onto the k-element histogram. For the comparison of the histograms they used the L_2-related metric. This way similarities between similar but not identical colour are taken into account. Later, in [93], Rubner *et al* introduced the earth mover's distance (EMD), as a metric between two distributions. The most important property of the EMD is that it can match perceptual similarity of colour better than other distance measures.

Mandal *et al* took a different approach to this matter, in [60], proposing the use of orthogonal Legendre moments to represent the histogram. The benefits of using the orthogonal Legendre moments is two-fold. First, differences between histograms due to noise have little effect, and second the computational efficiency is much improved.

A more recent attempt on global colour for CBIR is the ratio-based colour indexing by Adjeroh and Lee [2]. The basis of the method is that colour is treated using localised operations that consider the colour response at any point of an object, with respect to the colour of its neighbours. Images are transformed using these operators to the, so called, *ratio image.* The values of the ratio image pixels are used to calculated the colour ratio histogram.

While global histograms can capture the distribution of colour in images, all the spatial information is thrown away during the histogramming process. This lack of spatial information is the main reason of false positive retrievals, images of different content but very similar colour distribution are retrieved. The next step towards improving CBIR performance is to incorporate spatial information into the analysis.

Pass and Zabih [78] try to improve the performance of the colour histogram using a histogram refinement scheme named *colour coherence vector*. Each histogram bin is essentially subdivided into groups based on a coherence measure, injecting spatial information into the histogram. Coherent pixels are part of sizable continuous region, while incoherent are not. The concept of refinement is then extended to subdivide the bins of the histogram using additional features. In both cases histogram refinement resulted to improved performance.

The colour correlogram is introduced by Huang *et al* in [50]. Spatial information is extracted in the form of local spatial correlation of colour. The colour correlogram captures the change of spatial correlation of colour pairs over a distance d. It is noticed, that the choice of d affects the performance of the method in terms of retrieval accuracy and computational complexity. With respect to this they propose adaptive choice of d, with smaller values to be preferred. This way computational complexity remains low, while the more important local spatial correlation of colour is captured.

Rao *et al,* proposed the geometric histogram in [85] and then in [86]. By this term basically they established a framework where geometric configurations of pixels are used, along with their colour properties, to generate histograms. Each geometric configuration essentially represents a spatial relationship between the pixels involved. For instance, a configuration of a single pixel would give the same results as the colour histogram. The focus is on configurations of a sequence of isosceles right triangles of different side lengths.

2.2.2 Shape in CBIR

Shape methods in CBIR can be classified into two distinct categories, with respect to their input, *i*) boundary, and *ii*) region methods. Boundary methods work on shape contour extracted from images, ignoring whatever is enclosed by the boundaries. On the other hand, region based methods take internal (to the boundary) properties into account (e.g. holes, pixel colours).

Shape methods can be further classified into two categories, with respect to the methodology, *i*) measurement, and *ii*) transformation based methods. Measurement based methods range from simple calculations of shape primitives (*e.g.* area, circularity *etc.*) to more sophisticated methods, such as moment invariants. Transformation based methods range from functional transformations, such as Fourier descriptors, to structural transformations, such as chain codes.

A small sample of shape methods (falling into the above categories) follows. This list is by no means complete, but is long enough to demonstrate the trend of using shape in CBIR. A survey on shape matching is presented by Veltkamp and Hagedoorn in [121], while comparative studies on shape methods in the context of CBIR have been published by Mehtre *et al* [64], and later by Safar *et al* [95].

Paulik *et al* in [80] introduced a *spatially variant circular autoregressive* (SVCAR) model for the analysis and classification of closed shape boundaries. The parameters of the SVCAR model are used directly as a feature vector for shape matching. The method is invariant to object scale, rotation and translation, while an optimal feature matching algorithm is proposed to compensate for parameter changes due to different starting points. Application of the SVCAR model in CBIR may be possible, though the requirement for closed curves may be a limiting factor.

In [23], Daudi and Matusiak attempted CBIR using the multiscale description of user sketches. The curvature scale space (CSS) description is extracted from closed boundaries over multiple scales. These boundaries are then described and compared by the location of CSS contour maxima. Strong properties that allow this method to be used in CBIR include: tolerance to translation, rotation and scale, and robustness under small distortions. Again, the requirement for closed curves may be a limiting factor for general CBIR.

Berretti *et al* [7], proposed a shape similarity method based on local shape descriptors for CBIR. Shapes are partitioned into tokens at the points of minima of the curvature function. A set of perceptually salient attributes are used to model each token. The tokens are arranged into an M-Tree index structure to allow efficient comparisons. Similarity between two shapes is quantified by the use of token and shape measures modelling perceptual similarity. The method is not limited to closed contours, and allows partial matching of shapes. Experimental results showed tolerance to slight shape differences and partial occlusions.

Kim *et al* proposed a method invariant to translation, rotation and scale for similarity based image retrieval, in [55]. A modified form of the Zernike moment descriptors applied in such a way to take the perceptual qualities of the shape into account. The modification is based on the separation of the shape into inner and outer regions, with respect to the distance of pixels from the centre of mass of the shape. Zernike moment descriptors are calculated as normal for the two regions. During shape comparison the perceptual importance of the outer form of shape is taken into account by assigning a greater weight to the similarity of the outer Zernike moment descriptors. The way shape is treated by this method suggests suitability for single object images (that makes it suitable for applications such as trademark retrieval).

The *image shape spectrum* (ISS) is introduced by Nastar in [75]. Images are first transformed to the XYI space (X and Y the location of a pixel and I its intensity). Then the local shape of the intensity surface is calculated (called *image shape index*) for each pixel of the image as a function of the principle curvatures. The ISS is then generated by histogramming the image shape index values. ISS is fully invariant to in-plane translation and rotation of the image. That is due to use of the principal curvatures and the nature of the image. Application of the ISS in image retrieval is demonstrated in [75], while successful use of ISS in face recognition is demonstrated in [74]. Experimentation was performed on collections of relatively simple images (single objects) and so, performance on more realistic images is not known.

Local principal curvatures have also been used by Ravela and Manmatha in [87]. To compensate for the loss of spatial orientation information, when constructing histograms of curvature only, they also histogram the local phase. Calculating local curvature and phase at multiple scale improved the performance of the method. Experiments were conducted using a broader range than in [75] and [74], supporting the capability of the methods for more general CBIR.

In [118], Tao and Grosky proposed the point feature map representation. Images are partitioned to $M \times N$ non overlapping windows. Each window is then abstracted as a unique point in terms of its *spatial location*, *dominant hue* and *dominant saturation*. A Delaunay triangulation is then generated using groups of points with the same hue or saturation. The angles of the generated triangles are histogrammed over all the possible groups, resulting in the feature point histogram. Experiments showed promising results mainly addressed to the incorporated spatial information.

In [102] Sebastian *et al* attempt matching of shape outlines. Their method relies on finding an optimal correspondence between two curves. That is achieved by comparing the length and curvature of the curves. The notion of the alignment curve is introduced in the context of ensuring a symmetric treatment of the curves been matched. The basic assumption is that the goodness of the optimal correspondence can be expressed as the summation of goodness of matching subsegments. This allowed the authors to treat the search for the optimal correspondence as an energy minimisation problem, allowing them to use an efficient dynamic-programming algorithm to find a solution. Application in CBIR may be possible after some of the shortcomings of the method are addressed. Specifically, sensitivity to occlusion and spatial arrangement.

In [71], Mori *et al* propose the *shape context*. First edges are extracted from the image, and number of points are selected (roughly uniformly sampled) from the contours. Then the distribution of the distances to the remaining points is calculated, which constitutes the shape context. Shapes are matched by means of calculating the cost of finding the best correspondences of points between two shapes. Experiments were performed with image datasets and synthetic data, concluding that performance is robust under occlusion and intraclass variation.

2.2.3 Texture in CBIR

In a digital image, texture is defined by the spatial interrelationships between (and/or spatial arrangement of) the image pixels. Texture feature extraction can be classified into, *i*) statistical, and *ii*) structural methods [42]. In statistical approaches texture is modelled by statistical distributions in the spatial or in a transform (e.g. Fourier, wavelets *etc.*) domain. Structural methods are based on the assumption that texture is composed of primitives with different properties appearing in particular arrangements.

The most popular set of texture features are based on Haralick's cooccurrence matrix of gray levels [46]. Texture features calculated from the cooccurrence matrix have been extensively used for texture classification, while in [3] such features are used for CBIR with success.

In [61], Marie-Julie and Essafi use the fractal transform for CBIR. Images are compressed using a fractal compression. This is done by applying the fractal compression in windows, so a collection of local transformations is obtained. The similarity between two images is calculated by comparing the parameters of the local transformations. The main attraction of the method is that the local transformation parameters can be used to reconstruct an approximate version of the original image. A multi-scale version is also proposed, that allows (by multiple sets of local transformations) a richer signature, allowing some degree of scale and rotation tolerance.

In [24], Dimai proposes a rotation invariant descriptor for image retrieval. The method is a two phase process. First, Gabor filters are used to extract non-invariant texture descriptor. Then, using the framework of the general moment invariants, a rotation invariant descriptor is extracted. The rotation invariant feature was tested using a variety of datasets, and was found to perform better than the non-invariant version.

2.2.4 Spatial Information

Regardless to how effective is the use of the colour histogram for CBIR, any spatial information is ignored when only the distribution of colour is considered. Methods for combining colour along with spatial information can be separated into two categories, base on:

- *Predefined partitioning.* Images are partitioned using a fixed configuration. and colour properties are calculated for each partition.
- *Segmentation.* Images are segmented and the resulting regions are used to generate, typically, a graph representation that includes the colour properties of each region.

In [103], Sethi *et al* present Color-WISE, where fixed partitioning of 8×8 overlapping windows is used. For each partition a histogram is calculated for the hue, saturation and intensity. The histograms are then reduced in size by finding the peak areas. This reduction also acts as a filter to reduce the effects of noise, due to small amounts of isolated colour.

Matas *et al* introduced the colour adjacency graph (CAG) in [63]. Each node in the graph represents a single colour component. That is defined by a set of pixels forming a unimodal cluster in the chromatic histogram. Edges of the CAG represent the adjacency attributes of the colour components. The CAG therefore includes information, not only about the modes of the colour histogram, but also spatial information about the image.

In [105], Sinclair is also using colour regions to form a graph for CBIR. The main focus is on the segmentation of the image. Starting from a set of more reliable edges, they propose a multiscale edge detection. A series of Voronoi peaks are generated from the edge image. Pixels are then grouped by the distance to the closest peak. The edges act as region limits, so pixels are not assigned to a Voronoi peak past an edge. A merging step is then used to join regions not separated by an edge. Each region is then used as a node in a graph, with the mean colour as an attribute. Although, this method is presented for CBIR no evidence of its application is shown.

In [10], Carson *et al* present the *blobworld* representation, for region based image retrieval. A colour and texture descriptor is calculated for each pixel, constited of six values (3 for colour and 3 for texture). Each descriptor is treated as a point (6D vector) in a 6D space. The EM algorithm is then used to cluster these points. The resulting clustering is used to label the pixels of the image where regions are expected to be formed. Spatial (the centroid) and shape (scatter matrix) properties are extracted from each region and used (along with colour and texture descriptors) as the signature of the image. Comparing the images is just a matter of comparing their corresponding blobs.

2.2.5 Summary

The pioneering study of Swain and Ballard kick started the research interest on CBIR. Colour has been the most popular approach, that is mainly due to simple computational requirements and the easier understanding of its properties. Soon it was realised that other features (shape, texture *etc.*) may be of equal importance. This is why research headed towards adopting more and more methods from the image processing and machine vision literature, in an attempt to improve CBIR performance over the colour histogram.

The most common approach to this is combining multiple low-level features. This way authors attempt to reduce the limitations of one feature by combination with another feature that overcomes such limitations (transformation, occlusion *etc*). In some methods, features are extracted independently of each other, and they are combined using methods similar to those mentioned in section 2.1.5. Pass and Zabih for instance, in [79], attempt four combinations of *colour*, *edge density*, *texturedness*, *gradient magnitude* and *rank* features. On the other hand, multiple features can be combined implicitly, and treated as a single signature. The geometric histogram [86] can be considered such a combination of features exploiting of colour and texture.

Low-level features have been used successfully in CBIR applications. Regardless to this, they still fail to capture true semantic content. Queries, such as "*get me images of*

Everest" can not be served by systems based purely on low-level features. This inability directed the research closer to more semantic sensitive systems.

2.3 Semantic CBIR

Human perception can not easily be represented by a simple scheme such as the one presented in Section 1.1. To overcome this several techniques evolved in the research community of CBIR, roughly categorised as follows: *i*) using more feature vectors [79][106][38], *ii*) utilising relevance feedback techniques [22][119][94], or *iii*) perceptual feature vectors [101][78][21].

Regardless to the achieved improvements, no semantic knowledge is extracted or used explicitly by such methods. Semantic image analysis is a very active area of research in CBIR. Ultimately, semantic level features would replace the features based only on mathematical or computational convenience rather than their perceptual plausibility and constitute a tool for efficient and perceptually accurate image retrieval. Current advances in CBIR include methods towards that direction and a small sample of that follows.

In [57] Leow and Miikkulainen describe VISOR, a visual schema based system for object recognition and scene analysis. The main focus of that work is on methods for representing and learning visual schemas in neural networks. The approach was simplified by using simulated input. Clinque *et al* [19] extract regions of semantic importance based on the size/position of the region. Though there is no actual association with semantic level info, assuming that size and position of a region are enough for image retrieval. Similarly Martinez and Serra [62] associated semantic labels to objects and experimented with hierarchical structures to access the images using semantic labels, without the need of low level features. Although they assume that all of their images have at least one focus of attention (potential object) they avoid images that have many areas of attention as it is not realistic for their system to analyse them. In all cases accurate and complete object extraction from the input images was avoided, an indication that a solution is not yet possible.

Tao and Grosky [117] proposed the point feature map representation and attached semantic labels to objects for retrieval. In that work another aspect in image retrieval is addressed, the spatial relationships between the objects in the image. However, they assumed that all the objects in the image are extracted, which implies a perfect segmentation that is not realistic yet. In another piece of work Grosky and Zhao [44] expanded the feature maps method using semantic level information. They integrated lexical data along with local and global colour features. This resulted in improved system performance. However, the lexical information was passively used, no extra information about the relationships of different labels was used, and the annotation of the images was manual.

Dupplaw and Lewis, in [30], attempt CBIR using scale space object trees. Their method is based on the sieve. Images are decomposed through scale space using their morphology to form a tree. Each, node represents a region of the image that was removed at each scale of the sieve. Trees generated from real-life images are far too complicated for direct application to CBIR. A simplification of the tree is possible (and still under research) by merging similar sub-trees or eliminating redandunt sub-trees. Sub-trees in scale space may represent objects. With respect to CBIR, matching can be performed in two phases. First, graph matching enables identification of similar objects, then further matching based on low level features can take place for more specific object matching.

Tansley *et al,* in [116], discuss one of the key points of their system (MAVIS 2) on automatic linking of content with concepts following a two phase approach. In phase one they associate meta-data available for some media object with concept class descriptions. Then, in phase two, they use the above knowledge to classify media objects without meta-data through their similarity to media objects classified in phase one. The actual classification of the un-tagged images relies on a system that uses low level features only.

Santini *et al*, in [99], followed a different direction based on the meaning of the image and organisation of the database. The user starts by organising the images returned by the system (initially random) and the system tries to figure out the meaning of the image in question, which evolves over time as the user proceeds.

Activity towards semantic image analysis for CBIR is obvious, though it seems that most of the proposed approaches address only part of the problem, ignoring essential issues.

2.4 Performance Evaluation in CBIR

Standard methods from the area of document retrieval have also been used in performance evaluation in CBIR. The most common are the pair of *recall* R and *precision* P. Applied to the CBIR context, suppose an image database contains N_c relevant images. Given a query, B is the set of retrieved images, and A the set of relevant retrieved images ($A \subset B$). The above mentioned measures are defined as follows:

- *Precision.* Measures the ability of the system to present only the relevant items from the database.

$$P = \frac{\#\{A\}}{\#\{B\}} = \frac{number\,of\,relevant\,images\,retrieved}{number\,of\,retrieved\,images} \tag{2.18}$$

- *Recall.* Measures the ability of the system to retrieve all the relevant images in the database.

$$R = \frac{\#\{A\}}{N_c} = \frac{number\,of\,relevant\,images\,retrieved}{number\,of\,relevant\,images} \tag{2.19}$$

A common representation for precision and recall is the *recall-precision graph*, in which precision values are plotted against recall values.

Although these of methods allow automatic performance evaluation were available, for the first CBIR systems testing was often minimal and limited to results of a number of example queries [33]. As the community matured performance evaluation became a more serious issue. In [125] a survey on performance measures is conducted and in [73] guidelines for constructing performance measures are given.

The advances in performance evaluation in Information Retrieval (IR) and the realisation of the common ground with CBIR, triggered the necessity for more elaborate and automatic performance evaluation of CBIR systems. While in theory such methods may be applicable almost without change, in practice it is not as easy. That is mainly due to human subjectivity factors involved in the whole process (similarity between two images is conceived differently by different people).

Researchers tried to approach a solution to this problem in two different ways.

- *Fully automatic approach.* Human factors are excluded from the process, by the use of a carefully selected datasets. The COIL-100 [76] dataset provides instances of 100 rotated objects and has been used extensively for performance evaluation in CBIR. In [65], an alternative automatic ground-truth method is proposed, that provides more “real life” oriented test datasets through sampling images from video sequences. In both cases enough change in the image content was evident to test the feature methods, and disjoint image classes that were used as the ground-truth.

- *One-off use of human subjects.* Human subjects provide the ground-truth of the dataset. This is achieved usually by *i*) manual grouping of images, or *ii*) image descriptions. In the first case, the resulting grouping is based purely on semantic reasoning, although, there are potential bottlenecks when more than one human subjects are contributing to the grouping. The alternative approach is to assign descriptions to the images independently (e.g. [5]). A set of predefined terms is used (similarly to text annotation), and multiple terms are assigned to images. Grouping can then be achieved by means of common terms.

In both cases, human subjectivity is taken into account. On the downside the lack of scalability of the methods, specially as the datasets increase in size, makes the process of groundtruthing potentially impractical.

An alternative method to increasing scalability could be to apply and test CBIR methods on more specific domains of image retrieval. This approach often appears in computer vision, an example of such approach appears in [89] for the assessment of polygonal approximation of curves. Unfortunately, CBIR comprises an application by itself, and there is not much (if any) space of elevating the methodologies to a higher level. On the contrary, such approaches may oversimplify (e.g. use of CBIR methods for scene cut detection, a problem that can be approached with simpler than multi-feature extraction methods) and possibly diverge from the main problem of CBIR.

The difficulties posed against subjective evaluation (even when it comes to narrow down the application area, subjectivity is a major difficulty [59]) led researchers to prefer automatic (or semi-automatic) methods.

In [25], Dimai presents a new evaluation measure, based on rank-difference trend analysis. A target set (or more) need to be established first (that can be obtained by human subjects). The definition of the the target set depends on the aspect that is targeted by the evaluation. Two definitions are presented *i*) Stability (under transformation or noise), and *ii*) Discrimination (how relevant are the retrieved images). Rankings obtained by the CBIR system (or methods) are then compared against the target by means of statistical trend analysis. The methodology is shown to be scalable allowing easily repeatable testing. That is mainly due to the reusability of the target sets.

Current work on automatic performance evaluation includes [68] where a positive/negative feedback loop is involved. The automation of the process is achieved by the relevance information extracted by the image descriptions. This approach provides a powerful tool for evaluation and comparison of CBIR systems. However, we believe that the basis of the system, the relevance of images and eventually their grouping, can be ill defined, making the process potentially unreliable, mainly due to the subjectivity of the image description process.

A more recent attempt to improve performance evaluation of CBIR is *The Benchathlon Network* (http://www.benchathlon.net/), which started as an idea of a contest for CBIR systems. The idea is based on a common dataset and researchers to provide the mechanism to extract and compare image features. A number of different performance measures (mainly based on recall and precision) are used for performance evaluation.

While performance evaluation in CBIR has attracted some attention, it remains a difficult and still an unsolved problem, a recent survey is presented by Muller *et al* in [69].

Chapter 3

Feature Histograms for CBIR

Introduction

The main components of a histogram based CBIR system are *i*) the feature extraction and histogramming mechanism, and *ii*) the similarity measure (or measures). Both parts share similar importance in the performance of the final system. In this chapter we discuss a number of histogram methods focusing on the feature extraction and the histogramming process. This is related towards the completion of the first project objective (Section 1.3.1), in particular, part of understanding of CBIR methods. The second part of the primary objective, related to new histogram methods for CBIR, is also addressed, as a number of original features (covering a broad range of image aspects) are implemented.

The discussion of each method starts with a brief description of the idea(s) it is based on. Then the process of feature extraction is described, followed by notes that we have identified regarding the *pros* and *cons* of the method. Finally the discussion concludes with the illustration of histograms extracted by three test images and any conclusion drawn after a very brief visual inspection of these histograms.

The choice of the three images is such that, two of them are similar, picturing the same scene from slightly different angle, while the third pictures a completely different scene. Using these images as a rule we try to identify issues on the discriminatory attributes of each method.

The implemented methods are organised with respect to the main feature they are based on. More specifically, we start with colour based histograms. Then in Section 3.2, the shape histograms are discussed, followed by the texture histograms, Section 3.3. The discussion on the major features concludes with an attempt to incorporate spatial information into histograms, Section 3.4. Discussion continues with methods combining multiple features (*e.g.* colour and texture) into single histograms, Section 3.5. The discussion on histogram methods finishes with some hybrid methods, Section 3.6. In this Chapter we do not discuss details on combining the implemented histograms.

All the methods discussed in this chapter are based on established techniques applied in various areas in computer vision. Some of the methods (Sections 3.1.1, 3.1.2, 3.3.1) have already been implemented and used in CBIR systems. Their inclusion in this chapter is in the basis of completeness, regarding the features used in the prototype CBIR system. The methods discussed in Sections 3.2.4 and 3.3.2 have been used in CBIR, but not in the form of our implementation, while the remaining methods (Sections 3.2.1, 3.2.2, 3.2.3, 3.2.5, 3.2.6, 3.4, 3.5.1, 3.5.2, 3.5.3, 3.6.1, 3.6.2) have not been implemented for use in CBIR. These variant and original methods to CBIR comprise the main contribution of this chapter.

3.1 Colour

Colour, been the most popular feature in CBIR, was the first implemented for the prototype systems. Two different approaches were considered *a*) diverting little from the original approach of Swain and Ballard, based on opponent colour space [114], we implemented the *hue* histogram, and *b*) based on the work of [101] we experimented with the *colour labels* histogram. The main reason for experimenting so briefly with colour features is mainly the popularity of colour features in CBIR. This fact encouraged us to experiment more with other image features.

3.1.1 Hue Histogram

The basis of this method is the *Hue-Saturation-Luminosity* (HSL) colour space. Colours specified in the HSL coordinates are thought to correspond more closely to human colour intuition than spaces such as the RGB. In HSL one vector component is used to specify the luminance or lightness, and two, hue and saturation, to specify the chrominance. Hue is expressed by an angle and saturation by a fraction. The HSL space can be imagined as a cone where the *height* represents the luminosity, the distance from the centre the *saturation* and the angle around the vertical axis represents the *hue*. A partition of a HSL cone is illustrated in Figure 3.1.

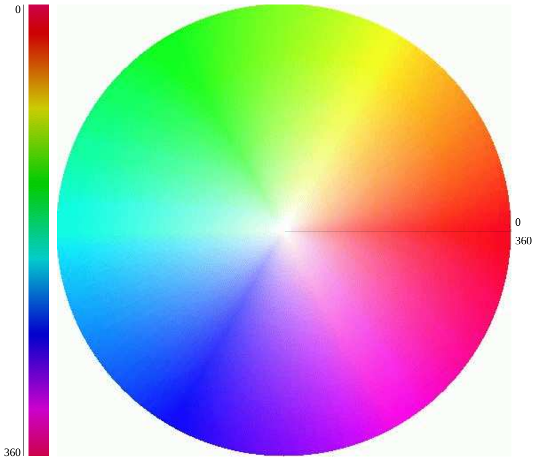


Figure 3.1: Portion of the HSL cone and a Hue stripe at a Saturation=1 and Luminosity=1.

The main initiative of using this method is the simplicity of extracting the colour information from the image. After transformation to the HSL colour space, just the hue component is retained. Then the hue values are used directly to generate the histogram representation of colour.

Advantages

- Colour representation is closer to human intuition than RGB.
- Keeping only the hue result in a single dimensional histogram representation. The resolution of the histogram can then be adjusted to meet any storage requirements. For our experiments we used hue histograms of 256 bins.

Disadvantages

- The grayscale is not represented by the hue histogram. That could easily be solved by incorporating quantised intensity values into the histograms. Regardless of this fact, we experimented without incorporating the grayscale information into the histogram.
- The hue histogram is not illumination invariant. At the same time, most of the colour based methods are not illumination invariant. Colour constancy may be a solution to this and constitutes a research area by itself.

- Still, images with different content may have similar hue histograms. Since no extra information is used but colour we expect this problem to be apparent.

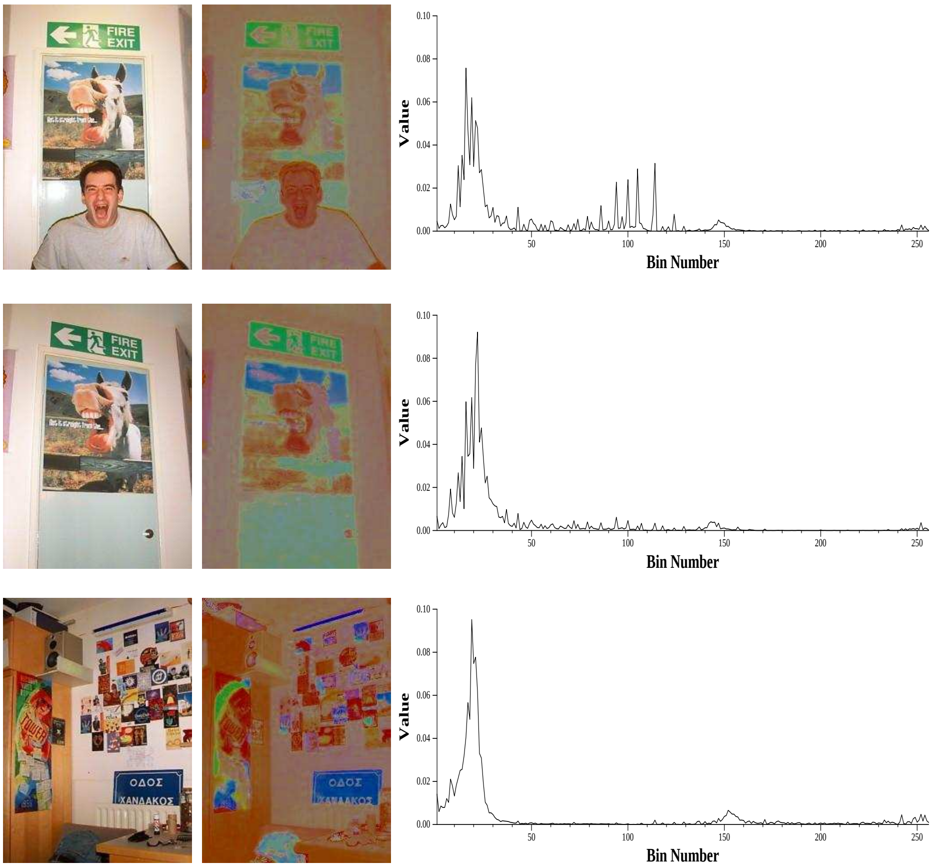
While each bin of the histogram represents a different range of hues, the actual perceptual value of adjacent bins is similar. Therefore, the choice of distance measure should be such that it takes that fact into account. For example, the *earth mover's distance* [93] may be more appropriate than the Euclidean. Furthermore, since the the bins represent angular values, one can say that the circularity of the histogram should be taken into account. Because of potential illumination variance between images we do not expect linear distortion, so a simple histogram shift would not solve the problem. The root of the problem is colour constancy which is out of the scope of this work.

Examples of hue histograms are illustrated in Figure 3.2. Overall, the shape of all the three histograms could be considered similar. The shape of the first histogram is not as smooth, differing to the second histogram. This is due to the appearance of the human figure. Closer visual inspection showed that the top two histograms share more similarities than the third one. Notice the shape of the large peak and the range of populated bins.

3.1.2 Colour Labels

Berlin and Kay studied colour terms in 100 different languages [6] (see [101] for a recent application). The result of that anthropological study was that even over a wide range of cultural backgrounds the colour space could be partitioned into a maximum of 11 "universal" categories. These can be subdivided into achromatic (Black, White, Gray) and chromatic (Red, Green, Blue, Purple, Orange, Pink and Brown) labels, distributed on the Munsell space, shown in the following Figure 3.3. For the generation of the colour labels histogram, the RGB pixel values of the image are mapped to the 11 Berlin and Kay colour terms and then histogrammed.

The process of mapping from RGB to Berlin and Kay colour terms is not straight forward. That is mainly due to uncertain areas of the Munsell space which could easily fall into more than one colour term. In the scope of our work we restricted the label mapping to crisp values, using a translation map provided by the authors of [101], where the mapping scheme is further refined by the use of fuzzy logic.



(a) RGB (b) Hue (c) Histogram

Figure 3.2: Hue histogram examples.

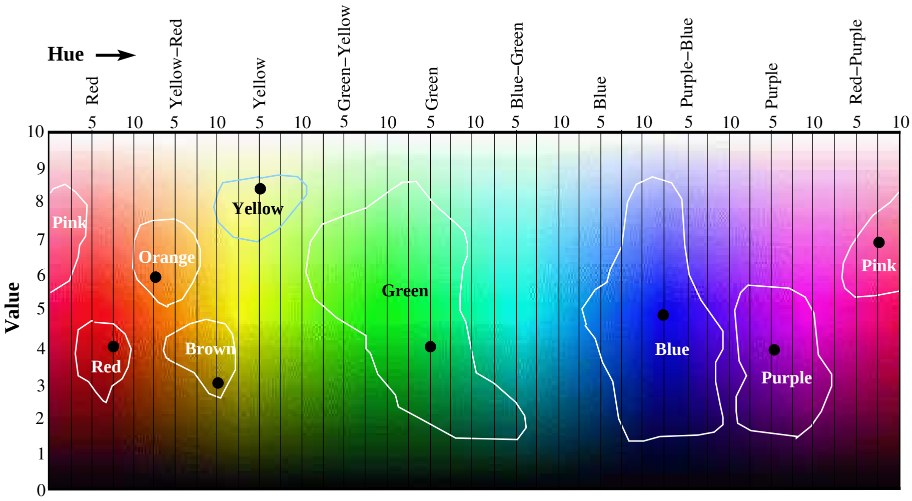


Figure 3.3: Berlin & Kay colour labels overlayed on the Munsell space.

Advantages

- The colour representation is even closer to human perception than hue.

- The resulting histogram representation is very compact, while gray scale are also incorporated.
- The method is less sensitive to illumination changes, while the problem is not eliminated.

Disadvantages

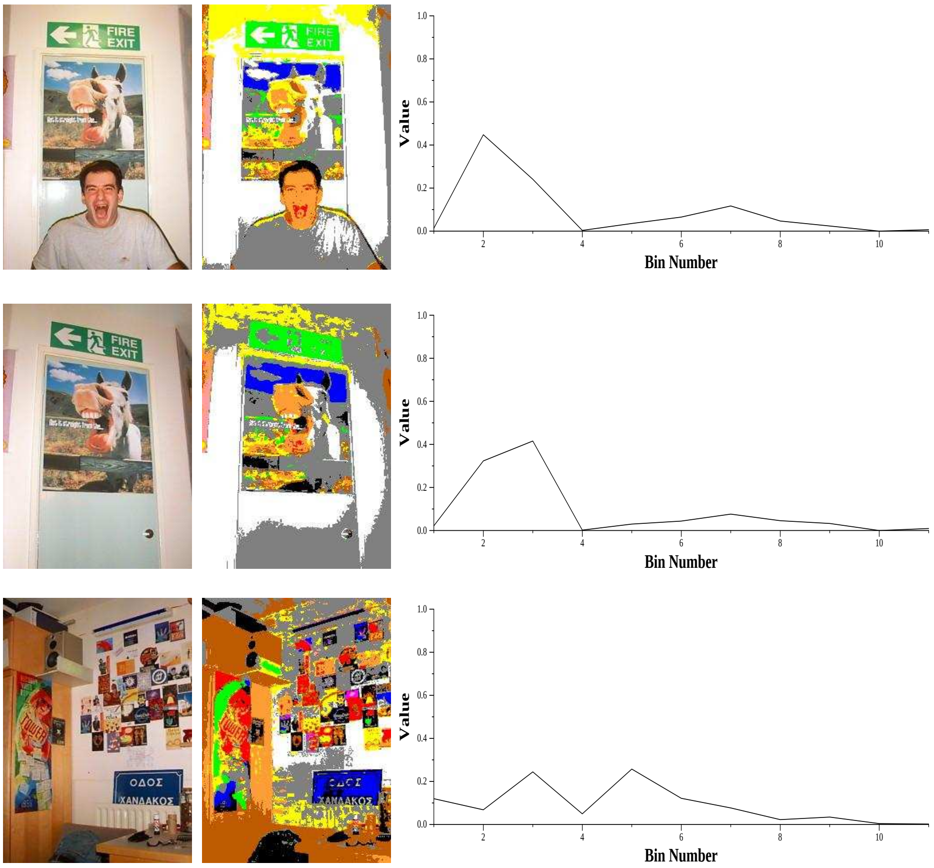
- The labelling process is not straight forward that may result to poor labelling. More accurate labelling is possible but the resulting solution may overly increase the complexity of the system.
- Images with different content may still have similar hue histograms.

Since each bin of the histogram comprises a disjoint class of colour, the choice of distance measure is not expected to influence the performance of the method significantly.

Brief experiments showed the colour labels histogram to give a better response than the hue histogram, when inspected visually. This is illustrated in figure 3.4. The histogram of the top two images show a greater similarity than in the case of the hue histograms. At the same time the histogram of the third image shows greater difference from the top two histograms.

3.2 Shape

An obvious approach to extract shape features from images for use in CBIR is as follows. Segmentation is performed on the image, then it is straightforward to measure region shape as well as determining spatial inter-relationships between regions. Segmentation is inherently such a difficult task that the performance of current algorithms falls far short of being able to provide an adequate input to such schemes [45][20]. In some contexts this may not be such a problem. For instance, in region-based classification it is acceptable to over-segment the image since the fragmentation does not disadvantage the classifier. However, for CBIR the region sizes, shapes, and positions are just as relevant as their underlying colour, intensity, and texture. Therefore the assumption made in many region based CBIR schemes on high quality segmentation is unrealistic [14], while more recent approaches that minimise their dependency on good segmentation are limited in scope



(a) RGB (b) Labelled (c) Histogram

Figure 3.4: Colour labels histogram examples.

[108]. In specific applications such as trademark retrieval it is feasible to reliably extract objects as a simple segmentation method such as thresholding is often sufficient, and then a host of shape measures can be applied [64].

Rather than perform region segmentation some researchers attempted the use of interest points to localise the processing to significant image windows. Properties, preferably invariant to intensity scaling, rotation, etc. can then be extracted, and used for retrieval within a voting scheme [100] or by histogram matching [124]. Corner detection and other interest operators are typically unreliable since they are looking for relatively complex features based on small windows of information.

This suggests edge detection as a more reliable approach (somewhere in between regions and interest operators). It is true that edges are detected using local window operators just like corners. However, the standard linking phase to generate connected edge curves provides a level of noise suppression, and easily enables isolated insignificant edges to be eliminated. Nevertheless, although edge detection may be more tractable than region or corner detection it is still prone to many errors such as mislocalisation, false edges, drop-out, incorrect linking, and so on, and therefore edge based CBIR techniques must be able to cope with these difficulties.

Since both extracting shapes and describing them is problematic we implemented methods to incorporate shape into histograms *i*) directly (Sections 3.2.2, 3.2.3, 3.5) and *ii*) indirectly eliminating the need for any meaningful segmentation or very accurate edge extraction (Sections 3.2.1, 3.2.5, 3.2.6). By indirect shape we mean that, although edges are the primary input, histograms are generated based on local characteristics of the image with respect to the edges, rather than explicit geometric calculations on the edges. In all cases where edge detection is required, edges were extracted as described in Appendix A.

3.2.1 K-Cos Edge Curvature

The *k-cos* is a computationally efficient method for estimation of curvature. It is derived from work by Rosenfeld and Johnston [88] and calculated as follows.

First we need to define two vectors, α and β. These vectors are directed from an edge point γ, to points k pixels ahead and k pixels behind, respectively. Clearly, the angle θ, enclosed by α and β, is closely related to the curvature at point γ on the edge. This angle θ (or rather its cosine) is obtained from the dot product of the two vectors, α and β. The value of $cos\,\theta$ is calculated by the vector cross product of α and β, and is within the $[-1..1]$ range, representing positive and negative curvature.

Note that k is a scale parameter. If k is too small the noisiness of the edge is measured rather than true curvature due to shape. Larger values of k provide curvature values due to shape, while too large k values result to essentially curvature values of a smoothed edge. An example of the *k-cos* calculation for a single edge point is illustrated in Figure 3.5.

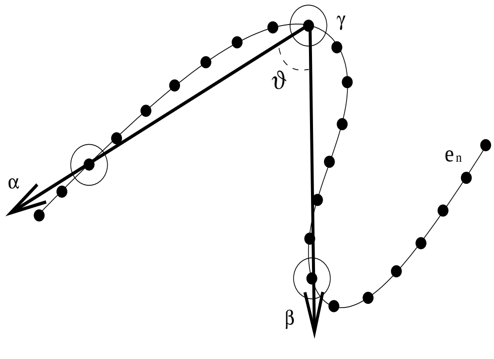


Figure 3.5: *k-cos* at an edge point, $k = 7$.

The *k-cos* histogram is calculated as follows. The images is transformed to grayscale that is used for edge detection. The resulting edge pixels are then linked and short edges are eliminated (we choose a threshold of 10 pixels). Curvature values are calculated for the pixels of the linked edges and then histogrammed. The effect of surviving spurious edges is limited by weighting the histogram votes by $w = \frac{\sum m_i}{L}$, where $0 \leq m_i \leq 1$ is the edge magnitude at pixel i, and L the length of the edge in pixels.

Advantages

- High order shape information is incorporated into the histogram.
- Because curvature is invariant to change and orientation it is expected that the histogram inherits these properties as well.

Disadvantages

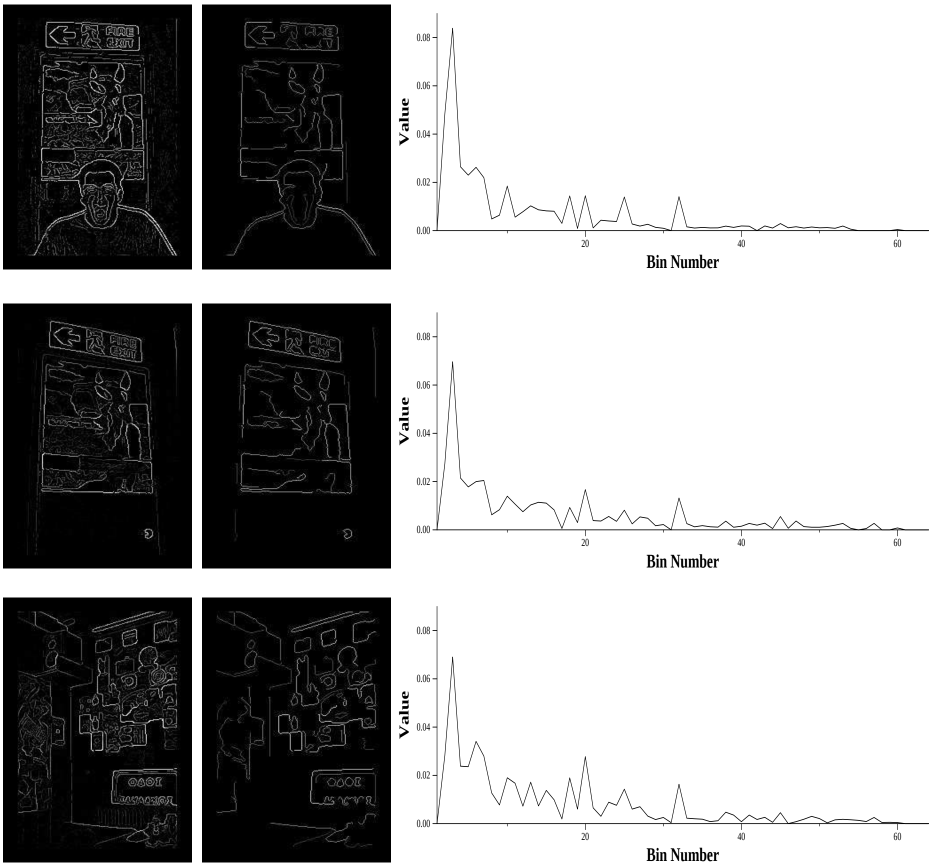
- Because of various problems that may arise while in the edge detection process (false edges, drop-out, incorrect linking and so on) the *k-cos* curvature histogram may perform poorly.
- As the scale k can have a great effect on the resulting curvature values, it may be important to use an adaptive scale.

Although an adaptive mechanism for choosing the scale variable k may be desirable, it is out of the scope of this project. We experimented with a range of k values and we found that for $k = 7$ we got the best response. The experimentation was based on visual inspection of histograms extracted from a small number of images.

Visual inspection of the histograms extracted from three images, as shown in Figure 3.6, showed subtle differences between the three cases. Although, parts of the histograms showed greater similarity for the top two images, the overall shape of the three histograms can be considered as visually similar. Therefore, conclusions cannot be drawn before further quantitative tests are performed.

3.2.2 Circular Arcs Histogram

Attempting to incorporate more direct shape to a histogram, we tried the *arcs* histogram. The basis of the arcs histogram is the work of Rosin and West [123] and [91]. Assuming



(a) Edges (b) Linked (c) Histogram

Figure 3.6: Edge curvature histogram examples.

that the edges of the image have already been extracted, the edge curves are segmented into circular arcs. Then the radii of the circular arcs are histogrammed.

Advantages

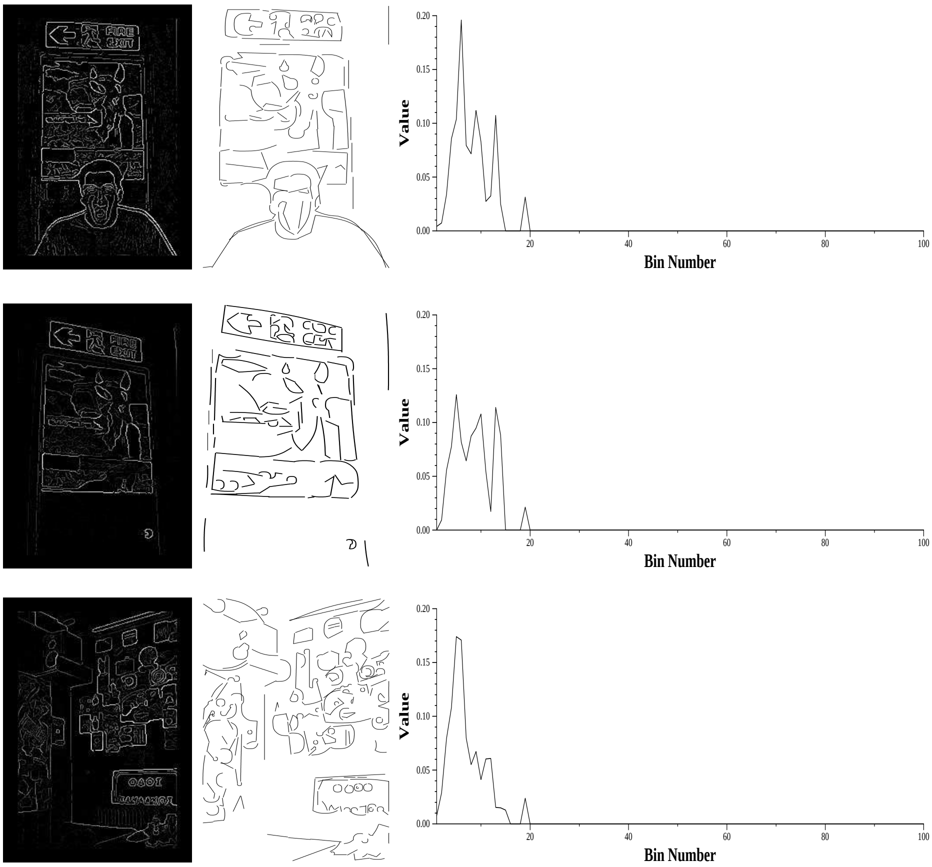
- Translation and Rotation invariance is inherited by the process of the circular arcs.
- Sensitivity to spurious edges is reduced by weighting the contribution of each circular arc (similarly to the edges for the k-cos histogram) by $w = \frac{\sum m_i}{L}$, where $0 \leq m_i \leq 1$ is the edge magnitude at pixel i, and L the length of the edge in pixels.

Disadvantages

- The scheme is not scale invariant, as the radius the circular arc is determined by the size of the edge.
- The circular arcs histogram is sensitive to the edge detection and linking process.

The algorithm of the circular arcs operates on the curves in a sequential manner. Disrupted edges may have a severe effect on the histograms of similar images.

During initial testing we noticed that the range of radius values was such that the upper limit was very large and the resulting histograms were very sparse. A solution to this was to histogram the logarithm of the radii. When applied to the three test images (Figure 3.7) we noticed that a relatively small number of bins (about 20-25) were populated. The histograms of the top two images showed greater similarity than the histogram of the third image, note the positions of the peaks.



(a) Edges (b) Arcs (c) Histogram

Figure 3.7: Circular arcs histograms.

3.2.3 Region Shape Histograms

So far, in Sections 3.2.2 and 3.2.3, histograms describe the distribution of abstract shape properties. These features are not related implicitly to the image content at a higher semantic level (e.g. natural or man-made scene). In this section we try to tackle this limitation by extracting more explicit shape information from images.

The first step is to split the image into regions, a meaningful segmentation would be the ideal tool. Because we want to keep complexity at low levels we preferred to use a simpler form of segmentation, binary thresholding. We expect that if the thresholding is robust enough, similar regions are to be extracted from similar images. For thresholding we used the method proposed by Tsai [120]. The choice of the particular algorithm was based on testing as described in Section 3.4.

After thresholding, majority filter is applied (with a mask size of 5×5 pixels) to eliminate very small regions. Then the edges of the black and white regions are extracted. Shape properties of those are then calculated and histogrammed for each region. A histogram is created for every property independently. We considered four different shape properties, as described in [90], namely, *i) convexity*, *ii) circularity*, *iii) ellipticity* and *iv) rectangularity.* In Figure 3.8, the above process is illustrated. The original images are shown in column (a) followed by the direct output of the thresholding algorithm, column (b). In columns (c) and (d) the filtered binary image is shown, with the extracted regions highlighted in black for both cases.

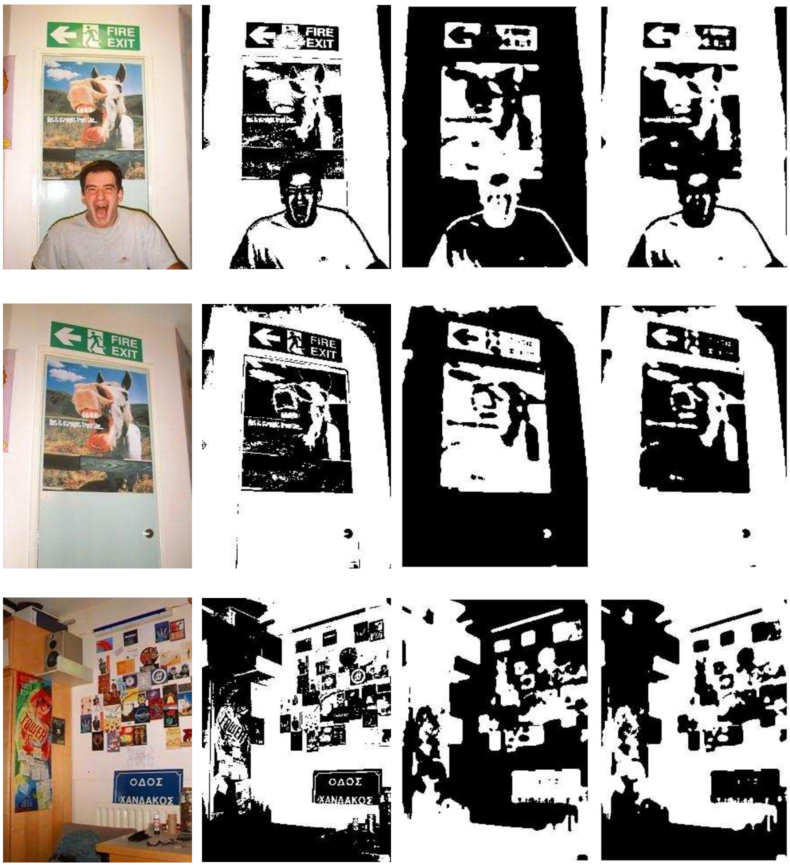
Advantages

- The extracted features are 2D rotation, scale and translation invariant.
- Meaningful segmentation is not necessary.
- The extracted properties can be related directly to higher level meanings. Having the man-made *vs* natural scene example in mind. For example a large number of concave regions suggest a natural area, e.g. a tree, while stronger geometrical shapes circular, elliptic or rectangular, are expected in man-made scenes, e.g. a building.

Disadvantages

- The resulting quality of the histograms is closely related to the quality of the segmentation. Although, meaningful regions are not a necessity, robustness is required so similar regions are extracted from similar images.

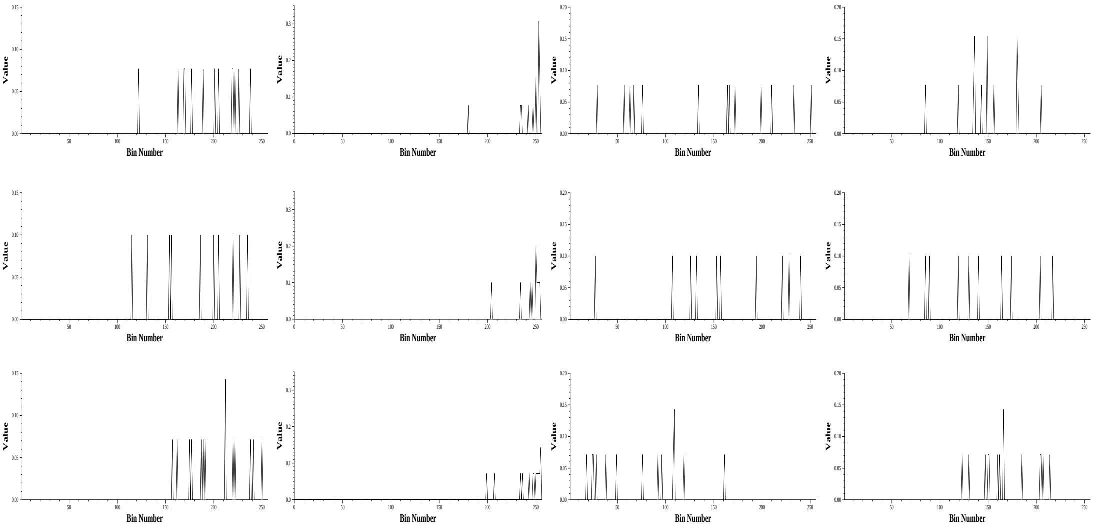
Initial testing, as illustrated in Figure 3.9, showed that in all cases of the mentioned region histograms performance was poor. We address this to the dependence of the features on



(a) RGB (b) Binary (c) White (d) Black

Figure 3.8: Thresholded and filtered test images.

the segmentation method. Even slight thresholding differences between similar images propagated to greater differences in the related histograms.



(a) Convexity (b) Circularity (c) Elipticity (d) Rectangularity

Figure 3.9: Region shape histograms of the three test images.

3.2.4 Orientation Histogram

Diverting from the above methods we tried to extract indirect shape from images. The implementation of the orientation histogram is a variant of the method proposed by Jain and Vailaya in [52]. In [52] the edge orientation histogram is used along with a colour histogram for image retrieval. The edge orientation histogram encodes some aspects of shape information, enabling querying to be more responsive to the shape content of the images. Moreover to extract the shape information only standard edge detection is required (e.g. Canny's algorithm [9]).

In the original form the edge orientation histogram was used against trademark logos. The simplicity of the images makes the edge extraction more reliable than in the case of more natural images. Because the prototype CBIR system is not purposed for a specific application we are expecting the edge extraction to be less reliable. For this reason we decided not to rely on arbitrarily thresholded edges, but to try and use as much available edge information as possible.

The by-products of the edge detection algorithm are, the results of the first order *vertical* and *horizontal* differentiation (X' and Y' maps), and the *edge magnitude* map. From the X' and Y' maps pixel orientations ($\theta = \arctan \frac{Y'_{xy}}{X'_{xy}}$) can be derived, while the edge magnitude map can be used as a weighting factor. For the orientation histograms we use the direction values for all the pixels. During histogramming the votes are weighted by the corresponding magnitude values. This emphasises pixels close to the area of the real edges, while pixels away from the edge areas have little or no effect.

Advantages

- Some aspects of shape information are encoded, enabling querying to be more responsive to the shape content of images.

- Avoiding edge thresholding reduces the reliability problems of the edges, which is directly related to the resolution of the histogram [52]. In the original version low resolution histograms would produce inaccurate results, while high resolution ones require a higher degree of accuracy of edge detection.

- The scheme is translation invariant and can also be orientation invariant, through histogram shift.

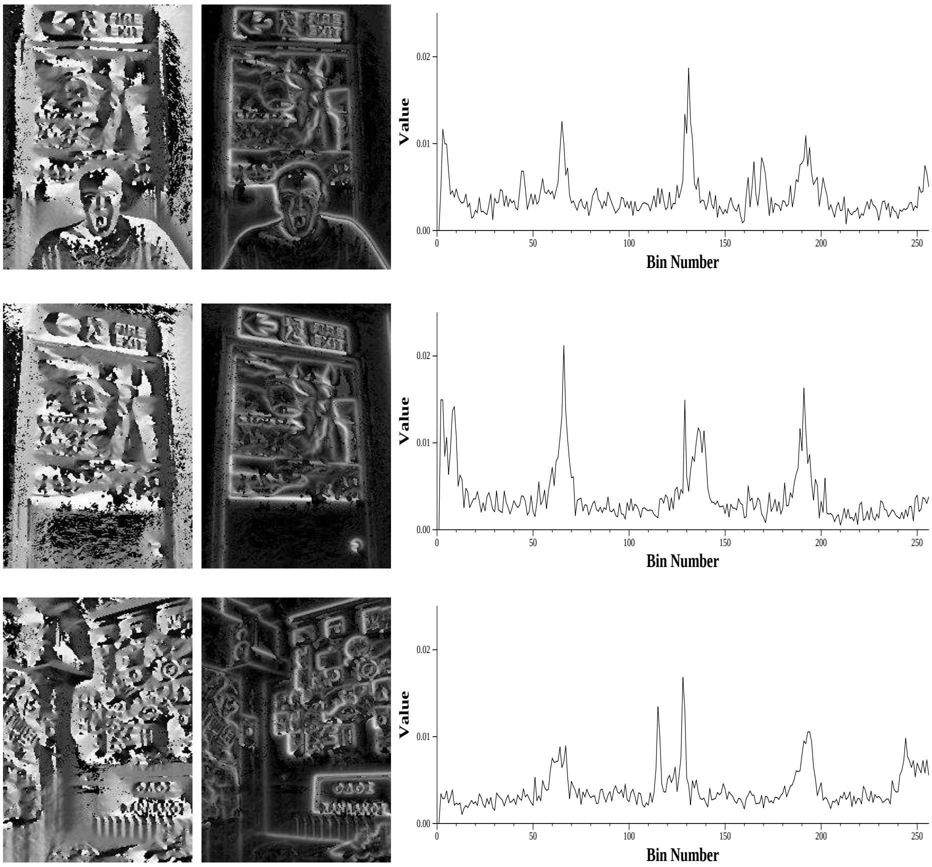
Disadvantages

- Regardless of the abovementioned measures for rotation invariance, the orientation histogram suffers for partial rotation, that is, one or more objects in the image are rotated at various orientations.

- Although the dependency to the edge detection is reduced, it is not eliminated. Spurious strong edges due to noise will affect the performance.

- Because no spatial information is included into the histogram we expect to see similar histograms for different images.

Brief visual inspection, Figure 3.10, showed a good response for the three test images. Peaks appear at close positions for the two top images, while an extra peak is apparent for the image with the human figure. The shape of the third image appears to be different from the other two, which is a desired behaviour.



(a) Orientation (b) Weighted (c) Histogram

Figure 3.10: Edge orientation histograms.

Although, the reduced dependency to an accurate edge detection, the histograms appear to be noisy. In [52], smoothing of the histogram is achieved by treating the histogram as a 1-Dimensional signal:

$$I_s[i] = \frac{\sum\limits_{j=i-k}^{i+k} I[j]}{2k+1} \tag{3.1}$$

3.2.5 Salience Distance Histogram

Another approach to extracting indirect shape is based on the *distance transform* (DT). The DT is a method for taking a binary image of feature (edges) and non-feature pixels and calculating at every pixel in the image the distance to the closest feature (edge). Although this is a potentially expensive operation, efficient algorithms have been developed that only require two passes through the image [8]. Just as edge thresholding poses a limitation for Jain and Vailaya's [52] approach, it also causes problems for distance transforms. Varying the threshold even slightly sometimes can generate large changes in the resulting edge image. To improve stability Rosin and West [92] developed a weighted version called the s*alience distance transform* (SDT). Rather than propagating out Euclidean (or quasi-Euclidean) distances from edges, the distances are weighted by the saliency of the edge. The effect of this was to downplay the effect of spurious edges by soft assignment while avoiding the sensitivity problems of thresholding.

Once the SDT has been performed the distance values are histogrammed. It can be seen that the histograms will respond differently to different types of shapes. First there is the crude distinction between cluttered, complex scenes and simple sparse scenes, which will result in different ends of the histogram being heavily populated. In that respect the distance histograms area provide an indication of image complexity, along the lines of Kawaguchi and Taniguchi's [54]. However, rather than return a single complexity measurement the shape of the histogram will indicate more subtle distinctions between shapes.

Advantages

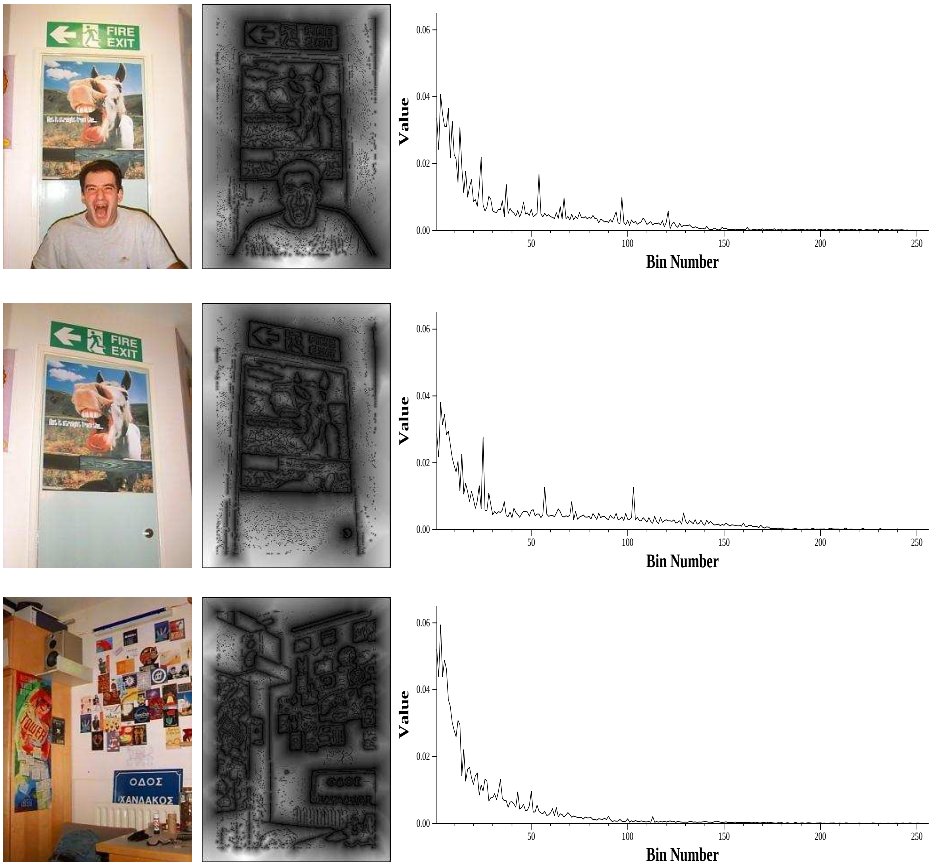
- The method is rotation and translation invariant.
- In this implementation, the effect of spurious edges is reduced by the ageing process of the edge detection.

- Spatial information, with respect to intensity changes, are also captured.

Disadvantages

- The quality of the resulting map is related to the quality of the edge detection, and the choice of threshold.

In Figure 3.11, salience distance histograms of the three test images are shown. The top two histograms show a greater similarity than when compared with the third histogram. The main differences from the third histogram are on *i*) the range of populated bins, and *ii*) on the scale of the histograms.



(a) RGB (b) Distances (c) Histogram

Figure 3.11: Multi-scale salience distance transform histogram examples.

3.2.6 Triangulation Histograms

In an attempt to extract alternative forms of shape from images we experimented with the Delaunay triangulation [83] of edge points. In [118] and [117], Tao and Grosky used the

Delaunay triangulation for image indexing. In [118] the scheme required isolated objects while in [117] rigid image content is assumed. While the proposed method is invariant to scale and rotation it does not perform very well when objects are rearranged.

Unlike Tao and Grosky, we did not try to extract any points of interest from the image edges. Instead, we used a subset of the edge pixels, avoiding introduction of more possible errors from the extra required operations. Additionally, our scheme does not exploit any object extraction methods and we do not try to capture any spatial dependencies explicitly. We also want to note that while this method uses similar tools with [118], it was conceived and implemented independently and in parallel.

Edge detection and linking is first carried out, as in Appendix A, which enables us to reduce spurious and short edges. The remainder edges are sub-sampled and Delaunay triangulation is applied on the selected edge pixels. Then we can histogram properties of the resulting triangles.

Advantages

- Delaunay triangulation is invariant to rotation and translation.
- Delaunay triangulation is fairly robust and small changes (missing or mispositioned pixels) have little effect on the resulting triangulation.
- The triangulation is not dependent on connectivity and therefore can cope with edge linking errors.

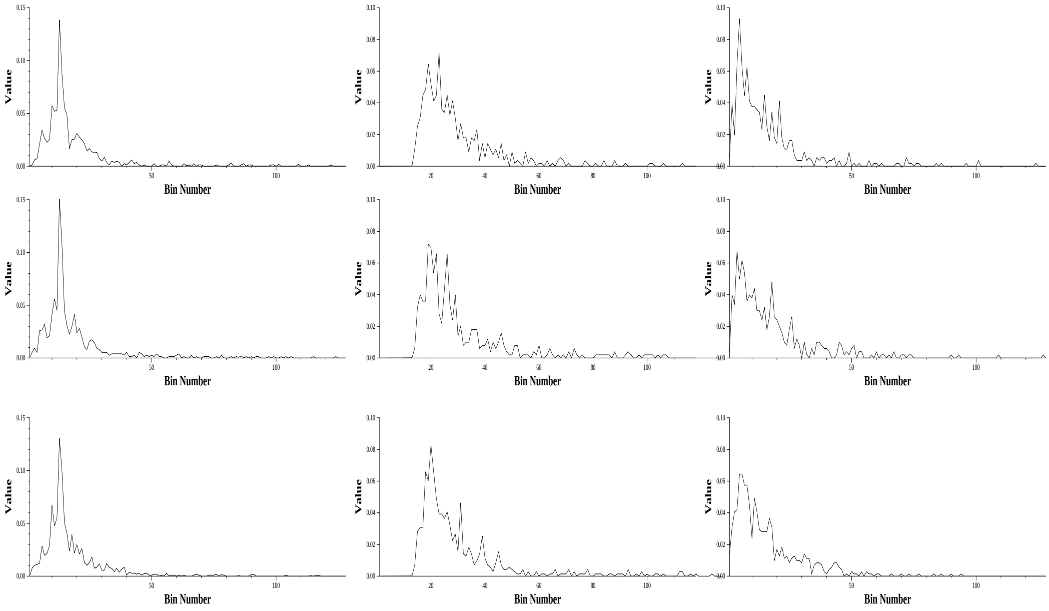
Disadvantages

- The selection of the edge pixels for triangulation is sensitive to the starting point of the sampling. Severely disrupted edges may be excluded from the sub-sampling process. This could be solved by post-processing of the extracted edges trying detect and connect such disruptions. In this implementation we did not make such an attempt avoiding the introduced complexity and the possibility of further problems related to such process.
- Fixed step sampling poses another limitation to this method, resulting to an increased sensitivity to scale. (e.g. As edges may get longer as scale increases, sampling will result to more edge pixels and eventually to more triangles). In this implementation

we sub-sampled the edges by a factor of 10 (every 10^{th} pixel). This value was selected after visual inspection of experimental triangulations.

We calculated and histogrammed three different triangle properties over the complete triangulation, *i*) the *area* of each triangle, ii) The ratio of the longest to the shortest edge of each triangle, and *iii*) the lengths of the triangle edges. In Figure 3.12, histograms of the three different triangle properties, extracted from the three test images are illustrated.

Initial testing showed only subtle similarities between the two similar images for all three histogram methods. Differences from the third image were also subtle, therefore further testing was required for assessment.



(a) Area

(b) Ratio

(c) Length

Figure 3.12: Histograms of triangle features.

3.3 Texture

A common extension to colour CBIR systems is to add textural information. There are many texture analysis methods available, and these can be applied either to perform segmentation of the image, or to extract texture properties from segmented regions or the whole image. In keeping with our general approach we use the standard co-occurrence method that produces a matrix analogous to a histogram.

3.3.1 Texture Spectrum

In [47] He and Wang proposed the texture spectrum. The first step is to analyse each pixel within its 3 × 3 neighbourhood. The vector $V = \{V_c, V_1, ..., V_8\}$ is constructed, where V_c is the intensity of the pixel at the centre of the 3 × 3 neighbourhood, and V_i the intensities of the neighbour pixels. V is then transformed to a texture unit $TU = \{E_1, ..., E_8\}$, with E_i defined as

$$E_i = \begin{cases} 0, & if\, V_i < V_c \\ 1, & if\, V_i = Vc \\ 2, & if\, V_i > V_c \end{cases} \tag{3.2}$$

Each texture unit is then mapped onto a unique integer, the texture unit number, by

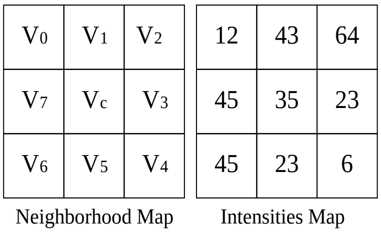
$$N_{TU} = \sum_{i=1}^{8} 3^{i-1} E_i \tag{3.3}$$

The frequencies of texture units form the texture spectrum, which is then treated as a texture histogram. In our implementation we used a simplified form of 3.2 and 3.3, as follows:

$$E_i = \begin{cases} 0, & if\, V_i < V_c \\ 1, & if\, V_i \geq V_c \end{cases} \tag{3.4}$$

$$N_{TU} = \sum_{i=1}^{8} 2^{i-1} E_i \tag{3.5}$$

The effect of this modification was a reduction of the number of unique texture unit numbers from 6561 to 256. An example of a texture unit utilising 3.4 is shown in Figure 3.13.



V_0	V_1	V_2
V_7	V_c	V_3
V_6	V_5	V_4

Neighborhood Map

12	43	64
45	35	23
45	23	6

Intensities Map

$TU = \{0, 1, 1, 0, 0, 0, 1, 1\}$
and
$N_{TU} = 0{+}2{+}4{+}0{+}0{+}0{+}64{+}128 = 198$

Figure 3.13: T-Unit example.

Advantages

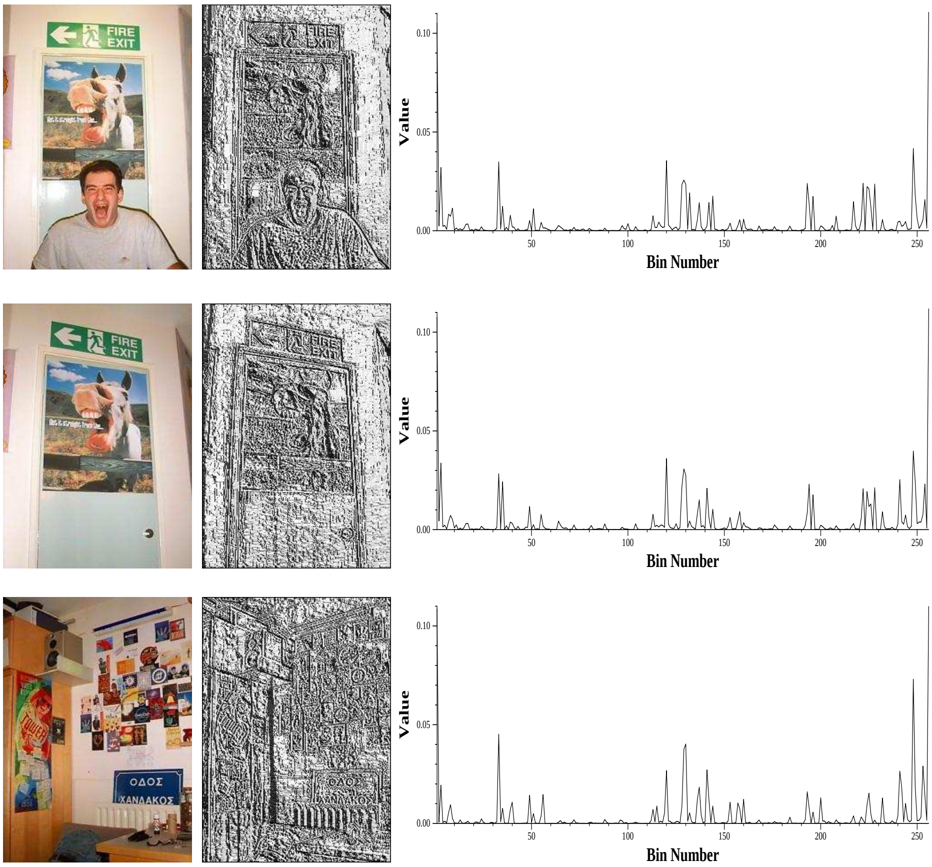
- Simple and efficient way to describe image texture.

Disadvantages

- Sensitivity to rotation. This is mainly because the texture unit number is dependent to the sequence of comparisons of the texture unit.

- Sensitivity to scale. Because the process of calculating the texture unit is very localised slight changes to scale may result to different texture unit numbers.

Initial testing, Figure 3.14, did not show any promising performance in CBIR. Differences between the histogram of the dissimilar images were subtle, while little visual difference was evidence. This is why we abandoned further study of the method.



(a) RGB (b) T-Units (c) Histogram

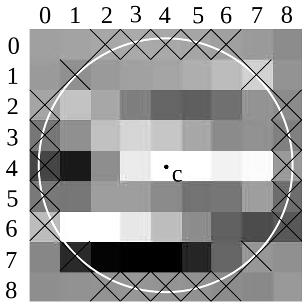
Figure 3.14: Texture spectrum examples.

3.3.2 Co-occurrence Histogram

The cooccurrence matrix was proposed by Haralick *et al.* in [46], as a method to represent the spatial distribution dependence of gray levels within an area. The cooccurrence matrix offers a very accurate tool for describing image characteristics and especially texture.

Starting from a point $\mathbf{c} = [x, y]$ and intensity $I(\mathbf{c})$ we define a digital circle of radius d obtaining a set of intensity pairs. This process is repeated for the centre located at each pixel in the image, Figure 3.15. These intensity pairs are then histogrammed to produce the co-occurrence matrix, as shown below

$$C(i, j|d) = \#([\mathbf{c}, \mathbf{p}]|I(\mathbf{c}) = i, I(\mathbf{p}) = j, \mathbf{p} = \mathbf{c} + d(\cos\theta, \sin\theta)) \tag{3.6}$$



The marked pixels define the digital circle. The intensities of the pixels (at the point c and on te digital circle) are quantised and used as indexes to the matrix. The quatisation is performed to reduce the resolution of the histogram to a practical size.

Figure 3.15: Digital circle.

In the original study the authors mention that, for practical reasons only some angles θ should be considered while maintaining the usefulness of the method. In our implementation we consider all the θs for the digital circle instead of just the two or so angles more commonly used. We think that this redundancy is useful as the pixels close to the image border are also considered.

The standard co-occurrence method produces a matrix that is analogous to a histogram. To maintain simplicity and consistency throughout the system we treated the matrix as a one-dimensional histogram.

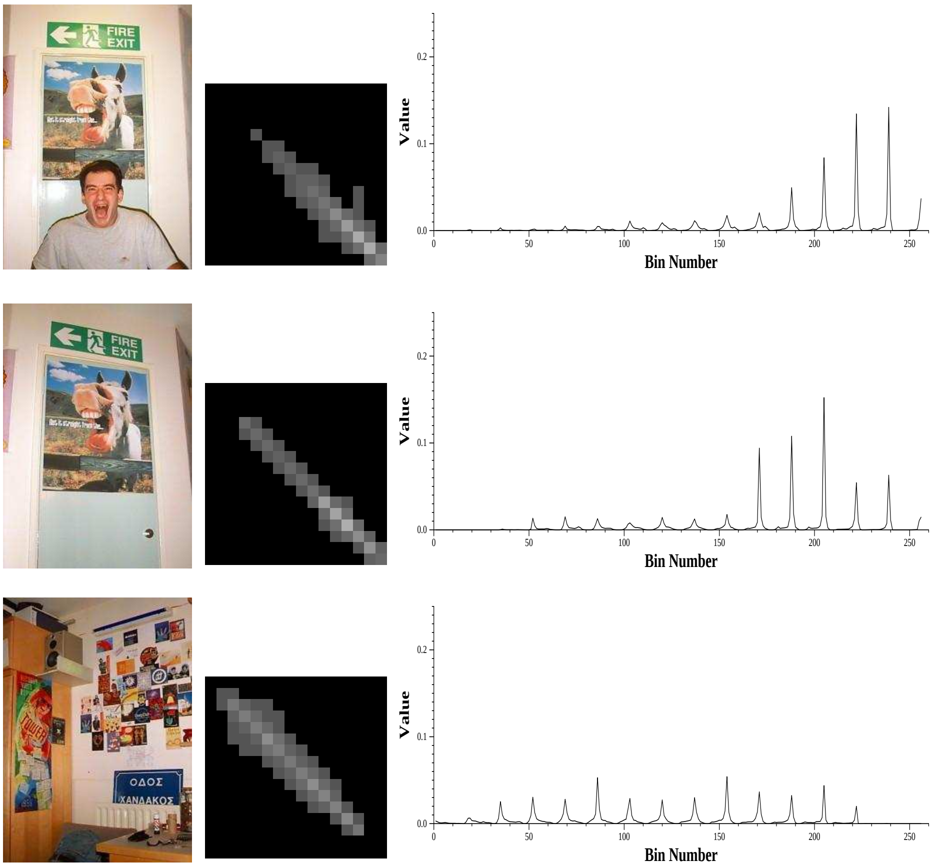
Advantages

- Reduced sensitivity to rotation. Because we take into account all the angles when calculating the co-occurrence matrix,
- Reduces sensitivity to scale. The parameter d, controlling the radius of the digital circle can be used to compensate for scale changes.

Disadvantages

- Sensitivity to scale is not eliminated. Although d can be adjusted, in this implementation it is fixed. A multiscale solution may help to overcome this problem.

The results of the initial testing on the three images were promising, Figure 3.16. The difference between the histograms of the top two similar images and the third image showed greater visual difference than many of the methods in previous sections.



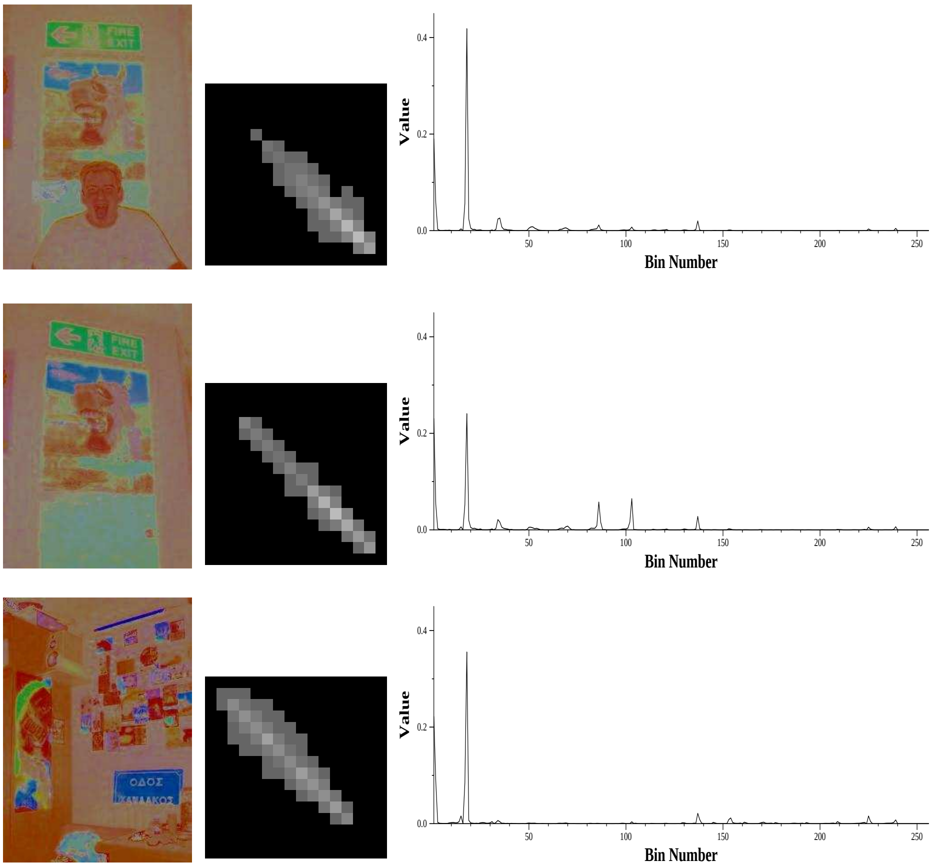
(a) RGB (b) Matrix (c) Histogram

Figure 3.16: Co-occurrence histograms of intensities.

The circular cooccurrence matrix method was also applied on hue images in an attempt to extract the dependences of hues within an area. Examples of histogram obtained from hue maps are illustrated in Figure 3.17.

3.4 Spatial Information by Soft-Thresholding

We previously stated that reliable segmentation is currently not possible. Therefore we do not attempt to perform true segmentation into meaningful regions. Instead the simplest



(a) Hue (b) Matrix (c) Histogram

Figure 3.17: Co-occurrence histograms of hue.

possible sort of segmentation is carried out, namely binary thresholding. While binarisation can produce meaningful regions for well-defined bimodal images it is unlikely to do so for less constrained, more naturalistic images. However, this is not our goal. All that is required is a reliable spatial partitioning of the image; the regions formed may be purely arbitrary in terms of their correspondence with objects in the scene.

The segmentation performs the same function as the partitioning based approach by Dimai and Stricker [112]. Spatial information is injected into the analysis so that standard feature based (e.g. non-spatial) methods can then be applied within each region. In fact we only consider each of the two classes black and white as single composite regions rather than treat each individual region separately. Not only does this avoid the need for component labelling but it also reduces sensitivity to variations in the thresholding process. The advantage of using thresholding to provide the image partitioning is that it is adaptive to the image content unlike proposed rigid grid methods.

There has been considerable work carried out on thresholding methods [96]. To ensure that the method used was reliable and repeatable tests were carried out to compare several thresholding algorithms. We compared three representative types of algorithm, based on entropy [53], statistics [77] and moments [120]. The test was performed using 5 images from which we extracted patches of random size and position. The size of each patche was in the range of 10% to 90% of the original image size in terms of number of pixels. The patch size range was quantised to 9 steps and for each step we extracted at least 100 patches from each image. Then we applied the thresholding algorithms to all the patches and images. As a robustness measure we calculated the difference between the binary patches and the related area of the thresholded image, i.e. the percentage of thresholded pixels that differed due to the differences in the global statistics. In Figure 3.18, the curves of the average thresholding performance for the three algorithms are shown. Tsai's moments based method worked best, and so it was selected for all future experiments.

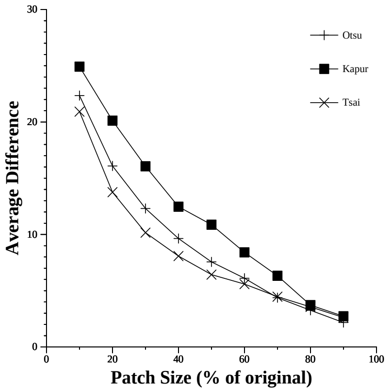


Figure 3.18: Thresholding algorithm comparison.

A potential drawback is that the process can sometimes be sensitive such that small changes in the threshold value produce large changes in the resulting binary images. To overcome this we introduced a *soft threshold.* Starting from the binary threshold value (obtained by the thresholding algorithm) the transition between black and white is fuzzified, Figure 3.19.

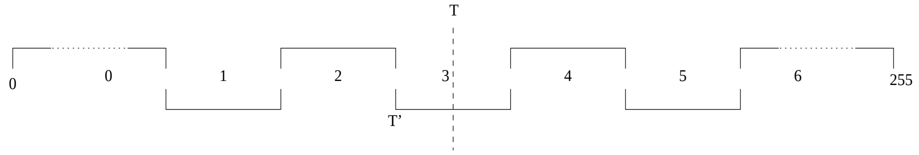


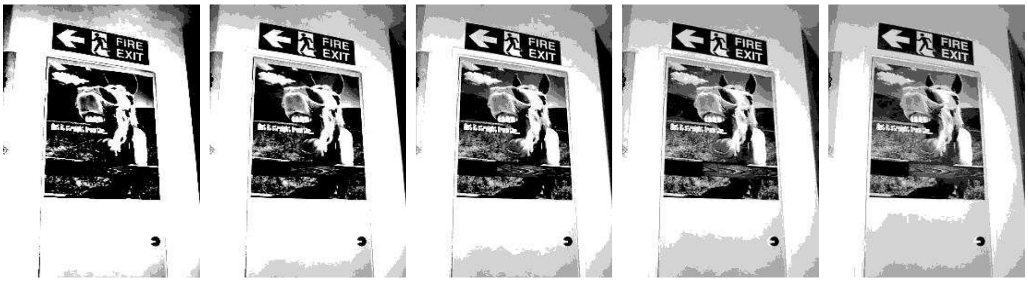
Figure 3.19: Soft thresholding ranges when $n = 6$.

Let T be the original threshold point and s the step size. Then we want to centralise T to a range of size s to offset the effect of threshold to the limits of the range rather than the threshold point. Thus we define a new point $T' = T - \frac{s}{2}$. Then for pixels with intensities close to the original threshold value the soft threshold maps (SM_B and SM_W) for the two classes is calculated as

$$SM_B(i,j|T',s) = \begin{cases} 0 & if\, I(i,j) \in [0...T' - 2s] \\ 1 & if\, I(i,j) \in (T' - 2s...T' - s] \\ 2 & if\, I(i,j) \in (T' - s...T'] \\ 3 & if\, I(i,j) \in (T'...T' + s] \\ 4 & if\, I(i,j) \in (T' + s...T' + 2s] \\ 5 & if\, I(i,j) \in (T' + 2s...T' + 3s] \\ 6 & if\, I(i,j) \in (T' + 3s...255] \end{cases} \tag{3.7}$$

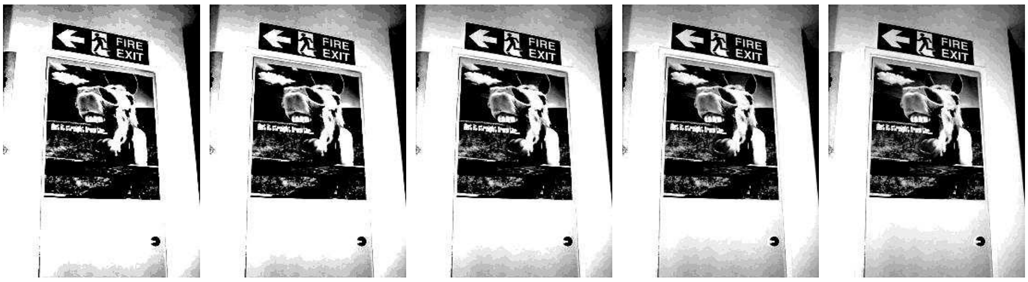
$$SM_W(i,j) = 6 - SM_B(i,j)$$

The resulting soft threshold map is used to weight the contribution of a feature at a specific point. For simplicity only two steps were used on either side resulting in six possible weight values, Figure 3.19. Informal experimentation showed that a practical range of values for the step (s) constant for real images to be $s \in [5...10]$. Otherwise, the fuzzy range is not large enough to be useful, or if the original threshold is very close to the intensity limits (usually for images with very high or very low illumination) the range is not symmetric on both sides of the binary threshold. In Figures 3.20 and 3.21 the effect of different values of n and s are illustrated. Adaptive selection of s may result in more accurate maps but was not implemented.



(a) $s = 5$ (b) $s = 10$ (c) $s = 15$ (d) $s = 20$ (e) $s = 25$

Figure 3.20: Effect of step size s on soft thresholding ($n = 6$).



(a) $n = 8$ (b) $n = 10$ (c) $n = 12$ (d) $n = 14$ (e) $n = 16$

Figure 3.21: Effect of number of steps n on soft thresholding ($s = 5$).

Advantages

- The process is rotation scale and translation invariant
- Spatial information is injected into the analysis.
- Soft-thresholding improves the robustness of the segmentation.
- Although some weaknesses of histogram methods used along with soft-thresholding are inherited, the ones related to spatial information are expected to have reduces effect.

Disadvantages

- Suffers from illumination variance.
- Spatial information is not explicit, and therefore information about individual entitie (e.g. a vehicle, the sky etc.) cannot be extracted from the histograms.
- The weakness of the histogram methods used along with soft-thresholding, not directly related to spatial information still remain.

Soft-thresholding was used along with four histogram methods, namely, *i*) hue, *ii*)colour labels, *iii*)texture spectrum, and *iv*)cooccurrence histogram. Related figures are located at the end of this chapter.

Spatial Hue Histogram. Visual inspection of the spatial hue histograms, Figure 3.30, showed better discrimination than the single hue histogram. While the shapes of the SM_B region resembled similar behaviour as in the case of the single hue histogram, the picture was not the same for the histograms of the SM_W region. The histograms of the top two images showed greater similarity than the third histogram.

Spatial Colour Labels. A similar improvement was also noticed for the case of the spatial colour labels, Figure 3.31.

Spatial Texture Spectrum. For the case of the spatial texture spectrum we did not notice any improvement, Figure 3.32. Therefore we stick with our decision not to do any further tests.

Spatial Cooccurrence Matrix. Similarly, for the case of the spatial cooccurrence matrix (on intensities), Figure 3.33, we did not notice any visual improvement.

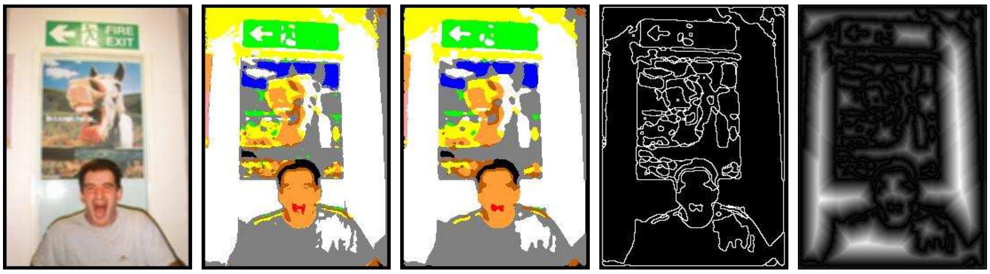
3.5 Colour and Distance Histograms

In Section 3.2.5 we introduced the distance histogram as a method to capture indirect shape from images. In this section we are attempting to extend the method to include information related to colour. The basis of the extension is a form of colour based segmentation. Similarly to Section 3.4, we do not try to extract meaningful regions from the image, but segment images into colour regions. Assuming a level of robustness, we expect regions or similar colour content to be treated the same way by the segmentation algorithm. We do not try to extract any particular properties of the regions, like in Section 3.2.3, but use the colour content to direct the analysis.

For segmentation we used the colour labels method, as described in Section 3.1.2. The main reasons for this are i) the low number of labels, and ii) the colour labels are very close to human perception. To reduce the effect of noise we used Gaussian smoothing on the image, prior to labelling. Furthermore to eliminate very small colour regions we applied a median filter on the resulting colour label map.

3.5.1 Distance of Colour Region Boundaries

Our first attempt was to apply the distance transform to the boundaries of the colour regions, followed by histogramming of the intensities of the distance map. An example of the process from image smoothing to the distance map is illustrated in Figure 3.22.



(a) RGB smooth (b) Labelled (c) Filtered (d) Edges (e) Distances

Figure 3.22: Processing stages to obtain the distances of colour region boundaries.

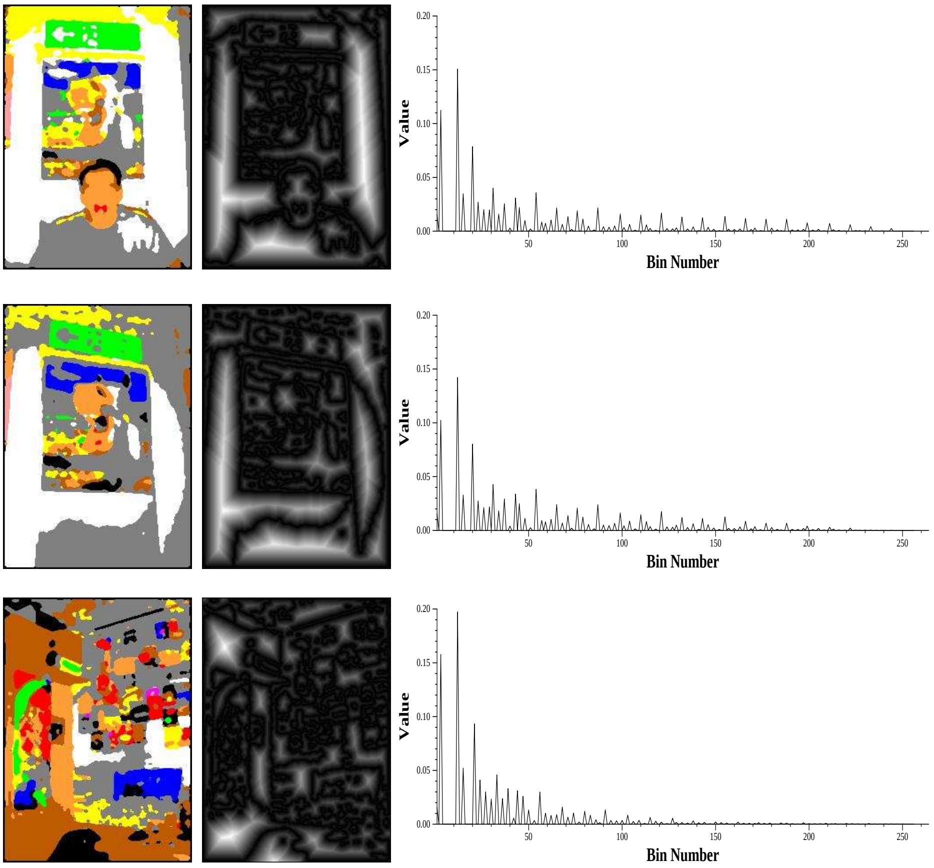
Advantages

- Colour and shape features are combined in a single histogram.
- Both features are invariant to rotation and translation.
- Spurious colour regions are reduced by applying Gaussian smoothing ($\sigma = 2.4$) of the original image and filtering of the resulting colour label map (we used a 5×5 mask for the median filter).
- The resulting distance maps are normalised so the intensities are in the range of [0..255] so the effect of scale is reduced.

Disadvantages

- The performance of the method is closely related to the performance of the segmentation algorithm.
- Edges are not weighted, so spurious edges that were not eliminated have a greater effect than in the case of the salience distance histogram.
- The current implementation is not multiscale. While spurious regions are reduced by filtering, they are not eliminated. A multi scale approach could result to a better segmentation, but so would an alternative colour based segmentation algorithm. This is why we did not attempt to implement a multiscale version.

Initial testing and visual inspection of the resulting histograms showed a similar behaviour as in the case of the salience distance histogram. Similarity between the top two histograms was greater than when compared to the histogram of the third (dissimilar) image. We also noticed a sparser histogram population than in the case of the salience distance histogram. That is due to the quantisation of the intensities of the distance map.



(a) Labelled (b) Distances (c) Histogram

Figure 3.23: Colour region boundary distance histogram.

3.5.2 Colour Labels *vs* Colour Region Boundary Distance

Trying to use colour in a more active way we expanded the process by using the pixel values of the colour and shape maps as indexes to a matrix. This way we formed a two dimensional histogram. In this histogram each bin represents the frequency of occurrence of a colour at some distance from a feature (based on the colour region boundaries).

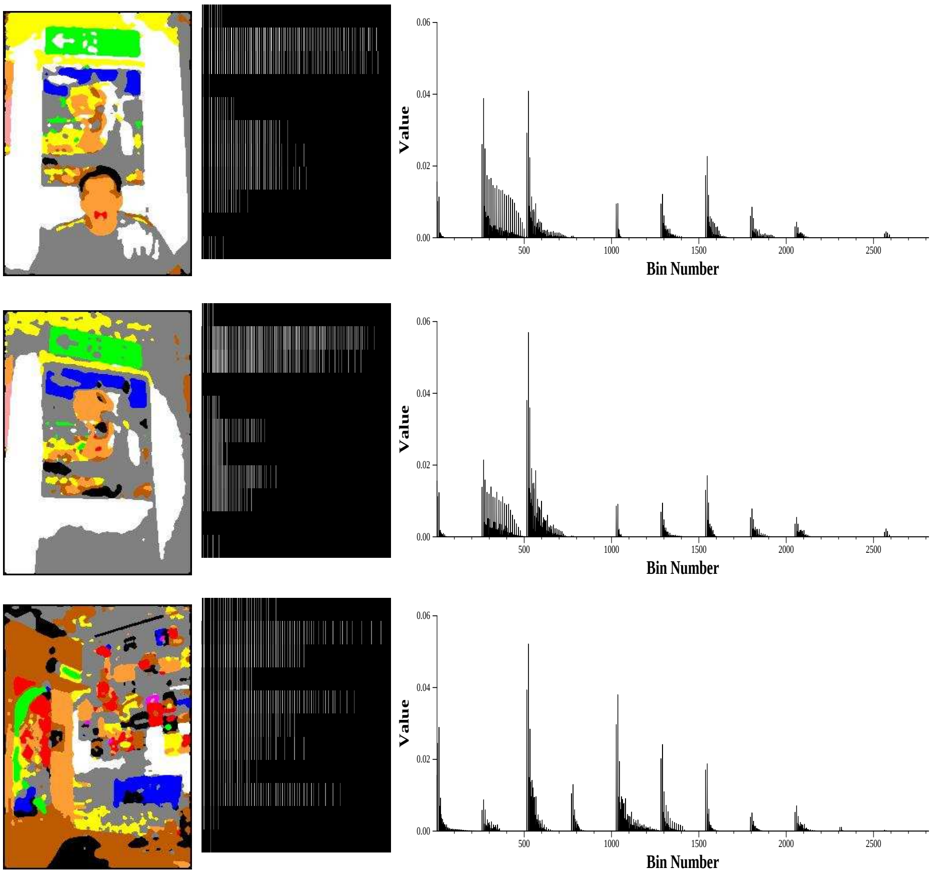
Advantages

- Both colour and shape information is used directly.

Disadvantages

- The performance of the method is still closely related to the performance of the colour labelling algorithm.
- Spurious edges may still cause a problem as their effect is not eliminated.
- Greater storage requirements for the histogram.

The histograms of the top two images showed greater similarity than when compared to the third histogram. Overall the population of the histogram bins showed a domination of the colour information. This suggests that coarser bins may be used for the shape dimension. In Figure 3.24, the three examples are illustrated, note that the 2D version of the histogram is gamma corrected to improve visibility, and it is shown just as a reference. For the experiments we used the 1D version where the rows are concatenated.



(a) Labelled (b) Matrix (c) Histogram

Figure 3.24: Examples of colour labels *vs* colour region boundary distance histograms.

3.5.3 Colour Labels *vs* Edge Multiscale Salience Distance

As an extension of the *colour labels* vs *colour region boundary distance* histogram we combined the colour information from the labelling process with the distance map obtained by the multiscale version of the salience distance transform.

Advantages

- Both colour and shape information is used directly.
- The shape information comes from a more reliable source. As the multiscale version of the salience distance transform algorithm is used, the effect of spurious edges is reduced.

Disadvantages

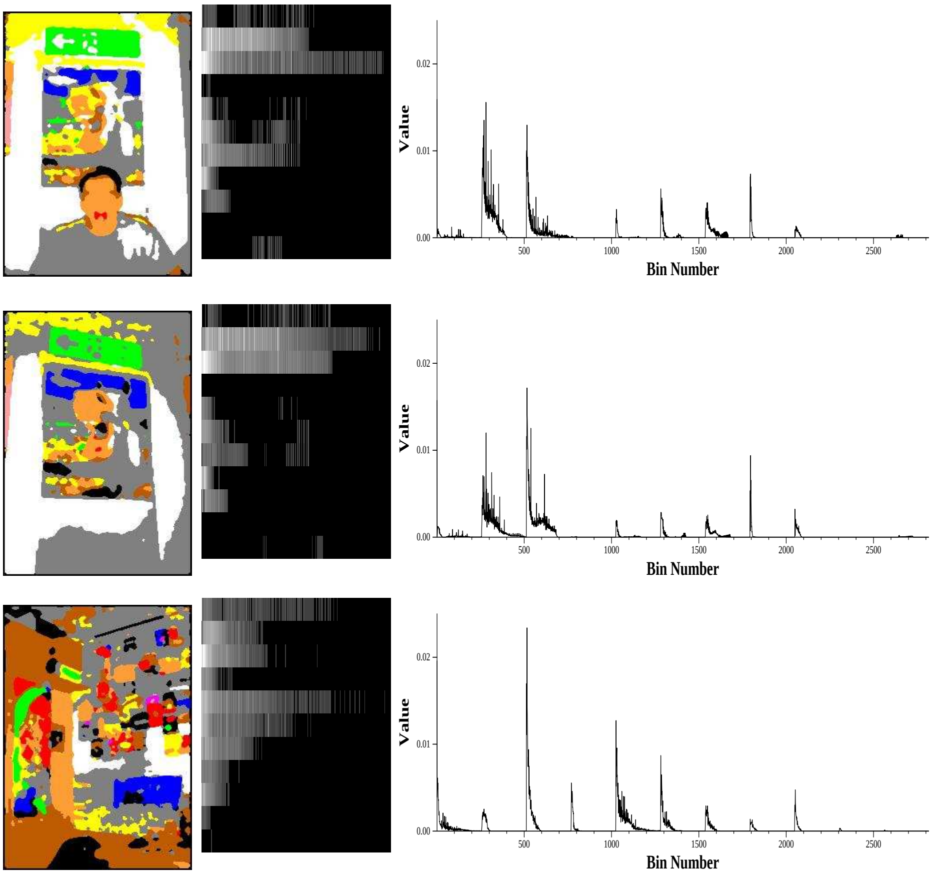
- The performance of the method is still closely related to the performance of the colour labelling algorithm.
- The histogram storage requirements are the same as in colour labels *vs* colour region boundary distance histogram.

The performance of colour labels *vs* edge multiscale salience distance histogram was similar to the colour labels *vs* colour region boundary distance histogram in terms of discrimination between the three test images. In Figure 3.25, the three examples are illustrated, note that the 2D version of the histogram is gamma corrected to improve visibility, and it is shown just as a reference. For the experiments we used the 1D version where the rows are concatenated.

3.6 Other Hybrid Methods

3.6.1 Hue Entropy Histogram

In an attempt to involve more colour in the feature histograms we calculated entropy in local windows over the hue map of the image. As a first step we transformed the image to HSV keeping only the hue channel, throwing away the rest of the information (as we are interested only on the plain colour aspect of the image). Then we define a set of



(a) Labelled (b) Matrix (c) Histogram

Figure 3.25: Examples of colour labels *vs* edge multiscale salience distance histograms.

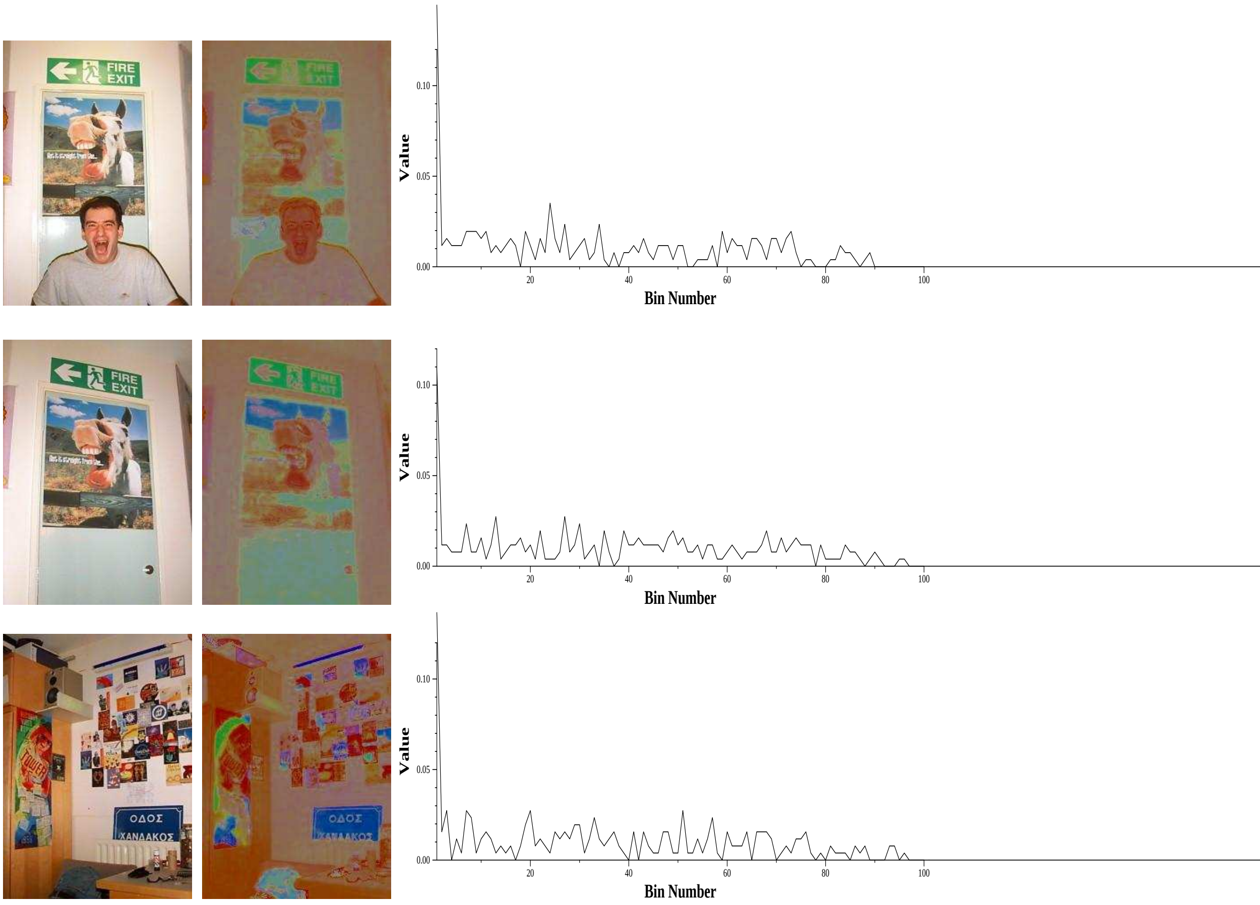
$N \times N$ overlapped windows. For each window, cantered at position (x,y), we generate the histogram of the hues and then calculate the entropy ($E_{H_{xy}} = -\sum p_i \log_2 p_i$) of the histogram. The desired image signature is then obtained by simply histogramming the entropy values over all the windows.

The size of the hue histogram affects the calculation of the colour entropy. We experimented with static and dynamic histogram sizes, and we found an optimal performance with dynamically sized histograms of $size = min(w^2, 256)$ bins[1] with w being the width of the window.

3.6.2 Local *vs* Global Difference Histogram

We used Tsai's moment preserving thresholding in order to capture local low-level differences on the image. The overall process is similar to the process used to test the robustness of the thresholding algorithms, in Section 3.4.

[1] This approach to dinamically choose the bin size of a histogram is also suggested by Wand in[122].



(a) RGB (b) Hue (c) Histogram

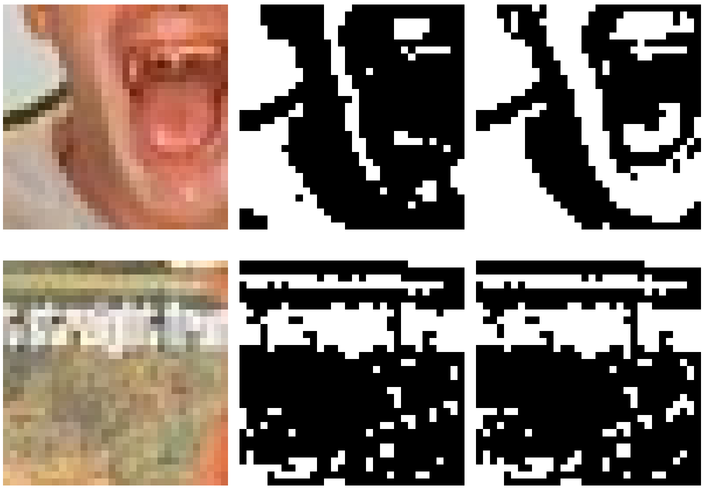
Figure 3.26: Hue entropy histogram examples.

First we defined a set of $N \times N$ windows. Then we applied thresholding to the whole image and obtained a binary image. The next step was to extract from the original RGB image the patches corresponding to the defined $N \times N$ windows and apply Tsai's thresholding on them. The last step is to measure the differences (at pixel level) between the binary patches, extracted from the RGB image, against their corresponding positions on the global binary image. Two instances of the process are illustrated in Figure 3.27.

By using this method we wish to capture similarities or differences of the local statistics of small windows against the corresponding global statistics. This can be considered as capturing information about global texture from images.

Advantages

- Because we do only try to capture differences and any kind of structural information the method is invariant to rotation and translation.



(a) RGB patch (b) Global threshold (c) Local threshold

Figure 3.27: Examples of thresholding patches. (b) is a patch from the global binary image, while patch (c) is the result of thresholding of the RGB patch (a).

Disadvantages

- Spurious edges may still cause a problem as their effect is not eliminated.
- Greater storage requirements for the histogram.

We used the same parameters as in the case of the hue entropy window size.

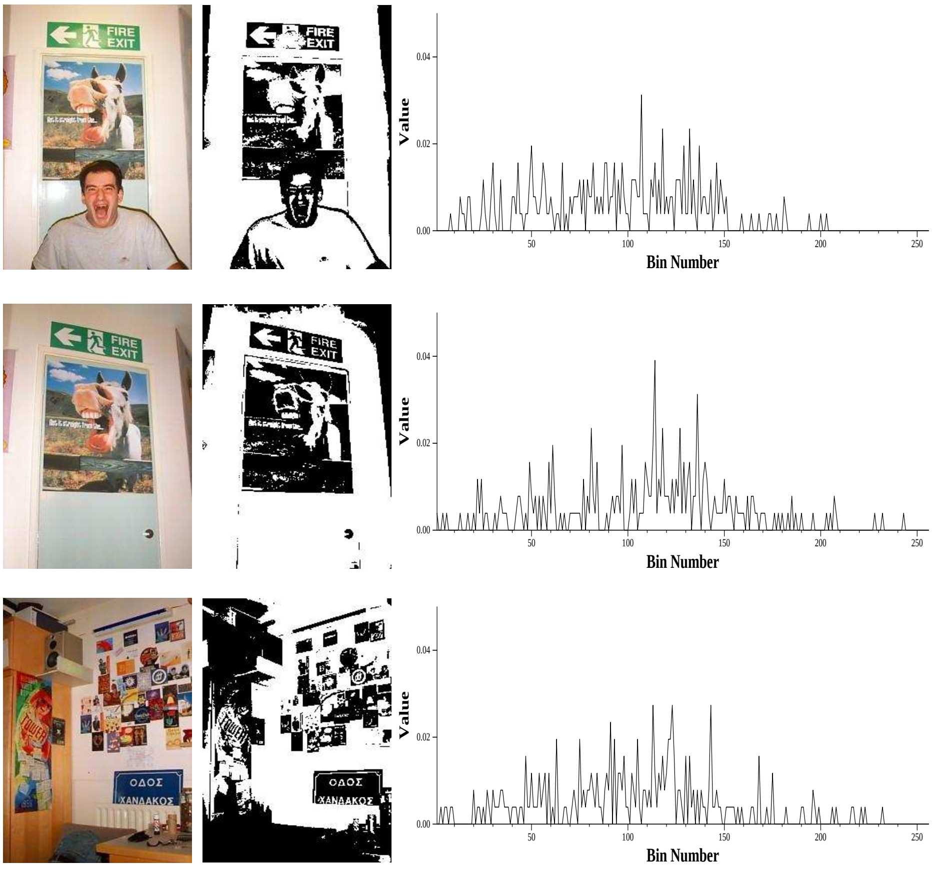
Thresholding is applied both globally to the image and locally to the individual windows. Then the percentage of pixels in the window that have different values according the two thresholds is histogrammed.

Additionally, we histogrammed the amount of *blackness* found in the window content after thresholding.

In both methods 2.2 and 2.3 we used overlapping windows. The difference caused by the amount of overlap was found to be negligible. So for our experiments we used an overlap of 25% of the window size in each direction. In Figure 3.29 an example of thresholding for measuring the *difference* is illustrated.

3.7 Summary

Feature extraction is the first step in CBIR, when new images are added in to a collection, and when queries-by-example are performed. In this chapter a number of existing and new histogram methods for CBIR were discussed. We demonstrated that a histogram is capable

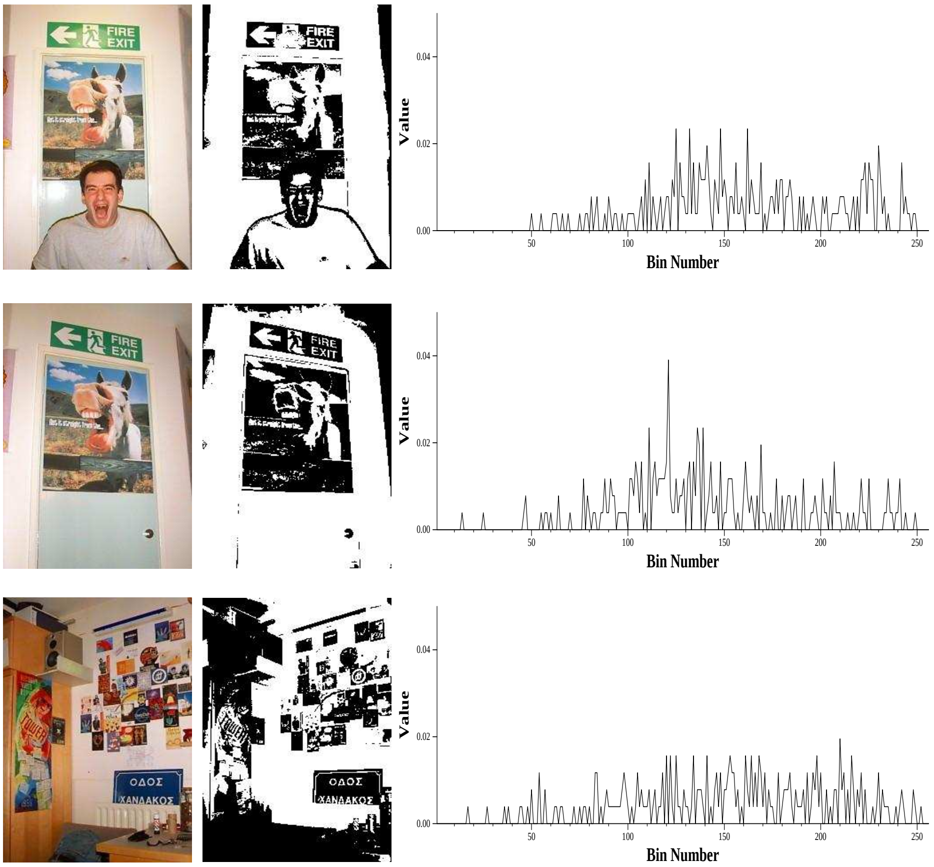


(a) RGB (b) Binary (c) Histogram

Figure 3.28: Local black histogram examples.

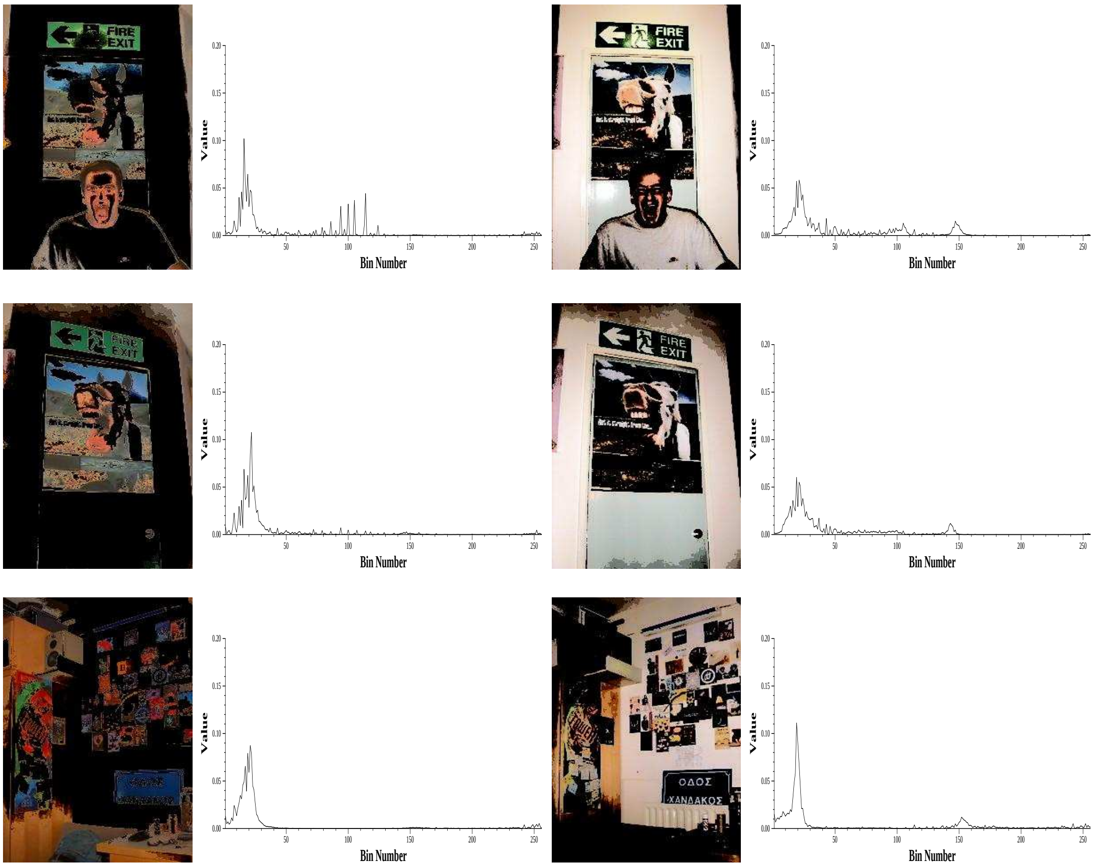
of summarising characteristics of image features. We also illustrated example histograms extracted from three test images and discussed their discriminatory characteristics based on their visual appearance.

Experiment results of the methods (and combinations) presented in this chapter are compiled in Chapter 6, using the evaluation framework presented in Chapter 5.



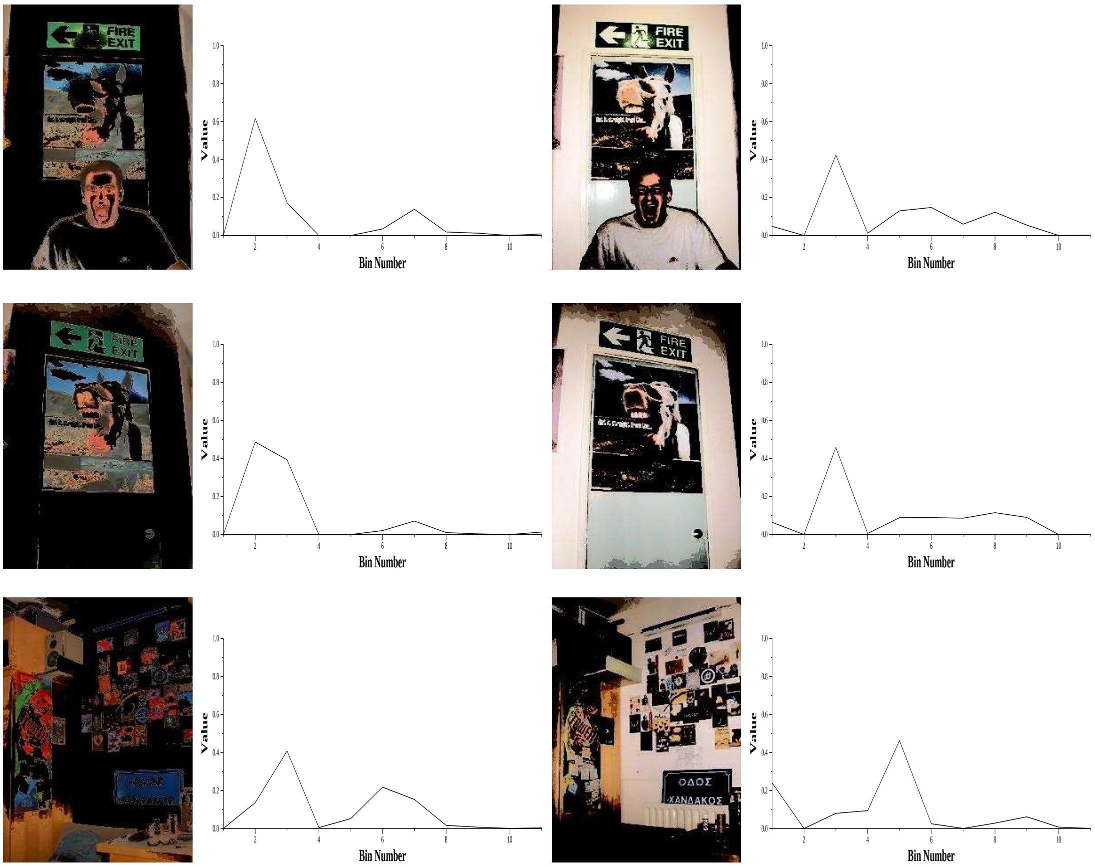
(a) RGB (b) Binary (c) Histogram

Figure 3.29: Local difference histogram examples.



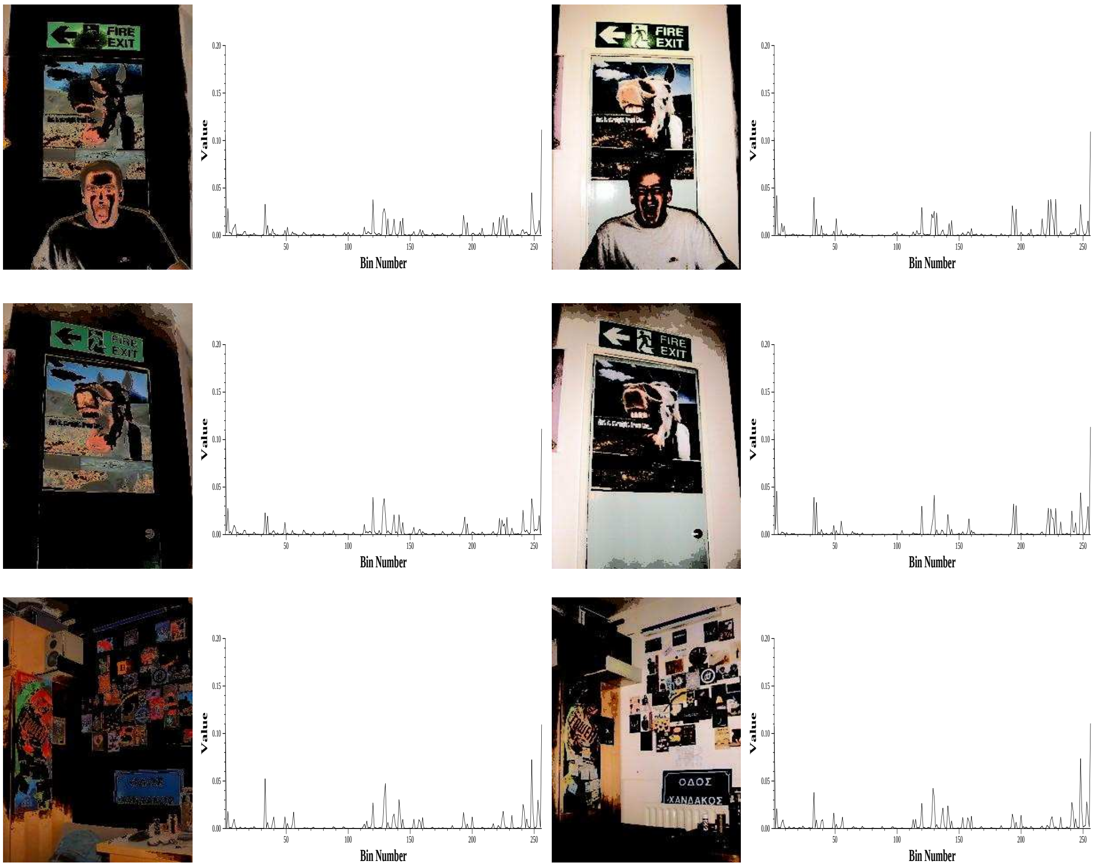
(a) SM_B (b) Histogram for SM_B (c) SM_W (d) Histogram for SM_W

Figure 3.30: Soft thresholding and the hue histogram.



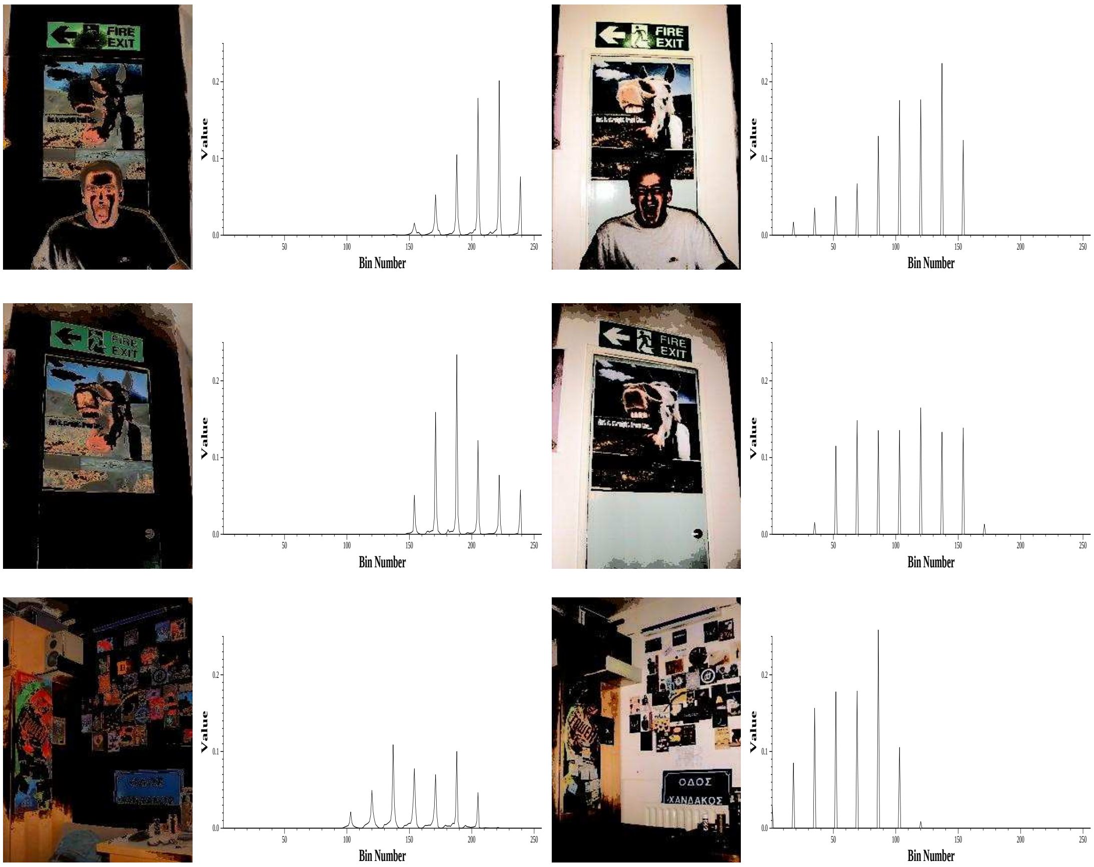
(a) SM_B (b) Histogram for SM_B (c) SM_W (d) Histogram for SM_W

Figure 3.31: Soft thresholding and the colour labels histogram.



(a) SM_B (b) Histogram for SM_B (c) SM_W (d) Histogram for SM_B

Figure 3.32: Soft thresholding and the T-unit histogram.



(a) SM_B (b) Histogram for SM_B (c) SM_W (d) Histogram for SM_W

Figure 3.33: Soft thresholding and the cooccurrence histogram.

Chapter 4

BEKAS - Closing the Gap

Introduction

One of the main challenges in CBIR is to extract and use semantic level information. For over a decade histograms of image features served CBIR applications. This CBIR model is based on the assumption that, if there is a similarity between low level features of two images, chances are that the images are of similar content. Despite the success of this model, histograms fail to extract image content in any meaningful form (e.g. people, animals *etc.*). Histograms provide just an indication of the low level similarity of images, without any explicit association to higher level terms.

Histograms, based purely on low level features (even for the cases of perceptually correct features), do not have a particular high level meaning unless they are given one. Under this condition a fully automated CBIR system would not be feasible, as the vast amount of information required to make it high level conscious would require infinite time. In an attempt to overcome that disability, early techniques were based on *i*) relevance feedback and iterative image querying, and *ii*) perceptually correct feature vectors. In the first case there is no use of high level content, while in the later, although higher level information is utilised, its use is passive limiting the potential of such knowledge.

Attempts at more active use of semantic content are evident, such as the VISOR, by Leow and Miikkulainen [57], the rich region descriptor by Clinque *et al.* [19], or the use of the focus of attention to associate sematic labels to objects, by Martinez and Serra [62]. Most of these techniques though, approach the problem partially (e.g. by simulated input) by using limiting assumptions (e.g. positioning of the semantically rich regions, or simple images with only one or two objects).

On the other hand, positive results while making use of semantic content have been reported. In [43] by Tao and Grosky, reported improved performance when they integrated lexical data along with their low-level techniques. Tansley *et al,* in [116], discuss the possibility of automatic tagging of images by the means of low level similarities with already tagged images, with positive results.

We think that one of the main bottlenecks to a more semantic sensitive CBIR system is to achieve associations between high and low-level features. We propose to approach this problem as a straight forward classification problem. We extract localised low level features (in the form of histograms) and using user input, we generate associations to semantic labels by the means of a classification algorithm. Furthermore, we use spatial relationships of labels (setups) as a link to a higher level concept hierarchy. The relationship between setups and concepts, allows retrieval based not only on classes of objects, but also on events.

In this chapter we present BEKAS[1], a CBIR architecture that combines low-level features, spatial information and semantic labels, along with a concept hierarchy. BEKAS is related with the completion of our second objective (Section 1.3.2). Compared to existing related work, in BEKAS *i*) an algorithm for meaningful segmentation is not required, *ii*) a concept hierarchy can be used to support high level knowledge, *iii*) local histograms are used to attach semantic labels to images, and *iv*) the Spatial Dependency matrix is proposed to represent spatial relationships in images or groups of images.

In the following Section (4.1) we define a number of targets, with respect to the features that we intend to use in the process of image similarity, along with a short overview of the system. In Section 4.2, the *knowledge structure* is presented, with respect to what kind of information is stored, and how. In Sections 4.3 and 4.4 we discuss the problems of training and use of the system. Training is regarded as the generation of the actual structures and their population, while use is concerned with how queries are formed and executed by the system, along with now the images are retrieved.

The Chapter concludes in Section 4.5, where we discuss issues regarding the implementation of BEKAS, and the effects of possible errors at different levels of the process.

[1] *Bekas* is the name of a famous character in a Greek 80's TV series. He was a very capable investigator that eventually solved the mystery of the series.

4.1 BEKAS: Targets and Overview

4.1.1 Targets

Before anything was attempted towards implementing BEKAS a set of architecture targets, with respect to exploited technologies, were set as follows:

Use of histogram features. It has been shown in the literature [106] that histograms are capable enough of capturing low level image content of different aspects, while moderate combination of features has been proven beneficial (as showin in Chapters 3 and 6 respectively) [38]. Experiments with the COIL library showed from the early steps of CBIR that histograms can cope quite well with images of individual objects. Additionally, histograms have been used on image partitions to extract localised features with some success [112][103].

When the histogram analysis is focused on small windows, the amount of information used to generate a histogram is reduced. This can be translated to potentially more distinct histograms (assuming that the complexity of the pictorial data remains low). Effectively, the localisation of the histogram analysis is potentially reflected by a more distinct feature space. In addition to this, we have implemented a number of histogram features that may capture positive localisation behaviour, and therefore we want to use as many of these features as possible.

Active use of spatial information. Different spatial arrangements of the same objects put images to different contexts (not only describing places but also events). It is therefore, important to extract and use spatial information. In Chapter 3, we introduced a method to include spatial information into histograms. We think that such a technique is not capable of providing spatial information, explicit enough, to differentiate between different contexts. A different approach should be used to achieve that goal.

Active use of semantic information. This target can be broken down into two steps, as follows:

- Association of semantic information to images. This effectively means that we need to classify regions of the image and attach semantic labels. A method is needed to break down the image into regions before the classification takes place. Although, methods to perform automatic (and potentially meaningful) segmentation to a reasonable degree currently exist, their robustness or their specialisation is such that it is out of the scope of use in BEKAS. We attempt this by partitioning the images to rectangular windows and treating each one individually.

- Use of a concept hierarchy. Using semantic labels attached to images is not much different than using keywords to perform CBIR. A dynamic structure can be used to place these labels into a semantic space. This way similarity can be extended from low level features and spatial information to the semantic level. The benefits of such an approach are discussed by Tansley *et al.* in [115] and later in Dobie *et al.* in [26].

4.1.2 Overview

If we ignore the actual image database, BEKAS consists of three main components *i*) a training module, *ii*) the knowledge structure, and *iii*) the query module, (as illustrated in Figure 4.1). The training module is in charge of *i*) gathering knowledge to enable associations between semantic labels and low level features, and *ii*) to collect knowledge about the spatial arrangements of semantic labels. The information generated by the training module is stored in the knowledge structure, in a predefined arrangement, so it is linked to a concept hierarchy. The information in the knowledge structure is used by the query module to process queries at a semantic level. The dashed arrow that goes back from the query module to the training module denotes a possible feedback loop, where the training modules update the knowledge structure.

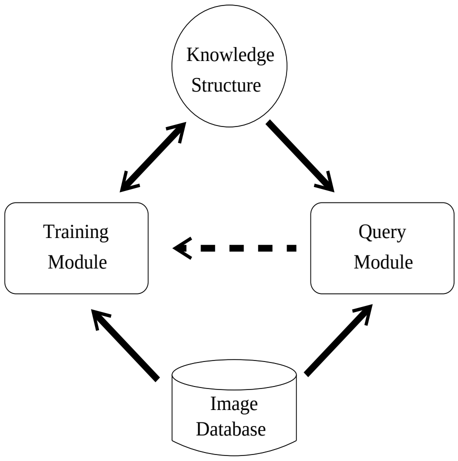


Figure 4.1: Bekas overview.

4.2 BEKAS: Knowledge Structure

The knowledge structure has three levels of information (Figure 4.2). Semantic labels comprise the lowest entry level to the knowledge structure, and they are associated with low-level features (histograms). In the middle layer, spatial arrangements are used to store semantic label configuration details and are used as a link to the higher level of the knowledge structure, the concept hierarchy.

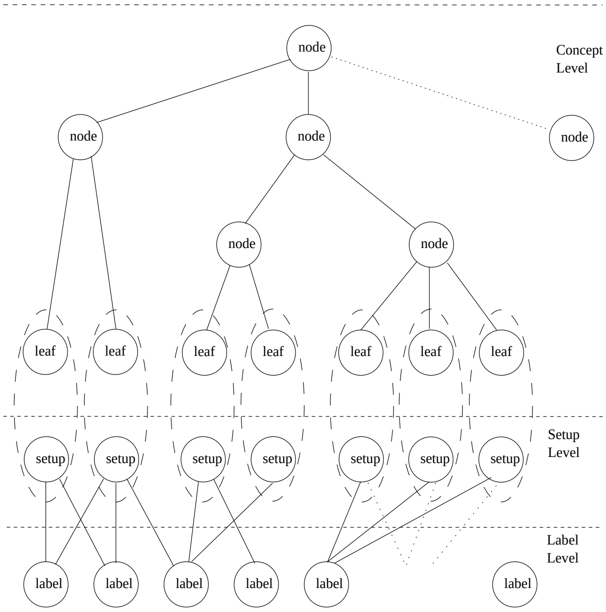


Figure 4.2: Levels of the knowledge structure.

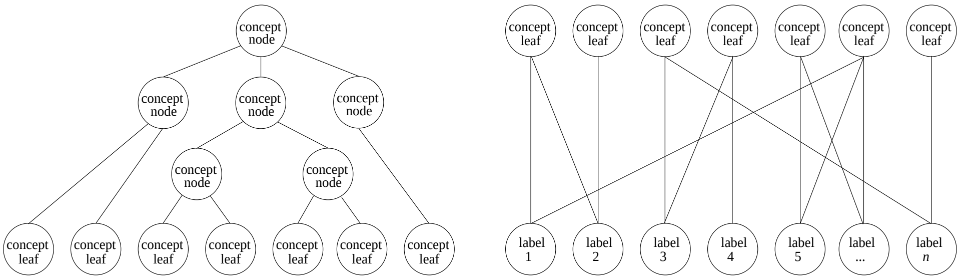
In this section we discuss the top two levels of the knowledge structure. Discussion of this section is limited to their role in the knowledge structure.

4.2.1 Concept Hierarchy

Concept hierarchies are used to organise data and concepts in a hierarchical form or some specific partial order. This way knowledge and data relationships can be expressed using high level terms. Therefore, concept hierarchies play an important role in the process of knowledge discovery.

At the bottom end of the concept hierarchy (as used in BEKAS, Figure 4.3a.) a label composition is attached to the concept leaves. This way a link between the concept hierarchy and images is achieved, using the semantic labels attached to the images, Figure 4.3b.

The benefits of using the concept hierarchy are two-fold:



(a) Concept hierarchy example.

(b) Concept composition.

Figure 4.3: Instances of two parts of the knowledge structure.

- Similarity between images can be quantified at a semantic level, using the semantic relationships of concepts. The use of the more general term *concept* (instead of just object[2]) means that the structure can be more flexible with respect to the nature of stored semantic content (e.g. nodes can represent events). The information we intend to use for identification of such concepts in the hierarchy, is the spatial information of semantic labels in images.

- Flexibility of the search is increased while reducing restrictions of user queries, and the system can be enabled to handle inaccurate user queries. That can be achieved by expanding the concept hierarchy to include synonyms.

As the concept hierarchy comprises a dynamic structure, it can be updated or rearranged as desired, to accommodate more semantic information and to expand its label specification.

There has been already a great volume of research on the area of concept hierarchies. Additionally, similar structures, have been used for CBIR [115]. For these reasons we present the concept hierarchy as a means to store incompatible semantic information (e.g. object name, event, season *etc.*).

4.2.2 Spatial Arrangement (Setup)

Spatial information can be extracted from images and has been used in the literature as an extra feature for indexing. Two different approaches are common. Through partitioning and histogramming [38][112][117], the spatial information is implicitly extracted along with other features. This approach allows passive use of the extracted spatial information, while no explicit access to this information is available. Alternative methods, represent

[2] In this context object is the semantic label attached to an image region (e.g. sky, snow, ball, car *etc.*).

the spatial relationships of objects or regions as graphs (region adjacency graphs [109]). Graph matching techniques are used to quantify similarity, while the explicit use of spatial information allows variable degrees of matching tolerance (more flexibility).

Spatial arrangements of objects (*setup*) in an image is important so one can put the image into a context. Relationships between sets of labels and concepts are made stronger, so ambiguities due to relationships of multiple concepts to a single labels set can be reduced or resolved. The spatial relationships can also help identify variations of a single concept, if that is desired.

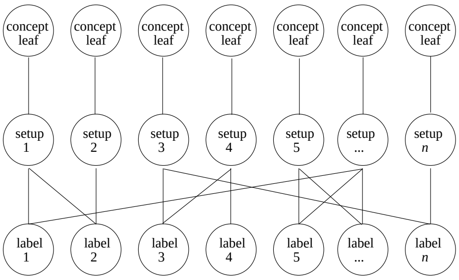


Figure 4.4: Setups.

The concept composition in the knowledge structure (Figure 4.3*b*) can be updated so spatial arrangement of labels are represented, Figure 4.4. This change allows the knowledge structure multiple means for creating relations to high level concepts (by arrangement and/or by composition).

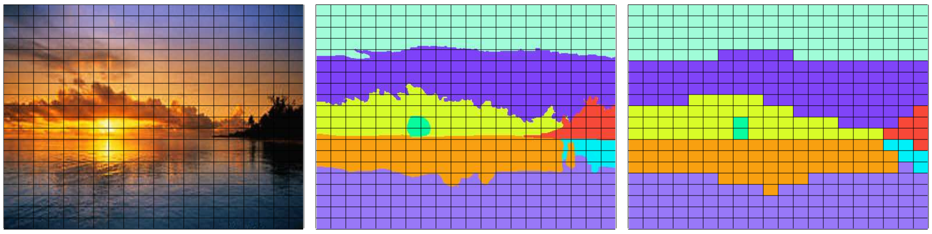
4.3 Teaching BEKAS

The starting point of building the basis of BEKAS is the construction of the knowledge structure. At the lowest level of the structure semantic labels are associated to histograms. At the highest level, a concept hierarchy provides relationships between various concepts and/or contexts. The lifecycle of BEKAS starts with a training process that is expected to lead to a usable knowledge structure. This training process is split into three phases, each one related to a different level in the knowledge structure. Discussion in the section is focused on the methodology used to achieve the target functionality of each level.

4.3.1 Semantic Labels

With a meaningful segmentation one could extract all the objects from an image to compose a desired semantic label set for that image. Meaningful segmentation is not realistic [45], so extracting all the objects from an image is not a viable approach. Our approach is inspired by the work of Feldman and Yakimovsky [31] where the segmentation assumption is replaced by a region labelling process that is based on *a priori* knowledge. Our aim is to model a mapping between histogram features and a predefined set of semantic labels. This problem falls into the category of classification and, given availability of *a priori* knowledge, it can be tackled in an automated fashion.

The training process starts with a set of training images. These images are given to human subjects for segmentation. Multiple segmentations are used so the classification is not biased by the judgement of a single person. To average the segmentations we used majority voting.



(a) Original. (b) Segmentation. (c) Label map.

Figure 4.5: Example of going from image to partition map.

Then the images, and their corresponding segmentations are partitioned to $M \times N$ rectangular sub-images (or windows), Figure 4.5. Histograms are generated from each image window, while a single semantic label is assigned to each segmentation partitions (by majority voting at the corresponding segmentation window). These $M \times N$ semantic labels form what we call the *label map* which is also used at a later process.

Each histogram generated from the training window pixels is associated with the corresponding single target semantic label. The collection of histograms and the target semantic labels, is passed to the classification module for learning, Figure 4.3.

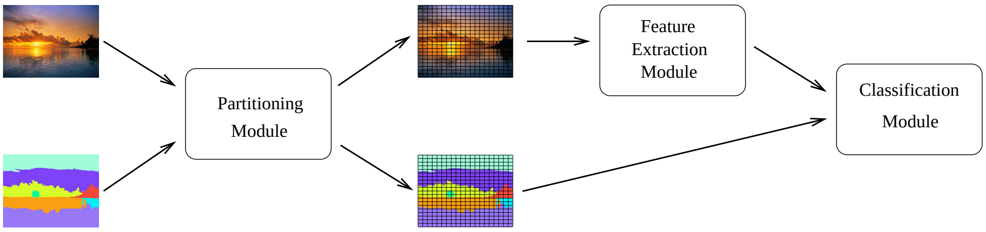


Figure 4.6: Overview of the semantic label learning process.

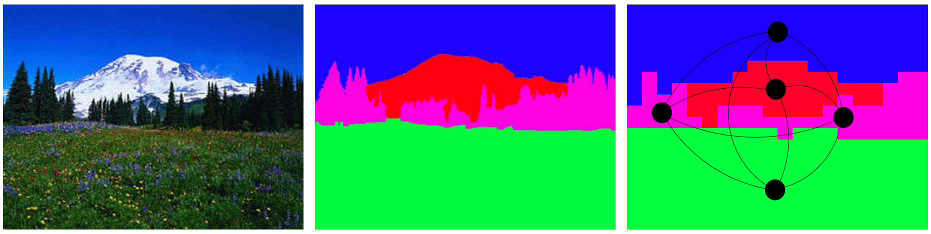
Two issues are obvious in this process:

- *Partition size and type of histograms*: The required quality expected from histograms useful in this process is good localisation potential. The choice of partition size is also important and it should be such that usable histograms can be extracted, that is, the histogram space can be discrete.

- *How the training data is fed to the classifier*: The number of histogram bins plays a major role in the complexity of the classification process. By summarising the histogram data the classification process is simplified, with some loss of accuracy. One method is to store a finite number of histogram moments [48]. The first five moments should be enough and capable of describing the originating histogram.

4.3.2 Spatial Arrangement

The different regions of a segmentation can be enumerated and the relative positioning recorded to generate a graph. In Fig.4.7, an example of this process is illustrated. The nodes of the graph, Fig.4.7c, can represent the various region properties (e.g. label, size etc.) and the arcs represent the spatial properties between the nodes (e.g. relative positioning).



(a) Original. (b) Segmentation. (c) Label map and graph.

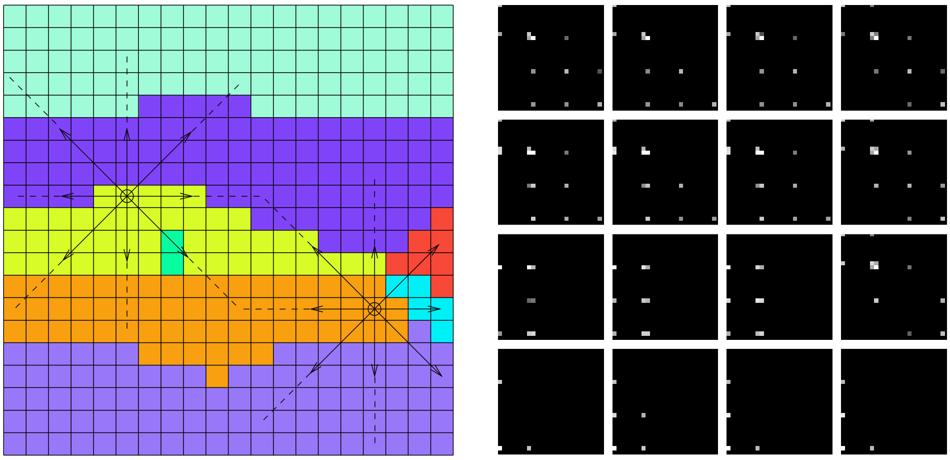
Figure 4.7: Example of going from image to partition map.

In BEKAS spatial arrangements of labels are extracted from the label map of the training images. We preferred an alternative representation to the a graph, that we call the *spatial dependency matrix* (SDM).

A spatial dependency matrix is essentially a two-dimensional histogram where the frequencies of label pairs positioned at a given orientation are registered,

$$C(i,j|\phi,d) = \#([\mathbf{c},\mathbf{p}]|L(\mathbf{c}) = i\,, L(\mathbf{p}) = j,\ \mathbf{p} = \mathbf{c} + d(\cos\phi, \sin\phi)) \tag{4.1}$$

where $\mathbf{c}$ and $\mathbf{p}$ are points in the label map. Given an orientation the $\mathbf{p}$ points at d (where $1 \leq d \leq \sqrt{w^2 + h^2}$) are considered. $L(\mathbf{x})$ returns the underlying label at a point $\mathbf{x}$. Spatial dependency matrices at orientation $\phi = \{0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o, 315^o\}$ are calculated to collect spatial information for the eight major connectivity types. In fact, only the first four ($\phi = \{0^o, 45^o, 90^o, 135^o\}$) are needed, due to symmetry. The range of distance d is quantised at four levels to increase the level of detail. SDMs calculated for every combination of distance d and angle ϕ, (4×4) are combined producing a setup representation. In Figure 4.8, an example of a setup representation of a training image is illustrated.



(a) Region Look-up. (b) Spatial Dependency Map.

Figure 4.8: Instances of tracing for the generation of a spatial dependency matrix. In (b) 4 orientations (left-to-right) and 4 distances (top-to-bottom) are shown.

Spatial dependency matrices of multiple images can be combined and normalised. If this normalised form of the SDM originates from similar images, representing a single concept, they can be used as setup templates.

To compare single SDMs, any similarity measure mentioned in Chapter 2 can be adopted. Similarly, to compare the setup of an image against a setup template, we just need take only the matching semantic labels into account. For the current implementation of BEKAS we use the Euclidean distance, as follows

$$D_{\mathbf{xT}} = n\frac{\sum\limits_{k=1}^{\phi}\sum\limits_{j=1}^{d}\sqrt{\sum\limits_{i=1}^{n}(\mathbf{x}_{kji} - \mathbf{T}_{kji})^2}}{N^4} \tag{4.2}$$

where $\mathbf{x}_{kji}$ and $\mathbf{T}_{kji}$ are the values of the SDMs for the matching semantic labels, N is is the total number of semantic labels in the setup template, d is the number of distance levels, ϕ the number of different angle steps, and n the number of matching labels.

4.3.3 Concept Hierarchy

Concept hierarchies can be provided by knowledge engineers, domain experts (or users), or derived automatically from data relations. While automatic generation of concept hierarchies is possible [28], it is outside the scope of this approach.

For the purpose of this study a concept hierarchy had to be generated manually and specifically in the context of BEKAS and CBIR. The aim is to initially generate a very simple concept hierarchy, using the theme of natural scenes (Figure 4.9), which later can be expanded.

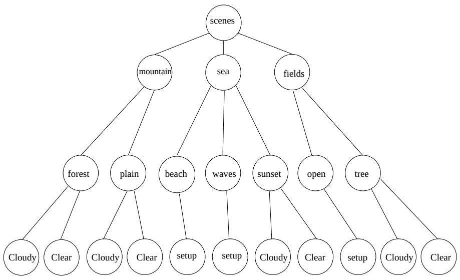


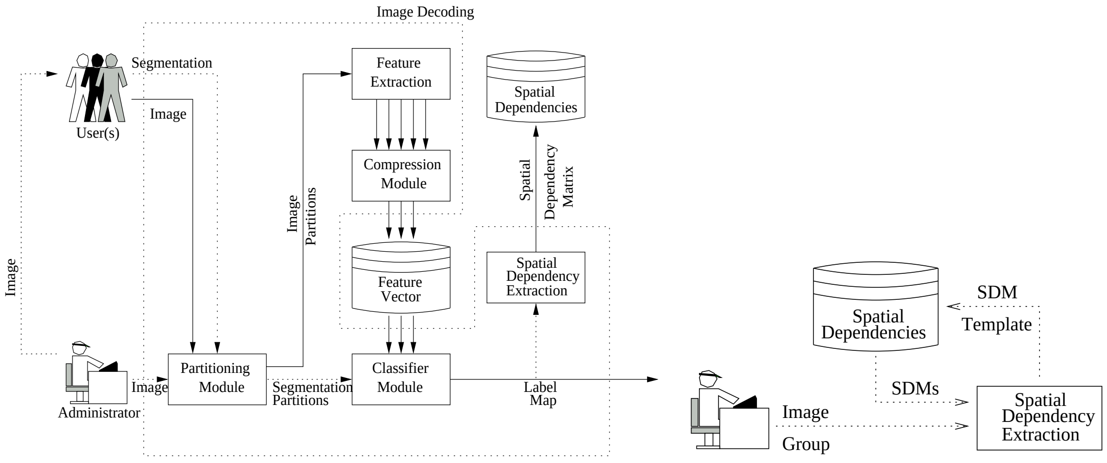
Figure 4.9: A simple concept hierarchy uses with BEKAS.

The leaves of the concept hierarchy are associated with setup templates representing two weather characteristics, *i*) clear sky, and *ii*) sky with clouds (that is wherever we have provided example images).

4.3.4 Teaching, Summarised

In the previous Section (4.2) we discussed the function of the knowledge structure in BEKAS, and in this section we presented how this structure is populated with information. The process of teaching BEKAS consisted of semi-automatic and manual sub processes.

The generation of *i*) *the low*-level semantic knowledge, and *ii*) the spatial dependency matrix is a semi-automatic process as the needed segmentations are provided by human subjects. After training is finished, the same modules can also be used to generate label maps and SDMs from images not exposed to the system before, and for cases where a segmentation is not available (e.g. query-by-example). In such cases the modules' function is fully automatic. An overview of the process is shown in Figure 4.10, where the sharing of resources is illustrated.



(a) Image Decoding

(b) SDM template

Figure 4.10: Overview of image decoding and generation of the SDM templates. Solid arrows indicate the typical data flow while the dotted arrows indicate the data flow required while training.

The process of creating the concept hierarchy and the linking of the SDM templates is manual, although we believe that it may be automated. Algorithms exist for automatic generation of concept hierarchies, therefore the remaining bottleneck is on the association with the SDM templates. Currently we do not have any evidence of an automatic method to link spatial configurations of images to a concept hierarchy, though it may be possible in the future. One possibility may be to use the spatial information along with the semantic relation data in the process of generating the concept hierarchy.

4.4 Questioning BEKAS

In the previous sections we showed how knowledge is stored in BEKAS using the three-layer knowledge structure. In this section we show how this structure is exploited to perform queries in BEKAS.

First the search process is outlined, followed by a more detailed discussion of the possibilities of query types that can be supported by BEKAS.

4.4.1 Searching

Performing a query with BEKAS involves a sequence of steps, as follows:

- *Query analysis.* The information submitted by a query is analysed to extract information compatible to the knowledge structure.
- *Starting point of search.* Depending on the information provided by the query, one or more points are selected in the knowledge structure, to initiate multiple searches.
- *Search paths.* For each starting point searching is performed, in terms of reaching the lowest part of the knowledge structure. If necessary, the search is also directed towards the root of the concept hierarchy, to search similarities at a higher level. Eventually, a set of paths (from a starting point to a spatial arrangement setup) is generated and sorted by their depth (number of steps). The depth of a path defines the similarity to the submitted query.
- *Define image sets.* The destination of each path (that is the leaves of the knowledge structure) contain information about the configuration of the related images. This information is used to define image sets, that are retrieved for each path (starting from the shortest path). In the current implementation, images are assigned to only one image set, it may be desired to retrieve all images though.
- *Image retrieval.* The user initiates image retrieval by browsing the image sets defined in the previous step. The images of each set can be sorted using any available resources (similarity to semantic label composition, arrangement, proportions *etc.*).

In essence, searching is nothing more that traversing the knowledge structure. Because the structure is composed of high level concepts, they also comprise the driving force of the search. Therefore, the user is exposed to images grouped based on their semantic properties. The process of questioning BEKAS is illustrated in Figure 4.11.

4.4.2 Query Types

The architecture of BEKAS allows exploitation of two kinds of queries, *simple* and *composite*. Simple type queries target a single layer in the knowledge structure, to find the

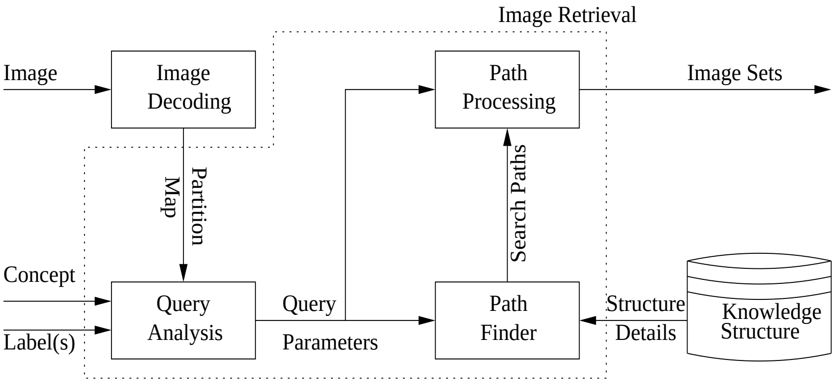


Figure 4.11: Query processing.

starting point(s) of the search. On the other hand, composite queries comprise an extension of the simple queries, by using information related to more than one of the levels of the knowledge structure. The information in composite queries can be used in such a way that a query language is possible (e.g. exclusion of semantic labels or configurations using logical operators). The focus of this section is on the execution of simple queries.

Query by Example. The expected input for queries of this type, is an example image (existing file or user sketch). This image where it is partitioned and the features of each partition are extracted, compressed and passed to the classifier. A semantic label map is generated and an image signature is calculated (SDM and label proportions). One starting search point (or more if desired) is defined by finding the SDM template that matches best the SDM derived by the example image. Then the search process continues as described in the previous section.

A simple example of the search process, from the point we have found a starting point (at the setup level) is illustrated in Figure 4.12. In this example the starting point is node (h), that will be used as a path of zero depth. Then, the search is directed up to the parent node (c) and then down to the remaining children of that node (g) and (i). Then the search is directed up again, so the children of the parent node (a) are visited. The resulting path set from this particular example would be in the following order $P = \{h, g, i, e, f, j, k\}$.

For cases with multiple starting nodes, a solution would be to score the starting points by how much the query SDM matches the SDM template of the starting point (the closer the matching the higher the score). Then the search can progress for all the nodes as normal, until a conflict occurs. In the case of a conflict the progress of the path that started from the node with the least score is terminated. When the search progress of all

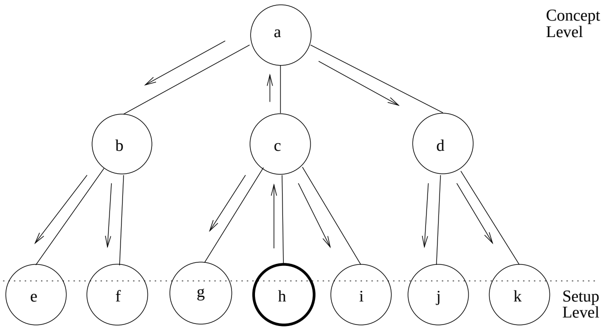


Figure 4.12: Query by Example or Spatial setup.

the paths is finished the process continues as described in the previous section.

Query by Spatial Setup. In this case the input can be given in two forms either by example, where the user provides a drawing similar to a segmentation (from which a label map is produced), or by access to the knowledge structure (e.g. selecting a setup from a list, exposing the knowledge structure to the user). The remaining of the search process is identical to the case of query by example.

Query by Concept. In such a type of query the BEKAS expects a concept name as an input. The starting point is defined by a lookup in the hierarchy. Then the search is directed towards the setup level first, and then up to a higher level concept. A simple example of a query by concept is illustrated in Figure 4.13. Starting from the concept node (c) the search is directed to nodes (g), (h) and and (i). Then is directed up to the parent node (a) for further search. The resulting path set from this particular example would be in the following order $P = \{g, h, i, e, f, j, k\}$.

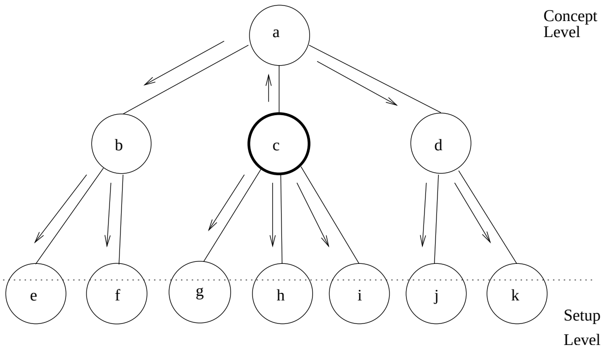


Figure 4.13: Query by Concept.

Query by Label(s). One or more labels are given to the system, without any spatial information (An alternative form of input could be an example image, that is treated as a single partition). This type of query results in potentially more than one starting points.

That is because all the setups of the knowledge structure that include the given labels are used as starting search points. From that point the search progresses as in the previous types of queries. An example of such a query type is illustrated in Figure 4.14. Starting from two labels ($L3$ and $L4$), the resulting path set is $P = \{g, h, i, f, e, j, k\}$. If the query is restricted to return images that contain *only* the labels included in the query the path set would be $P = \{g, h, i, f\}$.

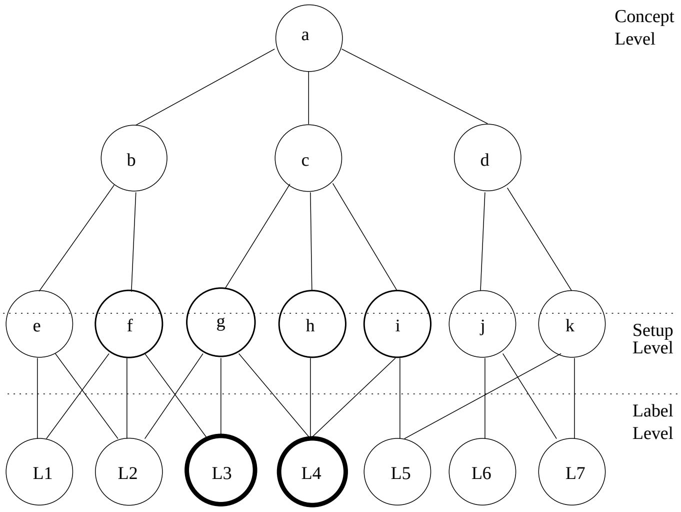


Figure 4.14: Query by Label(s)

4.4.3 Summary

In this section we presented in more detail the types of queries that can be supported by the BEKAS architecture and we showed examples of how these queries can be generated.

The main feature of query processing in BEKAS is that the distance measure is based on semantic information, rather than computationally convenient features. Furthermore, the semantic information is used to generate groups of images sharing the same semantic properties. The significance of ranking of images within a single group is not defined by the developer of the system, but by the user of BEKAS (depending on the amount of detail in a submitted query).

While the illustrated examples show very simple concept hierarchies, it is possible to use the same techniques for deeper structures. A natural extension of the knowledge structure would be to include synonyms of concepts, while the ultimate goal is the use of a thesaurus, in a similar manner as in [115].

4.5 Implementation Issues

In this section we discuss implementation issues of various components of BEKAS as well as the possible impact of error at different stages of training and query processing.

4.5.1 Semantic Labels

In this section we discuss issues regarding decisions taken with respect to the choice of the classification algorithm and the choice of parameters. Additionally we discuss the implications of errors at this level, and possible approaches to remedy these errors.

***Choice of classification algorithm* and parameters.** Two methods were attempted, *i*) Decision Trees, and *ii*) Multi-Layer Perceptron (MLP) and error back-propagation learning. Preliminary testing showed that both methods performed very similarly for this type of problem. The MLP method was preferred due to implementation conveniency.

Before feeding the training data to the classification module we applied majority filtering to each partition. This way slight segmentation errors are filtered, while a confidence level can be established to decide which partitions are suitable for training. Additionally, an amount of overlap is added to the partitioning algorithm to fuzzify the process. We choose a 75% coverage as the confidence level, and 10% overlap between partitions.

After testing the system with a range of MLPs of different structures, we found that the best accuracy performance we could achieve for labelling was in the range of 60-70% successful classifications of partitions. A reason for this may be the nature of the information used by the classifier, which may be improved by using a higher level summary of that information.

Effects of erroneous classification. Misclassification of partitions, result in errors in semantic label maps. Semantic label maps are used *i*) to generate spatial dependency matrices, and *ii*) to calculate the proportions of semantic labels in an image. Therefore, errors in the semantic label map would potentially affect the setup-templates stored in the knowledge structure, and the SDMs generated for images submitted for queries-by-example.

There are two cases we need to discuss, when the classification of the semantic label maps used for the generation of the setup-templates, *i*) had errors (the classification module was used to update the setup-templates), and *ii*) had no errors (only hand segmented images were used). In the first case, the impact of mis-classifications, would be severe, to a degree that the system could not be used with confidence. That is because errors would affect both the existing setup-templates, and the semantic label maps generated for query and new images.

In the second case, the system would only be affected when generating semantic label images, and the process of comparing the SDMs of an image against the setup-templates. Two possibilities exist, *i*) the assigned erroneous semantic label that exists in the setup-template, and *ii*) when the assigned erroneous semantic label does not exist in the setup-template. Assuming that these errors are isolated and small in number we expect to have little effect. That is because the distance measure takes into account the distance and direction characteristics, as well as the membership of a semantic label in the setup template. In addition to this, queries that do not use the classification mechanism would not be affected.

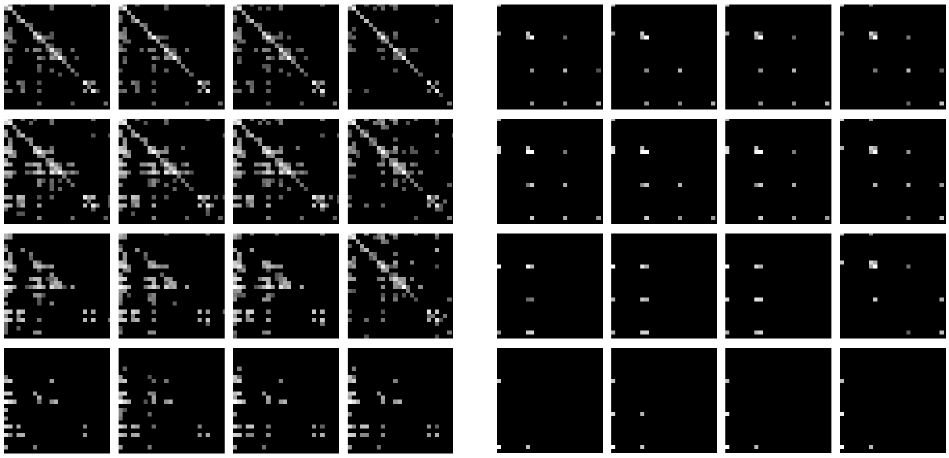
Possible remedies. We have considered two possible remedies, *i*) simplify the classification problem, and *ii*) post process the classification results.

Simplification of the classification problems may be achieved with the use of a different method for summarising the histogram features. In [70], Mojsilovic and Rogowich experiment with mapping of features to semantic categories. By exploiting a similar method more distinct variables may be possible, therefore making the process of classification more accurate.

Alternatively, relaxation techniques [109] may be used to post-process the classification results. An additional processing step of the setup-templates can be used as *a priori* knowledge to the relaxation labelling. That is to summarise and normalise the setup-template to a single structure, from which we can retrieve the probability of label pairs to be positioned at a given orientation and distance (Figure 4.15).

4.5.2 Representation of Spatial Relationships

Another decision had to be taken with respect to representing the spatial relationships of semantic labels. Initially we experimented with a graph representation and the Bron-Kerbosch clique finding algorithm to compare the spatial relationships of two images.



(a) Global setup-template

(b) SDM of a sigle image.

Figure 4.15: Example of a global template in contrast to SDMs of a single image.

We found difficulties implementing the algorithm so to allow comparisons of graphs from images against template-graphs. The basic comparison algorithm returns only the matching pairs of nodes (between two graphs). The result of this was the spatial dependency matrix as is introduced in Section 4.3.2.

The process of generating SDMs is simpler than the graph method, while the main advantage is the method used to compare SDMs. The main advantage is that more information is accounted for the comparison process, rather than just matching information. Furthermore, the process of comparing SDMs is also simple as a wider choice of distance measures can be adopted and used to experiment.

The proposed representation of the spatial dependency matrix is not expected to scale very well for hundreds or thousands of semantic labels. In such cases alternative methods should be considered (e.g. sparse matrix representations).

4.5.3 Query Processing

On the weak side, the process of generating the search paths is heavily dependent on the structure used to represent the relationships of semantic knowledge.

Errors in selecting start search point(s). There are three possible starting points in the knowledge structure, that is, from top to bottom:

- *Concept.* The effects of errors at this level depend on *i*) the depth of the concept hierarchy, and *ii*) the source of the concept hierarchy. In the first case, the error mainly depends on the distance between the mis-selected concept and the correct concept. The error is amplified by the depth of the concept hierarchy. Small errors at high levels in the hierarchy should have a greater impact in performance. In the second case, errors may be due to incorrect connectivity (relationships) in the concept structures. These errors are introduced when the structure is composed, and their effect in performance is more or less unpredictable. In this implementation, we are not concerned about such errors for two reasons, *a*) the search is a straight forward lookup through the structure, and *b*) the size and structure is simple.

- *Setup.* A erroneous selection of a setup is expected to have the same effect as in case (*i*) for errors in concept selection. The reasons for such errors to occur are *i*) misclassification in the labelling process, and *ii*) capacity limitations of the setup-template. In the first case, errors may be reduced by improving the classification process, while in the second case, the template may be over-generalised and a split may be required. Errors of the latter kind do not concern the current implementation, as the number of possible setups is limited.

- *Label.* Errors at this level are expected to appear during the classification process. Their effect and possible remedies are discussed in the previous Section 4.5.1.

Dependency to concept hierarchy representation structure. In this implementation of BEKAS we choose a tree structure to store the concept hierarchy. Scaling of the concept hierarchy to accommodate more information and more complex relationships, would require, redesign of the query processing methods. In particular, the selection of start search points for query-by-concept, and the path generation process would require changes.

Ranking of images within the returned image sets. As we mentioned, the query process returns a number of image sets, that the user can browse through. Depending on the type of query, the amount of information about the desired target image is different. The method used to rank the images within a return image set, depends on that information. A list with the method used for each type of query follows:

- Queries-by-example and -by-spatial setup, are the most information rich types of queries. From the input the system can extract information about semantic label membership, proportions and spatial relationships. The same information is used to generate SDMs. Therefore, we used the SDM distance measure to rank the images within the returned image sets.

- For the cases of queries-by-label, the system is only provided with semantic label membership information. This exact information is used to sort the images within the returned image sets. So images which include more semantic labels from the query are ranked first. It is expected to get ties, and as no further information is available to the system the images are returned in random order.

- Queries-by-concept do not provide any semantic label membership information. In this case images are again ranked randomly.

4.5.4 Testing

BEKAS provides a more complex approach compared to a system extracting and using features using the methods presented in Chapter 3. Because of this the performance evaluation framework presented in Chapter 5 cannot be applied directly. Testing BEKAS poses a challenge to CBIR as multiple aspects of the system need to be tested.

In this Chapter we have presented the idea of BEKAS. All the modules have been implemented and tested individually and not as a complete CBIR system. This would require further data collection (related to the composition of the concept hierarchy, training sets for the labeling module and the setup templates) and application development (design and implementation of a user interface) which would be outside the scope of this thesis.

Chapter 5

Evaluation Framework

The main objective of a CBIR system is to ease the process of image retrieval in order to increase productivity. Therefore, it is essential to quantify the effectiveness of a system as a measure to assess its usability. This is a challenging task in CBIR, due to the subjectivity factors involved in the process.

Extensive subjective evaluation would be the perfect solution to this problem. But due to difficulties (even when it comes to narrow down the application area, subjectivity is a major difficulty [59]), time consumption and inability to be reliably repeatable, subjective evaluation should be left for the last stage of the CBIR system development.

In testing low level computer vision algorithms it often appears that an author jumps to a higher level, using an application, in order to evaluate objectively the performance of the proposed method (*e.g.* [89]). As CBIR is an application by itself, it is not easy to follow such an approach without promoting simplification and possibly diverge from the main problem.

In [98], Santini mentions the three different modalities of evaluation of a perceptual and interactive system *physical performance*, *contextual* and *decontextual evaluation*, while emphasis is on decontextual evaluation. The nature of a histogram based CBIR system and parts of the above work was the main motivation towards experimenting on different aspects of the implemented prototype system. The goal is to identify methods for automatic and easily repeatable testing and validation of histogram methods for CBIR.

Our experimentation leads to a collection of qualitative and quantitative methods that can be used to evaluate various aspects of histogram methods for CBIR. In this chapter we have organised these methods to form an evaluation framework. This framework allows methodical and easily repeatable testing, which satisfies the testing objective (1.3.3). Although some of the methods may sound self-intuitive we did not find any reference to their use in CBIR evaluation.

Discussion starts with an overview (Section 5.1) of the testing framework where the information flow between parts of the framework is illustrated. Discussion continues on the acquisition of groundtruth and the use of multiple datasets.

The remainder of the chapter is separated into *qualitative* and *quantitative* evaluation methods for CBIR. In Section 5.2, we present a number of methods, consisting of graphs and plots, whose role is to assist qualitative assessment of histogram methods for CBIR, in terms of potential performance behaviour and sensitivity to noise and image transformations. Then, in Section 5.3 we present the quantitative methods we used in this framework.

The discussion finishes with Section 5.4, where the Pathfinder networks are introduced for use in image dataset visualisation and CBIR system evaluation. The work involved in Chapter 5.4 was a joint effort and is included in this thesis for completeness.

5.1 Overview

The scope of the testing framework is to provide the means for qualitative and quantitative evaluation for histogram methods for CBIR, an overview of which is illustrated in Figure 5.1. Starting from a pool of extracted histograms and the associated groundtruth, various quantities are calculated (distance / similarity measure and performance measure). These values are then used to generate graphs and visualisations for qualitative analysis. Resulting values are also used for further quantitative analysis (statistical significance). In Figure 5.1, the main information flow between the components of the framework is also illustrated.

The focus of this section is on the input to the feature extraction methods, the images, and the associated groundtruth. More details on the remaining components of the framework are discussed in later sections.

5.1.1 Groundtruth

A collection of information about an image comprises the groundtruth for that image. This information may describe various aspects of the image at various levels. A set of tags may

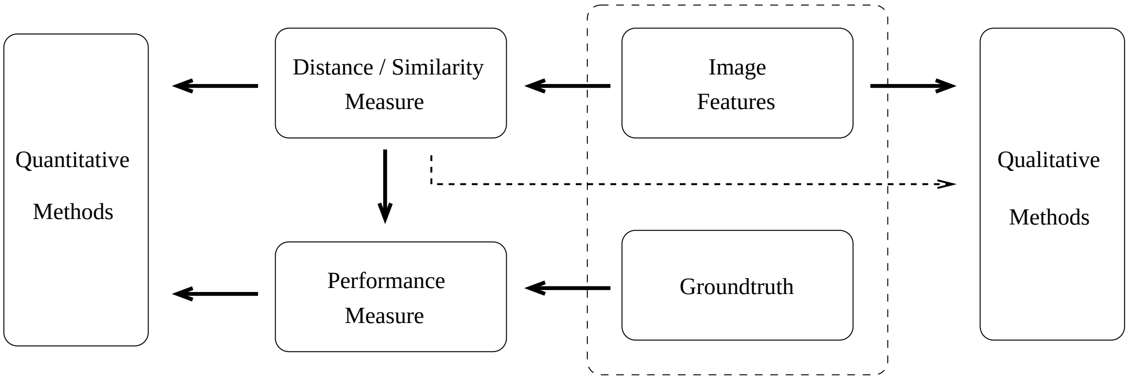


Figure 5.1: Obtaining results overview.

describe the content of the image while higher level concepts may be associated to the image. At lower level tags may be used to describe low level aspects of the image, e.g. a segmentation, and possibly an association layer that provides a spatial connection of the two levels.

In a collection of images intended for traditional CBIR (using keywords) this information is used to determine relationships between images. In modern CBIR such information is collected and used as part of the evaluation process of a method or system. The relationships between images can be summarised into classes of images, which may be disjoint or overlapping. These summaries can then be used to ease the calculation of performance measures such as *recall* and *precision*. The initial part of the process (tagging) is usually performed by human subjects, usually experts, while the image classes can be calculated automatically by using programs.

Fully automatic classification of images for groundtruth can also be achieved. Milanese (in [65]) used the fact that images belonging to the same scene to put images into classes. The relationships between images of different classes, although possible, are not captured by such a method. Still this kind of groundtruth may be used as a reference to determine relative performance between different methods for CBIR.

Recent research activity is noticed towards automatic labelling [116]. A hybrid system is built that uses feature extraction methods and the labels attached to a groundtruthed set. The low level feature similarity between the groundtruth and the new images is used to attach labels to these new images. Such interpolated groundtruth should not be used for the evaluation of CBIR systems as the accuracy error introduced by the automation may compromise the results.

As the whole process of groundtruthing is very subjective and the accuracy of the groundtruth is questionable (no matter how strict the tagging process is), we think that testing based on such groundtruth should be used only for reference performance evaluation and not as a tool for assessing the usability of a CBIR system. A full scale subjective test is more suitable of capturing the usability characteristics of a CBIR system.

A variable number of image datasets is important in testing a CBIR system. This way the system can be tested in various scenarios and so evaluate the behaviour under specific circumstances.

5.1.2 Images Sets

Collections of images of various content and associated groundtruth comprise the usual tool of a testing framework. Having different datasets with different levels of complexity allows isolation of the system (individual components or as a whole) to specific scenarios. In this framework, four different image sets are used. The classes were automatically created (apart from one case) by information of different semantic levels. In all cases the classes are disjoint.

Video Scene Samples We recorded scenes from broadcasted TV[1] programs on video tapes. Then we took 9 frames from each scene. A total of 43 scenes were stored giving a total of 387 images. The content continuity over a scene and the fact that each group of samples comes from a different scene provides a sound automatic ground-truth, similarly to [65]. Example images are shown in Figure 5.2.

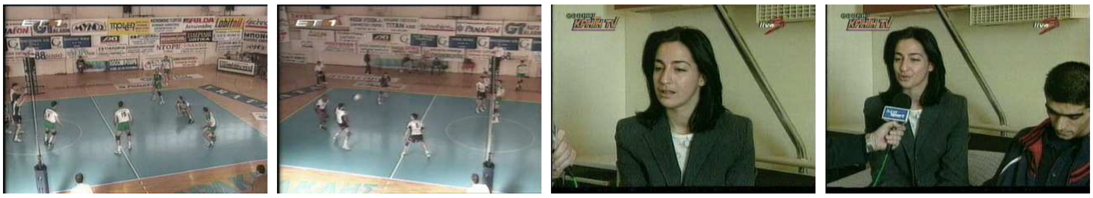


Figure 5.2: Example images from the video scene samples.

Chen Collection A total of 279 images were collected from the Internet and then grouped by human subjects. The groundtruth was used as granted and we are not aware about any further details related to the method used to summarise the subjects' groupings.

[1] The sequences were recorded under the permission of CRETA channel in Crete, Greece.

Example images are shown in Figure 5.3.

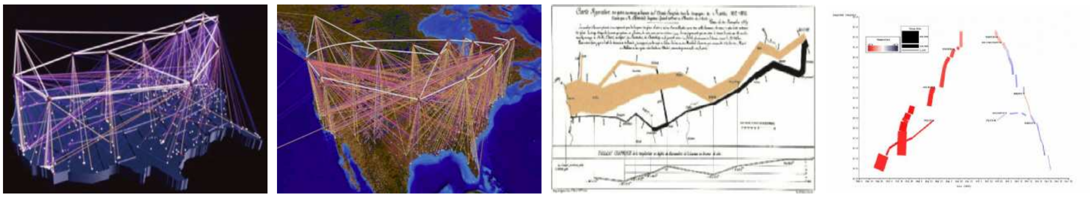


Figure 5.3: Example images from the Chen collection.

Coil-100 This particular dataset comes from [76]. It has been used extensively in the CBIR community and its inclusion in this framework was to provide a point of reference for the testing. The dataset includes 20 instances of 100 different objects, rotated around the Y axis. Example images are shown in Figure 5.4.

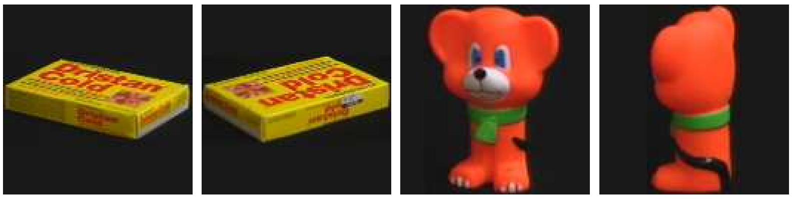


Figure 5.4: Example images from the Chen collection.

Artchive A collection of painting photographs (grouped by artist) was provided by the Patron Program of the Artchive (*http://www.artchive.com/*). The original grouping by artist was used directly as the groundtruth. The task of image retrieval on such a dataset with such abstract information is expected to be very difficult. It is expected that pictures from a single artist vary in content. Some aspects of the artist's style are expected to remain the same, while there is expectation to vary in time. Example images are shown in Figure 5.5.

Figure 5.5: Example images from the Chen collection.

5.1.3 Transformation Sequences

It is important during the development of a CBIR system (new methods, tuning existing methods etc.) to investigate the behaviour of a method under controlled conditions. Typically we are interested to study the tolerance of histogram methods under specific transformations, *rotation*, *translation* and *scale*.

For this purpose we set up a camera that could follow a predefined path. This enabled us to capture sequences of the desired transformations. Each sequence consisted of about 600 to 900 images. A smaller number would probably be enough to perform experiments. We choose a higher sample rate to enable us to control the amount of change by subsampling the sequences. Sample images are illustrated in Figures 5.6, 5.7 and 5.8.

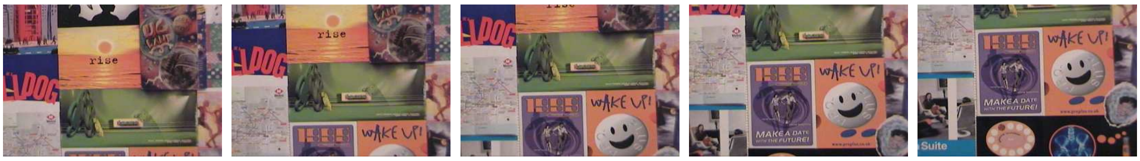


Figure 5.6: Translation on the y axis.

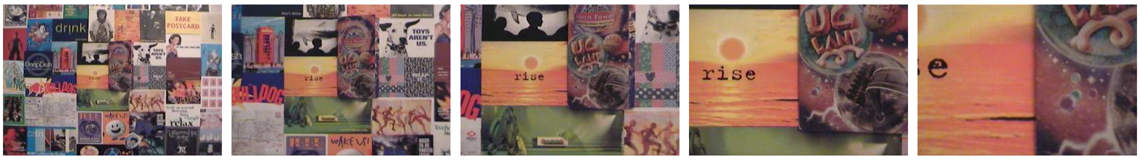


Figure 5.7: Zoom-in.

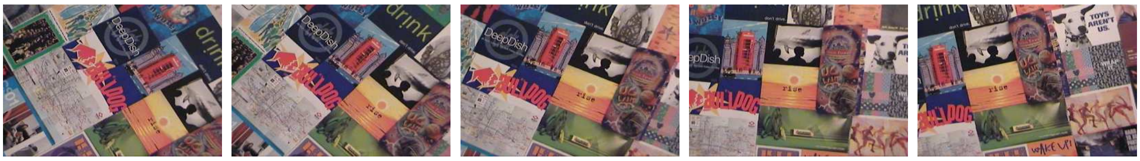


Figure 5.8: Rotation on the z axis.

5.2 Qualitative Methods

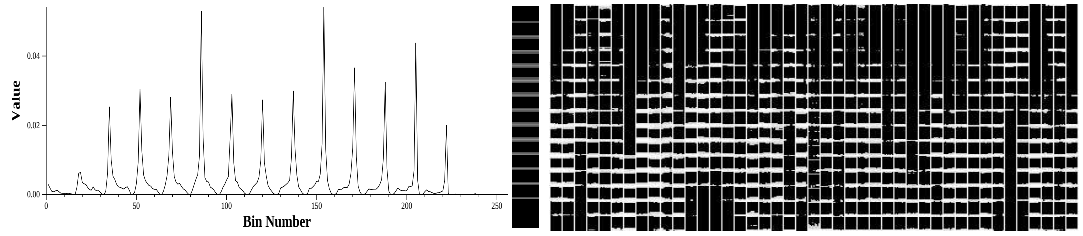
Decontextual evaluation is defined as performing tests on a system outside a specific context. It is desirable to study the behaviour of a method (a low level feature extraction routine) before its integration to a greater system.

In the CBIR context, various components, like feature extraction methods, distance measures or combinations of histograms can be considered for decontextual evaluation. Our objective is to study histogram methods with respect to their strengths and weaknesses.

In the following sections a series of tests and visualisations are described as part of the decontextual evaluation in this framework.

5.2.1 Histogram Plots

The objective of this method is to generate a visualisation of the histograms generated from a image data set using a single histograms. The values of each bin in a histogram are translated to pixel intensities (Figure 5.9*a*). These strips of pixels are generated for each extracted histogram and then used as columns to construct an image, Figure 5.9*b*. The illustrated histogram plots are post-processed for better viewing.



(a)

(b)

Figure 5.9: Histogram plot example, using the video scene samples set.

The use of this visualisation is separated into two different sections, when generated from *i*) grouped image collections, and *ii*) video sequences (controlled transformations).

Image Collections[2]. When investigating the histogram plots generated from the grouped image collections, we are looking for the following cases:

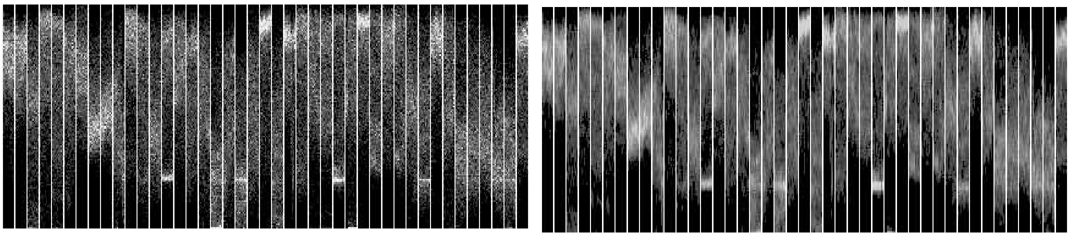
Global Level Uniformity. When very little changes across the whole range of the image collection the expectations for the particular method should be low. This uniformity denotes a limited discrimination power.

Narrow Band Bin-Population. When a small part of the histogram bin area is populated, through the whole range of the image collection, Figure 5.11a. Methods resulting to such histogram plots are not expected to show good performance, specially if there is very little variation of the populated bin range. Changing the bin membership function or rescaling the histograms may be a solution to reduce the negative effects of narrow banding. This solution may not show any improvement in cases of very noisy histograms.

[2] White lines in the illustrations of this section, are used to highlight the limits of image groups.

Noise. Regardless of similarity of bin range population within or between image groups, noisy histograms are not desirable. The source of the problem may be the bin membership function. In such case tuning, more advanced histogramming methods (like [56]) or distance measures may be beneficial. Generally, methods that exhibit noisy histogram plots are not expected to perform.

Dithering Effect. Slight errors are expected into histograms caused by the chosen histogram resolution (number of bins). This is a similar case as with noise; the difference is less noise. A possible fix for this is to use a distance measure that is not very sensitive to subtle differences (like the Kolmogorov-Smirnov test). An other alternative may be to just smooth the histogram. The effect of smoothing is illustrated in Figure 5.10.

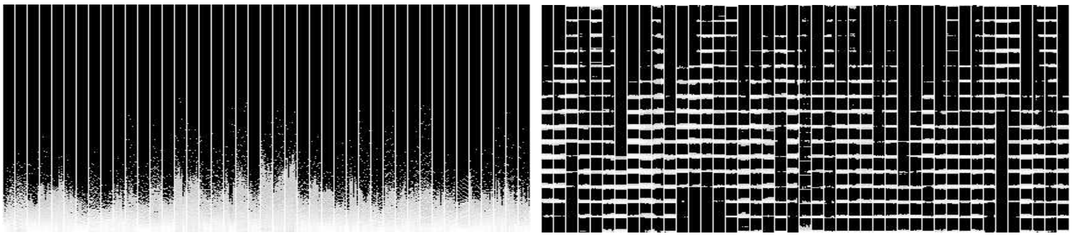


(a) Original histograms.

(b) Smoothed histograms.

Figure 5.10: Example of dithering effect in a histogram plot.

Global Level Variation. This is the kind of histogram plot expected from top performing features. Histogram differences within groups are very small while the differences between different groups are greater. Such an example is illustrated in Figure 5.11*b*. In the same figure, the histogram bins are also sparsely populated. A more complex bin membership function could be used to minimise this effect and essentially compress the histograms. Although, such a technique may be beneficial for CBIR, it is not investigated further.



(a) Narrow banding.

(b) Global level variation.

Figure 5.11: Histogram plots of two methods, using the video scene samples set.

Image sequences. In the case of the image sequences, histogram plots can be used to visualise how a feature histogram tolerates or gets affected by a transformation. We have identified three main behaviour cases:

Noisy. The histogram plots of this kind can be further classified into two general cases. In the first case a general noisy histogram plot is noticed, while the overall visual difference changes smoothly (Figure *5.12*a). This is similar to the noisy histogram plots of image sets. In the second case histogram plots have a large proportion of the plot very similar or even the same, while difference is noticed on a narrow band of the histogram (Figures *5.12*d, *5.11*a). Such behaviour suggests that the method fails to capture the image differences. The narrow banding also suggests a low discriminative power of the method.

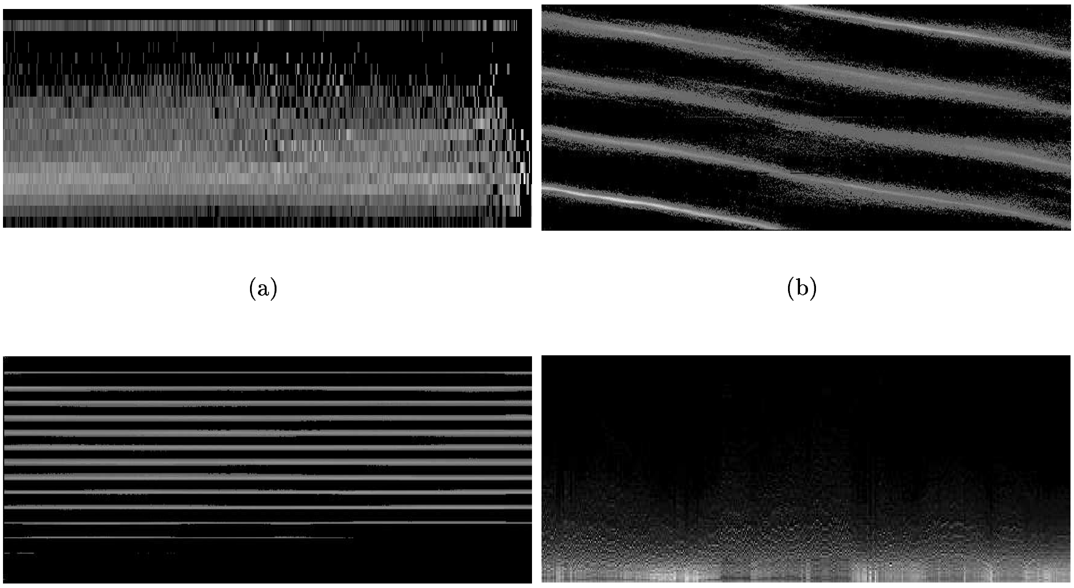
Shift. A shape of the histogram plot follows some trend (Figure *5.12*b), e.g. shifts on some direction. This suggests that the similarity of the images can be captured but extra effort is required. This is when the distance between images is calculated so to compensate for this trend. (e.g. the best match euclidian, where shifting the histograms to match is involved)

Smooth or Minimal Change. This is the ideal behaviour of a histogram plot. Depending on the transformation in question different conclusions can be drawn. For all transformation cases this would mean invariance to the transformation. Also, for the case of the zoom-in sequence, this would also mean high localisation potential. An example of a histogram plot using the rotation sequence is illustrated in Figure 5.12*c*.

5.2.2 Similarity at Distance D

The initiative on the use of the image sequence data sets is to control the amount of change when comparing the features of two images. The objective is to generate a graph where we can view the differences of features. These differences are calculated using a distance/similarity measure while we progress through the histograms of sequence frames. The variable D controls the amount of change between the compared a sequence of frames. So, if H_i is the first histogram of the comparison the H_{i+D} is the second. Histograms are compared in synch with the original sequence.

Using this graph we can also visualise the effectiveness of alternative distance/similarity measures when trying to compensate for sensitivity to a particular transformation or a negative behaviour of a method (e.g. dithering effect in histogram plots and the effect of

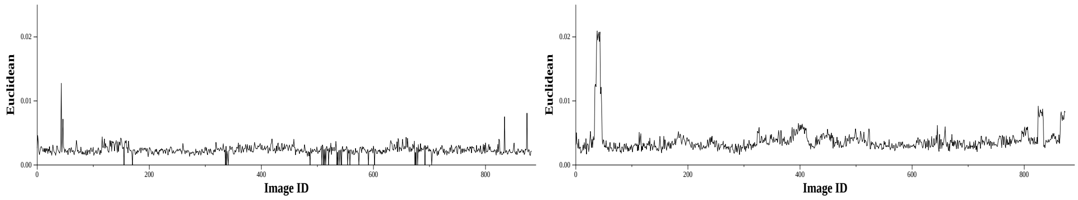


(a) (b)

(c) (d)

Figure 5.12: Examples of histogram plots using video sequences.

histogram smoothing). Examples of similarity plots are illustrated in Figure 5.13, using histograms of transformation sequences for two different D values.



(a) D=1 (b) D=10

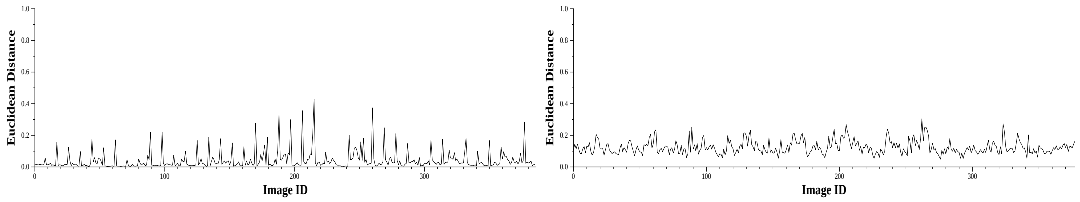
Figure 5.13: Similarity graphs of transformation sequences.

Although the use of the similarity plot makes more sense for video sequences we also experimented with the video scene samples. We used the lowest distance value $D = 1$ and as the images are grouped we compared the images in synch with the image groups and how the sequence they are stored in the groups. For histogram methods to be used in CBIR our expectations are:

- High distance values at the limits of the groups (when comparing the last image of a group with the first of the next group).
- Small distance values in between the limits of the groups (as the images are of similar

content).

Similarity plots fitting to the above description (Figure 5.14*a*) may be used as an indication of potential use of the histogram methods in CBIR. A case of a similarity plot not fitting to the above description is shown in Figure 5.14*b*. Note that similarity plots are used with grouped images only to get quick indication. As the method is based on a very small number of comparisons the result is not expected to be of a similar level of accuracy.



(a) D=1

(b) D=1

Figure 5.14: Examples of similarity graphs using the video samples.

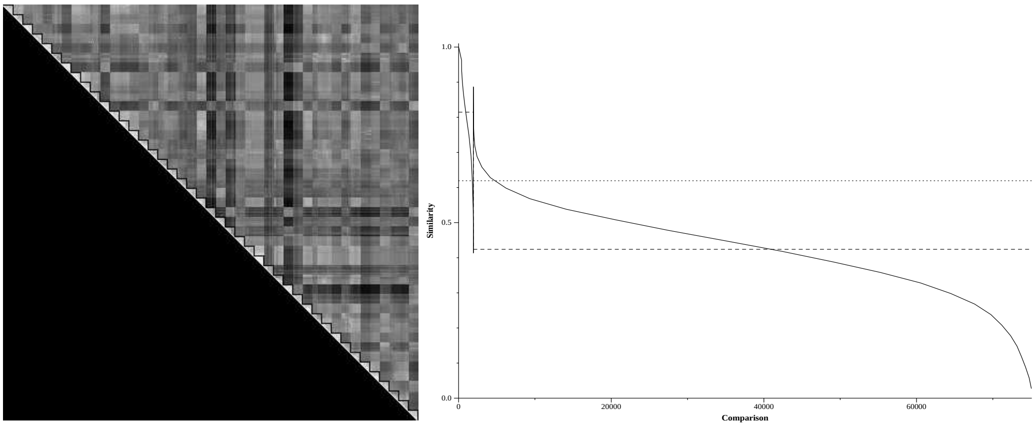
5.2.3 Distance Matrix

The distance matrix is used to generate a visual summary of distance values between images. The distance matrix is used along with grouped images sets. All the distances between image pairs are calculated and stored in a matrix, where the column and row indexes point to the compared images. The groundtruth is used to arrange the column and row indexes in such a way that images of the same group are adjacent. The values of the matrix are then translated to intensities to form an image. Higher intensities indicate greater similarity.

Assuming disjoint classes, this visualisation is capable of summarising the between classes (inter-class) and within classes (intra-class) similarities in one dataset. The intra-class similarities are arranged close to the diagonal while the inter-class similarities occupy the remaining area.

The expected behaviour of a good performing method is to produce high similarity values for the intra-class comparisons and lower values for the inter-class comparisons. This translates to a distance matrix with high intensities close to the diagonal and lower in the remaining area. An example is illustrated in Figure 5.15*a*.

Otherwise, high intensities in the inter-class area (and low intensities in the intra-class)



(a) (b)

Figure 5.15: Distance matrix of a good performing method.

indicate:

- The groundtruth was not properly defined which is not likely to happen (the prerformance is such cases can be compared to that obtained by experimenting with randomly classified images).
- The method fails to approach the ground-truth for the used dataset. This may be due to the high level of information used for the classification, which the feature(s) fail to pick up from the images, or
- The method fails to extract any useful information out of the image.

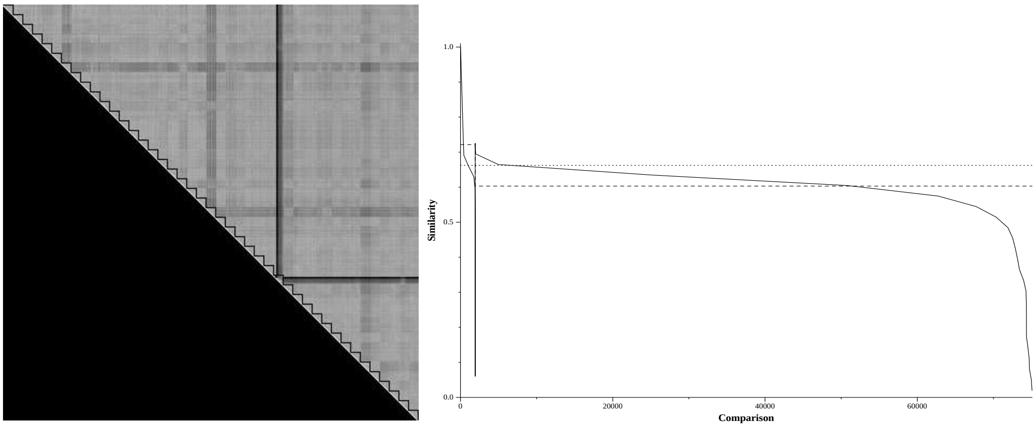
An example of such a method is illustrated in Figure 5.16a.

The above distance matrix can be summarised to a single value, indicating overall performance, as:

$$\begin{aligned} M &= \frac{\sum_{c=1}^{k}\sum_{i=1}^{O_c}\sum_{j=1}^{Q_c} f_{i,j}}{\sum_{c=1}^{k} O_c Q_c}, \; f_{ij} = \begin{cases} 1\,\text{if}\, i > j \\ 0\,\text{if}\, i \leq j \end{cases} \\ Q_c &= N - n_c, \; O_c = \frac{n_c(n_c-1)}{2} \text{ with } n_c \geq 2, \text{ and } n_c < N \end{aligned} \tag{5.1}$$

where, N the total number of images in the dataset, n_c the number of images in class c and k the number of classes.

As an additional indication we can plot the calculated similarities, in descending order, Figures 5.15b and 5.16b. The vertical line acts as a separator between the intra-class and



(a)

(b)

Figure 5.16: Distance matrix of a bad performing method.

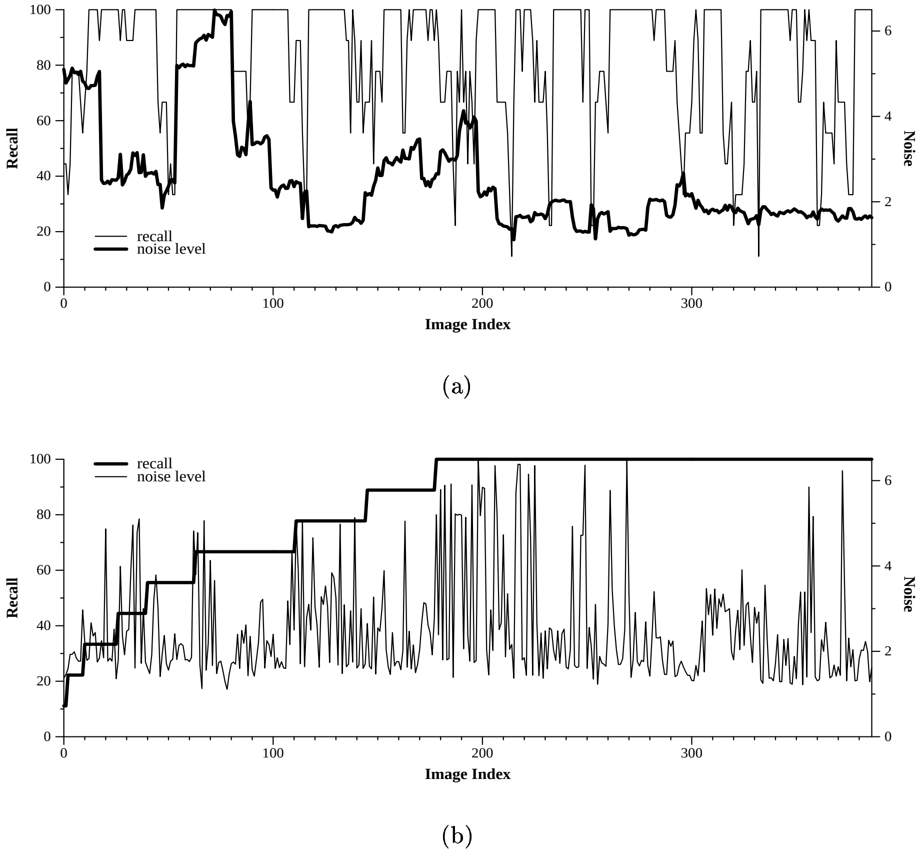
inter-class similarity values. The two dashed lines indicate the median similarity values of the two classes, while the dotted line indicates the mean value of the two medians. The closer the two medians the worse the performance. An ideal solution would result in a plot very close to a step curve.

As the distance matrix is composed of similarity values, it can be used on either single histogram methods or even combinations.

5.2.4 Noise Dependency

The fact that the images of the datasets were collected from various sources (possible variable quality) suggests that variable noise levels are to be expected in images. It is important to know if the existence of noise affects the behaviour of a method. In particular we want to see if there is any dependency between performance and noise (does a method capture the underlying feature, or the noise?).

In [51] the standard deviation of the grey-scale map is computed using the fast noise variance estimation. Testing for correlations between the observed noise and performance levels (using Pearson, Spearman or Kendal) can show potential dependencies to noise. Two instances of recall and noise observation curves are illustrated in Figure 5.17 (using the results from a single histogram method). In Figure 5.17a, the observations are sorted in the same order as the images are stored in the database. Similarity of the noise level within single image groups is apparent. In Figure 5.17b, the observations are sorted by performance (ascending order). With this graph a possible correlation between the two curves would be visible. We also used the *Pearson* product moment correlation coefficient to quantify a possible correlation between the two curves.In this particular example no obvious correlation between performance and noise level exist, suggesting that this particular method does not suffer from any dependency on noise.



(a)

(b)

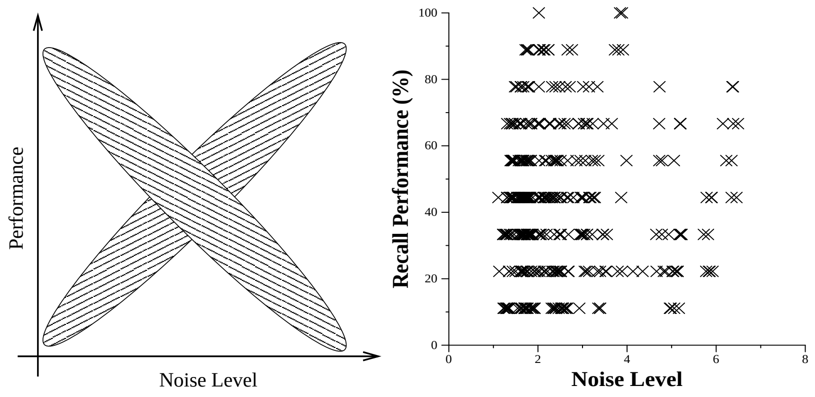
Figure 5.17: Noise dependency graphs Pearson product moment correlation coefficient $P = 0.1009$

As an alternative to Figure 5.17 we use a scatter plot of the observation, as shown in Figure 5.18. Clusters on the grayed area of $5.18a$ would signify a dependency. An example of the actual scatter plot of recall and noise observations is illustrated in $5.18b$.

We expect this graph to be more beneficial when used with the video scene samples image set. This is because the noise levels vary between groups, while remaining almost the same within groups.

5.3 Quantitative Measures

The evaluation methods presented in this section are intended to provide quantitative results of the evaluation:



(a) (b)

Figure 5.18: Performance vs Noise scatter plot.

- Comparing the results of various techniques (distance measures, single or combinations of histogram features) with a groundtruth (automatic or obtained by human subjects) to derive a quantitative measure of agreement.

- A statistical analysis is performed over a group of such measurements to find out if the detected differences are of any significance.

The abovementioned operations are performed using standard techniques.

5.3.1 Query Effectiveness

Having the query-by-example paradigm in mind, an example image is shown to the system. Low-level image processing techniques are used to extract several aspects from the image. These are summarised by histograms, which comprise the signature of the image. That signature is compared against the signatures previously extracted and stored in a database. The resulting distances are used to quantify the similarity between the query image and those stored in the database. Then the distances are sorted to associate a ranking to the images in the database. A rank of 1 is assigned to the most similar image while a rank of N is assigned to the least similar image. A performance measure can then be used to judge the performance of the query.

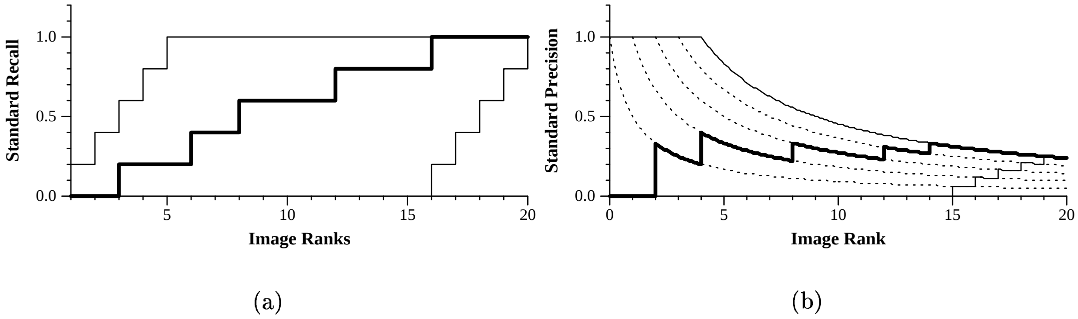
Standard methodology from the area of document retrieval have been used also in evaluating the performance of various image retrieval applications. The most common has been the pair of *recall* R and *precision* P. Applied to the CBIR context, suppose an image database containing N_c relevant images, B as the set of retrieved images at a time and A

as the set of relevant retrieved images at that time ($A \subset B$).

Although recall and precision are objective measures and easy to compute, they suffer from certain shortcomings when used in evaluation of a CBIR system:

- Image Relevance. The process of defining the relevance between images is ill defined.
- Sensitivity to a threshold. CBIR systems are generally based on ranked lists of retrieved images and the recall/precision values are calculated based on a threshold (cutoff) chosen by the operator of the system. Unfortunately, these measures are very sensitive to the choice of that threshold.

Salton, in [97], proposed normalised forms of recall and precision to eliminate the need of a threshold and any problems bound to that.



(a) (b)

Figure 5.19: Construction of normalised (a) Recall and (b) precision. The upper thin solid curves represent the ideal case. The thick solid curves represent a typical case while the lower thin curves represent the worst case.

Both measures represent the area between the ideal case and the observed case curves and they are calculated as follows:

$$R_{norm} = 1 - \frac{\sum_{i=1}^{n} r_i - \sum_{i=1}^{n} i}{n(N-n)} \tag{5.2}$$

$$P_{norm} = 1 - \frac{\sum_{i=1}^{n} \log r_i - \sum_{i=1}^{n} \log i}{\log\left[\frac{N!}{n!(N-n)!}\right]} \tag{5.3}$$

The normalised recall and precision can be interpreted as approximation to the average standard recall and precision measures over all the possible cutoff levels. A schematic example is shown in figure 5.19.

Salton also suggests the value ε to summarise the recall and precision measures into a single number,

$$\varepsilon = 1 - \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}, \text{ where } 0 \leq \alpha \leq 1 \tag{5.4}$$

the factor α is used as balancing weight between recall and precision. As α gets closer to 1 precision the contribution to the value of ε is greater than the contribution of the recall. The opposite holds when α gets close to 0.

Alternative methods based on statistical tests have been proposed (*e.g.* [25]) but improvements are due to independence of a threshold and while the problem of image relevance, remains.

Such performance measures work well in document retrieval mainly due to the large amount of textual information in the documents' content. When it comes to images such information is non-existent. Keywords are extracted based on subjective interpretation of the image content. That makes the whole process of groundtruthing ill-defined. Although, precision and recall can still be used as a reference and not as a panacea of testing.

This evaluation framework does not dependend on any particular performance measures. The abovementioned measures were arbitrarily chosen and alternative measures can be used without any fundamental changes.

For our initial experiments and published results we used a different, single value, performance measure. We called that method, the weighted class expansion, and it is calculated using the formula,

$$P_i = \left(1 - \frac{C_k}{N}\right)\frac{C_k}{L_k} \tag{5.5}$$

where, k is the class the query image belong to, and C_k the number of images in that class. L_k is the rank of the last retrieved image of class k, and N the total number of images in the dataset. The advantages of the weighted class expansion are i) a single value is generated, and ii) the resulting values are not biased by the size of the image classes. On the downside, we found that it may can be sensitive to outliers, (e.g. if the majority of images of class k are clustered and with a good ranking, a single, poorly ranked image will result in a low score, the opposite is also possible). For this reason and for consistency with the research community we switched to the normalised recall/precision, and the epsilon value as proposed by Salton [97].

5.3.2 Statistical Analysis (does the difference make any difference?)

Experiments, usually, consist of a number of trials and every trial, quite expectedly, will give different results. Assume two trials using two different histogram feature combinations F_1 and F_2. The resulting rankings of the two methods are compared with the given ground truth, resulting in a performance result P_1 and P_2 respectively. We cannot generalise and say with a certain confidence that method F_1 is better than F_2. A naive expansion of the above case would be a comparison of the average scores obtained by the the methods. Still that does not tell anything about the performance variance of a single method or between the two methods.

Performance values almost invariably differ somewhat between different histogram feature methods. The question is whether the difference among the samples signify genuine population differences or whether they represent merely random variations beyond the control of the experimenter. This type of question can be treated as a *significance test* (or hypothesis test) problem.

Statistical analysis provides a large pool of techniques for significance testing. These are generally grouped into two categories, the *parametric* and the *non-parametric* (or distribution free) tests. In parametric tests assumption is made about the population distribution having the characteristics of a normal distribution while in non-parametric tests such an assumption is not made.

Before going any further it is mandatory, for consistency, to put down the associations of the terms used in CBIR with the appropriate statistical terms. The result (performance value(s)) of a single query is the *observation* while the result of a set of queries comprises a *sample.*

Since there is not *a priori* knowledge about the shape of the distributions of the expected samples a non-parametric test is preferred. Additionally, since the samples under test are independent and contain ordinal observations three tests are suggested in the literature [104] the extension of the *median test*, the *Kruskal-Wallis* (KW) one-way analysis of variance by ranks and the *Jonckheere test* (J) for ordered alternatives.

The median test can be useful for testing but the Kruskal-Wallis test is preferred as more information from the samples are used. The Jonckheere test is not suitable for this test as the point of interest is if the difference of the samples makes any difference.

The Kruskal-Wallis one-way analysis of variance by ranks tests the hypothesis that k independent groups or samples are the same against the alternative hypothesis that one or more of the samples differ from the others. Following, the description of the KW test as derived from [104].

The null hypothesis and its alternative are specified as:

$H_0 : \vartheta_1 = \vartheta_2 = ... = \vartheta_k$ and

$H_1 : \vartheta_i \neq \vartheta_j$ for some groups/methods i and j

with ϑ_i the median for the j^{th} group. That is, if the alternative hypothesis is true, at least one pair of groups has different medians. Under the null hypothesis, the test assumes that the variables under study have the same underlying continuous distribution; thus it requires at least ordinal measurement of that variable.

In applying the Kruskal-Wallis one-way analysis of variance by ranks, the data are cast into a two-way table with each column representing each successive sample group. For simplicity, equally populated samples are assumed.

Group Value	1	2	$\cdots$	k
1	X_{11}	X_{12}	$\cdots$	X_{1k}
2	X_{21}	X_{22}	$\cdots$	X_{2k}
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
n	X_{n1}	X_{n2}	$\cdots$	X_{nk}

Table 5.1: Data arrangement in a two-way table, assuming $n = n1 = n2 = ... = nk$.

In the computation of the Kruskal-Wallis test, each of the N observations is replaced by ranks. That is, all of the scores from all the k samples are combined and ranked in a *single* series. The smallest score is replaced by rank 1, the next smallest by rank 2, and the largest score is replaced by rank N, where N is the total number of independent observations in the k samples.

When this is done, the sum of the ranks in each sample (column) is found. From these sums the average rank for each sample or group can be computed. Now, if the samples are from the same identical populations with different medians the average ranks should differ. The Kruskal-Wallis test assesses the differences among average ranks to determine whether they are so disparate that they are not likely to have come from samples which were drawn from the same population.

Two forms for the computation of the Kruskal-Wallis statistic are as follows:

$$
\begin{aligned}
KW &= \tfrac{12}{N(N+1)} \sum_{j=1}^{k} n_j \left(\overline{R_j} - \overline{R}\right)^2 \\
\text{or} \quad KW &= \left[\tfrac{12}{N(N+1)} \sum_{j=1}^{k} n_j \overline{R}_j^2\right] - 3(N+1) \\
\text{with} \quad N &= \sum_{j=1}^{k} n_j \\
\text{or} \quad N &= \sum_{1}^{k} n \text{ for equally sized samples} \\
\text{where} \quad k &= \text{number of samples or groups} \\
n_j &= \text{number of cases in the } i^{th} \text{ sample} \\
N &= \text{number of observations in the combined sample} \\
\overline{R_j} &= \text{average of the ranks in the } j^{th} \text{ sample or group} \\
\overline{R} &= \tfrac{(N+1)}{2} = \text{the average of the ranks in the combined sample}
\end{aligned}
\tag{5.6}
$$

If the k samples actually are drawn from the same population or from identical populations (H_0 is true) then the sampling distribution of the statistic KW can be calculated and the probability of observing different values of KW can be tabled. However, when there are more than $k = 3$ groups and when the number of observations in each group exceeds five, the sampling distribution of KW is well-approximated by the χ^2 distribution with $df = k - 1$ degrees of freedom. This approximation gets better as the number of groups, k, and the number of observations in each group, n_j, increase.

Tied observations are expected in the performance results. When ties occur between two or more scores (regardless of group), each score is given the mean of the ranks for which it is tied.

Since the variance of the sampling distribution of KW is influenced by ties, one may wish to correct for ties in the calculation of KW. This correction factor is computed as follows:

$$\begin{aligned} f &= 1-\frac{\sum\limits_{i=1}^{g}(t_i^3-t_i)}{N^3-N} \\ \text{where}\, g &= \text{number of groupings of different tied ranks} \\ t_i &= \text{number of tied ranks in the}\, i^{th}\, \text{groupping} \\ N &= \text{number of observations in the combined sample} \end{aligned} \tag{5.7}$$

The general expression of the KW statistic including the correction factor for ties is:

$$KW' = \frac{KW}{f} = \frac{\left[\frac{12}{N(N+1)}\sum\limits_{j=1}^{k} n_j\overline{R}_j^2\right] - 3(N+1)}{1-\frac{\sum\limits_{i=1}^{g}(t_i^3-t_i)}{N^3-N}} \tag{5.8}$$

When the result of the Kruskal-Wallis invalidates the null hypothesis, it indicates that at least one of the methods is different from at least one of the others. It is therefore more important to investigate further, to find which are the groups that have a significant difference. That is, we need to test the hypothesis

H_0 : $\vartheta_u = \vartheta_v$ and

H_1 : $\vartheta_u \neq \vartheta_v$ for some groups/methods u and v

In this case one could perform another non-parametric test like the Wilcoxon signed ranks test, but for computational conveniency a simple method due to Dunn is preferred. The rank differences $|\overline{R_u} - \overline{R_v}|$of all the pairs of groups are calculated. Then the significance of these pairs can be tested by the following inequality:

$$|\overline{R_u} - \overline{R_v}| \geq z_{a/k(k-1)}\sqrt{\frac{N(N+1)}{12}\left(\frac{1}{n_u}+\frac{1}{n_v}\right)} \tag{5.9}$$

The value of $z_{a/k(k-1)}$ is the abscissa value from the unit normal distribution above which lies $a/k(k-1)$ percent of the distribution.

The Kruskal-Wallis statistic along with Dunn's method are capable of indicating significant difference. Sometimes it is useful to investigate how distinct two methods (samples) are. *Spearman*'s r_s rank-order correlation coefficient provides a measure of association between two variables.

Suppose two samples $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$. The observations of the two samples are ranked independently. The disparity between the ranks of two observations is $d_i = x_i - y_i$. The Spearman rank-order correlation coefficient is then calculated by the formula:

$$r_s = \frac{\sum x^2 + \sum y^2 - \sum d^2}{2\sqrt{\sum x^2 \sum y^2}} \tag{5.10}$$

which does not require any correction for ties. Then the significance of an obtained r_s under the null hypothesis (for samples of $N > 50$) may be tested by the statistic

$$z = r_s\sqrt{N-1} \tag{5.11}$$

For large N, the value of z is approximately normally distributed with mean 0 and standard deviation 1.

5.4 Pathfinder Networks for CBIR Evaluation

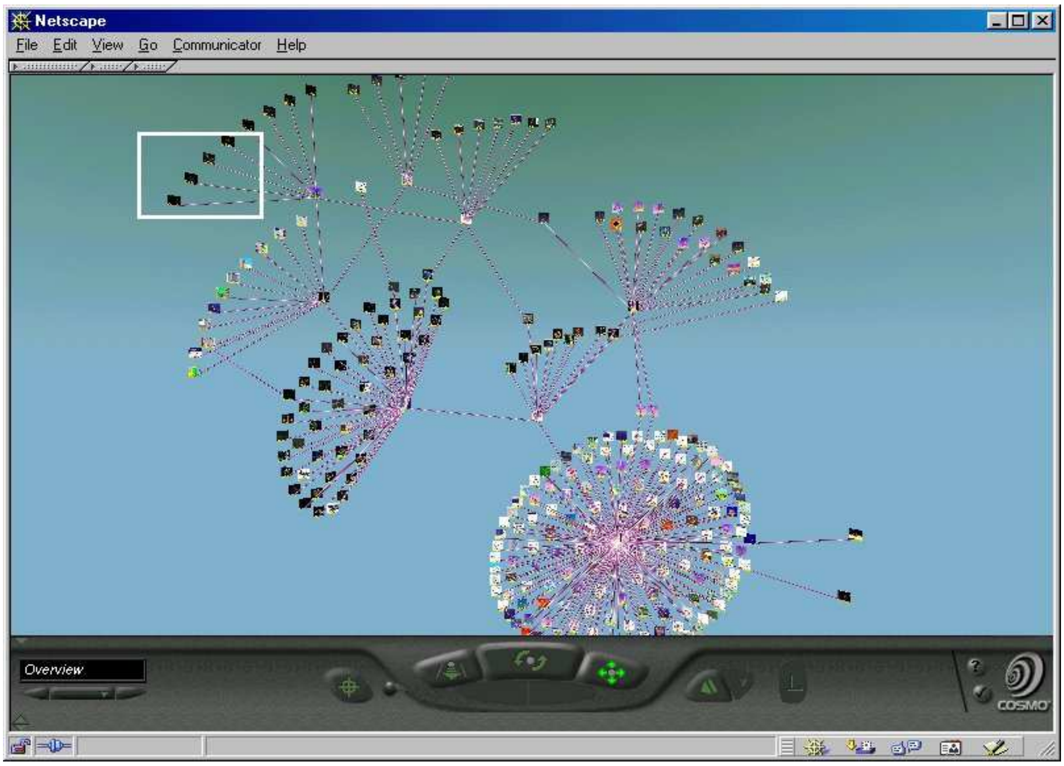
Pathfinder network scaling is a structural modelling technique originally developed for the analysis of proximity data in psychology [15]. This modelling technique is adapted to simplify and visualise the strongest interrelationships in proximity data. The triangular inequality condition is used to eliminate redundant or counter-intuitive links. The networks resulted from this process are called Pathfinder networks (PFNETs).

The topology of a PFNET is determined by two parameters r and q. The weight of the path is defined based on Minkowski metric with the r-parameter. The q-parameter specifies that the triangle inequality must be maintained against all the alternative paths with up to q links connecting nodes n_1 and n_k:

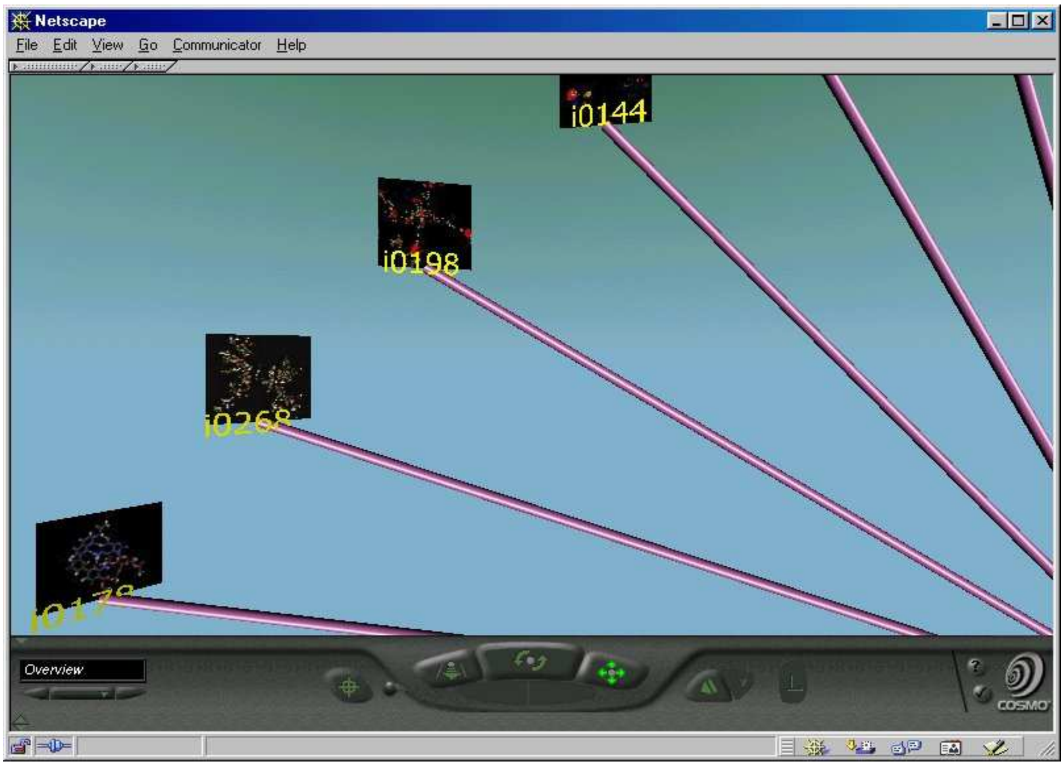
$$w_{n_1 n_k} \leq \left(\sum_{i=1}^{k-1} w^r_{n_i n_{i+1}} \right)^{\frac{1}{r}} \quad \forall k = 2, 3, \ldots\ q \tag{5.12}$$

Image similarity data, calculated by histogram methods (or combinations) described in Chapter 3, are passed to the Pathfinder network scaling technique. This process is similar to asking the CBIR system to group the images into classes. The resulting PFNETs are then rendered as virtual reality models in VRML. This allows navigation through the image classes created by the system. Furthermore it allows one to interrogate the image associations and verify their validity, Figure 5.20.

Further information about *generalised similarity analysis* can be found in [15], while more details on using PFNETs and CBIR methods can be found in [40][17][16][18].



(a) Overview



(b) Close-up to an image class

Figure 5.20: Example of a PFNET VRML rendering.

Chapter 6

Obtained Results

The work involved in this chapter is related to the objective that is discussed in Chapter 1.3.3, related to the performance evaluation framework. We present the results of experiments using the methodology discussed in Chapter 3 and the evaluation framework presented in Chapter 5. Discussion is focused on the obtained results, while a critique on the evaluation framework is included along the discussion.

The chapter starts with qualitative and quantitative analysis of individual histogram methods, Sections 6.1 and 6.2, respectively. The chapter concludes with the results of experiments with combination of histograms, in Section 6.3.

6.1 Qualitative Analysis of Individual Histograms

In this section we summarise our observations on the histogram plots and similarity graphs of the individual histogram methods.

Histogram plots generated from images of the grouped datasets are used to visualise the discriminatory potential. In particular we are looking at:

- *Narrow band bin population.* We want to see how the histogram bins are populated, through a range of grouped images. The greater the range of populated bins the better.

- *Variation or Uniformity.* We are interested to find methods with histogram shapes of high global variation (that is, between groups) and similarity within groups.

- *Dithering effect or Noise.* We are also interested to see if the histogram shapes

Label	Method	Label	Method
HUE	Hue	HUE-SP	Spatial Hue
CL	Colour Labels	CL-SP	Spatial Colour Labels
TU	Texture Unit	TU-SP	Spatial Texture Unit
COOC	Cooccurrence Matrix	HCC	Hue Cooccurrence Matrix
CC-SP	Spatial Cooccurence Matric	KCOS	K-Cos Curvature
CIRC	Circularity	ELIP	Elipticity
RECT	Rectancularity	CONV	Convexity
MSSDT	Edge Distance	CLSH	Colour Region Distance
BKDT	Colour Lavels *vs* Boundary Distance	BKDT2D	Colour Labels *vs* Edge Distance
ORNT	Orientation	TARE	Triangle Area
TRAT	Triangle Ratio	TLEN	Triangle Length
CARC	Circular Arcs	HENT	Hue Entropy
LDIFF	Local Difference	LBLK	Local Black

Table 6.1: Histogram methods.

through the range of grouped images are noisy. Also, we want to see is there are cases of methods with subtle differences between histograms (the histogram plots exhibit a dithering effect).

Furthermore, we generated histogram plots using three transformation video sequences (translation, rotation and scale). This way we can visualise the effects of the transformation to the shape of histograms, and the possible tolerance to the transformations.

In addition to the histogram plots we use the similarity graphs (Section 5.2.2) (generated from image sequences using the Euclidean distance) as another view of assessing the tolerance to the transformations.

In the Table 6.1, the methods along with their corresponding abbreviation are listed, for reference.

6.1.1 Hue Histogram (HUE)

The hue histogram plots (Figure B.1) show no particular problems with the bin population range for the Video samples and COIL-100 datasets. A limited range is noticed with the Artchive and even more with the Chen collection. This is due to the limited colour range and larger flat areas (similar if not the same colour) in the images of these two collections.

In terms of variation, the histogram plots from the Video samples and COIL-100 datasets show a healthier behaviour, with better group separation (between group difference), and within group similarities. For the case of the Artchive, we noticed more subtle differences between image groups. For the case of the Chen collection, we noticed sub-groups within the groups defined by the ground truth.

Overall, we did not notice noisy behaviour or dithering effects in the histogram plots.

The histogram plots from the transformation sequences showed smooth and subtle transitions. In the case of the zoom-in (scale) sequence, the transition is due to the changes in the range of hues. For the cases of the translation and rotation, it is due to changes in the actual content.

The similarity graphs (Figure C.1) confirmed this tolerance for the translation and rotation sequences, with the majority of distance values to be less than 0.01. Increasing the d parameter had little effect for the two transformations. The similarity graph of the Zoom-in sequence showed greater sensitivity.

The distance matrix using the video sequence (Figure D.16a) shows a highlighted diagonal (area comparing images within the same group), which indicates good recall potential, while the intensity of the remaining pixels (area comparing images between different groups) is lower than the diagonal, indicating good precision performance. The distance matrix using the Chen collection (Figure D.16b) showed a difficulty of the system to highlight the diagonal area, and therefore, low performance is expected.

6.1.2 Spatial Hue (HUE-SP)

The spatial hue histogram plots share the same characteristics as the hue histogram plots. A similar trend is also observed in the similarity graphs, with an increase of the distance values, closing the $0.02 - 0.03$ range, for the translation and rotation sequences. A smaller distance increase is noticed for the zoom-in sequence (close to the value of 0.1 for $d = 50$).

The distance matrices (Figure D.18) of the HUE-SP method we similar to HUE. For the case of the non diagonal area on the matrix of the video sequence, we noticed lower variation, indicating possible performance increase from HUE.

6.1.3 Colour Labels (CL)

The bin population of the colour label histogram shows no narrow banding, for all the grouped datasets (Figure B.2). This is mainly due to the categorisation of the colour space. Variation between the image groups is evident for the Video samples and the COIL-100 datasets, while similarities within the groups seem greater. For the case of the Artchive and the Chen collection, we noticed a noisy behaviour, with sub-groups for the case of the Chen collection.

Tolerance to transformation is expected to be high (higher than the Hue histogram), as we noticed more stable histogram plots for the transformation sequences.

The similarity graphs (Figure C.3) showed similar behaviour as in the case of the hue histogram. The majority of distances are lower than 0.04 for the translation and rotation sequences. Again, the distances obtained from the zoom-in sequence are higher.

The distance matrix using the video sequence (Figure D.1a) shows a highlighted diagonal, which indicates good recall potential, while the intensity of the remaining pixels seem to be high, indicating possible low precision performance. The distance matrix using the Chen collection (Figure D.1b) showed a difficulty of the system to highlight the diagonal area, and therefore, low performance is expected.

6.1.4 Spatial Colour Labels (CL-SP)

The spatial version (Figure B.18) of the colour labels histogram showed a very similar behaviour as in the case of the colour labels histogram. An increase of the distance values is noticed in the similarity graphs C.4, while the majority is well below 0.1.

The distance matrices of CL-SP (Figure D.2) were similar to CL, with a slight increase to the intensities of the non-diagonal area, indicating lower performance than CL.

6.1.5 Texture Unit (TU)

We did not notice any problems related to the bin population for all the histogram plots (Figure B.14) of the grouped image datasets. Evident of variation between the histograms of different image groups was noticed. Although, that was little compared to the overall uniformity over the range of the histogram plots. We also noticed noisy behaviour in the case of the Chen collection.

The histogram plots of the transformation sequences, showed stable behaviour, which is further confirmed by the range of distance values in the similarity graphs (Figure C.5) which was less than 0.02 for the majority of the comparisons for $d = 50$. We noticed a fluctuation at the similarity graph of the rotation sequence, which requires further investigation, to conclude on the tolerance of the T-Unit histogram under rotation.

The distance matrix using the video sequence (Figure D.21a) shows a highlighted diagonal, which indicates good recall potential, while the intensity of the remaining pixels seem to be high (indicating possible low precision) but with low variability (which indicates otherwise). The distance matrix using the Chen collection (Figure D.21b) showed (again) characteristics of possible low performance.

6.1.6 Spatial Texture Unit (TU-SP)

The spatial version of the T-Unit histogram showed very similar behaviour across all the histogram plots (Figure B.19) and similarity graphs (Figure C.6), with a slightly elevated distance value range. The fluctuation was again evidence for the rotation sequence.

The distance matrices of TU-SP (Figure D.22) were similar to TUNIT.

6.1.7 Cooccurrence Matrix (COOC)

The bin population of the histogram plots (Figure B.15) showed no problems throughout the range of grouped image datasets. We noticed a trend of peaks appearing on the diagonal of the 2D matrix, which was also evident as peaks in the 1D representation. It may be possible to use this information to derive a more compact version of the method.

Closer examination of the histogram plots showed strong similarities of the histograms within groups, while there was great variation between the image groups. The same cannot be said for the case of the Chen collection where we noticed sub-groups, less within group similarities. Overall we did not notice dithering of noisy behaviour.

The overall picture of the histogram plots from the video sequences was that the method shows tolerance to translation and rotation transformation. A gradual change of the histogram bin population is noticed for the zoom-in sequence, which we find to be normal as the amount of texture reduces as the sequence progresses. The similarity graphs (Figure C.7) confirmed this finding with distance values raising to over 0.3 for $d = 50$ towards the end of the zoom-in sequence. This sensitivity to scale may be reduced by considering multiple scales.

We also noticed some fluctuation in the similarity graph of the rotation sequence. We do not think that it is very important as the curve is close to a noisy behaviour, rather than a trend (as in the case of the T-Unit histogram, Figure C.5).

The distance matrix using the video sequence (Figure D.3a) shows a highlighted diagonal indicating good recall potential, while the intensity of the remaining pixels is lower than the diagonal, indicating good precision performance. The distance matrix using the Chen collection (Figure D.3b) showed a difficulty of the system to highlight the diagonal area, and therefore, low performance is expected.

6.1.8 Spatial Cooccurrence Matrix (CC-SP)

The histogram plots of the spatial version of the cooccurrence matrix (Figure B.20) vary very little. At least, that is the case for the grouped images sets. The picture is different for the sequences histogram plots, where more distortion is evident. We address this to the thresholding process, which may not be as reliable as we expected. This lower tolerance to transformations, compared to the non-spatial version, is confirmed by the similarity graphs (Figure C.9), where higher distance values are noticed.

The distance matrices of CC-SP (Figure D.20) were similar to CC.

6.1.9 Hue Cooccurrence Matrix (HCC)

When we applied the cooccurrence matrix to hue, we noticed slightly greater variation in the shape of the histogram plots (Figure B.16). Variations between image groups seem to be greater for the Video samples and the COIL-100 datasets. The picture remained almost the same (as in the case of intensity cooccurrence matrix) for the Chen and Artchive datasets. Histogram plots using the video sequences showed a similar level of tolerance to images transformations, as in the case of the intensity version.

The similarity graphs showed showed, again, similar distance values. The main difference was a greater fluctuation of distance values for the zoom-in and translation sequences (Figures C.8e and f). This suggest a possible greater sensitivity to these transformation in comparison to the intensity version.

Again, the distance matrices of HCC (Figure D.17) were similar to CC.

6.1.10 K-Cos Edge Curvature (KCOS)

The range of histogram bin population is noticed to be low for the K-Cos histogram plots (Figure B.3). Between group differences are not as distinct, as in previous methods, for most of the grouped datasets, with the histogram plots of the Artchive (Figure B.3d) being the worst case. The picture is better for the COIL-100 dataset.

The histogram plots of the video sequences were not particularly helpful. We noticed a noisy behaviour, with occasional peaks, possibly caused by the quantisation of the curvature values.

The similarity graphs (Figure C.10) showed the majority of the distance values to be at low levels, below 0.05, while fluctuations of the distance values are noticed. In all cases, these fluctuations do not follow a particular trend, and considering the behaviour of the method on the grouped image sets, we can not really conclude about the tolerance on image transformations.

The distance matrices of KCOS (Figure D.19) showed very little difference between the intensities of the two areas, indicating low performance.

6.1.11 Boundary Shape Histograms

This section includes the histogram plots of region boundary, Circularity (CIRC) (Figure B.6), Ellipticity (ELIP) (Figure B.7), Convexity (CONV) (Figure B.5) and Rectangularity (RECT) (Figure B.8). In all cases, the histogram plots showed sparse bin population and they were very noisy, suggesting poor performance. With a closer inspection of the histogram plots we noticed subtle variation between image groups and subtle similarities within image groups. This suggests that a careful selection of distance measure may improve the methods potential. The sparse population of the histograms was more evident for the COIL-100 (and less for the Video samples) dataset where images are simple with potentially small number of regions. More complex images from the artchive and Chen collection resulted to more densely populated histograms.

The histogram plots of the sequences were relatively stable but with noisy behaviour, not enabling us to comment any further before further examination. . The similarity graphs (Figures C.11, C.12, C.14 and C.13) confirmed the noisy behaviour of the histogram plots. The recorded distances for the same sequences were in a higher range $0.1 - 0.2$, while very high distances were noticed towards the end of the zoom-in sequence (over 0.4).

In all cases of boundary shape histograms (Figures D.26, D.24, D.25 and D.23), the distance matrices showed similar characteristics. Overall the intensities of the two areas were low, with very little difference.

6.1.12 Edge Distance (MSSDT)

While the range of bin population of the edge distance histogram plots (Figure B.10) is covering the whole range of the histogram bins, a domination of the lower numbered bins is evident. Although, we noticed between group variations (mainly due to differences in the bin population range) for the Video samples and Coil datasets. Less variation is noticed for the Artchive dataset, while for the Chen collection poor within group similarity is noticed. The histogram plots using the video sequences showed only subtle changes for the translation and rotation sequences. A greater change is noticed in the zoom-in sequence, which is attributed to distortions caused by the digital zoom of the camera.

Low distance values were recorded in the similarity graphs (Figure C.15), ranging between 0.01 and 0.02, suggesting possible tolerance to the applied transformations.

The diagonal area on the distance matrix of the Video samples (Figure D.4a) showed higher intensities, but little difference from the remaining area. For the case of the Chen collection (Figure D.4b), low differentiation between the two areas is more apparent.

6.1.13 Colour Region Distance (CLSH)

Limited bin population is noticed in the colour region distance histogram plots (Figure B.21), that is due to the use of the colour label information during the histogramming process. If the histogram is split into sections (each corresponding to a colour label) a behaviour similar to the MSSDT histogram plots is noticed for each histogram section.

Variation between image groups is mainly due to the use of the colour information. Similarities within groups are strong (for the Video samples and COIL-100 dataset), again due to the colour information, while slight variations are evident due to the shape information. Within group variations are not as subtle for the Artchive histogram plots, while they get even greater for the Chen collection. We attribute this to the high level information used for to group the images.

Histogram plots of the video sequences showed smooth transition for the cases the translation and rotation transformations. The behaviour of the zoom-in sequence is not as smooth, indicating possible sensitivity to scale.

The range of distances in the similarity graphs (Figure C.16) is low, within the range of 0.01 and 0.02 for the translation and rotations sequences, confirming the tolerance to the transformations. The range is greater for the zoom-in sequence (0.01 to 0.04).

The distance matrix of CLSH using the Video samples (Figure D.7a) showed a better differentiation than in MSSDT, indicating better performance potential. The picture remained the same for the case of the Chen collection (Figure D.7b).

6.1.14 Colour Labels and Boundary Distance (BKDT)

The colour labels and boundary distance histogram plots (Figure B.22), show a similar behaviour as in the case of the MSSDT histogram plot for all the histogram plots.

Furthermore a sparser population of the histograms is noticed, due to the different quantisation of the distance map (from the case of the MSSDT). Since the colour label boundaries are not weighted, more strong edges are expected in the edge map. This results in a shorter range of distances in the distance map.

A greater range of fluctuating distances is noticed in the similarity graphs (Figure C.17) suggesting poor tolerance to transformations.

The distance matrix of BKDT using Video samples(Figure D.10b), confirmed the possibility of poor performance. Indications of a possible fix are evident, as in some areas on the diagonal area are of higher intensity.

We think that this method may benefit from a lower histogram resolution.

6.1.15 Colour Labels and Edge Distance (BKDT2D)

The colour labels and edge distance histogram plots (FigureB.23) share (again) similar behaviour to the MSSDT histogram plots. Initial examination showed the histogram plots of the transformation video sequences to be sparsely populated as in the case of BKDT. Closer examination showed that it is not the case. That was further supported by the significantly lower distances in the similarity graphs (Figure C.18) that are about ten times lower than in the case of BKDT. This supports the view that the BKDT2D histograms are potentially tolerant to transformations.

The distance matrices of BKDT2D using Video samples(Figure D.11), show similar symptoms as in the case of BKDT.

6.1.16 Orientation (ORNT)

In the orientation histogram plots (Figure B.9) we did not notice any problems regarding the range of histogram bin population. We noticed three dominating peaks throughout all the histogram plots. Differences between histograms were reflected by the change on the peak (intensity and spread). For the Video samples and the COIL-100 datasets we see that great variation between the image groups is evident, while similarities within image groups are noticed. Narrower peaks are noticed in the histogram plots of the Archive dataset.

For the case of the Chen collection, variation is great throughout the range of the images, while we noticed very little similarity within groups.

The histogram plots of the translation and zoom-in video sequences suggest tolerance to the corresponding translations. That is confirmed by the corresponding similarity graphs (Figure C.19), showing low distance values and fluctuations. For the case of the rotation sequence, we noticed a change that followed a trend as the sequence progresses. This confirms the sensitivity of the method to rotation.

The distance matrix of ORNT using the Video samples (Figure D.5a) shows a highlighted diagonal area. While there is a difference between the intensities of the two areas, it is not great, indicating possible low precision performance.

6.1.17 Triangle Properties

The histogram plots of the triangle properties histograms (Figures B.11, B.12 and B.13) exhibited almost identical behaviour. Overall, noisy behaviour is noticed, with subtle similarities between the histograms within image groups. Variation between image groups is mainly due to difference in the bin range population.

In the histogram plots of the transformation video sequences we observed a number of peaks that remained stable throughout the sequence, while other peaks appeared to change following some trend, which may have a negative effect to the tolerance to image transformations.

In the similarity graphs (Figures C.20, C.22 and C.21) we observed low distance values (with the triangle area, the lowest), with low fluctuations.

The distance matrices of triangle property histograms (Figures D.12, D.14 and D.13) indicated poor performance as the intensity difference of the diagonal area was very small.

6.1.18 Circular Arcs (CARC)

The circular arc histogram plots (Figure B.4) showed a noisy behaviour, indicating low performance potential. This noisy behaviour was also noticed in the histogram plots of the transformation video sequences. We were not able to find any variations, similarities or transformation tolerance, by visual inspection.

Furthermore, the similarity graphs (Figure C.23) showed large fluctuations of distance values, close to the range between 0.1 to 0.4. This further supported the possibility for poor performance of the method.

The distance matrices of CARC (Figure D.15) also supported the possibility for poor performance, as there was no clear differentiation between the diagonal and the remaining region.

6.1.19 Hue Entropy (HENT)

A limited range of the histogram bins was populated in the case of the hue entropy histogram plots, mainly the lower half of the histogram range (Figure B.24). While some noise is noticed, similarities within the image groups and variation between image groups are also noticed. That is stronger in the cases of the Video samples and COIL-100 collections.

The histogram plots of the transformation sequences are also noisy, but a general stability is noticed, at least for the cases of rotation and translation. For the case of the zoom-in sequence we noticed a change to the histogram shape. The hue entropy histogram seems to have some tolerance to rotation and translation, while a sensitivity to scale is evident.

Low distance values were recorder in the similarity graphs (Figure C.24), with small fluctuations. This may suggest tolerance to the image transformations, but considering the noisy histogram plots, we need to investigate further.

The distance matrix of HENT using the Video samples (Figure D.6a), shows the diagonal area to be of slightly higher intensity than the remaining area.

6.1.20 Local *vs* Global Difference (LBLK & LDIFF)

The histogram plots of the local difference (Figure B.25) and local black histograms (Figure B.26) exhibit very similar characteristics to the hue entropy histogram. The difference is on the range of the histograms, with the local black histogram showing the greater range of the three.

Regardless to the noisy behaviour, low distance values were recorder in the similarity graphs (Figures C.25 and C.26), with small fluctuations.

The distance matrices in both cases of LBLK and LDIFF, using the Video samples (Figures D.8a and D.9a), show very little difference between the two areas. In the plot of the distance though, we noticed that the distances in the between groups region, are very similar and slightly lower than those in the within group distances. The case of LDIFF was slightly better than LBLK.

6.1.21 Summary

We presented the results obtained from histogram methods presented in Chapter 3, using qualitative methods presented in Chapter 5. In this section we summarise our observations as follows:

Features. Shape features are the most difficult to extract. This is supported by the poor histogram characteristics exhibited by most of the shape features (compared to the rest of the presented features). Colour remains the best performing feature, while methods combining colour and texture showed similarly good characteristics.

Transformations. Most of the methods that showed good histogram characteristics, also showed some tolerance to image transformations. Out of the three transformations, scale seem to be the most difficult to cope with. Most of the methods showed increased distance values under scaling. A possible solution to this may be to consider multiple scales during the histogramming process.

Spatial Information. With the introduction of spatial information we noticed and increase to the distance values of the similarity graphs. This extra information resulted to an expansion of the histogram space, therefore it is expected to see greater distance values. This expansion of the histogram space may be of interest when combining multiple histogram methods.

Dependency to Noise. In Appendix E, the scatter plots of noise level *vs* performance are illustrated. Over all the tested methods we did not find any strong correlation to noise. Therefore it is safe to say that the methods do not capture the noise level of images, but the features they are designed to capture.

Histogram Resolution. The number of bins in a histogram define its resolution. In the case of the colour and shape histogram 6.1.13 we noticed a sparse population of the histogram, which can potentially reduce the discriminatory power of the histogram. Subtle differences can be reduced by applying a different resolution, essentially filling the gaps that cause the problem. In Section 6.1.7, a number of peaks are noticed to be located at almost fixed areas in the histogram. A similar strategy to reducing the histogram resolution may be possible to be applied.

Datasets and Expectations. The initiative of using multiple datasets of grouped images is to test the histogram methods under a variable level of difficulty. As the level of information used to group the images gets higher, we have lower expectations. This was noticed for the Chen collection and the Artchive datasets, where most of the methods showed poor characteristics. This was further supported by the distance matrices, where the overlap between the two regions of the matrix are evident in the majority of the histogram methods using the Chen collection and Artchive datasets.

Qualitative Methods. We used the histogram plots, successfully, to get a feel about the shape of histograms and identify noisy or sparse behaviour. Also, we managed to visualise similarities and differences between images (within and between groups). At last, we used the histogram plots for assessment of histogram methods regarding their tolerance to various transformations. On the negative side, we often had to manipulate the histogram plot images, in order to pick certain features.

We used the similarity graphs, successfully, to support observations made on the histogram plots. We were also able to visualise potential trends, with respect to transformation tolerance.

Distance matrices were also useful, in particular the ones generated from Video samples and COIL-100 datasets. That is because the group size was the same for all groups making the limits of the diagonal area better defined. The only problem we encountered while using distance matrices was handling large datasets. This is because the dimensions of the distance matrix image are such that we had to scroll though the matrix. This also posed a typesetting problem, which is the reason we only included distance matrices of the Video samples and Chen collection datasets.

6.2 Quantitative Analysis of Individual Histograms

In this section we use the individual methods to perform image retrieval queries and extract performance values. We tested the individual methods using the four grouped datasets,

ID	Method
d0	Euclidean
d1	Cumulative Euclidean
d2	χ^2
d3	Kolmogorov-Smirnov
d4	Kuiper
d5	Pearson Product Moment Correlation
d6	Edit Distance

Table 6.2: Distance measures.

and 7 different distance measures (listed in Table 6.2). In particular, we wish to answer the following two questions:

- Which features perform best for each dataset?
- Is there any performance benefit by using different distance measures?

Furthermore, we applied statistical analysis tests on the obtained results, to verify that any detected performance differences are significant.

6.2.1 Single method performance

Video Samples. The colour feature histograms are the top performing methods. Following, the T-Unit histogram competes with the Colour Labels. The texture features dominate the low end of the top 10 performing features, while the hybrid method CLSH manages to reach the top 10. The spatial variants (HUE-SP, CL-SP and COOC-SP) outperform their non-spatial counterparts in the majority of cases. The performance values of the methods, using the 7 distance measures, are listed in Table 6.3, their ranking is listed in Table 6.12.

COIL-100. Again the colour features dominated the top ranks, CL and HUE variants in particular. The CLSH method got a higher rank this time, followed by the texture feature methods, with COOC outperforming the TU variants. We also noticed the HENT and BKDT2D methods at the low end of the top 10. The raw performance values are listed in Table 6.4, and the method ranking in Table 6.13.

Chen Collection. Colour is again the top performing feature (HUE variants). The colour label methods seem to compete with texture for four positions. The remainder of

Feature	**d0**	**d1**	**d2**	**d3**	**d4**	**d5**	**d6**
CL	0.959	0.9562	0.9729	0.9541	0.9612	0.9518	0.9554
CL-SP	0.9481	0.951	0.9656	0.9418	0.9409	0.944	0.9572
COOC	0.9579	0.9308	0.9744	0.9381	0.9532	0.9607	0.9277
MSSDT	0.6563	0.5858	0.6355	0.5922	0.6181	0.6711	0.5817
ORNT	0.8704	0.8588	0.9143	0.8533	0.8888	0.8567	0.8478
HENT	0.8912	0.9097	0.9049	0.9065	0.9253	0.8799	0.909
CLSH	0.92	0.9333	0.9481	0.9223	0.9376	0.9284	0.9307
LBLK	0.7592	0.8574	0.7748	0.8384	0.8472	0.7191	0.8644
LDIFF	0.722	0.8303	0.7327	0.8127	0.8278	0.6732	0.8361
BKDT	0.577	0.6906	0.5561	0.6691	0.7379	0.5893	0.6843
BKDT2D	0.6812	0.8218	0.7046	0.8332	0.8471	0.6573	0.818
TARE	0.6178	0.6521	0.5773	0.6419	0.6663	0.5972	0.6552
TRAT	0.5397	0.6724	0.5609	0.6668	0.6767	0.5416	0.6739
TLEN	0.607	0.6475	0.6203	0.6516	0.675	0.5958	0.6382
CARC	0.5344	0.6245	0.5426	0.5855	0.5697	0.5412	0.6387
HUE	0.9815	0.9614	0.9887	**0.9692**	0.9761	0.9838	0.9556
HCC	0.9701	0.9593	0.9865	0.9674	0.9743	0.9671	0.9532
HUE-SP	**0.9834**	**0.967**	**0.9899**	**0.9692**	**0.9775**	**0.9853**	**0.9636**
KCOS	0.6551	0.6442	0.6924	0.6511	0.6621	0.6536	0.6296
COOC-SP	0.9591	0.9385	0.9718	0.9425	0.9522	0.9627	0.9368
TU	0.9601	0.9253	0.9786	0.9467	0.9594	0.9658	0.9179
TU-SP	0.9611	0.9309	0.9775	0.9373	0.9481	0.9652	0.9301
CIRC	0.5101	0.5888	0.5142	0.54	0.539	0.5025	0.5954
CONV	0.4571	0.5579	0.4956	0.5309	0.5224	0.4817	0.5784
ELIP	0.4937	0.6341	0.5798	0.6198	0.6345	0.5434	0.6297
RECT	0.4435	0.5726	0.4933	0.5326	0.5234	0.4734	0.5987

Table 6.3: Single histogram performance VIDEO.

the top 10 is occupied by more colour and texture methods. The raw performance values are listed in Table 6.5, and the method ranking in Table 6.14.

Artchive. While colour again dominates the high end of the top ten, CLSH manages to reach the fourth position. Overall the top ten is dominated by colour and texture features. Raw performance values are listed in Table 6.6, and the method ranking in Table 6.15.

Distance Measure. Throughout the range of datasets and we observed differences in performance when using different distance measures. Looking at the raw scores, (Tables 6.3 - 6.6), we counted how many times each distance appeared to perform best. The top three distance measures that appeared consistently throughout the testing were, *i*) the χ^2 test, *ii*) the *Kuiper* test, and *iii*) the *edit* distance. We did not notice any correlation between types of features and distance measures.

6.2.2 Statistical Test of Significance

The obtained results showed that performance differences exist *i*) when using different histogram methods, and *ii*) when using different distance measures with the same histogram feature. In this section we want to find out if these differences are significant. For this

Feature	**d0**	**d1**	**d2**	**d3**	**d4**	**d5**	**d6**
CL	0.874	0.8403	0.9324	0.8541	0.8826	0.8744	0.8369
CL-SP	**0.9175**	**0.9095**	**0.9475**	**0.9042**	0.914	**0.9141**	**0.9121**
COOC	0.7475	0.7675	0.8661	0.7663	0.7965	0.7653	0.7761
MSSDT	0.5487	0.4854	0.5372	0.495	0.5275	0.5403	0.483
ORNT	0.5646	0.6287	0.6166	0.5887	0.6351	0.6231	0.6223
HENT	0.6354	0.6724	0.652	0.6671	0.6807	0.5657	0.6734
CLSH	0.8431	0.797	0.8686	0.7975	0.8298	0.842	0.7948
LBLK	0.5874	0.6329	0.5862	0.6317	0.6485	0.5269	0.6301
LDIFF	0.5971	0.6355	0.5869	0.6323	0.643	0.5288	0.6329
BKDT	0.4938	0.5833	0.4915	0.5605	0.611	0.5124	0.5815
BKDT2D	0.4883	0.6683	0.4793	0.6814	0.6959	0.5014	0.6641
TARE	0.4363	0.4888	0.4583	0.465	0.488	0.4417	0.4934
TRAT	0.4168	0.501	0.4554	0.4746	0.4839	0.4426	0.503
TLEN	0.445	0.496	0.4791	0.4794	0.4935	0.4618	0.4913
CARC	0.4412	0.462	0.4446	0.4451	0.4532	0.4379	0.4647
HUE	0.8255	0.8637	0.9584	0.8741	**0.9148**	0.8856	0.8497
HCC	0.8547	0.8482	0.9336	0.8494	0.8834	0.8569	0.8403
HUE-SP	0.7886	0.8129	0.8975	0.7088	0.742	0.8336	0.8118
KCOS	0.5324	0.5578	0.5519	0.547	0.5539	0.5183	0.5477
COOC-SP	0.7492	0.7623	0.8165	0.744	0.7702	0.7647	0.7717
TU	0.6995	0.6642	0.8032	0.6835	0.7114	0.7472	0.6573
TU-SP	0.6761	0.6502	0.7525	0.5866	0.6055	0.6945	0.6479
CIRC	0.3949	0.4903	0.4104	0.4098	0.4144	0.3875	0.4922
CONV	0.3874	0.5247	0.4072	0.4255	0.4153	0.3834	0.5333
ELIP	0.4045	0.542	0.4274	0.4064	0.4424	0.404	0.5387
RECT	0.3847	0.5236	0.4029	0.4029	0.4022	0.3799	0.5416

Table 6.4: Single histogram performance COIL.

Feature	**d0**	**d1**	**d2**	**d3**	**d4**	**d5**	**d6**
CL	**0.6581**	**0.6526**	**0.6693**	**0.6658**	**0.6752**	0.6242	**0.6412**
CL-SP	0.6353	0.6326	0.6388	0.6372	0.6218	0.6116	0.6293
COOC	0.6067	0.6449	0.6313	0.6282	0.6311	0.6067	0.6485
MSSDT	0.4751	0.4656	0.4749	0.468	0.4727	0.4752	0.4643
ORNT	0.5324	0.543	0.5569	0.536	0.5382	0.5358	0.5435
HENT	0.5509	0.5476	0.5553	0.548	0.5539	0.532	0.5433
CLSH	0.632	0.6415	0.6541	0.6491	0.6562	**0.6345**	0.6323
LBLK	0.5599	0.5937	0.5877	0.5873	0.5845	0.5537	0.5966
LDIFF	0.5285	0.5408	0.535	0.5357	0.5305	0.5198	0.543
BKDT	0.5079	0.5069	0.5026	0.5222	0.5409	0.502	0.501
BKDT2D	0.5606	0.5602	0.5618	0.576	0.579	0.5753	0.5526
TARE	0.5286	0.5078	0.5157	0.5084	0.5099	0.5293	0.5082
TRAT	0.4984	0.4964	0.5021	0.4971	0.5078	0.513	0.4968
TLEN	0.511	0.5062	0.5072	0.5058	0.5138	0.5008	0.5058
CARC	0.5072	0.5129	0.5208	0.5167	0.5221	0.5173	0.5113
HUE	0.4877	0.4874	0.4975	0.4847	0.488	0.4921	0.4871
HCC	0.4858	0.4857	0.4929	0.4837	0.4882	0.4882	0.4859
HUE-SP	0.4884	0.4875	0.4969	0.4841	0.4862	0.4923	0.4881
KCOS	0.5007	0.5072	0.5212	0.5029	0.5063	0.5022	0.506
COOC-SP	0.4822	0.4752	0.492	0.479	0.4864	0.4904	0.4741
TU	0.502	0.5041	0.5322	0.5023	0.5038	0.5171	0.5039
TU-SP	0.5021	0.5046	0.5319	0.5006	0.5031	0.5132	0.5044
CIRC	0.4621	0.4519	0.4588	0.4532	0.4553	0.4606	0.4516
CONV	0.456	0.4524	0.4595	0.4567	0.4597	0.4592	0.4526
ELIP	0.47	0.4708	0.4716	0.4725	0.471	0.4772	0.4694
RECT	0.4565	0.4593	0.4664	0.4573	0.4625	0.4665	0.4607

Table 6.5: Single histogram performance CHEN.

Feature	d0	d1	d2	d3	d4	d5	d6
CL	0.4372	0.4193	0.4524	**0.4244**	**0.4283**	0.4313	0.4162
CL-SP	**0.4391**	**0.4278**	0.4526	0.4233	**0.4283**	0.4376	**0.4291**
COOC	0.4243	0.4124	**0.4542**	0.413	0.4215	0.4325	0.4136
MSSDT	0.3587	0.3537	0.3584	0.3555	0.359	0.3602	0.3523
ORNT	0.406	0.4007	0.4225	0.3998	0.4104	0.3913	0.3985
HENT	0.3869	0.3933	0.3956	0.3903	0.3928	0.3862	0.3952
CLSH	0.4313	0.4136	0.4497	0.4197	0.4246	**0.4409**	0.4107
LBLK	0.389	0.4003	0.3981	0.3959	0.4027	0.3754	0.4032
LDIFF	0.3826	0.3937	0.3882	0.3887	0.3983	0.3736	0.3966
BKDT	0.3495	0.3636	0.3525	0.3648	0.3758	0.3489	0.3643
BKDT2D	0.3968	0.3975	0.4102	0.4016	0.4051	0.3823	0.3963
TARE	0.4078	0.3925	0.3983	0.3942	0.4026	0.4033	0.3928
TRAT	0.3898	0.3869	0.3868	0.3887	0.3957	0.3936	0.3859
TLEN	0.3953	0.3948	0.3964	0.3944	0.3999	0.3933	0.3944
CARC	0.3904	0.3983	0.4003	0.3967	0.4031	0.395	0.3978
HUE	0.3872	0.382	0.4085	0.3824	0.3903	0.3899	0.3815
HCC	0.3876	0.381	0.406	0.3843	0.3913	0.3844	0.3808
HUE-SP	0.3862	0.3824	0.4081	0.3816	0.3897	0.3896	0.3823
KCOS	0.3921	0.3899	0.4071	0.3911	0.3954	0.4009	0.3857
COOC-SP	0.4182	0.4125	0.443	0.4102	0.4173	0.4267	0.4142
TU	0.4117	0.4009	0.4236	0.4077	0.4045	0.4169	0.3989
TU-SP	0.4121	0.4019	0.4234	0.4071	0.4036	0.417	0.4004
CIRC	0.3654	0.3751	0.368	0.3714	0.3742	0.3687	0.3735
CONV	0.3398	0.3787	0.3643	0.3669	0.3686	0.3562	0.3811
ELIP	0.3412	0.3668	0.364	0.3616	0.3677	0.3536	0.3666
RECT	0.3403	0.3803	0.3614	0.3663	0.3676	0.3557	0.3842

Table 6.6: Single histogram performance ART.

reason we performed the Kruskall-Wallis statistical test using $\alpha = 0.005$. The value of the statistic gives only an indication if there *is* or *is not* a significant difference between the tested samples. Then, we applied the Dunn multiple comparisons to find which pairs of methods (or distance measures) are significantly different.

Histogram Methods. The hypothesis is that all the methods perform similarly. This is translated to "The distributions of the performance values of different histogram methods, using a single distance measure, are the same". We performed the KS test for all the methods, for all the datasets, for all the distance measures. In all cases the hypothesis is rejected, indicating significant difference in the samples. The tables of the Dunn multiple comparisons are located in Appendix F.

Distance Measure. In this case the hypothesis is that the performance of a histogram method, using different distance measures, is the same. This is translated to "The distributions of the performance values of a single histogram method, using different distance measures, are the same". The results of the test on the four datasets, are listed in Table 6.7. The symbol "$\surd$" indicates that the hypothesis is rejected, while the symbol "$\times$" indicates otherwise.

Feature	Video	COIL	Chen	Artchive	Feature	Video	COIL	Chen	Artchive
CL	√	√	×	√	TLEN	√	√	×	×
CL-SP	√	√	×	√	CARC	√	√	×	×
COOC	√	√	×	√	HUE	√	√	×	√
MSSDT	√	√	×	√	HCC	√	√	×	√
ORNT	√	√	×	√	HUE-SP	√	√	×	√
HENT	√	√	×	×	KCOS	√	√	×	√
CLSH	√	√	×	√	COOC-SP	√	√	×	√
LBLK	√	√	√	√	TU	√	√	√	√
LDIFF	√	√	×	√	TU-SP	√	√	√	√
BKDT	√	√	√	√	CIRC	√	√	×	√
BKDT2D	√	√	×	√	CONV	√	√	×	√
TARE	√	√	×	√	ELIP	√	√	×	√
TRAT	√	√	×	×	RECT	√	√	×	√

Table 6.7: Does a different distance measure make any difference?

6.3 Histogram Combinations

In this section we present the results obtained by combining histogram methods. As an exhaustive search would be unrealistic (2^{26} possible histogram combinations to evaluate), we used a sequential forward search strategy. While the resulting solutions using this method may be sub-optimal, the search process is more reliable than combining the histograms in terms of their individual performance. In the previous section we found that the χ^2 goodness of fitness test performed best for the majority of the tests, but in this section we use the Euclidean distance since we wanted to use a more standard method to CBIR.

In the previous section we noticed cases of competition between spatial histogram methods and their non-spatial counterparts. Because of this, we performed more tests by excluding (in turn) the spatial and non-spatial versions. This allows us to investigate if the combination sequence changes and which of the two variants performs better.

6.3.1 Sequential Forward Search

The results obtain from the methods selected by the sequential forward search (SFS) are presented in tables 6.8, 6.9, 6.10 and 6.11. We run three experiments for each dataset, one that included all the methods presented in Chapter 3, and two using different subsets of histogram methods. In particular we want to see if there is any difference in performance contribution between the spatial and non-spatial versions.

6.3.2 Statistical Test of Significance

Video Samples. The Kruscal-Wallis test showed that there is no significant difference between the performance of the histogram combinations using the video samples dataset.

Features	**All**	ε	**Spatial**	ε	**Non-Spatial**	ε
2	HUE-SP + COOC	0.9856	HUE-SP + CC-SP	0.9869	HUE + COOC	0.9867
3	HENT	0.9874	CLSH	0.9883	CLSH	0.9882
4	BKDT2D	0.9875	BKDT2D	0.9888	BKDT2D	0.9884
5	CLSH	0.9887	LDIFF	0.9892	HENT	0.9896
6	LDIFF	0.9889	HCC	0.9896	LDIFF	0.9898
7	HUE	0.9895	TU-SP	0.9904	TARE	0.9899
8	ELIP	0.9894	BKDT	0.9901	TU	**0.9909**
9	TARE	0.9893	TARE	0.9902	ELIP	**0.9909**
10	CC-SP	0.9895	ELIP	0.9901	HCC	0.9908
11	CONV	0.9892	CIRC	0.9899	CL	0.9907
12	TU	0.9900	HENT	0.9900	BKDT	0.9906
13	HCC	0.9898	CONV	0.9899	CONV	0.9908
14	CL	**0.9904**	RECT	0.9894	CIRC	0.9906
15	BKDT	0.9898	TRAT	0.9891	RECT	0.9903
16	CIRC	0.9903	KCOS	0.9883	CARC	0.9900
17	RECT	0.9901	CARC	0.9878	TLEN	0.9894
18	KCOS	0.9899	ORNT	0.9867	KCOS	0.9892
19	TLEN	0.9900	TLEN	0.9864	LBLK	0.9887
20	TU-SP	0.9899	CL-SP	0.9851	ORNT	0.9886
21	LBLK	0.9895	MSSDT	0.9843	MSSDT	0.9880
22	ORNT	0.9893	LBLK	0.9815	TRAT	0.9870
23	CL-SP	0.9885				
24	MSSDT	0.9882				
25	CARC	0.9878				
26	TRAT	0.9860				

Table 6.8: Performance of combinations of histograms using the Video samples of image dataset.

Features	**All**	ε	**Spatial**	ε	**Non-Spatial**	ε
2	CL-SP + CLSH	0.9281	CL-SP + CLSH	0.9324	CL + CLSH	0.8980
3	ONRNT	0.9300	ORNT	0.9342	HCC	0.9125
4	HCC	**0.9369**	HCC	**0.9411**	BKDT2D	**0.9132**
5	BKDT2D	0.9361	BKDT2D	0.9403	CONV	0.9091
6	CONV	0.9316	CONV	0.9359	ORNT	0.9101
7	TRAT	0.9301	TRAT	0.9346	TRAT	0.9089
8	RECT	0.9265	RECT	0.9312	RECT	0.9058
9	CARC	0.9235	CARC	0.9285	BKDT	0.8989
10	BKDT	0.9166	TLEN	0.9253	CARC	0.8969
11	TLEN	0.9136	ELIP	0.917	RLIP	0.8897
12	ELIP	0.9056	TARE	0.9125	TLEN	0.887
13	KCOS	0.9039	KCOS	0.9103	KCOS	0.8869
14	TARE	0.902	BKDT	0.9047	HENT	0.8848
15	HUE	0.8988	CC-CP	0.9029	HUE	0.8847
16	CIRC	0.8835	CIRC	0.8875	TARE	0.8856
17	HENT	0.879	MSSDT	0.8806	CIRC	0.8688
18	CC-SP	0.8753	HDIFF	0.8765	LBLK	0.8581
19	COOC	0.8681	LBLK	0.8609	TU	0.8539
20	MSSDT	0.8555	HENT	0.8392	LDIFF	0.8325
21	LDIFF	0.8581	HCC-SP	0.8384	COOC	0.8232
22	TU	0.8533	TU-SP	0.8293	MSSDT	0.819
23	LBLK	0.8339				
24	CL	0.825				
25	HUE-SP	0.8207				
26	TU-SP	0.8122				

Table 6.9: Performance of combinations of histograms using the COIL-100 of image dataset.

Features	**All**	ε	**Spatial**	ε	**Non-Spatial**	ε
2	CL + TARE	0.6479	CL-SP + CLSH	0.6303	CL + TARE	0.479
3	COOC	0.6498	TLEN	0.6408	BKDT	0.6507
4	BKDT	0.6598	HENT	0.6424	CLSH	0.6565
5	CC-SP	0.6643	BKDT	0.6503	HENT	0.6600
6	HENT	0.6696	CC-CP	0.6567	TU	0.6636
7	CLSH	0.6732	BKDT2D	0.6568	COOC	0.6659
8	TLEN	0.6754	TARE	0.6559	TLEN	0.6752
9	CL-SP	0.6775	ORNT	**0.6578**	BKDT2D	0.6755
10	TU-SP	0.6799	TU-SP	0.6554	ELIP	**0.6761**
11	BKDT2D	0.6813	LBLK	0.6557	CARC	0.6751
12	ELIP	**0.6811**	CONV	0.655	RECT	0.6733
13	CARC	0.6804	CARC	0.6523	TRAT	0.6691
14	RECT	0.6783	ELIP	0.6497	CONV	0.6659
15	ONRNT	0.6752	RECT	0.6453	CIRC	0.6609
16	TRAT	0.6746	CIRC	0.6399	HUE	0.6566
17	CONV	0.6714	KCOS	0.6363	KCOS	0.6488
18	CIRC	0.6677	HCC	0.6279	LDIFF	0.6348
19	HUE-SP	0.6615	HDIFF	0.6214	HCC	0.6419
20	KCOS	0.6621	TRAT	0.6216	MSSDT	0.6384
21	HCC	0.6514	HCC-SP	0.6208	ORNT	0.6344
22	TU	0.648	MSSDT	0.6072	LBLK	0.6295
23	LBLK	0.6405				
24	MSSDT	0.6405				
25	HUE	0.6367				
26	LDIFF	0.6284				

Table 6.10: Performance of combinations of histograms using the Chen collection of images.

Features	**All**	ε	**Spatial**	ε	**Non-Spatial**	ε
2	CL-SP - TARE	0.4303	CL-SP + TARE	0.4303	CL - TARE	0.4301
3	BKDT2D	0.4348	BKDT2D	0.4348	TU	0.4454
4	TU	0.4508	TU-SP	0.4506	BKDT2D	0.4501
5	CLSH	0.4555	CLSH	0.4551	HENT	0.4539
6	HENT	0.4589	CARC	0.4582	CARC	0.4568
7	CARC	0.4619	HDIFF	0.4598	LDIFF	0.4582
8	LDIFF	0.4632	TLEN	0.46	CIRC	0.4573
9	TLEN	0.4634	CC-CP	0.4616	LBLK	0.4562
10	CL	0.4636	HCC	**0.4618**	TLEN	0.4573
11	TU-SP	0.4667	KCOS	0.4613	COOC	0.458
12	KCOS	0.4666	CIRC	0.46	HCC	0.4602
13	CC-SP	0.4683	ELIP	0.4591	KCOS	0.4597
14	HCC	**0.4688**	CONV	0.4574	CLSH	0.4601
15	LBLK	0.4681	HCC-SP	0.4557	BKDT	0.4616
16	CIRC	0.4676	RECT	0.4538	ORNT	**0.4636**
17	CONV	0.467	ORNT	0.4531	CONV	0.4631
18	ELIP	0.4659	LBLK	0.4535	RECT	0.4621
19	RECT	0.4644	HENT	0.4502	ELIP	0.4605
20	ONRNT	0.4627	BKDT	0.4511	HUE	0.4598
21	HUE-SP	0.4633	MSSDT	0.4526	TRAT	0.4553
22	COOC	0.4627	TRAT	0.4498	MSSDT	0.4476
23	TRAT	0.4612				
24	HUE	0.4592				
25	BKDT	0.4565				
26	MSSDT	0.4557				

Table 6.11: Performance of combinations of histograms using the Artchive collection of images.

The hypothesis was not rejected for all the cases (including the Spatial only, and Non-Spatial only, experiments).

COIL-100. In the case of the performance values obtained using the COIL-100 dataset, the Kruskall-Wallis found significant differences. The Dunn multiple comparisons showed significant differences throughout the range of methods, for all three cases.

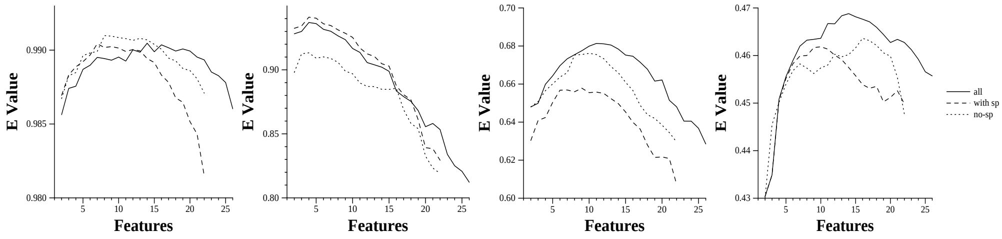
Chen Collection. Significant differences were also found with the Chen collection dataset. For the case where *all* the histograms were considered for combinations (Figure F.4), the majority of significant differences were found between methods utilising 5 to 20 histograms and 22 to 26 histograms. For the case where *spatial only* histograms are considered (Figure F.6), the majority of significant differences were found between methods utilising 6 to 14 histograms and 18 to 22 histograms. For the case where the *non-spatial only* histogram are considered (Figure F.5), the majority of significant differences were found between methods utilising 3 to 16 histograms and 19 to 22 histograms.

Artchive. Again, we found significant differences with the Artchive dataset. For the case where *all* the histograms were considered for combinations (Figure F.7), the majority of significant differences were found between methods utilising up to 5 histograms. For the case where *spatial only* histograms are considered (Figure F.9), the majority of significant differences were found between methods utilising 13 to 15 histograms and 25 histograms. For the case where the *non-spatial only* histogram are considered (Figure F.8), the majority of significant differences were found between methods utilising 11 to 17 histograms and 26 histograms.

6.3.3 Summary

We noticed that different feature sequences were returned by the sequential forward search for the four different datasets. This confirms the strong relationship between the performance of a histogram method and a dataset. We believe that this relationship is not only influenced by how well the features are extracted from the images, but also, on how the ground truth is established.

For the Video samples and the COIL-100 datasets the groundtruth is very distinctive. The groups of images are very well defined and their overlap is very limited. On the other



(a) Video (b) Coil (c) Chen (d) Artchive

Figure 6.1: Number of combined histograms and performance.

side, for the Chen collection and the Artchive a high level approach was used to establish the groundtruth. For the case of the Chen collection, image subjects grouped the images. The similarity between images of a group is based on high level information, that does not necessarily reflect the similarities/dissimilarities of their low-level features. For the case of the Artchive images, grouping was based solely on the name of the artist that painted the picture. The use of such small amount of information (of high semantic value) makes the possible association between low level feature similarity and image grouping even harder. The levels of performance of the histogram combinations, reflect the above points.

When we tested the individual histogram methods we noticed that the colour and texture based features dominated the Top-10 performing ranks. When we tested combinations of histograms the variety of methods appearing early in the search, increased. We also observed histograms that performed poorly (in the individual tests), to be members of the best performing combinations (e.g. TARE). While searching through combinations and adding more histograms, the amount of information for image comparisons increases. This extra information typically improved the performance. Of course we can not add features indefinetly, as after some point in the search this extra information becomes counterproductive. This peaking phenomenon is illustrated in Figure 6.1, where the numbers of histograms *vs* performance is plotted, for the three group of histogram methods and the four datasets.

Rank	d0	d1	d2	d3	d4	d5	d6
1	HUE-SP	HUE-SP	HUE-SP	HUE-SP	HUE-SP	HUE-SP	HUE-SP
2	HUE	HUE	HUE	HUE	HUE	HUE	CL-SP
3	HCC	HCC	HCC	HCC	HCC	HCC	HUE
4	TU-SP	CL	TU	CL	CL	TU	CL
5	TU	CL-SP	TU-SP	TU	TU	TU-SP	HCC
6	COOC-SP	COOC-SP	COOC	COOC-SP	COOC	COOC-SP	COOC-SP
7	CL	CLSH	CL	CL-SP	COOC-SP	COOC	CLSH
8	COOC	TU-SP	COOC-SP	COOC	TU-SP	CL	TU-SP
9	CL-SP	COOC	CL-SP	TU-SP	CL-SP	CL-SP	COOC
10	CLSH	TU	CLSH	CLSH	CLSH	CLSH	TU
11	HENT	HENT	ORNT	HENT	HENT	HENT	HENT
12	ORNT	ORNT	HENT	ORNT	ORNT	ORNT	LBLK
13	LBLK	LBLK	LBLK	LBLK	LBLK	LBLK	ORNT
14	LDIFF	LDIFF	LDIFF	BKDT2D	BKDT2D	LDIFF	LDIFF
15	BKDT2D	BKDT2D	BKDT2D	LDIFF	LDIFF	MSSDT	BKDT2D
16	MSSDT	BKDT	KCOS	BKDT	BKDT	BKDT2D	BKDT
17	KCOS	TRAT	MSSDT	TRAT	TRAT	KCOS	TRAT
18	TARE	TARE	TLEN	TLEN	TLEN	TARE	TARE
19	TLEN	TLEN	ELIP	KCOS	TARE	TLEN	CARC
20	BKDT	KCOS	TARE	TARE	KCOS	BKDT	TLEN
21	TRAT	ELIP	TRAT	ELIP	ELIP	ELIP	ELIP
22	CARC	CARC	BKDT	MSSDT	MSSDT	TRAT	KCOS
23	CIRC	CIRC	CARC	CARC	CARC	CARC	RECT
24	ELIP	MSSDT	CIRC	CIRC	CIRC	CIRC	CIRC
25	CONV	RECT	CONV	RECT	RECT	CONV	MSSDT
26	RECT	CONV	RECT	CONV	CONV	RECT	CONV

Table 6.12: Method ranking based on performance values using the Video samples dataset.

Rank	d0	d1	d2	d3	d4	d5	d6
1	CL-SP	CL-SP	HUE	CL-SP	HUE	CL-SP	CL-SP
2	CL	HUE	CL-SP	HUE	CL-SP	HUE	HUE
3	HCC	HCC	HCC	CL	HCC	CL	HCC
4	CLSH	CL	CL	HCC	CL	HCC	CL
5	HUE	HUE-SP	HUE-SP	CLSH	CLSH	CLSH	HUE-SP
6	HUE-SP	CLSH	CLSH	COOC	COOC	HUE-SP	CLSH
7	COOC-SP	COOC	COOC	COOC-SP	COOC-SP	COOC	COOC
8	COOC	COOC-SP	COOC-SP	HUE-SP	HUE-SP	COOC-SP	COOC-SP
9	TU	HENT	TU	TU	TU	TU	HENT
10	TU-SP	BKDT2D	TU-SP	BKDT2D	BKDT2D	TU-SP	BKDT2D
11	HENT	TU	HENT	HENT	HENT	ORNT	TU
12	LDIFF	TU-SP	ORNT	LDIFF	LBLK	HENT	TU-SP
13	LBLK	LDIFF	LDIFF	LBLK	LDIFF	MSSDT	LDIFF
14	ORNT	LBLK	LBLK	ORNT	ORNT	LDIFF	LBLK
15	MSSDT	ORNT	KCOS	TU-SP	BKDT	LBLK	ORNT
16	KCOS	BKDT	MSSDT	BKDT	TU-SP	KCOS	BKDT
17	BKDT	KCOS	BKDT	KCOS	KCOS	BKDT	KCOS
18	BKDT2D	ELIP	BKDT2D	MSSDT	MSSDT	BKDT2D	RECT
19	TLEN	CONV	TLEN	TLEN	TLEN	TLEN	ELIP
20	CARC	RECT	TARE	TRAT	TARE	TRAT	CONV
21	TARE	TRAT	TRAT	TARE	TRAT	TARE	TRAT
22	TRAT	TLEN	CARC	CARC	CARC	CARC	TARE
23	ELIP	CIRC	ELIP	CONV	ELIP	ELIP	CIRC
24	CIRC	TARE	CIRC	CIRC	CONV	CIRC	TLEN
25	CONV	MSSDT	CONV	ELIP	CIRC	CONV	MSSDT
26	RECT	CARC	RECT	RECT	RECT	RECT	CARC

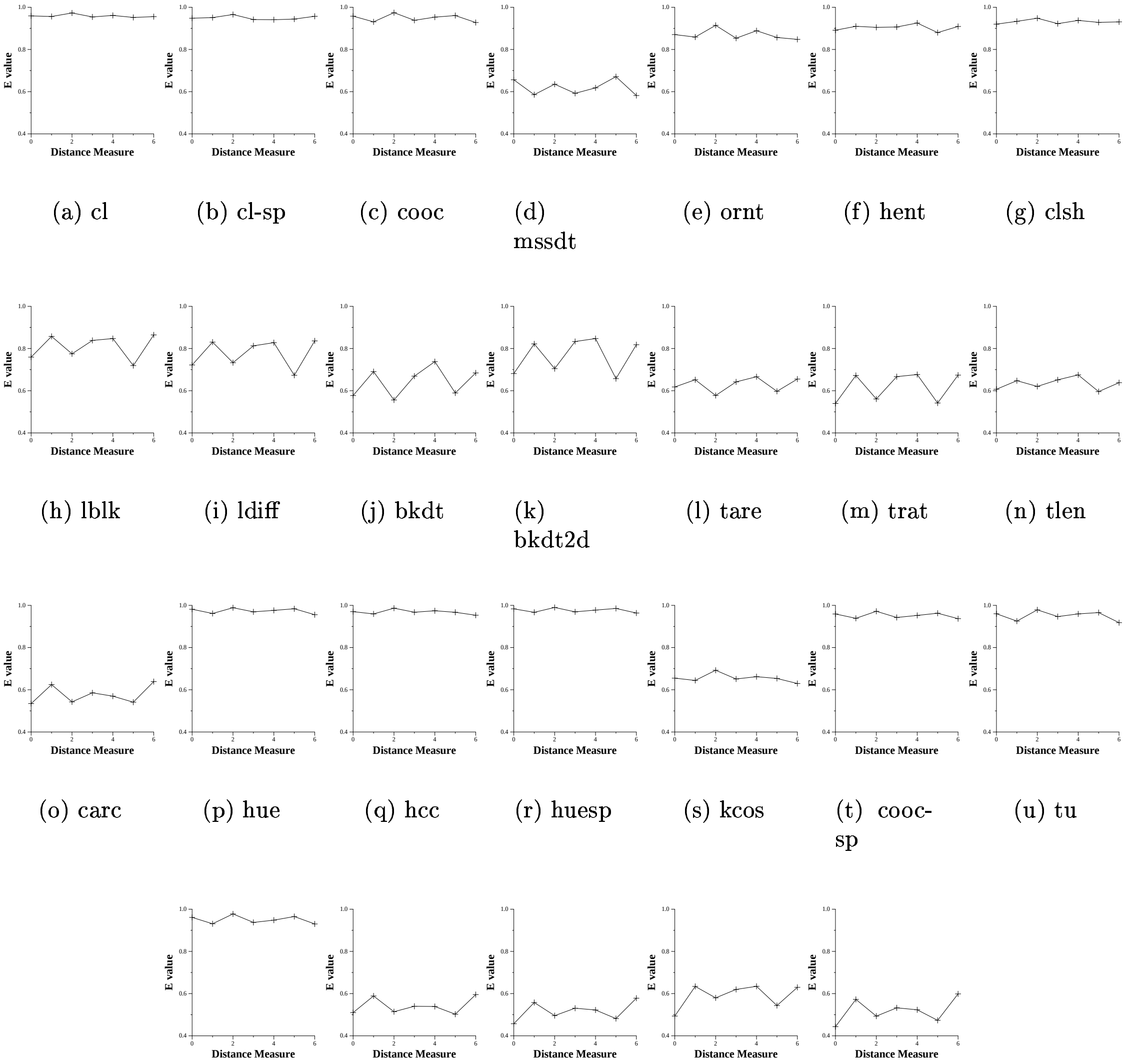
Table 6.13: Method ranking based on performance values using the Coil-100 dataset.

Rank	**d0**	**d1**	**d2**	**d3**	**d4**	**d5**	**d6**
1	HUE-SP	HUE-SP	HUE-SP	HUE-SP	HUE-SP	HUE-SP	HUE-SP
2	HUE	HUE	HUE	HUE	HUE	HUE	CL-SP
3	HCC	HCC	HCC	HCC	HCC	HCC	HUE
4	TU-SP	CL	TU	CL	CL	TU	CL
5	TU	CL-SP	TU-SP	TU	TU	TU-SP	HCC
6	COOC-SP	COOC-SP	COOC	COOC-SP	COOC	COOC-SP	COOC-SP
7	CL	CLSH	CL	CL-SP	COOC-SP	COOC	CLSH
8	COOC	TU-SP	COOC-SP	COOC	TU-SP	CL	TU-SP
9	CL-SP	COOC	CL-SP	TU-SP	CL-SP	CL-SP	COOC
10	CLSH	TU	CLSH	CLSH	CLSH	CLSH	TU
11	HENT	HENT	ORNT	HENT	HENT	HENT	HENT
12	ORNT	ORNT	HENT	ORNT	ORNT	ORNT	LBLK
13	LBLK	LBLK	LBLK	LBLK	LBLK	LBLK	ORNT
14	LDIFF	LDIFF	LDIFF	BKDT2D	BKDT2D	LDIFF	LDIFF
15	BKDT2D	BKDT2D	BKDT2D	LDIFF	LDIFF	MSSDT	BKDT2D
16	MSSDT	BKDT	KCOS	BKDT	BKDT	BKDT2D	BKDT
17	KCOS	TRAT	MSSDT	TRAT	TRAT	KCOS	TRAT
18	TARE	TARE	TLEN	TLEN	TLEN	TARE	TARE
19	TLEN	TLEN	ELIP	KCOS	TARE	TLEN	CARC
20	BKDT	KCOS	TARE	TARE	KCOS	BKDT	TLEN
21	TRAT	ELIP	TRAT	ELIP	ELIP	ELIP	ELIP
22	CARC	CARC	BKDT	MSSDT	MSSDT	TRAT	KCOS
23	CIRC	CIRC	CARC	CARC	CARC	CARC	RECT
24	ELIP	MSSDT	CIRC	CIRC	CIRC	CIRC	CIRC
25	CONV	RECT	CONV	RECT	RECT	CONV	MSSDT
26	RECT	CONV	RECT	CONV	CONV	RECT	CONV

Table 6.14: Method ranking based on performance values using the Chen dataset.

Rank	**d0**	**d1**	**d2**	**d3**	**d4**	**d5**	**d6**
1	CL-SP	CL-SP	COOC	CL	CL	CLSH	CL-SP
2	CL	CL	CL-SP	CL-SP	CL-SP	CL-SP	CL
3	CLSH	CLSH	CL	CLSH	CLSH	COOC	COOC-SP
4	COOC	COOC-SP	CLSH	COOC	COOC	CL	COOC
5	COOC-SP	COOC	COOC-SP	COOC-SP	COOC-SP	COOC-SP	CLSH
6	TU-SP	TU-SP	TU	TU	ORNT	TU-SP	LBLK
7	TU	TU	TU-SP	TU-SP	BKDT2D	TU	TU-SP
8	TARE	ORNT	ORNT	BKDT2D	TU	TARE	TU
9	ORNT	LBLK	BKDT2D	ORNT	TU-SP	KCOS	ORNT
10	BKDT2D	CARC	HUE	CARC	CARC	CARC	CARC
11	TLEN	BKDT2D	HUE-SP	LBLK	LBLK	TRAT	LDIFF
12	KCOS	TLEN	KCOS	TLEN	TARE	TLEN	BKDT2D
13	CARC	LDIFF	HCC	TARE	TLEN	ORNT	HENT
14	TRAT	HENT	CARC	KCOS	LDIFF	HUE	TLEN
15	LBLK	TARE	TARE	HENT	TRAT	HUE-SP	TARE
16	HCC	KCOS	LBLK	LDIFF	KCOS	HENT	TRAT
17	HUE	TRAT	TLEN	TRAT	HENT	HCC	KCOS
18	HENT	HUE-SP	HENT	HCC	HCC	BKDT2D	RECT
19	HUE-SP	HUE	LDIFF	HUE	HUE	LBLK	HUE-SP
20	LDIFF	HCC	TRAT	HUE-SP	HUE-SP	LDIFF	HUE
21	CIRC	RECT	CIRC	CIRC	BKDT	CIRC	CONV
22	MSSDT	CONV	CONV	CONV	CIRC	MSSDT	HCC
23	BKDT	CIRC	ELIP	RECT	CONV	CONV	CIRC
24	ELIP	ELIP	RECT	BKDT	ELIP	RECT	ELIP
25	RECT	BKDT	MSSDT	ELIP	RECT	ELIP	BKDT
26	CONV	MSSDT	BKDT	MSSDT	MSSDT	BKDT	MSSDT

Table 6.15: Method ranking based on performance values using the Artchive dataset.



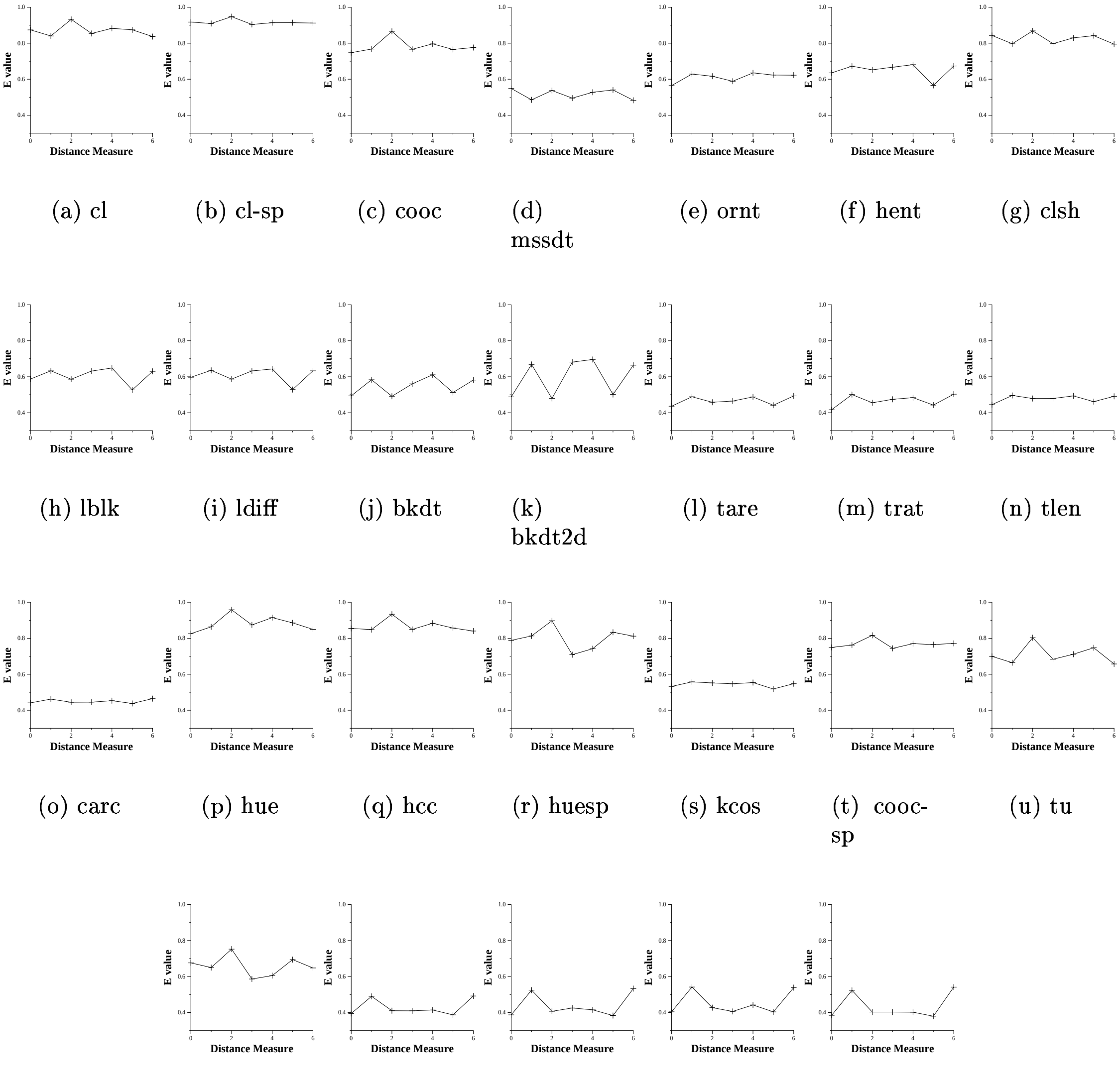
(a) cl (b) cl-sp (c) cooc (d) mssdt (e) ornt (f) hent (g) clsh

(h) lblk (i) ldiff (j) bkdt (k) bkdt2d (l) tare (m) trat (n) tlen

(o) carc (p) hue (q) hcc (r) huesp (s) kcos (t) cooc-sp (u) tu

(v) tu-sp (w) circ (x) conv (y) elip (z) rect

Figure 6.2: Performance levels of distance measures using the Video samples.



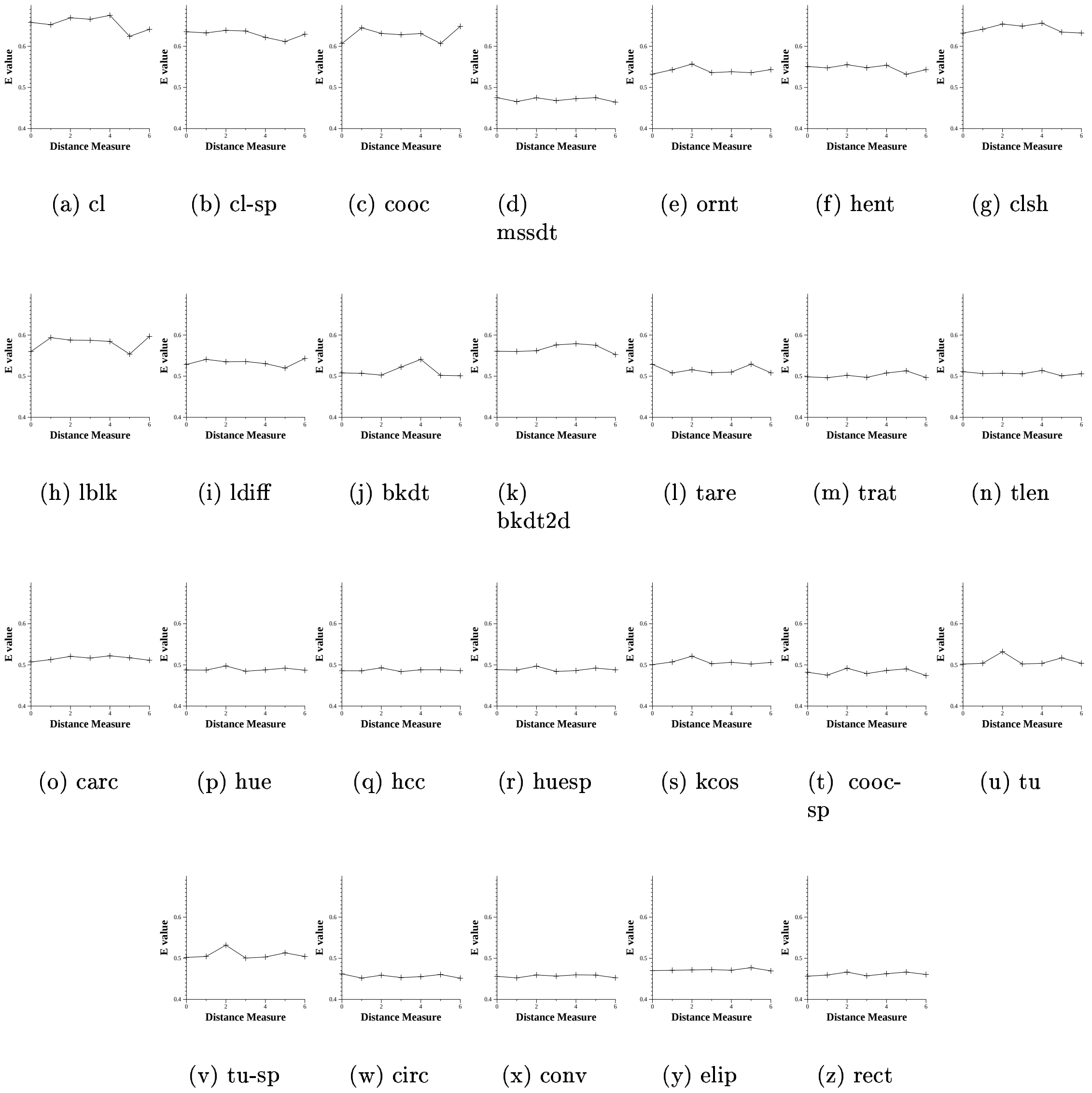
(a) cl (b) cl-sp (c) cooc (d) mssdt (e) ornt (f) hent (g) clsh

(h) lblk (i) ldiff (j) bkdt (k) bkdt2d (l) tare (m) trat (n) tlen

(o) carc (p) hue (q) hcc (r) huesp (s) kcos (t) cooc-sp (u) tu

(v) tu-sp (w) circ (x) conv (y) elip (z) rect

Figure 6.3: Performance levels of distance measures using the COIL-100 Library.



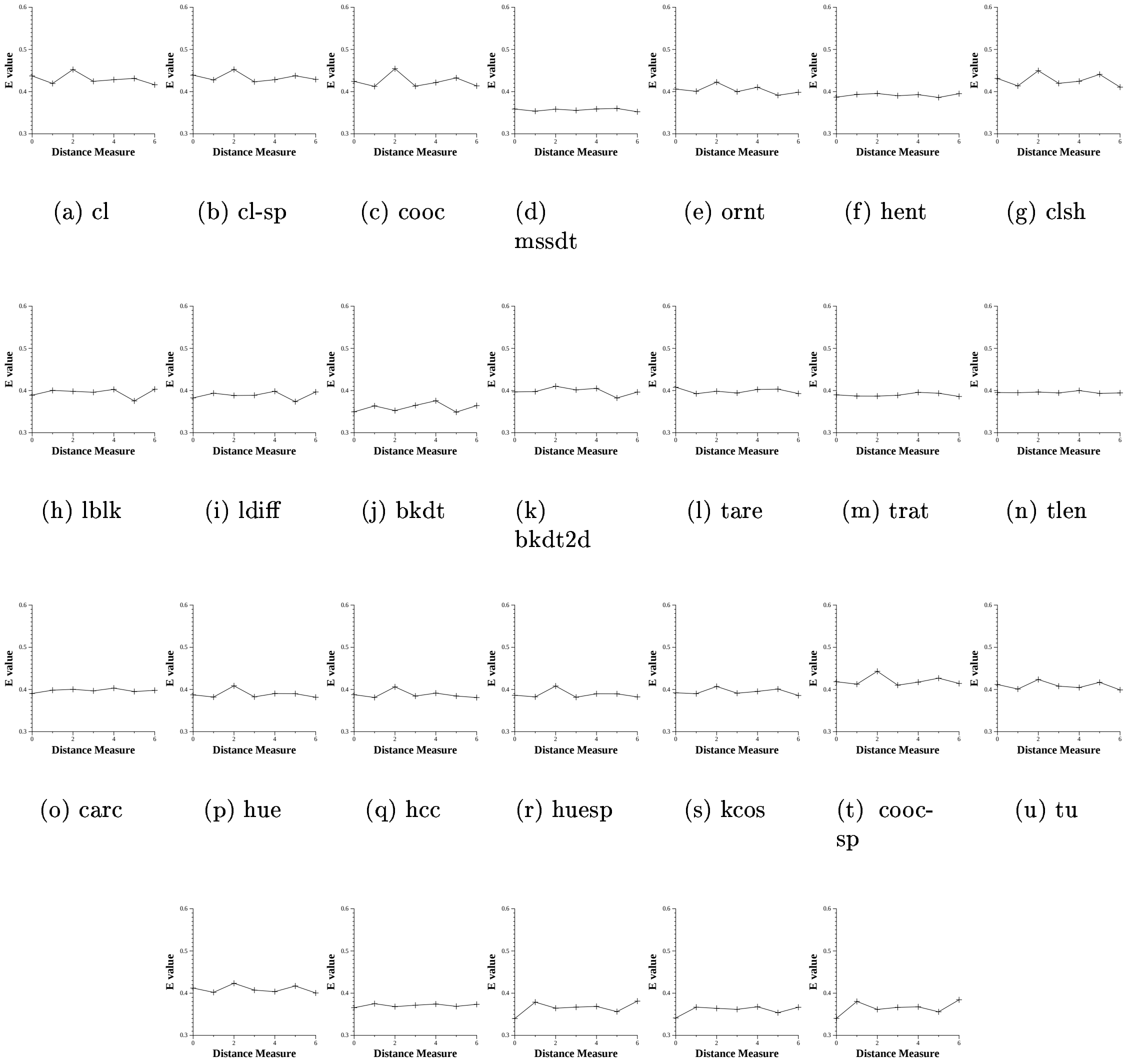
(a) cl (b) cl-sp (c) cooc (d) mssdt (e) ornt (f) hent (g) clsh

(h) lblk (i) ldiff (j) bkdt (k) bkdt2d (l) tare (m) trat (n) tlen

(o) carc (p) hue (q) hcc (r) huesp (s) kcos (t) cooc-sp (u) tu

(v) tu-sp (w) circ (x) conv (y) elip (z) rect

Figure 6.4: Performance levels of distance measures using the Chen collection.



(a) cl (b) cl-sp (c) cooc (d) mssdt (e) ornt (f) hent (g) clsh

(h) lblk (i) ldiff (j) bkdt (k) bkdt2d (l) tare (m) trat (n) tlen

(o) carc (p) hue (q) hcc (r) huesp (s) kcos (t) cooc-sp (u) tu

(v) tu-sp (w) circ (x) conv (y) elip (z) rect

Figure 6.5: Performance levels of distance measures using the Artchive.

Chapter 7

Conclusions

In this chapter we draw conclusions from the results presented in the preceeding Chapters and we discuss the possibilities for future work.

The ideas presented in each of the Chapters 3, 4 and 5 advance the state of the art. In two cases, specific histogram methods for extracting shape from images have led to published conference [36] and later in journal [38] publication, some of the remaining shape histogram methods have been published in a conference paper [37] and later, with the addition of parts of the evaluation framework in a journal paper[39]. Furthermore, the use of CBIR methods with Pathfinder networks, presented in Section 5.4, is published as a conference paper [40], which led to a series of further conference publications [18][16][17].

Using the three main objectives as a ruler, in the following sections we summarise the drawn conclusions, *i*)Low level feature extraction (Section 7.1), *ii*) Concept-based image retrieval (Section 7.2), and *iii*) Performance evaluation (Section 7.3). The thesis closes with some brief considerations about future work (Section 7.4).

7.1 Low level Features

The majority of the methods presented in Chapter 3 advance the state of the art. The histograms presented in Sections 3.2.4 and 3.3.2 have been used in CBIR (but not in the form presented in this thesis), while the histograms in Sections 3.2.1, 3.2.2, 3.2.3, 3.2.5, 3.2.6, 3.4, 3.5.1, 3.5.2, 3.5.3, 3.6.1, 3.6.2 are original in CBIR.

Evaluation of the obtained results led to the following conclusions:

- Histograms can be used with some success to summarise aspects of *texture* and *shape* from images, and then utilised for CBIR.

- Experiments of individual histograms showed that colour features were the best performing features. Overall, texture histograms performed better than shape histogram in the majority of the experiments. Promising performance was noticed by colour and distance (Section) and hybrid histogram methods (Section) outperforming plain shape and in some cases texture histogram methods. While the observed mean performance differences were small (in some cases 1%), statistical analysis using non-parametric tests indicated that the differences were significant in many cases.

- Experimentation of the individual histograms using different distance measures showed that the Euclidean distance (a more standard method in CBIR) was outperformed by the χ^2, *Kuiper*, and the *edit distance*. We did not notice any strong correlation between these distance measures and the feature aspects (colour, texture, shape *etc.*) captured by the histograms.

- The majority of the histogram methods showed tolerance to transformations, particularly to translation and less to rotation, while the majority of the histograms showed increased sensitivity to scale.

- Combining histograms resulted in improved performance. We used a sequential forward search (SFS) to find the best performing combinations. While the solutions may be sub-optimal, SFS is a better approach than arbitrary selection of histogram combinations, while a fully exhaustive search would be impractical. A peaking phenomenon is noticed in CBIR performance as the number of combined histograms increases, with the optimum performance in the range of 4 to 14 histograms. Statistically significant differences were noticed when combining a small (2-4) or a medium (6-12) number of histograms.

- Individual histograms incorporating spatial information by soft-thresholding did not outperform their non-spatial versions. The outcome was reversed when in the many cases the spatial versions showed greater contribution to performance of histogram combinations.

- A performance difference was noticed when we used different image datasets. That was expected as the groundtruth of each dataset was defined using information of different levels of abstraction. While performance was better when the groundtruth was automatically generated (Video samples and COIL-100), the weakness of the histogram approach to extract semantic features became apparent with the other two datasets (Chen Collection and Artchive).

7.2 Concept-Based Image Retrieval

BEKAS is proposed as an approach to overcome the deficiency of the histogram methods (presented in Chapter 3), to extract semantic features. The advances to the state of the art are due to, *i*) an algorithm for meaningful segmentation is not required, and *ii*) the use of the knowledge structure.

Study of the implemented modules of BEKAS led to the following conclusions:

- Histograms can be used along with a learning technique to assign semantic labels to regions in images. With our experimental dataset successful assignments of labels in the range of 60%-70% were achieved, using *i*) Decision Trees, and *ii*) Multi-Layer Perceptron.
- The spatial dependency matrix (SPD) is essentially a histogram that counts the instances of labels at a given direction (multiple SPDs are used to cover a range of directions) and distance range. Multiple SPD sets can be combined together to form a memory, that allows use of SPDs as templates.

7.3 Performance Evaluation

In Chapter 5 we discuss about a performance evaluation framework for histogram based CBIR systems. The framework is of the form of a compilation of methods rather than a strict testing protocol. The methodology presented advances the state of the art, in providing a broader range of methodologies that allows a more thorough testing of histogram based CBIR methods (while part of the methodology may be adopted to non-histogram based CBIR methods).

Evaluation of the obtained results led to the following conclusions, with respect to the usefulness of the methodologies included in the framework:

- *Groundtruth.* The quality of the groundtruth plays a major role, mainly, in quantitative performance evaluation. Subjective grouping, if done poorly, may lead to unreliable results, while automatic groundtruth methods my be misleading if oversimplified (very strong similarities between images of the same group). On the other hand, automatic groundtruth allows use of image sets of greater volume, as the time required to establish such groundtruth, is substantially lower than manual methods (subjective grouping).

- *Image sets.* A variety of image sets allowed us to test the histogram methods within different scenarios in the CBIR context. This way we obtained a broader picture of performance.

- *Transformation sequences.* The use of transformation sequences enabled us to extract from the system information about the tolerance of histogram methods to a number of image transformations (translation, rotation and scale). (Could be better if the sequences were selected more carefully, like specific shapes, various levels of complexity etc.)

- *Histogram plots* were more useful when created from the grouped image sets. This way we managed to get a feel about the histogram bin range occupancy, and visualise differences or similarities between histograms within single groups, or between different groups. Histogram plots using transformation sequences enabled us to visualise the tolerance of the histogram shape under different transformations. The whole process of assessment was not very easy, specially for the transformation sequences. Often, the only way to get enough information was to apply numerous image enhancing techniques.

- *Similarity Graph.* In some cases it was easier to pick up information about tolerance to transformations from the distance graphs (e.g. rotation sensitivity was related to a 'wave' like curve). Generating similarity graphs from grouped image sets did not provide much useful information with respect to CBIR.

- The *Distance matrix* provided a better picture of system performance when generated from grouped images. Occasionally a distance matrix of a method would suffer from poor visibility. While some image enhancement helped, in many cases the plot of the two sets of distances (within and between groups) provided extra information, reducing the need for image enhancement.

- The *Noise Dependency* scatter plots were useful as we verified that the histograms were not summarising the noise aspects of images.
- *Query Effectiveness.* We favoured average performance values (E-measure) as it allowed us easier comparisons between the different histogram methods we developed.
- *Statistical analysis*, proved to be a very useful tool in verifying the differences in performance of methods. In particular the *Kruskal-Wallis* test allowed very easy comparisons of multiple methods in single runs of a program. Furthermore, if a significant difference between the methods was identified, pair-wise tests (Dunn multiple comparisons) allowed us to find the pairs of methods with significant differences.
- *Pathfinder networks.* The generation of PFNETs allowed us to "walk" through image datasets as they were grouped by the CBIR system.

Overall, it is important to test CBIR methods from as many angles as possible, using both qualitative and quantitative methods.

7.4 Future Work

In this thesis we presented a number of histogram features for use in CBIR, covering a broad spectrum of image aspects. Some issues outside the scope of this thesis still remain (related to histogram resolution, binning process, *etc.*), but we think that interest on these issues is more application related rather than research related.

Research in CBIR shows a trend towards semantic sensitive CBIR. We think that BEKAS is a step towards that direction. At its current state we have identified several issues, which suggest lines for future research:

- Erroneous classification of semantic labels may affect the performance of the system. Classification accuracy may be improved by reducing the dimensionality of the histogram. In this thesis we used the first five moments to achieve that. Alternative ways are also possible using some perceptual mapping (association of histogram characteristics to simple, coarsely categorised, semantic terms [70]). Furthermore, post processing techniques (e.g. relaxation labelling) may improve the performance of classification.

- Training of the semantic labelling module is a step that currently requires training data segmented by human subjects. It may be possible to avoid manual segmentation by developing novel data collection techniques that target individual images to specific semantic labels.

- The process of generating the concept hierarchy is manual, while fully automated techniques for generating concept hierarchies do exist. A possible improvement on the latter would be to generate concept hierarchies specific to CBIR. Still the task of associating SPD setup templates to the concept hierarchy remains manual.

- The performance evaluation framework presented in Chapter 5 provided a broad range of beneficial information regarding the performance characteristics of histogram methods. However, new techniques need to be developed for the evaluation of more complex systems, such as BEKAS.

- Scalability is another essential characteristic in CBIR systems. BEKAS is composed of a number of technologies that may pose as bottlenecks regarding scalability. This area would be a starting point for further work.

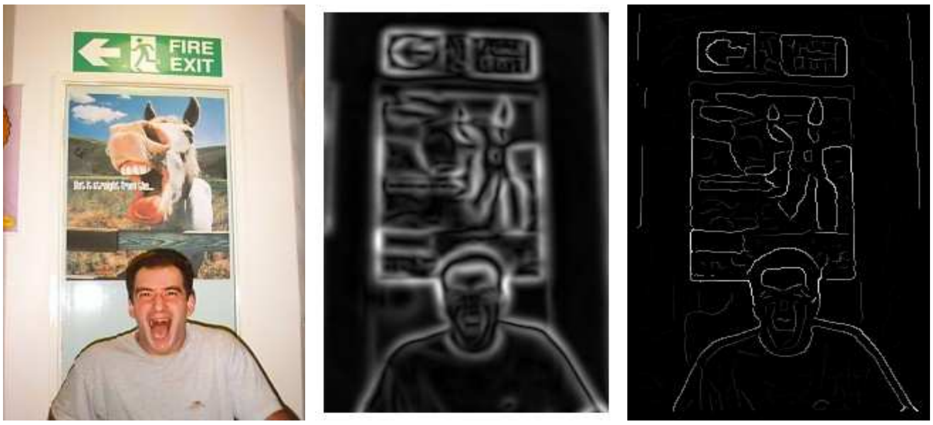
- The performance evaluation methods presented in this thesis required little human involvement. That is because the main objective related to performance evaluation is methods with high repeatability. A possible next step would be towards more active human involvement, targeting a mechanism which would generalise a small amount of information provided by human subject, thus allowing subjective tests on a greater scale and (after training) in a fully automatic fashion.

Appendix A

Multiscale Edge Detection

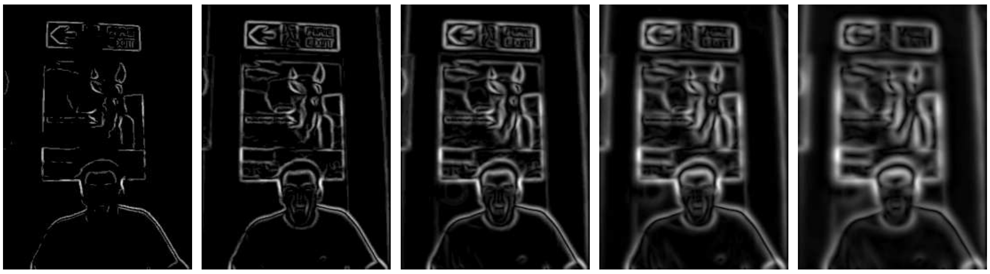
For the extraction of edges we used a multiscale variant of the Canny edge detector. The image we want to extract edges from, is smoothed using a Gaussian kernel, over a range of σ, and gradient maps are generated for each scale. The gradients maps are then averaged, to create a single map, on which we then apply non-maximal suppression. An example of this process is illustrated in Figure A.1.

The multiscale process (Figure A.2), acts as an ageing factor. Gaussian smoothing deteriorates the edge information gradually, so stronger edges are expected to survive to greater scales σ.



(a) RGB (b) Average Gradient (c) Extracted Edges

Figure A.1: Edge detection example.



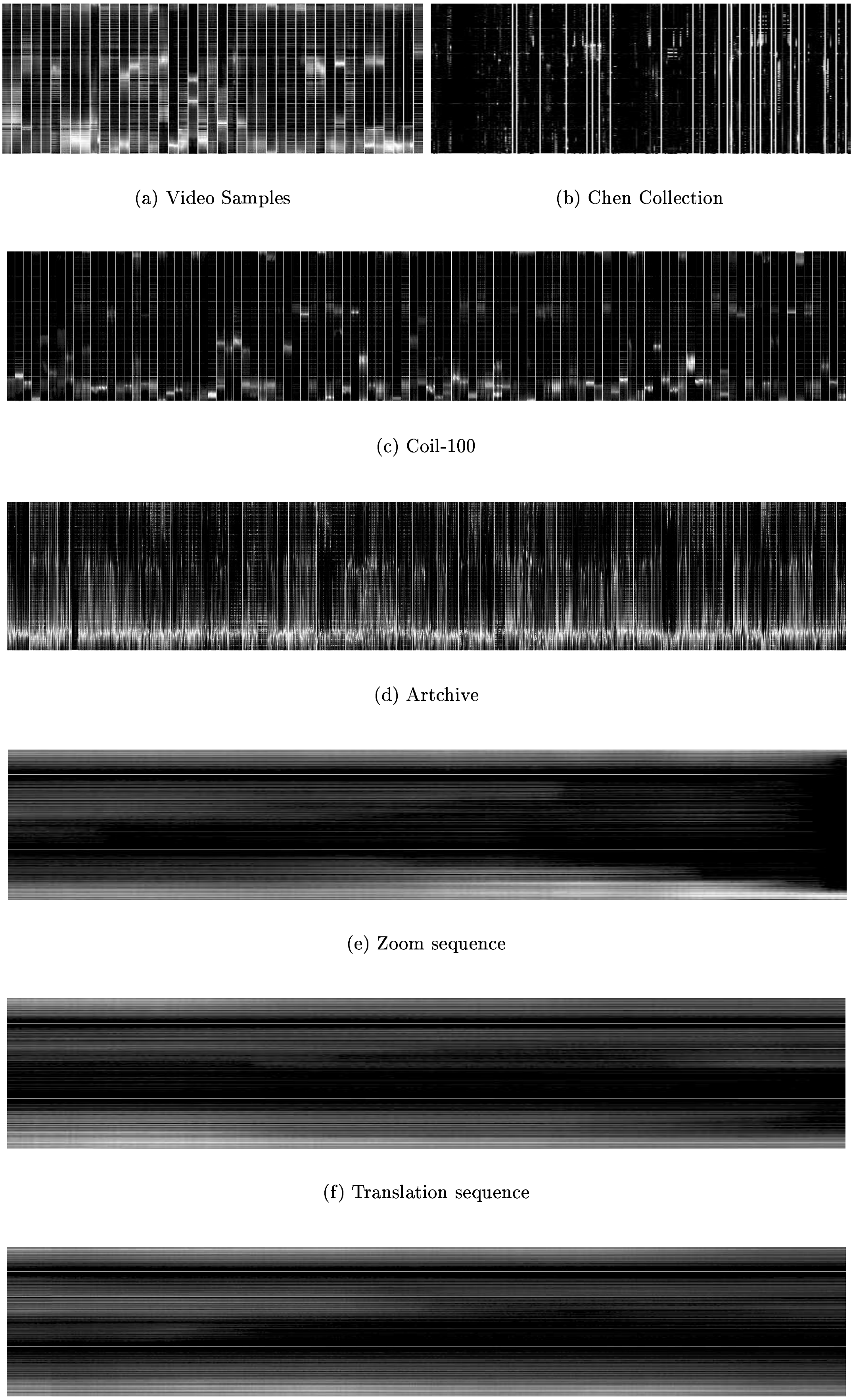
(a) $\sigma = 1.4$ (b) $\sigma = 2.0$ (c) $\sigma = 2.8$ (d) $\sigma = 4.0$ (e) $\sigma = 8.0$

Figure A.2: Edge detection example.

Appendix B

Results: Histogram Plots

In this section we have included the histogram plots referenced in Chapter 6. The images were enhanced for better printing quality and may be distorted. Histogram plots are more useful when printed in larger dimension of when studied on a computer screen.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

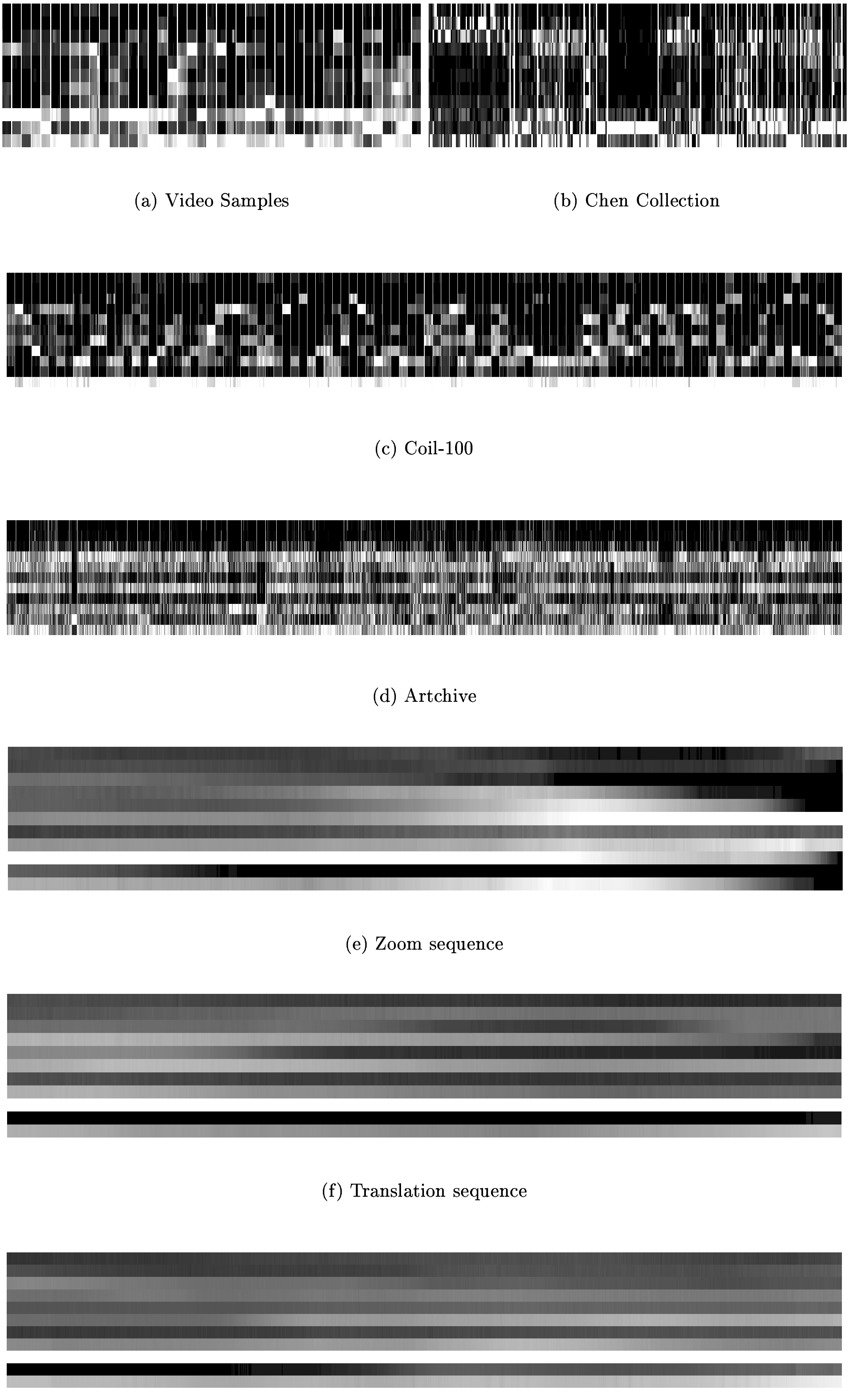
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.1: (HUE) Hue Histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

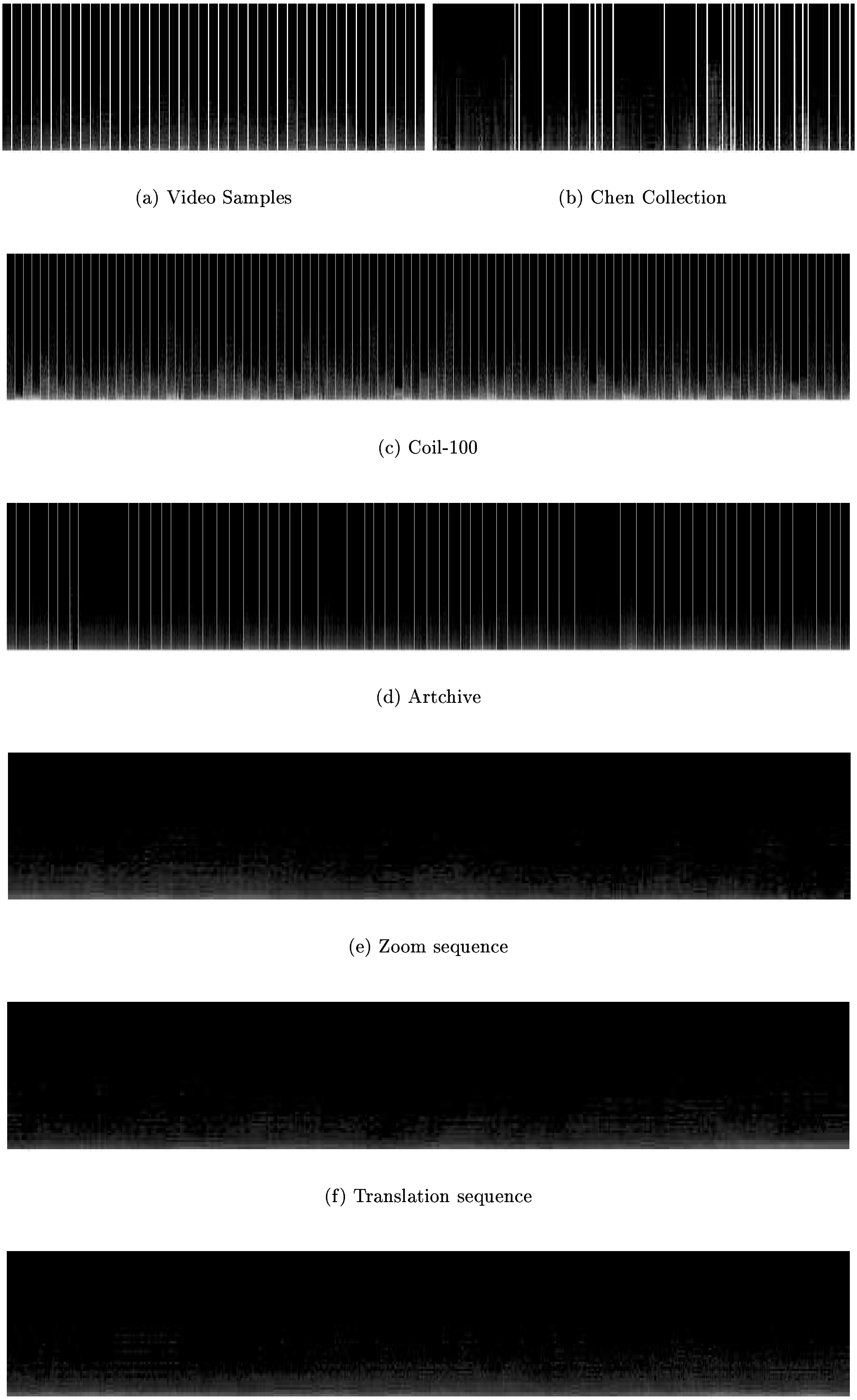
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.2: (CL) Berlin and Kay Colour Labels histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

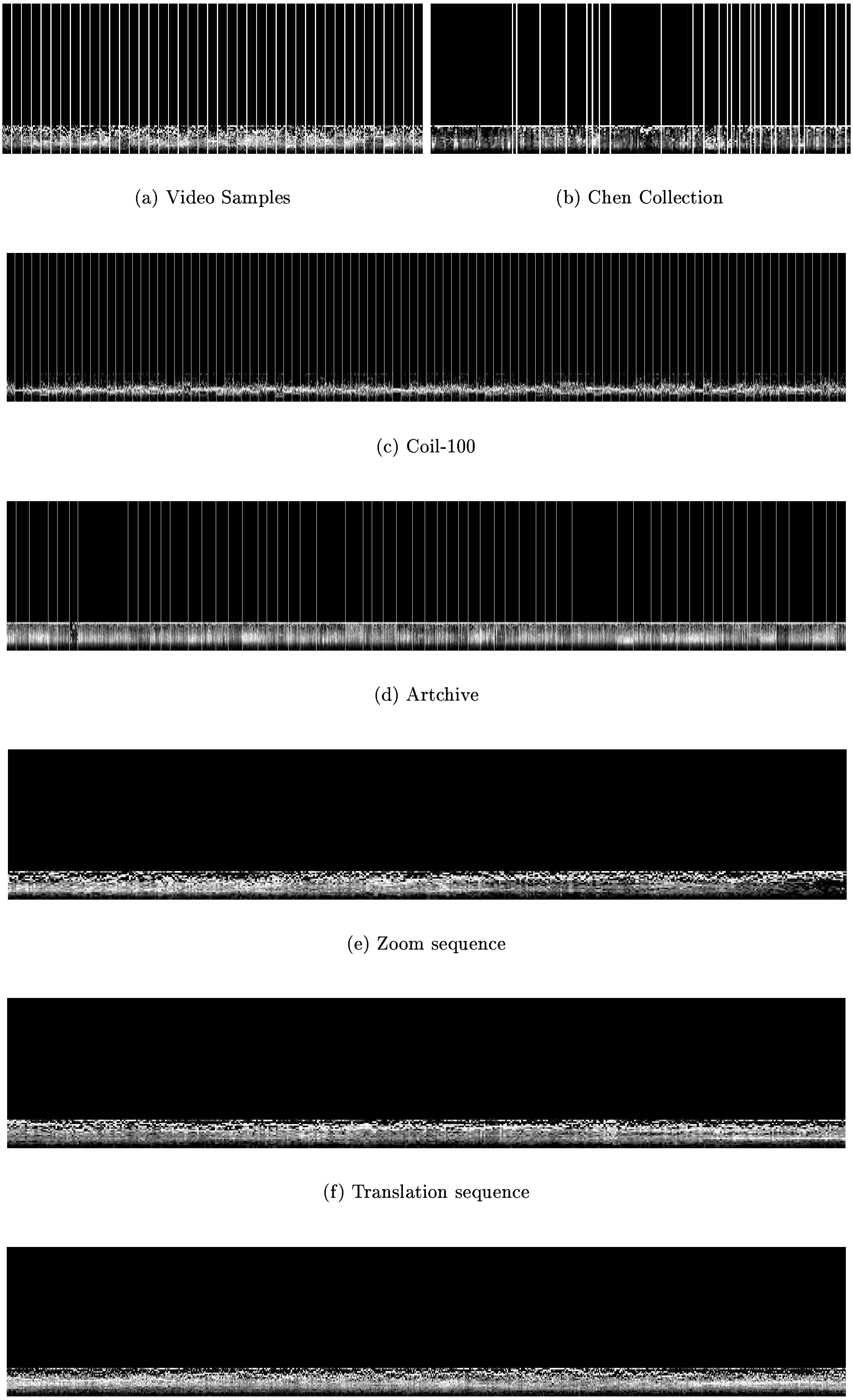
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.3: (KCOS) K-Cos edge curvature histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

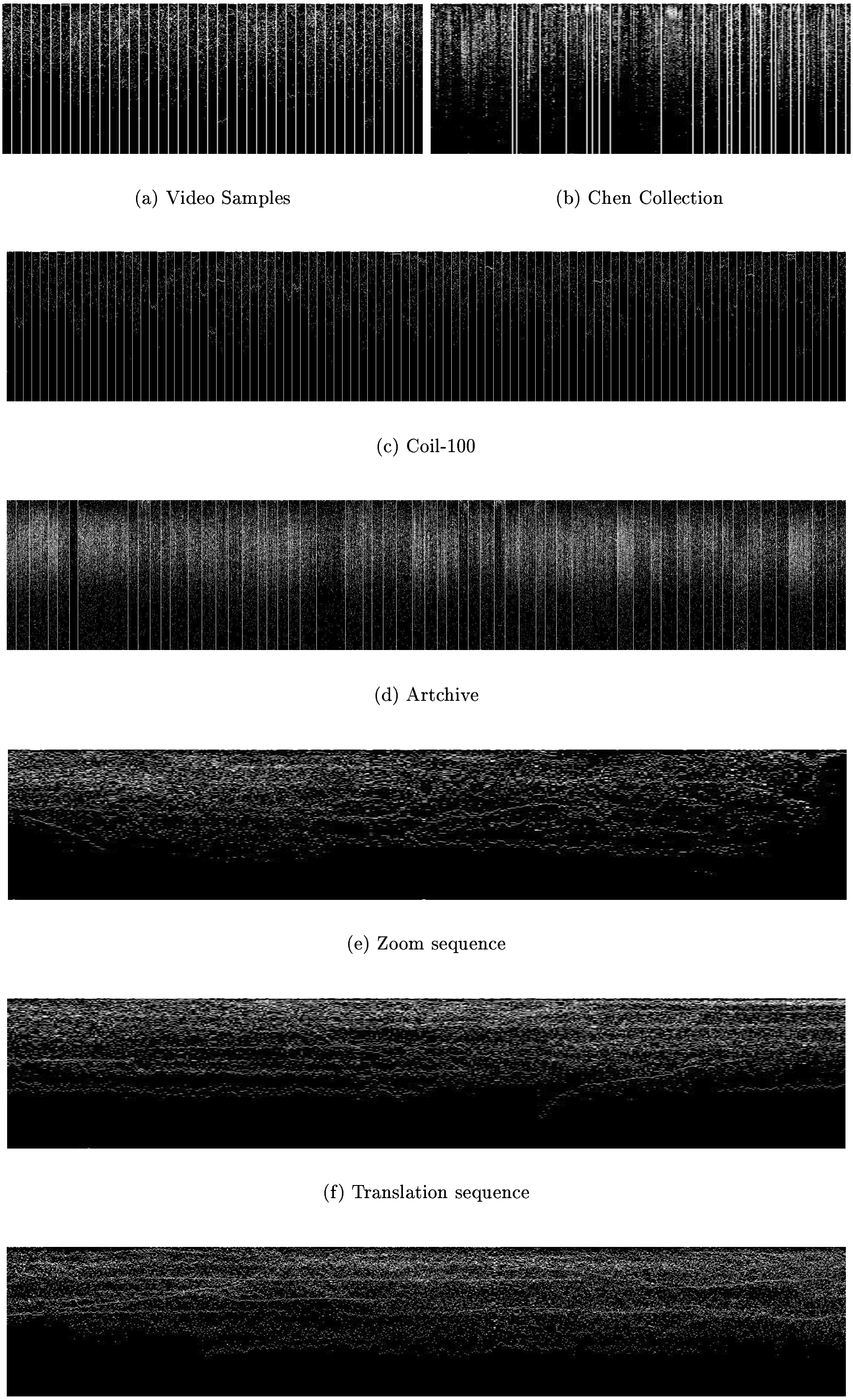
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.4: (CARC) Circular Arcs histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

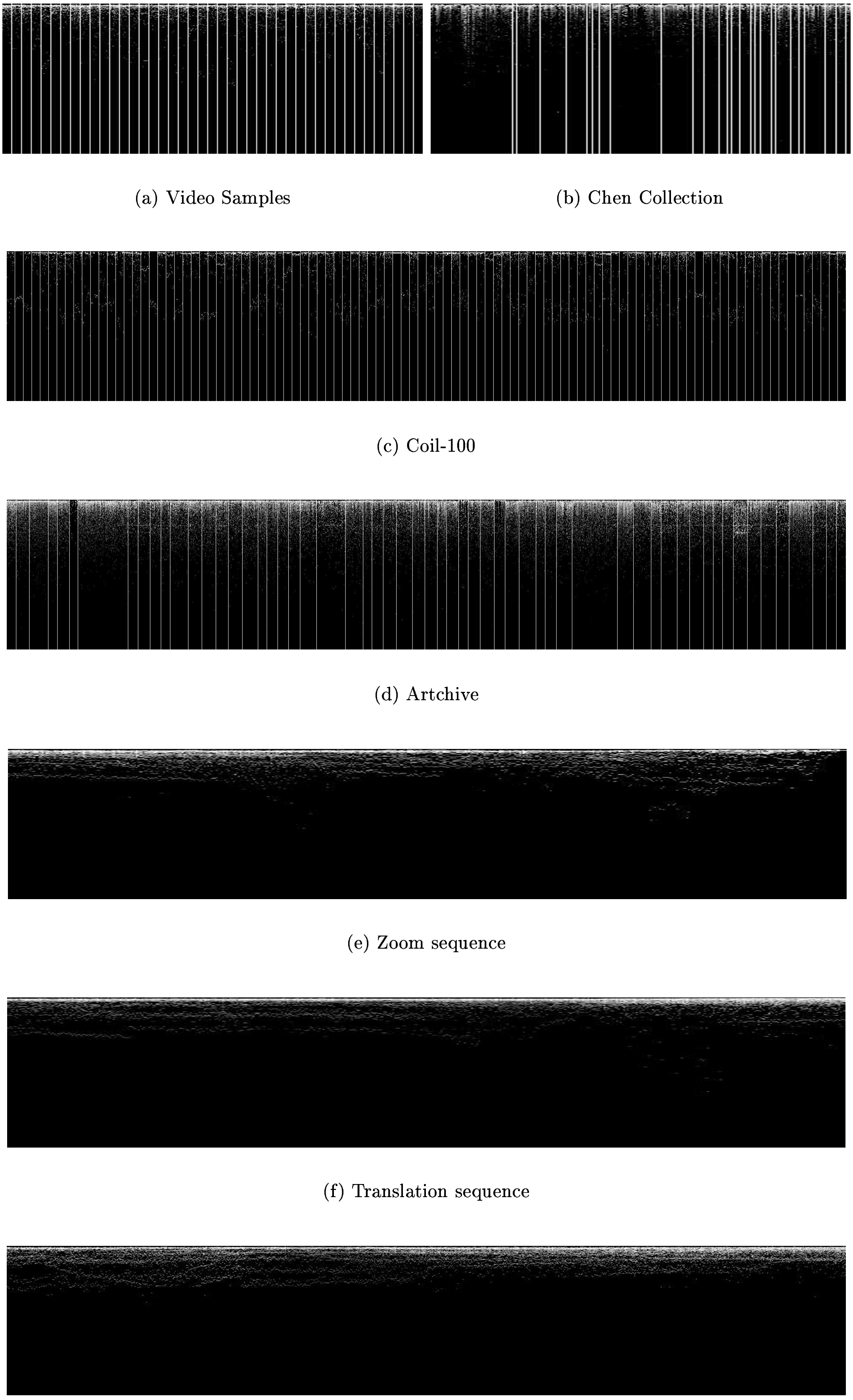
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.5: (CONV) Convexity histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

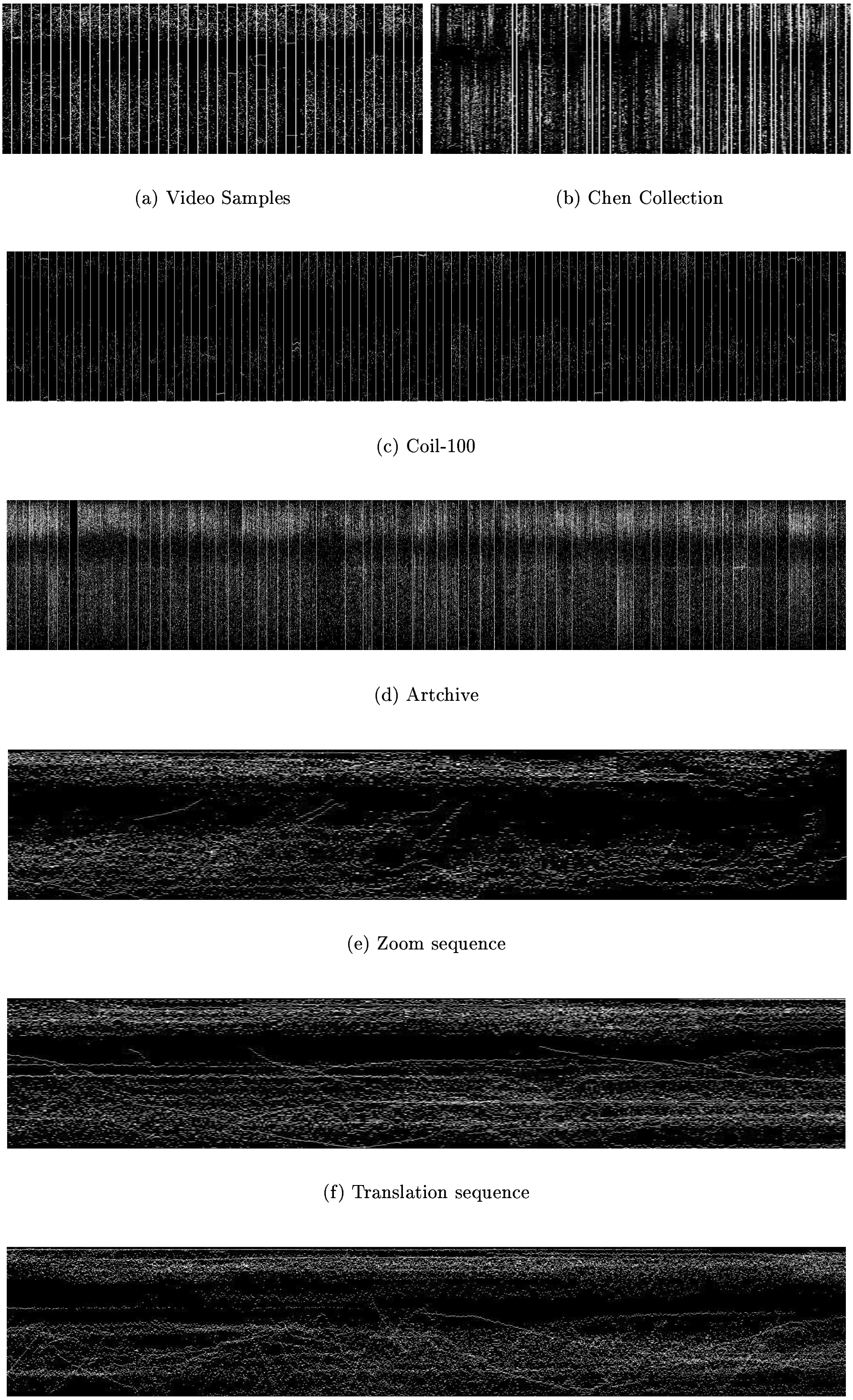
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.6: (CIRC) Circularity histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

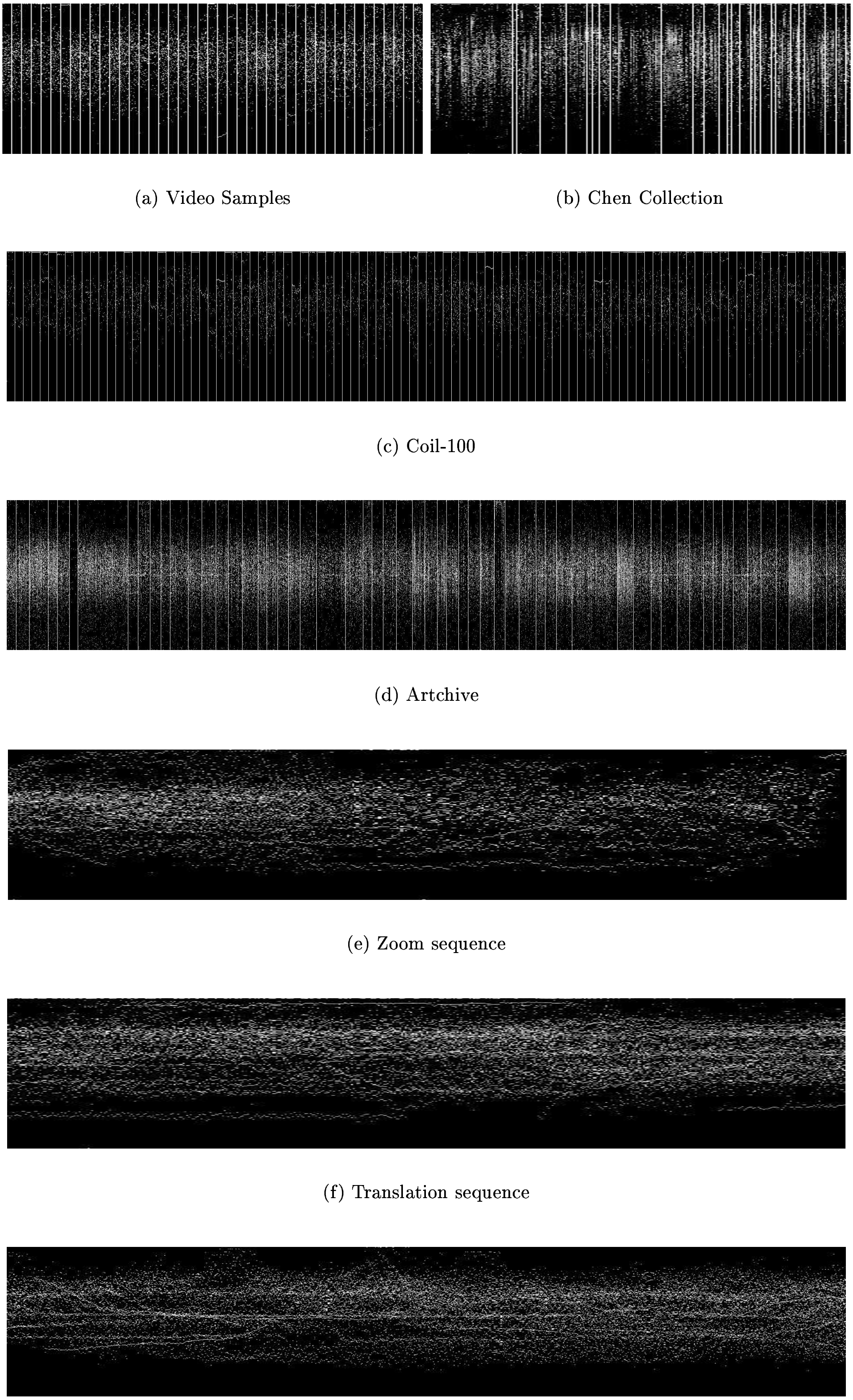
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.7: (ELIP) Elipticity histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

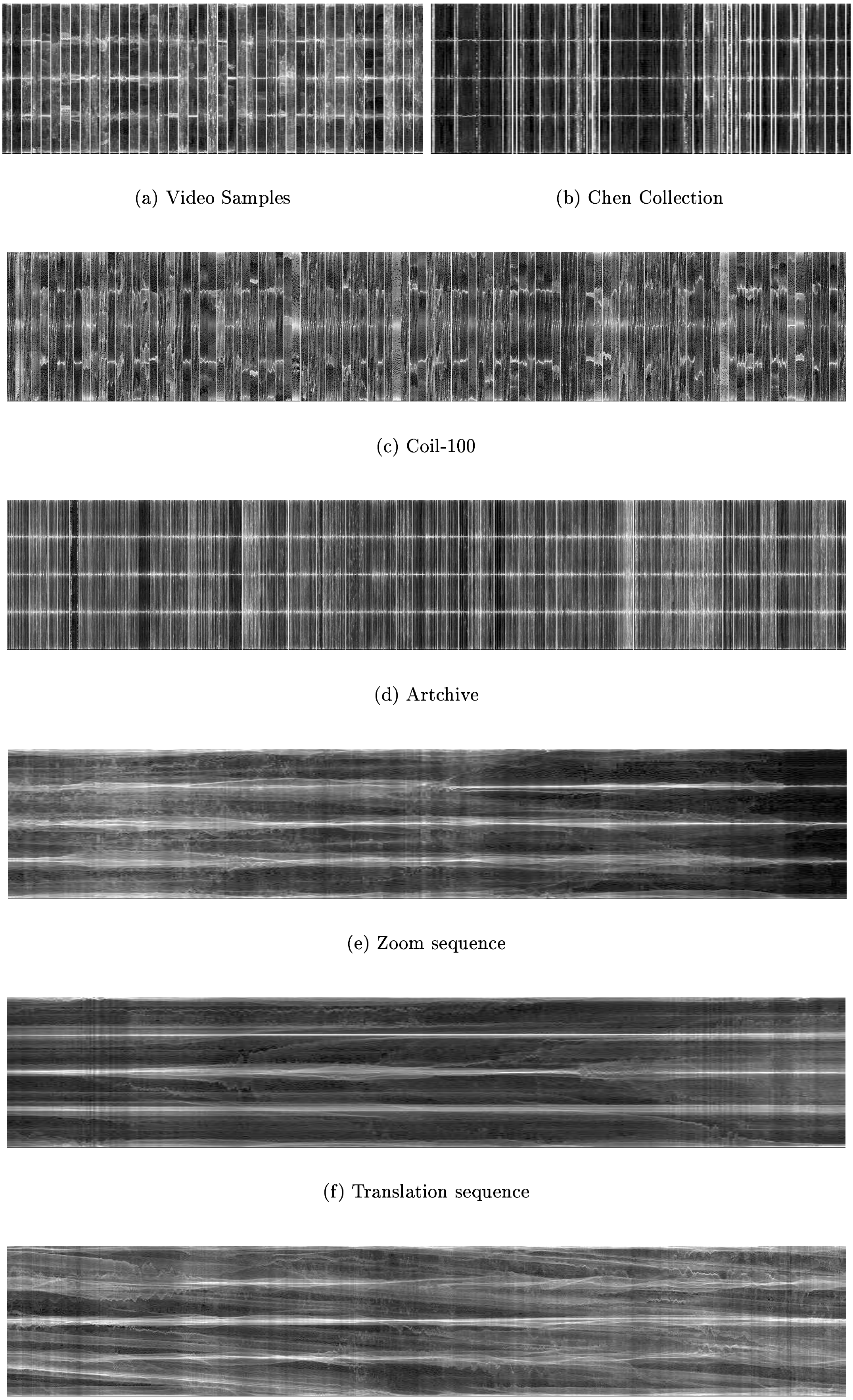
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.8: (RECT) Rectangularity histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

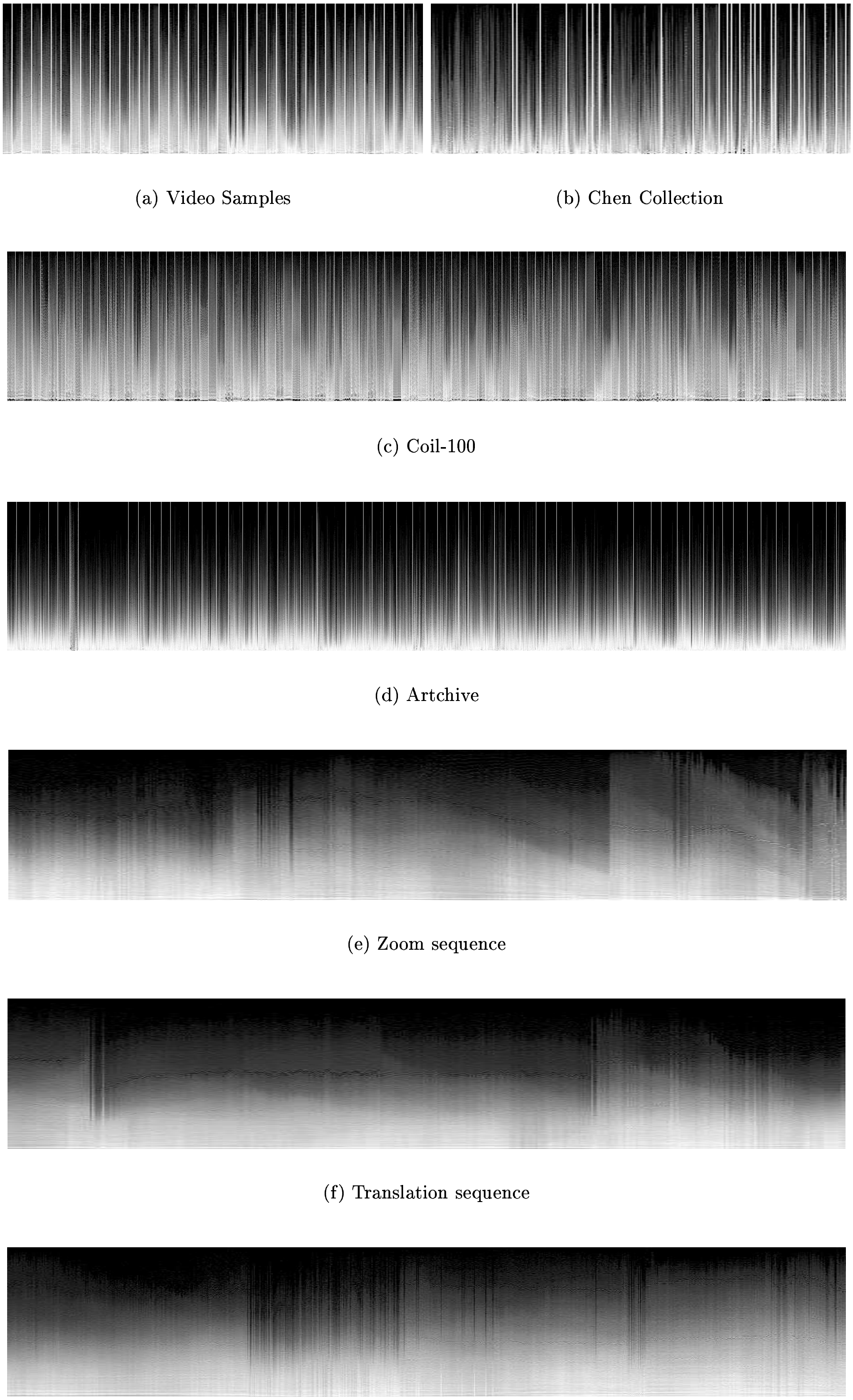
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.9: (ORNT) Orientation histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

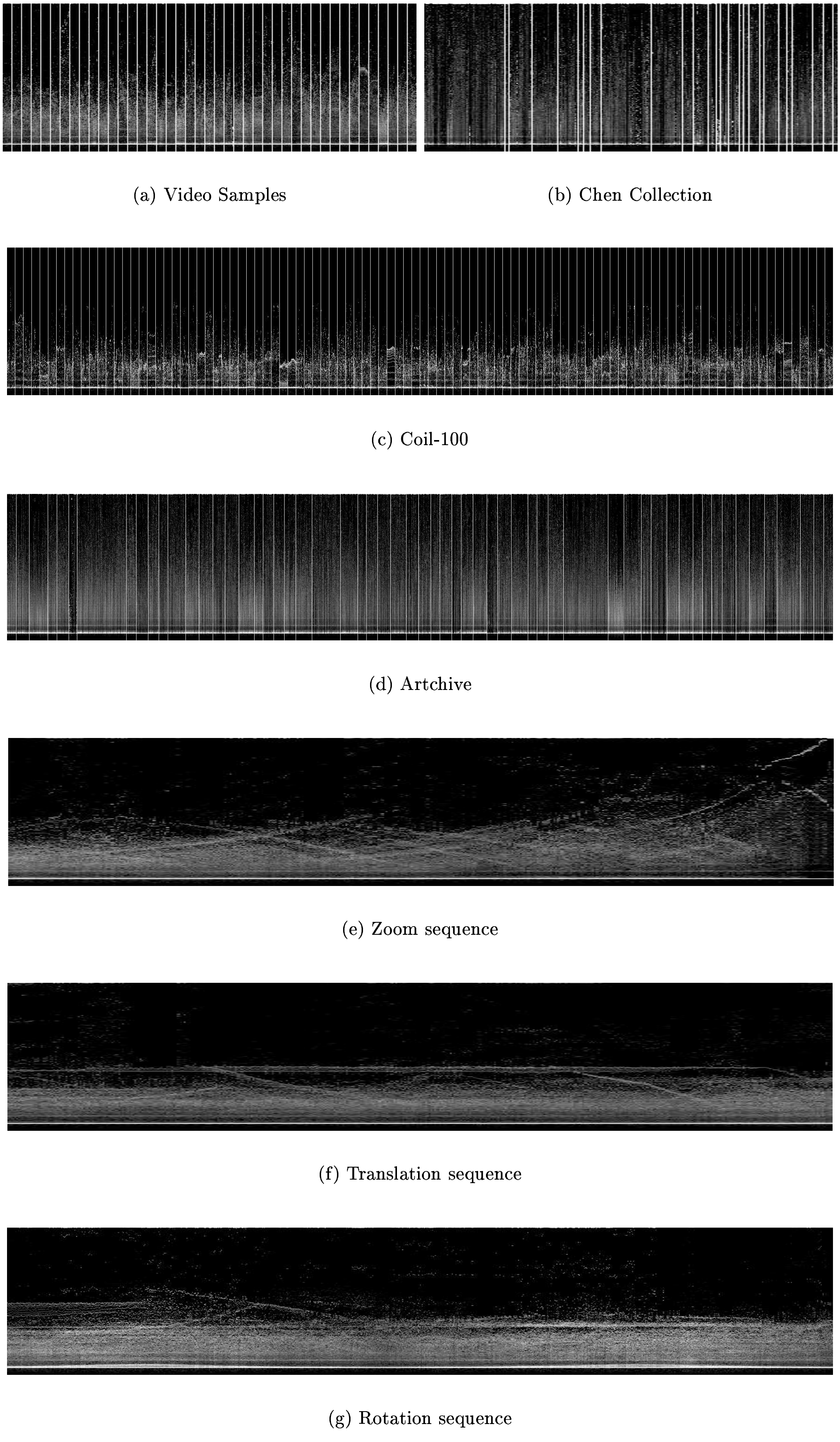
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.10: (MSSDT) Multiscale Salience Distance histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

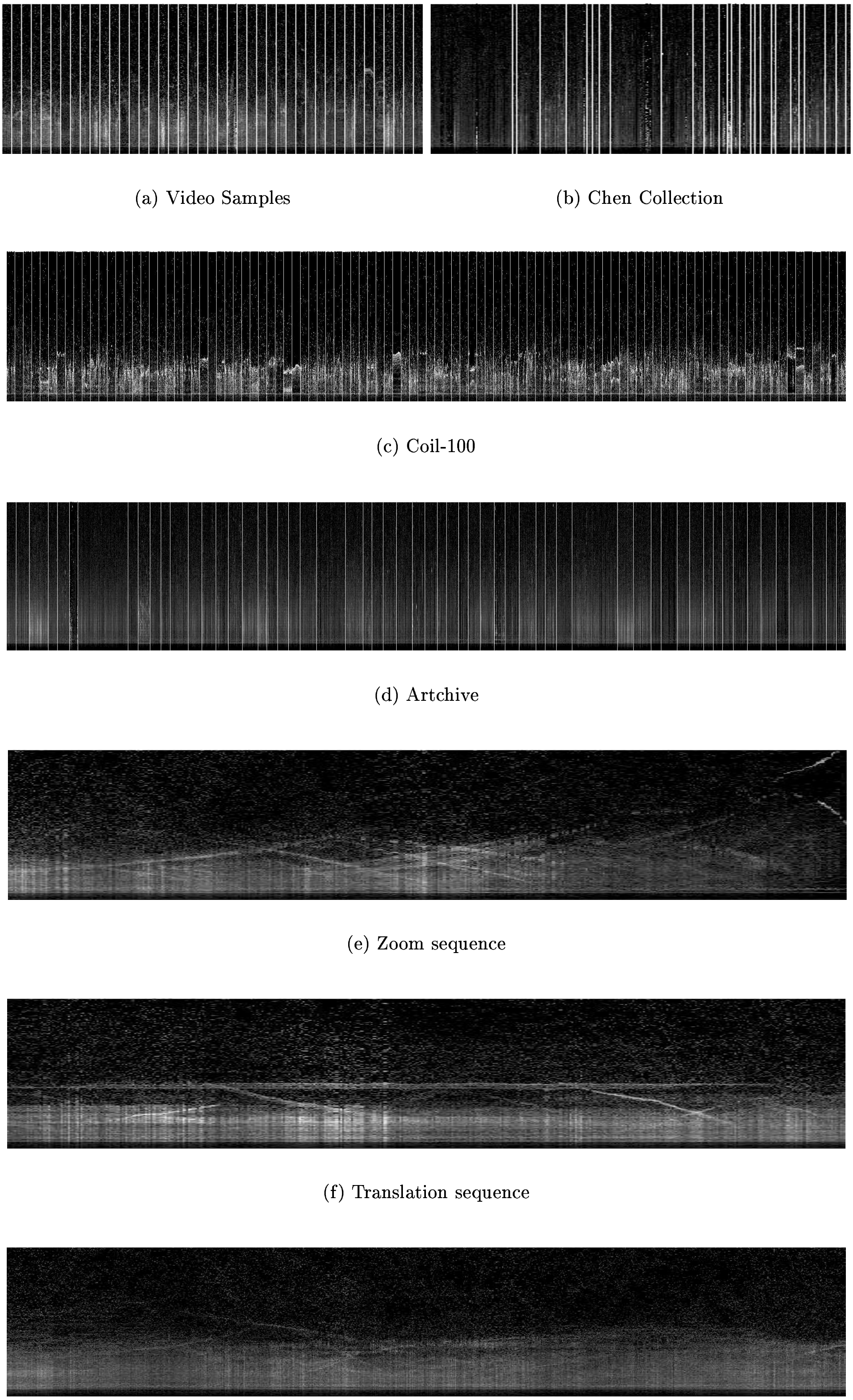
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.11: (TARE) Triangle Area histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

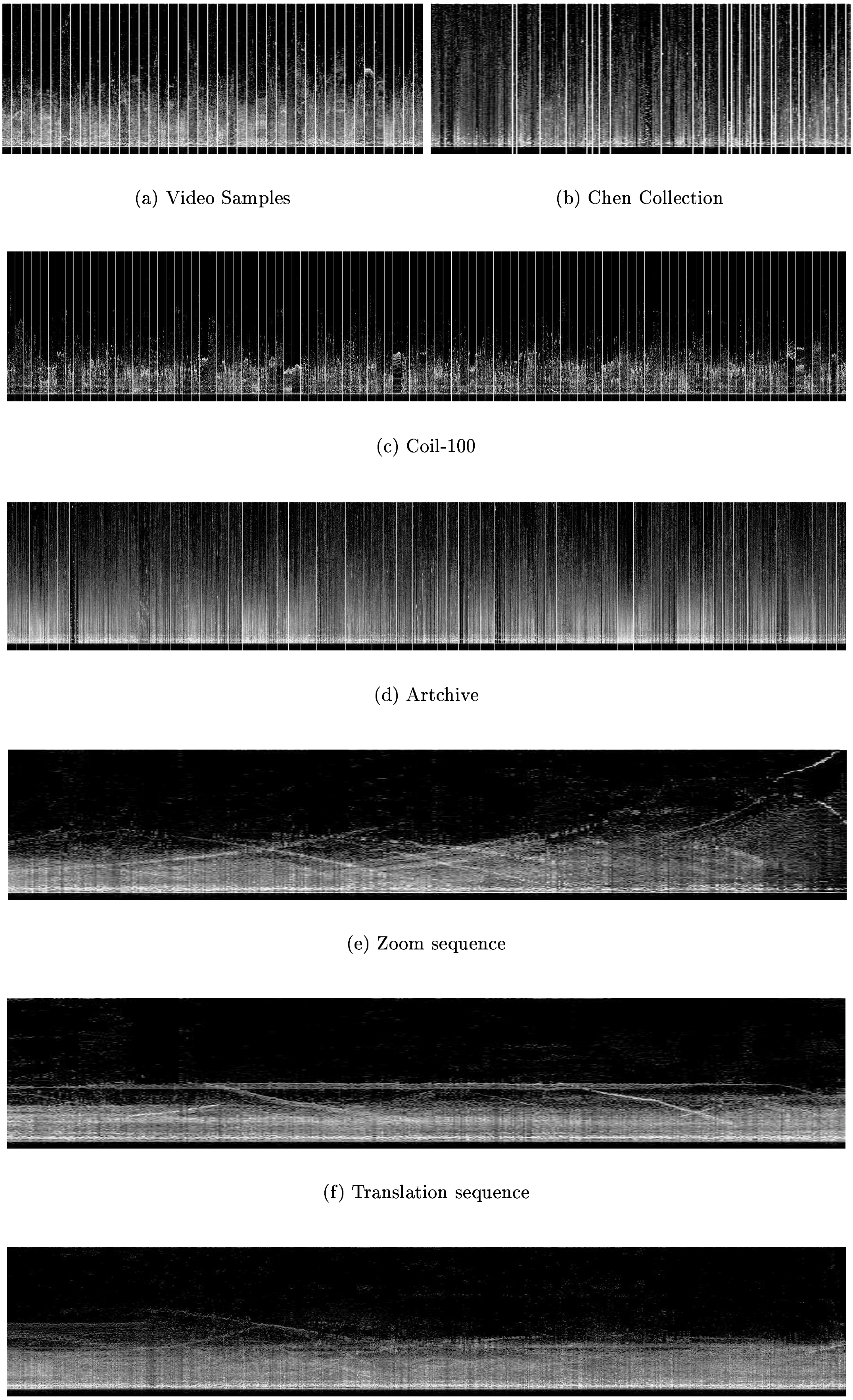
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.12: (TLEN) Triangle Length histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

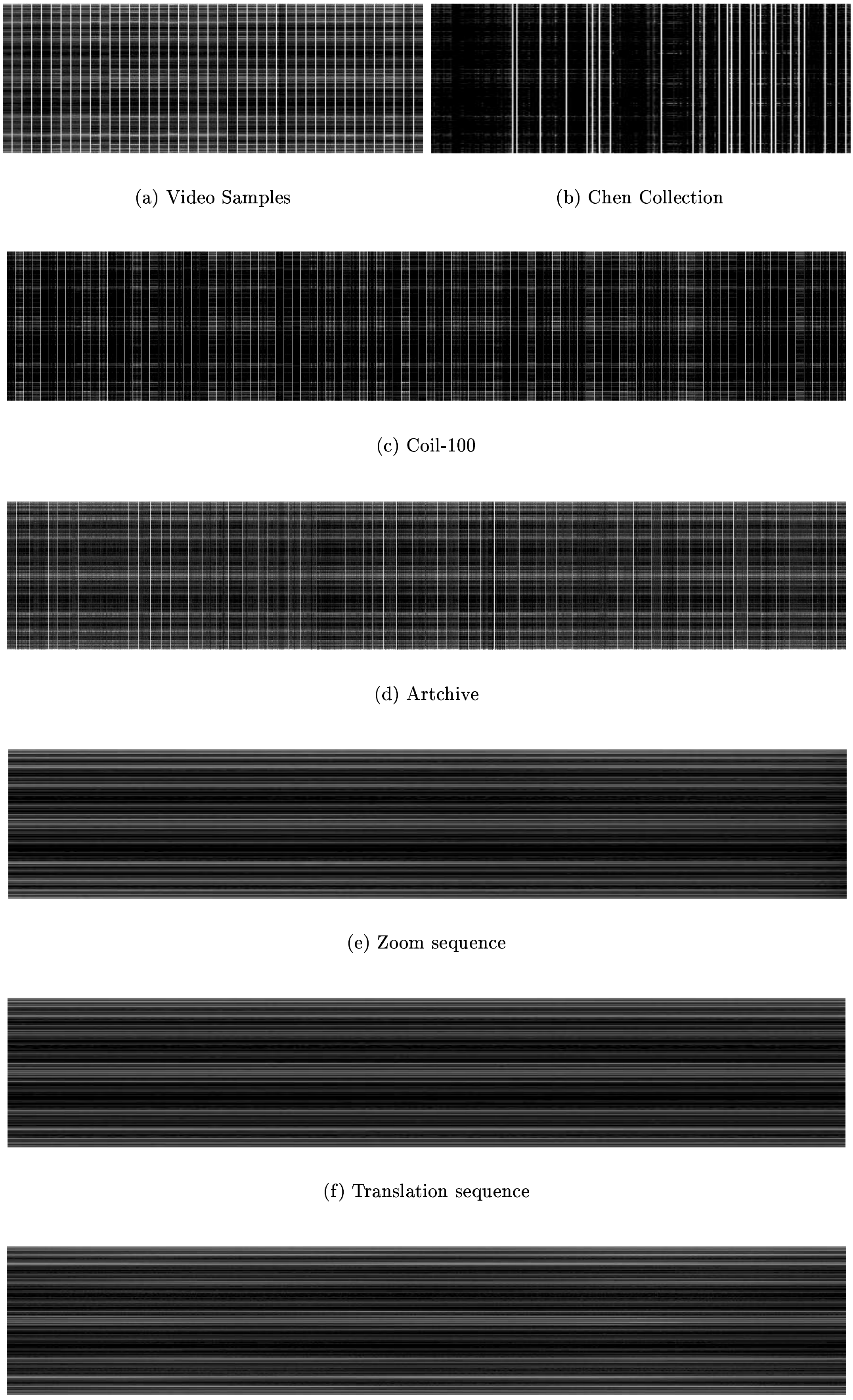
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.13: (TRAT) Triangle Ratio histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

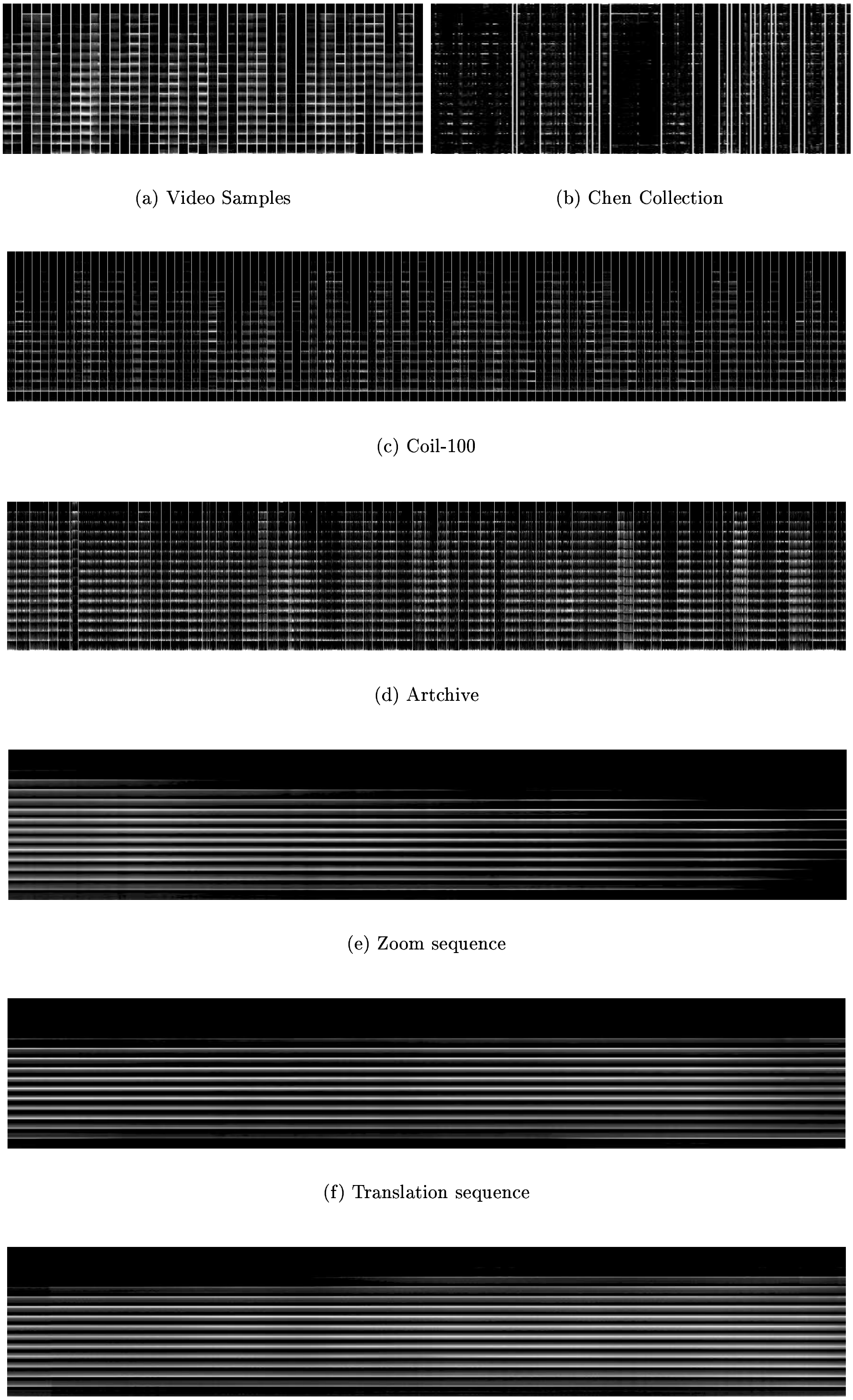
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.14: (TU) Texture Unit histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

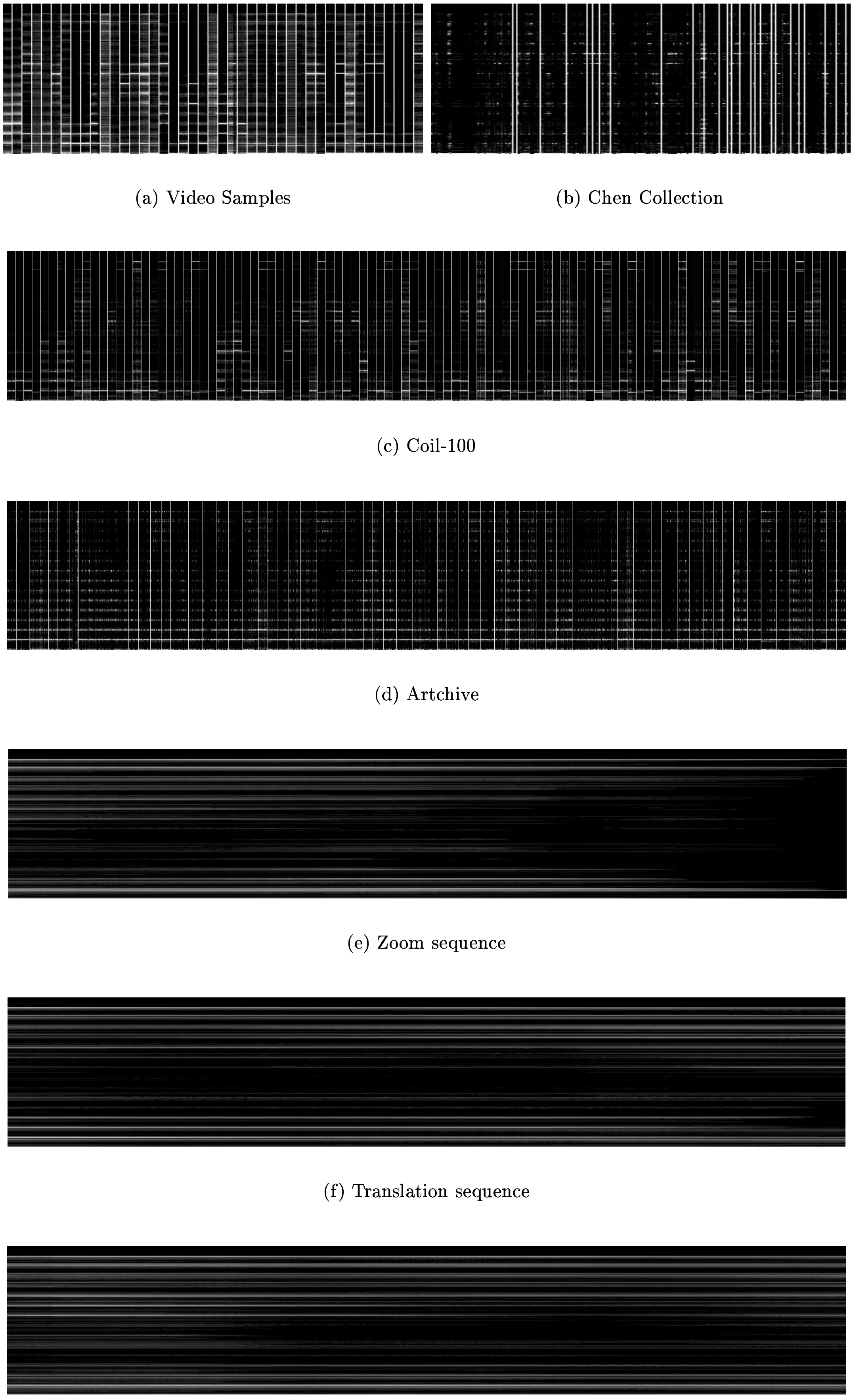
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.15: (COOC) Cooccurrence Matrix (intensity) plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

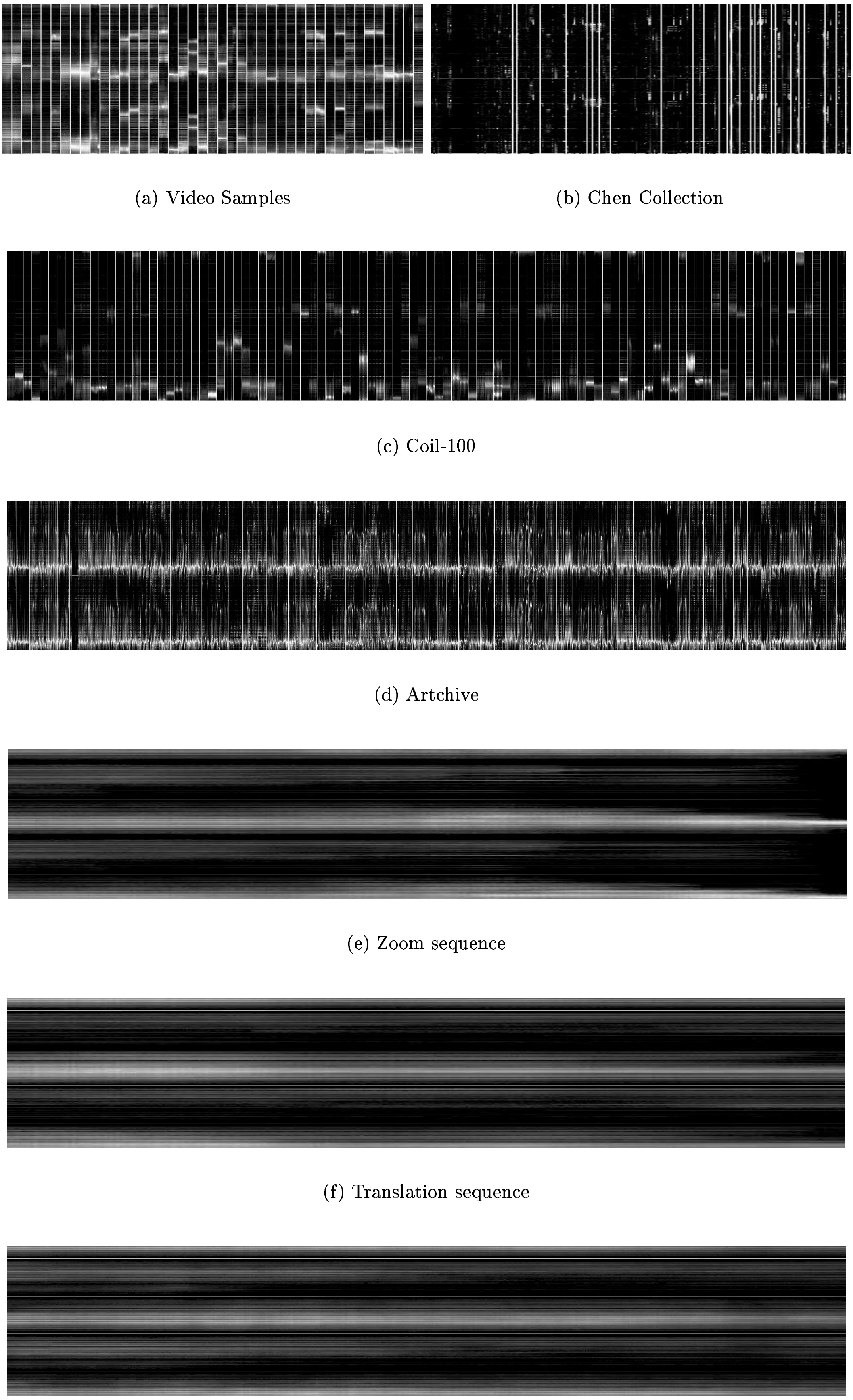
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.16: (HCC) Cooccurrence matrix (hue) plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

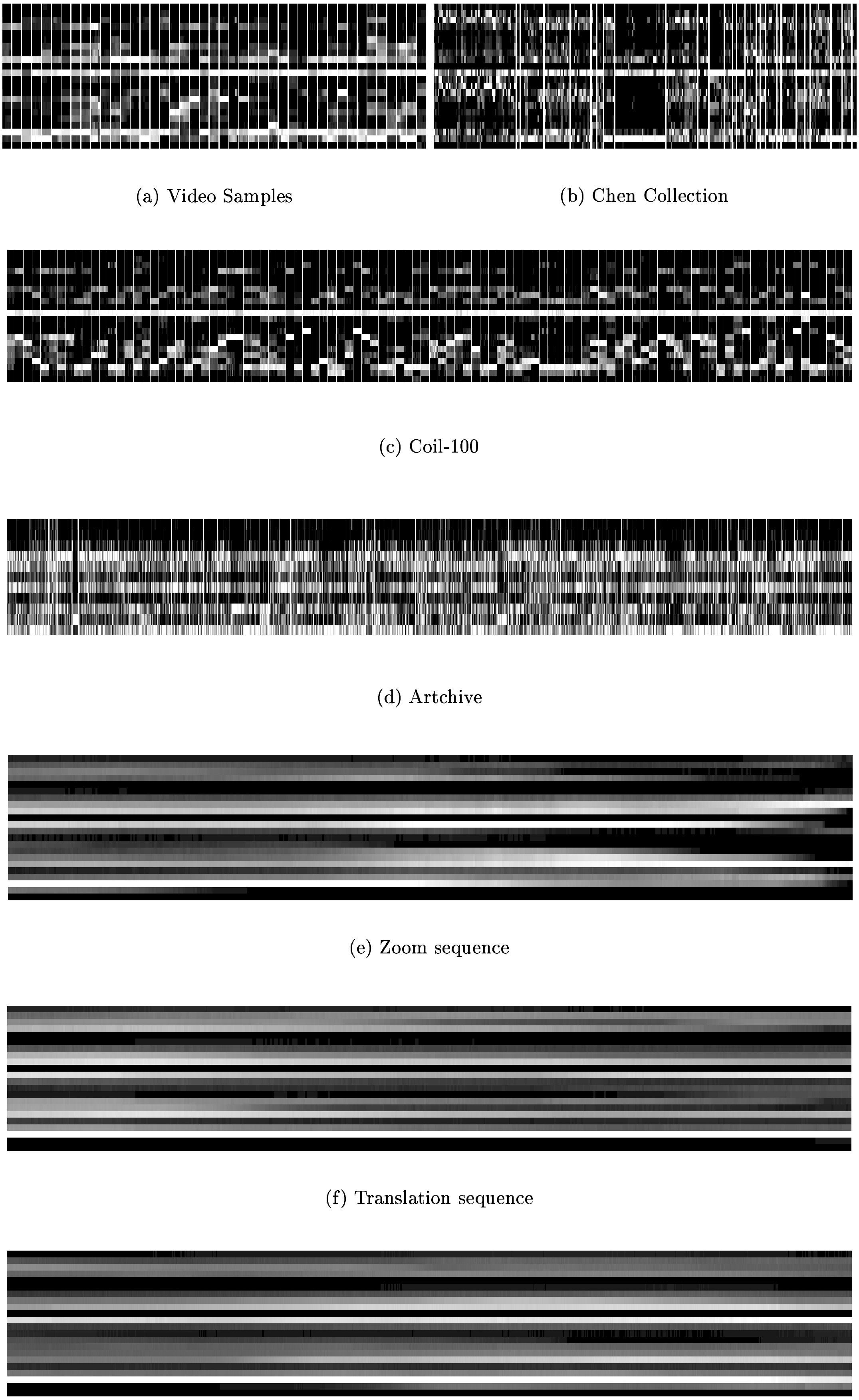
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.17: (HUE-SP) Spatial Hue histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

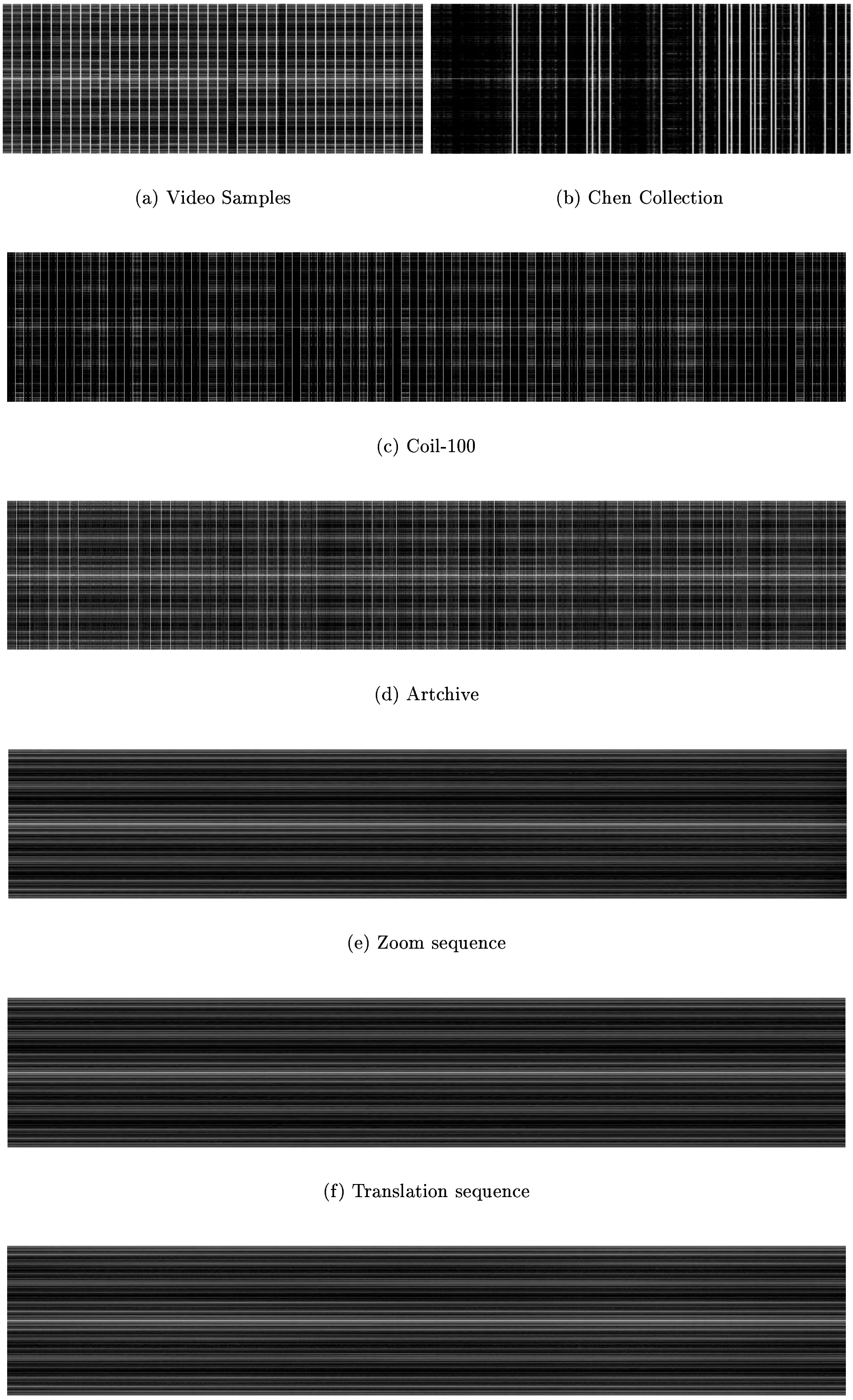
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.18: (CL-SP) Spatial Berlin and Kay Colour Labels histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

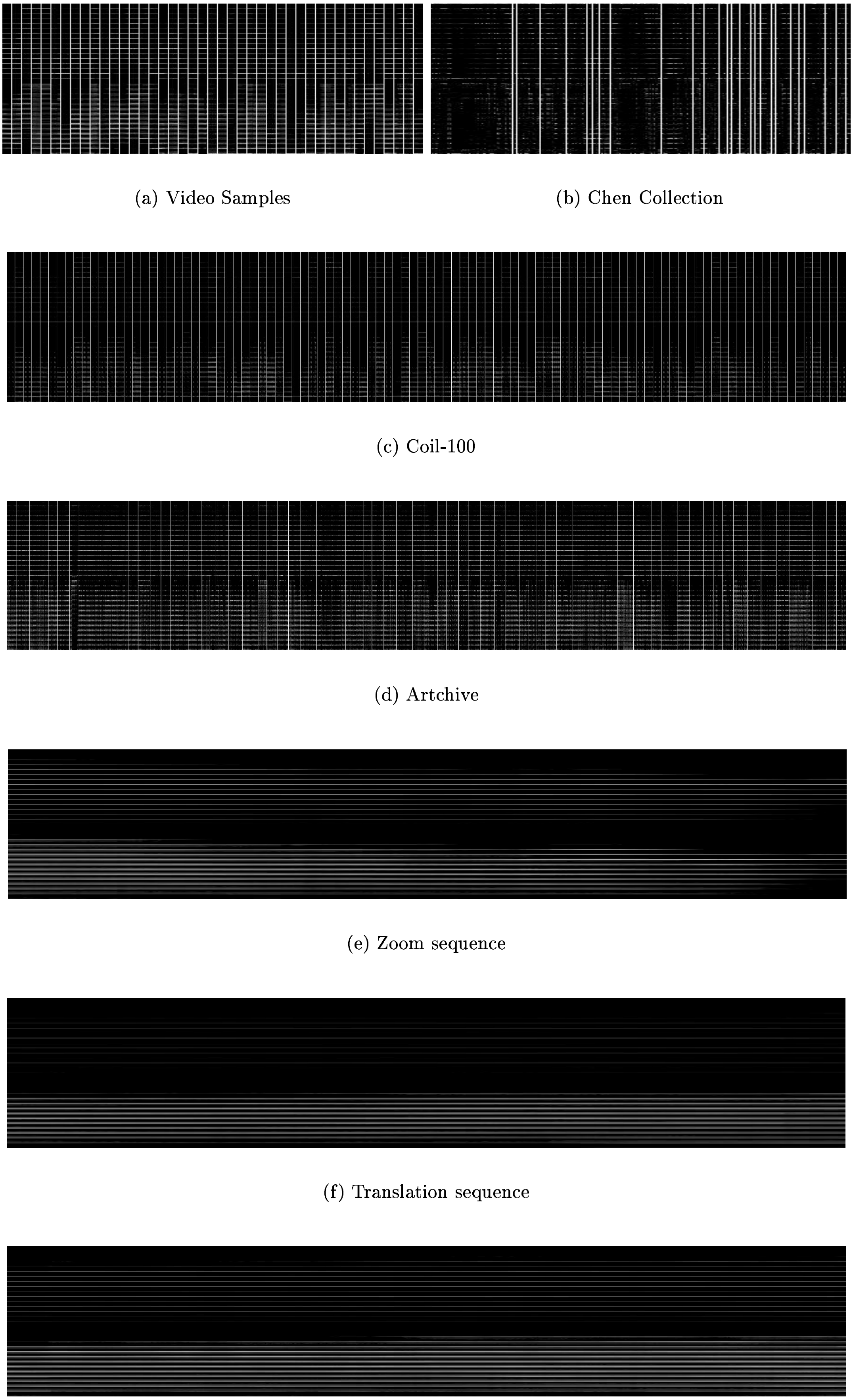
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.19: (TU-SP) Spatial Texture Unit histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

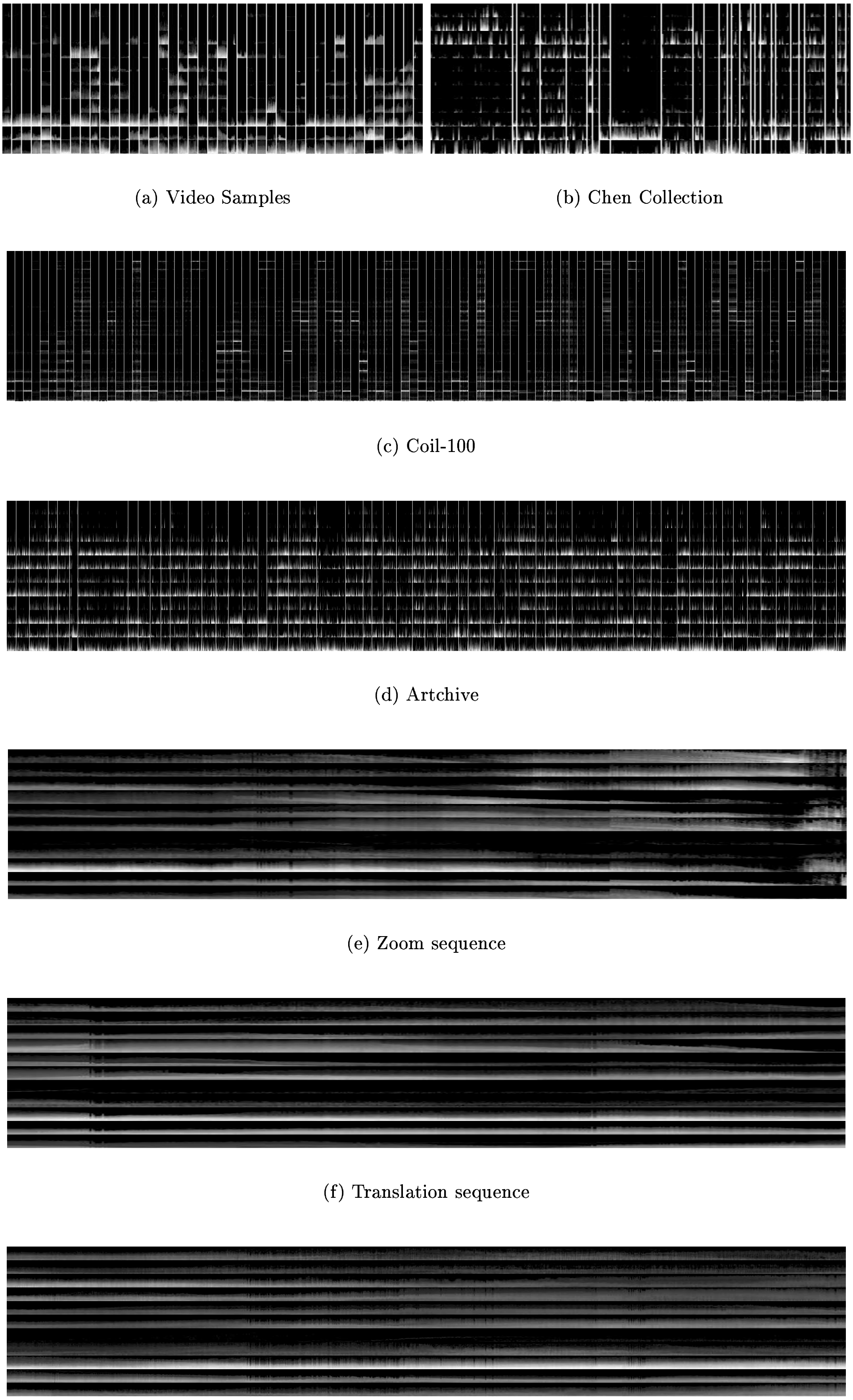
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.20: (CC-SP) Spatial Cooccurrence Matrix (intensity) plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

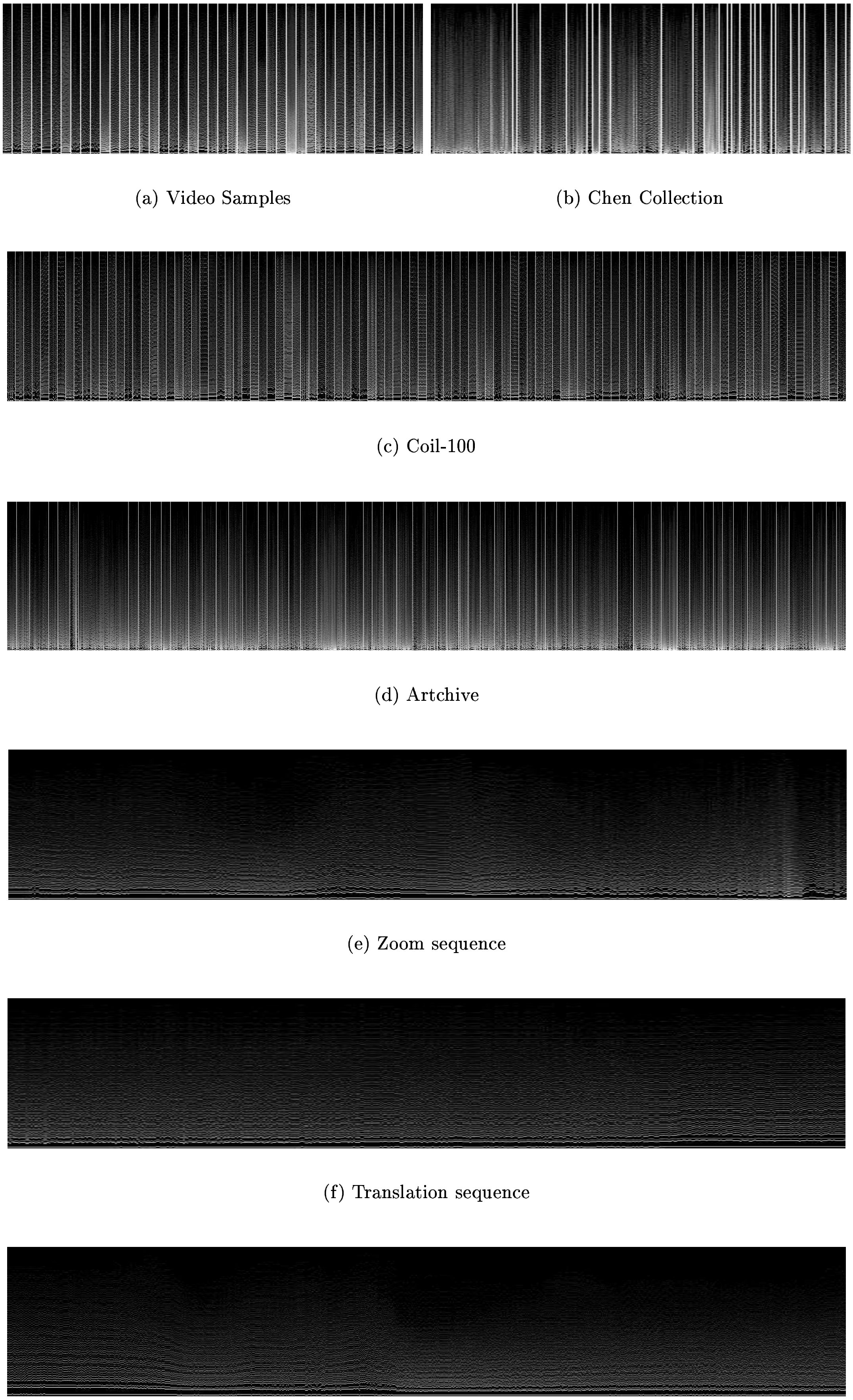
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.21: (CLSH) Distance from Colour Region Boundaries histogram plots.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

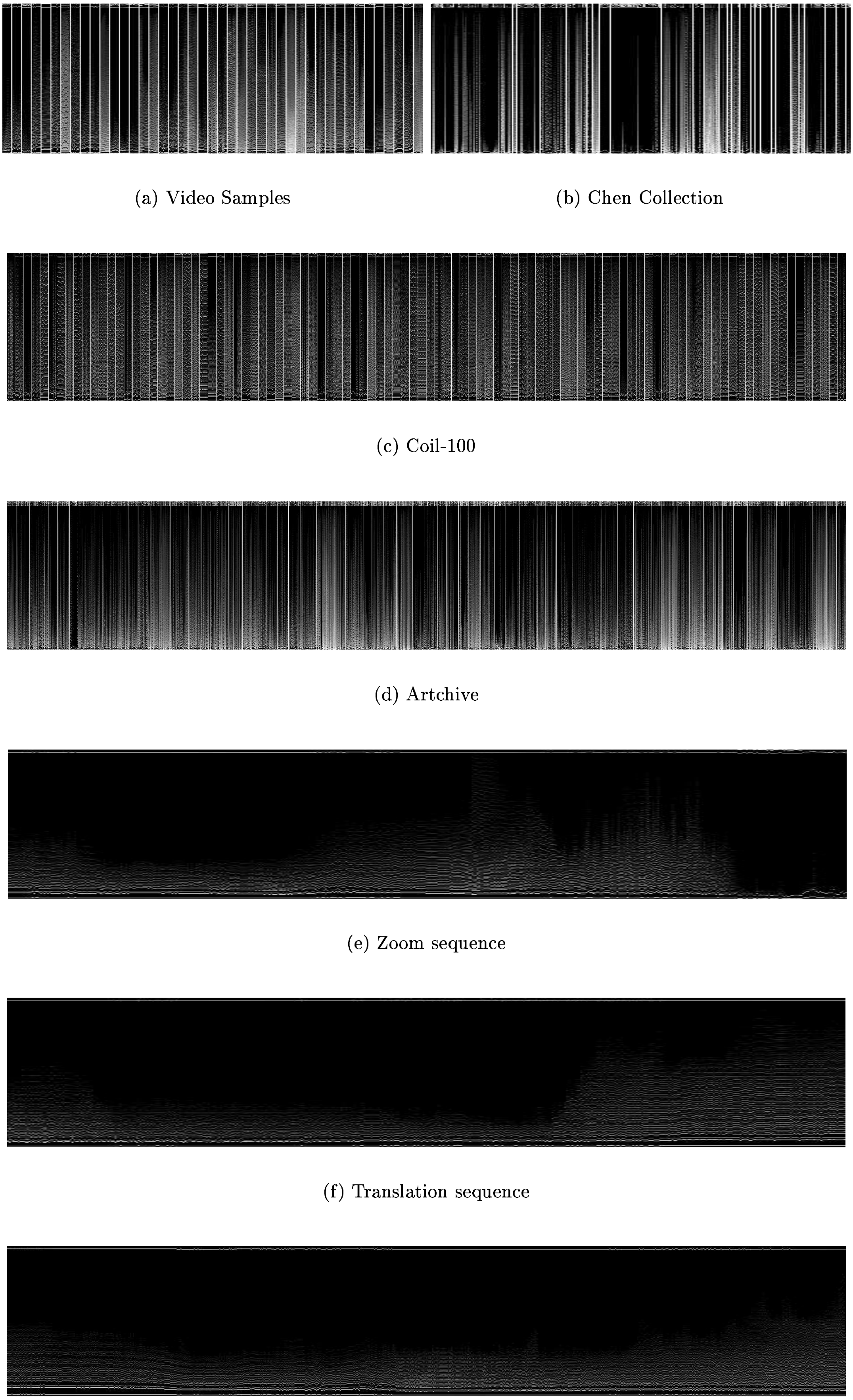
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.22: (BKDT) Colour Labels and Colour Region Boundary distance histogram plot.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

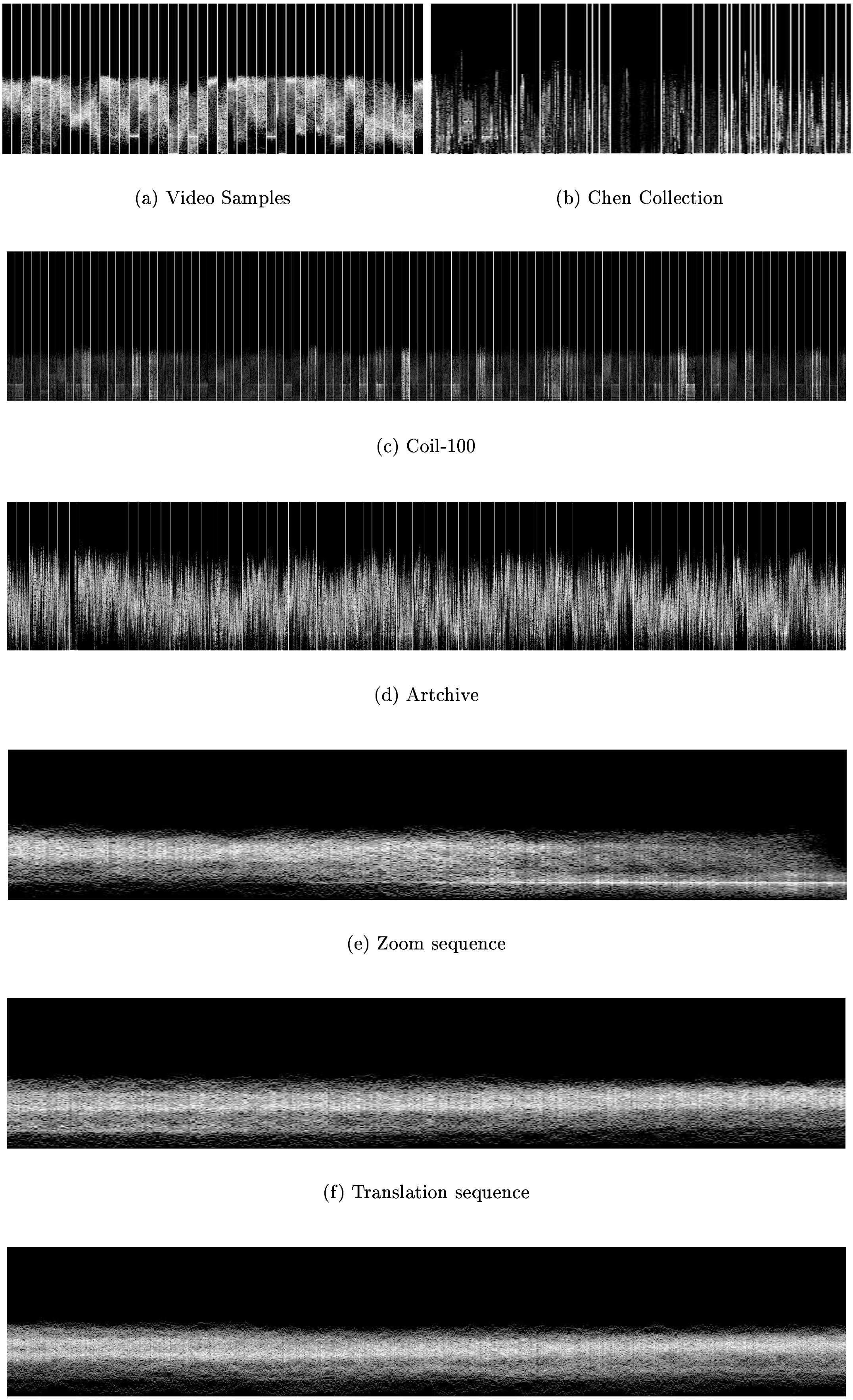
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.23: (BKDT2D) Colour labels and Edge Distance Histogram plot.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

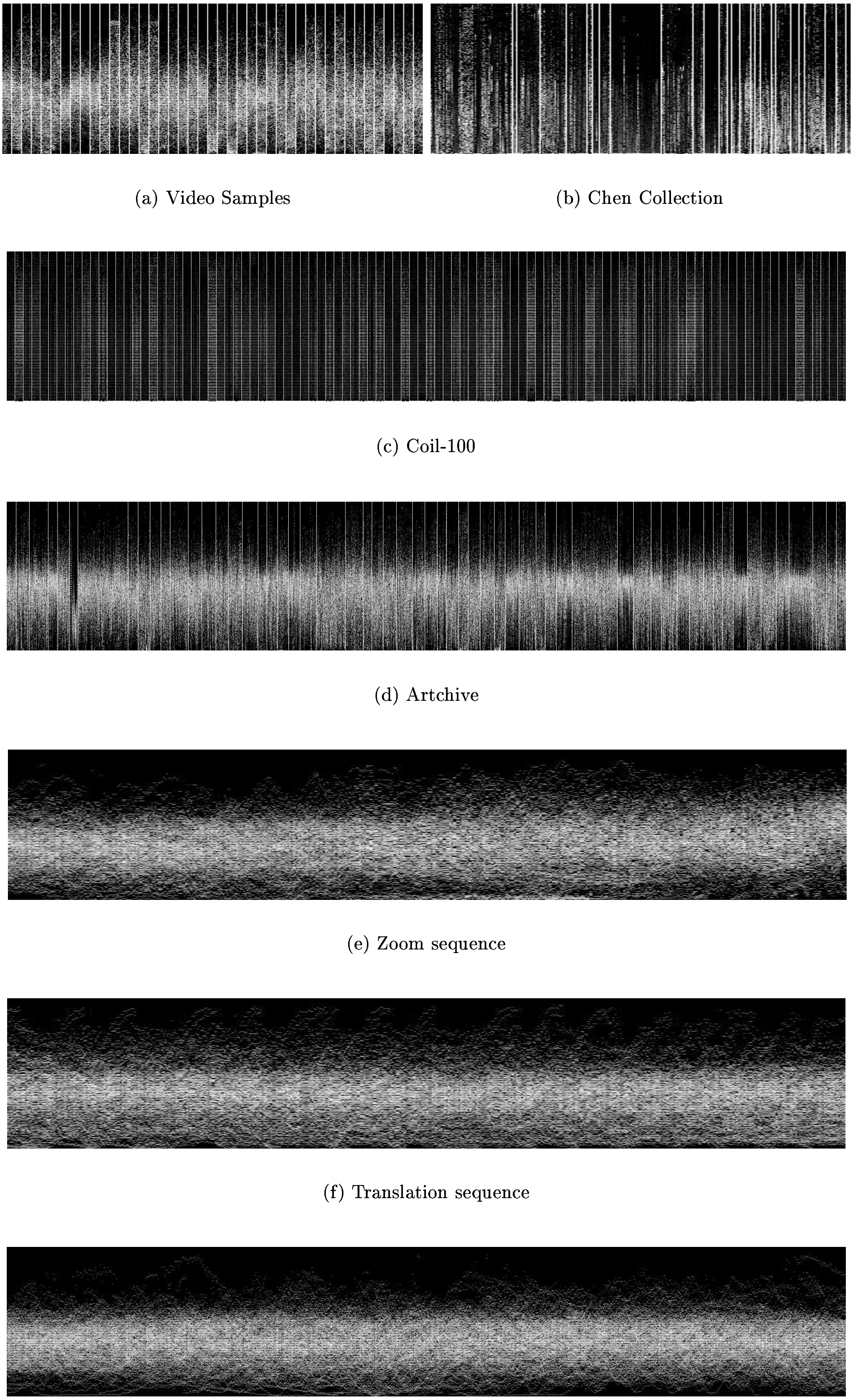
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.24: (HENT) Hue Entropy histogram plot.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

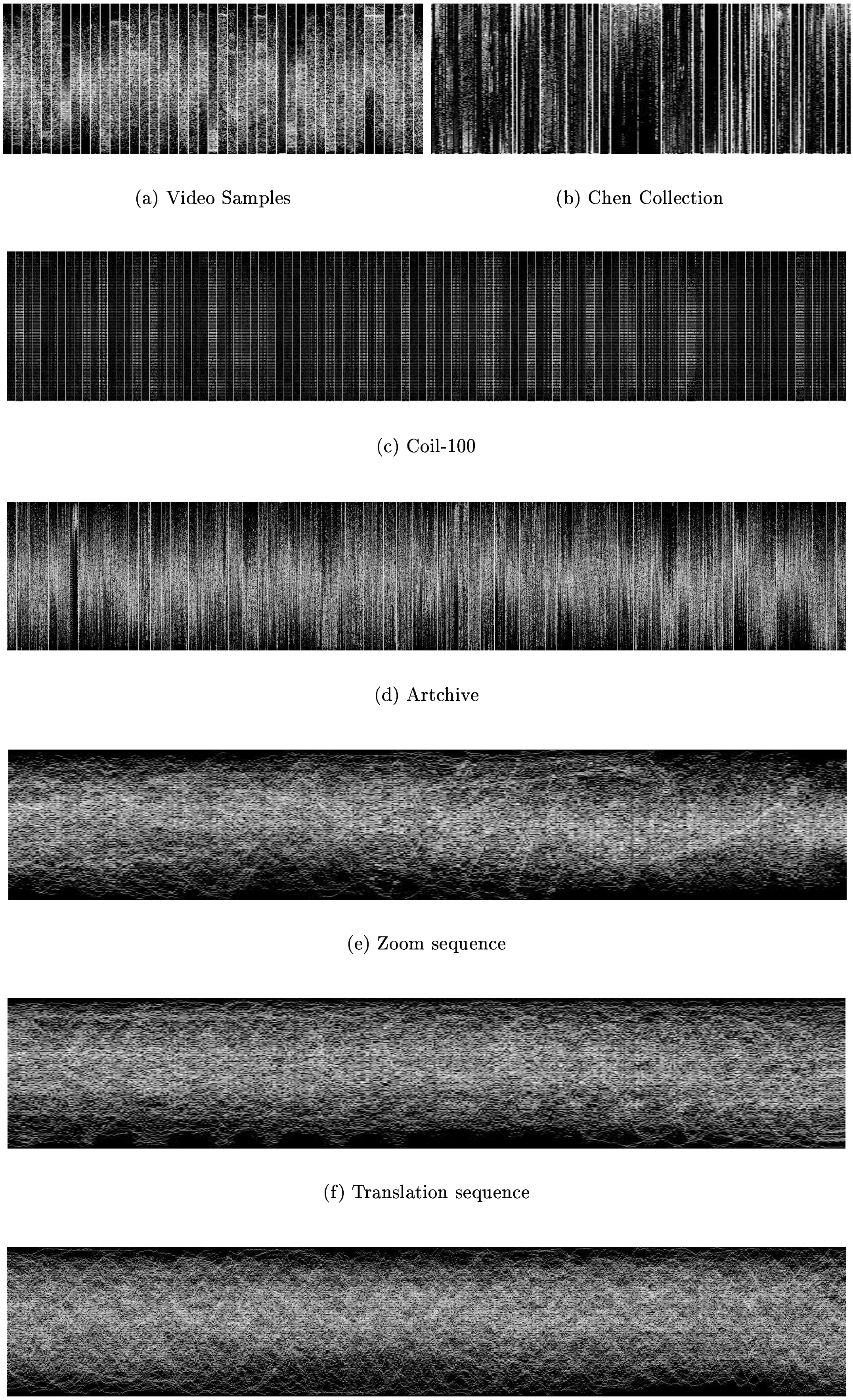
(d) Artchive

(e) Zoom sequence

(f) Translation sequence

(g) Rotation sequence

Figure B.25: (LDIFF) Local Difference histogram plot.



(a) Video Samples

(b) Chen Collection

(c) Coil-100

(d) Artchive

(e) Zoom sequence

(f) Translation sequence

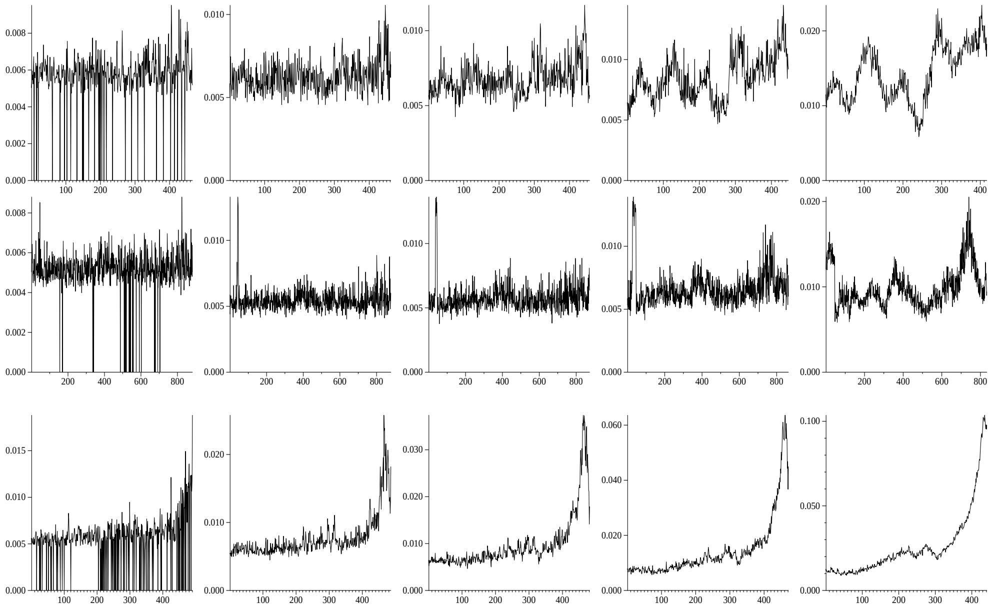
(g) Rotation sequence

Figure B.26: (LBLK) Local Black histogram plot.

Appendix C

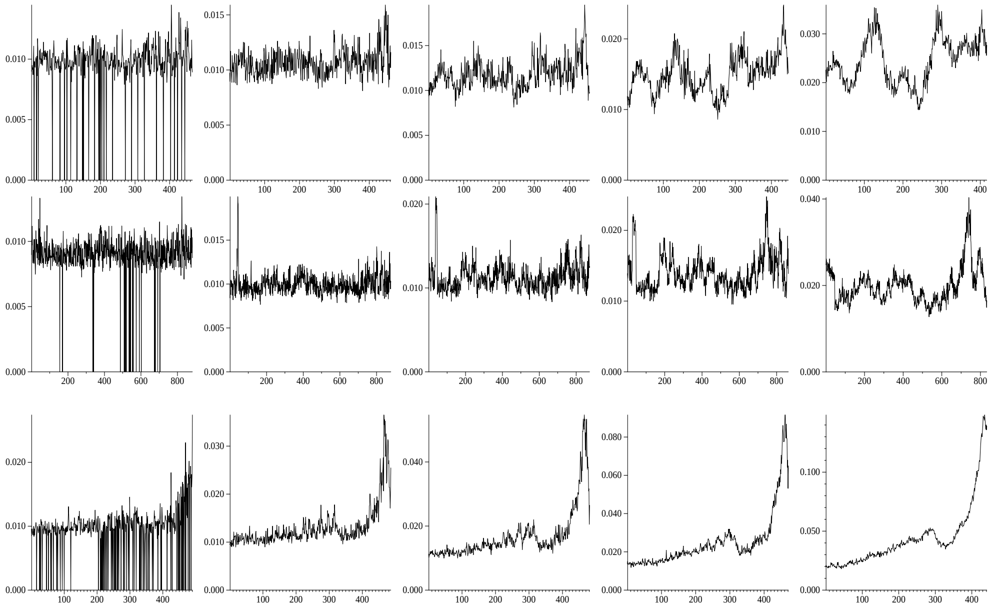
Results: Similarity Graphs

In this section we have included the similarity graphs studied in Chapter 6. The similarity graphs are generated from video sequences of transformations, from top to bottom, *translation*, *rotation*, and *scale*. We would like to note that at this instance we have used the Euclidean distance for the calculations. Furthermore, the graphs are not plotted on the same axis, they are re-scaled to enhance the range of distance values.



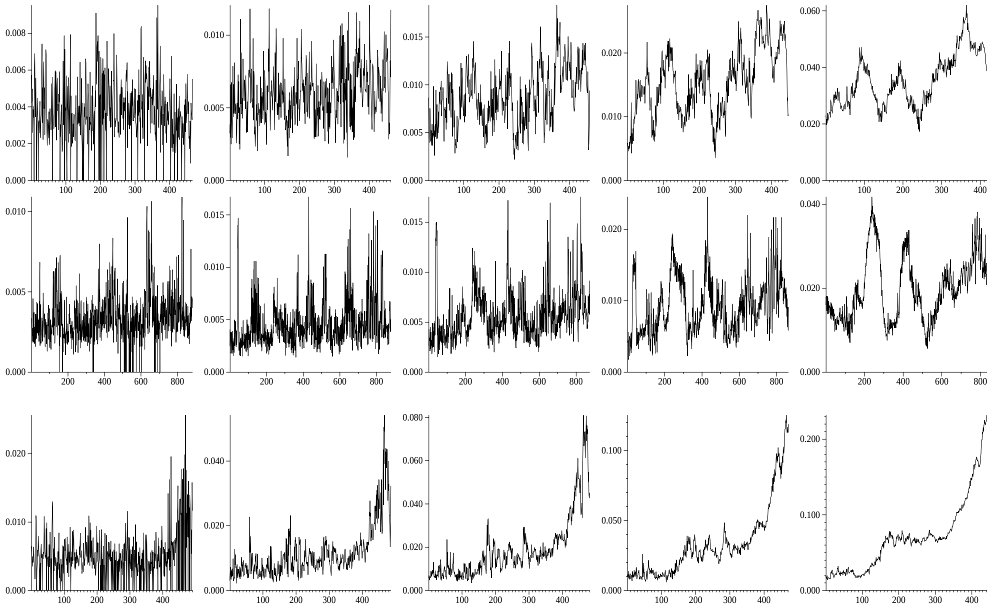
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.1: Similarity graph of the Hue Histogram.



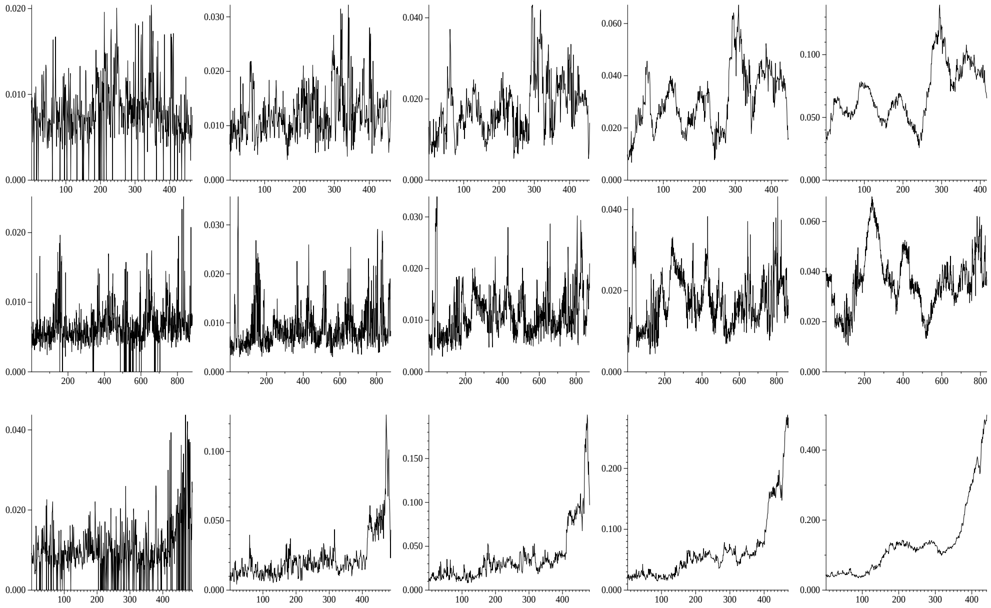
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.2: Similarity graph of the Spatial Hue Histogram.



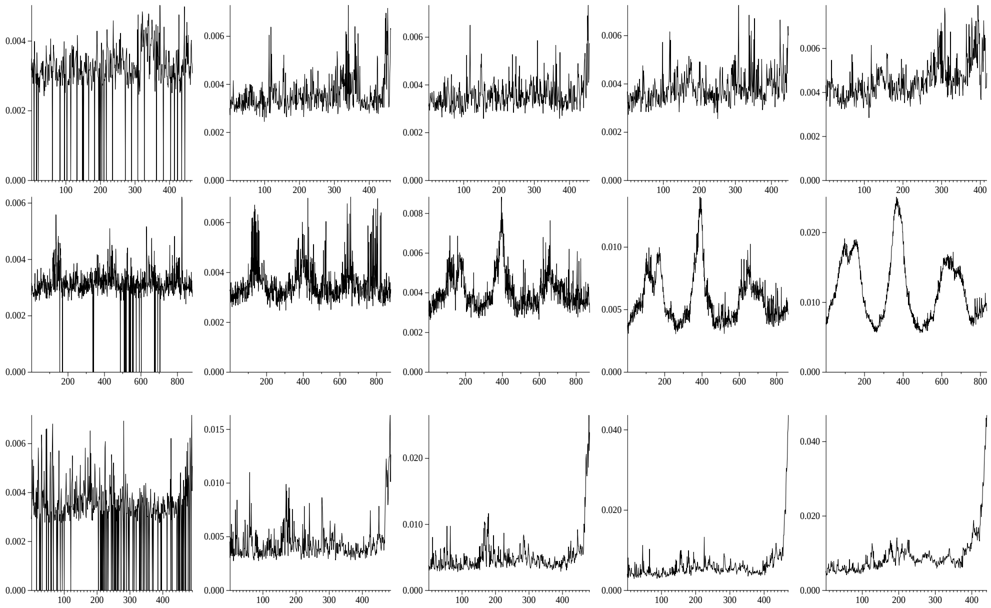
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.3: Similarity graph of the Colour Labels Histogram.



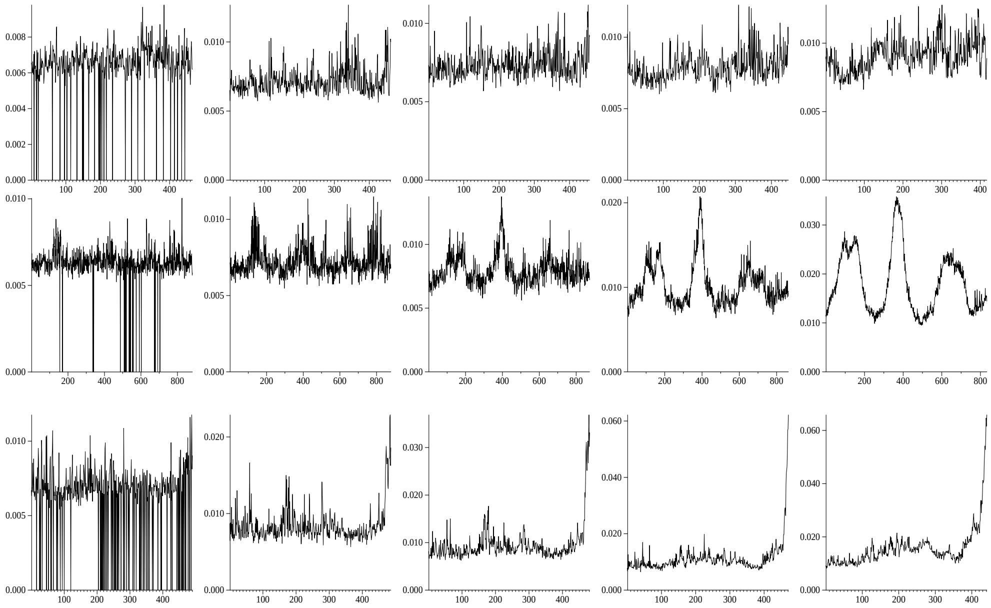
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.4: Similarity graph of the Spatial Colour Labels Histogram.



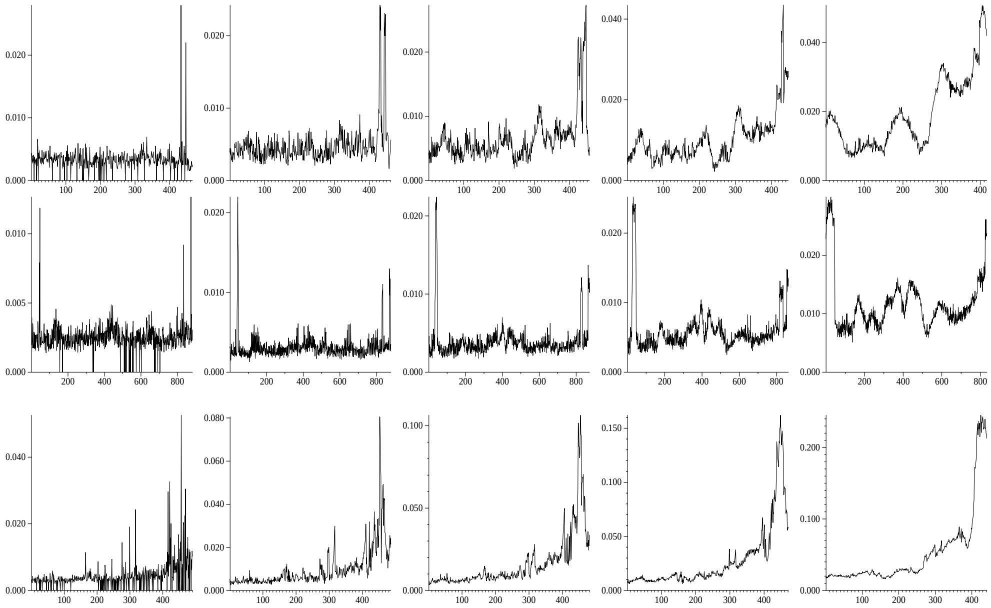
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.5: Similarity graph of the Texture Unit Histogram.



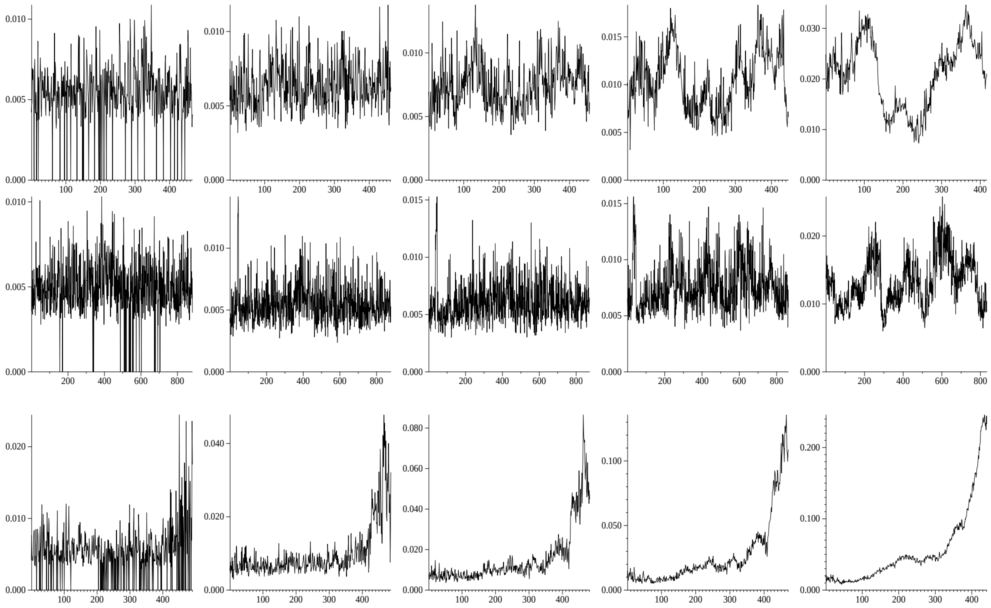
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.6: Similarity graph of the Spatial Texture Unit Histogram.



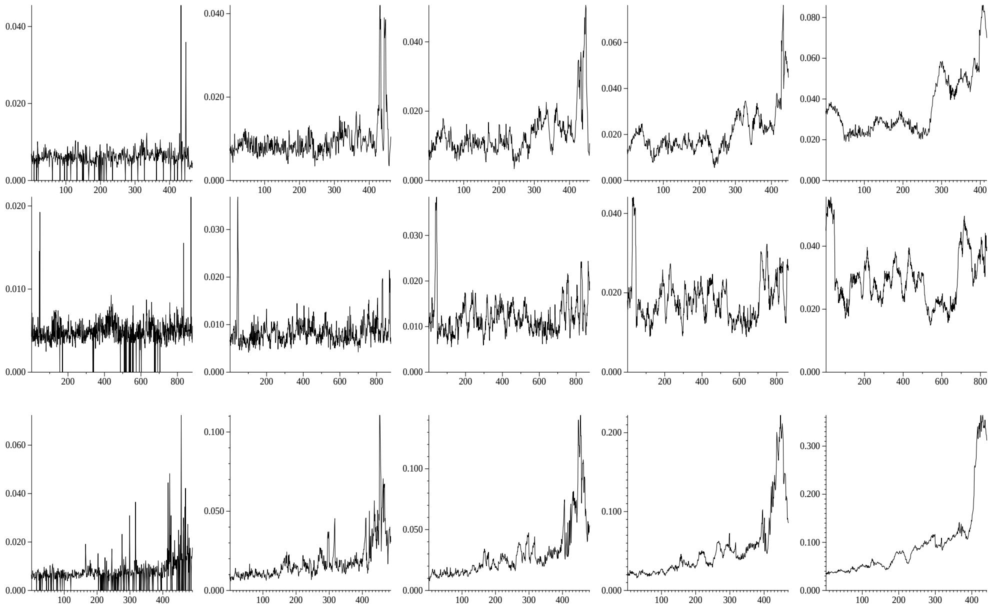
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.7: Similarity graph of the Cooccurrence Matrix of Intensities.



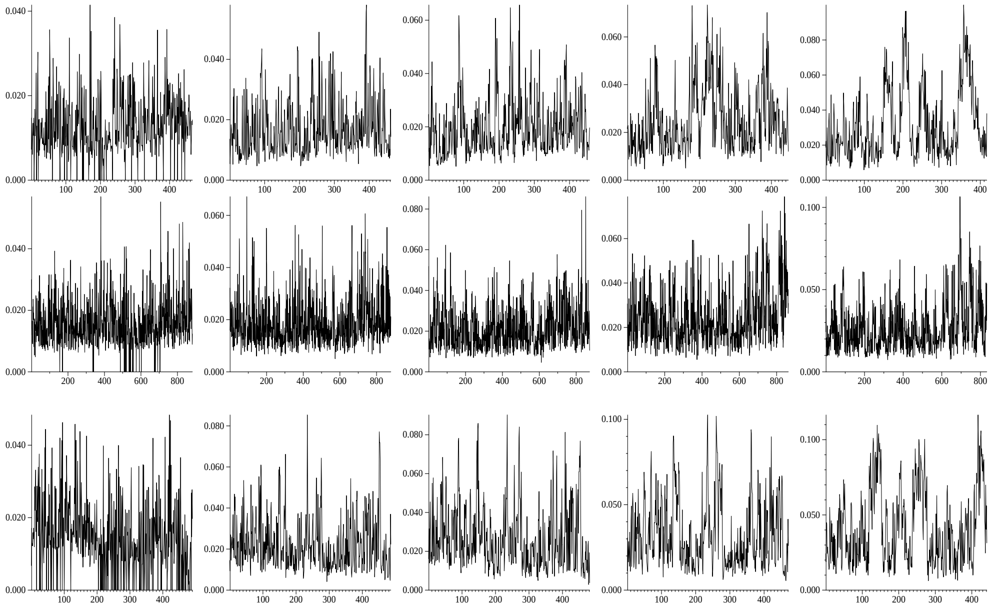
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.8: Similarity graph of the Cooccurrence Matrix of Hue.



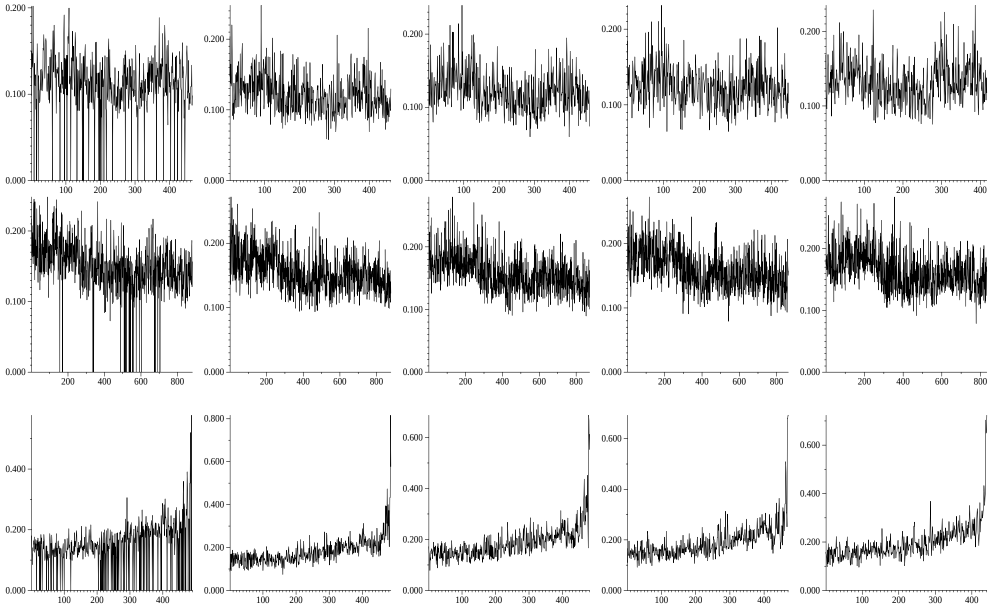
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.9: Similarity graph of the Spatial Cooccurrence Matrix of Intensities.



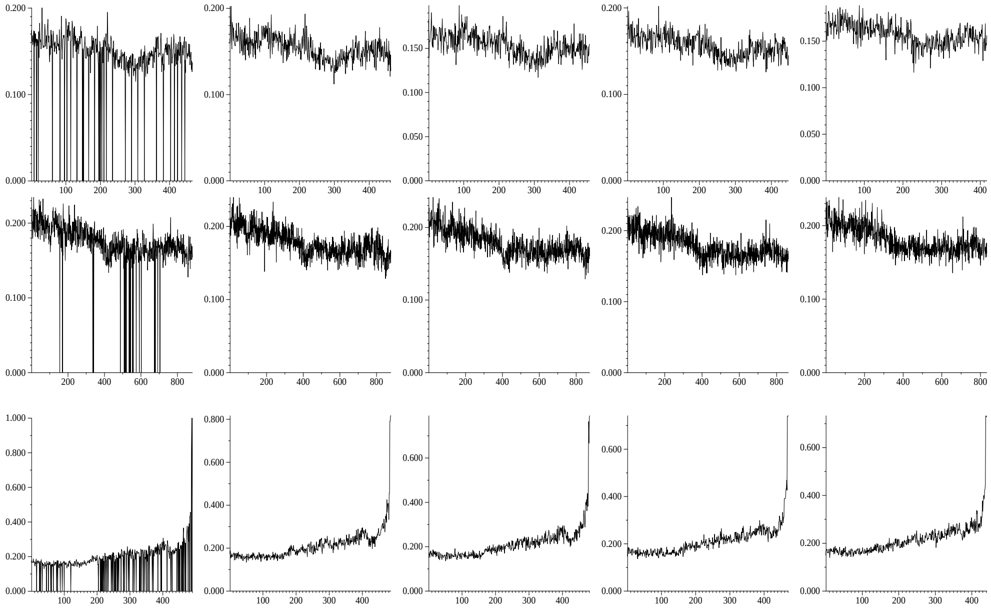
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.10: Similarity graph of the K-Cos edge curvature.



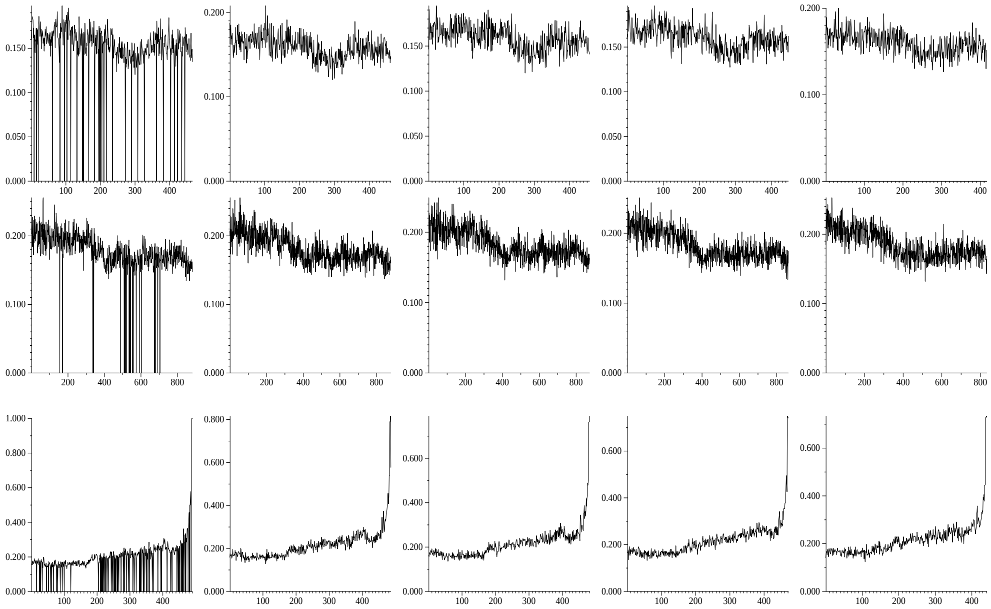
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.11: Similarity graph of the Circularity Histogram.



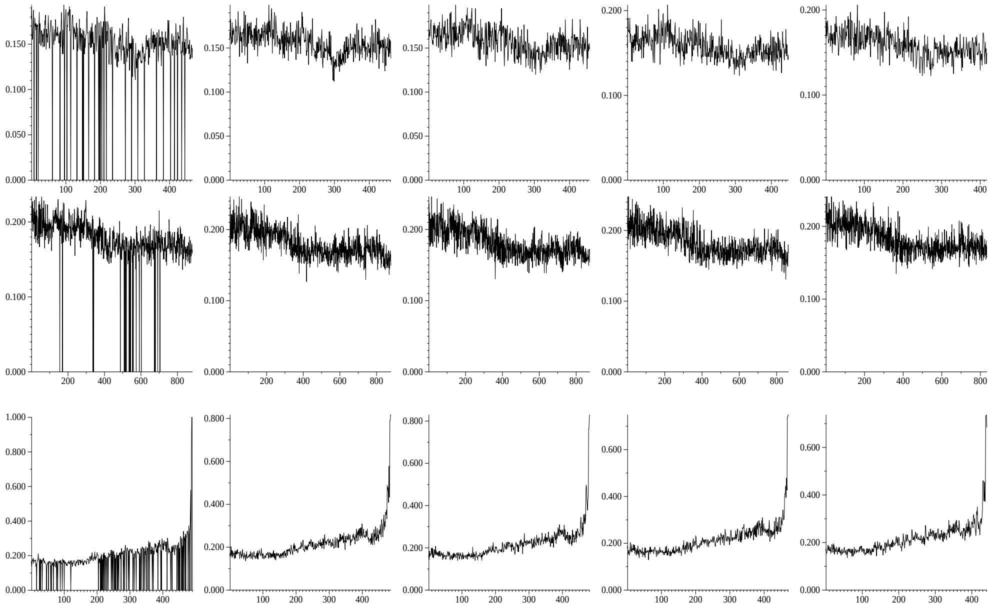
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.12: Similarity graph of the Ellipticity Histogram.



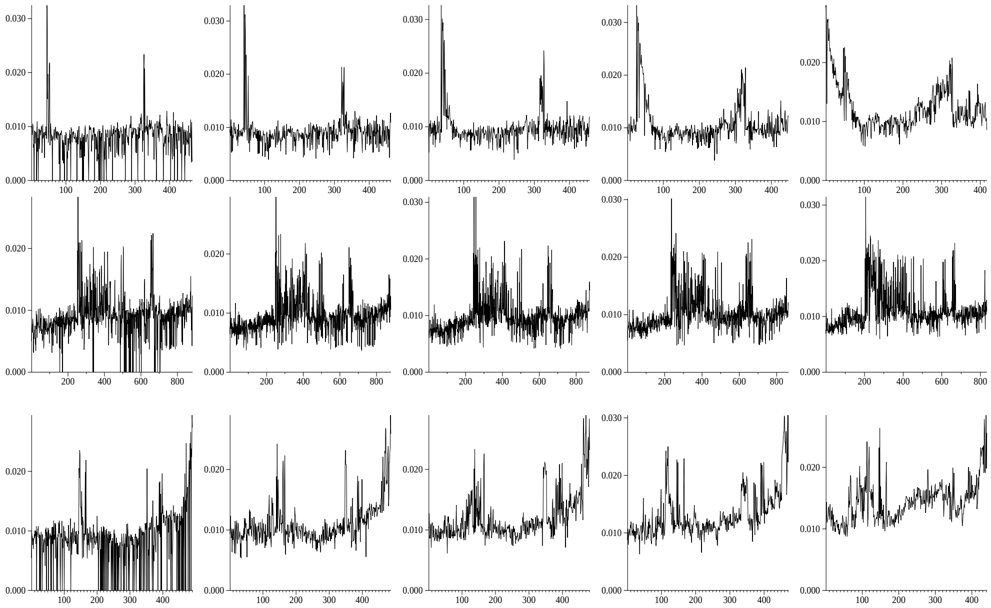
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.13: Similarity graph of the Rectangularity Histogram.



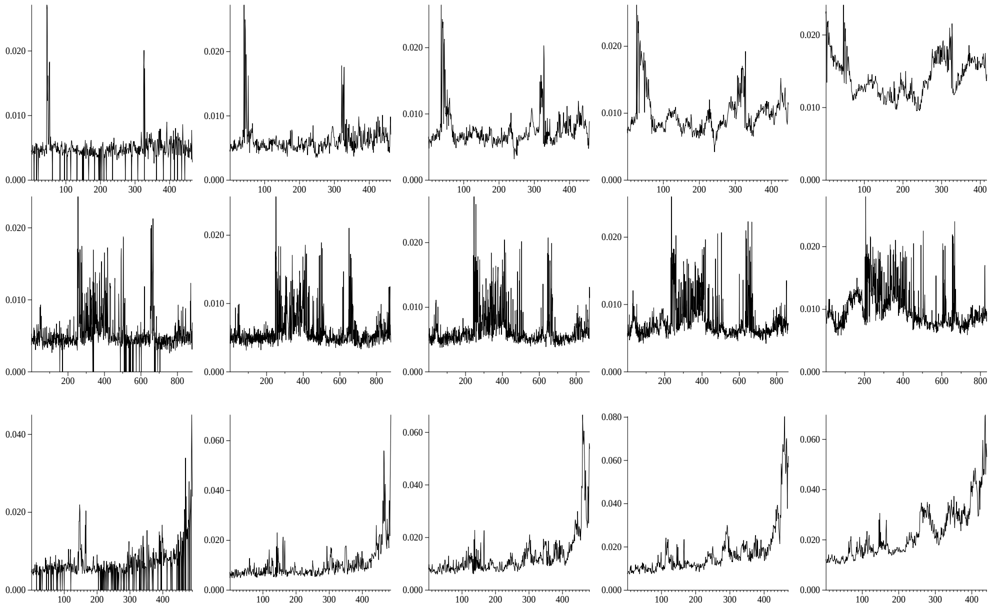
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.14: Similarity graph of the Convexity Histogram.



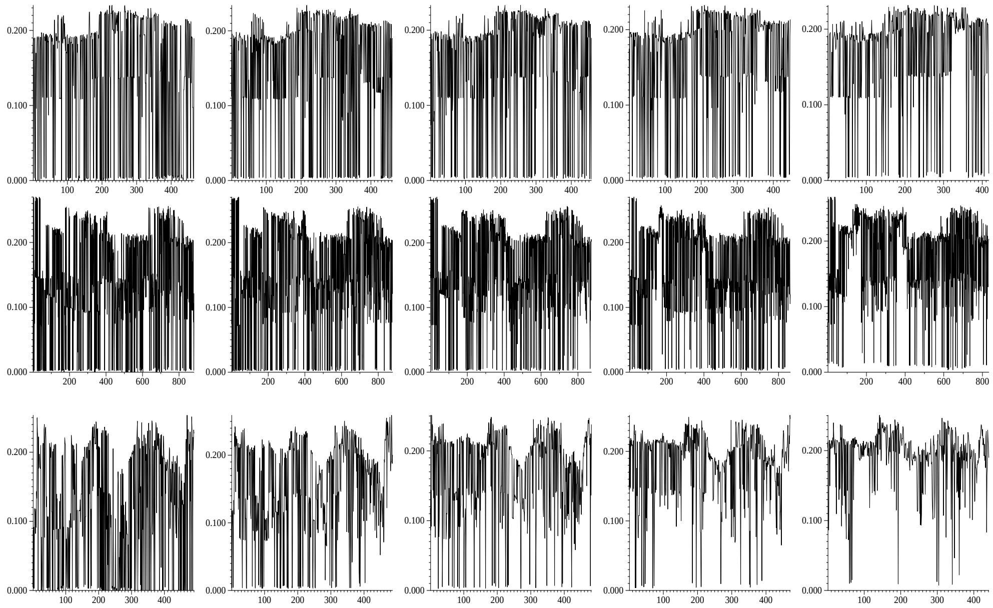
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.15: Similarity graph of the Multiscale Salience Distance Histogram.



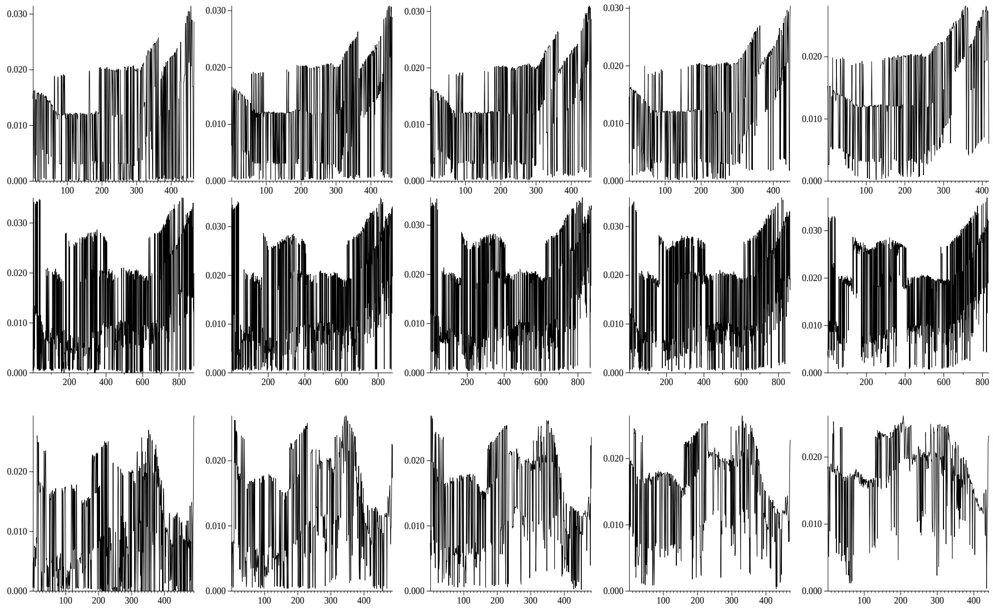
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.16: Similarity graph of the Colour Region Distance Histogram.



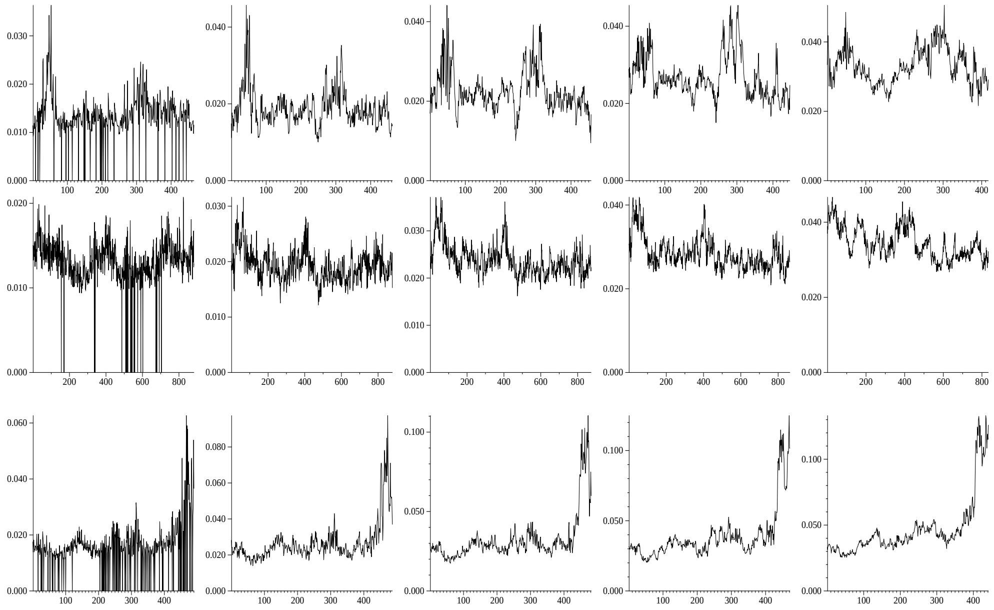
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.17: Similarity graph of the Colour Label & Boundary Distance Histogram.



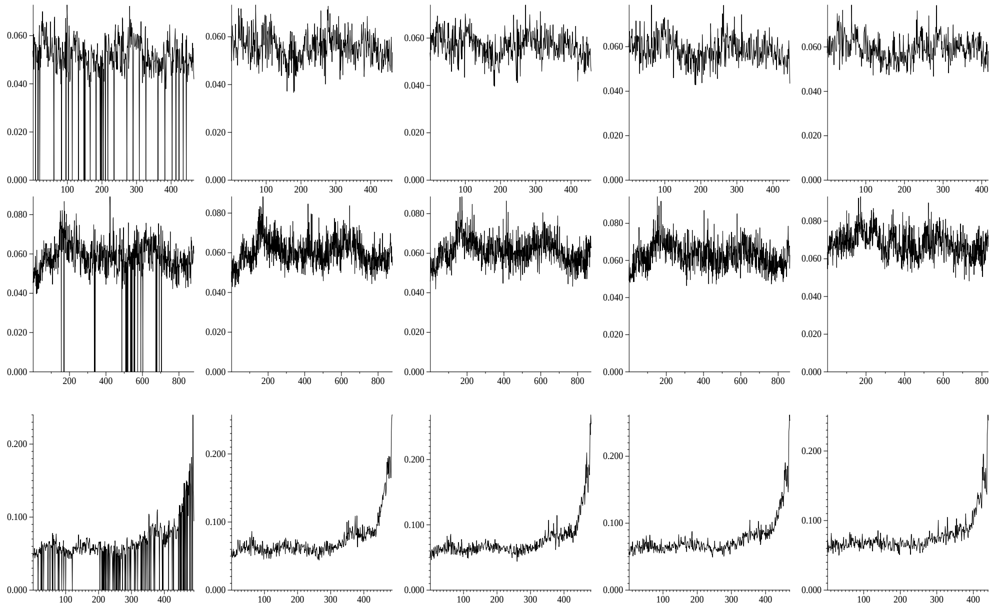
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.18: Similarity graph of the Colour Label & Edge Distance Histogram.



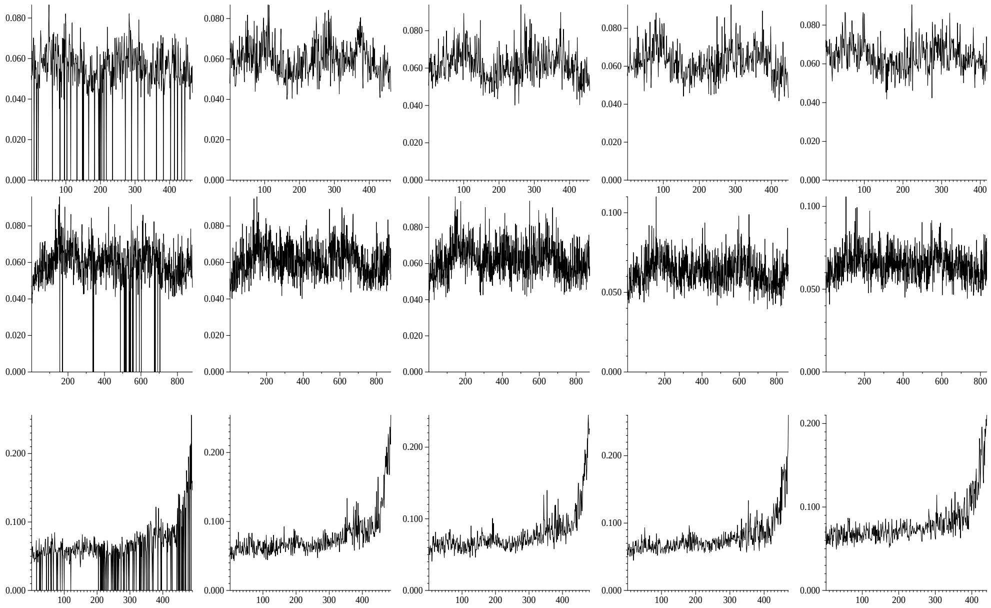
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.19: Similarity graph of the Orientation Histogram.



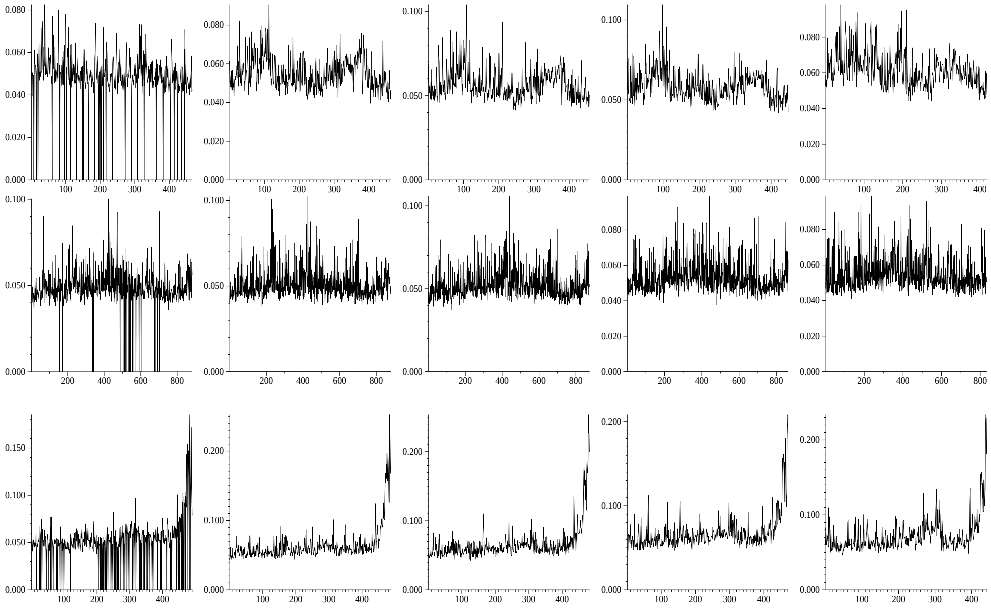
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.20: Similarity graph of the Triangle Area Histogram.



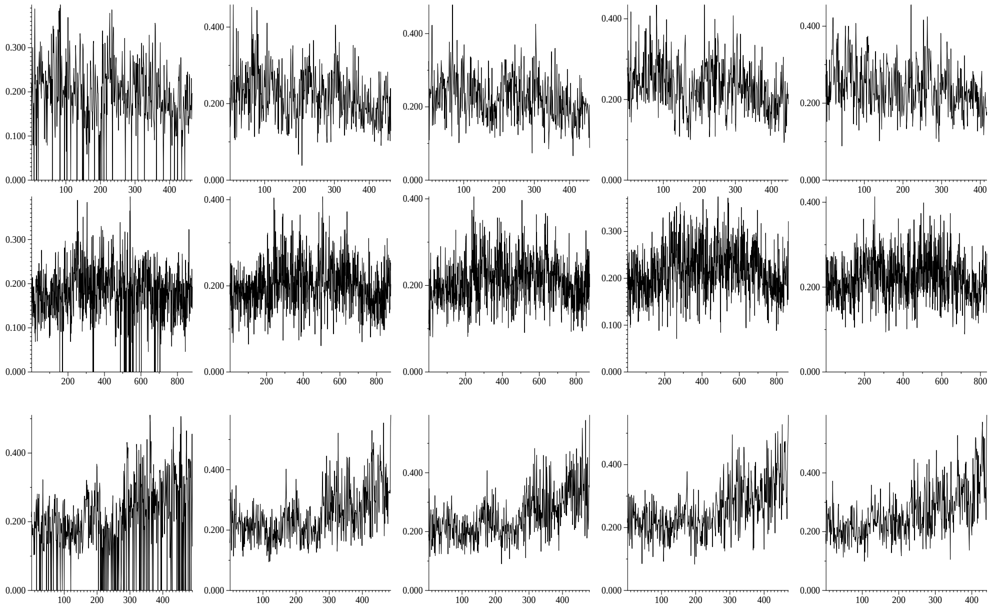
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.21: Similarity graph of the Triangle Ratio Histogram.



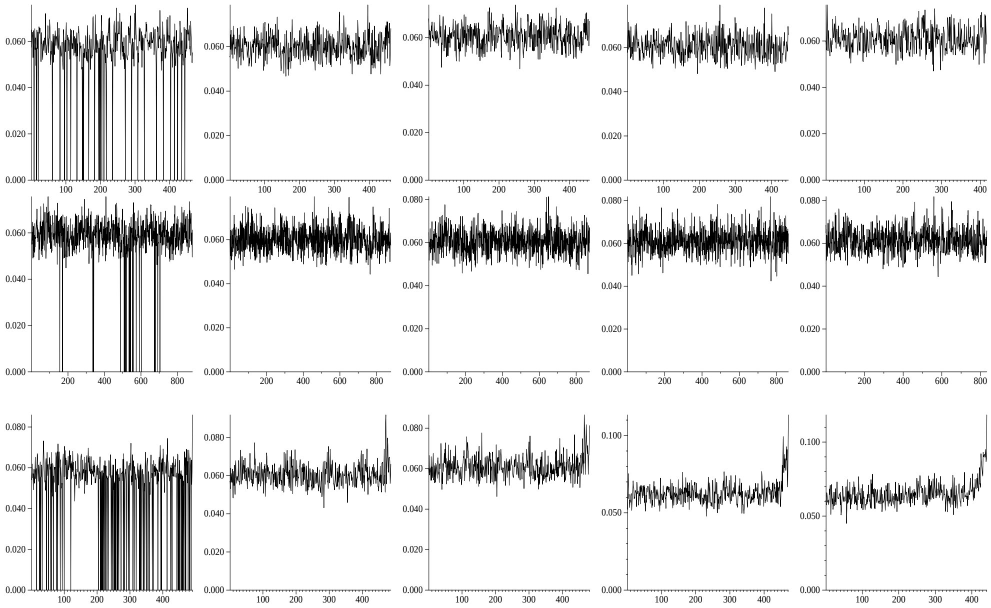
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.22: Similarity graph of the Triangle Length Histogram.



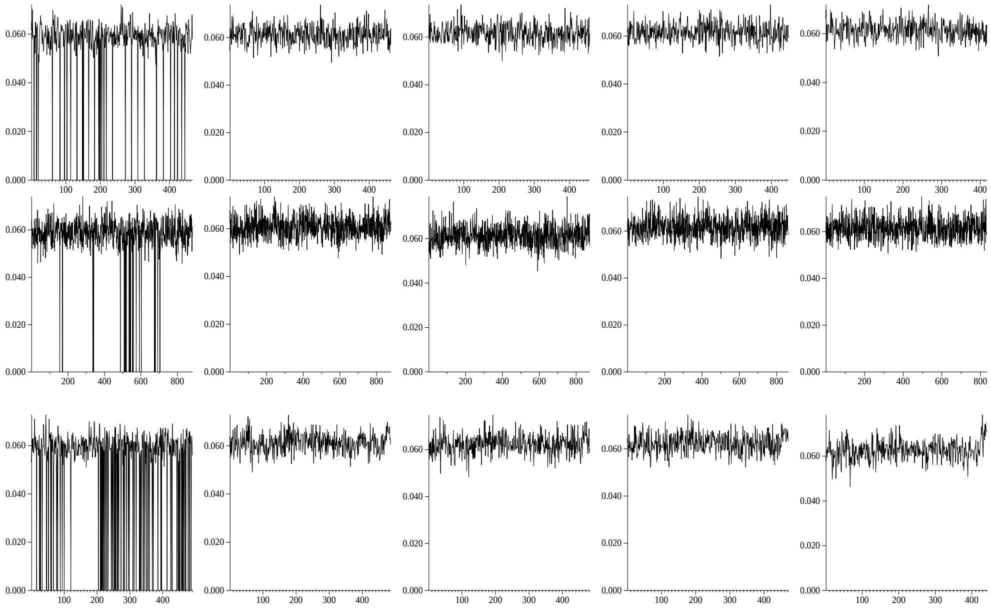
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.23: Similarity graph of the Circular Arcs Histogram.



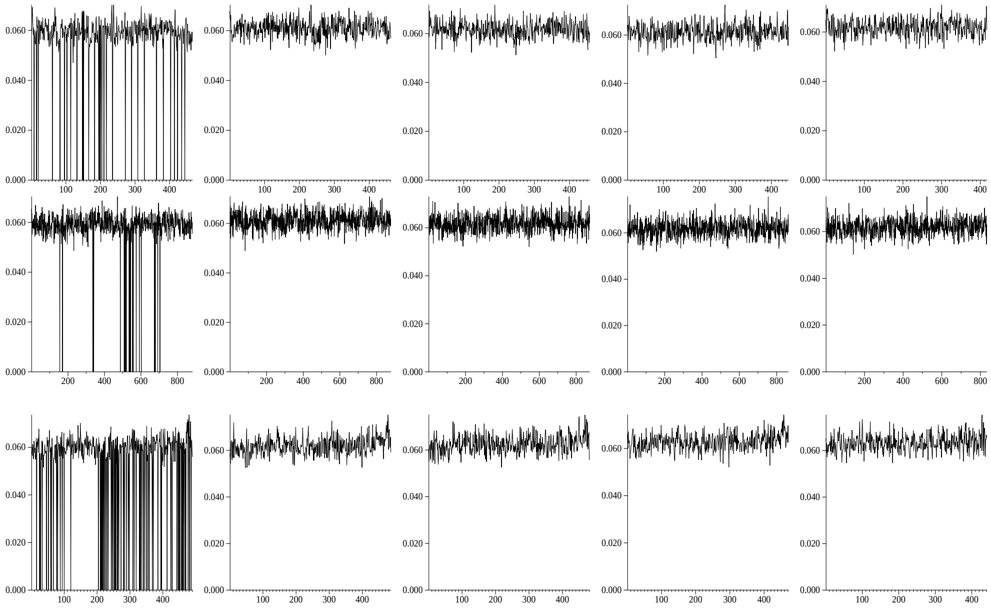
(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.24: Similarity graph of the Hue Entropy Histogram.



(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

Figure C.25: Similarity graph of the Local Difference Histogram.



(a) $d = 1$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$ (e) $d = 50$

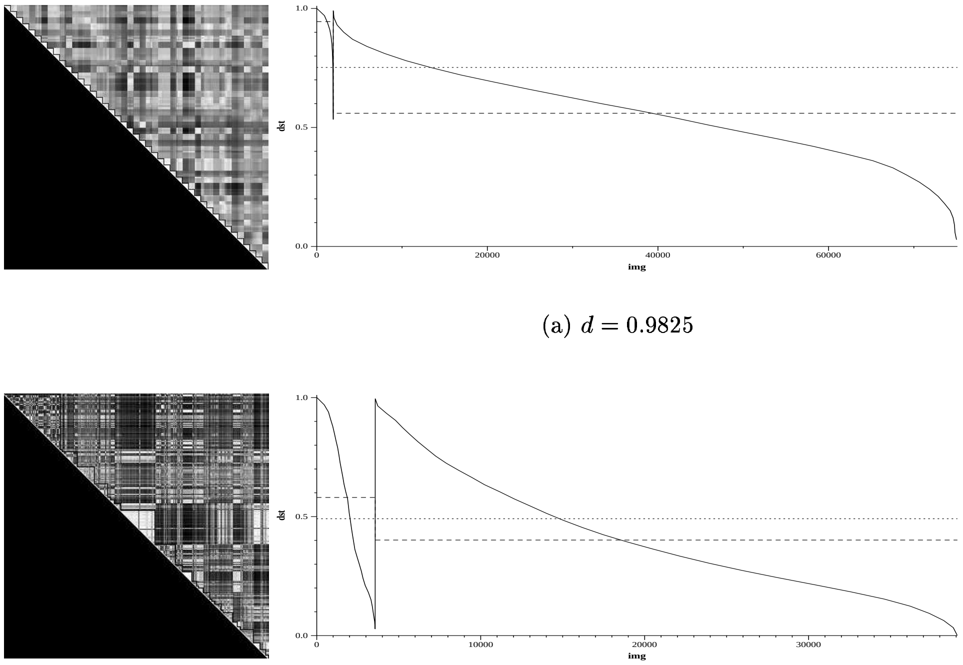
Figure C.26: Similarity graph of the Local Black Histogram.

Appendix D

Results: Distance Matrix

In this section we include the Distance matrices used in Chapter 6. As in the case of the histogram plots we had to post process the images for better printing.

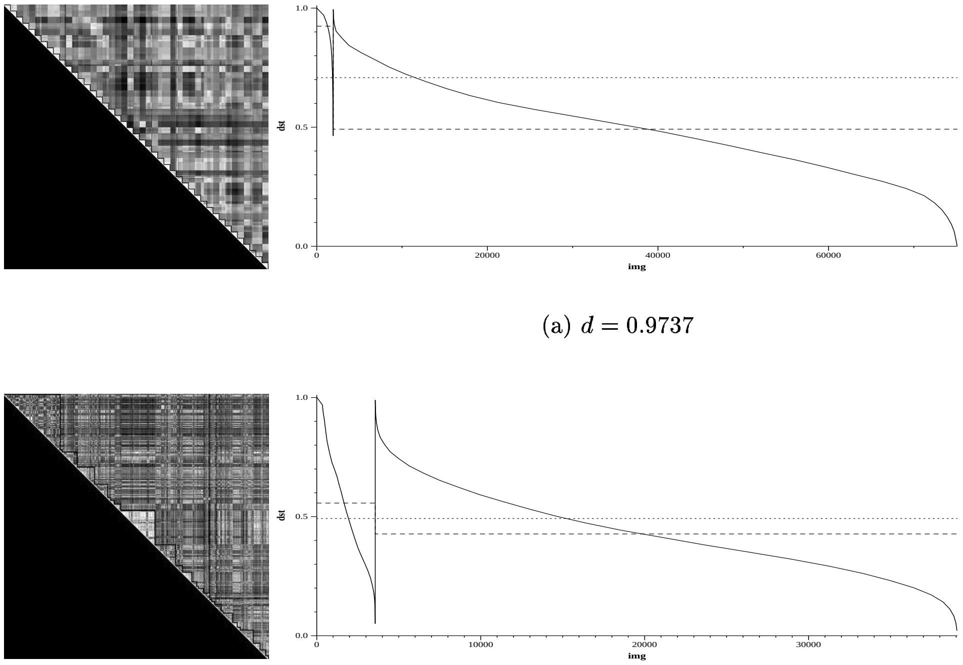
We have also included the graph of the distances and the corresponding summary value, as presented in Section 5.2.3.



(a) $d = 0.9825$

(b) $d = 0.6731$

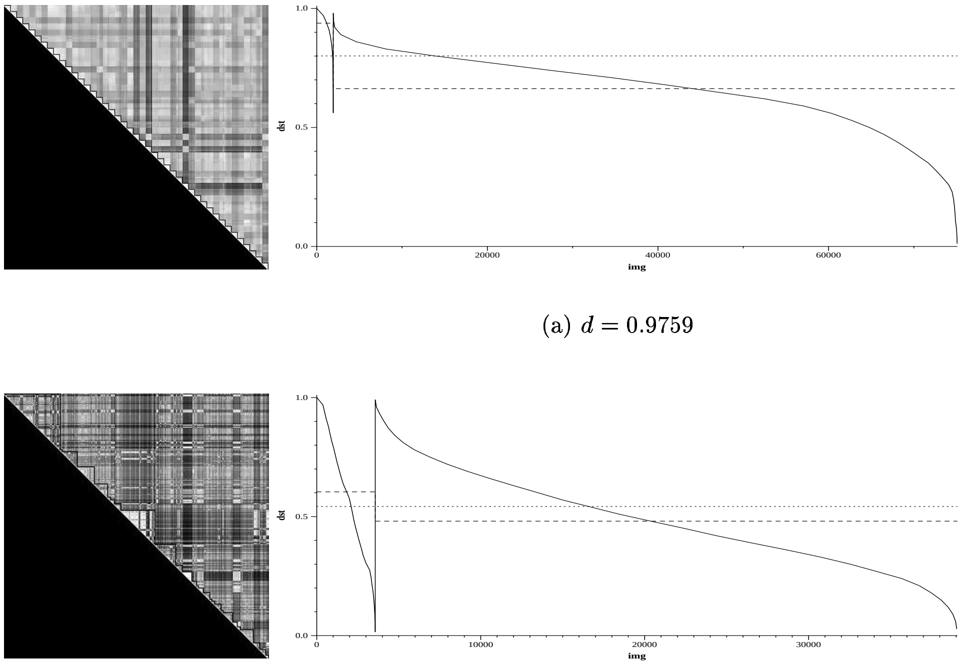
Figure D.1: Colour Labels (CL)



(a) $d = 0.9737$

(b) $d = 0.6583$

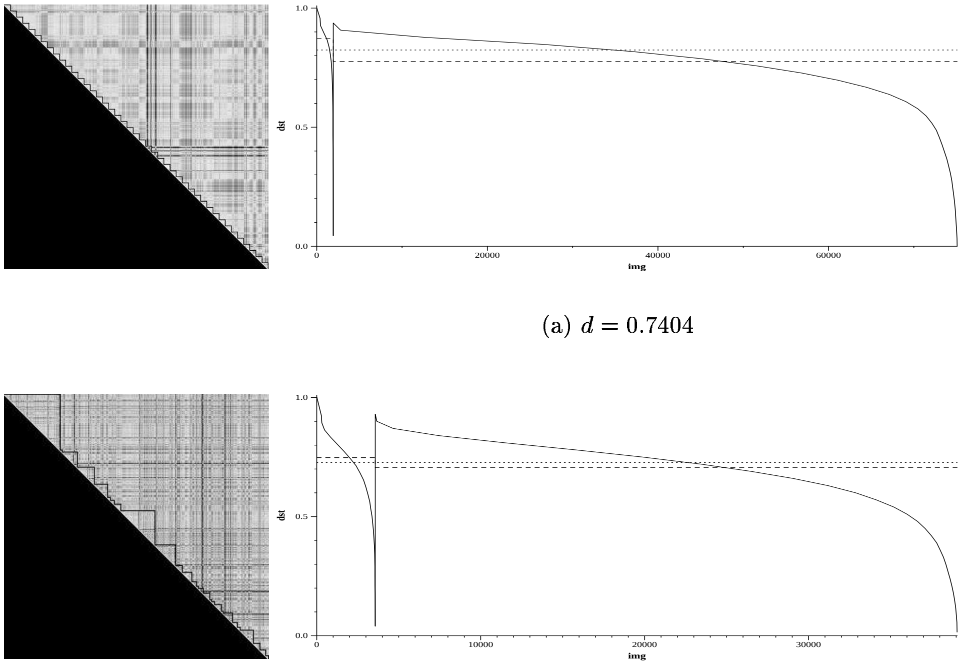
Figure D.2: Spatial Colour Labels (CL-SP)



(a) $d = 0.9759$

(b) $d = 0.6101$

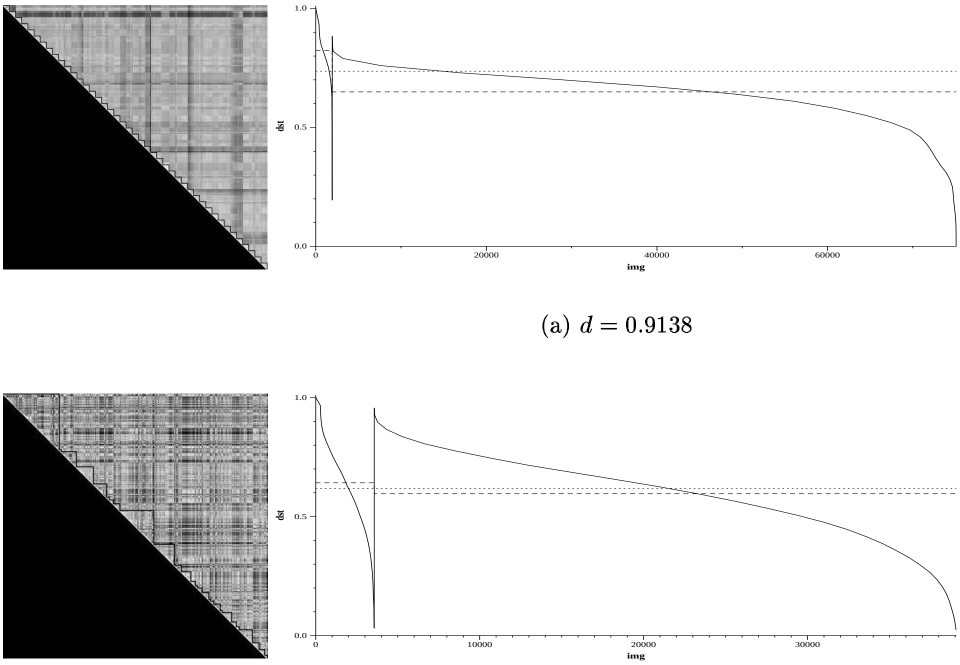
Figure D.3: Cooccurrence Matrix (intensity) (COOC)



(a) $d = 0.7404$

(b) $d = 0.5372$

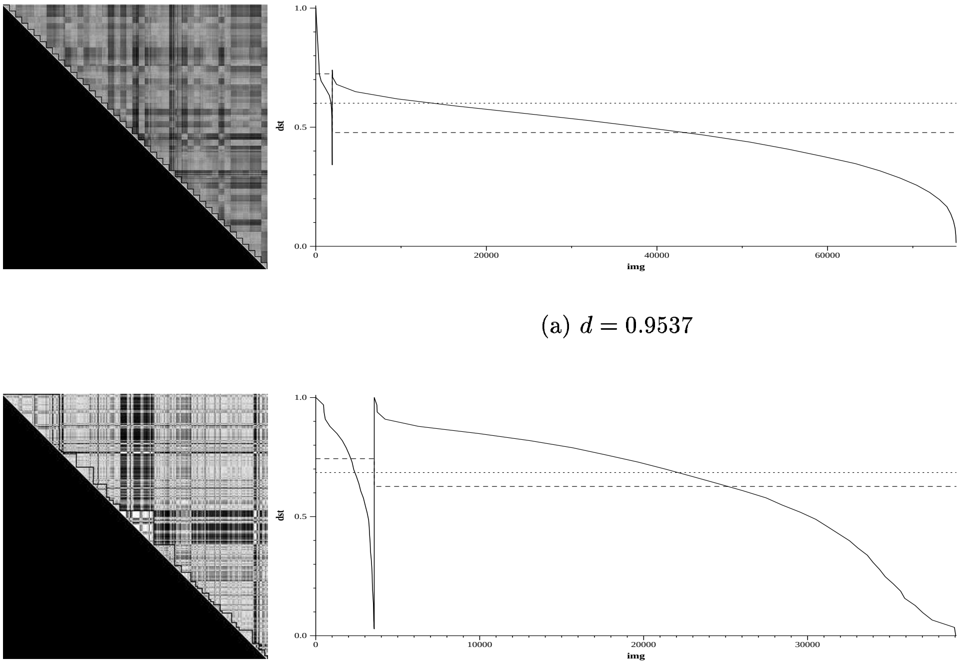
Figure D.4: Edge Distance (MSSDT)



(a) $d = 0.9138$

(b) $d = 0.5502$

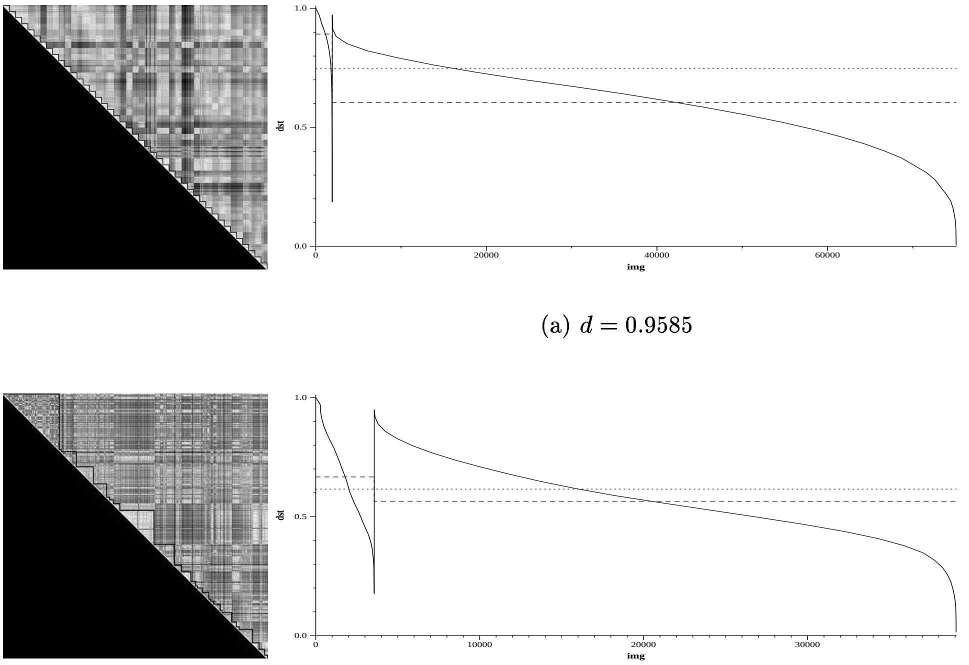
Figure D.5: Orientation (ORNT)



(a) $d = 0.9537$

(b) $d = 0.6073$

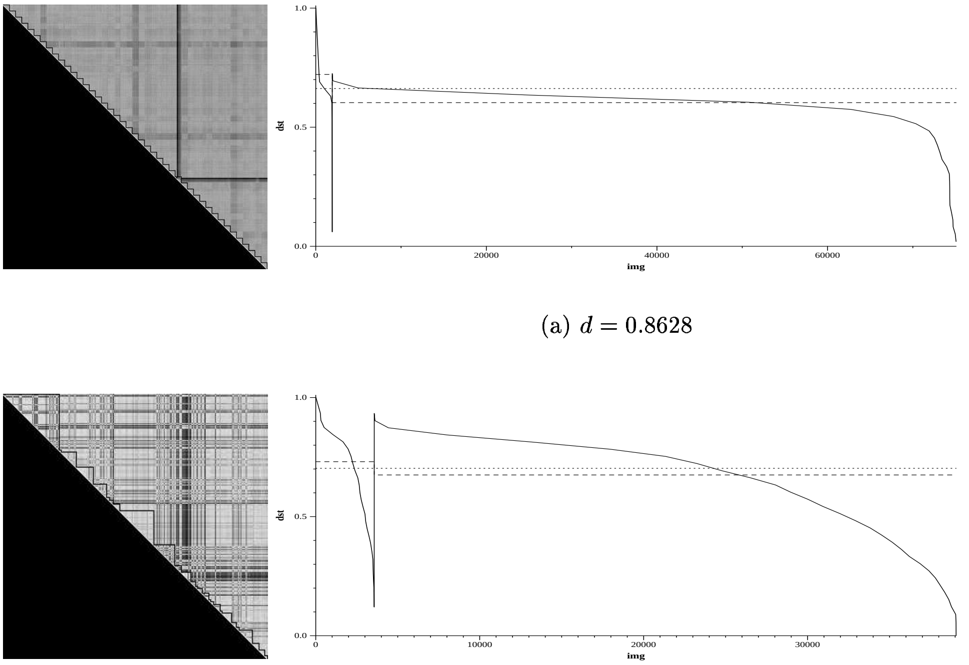
Figure D.6: Hue Entropy (HENT)



(a) $d = 0.9585$

(b) $d = 0.6463$

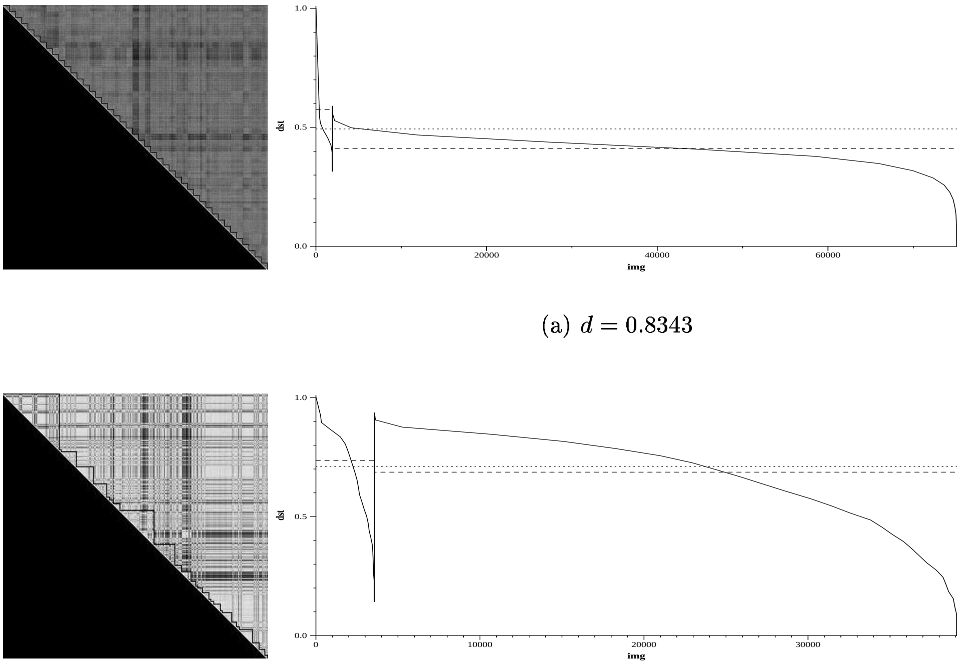
Figure D.7: Clsh (CLSH)



(a) $d = 0.8628$

(b) $d = 0.5887$

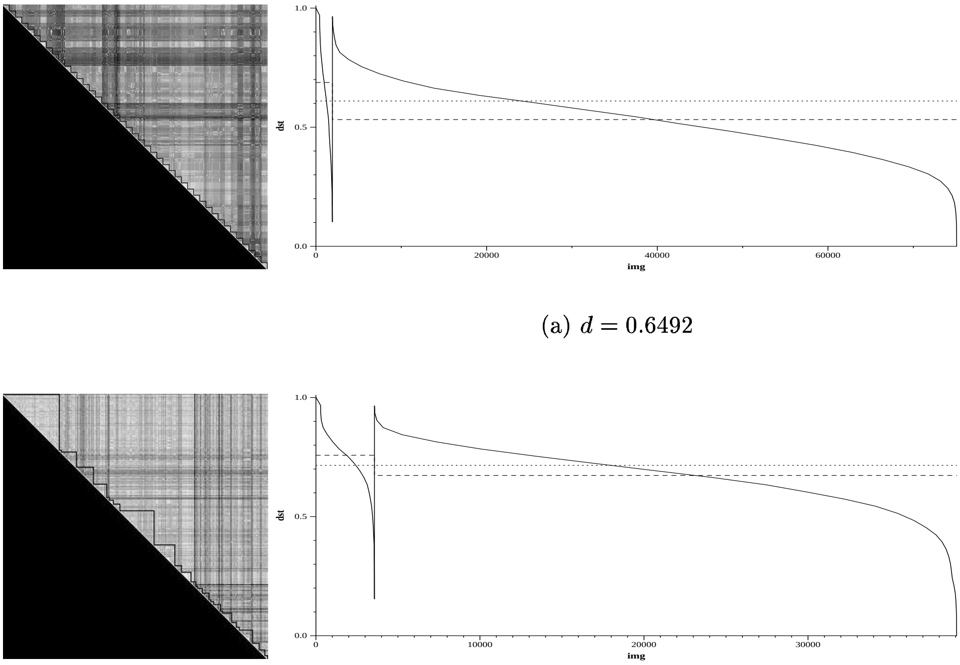
Figure D.8: Local Black (LBLK)



(a) $d = 0.8343$

(b) $d = 0.5650$

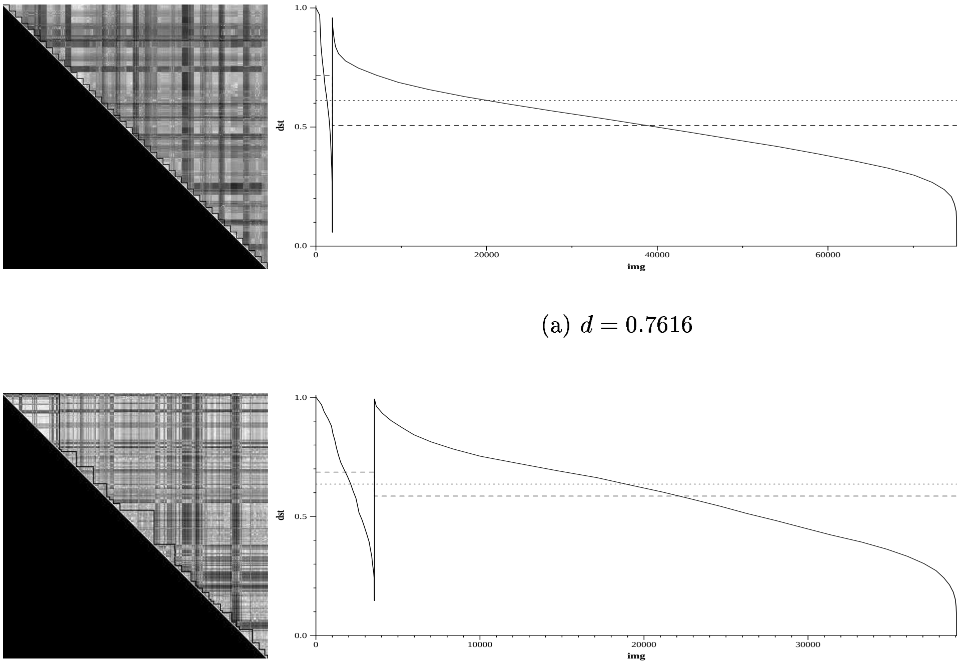
Figure D.9: Local Difference (LDIFF)



(a) $d = 0.6492$

(b) $d = 0.6239$

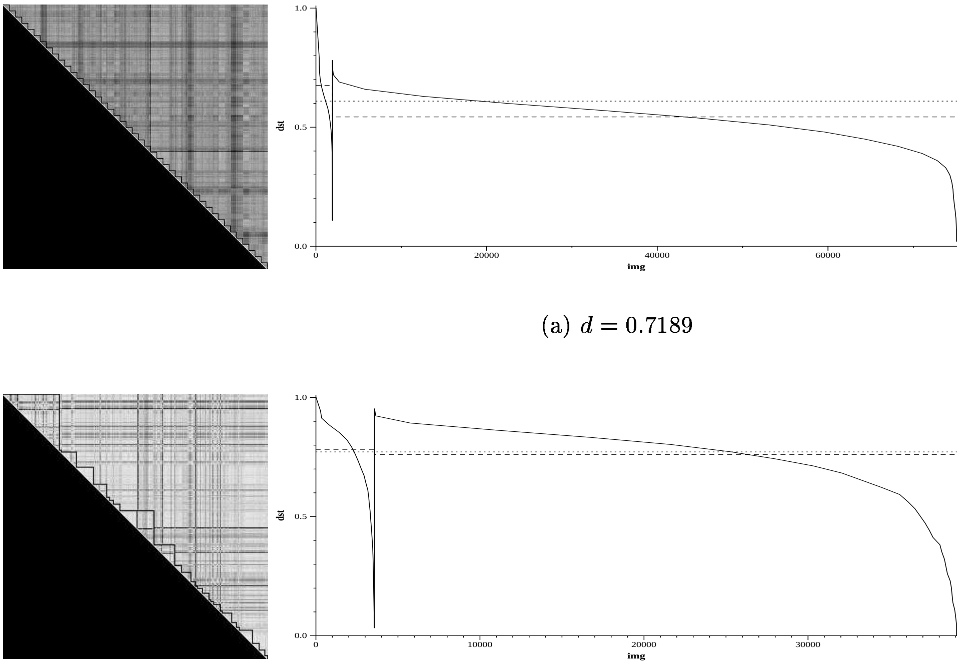
Figure D.10: HBKDT(HBKDT)



(a) $d = 0.7616$

(b) $d = 0.6292$

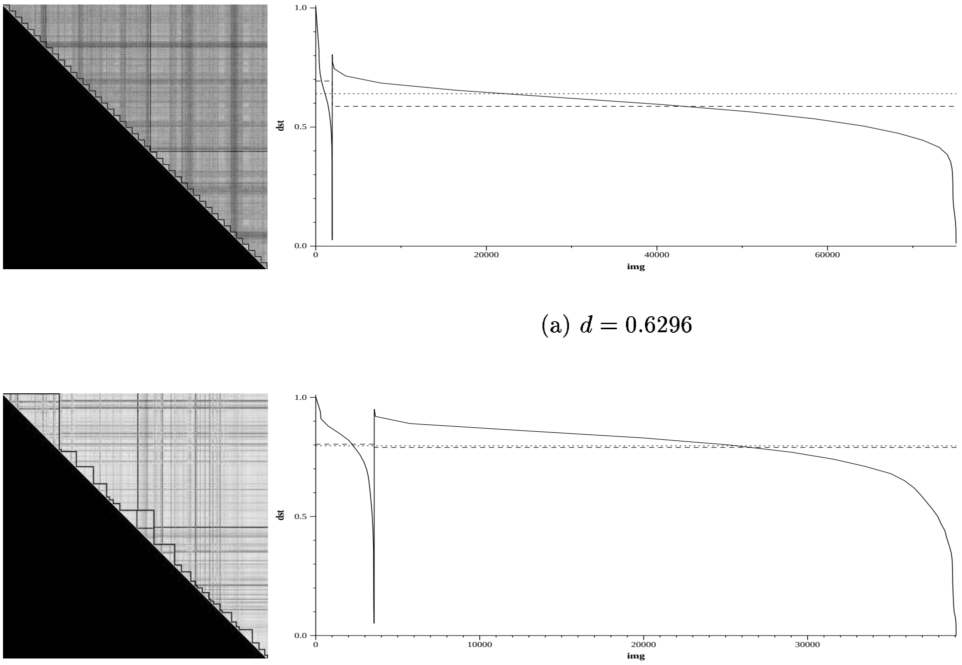
Figure D.11: HBKDT2D(HBKDT2D)



(a) $d = 0.7189$

(b) $d = 0.5817$

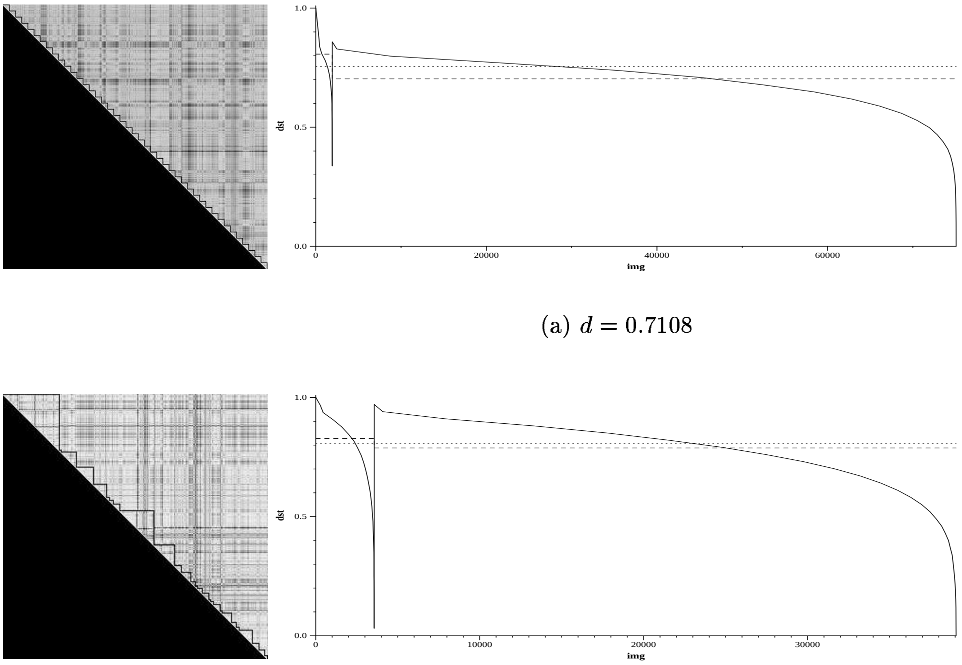
Figure D.12: Triangle Area (TARE)



(a) $d = 0.6296$

(b) $d = 0.5539$

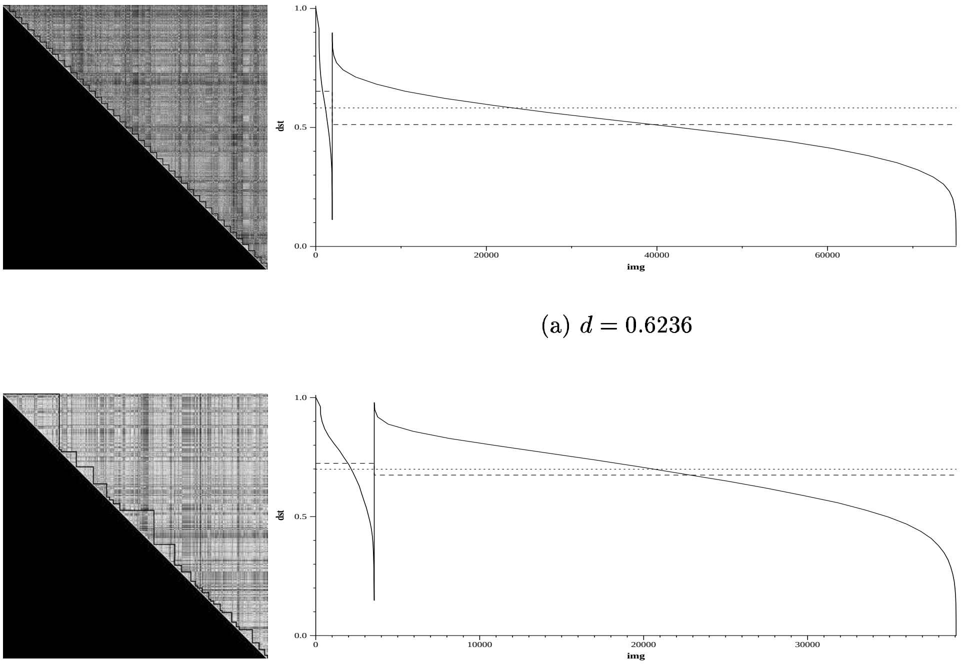
Figure D.13: Triangle Ratio (TRAT)



(a) $d = 0.7108$

(b) $d = 0.5848$

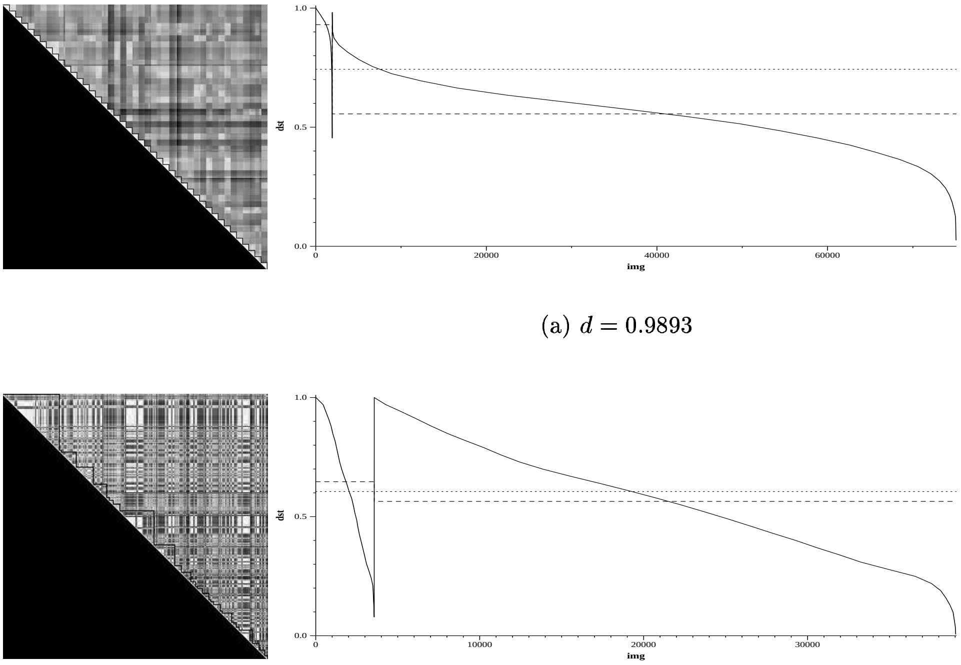
Figure D.14: Triangle Length (TLEN)



(a) $d = 0.6236$

(b) $d = 0.5602$

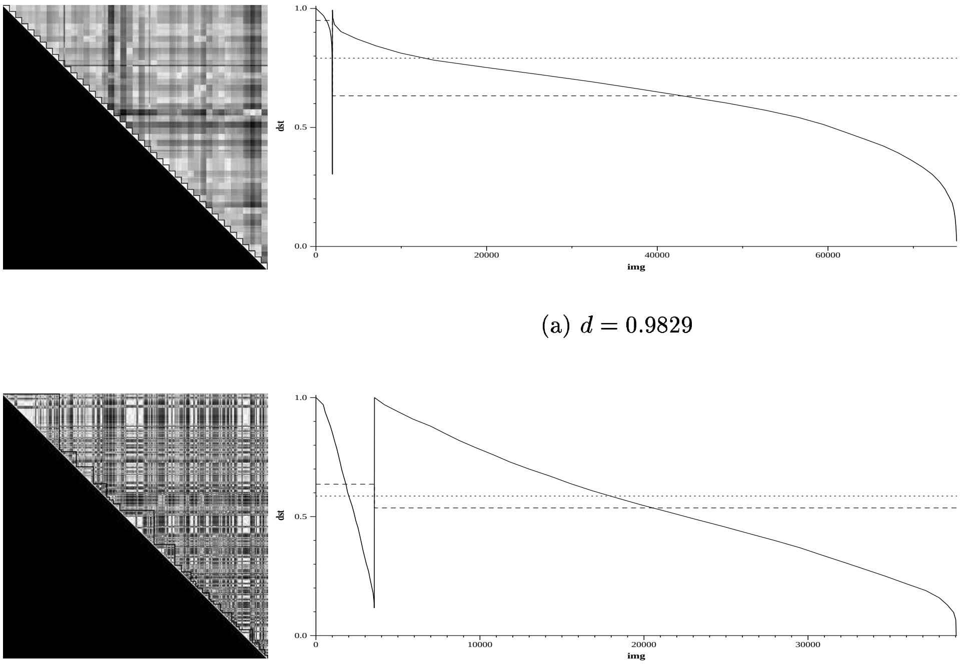
Figure D.15: Circular Arcs (CARC)



(a) $d = 0.9893$

(b) $d = 0.5399$

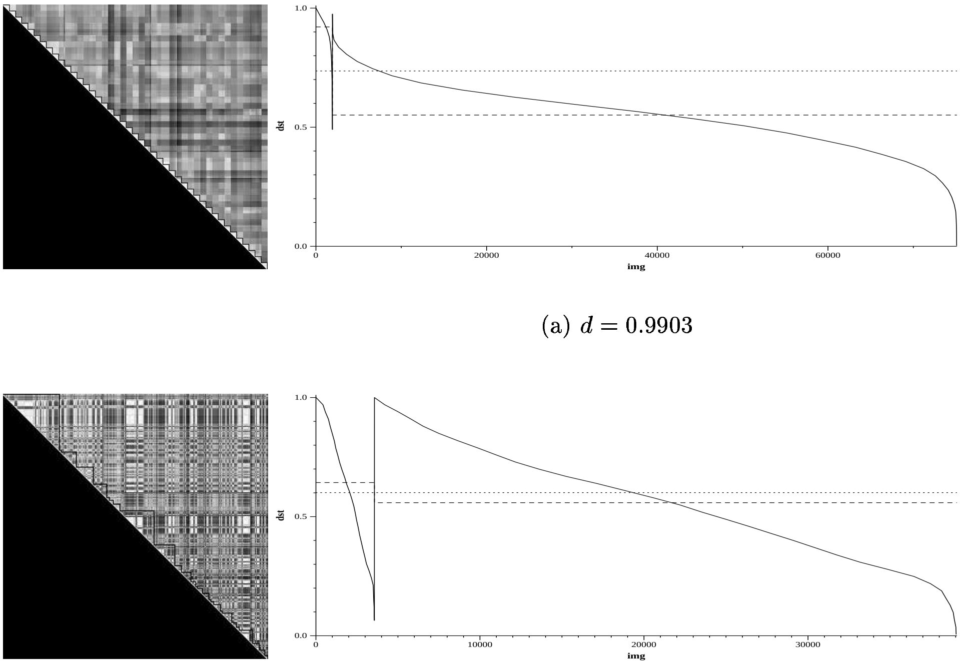
Figure D.16: Hue (HUE)



(a) $d = 0.9829$

(b) $d = 0.5468$

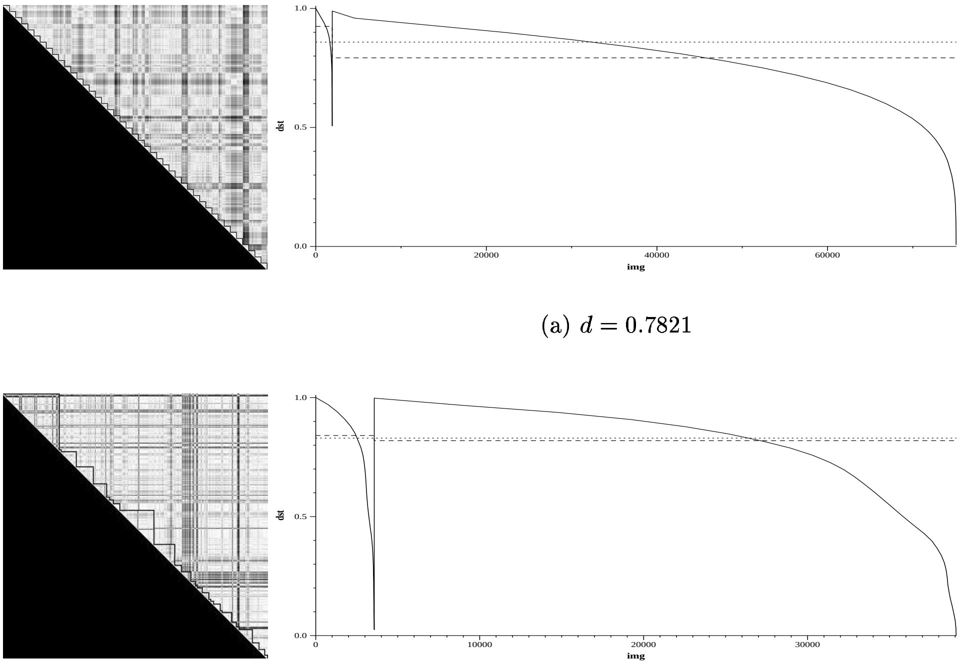
Figure D.17: Cooccurrence Matrix Hue (HCC)



(a) $d = 0.9903$

(b) $d = 0.5413$

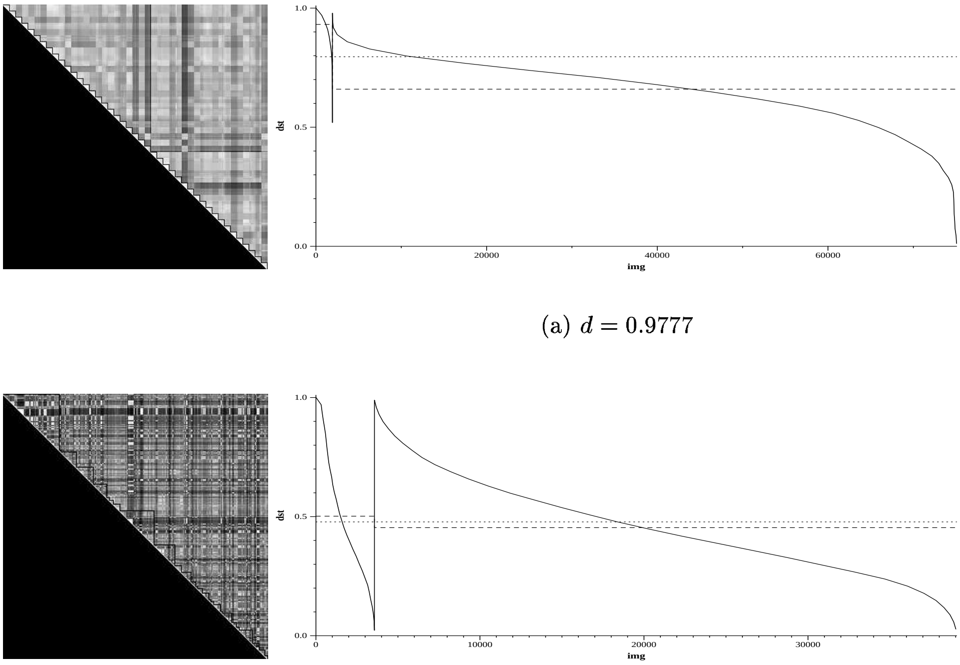
Figure D.18: Spatial Hue (HUE-SP)



(a) $d = 0.7821$

(b) $d = 0.5297$

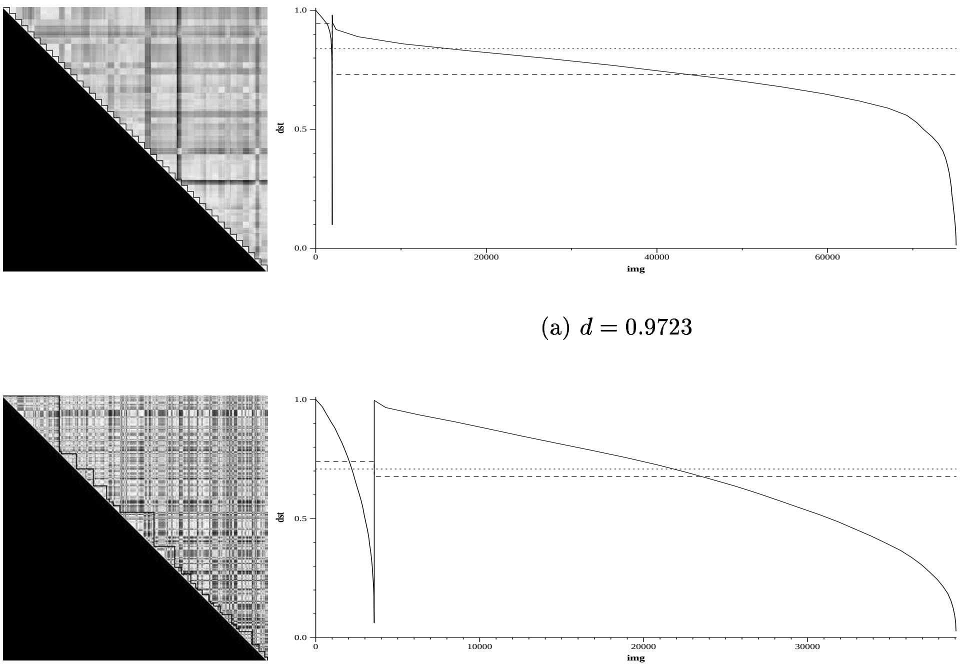
Figure D.19: K-Cos Edge Curvature (KCOS)



(a) $d = 0.9777$

(b) $d = 0.5134$

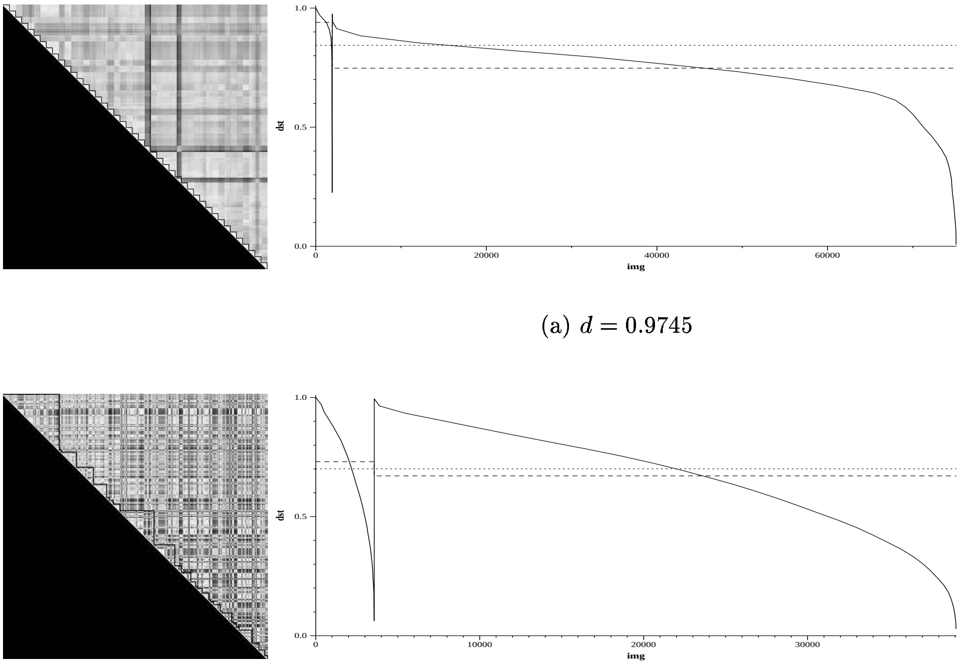
Figure D.20: Spatial Cooccurrence Matrix (CC-SP)



(a) $d = 0.9723$

(b) $d = 0.5448$

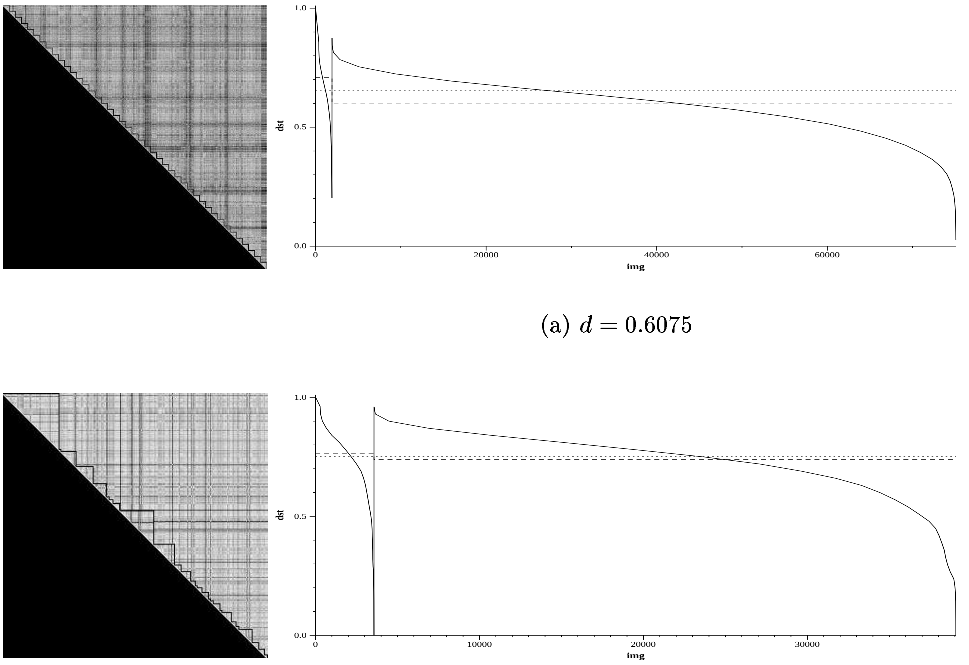
Figure D.21: T-Unit (TUNIT)



(a) $d = 0.9745$

(b) $d = 0.5443$

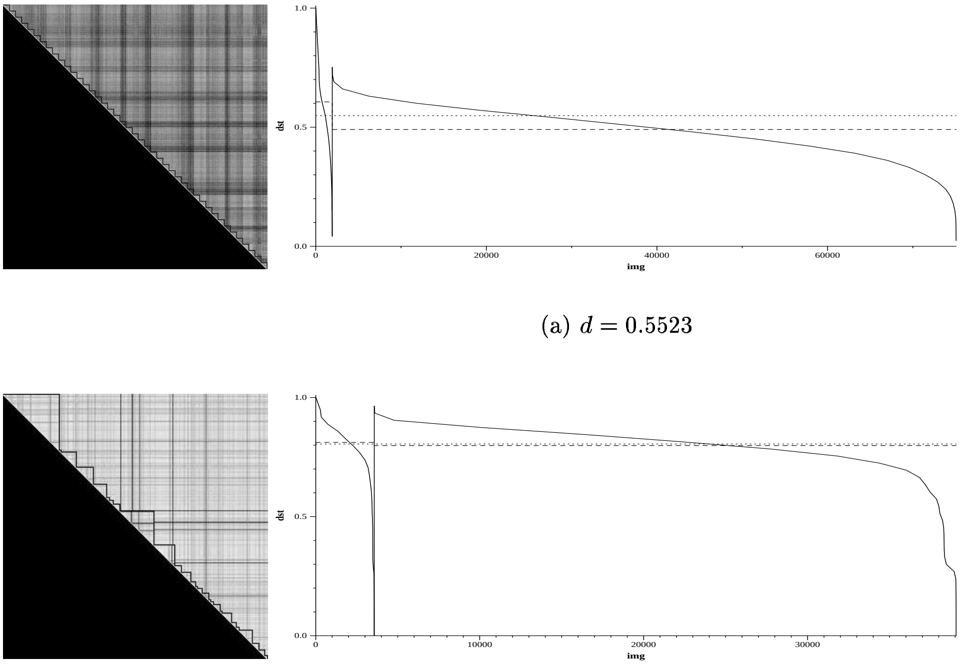
Figure D.22: Spatial T-Unit (TU-SP)



(a) $d = 0.6075$

(b) $d = 0.5202$

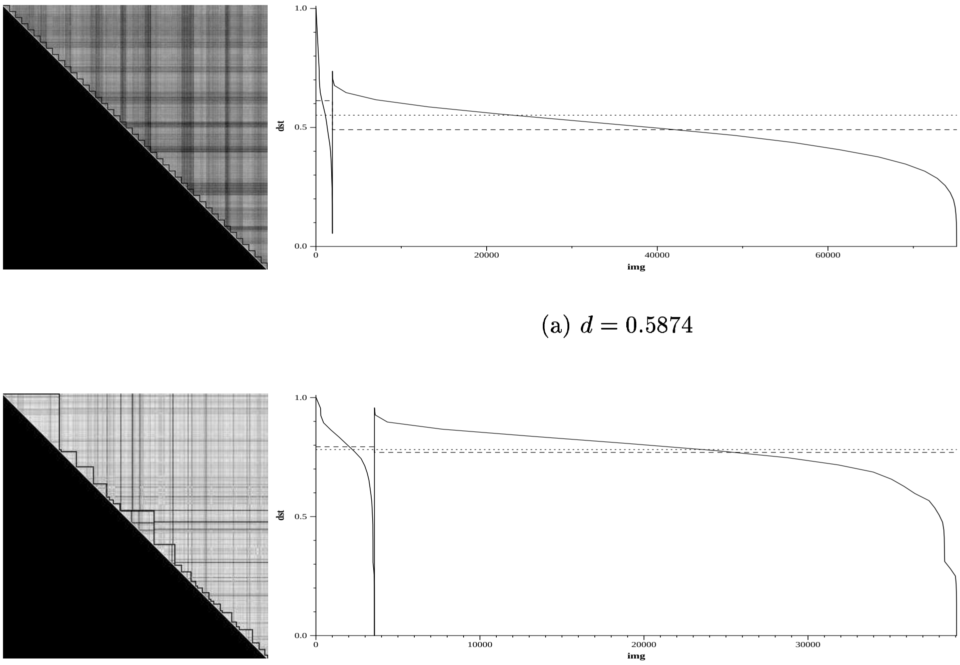
Figure D.23: Circularity (CIRC)



(a) $d = 0.5523$

(b) $d = 0.5523$

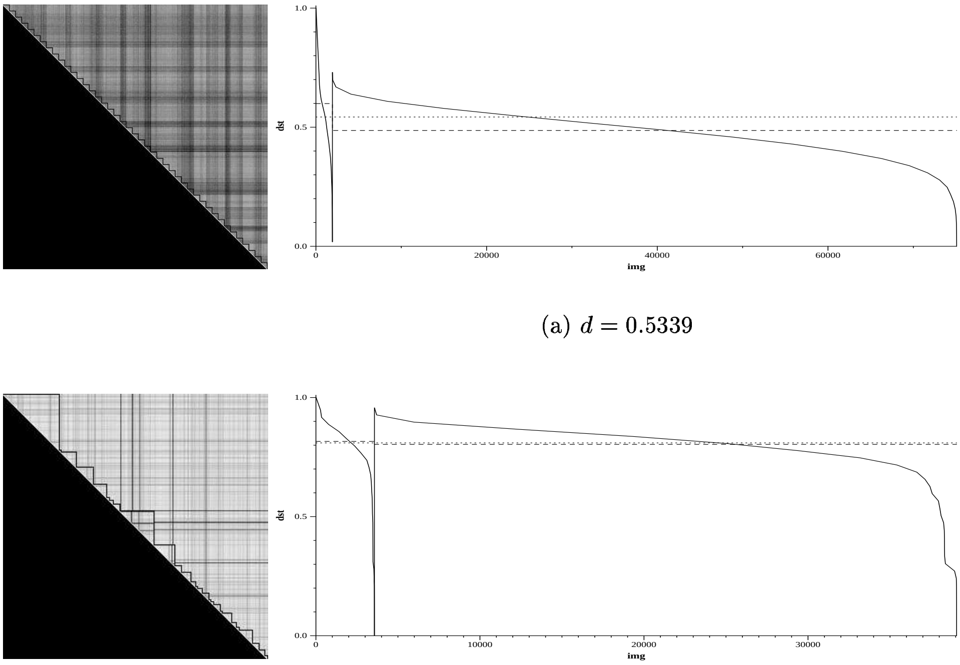
Figure D.24: Convexity (CONV)



(a) $d = 0.5874$

(b) $d = 0.5509$

Figure D.25: Ellipticity (ELIP)



(a) $d = 0.5339$

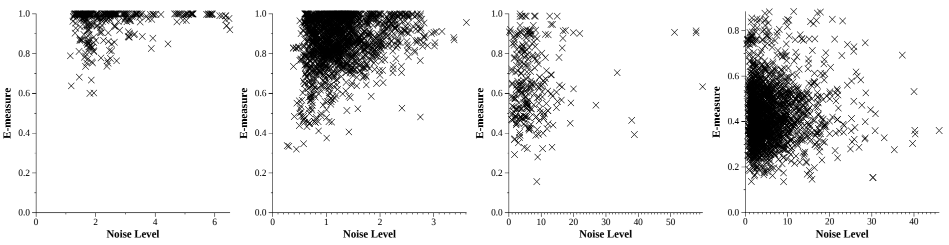
(b) $d = 0.5271$

Figure D.26: Rectangularity (RECT)

Appendix E

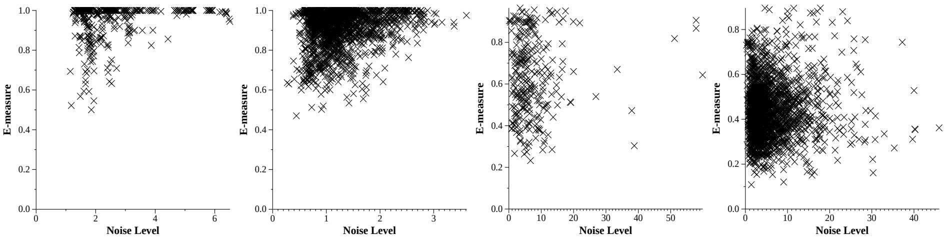
Results: Noise *vs* Performance

In this section we include the Noise *vs* Performance plots used in Chapter 6. As in the case of the similarity graphs we have changed the scaling of the graphs to enhance the boundaries of the clustering. Performance is calculated using the E-measure, proposed by Salton, as shown in Chapter 5.



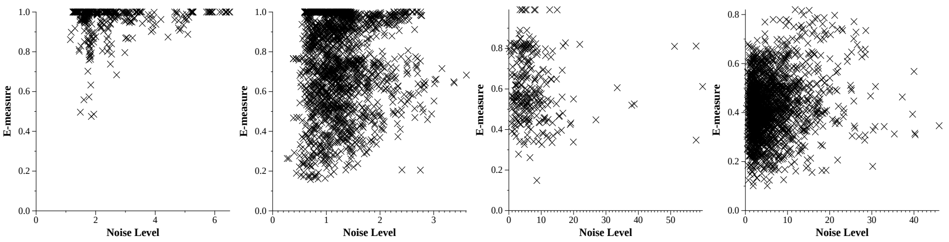
(a) $r = 0.160$ (b) $r = 0.198$ (c) $r = 0.069$ (d) $r = 0.044$

Figure E.1: Colour Labels (CL).



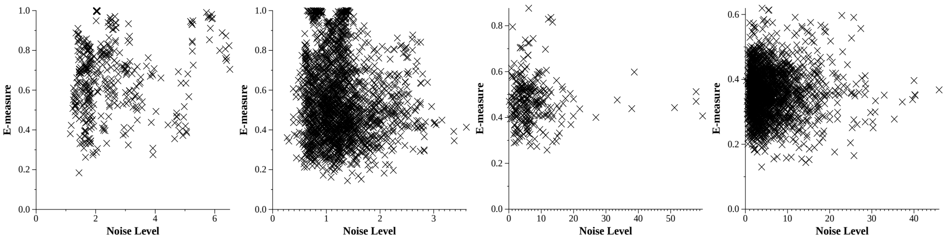
(a) $r = 0.202$ (b) $r = 0.176$ (c) $r = 0.069$ (d) $r = 0.059$

Figure E.2: Spatial Colour Labels (CL-SP).



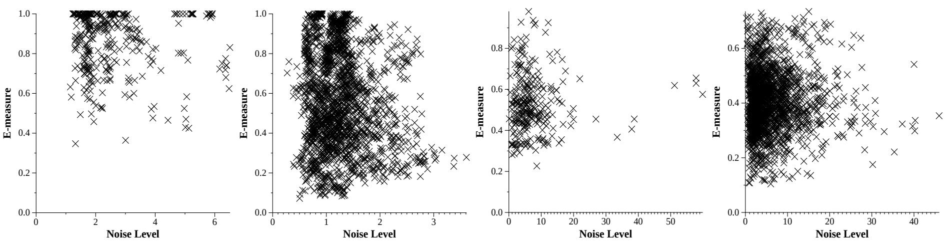
(a) $r = 0.134$ (b) $r = 0.038$ (c) $r = -0.062$ (d) $r = 0.192$

Figure E.3: Cooccurrence Matrix of intensities (COOC).



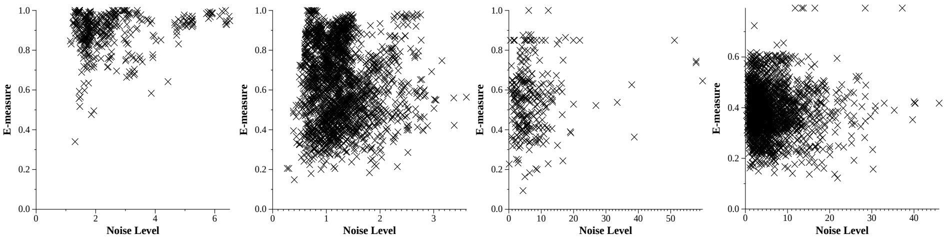
(a) $r = 0.168$ (b) $r = -0.079$ (c) $r = -0.015$ (d) $r = 0.013$

Figure E.4: Edge Distance (MSSDT).



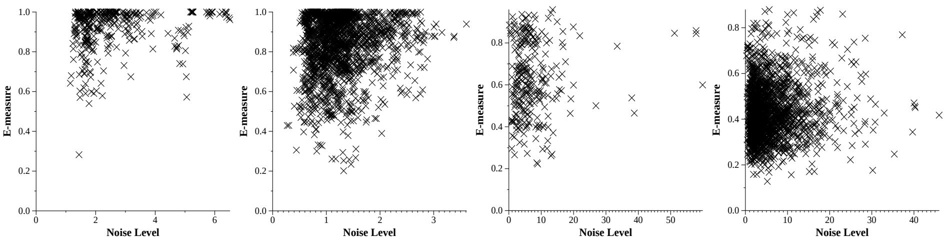
(a) $r = -0.092$ (b) $r = -0.138$ (c) $r = 0.030$ (d) $r = 0.028$

Figure E.5: Orientation (ORNT).



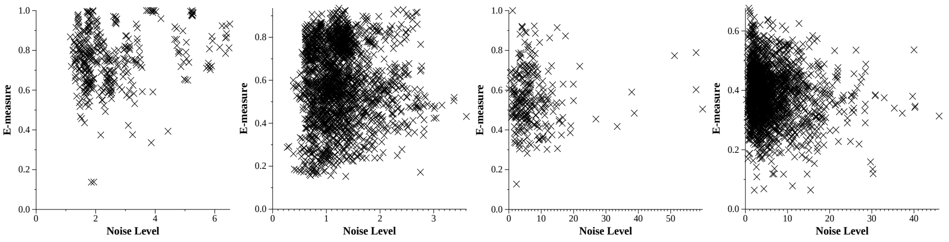
(a) $r = 0.166$ (b) $r = -0.044$ (c) $r = 0.126$ (d) $r = -0.008$

Figure E.6: Hue Entropy (HENT).



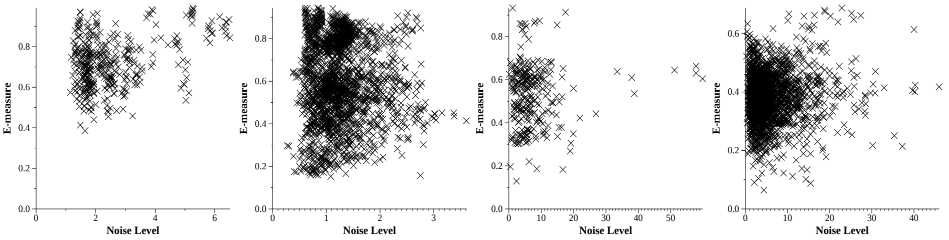
(a) $r = 0.166$ (b) $r = -0.044$ (c) $r = 0.126$ (d) $r = -0.008$

Figure E.7: Clsh (CLSH).



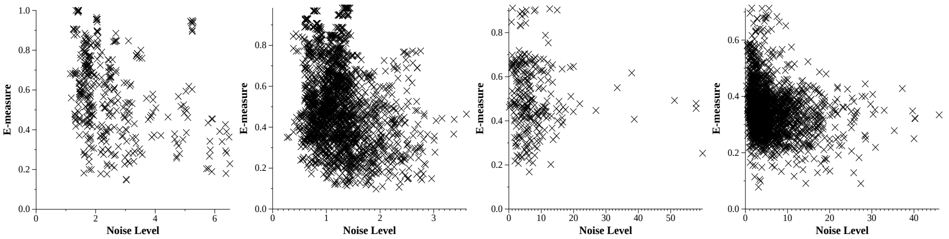
(a) $r = 0.152$ (b) $r = 0.099$ (c) $r = 0.071$ (d) $r = 0.091$

Figure E.8: Local Black (LBLK).



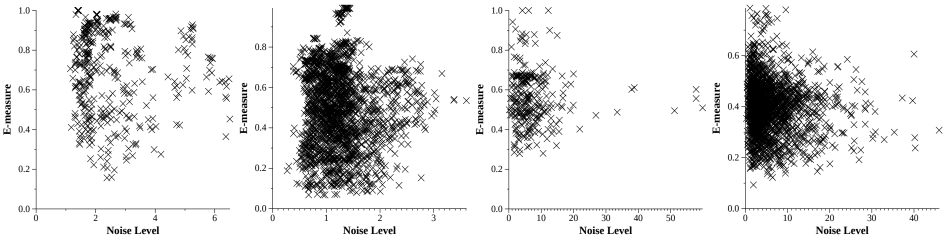
(a) $r = 0.167$ (b) $r = 0.015$ (c) $r = 0.015$ (d) $r = -0.113$

Figure E.9: Local Difference (LDIFF).



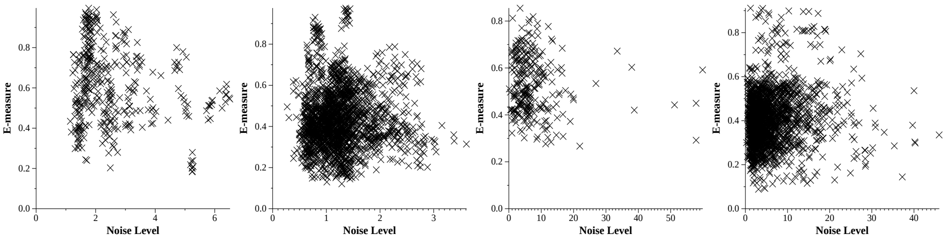
(a) $r = 0.342$ (b) $r = -0.023$ (c) $r = 0.056$ (d) $r = 0.080$

Figure E.10: BKDT (BKDT).



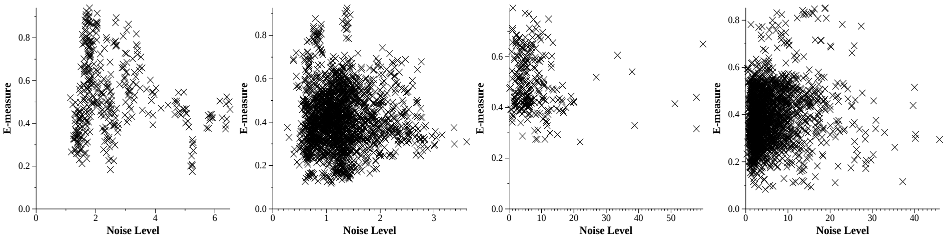
(a) $r = -0.311$ (b) $r = -0.266$ (c) $r = -0.084$ (d) $r = -0.201$

Figure E.11: BKDT2D (BKDT2D).



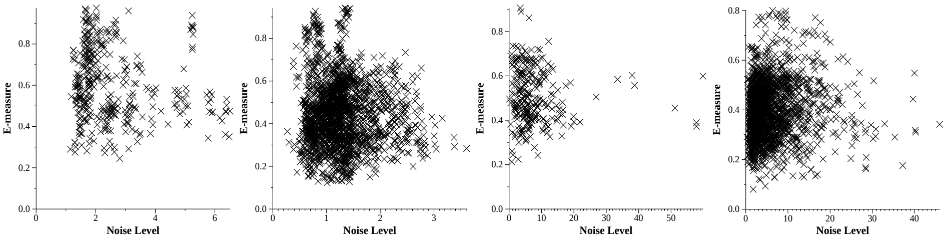
(a) $r = -0.068$ (b) $r = -0.036$ (c) $r = -0.038$ (d) $r = -0.103$

Figure E.12: BKDT2D (BKDT2D).



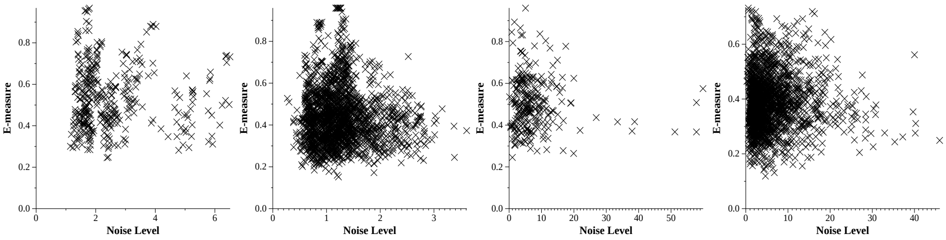
(a) $r = -0.068$ (b) $r = -0.036$ (c) $r = -0.038$ (d) $r = -0.103$

Figure E.13: Triangle Area (TARE).



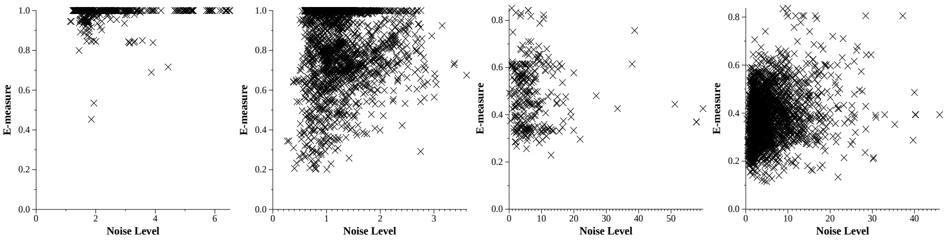
(a) $r = -0.175$ (b) $r = -0.014$ (c) $r = -0.139$ (d) $r = 0.117$

Figure E.14: Triangle Ratio (TRAT).



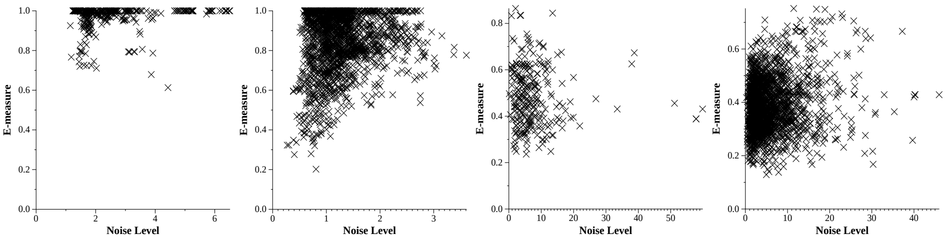
(a) $r = -0.166$ (b) $r = -0.053$ (c) $r = -0.082$ (d) $r = 0.099$

Figure E.15: Triangle Length (TLEN).



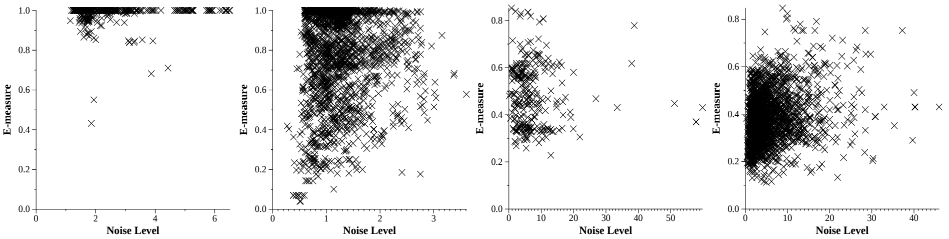
(a) $r = -0.009$ (b) $r = -0.056$ (c) $r = -0.073$ (d) $r = 0.005$

Figure E.16: Circular Arcs (CARC).



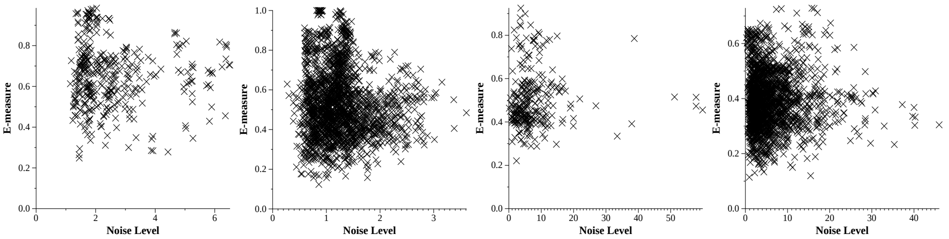
(a) $r = 0.076$ (b) $r = 0.043$ (c) $r = -0.064$ (d) $r = 0.216$

Figure E.17: Hue (HUE).



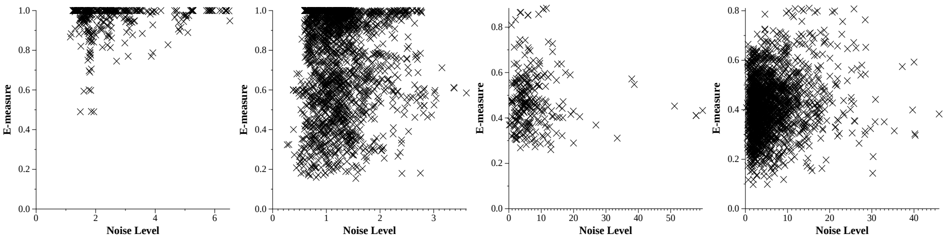
(a) $r = 0.116$ (b) $r = 0.120$ (c) $r = -0.064$ (d) $r = 0.169$

Figure E.18: Cooccurrence Matrix (Hue) (HCC).



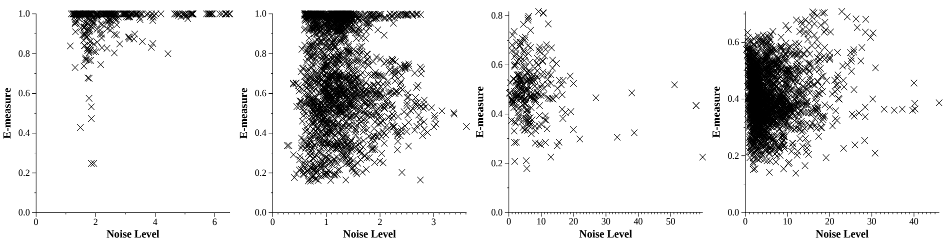
(a) $r = 0.054$ (b) $r = 0.045$ (c) $r = -0.056$ (d) $r = 0.227$

Figure E.19: Spatial Hue (HUE-SP).



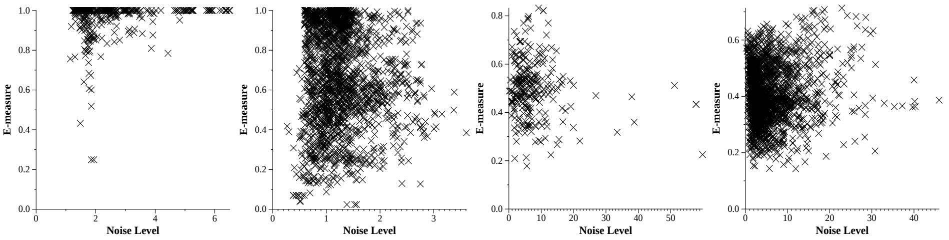
(a) $r = -0.063$ (b) $r = -0.093$ (c) $r = 0.038$ (d) $r = 0.009$

Figure E.20: K-Cos Edge Curvature (KCOS).



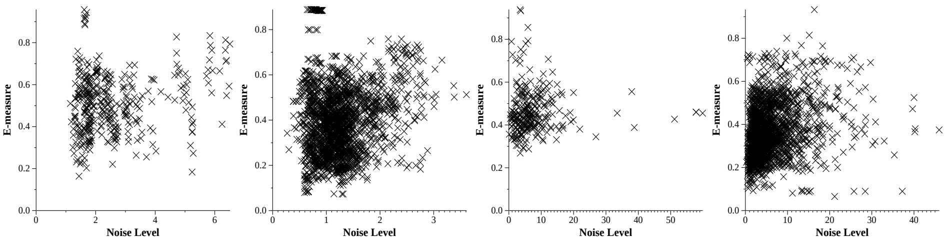
(a) $r = 0.117$ (b) $r = 0.036$ (c) $r = -0.041$ (d) $r = 0.164$

Figure E.21: Spatial Cooccurrence Matrix (CC-SP).



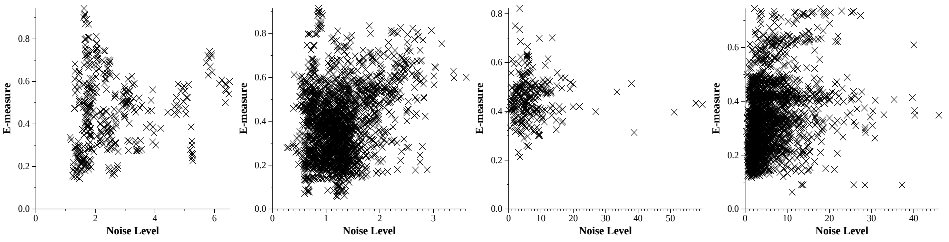
(a) $r = 0.171$ (b) $r = -0.061$ (c) $r = -0.165$ (d) $r = 0.095$

Figure E.22: T-Unit (TUNIT).



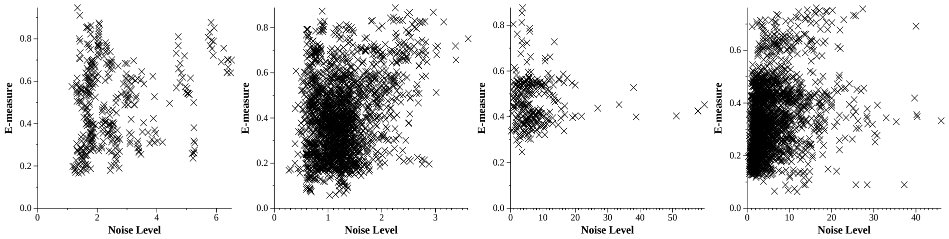
(a) $r = 0.177$ (b) $r = -0.046$ (c) $r = -0.156$ (d) $r = 0.095$

Figure E.23: Spatial T-Unit (TU-SP).



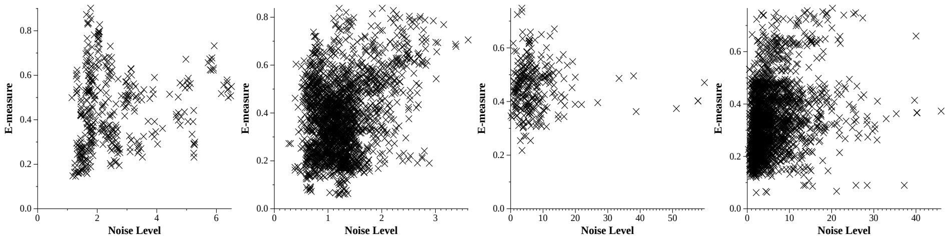
(a) $r = 0.178$ (b) $r = 0.163$ (c) $r = 0.009$ (d) $r = 0.269$

Figure E.24: Circularity (CIRC).



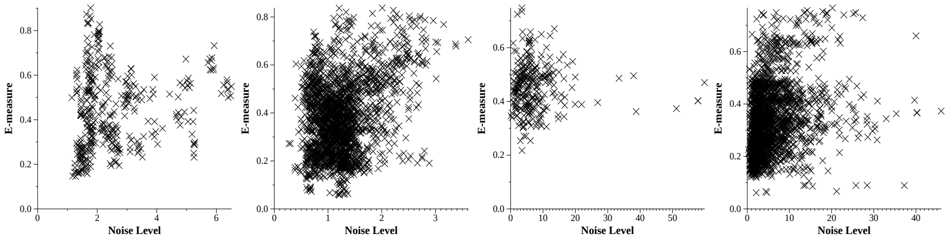
(a) $r = 0.135$ (b) $r = 0.288$ (c) $r = -0.039$ (d) $r = 0.266$

Figure E.25: Ellipticity (ELIP).



(a) $r = 0.195$ (b) $r = 0.291$ (c) $r = -0.044$ (d) $r = 0.278$

Figure E.26: Rectangularity (RECT).



(a) $r = 0.150$ (b) $r = 0.337$ (c) $r = -0.031$ (d) $r = 0.265$

Appendix F

Results: Statistical Tests Tables

The resulting tables of multiple comparisons, related to the statistical analysis are presented in this Appendix. The numbers of the first row and column indicate the number of combined histograms.

	2	3	4	5	6	7	8	9	10
2		0.978	**4.477**	**3.365**	0.583	0.063	1.197	2.291	**5.544**
3			**3.498**	**2.387**	0.395	0.915	2.175	**3.269**	**6.522**
4				1.111	**3.894**	**4.414**	**5.673**	**6.767**	**10.021**
5					**2.782**	**3.302**	**4.562**	**5.656**	**8.909**
6						0.520	1.780	**2.874**	**6.127**
7							1.260	**2.354**	**5.607**
8								1.094	**4.347**
9									**3.253**

	11	12	13	14	15	16	17	18	19
2	**6.320**	**8.661**	**8.975**	**8.858**	**8.368**	**13.926**	**14.271**	**13.994**	**14.780**
3	**7.298**	**9.639**	**9.953**	**9.836**	**9.346**	**14.905**	**15.249**	**14.973**	**15.758**
4	**10.797**	**13.138**	**13.452**	**13.335**	**12.845**	**18.403**	**18.748**	**18.471**	**19.257**
5	**9.685**	**12.026**	**12.340**	**12.223**	**11.733**	**17.292**	**17.636**	**17.360**	**18.145**
6	**6.903**	**9.244**	**9.558**	**9.441**	**8.951**	**14.510**	**14.854**	**14.578**	**15.363**
7	**6.383**	**8.724**	**9.038**	**8.921**	**8.431**	**13.989**	**14.334**	**14.058**	**14.843**
8	**5.123**	**7.464**	**7.778**	**7.661**	**7.171**	**12.730**	**13.074**	**12.798**	**13.583**
9	**4.029**	**6.370**	**6.684**	**6.567**	**6.077**	**11.636**	**11.980**	**11.704**	**12.489**
10	0.776	**3.117**	**3.431**	**3.314**	**2.824**	**8.383**	**8.727**	**8.451**	**9.236**
11		**2.341**	**2.655**	**2.538**	2.048	**7.606**	**7.951**	**7.674**	**8.460**
12			0.314	0.197	0.293	**5.265**	**5.610**	**5.334**	**6.119**
13				0.117	0.607	**4.951**	**5.296**	**5.020**	**5.805**
14					0.490	**5.069**	**5.413**	**5.137**	**5.922**
15						**5.558**	**5.903**	**5.627**	**6.412**
16							0.345	0.068	0.854
17								0.276	0.509
18									0.785

	20	21	22	23	24	25	26
2	**17.612**	**15.909**	**16.419**	**18.686**	**19.643**	**21.640**	**22.168**
3	**18.591**	**16.888**	**17.398**	**19.664**	**20.622**	**22.619**	**23.146**
4	**22.089**	**20.386**	**20.896**	**23.163**	**24.120**	**26.117**	**26.645**
5	**20.978**	**19.275**	**19.785**	**22.051**	**23.009**	**25.006**	**25.533**
6	**18.196**	**16.492**	**17.003**	**19.269**	**20.227**	**22.224**	**22.751**
7	**17.676**	**15.972**	**16.482**	**18.749**	**19.706**	**21.703**	**22.231**
8	**16.416**	**14.713**	**15.223**	**17.489**	**18.447**	**20.444**	**20.971**
9	**15.322**	**13.619**	**14.129**	**16.395**	**17.353**	**19.350**	**19.877**
10	**12.069**	**10.365**	**10.876**	**13.142**	**14.100**	**16.097**	**16.624**
11	**11.292**	**9.589**	**10.099**	**12.366**	**13.323**	**15.320**	**15.848**
12	**8.952**	**7.248**	**7.758**	**10.025**	**10.982**	**12.979**	**13.507**
13	**8.637**	**6.934**	**7.444**	**9.711**	**10.668**	**12.665**	**13.193**
14	**8.755**	**7.051**	**7.562**	**9.828**	**10.786**	**12.783**	**13.310**
15	**9.245**	**7.541**	**8.052**	**10.318**	**11.276**	**13.272**	**13.800**
16	**3.686**	1.983	**2.493**	**4.759**	**5.717**	**7.714**	**8.241**
17	**3.342**	1.638	2.148	**4.415**	**5.372**	**7.369**	**7.897**
18	**3.618**	1.915	**2.425**	**4.691**	**5.649**	**7.646**	**8.173**
19	**2.833**	1.129	1.640	**3.906**	**4.863**	**6.860**	**7.388**
20		1.703	1.193	1.073	2.031	**4.028**	**4.555**
21			0.510	**2.777**	**3.734**	**5.731**	**6.259**
22				2.266	**3.224**	**5.221**	**5.748**
23					0.958	**2.955**	**3.482**
24						1.997	**2.524**
25							0.527

Table F.1: COIL-100. Multiple comparisons of combinations utilising *all* the histograms.

	2	3	4	5	6	7	8
2		**6.226**	**5.451**	**3.298**	**4.512**	**3.457**	**2.468**
3			0.775	**2.928**	1.714	**2.769**	**3.758**
4				2.152	0.938	1.993	**2.983**
5					1.214	0.159	0.831
6						1.055	2.045
7							0.990

	9	10	11	12	13	14	15
2	0.684	1.202	**3.209**	**3.743**	**3.567**	**3.214**	2.101
3	**6.910**	**7.428**	**9.435**	**9.969**	**9.793**	**9.440**	**8.327**
4	**6.135**	**6.653**	**8.660**	**9.194**	**9.018**	**8.665**	**7.551**
5	**3.982**	**4.500**	**6.507**	**7.042**	**6.865**	**6.512**	**5.399**
6	**5.196**	**5.714**	**7.721**	**8.256**	**8.079**	**7.726**	**6.613**
7	**4.141**	**4.659**	**6.666**	**7.201**	**7.024**	**6.671**	**5.558**
8	**3.152**	**3.670**	**5.677**	**6.211**	**6.035**	**5.682**	**4.569**
9		0.518	**2.525**	**3.059**	**2.883**	**2.530**	1.417
10			2.007	**2.541**	**2.365**	2.012	0.899
11				0.534	0.358	0.005	1.108
12					0.176	0.529	1.642
13						0.353	1.466
14							1.113

	16	17	18	19	20	21	22
2	1.323	**7.453**	**8.838**	**9.185**	**12.032**	**12.948**	**14.176**
3	**7.549**	**13.679**	**15.064**	**15.411**	**18.258**	**19.174**	**20.402**
4	**6.773**	**12.904**	**14.289**	**14.635**	**17.482**	**18.399**	**19.627**
5	**4.621**	**10.751**	**12.136**	**12.483**	**15.330**	**16.247**	**17.474**
6	**5.835**	**11.965**	**13.350**	**13.697**	**16.544**	**17.461**	**18.688**
7	**4.780**	**10.910**	**12.295**	**12.642**	**15.489**	**16.406**	**17.633**
8	**3.790**	**9.921**	**11.306**	**11.652**	**14.499**	**15.416**	**16.644**
9	0.639	**6.769**	**8.154**	**8.501**	**11.348**	**12.264**	**13.492**
10	0.121	**6.251**	**7.636**	**7.983**	**10.830**	**11.746**	**12.974**
11	1.886	**4.244**	**5.629**	**5.976**	**8.823**	**9.739**	**10.967**
12	**2.421**	**3.710**	**5.095**	**5.441**	**8.289**	**9.205**	**10.433**
13	2.244	**3.886**	**5.271**	**5.618**	**8.465**	**9.381**	**10.609**
14	1.891	**4.239**	**5.624**	**5.971**	**8.818**	**9.734**	**10.962**
15	0.778	**5.352**	**6.737**	**7.084**	**9.931**	**10.847**	**12.075**
16		**6.130**	**7.515**	**7.862**	**10.709**	**11.626**	**12.853**
17			1.385	1.732	**4.579**	**5.495**	**6.723**
18				0.347	**3.194**	**4.110**	**5.338**
19					**2.847**	**3.764**	**4.991**
20						0.917	2.144
21							1.228

Table F.2: COIL-100. Multiple comparisons of combinations utilising *non-spatial only* the histograms.

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
2		1.004	**4.653**	**3.509**	0.579	0.051	1.244	**2.347**	**3.429**	**6.188**	**7.610**	**7.990**	**10.301**	**9.265**	**15.001**	**16.720**	**16.205**	**18.270**	**23.013**	**21.951**	**21.939**
3			**3.649**	**2.505**	0.425	0.953	2.248	**3.351**	**4.433**	**7.192**	**8.614**	**8.994**	**11.305**	**10.269**	**16.004**	**17.724**	**17.209**	**19.274**	**24.016**	**22.954**	**22.943**
4				1.144	**4.074**	**4.601**	**5.896**	**7.000**	**8.082**	**10.840**	**12.263**	**12.643**	**14.953**	**13.918**	**19.653**	**21.372**	**20.858**	**22.923**	**27.665**	**26.603**	**26.592**
5					**2.930**	**3.458**	**4.753**	**5.856**	**6.938**	**9.697**	**11.119**	**11.499**	**13.810**	**12.774**	**18.510**	**20.229**	**19.714**	**21.779**	**26.522**	**25.460**	**25.448**
6						0.528	1.823	**2.926**	**4.008**	**6.767**	**8.189**	**8.569**	**10.880**	**9.844**	**15.579**	**17.299**	**16.784**	**18.849**	**23.591**	**22.529**	**22.518**
7							1.295	**2.398**	**3.480**	**6.239**	**7.661**	**8.042**	**10.352**	**9.316**	**15.052**	**16.771**	**16.256**	**18.321**	**23.064**	**22.002**	**21.991**
8								1.103	2.185	**4.944**	**6.366**	**6.747**	**9.057**	**8.021**	**13.757**	**15.476**	**14.961**	**17.026**	**21.769**	**20.707**	**20.696**
9									1.082	**3.841**	**5.263**	**5.643**	**7.954**	**6.918**	**12.653**	**14.373**	**13.858**	**15.923**	**20.665**	**19.603**	**19.592**
10										**2.759**	**4.181**	**4.561**	**6.872**	**5.836**	**11.571**	**13.291**	**12.776**	**14.841**	**19.583**	**18.521**	**18.510**
11											1.422	1.803	**4.113**	**3.077**	**8.813**	**10.532**	**10.017**	**12.082**	**16.825**	**15.763**	**15.752**
12												0.380	**2.691**	1.655	**7.390**	**9.110**	**8.595**	**10.660**	**15.402**	**14.340**	**14.329**
13													2.310	1.275	**7.010**	**8.730**	**8.215**	**10.280**	**15.022**	**13.960**	**13.949**
14														1.036	**4.700**	**6.419**	**5.904**	**7.969**	**12.712**	**11.650**	**11.639**
15															**5.735**	**7.455**	**6.940**	**9.005**	**13.747**	**12.685**	**12.674**
16																1.719	1.205	**3.269**	**8.012**	**6.950**	**6.939**
17																	0.515	1.550	**6.293**	**5.231**	**5.219**
18																		2.065	**6.807**	**5.745**	**5.734**
19																			**4.742**	**3.681**	**3.669**
20																				1.062	1.073
21																					0.011

Table F.3: COIL-100. Multiple comparisons of combinations utilising *spatial only* the histograms.

	2	3	4	5	6	7	8	9	10
2		0.063	0.772	0.988	1.368	1.639	1.804	1.930	2.052
3			0.709	0.925	1.304	1.575	1.741	1.866	1.989
4				0.216	0.595	0.866	1.032	1.157	1.280
5					0.380	0.651	0.816	0.942	1.064
6						0.271	0.436	0.562	0.684
7							0.166	0.291	0.414
8								0.125	0.248
9									0.123

	11	12	13	14	15	16	17	18	19
2	2.210	2.203	2.155	2.009	1.771	1.711	1.491	1.227	0.837
3	2.147	2.140	2.092	1.946	1.707	1.648	1.428	1.164	0.773
4	1.438	1.431	1.383	1.237	0.998	0.939	0.719	0.455	0.065
5	1.222	1.215	1.167	1.021	0.783	0.723	0.503	0.239	0.151
6	0.843	0.835	0.787	0.641	0.403	0.343	0.123	0.141	0.531
7	0.572	0.565	0.517	0.371	0.132	0.073	0.148	0.412	0.802
8	0.406	0.399	0.351	0.205	0.033	0.093	0.313	0.577	0.967
9	0.281	0.274	0.226	0.080	0.159	0.218	0.439	0.703	1.093
10	0.158	0.151	0.103	0.043	0.281	0.341	0.561	0.825	1.215
11		0.007	0.055	0.201	0.440	0.499	0.719	0.983	1.374
12			0.048	0.194	0.432	0.492	0.712	0.976	1.366
13				0.146	0.384	0.444	0.664	0.928	1.318
14					0.239	0.298	0.518	0.782	1.172
15						0.060	0.280	0.544	0.934
16							0.220	0.484	0.874
17								0.264	0.654
18									0.390

	20	21	22	23	24	25	26
2	0.810	0.106	0.213	0.692	0.680	0.929	1.545
3	0.747	0.042	0.276	0.756	0.743	0.992	1.609
4	0.038	0.666	0.985	1.465	1.452	1.701	2.318
5	0.178	0.882	1.201	1.680	1.668	1.917	**2.533**
6	0.558	1.262	1.581	2.060	2.048	2.297	**2.913**
7	0.829	1.533	1.852	**2.331**	2.318	**2.568**	**3.184**
8	0.994	1.698	2.017	**2.497**	**2.484**	**2.733**	**3.350**
9	1.120	1.824	2.143	**2.622**	**2.609**	**2.859**	**3.475**
10	1.242	1.946	2.265	**2.744**	**2.732**	**2.981**	**3.598**
11	1.400	2.105	**2.423**	**2.903**	**2.890**	**3.139**	**3.756**
12	1.393	2.097	**2.416**	**2.896**	**2.883**	**3.132**	**3.749**
13	1.345	2.049	**2.368**	**2.848**	**2.835**	**3.084**	**3.701**
14	1.199	1.903	2.222	**2.702**	**2.689**	**2.938**	**3.555**
15	0.961	1.665	1.984	**2.463**	**2.450**	**2.700**	**3.316**
16	0.901	1.605	1.924	**2.404**	**2.391**	**2.640**	**3.257**
17	0.681	1.385	1.704	2.183	2.171	**2.420**	**3.036**
18	0.417	1.121	1.440	1.919	1.907	2.156	**2.772**
19	0.027	0.731	1.050	1.529	1.517	1.766	**2.382**
20		0.704	1.023	1.502	1.490	1.739	**2.355**
21			0.319	0.798	0.786	1.035	1.651
22				0.479	0.467	0.716	1.332
23					0.013	0.237	0.853
24						0.249	0.866
25							0.616

Table F.4: CHEN. Multiple comparisons of combinations utilising *all* the histograms.

	2	3	4	5	6	7	8
2		0.184	0.623	0.826	0.982	1.073	1.807
3			0.439	0.642	0.797	0.889	1.622
4				0.203	0.359	0.450	1.184
5					0.155	0.247	0.981
6						0.092	0.825
7							0.733

	9	10	11	12	13	14	15
2	1.880	1.934	1.875	1.720	1.423	1.181	0.800
3	1.695	1.749	1.691	1.536	1.239	0.997	0.616
4	1.257	1.311	1.252	1.097	0.800	0.558	0.177
5	1.054	1.107	1.049	0.894	0.597	0.355	0.026
6	0.898	0.952	0.894	0.738	0.442	0.200	0.181
7	0.806	0.860	0.802	0.647	0.350	0.108	0.273
8	0.073	0.127	0.069	0.087	0.383	0.626	1.006
9		0.054	0.005	0.160	0.456	0.699	1.079
10			0.058	0.214	0.510	0.752	1.133
11				0.155	0.452	0.694	1.075
12					0.297	0.539	0.920
13						0.242	0.623
14							0.381

	16	17	18	19	20	21	22
2	0.574	0.062	0.445	0.517	0.777	1.076	1.400
3	0.390	0.246	0.630	0.701	0.961	1.261	1.585
4	0.049	0.685	1.068	1.140	1.400	1.699	2.023
5	0.252	0.888	1.272	1.343	1.603	1.902	2.227
6	0.407	1.043	1.427	1.498	1.759	2.058	**2.382**
7	0.499	1.135	1.519	1.590	1.851	2.150	**2.474**
8	1.232	1.868	2.252	2.323	**2.584**	**2.883**	**3.207**
9	1.305	1.941	2.325	**2.396**	**2.657**	**2.956**	**3.280**
10	1.359	1.995	**2.379**	**2.450**	**2.711**	**3.010**	**3.334**
11	1.301	1.937	2.321	**2.392**	**2.652**	**2.951**	**3.276**
12	1.146	1.782	2.165	2.237	**2.497**	**2.796**	**3.121**
13	0.849	1.485	1.869	1.940	2.201	**2.500**	**2.824**
14	0.607	1.243	1.627	1.698	1.958	2.257	**2.582**
15	0.226	0.862	1.246	1.317	1.578	1.877	2.201
16		0.636	1.020	1.091	1.352	1.651	1.975
17			0.384	0.455	0.716	1.015	1.339
18				0.071	0.332	0.631	0.955
19					0.260	0.560	0.884
20						0.299	0.623
21							0.324

Table F.5: CHEN. Multiple comparisons of combinations utilising *non-spatial only* the histograms.

	2	3	4	5	6	7	8
2		0.843	0.871	1.465	1.837	1.901	1.848
3			0.028	0.622	0.994	1.058	1.004
4				0.594	0.966	1.030	0.976
5					0.372	0.437	0.383
6						0.064	0.010
7							0.054

	9	10	11	12	13	14	15
2	1.940	1.772	1.788	1.752	1.572	1.390	1.051
3	1.097	0.928	0.945	0.909	0.729	0.547	0.208
4	1.069	0.900	0.917	0.881	0.701	0.519	0.180
5	0.475	0.307	0.323	0.287	0.107	0.075	0.414
6	0.103	0.066	0.049	0.086	0.265	0.447	0.786
7	0.039	0.130	0.113	0.150	0.329	0.511	0.851
8	0.092	0.076	0.060	0.096	0.275	0.458	0.797
9		0.168	0.152	0.188	0.368	0.550	0.889
10			0.016	0.020	0.199	0.382	0.721
11				0.036	0.216	0.398	0.737
12					0.179	0.362	0.701
13						0.182	0.521
14							0.339

	16	17	18	19	20	21	22
2	0.631	0.336	0.234	0.707	0.673	0.690	1.711
3	0.212	0.508	1.077	1.550	1.516	1.533	**2.554**
4	0.240	0.536	1.105	1.578	1.544	1.561	**2.582**
5	0.833	1.129	1.699	2.172	2.138	2.155	**3.176**
6	1.206	1.502	2.071	**2.544**	**2.510**	**2.527**	**3.548**
7	1.270	1.566	2.135	**2.609**	**2.574**	**2.591**	**3.612**
8	1.216	1.512	2.081	**2.555**	**2.520**	**2.537**	**3.559**
9	1.309	1.604	2.174	**2.647**	**2.613**	**2.630**	**3.651**
10	1.140	1.436	2.005	**2.479**	**2.444**	**2.461**	**3.483**
11	1.157	1.452	2.022	**2.495**	**2.461**	**2.478**	**3.499**
12	1.120	1.416	1.985	**2.459**	**2.425**	**2.442**	**3.463**
13	0.941	1.237	1.806	2.279	2.245	2.262	**3.283**
14	0.759	1.054	1.624	2.097	2.063	2.080	**3.101**
15	0.419	0.715	1.285	1.758	1.724	1.741	**2.762**
16		0.296	0.865	1.339	1.304	1.321	**2.342**
17			0.569	1.043	1.008	1.025	2.047
18				0.473	0.439	0.456	1.477
19					0.034	0.017	1.004
20						0.017	1.038
21							1.021

Table F.6: CHEN. Multiple comparisons of combinations utilising *spatial only* the histograms.

	2	3	4	5	6	7	8	9	10
2		1.861	**5.886**	**6.776**	**7.689**	**8.340**	**8.693**	**8.639**	**8.582**
3			**4.025**	**4.916**	**5.828**	**6.479**	**6.832**	**6.779**	**6.721**
4				0.891	1.803	**2.454**	**2.807**	**2.754**	**2.696**
5					0.912	1.564	1.916	1.863	1.805
6						0.651	1.004	0.951	0.893
7							0.352	0.299	0.242
8								0.053	0.111
9									0.058

	11	12	13	14	15	16	17	18	19
2	**9.280**	**9.161**	**9.551**	**9.634**	**9.567**	**9.423**	**9.321**	**9.044**	**8.712**
3	**7.420**	**7.300**	**7.690**	**7.773**	**7.706**	**7.562**	**7.460**	**7.183**	**6.851**
4	**3.395**	**3.275**	**3.665**	**3.748**	**3.681**	**3.537**	**3.435**	**3.158**	**2.826**
5	**2.504**	**2.384**	**2.774**	**2.858**	**2.791**	**2.647**	**2.545**	2.267	1.935
6	1.592	1.472	1.862	1.945	1.878	1.734	1.632	1.355	1.023
7	0.940	0.821	1.210	1.294	1.227	1.083	0.981	0.704	0.371
8	0.588	0.468	0.858	0.941	0.874	0.730	0.628	0.351	0.019
9	0.641	0.522	0.911	0.995	0.928	0.784	0.682	0.404	0.072
10	0.699	0.579	0.969	1.052	0.985	0.841	0.739	0.462	0.130
11		0.120	0.270	0.354	0.286	0.143	0.041	0.237	0.569
12			0.390	0.473	0.406	0.262	0.160	0.117	0.449
13				0.083	0.016	0.128	0.230	0.507	0.839
14					0.067	0.211	0.313	0.590	0.922
15						0.144	0.246	0.523	0.855
16							0.102	0.379	0.711
17								0.277	0.609
18									0.332

	20	21	22	23	24	25	26
2	**8.504**	**8.540**	**8.312**	**8.115**	**7.707**	**7.218**	**6.947**
3	**6.643**	**6.679**	**6.451**	**6.254**	**5.846**	**5.358**	**5.086**
4	**2.618**	**2.654**	**2.426**	2.229	1.821	1.333	1.061
5	1.727	1.763	1.536	1.339	0.930	0.442	0.170
6	0.815	0.851	0.623	0.426	0.018	0.470	0.742
7	0.163	0.200	0.028	0.225	0.634	1.122	1.394
8	0.189	0.153	0.380	0.578	0.986	1.474	1.746
9	0.136	0.100	0.327	0.524	0.933	1.421	1.693
10	0.078	0.042	0.270	0.467	0.875	1.363	1.635
11	0.777	0.741	0.968	1.165	1.574	2.062	**2.334**
12	0.657	0.621	0.849	1.046	1.454	1.943	2.214
13	1.047	1.011	1.238	1.436	1.844	**2.332**	**2.604**
14	1.130	1.094	1.322	1.519	1.927	**2.416**	**2.687**
15	1.063	1.027	1.255	1.452	1.860	**2.349**	**2.620**
16	0.919	0.883	1.111	1.308	1.716	2.205	**2.476**
17	0.817	0.781	1.009	1.206	1.614	2.103	**2.374**
18	0.540	0.504	0.732	0.929	1.337	1.825	2.097
19	0.208	0.172	0.399	0.596	1.005	1.493	1.765
20		0.036	0.191	0.389	0.797	1.285	1.557
21			0.228	0.425	0.833	1.321	1.593
22				0.197	0.606	1.094	1.366
23					0.408	0.897	1.168
24						0.488	0.760
25							0.272

Table F.7: Artchive. Multiple comparisons of combinations utilising *all* the histograms.

	2	3	4	5	6	7	8
2		**4.034**	**5.665**	**6.713**	**7.327**	**7.732**	**7.593**
3			1.631	**2.679**	**3.292**	**3.698**	**3.558**
4				1.048	1.661	2.067	1.928
5					0.613	1.019	0.880
6						0.406	0.266
7							0.140
	9	10	11	12	13	14	15
2	**7.466**	**7.504**	**7.570**	**8.030**	**7.828**	**7.836**	**8.130**
3	**3.432**	**3.469**	**3.536**	**3.996**	**3.794**	**3.802**	**4.096**
4	1.801	1.838	1.905	**2.365**	2.163	2.171	**2.465**
5	0.753	0.790	0.857	1.317	1.115	1.123	1.417
6	0.139	0.177	0.244	0.704	0.501	0.510	0.803
7	0.266	0.229	0.162	0.298	0.096	0.104	0.398
8	0.127	0.089	0.023	0.438	0.235	0.244	0.537
9		0.038	0.104	0.565	0.362	0.371	0.664
10			0.067	0.527	0.324	0.333	0.626
11				0.460	0.258	0.266	0.560
12					0.203	0.194	0.099
13						0.008	0.302
14							0.293
	16	17	18	19	20	21	22
2	**8.510**	**8.576**	**8.359**	**8.003**	**7.843**	**6.776**	**5.114**
3	**4.475**	**4.542**	**4.324**	**3.969**	**3.809**	**2.742**	1.080
4	**2.845**	**2.911**	**2.694**	**2.338**	2.178	1.111	0.551
5	1.797	1.863	1.646	1.290	1.130	0.063	1.599
6	1.183	1.249	1.032	0.677	0.516	0.551	2.212
7	0.777	0.844	0.626	0.271	0.111	0.956	**2.618**
8	0.917	0.983	0.766	0.410	0.250	0.817	**2.478**
9	1.044	1.110	0.893	0.537	0.377	0.690	**2.351**
10	1.006	1.072	0.855	0.500	0.339	0.728	**2.389**
11	0.940	1.006	0.789	0.433	0.273	0.794	**2.456**
12	0.479	0.545	0.328	0.027	0.188	1.255	**2.916**
13	0.682	0.748	0.531	0.175	0.015	1.052	**2.713**
14	0.673	0.739	0.522	0.167	0.006	1.061	**2.722**
15	0.380	0.446	0.229	0.127	0.287	1.354	**3.015**
16		0.066	0.151	0.506	0.667	1.734	**3.395**
17			0.217	0.573	0.733	1.800	**3.461**
18				0.355	0.516	1.583	**3.244**
19					0.160	1.227	**2.889**
20						1.067	**2.728**
21							1.661

Table F.8: Artchive. Multiple comparisons of combinations utilising *spatial only* the histograms.

	2	3	4	5	6	7	8
2		1.921	**5.872**	**6.742**	**7.450**	**7.893**	**7.785**
3			**3.951**	**4.821**	**5.529**	**5.973**	**5.864**
4				0.870	1.578	2.022	1.914
5					0.708	1.152	1.044
6						0.444	0.336
7							0.108

	9	10	11	12	13	14	15
2	**8.146**	**8.208**	**8.020**	**7.722**	**7.525**	**7.127**	**6.767**
3	**6.225**	**6.287**	**6.099**	**5.801**	**5.604**	**5.206**	**4.846**
4	2.275	**2.336**	2.148	1.850	1.653	1.255	0.895
5	1.405	1.466	1.278	0.980	0.783	0.385	0.026
6	0.697	0.758	0.570	0.272	0.075	0.323	0.682
7	0.253	0.314	0.126	0.172	0.369	0.766	1.126
8	0.361	0.422	0.235	0.064	0.261	0.658	1.018
9		0.061	0.126	0.424	0.622	1.019	1.379
10			0.188	0.486	0.683	1.081	1.440
11				0.298	0.495	0.893	1.253
12					0.197	0.595	0.955
13						0.398	0.757
14							0.360

	16	17	18	19	20	21	22
2	**6.327**	**6.037**	**6.414**	**5.694**	**5.980**	**6.316**	**5.634**
3	**4.406**	**4.116**	**4.493**	**3.773**	**4.060**	**4.395**	**3.713**
4	0.455	0.165	0.542	0.178	0.109	0.444	0.237
5	0.415	0.705	0.328	1.048	0.761	0.425	1.107
6	1.123	1.413	1.036	1.756	1.469	1.134	1.815
7	1.567	1.857	1.479	2.200	1.913	1.577	2.259
8	1.458	1.749	1.371	2.091	1.805	1.469	2.151
9	1.819	2.110	1.732	**2.452**	2.166	1.830	**2.512**
10	1.881	2.171	1.794	**2.514**	2.227	1.891	**2.573**
11	1.693	1.983	1.606	**2.326**	2.039	1.704	**2.386**
12	1.395	1.685	1.308	2.028	1.741	1.406	2.087
13	1.198	1.488	1.111	1.831	1.544	1.208	1.890
14	0.800	1.090	0.713	1.433	1.147	0.811	1.493
15	0.440	0.731	0.353	1.073	0.787	0.451	1.133
16		0.290	0.087	0.633	0.346	0.011	0.693
17			0.377	0.343	0.056	0.280	0.402
18				0.720	0.434	0.098	0.780
19					0.287	0.622	0.060
20						0.336	0.346
21							0.682

Table F.9: Artchive. Multiple comparisons of combinations utilising *spatial only* the histograms.

Bibliography

[1] C. Achard, J. Devars, and L. Lacassagne. Object image retrieval with image compactness vectors. In *ICPR00*, page Vol IV: 1A, 2000.

[2] D.A. Adjeroh and M.C. Lee. On ratio-based color indexing. *IP*, 10(1):36–48, January 2001.

[3] S. Aksoy and R. Haralick. Textural features for image database retrieval. In *In Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries, in conjunction with CVPR'98*, pages 45–49, 1998.

[4] Selim Aksoy and Robert M. Haralick. Probabilistic vs. geometric similarity measures for image retrieval. In *Conf. Computer Vision and Pattern Recognition*, 2000.

[5] Y. A. Aslandogan, C. Thier, C. T. Yu, C. Liu, and K. R. Nair. Design, implementation and evaluation of SCORE (a system for COntent based REtrieval of pictures). In *International Conference on Data Engineering*, pages 280–287, 1995.

[6] B. Berlin and P. Kay. *Basic Color Terms: their universality and evolution.* University of California Press, 1969.

[7] Stefano Berreti, Alberto Del Bimbo, and Pietro Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Transactions on Multimedia*, 2(4):225–239, 2000.

[8] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, 34(3):344–371, 1986.

[9] J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, 8:679–698, 1986.

[10] C. Carson, M. Thomas, S. Belongie, J. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Visual99*, pages 509–516, 1999.

[11] S. Cha, Y. Shin, and S. Srihari. Algorithm for the edit distance between angular type histograms. Technical Report CEDAR-TR-99-1, 4 1999.

[12] S. Cha, Y. Shin, and S. Srihari. Distance between histograms of angular measurements and its application to handwritten character similarity. In *ICPR*, volume 2, page 2021, 2000.

[13] J.R and Beigi Chang, S-F and Smith. Visual Information Retrieval from Large Distributed On-line Repositories. In *Communications of the ACM 40*, number 12, 1997.

[14] S.K. Chang, Q.Y. Shi, and C.W. Yan. Iconic indexing by 2-D strings. *PAMI*, 9(3):413–428, 1987.

[15] C. Chen. Generalised similarity analysis and pathfinder network scaling. *Interacting with Computers*, 10(2):107–128, 1998.

[16] C. Chen, G. Gagaudakis, and P.L Rosin. Content-based image visualisation. In *IEEE Int. Conf. on Information Visualisation*, 2000.

[17] C. Chen, G. Gagaudakis, and P.L Rosin. Generalised similarity analysis revisited: Image visualisation. In *SSGRR, Int. Conf. on Advances on Infrastructure for Electronic Business, Science and Education on the Internet*, 2000.

[18] C. Chen, G. Gagaudakis, and P.L Rosin. Similarity-based image browsing. In *16th IFIP World Computer Congress (Int. Conf. on Intelligent Information Processing)*, pages 206–213, 2000.

[19] L. Clinque, F Lecca, S. Levialdi, and S. Tanimoto. Retrieval of images using rich-region descriptions. *Visual Languages and Computing*, 11:303–321, 2000.

[20] M.C. Cooper. The tractability of segmentation and scene analysis. *IJCV*, 30(1):27–42, 1998.

[21] J.M. Corridoni, A.Del Bimbo, and E. Vicario. Painting retrieval based on color semantics. *Image Databases and Multimedia Search*, 8:13–23, 1997.

[22] I.J. Cox, M.L Miller, S.M. Omohundro, and P.N. Yianilos. Pichunter: Bayesian relevance feedback for image retrieval. In *Proc. Int. Conf. on Pattern Recognition*, volume 3, pages 361–369, 1996.

[23] Mohamed Daoudi and Stanislaw Matusiak. Visual image retrieval by multiscale description of user sketches. *Visual Languages and Computing*, 11:287–301, 2000.

[24] A. Dimai. Rotation invariant texture description using general moment invariants and gabor filters. In *SCIA99*, page Pattern Recognition I, 1999.

[25] Alexander Dimai. Assessment of effectiveness of content based image retrieval systems. In *Visual Information and Information Systems*, pages 525–532, 1999.

[26] M. Dobie, R. Tansley, D. Joyce, M. Weal, P. Lewis, and W. Hall. A flexible architecture for content and concept based multimedia information exploration. In *Proc. Of the Challenge of Image Retrieval*, pages 1–12, 1999.

[27] Mark S. Drew, Jie Wei, and Ze-Nian Li. Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalized images. In *ICCV*, pages 533–540, 1998.

[28] Darren T. Drewry, Lin Gu, A. Benjamin Hocking, Kyoung-Don Kang, Robert C. Schutt, Christopher M. Taylor, and J Pfaltz. Current state of data mining. Technical Report UVa CS-2002-15, 5 2002.

[29] I.L. Dryden and K.V. Mardia. *Statistical Shape Analysis.* John Wiley and Sons, 1998.

[30] D. Dupplaw and P. H. Lewis. Content-based image retrieval with scale-spaced object trees. In M. M. Yeung, B.-L. Yeo, and C. A. Bouman, editors, *Proceedings of SPIE: Storage and Retrieval for Media Databases 2000*, volume 3972, pages 253–261, 2000.

[31] J.A. Feldman and Y. Yakimovsky. Decision theory and artificial intelligence: A semantic-based region analyser. *Artificial Intelligence*, 5:349–371, 1974.

[32] Graham D. Finlayson, Subho S. Chatterjee, and Brian V. Funt. Color angular indexing. In *ECCV (2)*, pages 16–27, 1996.

[33] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, 1995.

[34] David A. Forsyth, Jitendra Malik, Margaret M. Fleck, Hayit Greenspan, Thomas K. Leung, Serge Belongie, Chad Carson, and Chris Bregler. Finding pictures of objects in large collections of images. In *Object Representation in Computer Vision*, pages 335–360, 1996.

[35] Brian Funt, Kobus Barnard, and Lindsay Martin. Is machine colour constancy good enough? *Lecture Notes in Computer Science*, 1406, 1998.

[36] G. Gagaudakis and P.L Rosin. Incorporating shape into histograms for CBIR. In *10th British Machine Vision Conf.*, volume 35, pages 372–381, 1999.

[37] G. Gagaudakis and P.L Rosin. Shape measures for CBIR. In *ICIP*, 2001.

[38] G. Gagaudakis and P.L Rosin. Incorporating shape into histograms for CBIR. *Pattern Recognition*, 35:81–91, January 2002.

[39] G. Gagaudakis and P.L Rosin. Shape measures for CBIR. *To appear - Pattern Recognition Letters*, 2003.

[40] G. Gagaudakis, P.L Rosin, and C. Chen. Using CBIR and pathfinder networks for image database visualisation. In *15th Int. Conf. Pattern Recognition*, volume 1, pages 1052–1055, 2000.

[41] Th. Gevers. *Color Based Image Retrieval.* Springer Verlag, January 2001.

[42] L. Van Gool, P. Dewaele, and Oosterlinck. Texture analysis anno 1983. *Computer Vision, Graphics and Image Processing*, 29:336–357, 1985.

[43] W.I. Grosky and R. Zhao. From feature maps to semantic landscapes: Negotiating the semantic gap. *To Appear in Proc. of the First Int. Workshop on Multimedia Annotation, Tokyo, Japan*, Jan. 2001.

[44] William I. Grosky and Rong Zhao. Negotiating the semantic gap: From feature maps to semantic landscapes. *Lecture Notes in Computer Science*, 2234, 2001.

[45] E.M. Gurari and H. Wechsler. On the difficulties involved in the segmentation of pictures. *IEEE Transaction PAMI*, 4(3):304–306, 1982.

[46] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.

[47] Dong-Chen He and Li Wang. Texture unit, texture spectrum and texture analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 28(4):509–512, 1990.

[48] G. Healey and D. Slater. Global color constancy: Recognition of objects by use of illumination-invariant properties of color distribution. *Optical Society of America*, 11(11):3003–3010, 1994.

[49] F.S. Hillier and G.J. Lieberman. *Introduction to Mathematical Programming.* McGraw-Hill, 1990.

[50] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR97*, pages 762–768, 1997.

[51] J. Immerkaer. Fast noise variance-estimation. *CVIU*, 64(2):300–302, September 1996.

[52] A.K. Jain and A. Vailaya. Image Retrieval Using Color and Shape. *Pattern Recognition*, 29(8):1233–1244, 1996.

[53] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics and Image Processing*, 29(3):273–285, 1985.

[54] E. Kawaguchi and R.I. Taniguchi. The depth first picture-expression as an image thresholding strategy. *IEEE Trans. on Sys., Man, and Cybernetics*, 19(5):1321–1329, 1989.

[55] H.K. Kim, J.D. Kim, D.G. Sim, and D.I. Oh. A modified zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval. In *ICME00*, page MP5, 2000.

[56] Mineichi Kudo, Hideyuki Imai, and Masaru Shimbo. A histogram-based classifier on overlapped bins. In *ICPR 2000*, volume 2, page 2029, 2000.

[57] Wee Kheng Leow and Risto Miikkulainen. Visual schemas in neural networks for object recognition and scene analysis. *Connection Science*, 9(2):161–200, 1997.

[58] W.Y. Ma and H. Zhang. Benchmarking of image features for content-based retrieval. In *32nd Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 253–257, 1998.

[59] J. Malo, A.M. Pons, and J.M. Artigas. Subjective image fidelity metric based on bit allocation of the human visual-system in the DCT domain. *IVC*, 15(7):535–548, July 1997.

[60] M. Mandal, T. Aboulnasr, and S. Panchanathan. Image indexing using moments and wavelets. *IEEE Transactions on Consumer Electronics*, 42(3), 1996.

[61] Jean Michel Marie-Julie and Hassane Essafi. Digital image indexing and retrieval by content using the fractal transform for multimedia databases. In *4th International Forum on Research and Technology Advances in Digital Libraries*, 1997.

[62] Aleix Martinez and Joan R. Serra. An approach to object-related image retrieval. *Journal of Visual Languages and Computing*, 11:345–363, 2000.

[63] J. Matas, R. Marik, and J. Kittler. The color adjacency graph representation of multi-coloured objects, 1995.

[64] Babu M. Mehtre, Mohan S. Kankanhalli, and Wing Foon Lee. Shape measures for content based image retrieval: A comparison. *Information Processing and Management*, 33(3):319–337, 1997.

[65] R. Milanese and M. Cherbuliez. A rotation, translation, and scale-invariant approach to content-based image retrieval. *Journal of Visual Communication and Image Representation*, 10:186–196, 1999.

[66] Thomas Minka. An image database browser that learns from user interaction. Master's thesis, Cambridge, MA, 1996.

[67] T.P. Minka and R.W. Picard. Interactive learning with a society of models. In *Vismod*, 1996.

[68] H. Müller, W. Müller, S. Marchand-Maillet, T. Pun, and D.M. Squire. Automated benchmarking in content-based image retrieval. In *IEEE International Conference on Multimedia and Expo*, 2001.

[69] H. Müller, W. Müller, S. Marchand-Maillet, T. Pun, and D.M. Squire. Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letters - special issue on Image/Video Indexing and Retrieval*, 22(5):593–601, 2001.

[70] Aleksandra Mojsilovic and Bernice Rogowitz. Capturing image semantics with low-level descriptors. In *ICIP 2001*, 2001.

[71] G. Mori, S. Belongie, and H. Malik. Shape contexts enable efficient retrieval of similar shapes. In *CVPR*, pages 723–730, 2001.

[72] E.N. Mortensen and W.A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384, 1998.

[73] A. D. Narasimhalu, M. S. Kankanhalli, and J. Wu. Benchmarking multimedia databases. *Multimedia Tools and Applications*, 4:333–356, 1997.

[74] C. Nastar and M. Mitschke. Real-time face recognition using feature combination. In *AFGR98*, pages 312–317, 1998.

[75] Chahab Nastar. The image shape spectrum for image retrieval. Technical Report RR-3206, 1997.

[76] S. A. Nene, Nayar S. K., and H. Murase. Columbia object image library (coil-100). In *Technical Report CUCS-006-96*, 1996.

[77] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. on Sys., Man, and Cybernetics*, 9:62–66, 1979.

[78] G. Pass and R. Zabih. Histogram refinement for content-based image retrieval. In *WACV96*, pages 96–102, 1996.

[79] Greg Pass and Ramin Zabih. Comparing images using joint histograms. *Multimedia Systems*, 7(3):234–240, 1999.

[80] Mark J. Paulik, Manohar Das, and N. K. Loh. Nonstationary autoregressive modeling of object conturs. *IEEE Transactions on Signal Processing*, 40(3):660–675, 1992.

[81] M. Peura and J. Iivarinen. Efficiency of simple shape descriptors. In C. Arcelli *et al.*, editor, *Aspects of Visual Form Processing*. World Scientific, 1997.

[82] A. Charles Poynton. Frequently Asked Quenstions about Color. http://ftp.iasi.roedu.net/mirrors/ftp.gimp.org/gimp/docs/color/color.faq.pdf.

[83] F.P. Preparata and M.I. Shamos. *Computational Geometry*. Springer-Verlag, 1985.

[84] J. Puzicha, T. Hofmann, and J. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Conference on Computer Vision and Pattern Recognition*, 1997.

[85] A. Rao, R. Srihari, and Z Zhang. Spatial color histograms for content-based image retrieval. In *IEEE International Conference on Tools with Artificial Intelligence*, pages 183–186, 1999.

[86] A. Rao, R. Srihari, and Z. Zhang. Geometric histogram: A distribution of geometric configurations of color subsets. In *SPIE: Internet Imaging*, 2000.

[87] S. Ravela and R. Manmatha. Retrieving images by appearance. In *ICCV98*, pages 608–613, 1998.

[88] A. Rosenfeld and E. Johnston. Angle detection on digital curves. *TC*, 22:875–878, 1973.

[89] P. L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Trans. PAMI*, 19(6):659–666, 1997.

[90] P.L. Rosin. Measuring shape: Ellipticity, rectangularity, and triangularity. In *ICPR00*, pages Vol I: 952–955, 2000.

[91] P.L. Rosin and G.A.W. West. Nonparametric segmentation of curves into various representations. *PAMI*, 17(12):1140–1153, December 1995.

[92] P.L. Rosin and G.A.W. West. Salience distance transforms. *GMIP*, 57(6):483–521, November 1995.

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

[94] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Relevance feedback techniques in interactive content-based image retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 25–36, 1998.

[95] Maytham Safar, Cyrus Shahabi, and Xiaoming Sun. Image retrieval by shape: A comparative study. In *Proceedings of international conference on Multimedia and Exposition*, pages xx–yy, 2000.

[96] P.K. Sahoo, S. Soltani, A.K.C. Wong, and Y.C. Chen. A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41(2):233–260, 1988.

[97] G. Salton and M.J. McGill. *Introduction to modern information retrieval.* McGraw-Hill, 1983.

[98] Simone Santini. Evaluation vademecum for visual information systems. In *Proceedings of SPIE , Storage and Retrieval for Image and Video Databases VIII*, volume 3972, 2000.

[99] Simone Santini, Amarnath Gupta, and Ramesh Jain. Emergent semantics through interaction in image databases. *Knowledge and Data Engineering*, 13(3):337–351, 2001.

[100] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *PAMI*, 19(5):530–535, 1997.

[101] M. Seaborn, L. Hepplewhite, and J. Stonham. Fuzzy colour category map for content based image retrieval. In *BMVC99*, page Image Matching and Retrieval, 1999.

[102] Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. Alignment-based recognition of shape outlines. *Lecture Notes in Computer Science*, 2059, 2001.

[103] I. Sethi, I. Coman, B. Day, F. Jiang, D. Li, J. Segovia-Juarez, G. Wei, and B. You. Color-wise: A system for image similarity retrieval using color. In *Proceedings of*

SPIE Storage and Retrieval for Image and Video Databases, volume 3312, pages 140–149, 1998.

[104] J. R. Jr. Siegel, Sidney and Castellan. *Nonparametric statistics for the Behavioural Science, 2nd Edition.* McGraw-Hill Book Co., 2000.

[105] David Sinclair. Image parsing for image retrieval from large image data bases: From coloured image to coloured regions. Technical Report ORL Technical Report 97.4, 1997.

[106] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R.C. Jain. Content-based image retrieval at the end of the early years. *PAMI*, 22(12):1349–1380, December 2000.

[107] J.R. Smith and S-F Chang. Single color extraction and image query. In *ICIP*, 1995.

[108] J.R. Smith and C.S. Li. Image classification and querying using composite region templates. *CVIU*, 75(1/2):165–174, 1999.

[109] M. Sonka and R. Hlanac, V. Boyle. *Image Processing Analysis and Machine Vision (Second Edition).* PWS Publishing, 1999.

[110] David McG. Squire. Learning a similarity-based distance measure for image database organization from human partitionings of an image set. In *Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision (WACV'98)*, pages 88–93, 1998.

[111] D.M. Squire and T. Pun. Assessing agreement between human and machine clusterings of image databases. *PR*, 31(12):1905–1919, December 1998.

[112] M. Stricker and A. Dimai. Spectral covariance and fuzzy regions for image indexing. *MVA*, 10(2):66–73, 1997.

[113] Markus A. Stricker and Markus Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.

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

[115] R. Tansley. The multimedia thesaurus: An aid for multimedia information retrieval and navigation, 1998.

[116] R. Tansley, C. Bird, W. Hall, P. Lewis, and M Weal. Automating the linking of content and concept. In *ACM Multimedia*, pages 445–447, 2000.

[117] Y. Tao and W. Grosky. Object-based image retrieval using point feature maps proc. In *Proc. of the International Conference on Database Semantics (DS-8)*, pages 59–73, 1999.

[118] Yi Tao and William I. Grosky. Spatial color indexing: A novel approach for content-based image retrieval. In *ICMCS, Vol. 1*, pages 530–535, 1999.

[119] Leonid Taycher, Marco La Cascia, and Stan Sclaroff. Image digestion and relevance feedback in the imagerover WWW search engine. Technical Report 1997-014, 14, 1997.

[120] W.H. Tsai. Moment-preserving thresholding. *Computer Vision, Graphics and Image Processing*, 29:377–393, 1985.

[121] R. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Utrecht University, the Netherlands, 1999.

[122] M. P. Wand. Data-based choice of histogram bin width. *The American Statistician*, 51(1):59–??, 1997.

[123] G.A.W. West and P.L. Rosin. Techniques for segmenting image curves into meaningful descriptions. *PR*, 24(7):643–652, 1991.

[124] C. Wolf, J. Jolion, W. Kropatsch, and H. Bischof. Content based image retrieval using interest points and texture features. In *International Conference Pattern Recognition*, page Vol IV: 1A, 2000.

[125] Esprit working group 20039. Mira: Evaluation frameworks for interactive multimedia retrieval applications. *http://www.dcs.gla.ac.uk/mira/*, 1996.