Textural Features For Content-Based Image Database Retrieval

by

Selim Aksoy

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science in Electrical Engineering

University of Washington

1998

Approved by_____

(Chairperson of Supervisory Committee)

_____

_____

Program Authorized
to Offer Degree_____

Date_____

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Any other reproduction for any purposes or by any means shall not be allowed without my written permission.

Signature_____

Date_____

University of Washington

Abstract

Textural Features For Content-Based Image Database Retrieval

by Selim Aksoy

Chairperson of Supervisory Committee
Professor Robert M. Haralick
Department of Electrical Engineering

Image database retrieval has received significant attention in recent years. This thesis describes a system to retrieve all database images having some section similar to the query image. We develop efficient features for image representation and effective metrics that use these representations to establish similarity between images. The first set of features we use are the line-angle-ratio statistics constituted by 2-D texture histograms of the angles between intersecting/near-intersecting lines and the ratios of mean gray levels inside and outside the regions spanned by those angles. A line selection algorithm using hypothesis testing is developed to eliminate insignificant lines. The second set of features used are the variances of gray level spatial dependencies computed from co-occurrence matrices at different distances and orientations. Statistical feature selection methods are used to select the parameters of the feature extraction algorithms. We also combine these macro and micro texture features to make use of their different advantages.

We define two classes, the relevance class and the irrelevance class, and design an automatic groundtruth construction protocol to associate each image pair with one of the classes. To rank-order the database images according to their similari-

ties to the query image, a likelihood ratio and the k-nearest neighbor rule are used. To evaluate the performance, classification effectiveness and retrieval performance experiments are done on a large database with many different kinds of complex images. More than 450,000 image pair classifications using a Gaussian classifier and a nearest neighbor classifier showed that approximately 80% of the relevance class groundtruth pairs were assigned to the relevance class correctly. To compensate for the effects of the mislabeling probabilities of the groundtruth construction protocol, we develop a statistical framework that estimates the correct classification results. Hence, some of the assignments which we count as incorrect are not in fact incorrect. In the retrieval performance tests, which use more than 300,000 queries, we observed that combined feature sets and the likelihood ratio distance measure outperformed all the other feature and distance measure combinations we use. The results show that low-level textural features can help in grouping images into semantically meaningful categories, and multi-scale texture representation provides a significant improvement in both classification effectiveness and retrieval performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my advisor Prof. Robert M. Haralick for his support and guidance throughout my study. His approaches to research problems have been an invaluable experience for me.

I am very thankful to Prof. Linda G. Shapiro for her valuable comments, help and advice at every stage of my work. I also thank Prof. Jenq-Neng Hwang for being in my advisory committee and for his suggestions about the draft versions of my thesis.

The Intelligent Systems Laboratory has been a very pleasant place to work. I would like to thank all my colleagues, especially Mike Schauf for his partnership in both research and system administration, and Gang Liu, Qiang Ji, Desikachari Nadadur and Jisheng Liang for their help and valuable discussions. I am also thankful to Gokhan Sahin for his help and support.

I am also grateful to my professors at the Middle East Technical University, Turkey, for the background and motivation they have given me.

Finally, I would like to thank my family for their endless love and support.

Chapter 1

INTRODUCTION

Image databases are becoming increasingly popular due to large amount of images that are generated by various applications and the advances in computation power, storage devices like CD-ROM, scanning, networking, image compression, desktop publishing and the World Wide Web. Because of this popularity, image database research has become a very hot area. The advances in this area contribute to an increase in the number, size, use, and availability of on-line image databases and new tools are required to help users create, manage, and retrieve images from these databases. The value of these systems can greatly increase if they can provide the ability of searching directly on non-textual data, "content" of the image, instead of searching only on the associated textual information. Main purpose of a content-based image database retrieval system is to effectively and efficiently use the information stored in the database.

1.1 Image Databases

In a typical content-based image database retrieval application, the user has an image and/or just a subject he or she is interested in and wants to find images from the database that are similar to the example image and/or related to the subject. For example, a fashion designer needs images of fabrics with a particular mixture of colors, a museum cataloger looks for artifacts of a particular shape and textured pattern and a movie producer needs a video clip of a red car moving from right to

left with the camera zooming. Other application areas can be architectural and engineering design, interior design, remote sensing and management of earth resources, geographic information systems, scientific database management, weather forecasting, retailing, trademark and copyright database management, law enforcement, criminal investigation, picture archiving and communication systems [24].

Conventional database retrieval methods will not be sufficient to retrieve this kind of data because they depend on file IDs, keywords, or text associated with the images. They do not allow queries based directly on the visual properties of the images, they depend on the particular vocabulary used, and they do not provide queries for images "similar" to a given image. In conventional databases, retrieval is based on an exact match of the attribute values so they do not have the ability to rank-order results by the degree of similarity with the query image. There is an old saying "A picture is worth a thousand words." [47]. Unfortunately, it is impossible to represent the content of an image in a few words. For example, an image annotated as containing "woman" and "children" cannot be retrieved by a query searching for the keyword "people".

Establishing "similarity" between two images is a very hard and abstract concept. At first glance, content-based retrieval seems as if it should be very simple because humans are so good at it. Also, since we have an almost perfect text-search technology, it will be very easy to retrieve images if we can assign semantic descriptions to them. Unfortunately, assigning semantic descriptions is an unsolved problem in image understanding. In the literature, approaches to content-based retrieval have taken two directions [24]. In the first, image contents are modeled as a set of attributes extracted manually and managed within a conventional database management system. Queries are specified using these attributes. This attribute-based retrieval is advanced primarily by database researchers.

The second approach is to apply a feature-extraction and/or object-recognition algorithm to automate the feature-extraction and object-recognition tasks that need to

be done when the image is inserted into the database. This approach is advanced primarily by image understanding and computer vision researchers. The main goal is to combine ideas from areas such as knowledge-based systems, cognitive science, modeling, computer graphics, image processing, pattern recognition, database-management systems and information retrieval. Ideally, object recognition should be automatic, but this is generally difficult. The alternative manual identification is almost infeasible and also inhibits the query-by-content idea.

After images are added to the database and features are extracted, queries can be formed to allow users retrieve images. Researchers have used different distance measures to compute the similarity between two images. The idea is to find an image which has the most similar features to the features extracted from the query image. This similarity between two features is computed by measuring the closeness of the two using distance metrics.

Queries for a content-based image database retrieval system can be based on different features and can be from different classes like color, texture, sketch, shape, volume, spatial constraints, browsing (interactive), objective attributes, subjective attributes, motion, text, and domain concepts [24]. Color queries let users retrieve images containing specific colors as input by the user. The user can specify percentages and locations of colors in the image. Texture queries allow retrieving images containing a specific texture. Retrieval by sketch lets users outline an image and then retrieves a similar image from the database. The spatial constraints category deals with a class of queries based on spatial and topological relationships among the objects in an image. These relationships may span a broad spectrum ranging from directional relationships to adjacency, overlap, and containment involving a pair of objects or multiple objects. Retrieval by browsing (interactive retrieval) is performed when users are not exactly clear about their retrieval needs or are unfamiliar with the structure and types of information available in the image database. All of these queries can be formed either using some predefined options or using another image,

which is also called query-by-example.

These queries should be integrated with a graphical user interface for easier access. Ideally, a natural language understanding tool will be the most user friendly interface. For example, it is easier to express a query such as "Show me images of snow-covered mountains" in natural language than it is to sketch an image of a mountain and sprinkle it with snow texture.

In practice, there are also other issues like indexing problems in very large databases [9, 10, 11] and user interaction [44, 48]. Barros *et al.* [9] investigated the effect of triangle-inequality using single keys and pairs of keys in reducing the number of comparisons to search the database. Berman and Shapiro [10] used polynomial combinations of predefined distance measures to create new distance measures and extended the triangle-inequality to compute lower bounds for these new measures to prune the database. In [11] they investigated the performance of different key selection algorithms like random selection, selection according to density variance, selection according to separation, a greedy thresholding algorithm and clustering. Minka and Picard [44] used machine learning in terms of self organizing feature maps to automatically select and combine available features based on positive and negative examples from the user. Picard and Pentland [48] addressed the need for having a human in an interactive loop with the system and designing systems that can infer which features are the most relevant for a search guided by user's examples.

In this thesis we will concentrate on feature extraction methods, specifically textural features, for image representation, and on decision methods to establish similarity between these representations. In the following section we discuss some of the previous work done on texture analysis and its use in content-based image retrieval.

1.2 Literature Overview

1.2.1 Textural Features

Texture has been one of the most important characteristics which have been used to classify and recognize objects and scenes. It can be characterized by textural primitives as unit elements and neighborhoods in which the organization and relationships between the properties of these primitives are defined. Numerous methods, that were designed for a particular application, have been proposed in the literature. However, there seems to be no general method or a formal approach which is useful in a broad range of images.

Haralick and Shapiro [28] defined texture as the uniformity, density, coarseness, roughness, regularity, intensity and directionality of discrete tonal features and their spatial relationships. Although no generally applicable definition of texture exists, some common elements in the definitions found in the literature are primitives and/or properties that are defined in a neighborhood and the statistical and/or structural relationships between these primitives and/or properties that are measured at a scale of interest.

In his texture survey, Haralick [26] characterized texture as a concept of two dimensions, the tonal primitive properties and the spatial relationships between them. He pointed out that tone and texture are not independent concepts, but in some images tone is the dominating one and in others texture dominates. Then, he gave a review of two kinds of approaches to characterize and measure texture: *statistical* approaches like autocorrelation functions, optical transforms, digital transforms, textural edgeness, structuring elements, spatial gray level run lengths and autoregressive models, and *structural* approaches that use the idea that textures are made up of primitives appearing in a near-regular repetitive arrangement.

Rosenfeld and Troy [50] also defined texture as a repetitive arrangement of a unit pattern over a given area and tried to measure coarseness of texture using amount of

edge per unit area, gray level dependencies, autocorrelation, and number of relative extrema per unit area.

Rosenfeld [49] reviewed some of the texture measures in the literature; autocorrelation, power spectrum, second-order gray level statistics, first-order local feature statistics (features like edges and straight lines) and texture segmentation using local features.

Laws [36] filtered the image using 5×5 kernels and then applied a non-linear moving-window averaging operation to compute the texture energy in a neighborhood. These 5×5 kernels are constructed using outer products of five 1-D kernels, which he called level, edge, spot, wave and ripple, each of length 5. He used these texture energy values with a linear discriminator to classify pixels into different texture classes.

Mao and Jain [43] modeled texture in terms of the parameters of a simultaneous autoregressive model (SAR) fit. First, a rotation invariant SAR model is introduced by taking the gray level samples at a circular grid. Then, a multiresolution SAR model (MR-SAR) is developed to overcome the problems in choosing the neighborhood size in which the pixel gray levels are regarded as independent, and selecting the window size in which texture is regarded as being homogeneous and the parameters of the SAR model are estimated. MR-SAR performed better than single resolution SAR in both texture classification and texture segmentation.

A more recent texture survey was done by Tuceryan and Jain [57], where they reviewed the basic concepts and various methods for texture processing. They summarized the applications of texture as texture classification (recognition of image regions), texture segmentation (finding texture boundaries), texture synthesis (generation of images for special purposes) and shape from texture (recognition of shape from the distortion in texture elements). They argued that texture perception and analysis are motivated from two viewpoints; psychophysics, which motivated the use of first-order and second-order statistics and also multiresolution analysis, and ma-

chine vision applications, which include industrial inspection, medical image analysis, document processing and remote sensing. They classified the texture models into statistical methods (co-occurrence matrices, autocorrelation features, etc.), geometrical methods (Voronoi tessellation features, structural methods, etc.), model-based methods (random field methods, fractals, etc.) and signal processing methods (spatial domain filters, Fourier domain filtering, Gabor and wavelet models, etc.).

1.2.2 Texture in Image Database Retrieval

Many researchers used texture in finding similarities between images in a database. In the IBM's QBIC Project, Niblack *et al.* [22] used features like color, texture and shape that are computed for each object in an image as well as for each image. For texture, they used features based on coarseness, contrast, and directionality which were proposed by Tamura *et al.* [55]. In [7], they developed semi-automatic tools to aid manual outlining of the objects during database population.

In the MIT Photobook Project, Pentland *et al.* [47] emphasized the fact that features used in a database retrieval system should provide a perceptually complete representation, that allows reconstruction of the images, in order them to be semantically meaningful. They used the Karhunen-Loeve transform to select eigenvectors to represent variations from the prototypical appearance as appearance-specific descriptions in the Appearance Photobook; modeled the connections in a shape using stiffness matrices produced by the finite element method as shape descriptions in the Shape Photobook; and used 2-D Wold-based decompositions that are described in [39] to measure periodicity, directionality and randomness as texture descriptions in the Texture Photobook.

In the Los Alamos National Lab.'s CANDID Project, Kelly *et al.* [34] used Laws' texture energy maps to extract textural features from pulmonary CT images and introduced a global signature based on a sum of weighted Gaussians to model the texture. They also used these Gaussian distributions to visualize which pixels con-

tribute more to the similarity score. In [35] they applied these methods to LANDSAT TM data.

Barros *et al.* [8] tried to retrieve multi-spectral satellite images by first clustering image pixels, according to their spectral values, using a modified k-means clustering procedure, then using the spectral distribution information as features for each connected region.

Jacobs *et al.* [29] used Haar wavelet decompositions and a distance measure that compares how many wavelet coefficients that two images have in common for image retrieval. They used only a few significant wavelet coefficients and also quantized them to improve the speed of the system.

Manjunath and Ma [42] used Gabor filter-based multiresolution representations to extract texture information. They used means and standard deviations of Gabor transform coefficients, computed at different scales and orientations, as features. Gabor filters performed better than the pyramid-structured wavelet transform, tree-structured wavelet transform and the multiresolution simultaneous autoregressive model (MR-SAR) in the tests performed on the Brodatz database.

In [39], Liu and Picard treated images as 2-D homogeneous random fields and used the Wold theory to decompose them into three mutually orthogonal components. These components correspond to the perceptually important "periodicity", "directionality" and "randomness" properties. They compared the features that they compute from the 2-D Wold model to other models, namely the shift-invariant principal component analysis, the multiresolution simultaneous autoregressive model, the tree-structured wavelet transform and Tamura *et al.*'s [55] features that were used in [22]. The Wold-based features performed better than others in terms of average recall for a Brodatz texture dataset.

Li *et al.* [37] used 21 different spatial features like gray level differences (mean, contrast, angular second moments, directional derivatives, etc.), co-occurrence matrices, moments, autocorrelation functions, fractals and Robert's gradient on the Brodatz

image set and on remote sensing images. The spatial features they extracted outperformed some transform-based features like the discrete cosine transform, Gabor filters, quadrature mirror filters and uniform subband coding.

Carson *et al.* [13] developed a region-based query system called "Blobworld" by first grouping pixels into regions based on color and texture using expectation-maximization and minimum description length principles, then by describing these regions using color, texture, location and shape properties. Texture features they used are anisotropy, orientation and contrast computed for each region.

Smith [54] developed a system that uses color, texture and spatial location information for image retrieval. For texture, he used the quantized energies of the quadrature mirror filter wavelet filter bank outputs at different resolutions as features. He showed that these features are gray level shift invariant because the filters have zero mean and they are size invariant because the features are normalized by the size of the resolutions. In the classification tests done using the Brodatz dataset, these features performed better than the DCT-based features.

Ma and Manjunath [41] described a system called "Netra" that also uses color, texture, shape and spatial location information. They developed an "edge flow model" that identifies the direction of changes in the feature values to segment the image into non-overlapping segments and computed the color, texture, shape and location information for each region. For texture, they used Gabor filters which are orientation and scale tunable edge detectors.

Vailaya and Jain [58] compared the effectiveness of different features like color histogram, color coherence vector, discrete cosine transform coefficients, edge direction histogram and edge direction coherence vector in classifying images into two classes: city and landscape, using a weighted nearest neighbor classifier. Edge-based features performed better than the others. They suggested building a hierarchical classifier that uses multiple two-class classifiers for image grouping.

The approaches reviewed above and the ones that will be reviewed in Chapters 2

and 3 are only a few examples from the extensive texture and content-based retrieval literature. The textural features in these approaches can be grouped into categories like micro texture-based [50, 36, 22, 34, 35, 37, 13, 58], random field modeling-based [43, 47, 39] and signal processing and transform-based [49, 8, 29, 42, 37, 41, 58].

1.3 Problem Definition

The image retrieval scenario addressed here begins with a query expressed by an image. The user inputs an image or a section of an image and desires to retrieve images from the database that have some section that is similar to the user input image. An object/process diagram, where rectangles represent objects and ellipses represent processes, of the system that will be described here is given in Figure 1.1.

Selecting features that are suitable for an application is one of the most important parts in solving the problem. Shape descriptors, color features and texture measures are all able to represent some information about an image, but the way in which they are used determine the concept "similarity" between two images. The main problem is first to find efficient features for image representation, then to find effective measures that use these representations, individually or as a combination, to establish similarity between two images. The features and the similarity measures should be efficient enough to match similar images and also should be able to discriminate dissimilar ones.

The goal of this thesis is to develop textural features for image representation and statistical measures for similarity computation. We will evaluate the performance of the proposed algorithms in terms of the effectiveness to classify image pairs as similar or dissimilar, as well as the capability to retrieve perceptually similar images as best matches while eliminating irrelevant ones. The groundtruth for the experiments are generated both by automatic methods and by human annotation.

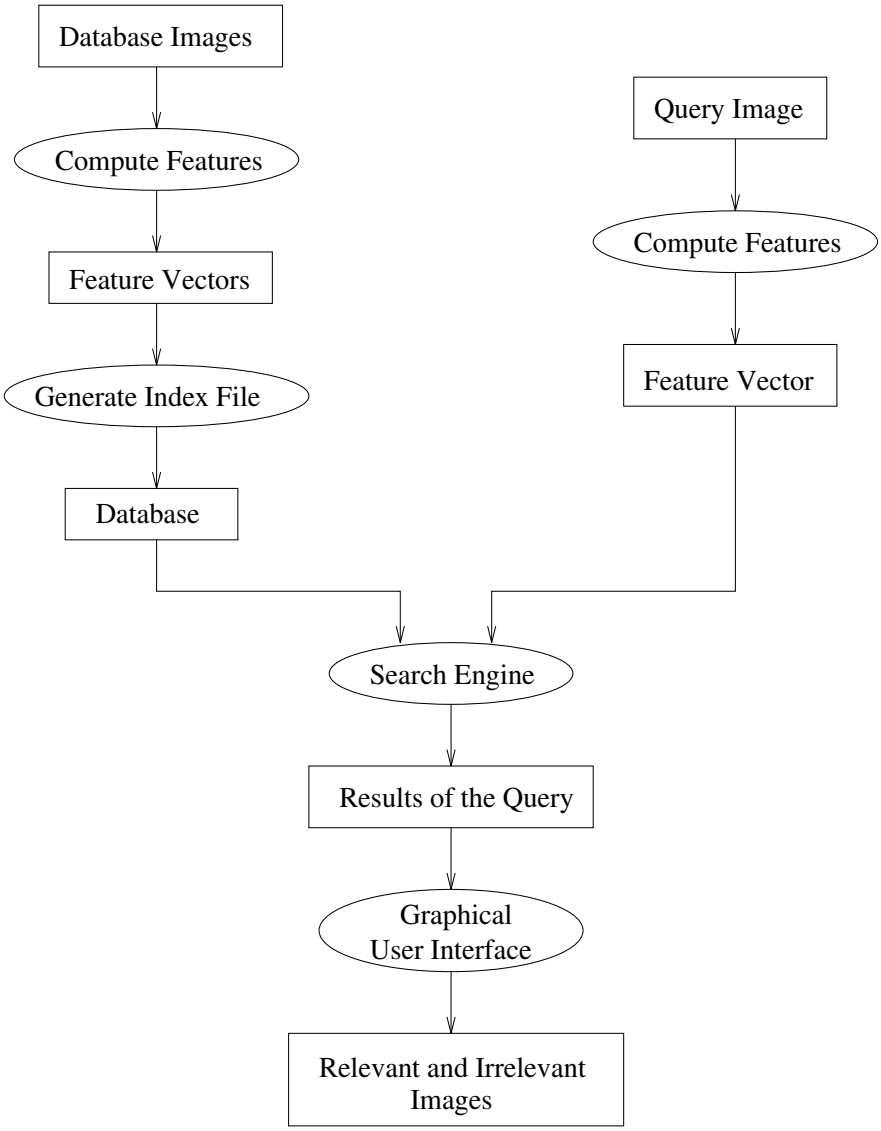


Figure 1.1: Object/process diagram for the database retrieval system.

1.4 Contributions and Thesis Organization

In this thesis, we

- develop an easy-to-compute texture histogram method that uses the spatial relationships between lines as well as the properties of their surroundings as textural features,
- develop a line selection algorithm that performs hypothesis tests to eliminate lines that are not significant enough,
- integrate macro and micro textural feature extraction methods for a multi-scale texture analysis,
- perform feature selection based on statistical methods in order to avoid heuristic selection of the parameters,
- describe a classifier that associates image pairs as relevant or irrelevant,
- design an automatic groundtruth construction protocol both to train the classifier and to evaluate the effectiveness of the features,
- develop a statistical framework to compansate the effect of "learning from an imperfect teacher" in the automatic groundtruth construction protocol,
- present experiments and performance evaluation on a large database of complex images including aerial images, remote sensing images, natural scenes, etc., not only a small set of constrained images.

The rest of the thesis is organized as follows. In the next chapter we introduce the first feature extraction algorithm, line-angle-ratio statistics, which is a macro texture measure that uses spatial relationships between lines as well as the properties of their

surroundings and is motivated by the fact that line content of an image can be used to represent texture. Chapter 3 describes the second feature extraction algorithm, variances of gray level spatial dependencies, which in turn is a micro texture measure that uses second-order (co-occurrence) statistics of gray levels of pixels at particular spatial relationships and is motivated by the fact that gray level spatial dependencies are proved to carry a significant amount of texture information and are consistent with human visual perception. Chapter 4 addresses the problem of combining these features in order to make use of their different advantages. A third feature extraction algorithm is also introduced in this chapter in order to capture the texture information that cannot be described by the first two methods. A multi-scale analysis is crucial for a compact representation, especially for large databases containing different types of complex images.

In Chapter 5, we discuss how we select the parameters for the algorithms, that best reflect the underlying texture in the images. First an automatic groundtruth construction protocol is described and is followed by a statistical decision rule that uses a Gaussian classifier. A statistical framework to estimate the true classification results using the mislabeling probabilities of the automatic groundtruth construction protocol is also described in this chapter. Chapter 6 describes the decision methods used for rank-ordering the database images according to their similarity to the query image. In Chapter 7, first we describe the database population, then we present the results of the feature selection algorithms described in Chapter 5, and finally we demonstrate the effectiveness of our features in both image classification and image retrieval. Example queries are presented and analysis of error pictures is also made in this chapter. Finally, Chapter 8 concludes by first giving a summary and then discussing some future research directions. Details of the definitions used in the derivations in Chapter 2 can be found in the Appendices.

Chapter 2

LINE-ANGLE-RATIO STATISTICS

In Section 1.3 we defined the problem as finding efficient textural representations for images. Experiments on various types of images showed us that one of the strongest spatial features of an image is its line segments and the angles between intersecting line segments [62]. Edge and line information has been extensively used since very early approaches to texture. Rosenfeld [50, 49] discussed that line content of an image can be used to represent texture in the image. Therefore, an image can be roughly represented by the lines extracted from it.

In this chapter, we describe how we extract texture information using lines extracted from an image as textural primitives as well as using the properties of their surroundings to assign signatures to images for content-based retrieval. In doing so, first we discuss the pre-processing steps, then we present the details of feature extraction. In this work we assume that images in the database have some line content.

2.1 Pre-processing

Since our goal is to find a section in the database which is relevant to the input query, before retrieval, each image in the database is divided into overlapping sub-images. The protocol for database population will be described in detail in Chapter 7. Each sub-image is then processed offline by Canny's edge detector [12], Etemadi's edge linker [21], line selection operator and line grouping operator to detect line pairs to associate with each sub-image in each image a set of feature records. Goal of the line selection operator is to perform hypothesis tests to eliminate lines that do not have

significant difference between the gray level distributions on both sides and goal of the line grouping operator is to find intersecting and/or near-intersecting line pairs. In the following sections we describe the line selection and line grouping operators in more detail.

2.1.1 Line Selection

Edge detection followed by line detection operators often result in many false alarms. It is especially hard to select proper parameters for these operators if one does not have groundtruth information as training data. After line detection, we use hypothesis testing to eliminate lines that do not have significant difference between the gray level distributions in the regions on their right and left.

Yakimovsky [61] used a similar approach to find edges for object boundary detection. He used a maximum-likelihood test to decide on whether two sets of mutually exclusive neighborhoods of a pixel, which are assumed to have normally distributed gray levels, have the same distributions or not. If the hypothesis that two neighborhoods have the same distributions can be rejected, the pixel is labeled as an edge pixel.

The algorithm we develop for line selection can be described as follows. Let the set of N gray levels $x_1, x_2, \ldots, x_N$ are samples from the region to the right of a line and the set of M gray levels $y_1, y_2, \ldots, y_M$ are samples from the region to the left of that line. To select these samples, first, Definition 3 in Appendix A is used to find pixels that are within 6 pixel neighborhood of the line and then, Definition 5 is used to determine which ones are on the left and which ones are on the right. We assume that the samples in both sets are drawn from normal distributions

$$x_1, x_2, \ldots, x_N \quad \sim \; N(\mu_x, \sigma_x^2) \tag{2.1}$$

and

$$y_1, y_2, \ldots, y_M \quad \sim \; N(\mu_y, \sigma_y^2). \tag{2.2}$$

We want to test whether these two sets of values come from the same distribution or not.

Let's define $\bar{x}$ and $\bar{y}$, which are averages of $x_n, n = 1, \ldots, N$ and $y_m, m = 1, \ldots, M$ respectively, as

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n \tag{2.3}$$

and

$$\bar{y} = \frac{1}{M} \sum_{m=1}^{M} y_m. \tag{2.4}$$

Then, we have

$$\bar{x} \ \sim \ N\left(\mu_x, \frac{\sigma_x^2}{N}\right) \tag{2.5}$$

and

$$\bar{y} \ \sim \ N\left(\mu_y, \frac{\sigma_y^2}{M}\right). \tag{2.6}$$

Then the random variable z

$$z = \bar{x} - \bar{y} \tag{2.7}$$

has a distribution

$$N(\mu_z, \sigma_z^2) = N\left(\mu_x - \mu_y, \frac{\sigma_x^2}{N} + \frac{\sigma_y^2}{M}\right). \tag{2.8}$$

For the hypothesis testing, let's define the null hypothesis as

$$H_0 : \mu_x = \mu_y = \mu \quad \text{and} \quad \sigma_x = \sigma_y = \sigma \tag{2.9}$$

which means gray levels $x_n, n = 1, \ldots, N$ and $y_m, m = 1, \ldots, M$ come from the same distribution, and the alternative hypothesis as

$$H_1 : H_0 \text{ not true} \tag{2.10}$$

which means two sets of gray levels come from different distributions. We do not know the parameters μ and σ but it is not important because they cancel out in the derivations.

To form the test statistic, we define two random variables A and B as

$$A = \left(\frac{z-\mu_z}{\sigma_z}\right)^2 \tag{2.11}$$

and

$$B = \frac{1}{N-1}\sum_{n=1}^{N}\left(\frac{x-\bar{x}}{\sigma_x}\right)^2 + \frac{1}{M-1}\sum_{m=1}^{M}\left(\frac{y-\bar{y}}{\sigma_y}\right)^2. \tag{2.12}$$

Under the null hypothesis, the random variables in equations (2.11) and (2.12) become

$$A = \frac{z^2}{(\frac{1}{N}+\frac{1}{M})\sigma^2} \tag{2.13}$$

and

$$B = \frac{1}{(N-1)\sigma^2}\sum_{n=1}^{N}(x-\bar{x})^2 + \frac{1}{(M-1)\sigma^2}\sum_{m=1}^{M}(y-\bar{y})^2. \tag{2.14}$$

We have

$$A \sim \chi_1^2 \tag{2.15}$$

and

$$B \sim \chi_{N+M-2}^2. \tag{2.16}$$

Then, we form the test statistic F as

$$\begin{aligned} F &= \frac{A/1}{B/(N+M-2)} \\ &= \frac{(\bar{x}-\bar{y})^2}{\frac{1}{N-1}\sum_{n=1}^{N}(x-\bar{x})^2 + \frac{1}{M-1}\sum_{m=1}^{M}(y-\bar{y})^2}\,\frac{N+M-2}{\frac{1}{N}+\frac{1}{M}} \end{aligned} \tag{2.17}$$

which follows the distribution $F_{1,N+M-2}$ [14].

Given a threshold α, if $P(F|1, N+M-2) < \alpha$, the alternative hypothesis is accepted; otherwise, the null hypothesis is accepted. If the null hypothesis H_0 is true, the line is rejected, if the alternate hypothesis H_1 is true, the line is accepted as a significant one because the distributions on either sides of it are significantly different.

2.1.2 Line Grouping

After hypothesis testing, remaining are the lines that are significant enough according to our test statistic. Now, we want to find intersecting and near-intersecting ones among them.

Given two lines L_1 and L_2 with end points $(P_1, P_2) = \left(\begin{bmatrix} r_1 \\ c_1 \end{bmatrix}, \begin{bmatrix} r_2 \\ c_2 \end{bmatrix} \right)$ and $(P_3, P_4) = \left(\begin{bmatrix} r_3 \\ c_3 \end{bmatrix}, \begin{bmatrix} r_4 \\ c_4 \end{bmatrix} \right)$ respectively, equations of them can be written as

$$L_1 : \quad P = P_1 + \lambda_1(P_2 - P_1), \tag{2.18}$$

$$L_2 : \quad P = P_3 + \lambda_2(P_4 - P_3) \tag{2.19}$$

using Definition 1 in Appendix A. The following conditions should be satisfied for intersection:

$$r_1 + \lambda_1(r_2 - r_1) = r_3 + \lambda_2(r_4 - r_3), \tag{2.20}$$

$$c_1 + \lambda_1(c_2 - c_1) = c_3 + \lambda_2(c_4 - c_3). \tag{2.21}$$

If $(r_4 - r_3)(c_2 - c_1) = (r_2 - r_1)(c_4 - c_3)$, lines L_1 and L_2 are parallel. If also $(r_2 - r_1)(c_3 - c_1) = (r_3 - r_1)(c_2 - c_1)$, end points P_1, P_2, P_3, P_4 are colinear. If neither of these cases are true, λ_1 and λ_2 can be derived from equations (2.20) and (2.21) as

$$\lambda_2 = \frac{(r_2 - r_1)(c_3 - c_1) - (r_3 - r_1)(c_2 - c_1)}{(r_4 - r_3)(c_2 - c_1) - (r_2 - r_1)(c_4 - c_3)} \tag{2.22}$$

and

$$\begin{aligned} \lambda_1 &= \frac{r_3 - r_1}{r_2 - r_1} + \lambda_2 \frac{r_4 - r_3}{r_2 - r_1} \text{ if } r_1 \neq r_2 \\ &\text{or} \\ &= \frac{c_3 - c_1}{c_2 - c_1} + \lambda_2 \frac{c_4 - c_3}{c_2 - c_1} \text{ if } c_1 \neq c_2. \end{aligned} \tag{2.23}$$

Let's define *Tol* as the tolerance, in number of pixels, for the end points of the lines to intersect. We need to define this tolerance to allow near-intersection instead

of exact end point intersection. To determine the tolerances for λ_1 and λ_2, two new tolerances τ_1 and τ_2 can be defined as

$$\begin{aligned} \tau_1 &= \frac{Tol}{||P_2P_1||} \\ &= \frac{Tol}{\sqrt{(r_2 - r_1)^2 + (c_2 - c_1)^2}} \end{aligned} \tag{2.24}$$

and

$$\begin{aligned} \tau_2 &= \frac{Tol}{||P_4P_3||} \\ &= \frac{Tol}{\sqrt{(r_4 - r_3)^2 + (c_4 - c_3)^2}}. \end{aligned} \tag{2.25}$$

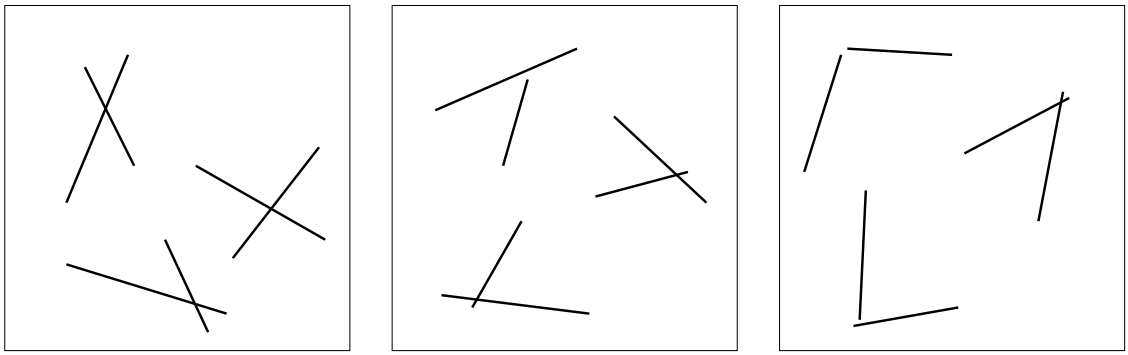
If $\tau_1 \leq \lambda_1 \leq 1 - \tau_1$ and $\tau_2 \leq \lambda_2 \leq 1 - \tau_2$, two lines cross each other, if ($\tau_1 \leq \lambda_1 \leq 1-\tau_1$ and ($|\lambda_2| < \tau_2$ or $|\lambda_2 - 1| < \tau_2$)) or ($\tau_2 \leq \lambda_2 \leq 1-\tau_2$ and ($|\lambda_1| < \tau_1$ or $|\lambda_1 - 1| < \tau_1$)), two lines have a T-like intersection, and if ($|\lambda_1| < \tau_1$ or $|\lambda_1 - 1| < \tau_1$) and ($|\lambda_2| < \tau_2$ or $|\lambda_2 - 1| < \tau_2$), two lines intersect at the end points within the given tolerance. Then, the intersection point $\begin{bmatrix} r \\ c \end{bmatrix}$ can be found by substituting λ_1 into the equation (2.18) as

$$\begin{bmatrix} r \\ c \end{bmatrix} = \begin{bmatrix} r_1 \\ c_1 \end{bmatrix} + \lambda_1 \begin{bmatrix} r_2 - r_1 \\ c_2 - c_1 \end{bmatrix}. \tag{2.26}$$

Examples of, what we call, crossing lines, T-like intersections and lines intersecting at end points are given in Figure 2.1. Examples for the pre-processing steps are given in Figure 2.2.

2.2 Computation of Features

The features for each pair of intersecting and near-intersecting line segments consist of the angle between two lines and the ratio of mean gray level inside the region spanned by that angle to the mean gray level outside that region.



(a) Crossing lines. (b) T-like intersections. (c) Intersection at end points.

Figure 2.1: Line grouping examples.

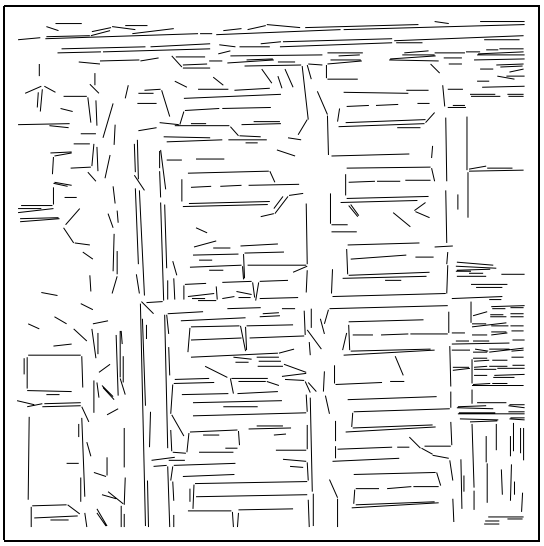
The feature extraction process is done as follows. After the line segments are extracted from a sub-image, they are grouped into pairs that have intersection/near-intersection inside that sub-image. Then, for each pair, the angle between the lines and the corresponding mean gray level ratio are computed. Details of the feature extraction process are given in the following sections.

2.2.1 Angle Computation

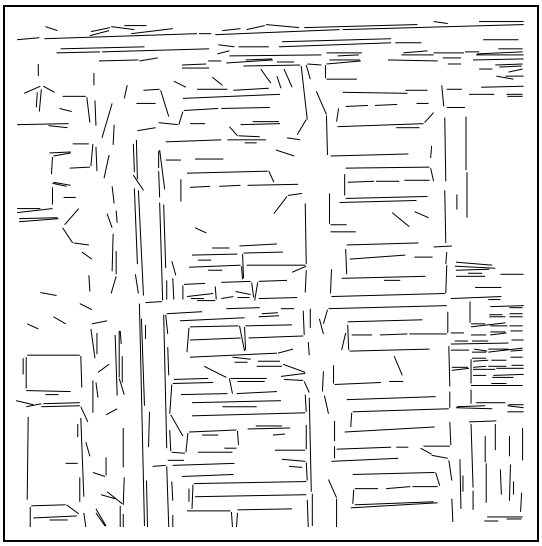
The angle between two lines is given by Definition 2 in Appendix A. This formula results in angle values that are in the range $[0°, 180°]$. Angles that are less than $10°$ or greater than $170°$ are ignored because these line pairs usually are broken segments of longer lines. A proper approach to avoid broken lines can be to use a line-fitting and noise cleaning algorithm as a pre-processing step. An example for noise cleaning on lines can be found in [62] where Zhou first assumed a line perturbation model and then used least-squares line-fitting to connect the broken lines. We do not use any line-fitting step in order not to decrease the speed of the feature extraction process.

(a) Grayscale image.

(b) Gradient image.

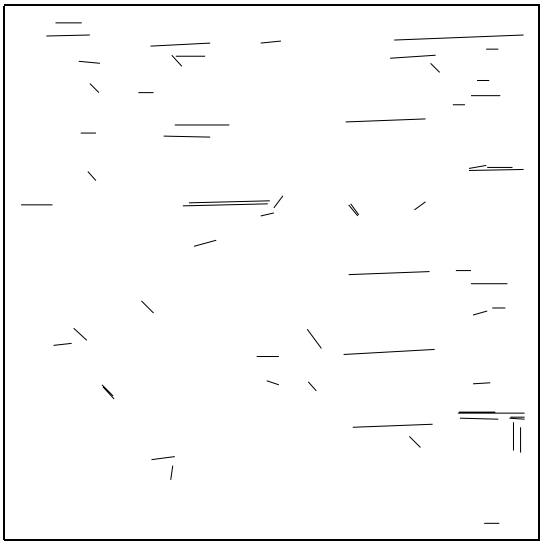


(c) Extracted lines after line detection operator.

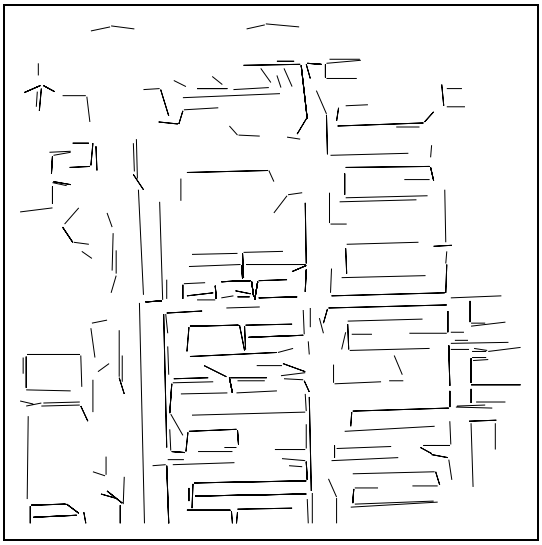


(d) Accepted lines after line selection operator.

Figure 2.2: Line selection and grouping pre-processing steps.



(e) Rejected lines after line selection operator.



(f) Resulting lines after line grouping operator.

Figure 2.2: Line selection and grouping pre-processing steps (cont.).

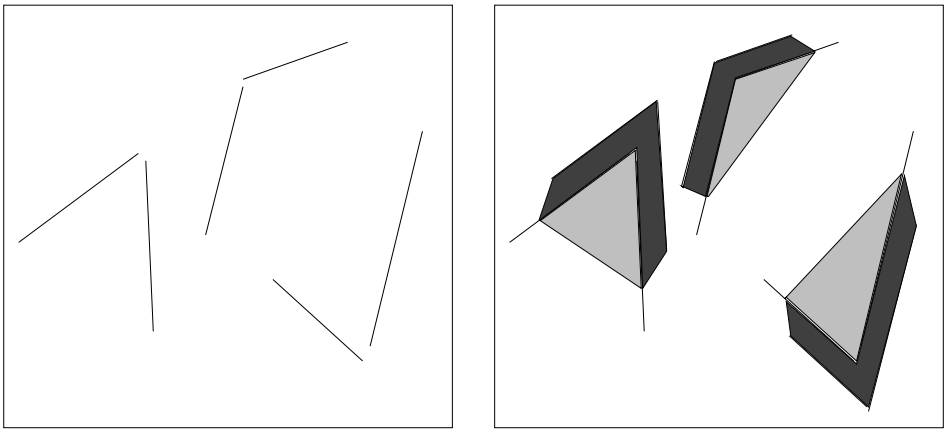
2.2.2 Ratio Computation

The second feature computed for each pair of intersecting/near-intersecting lines is the ratio of the mean gray level inside the region spanned by them to the mean gray level outside that region. The regions used to compute the means are found by the convention below:

- *in* region is defined as the pixels that fall into the region bounded by segments with length that is 80 percent of the length of the bounding lines

- *out* region is defined as the pixels that fall in the region bounded by any one of the line segments and the shifted version of that line segment by a defined amount.

Details of this region convention are explained in Appendix B. An example is given in Figure 2.3. The light shaded regions and dark shaded regions show the *in* and *out* regions respectively. The means are computed for the regions that are within the sub-image borders.



(a) Pairs of intersecting lines.

(b) Regions used for mean calculation.

Figure 2.3: Examples of region convention for mean ratio calculation. Light and dark shaded regions show the *in* and *out* regions respectively.

Since the possible range of ratio values is infinite, we restrict them to the range $[0, 1)$. To make a ratio value fall into this range, we take the reciprocal of it if the inner region is brighter than the outer region. The resulting ratios cannot be 1 because we guarantee to have lines that are significant enough by hypothesis testing during the line extraction process. To restrict the range, one might consider saturating ratio values at a value L which is greater than 1. We observed that this solution does not work well because ratio values are not equally probable since they are collected in the range $[0, 1)$ if the inner region is brighter, and spreaded out in the range $(1, L]$ if the outer region is brighter.

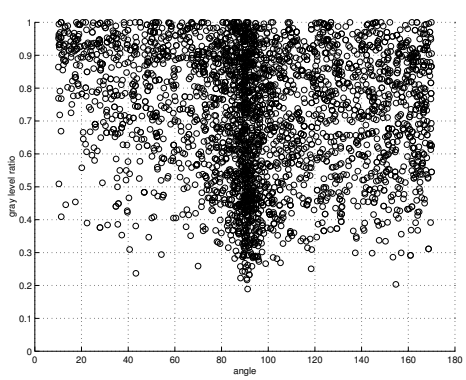
2.3 Texture Histogram

The features that are extracted from the image form a two-dimensional space of angles and the corresponding ratios. This two-dimensional feature space is then partitioned into a fixed set of Q cells. The signature vector, that we will call as feature vector in the rest of the thesis, for each sub-image is then the Q-dimensional vector which has for its q'th component the number of angle-ratio pairs that fall into that q'th cell. As can be seen in Figure 2.4(a), these features do not have a uniform distribution. Therefore, to form this partition of Q cells, the standard vector quantization algorithm [38] is used as the training algorithm. Then, a feature vector can be formed by counting the number of angle-ratio pairs that are assigned to each cell according to the Euclidean distance. This forms the texture histogram.

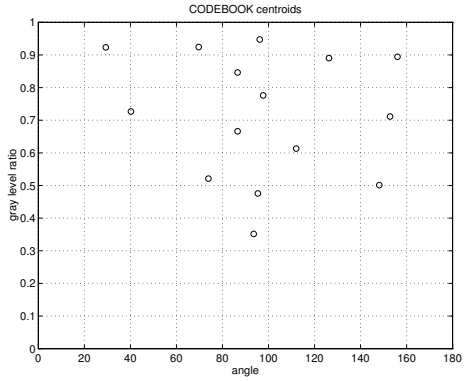
2.4 Feature Selection

One can select Q, the number of quantization cells, either heuristically or using some formal methods. In Chapter 5, we will first review some previous work on feature selection, then describe the statistical methods we use. In order to reduce the search space for Q, we consider only 15, 10 and 25 as the possible number of quantization cells. In Section 7.2.1, we will present the results of the tests and select the value we use in the rest of the experiments.

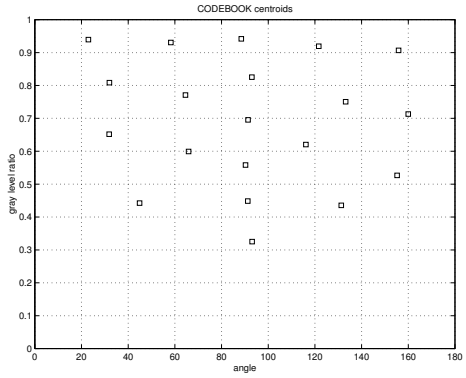
To give an idea how the line-angle-ratio feature space looks like, distribution of the training samples used in the experiments are shown in Figure 2.4(a). Centroids of the resulting partitions using 15, 20 and 25 quantization cells are given in Figures 2.4(b), 2.4(c) and 2.4(d) respectively.



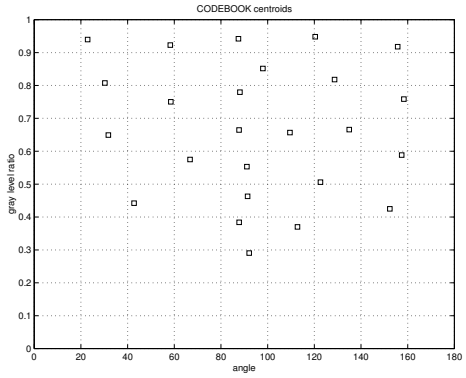
(a) Distribution of the training samples.



(b) Resulting partitions for 15 quantization cells.



(c) Resulting partitions for 20 quantization cells.



(d) Resulting partitions for 25 quantization cells.

Figure 2.4: Line-Angle-Ratio feature space distribution and centroids of the resulting partitions with different number of quantization cells.

2.5 Summary

The pre-processing and feature extraction steps for the line-angle-ratio statistics method are summarized in the object/process diagram in Figure 2.5. Given a sub-image, the features are computed using the following steps:

- Perform edge detection.
- Perform edge linking.
- Perform line selection to eliminate insignificant lines.
- Perform line grouping to find intersecting/near-intersecting lines.
- Compute angles between pairs of intersecting/near-intersecting lines.
- Compute ratios of mean gray levels inside and outside the regions spanned by these lines.
- Compute the texture histograms by counting the angle-ratio pairs that fall into each partition in the two-dimesional feature space.

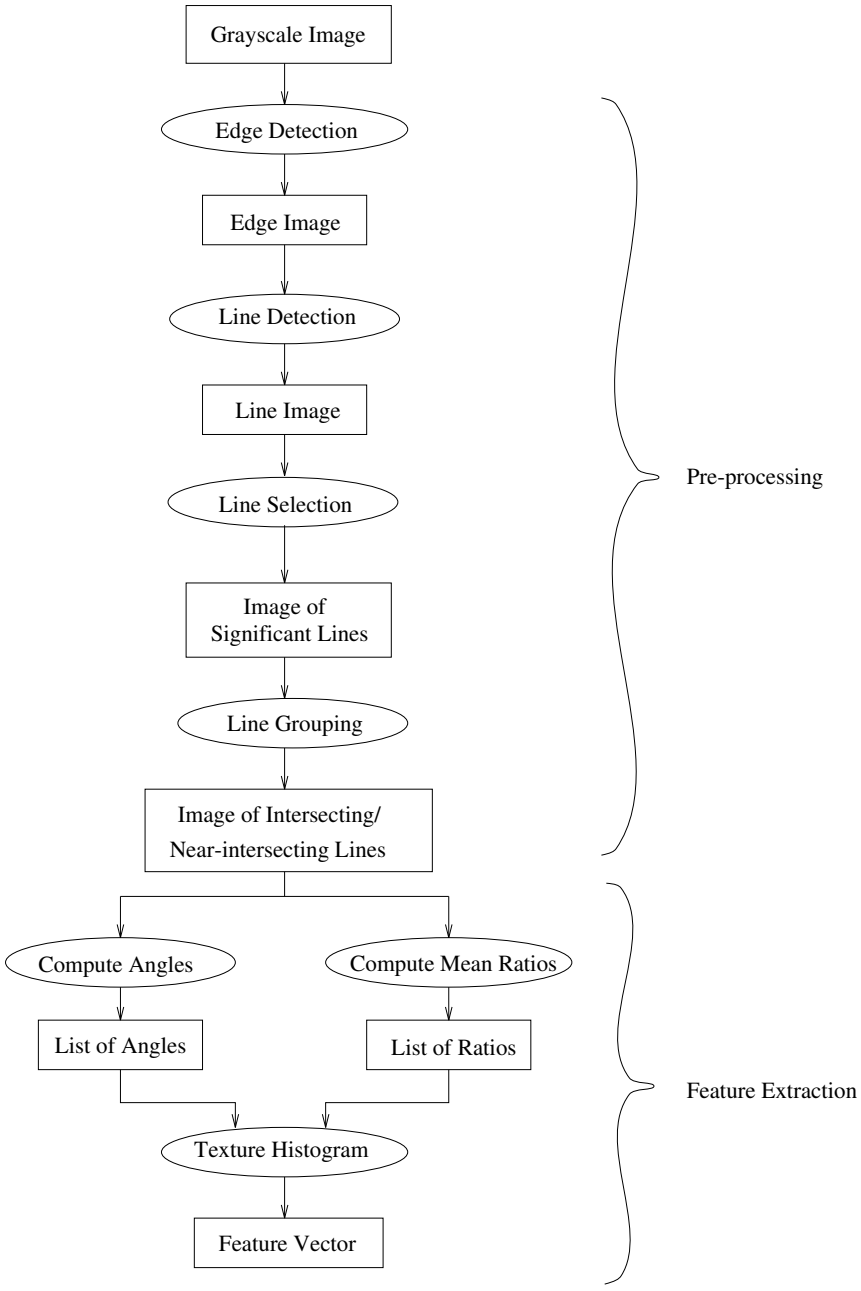


Figure 2.5: Object/process diagram for the pre-processing and feature extraction steps of the Line-Angle-Ratio Statistics method.

Chapter 3

VARIANCES OF GRAY LEVEL SPATIAL DEPENDENCIES

Structural approaches have been one of the major research directions for texture analysis. They use the idea that texture is composed of primitives with different properties appearing in particular spatial arrangements. On the other hand, statistical approaches try to model texture using statistical distributions either in the spatial domain or in a transform domain. One way to combine these two approaches is to define texture as being specified by the statistical distribution of the properties of different textural primitives occurring at different spatial relationships.

A pixel, with its gray level as its property, is the simplest primitive that can be defined in a digital image. Consequently, distribution of pixel gray levels can be described by first-order statistics like mean, standard variation, skewness and kurtosis or second-order statistics like the probability of two pixels having particular gray levels occurring at particular spatial relationships. This information can be summarized in two-dimensional co-occurrence matrices computed for different distances and orientations. Coarse textures are ones for which the distribution changes slightly with distance, whereas for fine textures the distribution changes rapidly with distance.

In the following sections, first we review some of the previous approaches to gray level spatial dependencies, then this is followed by a formal definition for gray level co-occurrence, next we describe the pre-processing algorithm, and finally we discuss the features we compute from these co-occurrence matrices.

3.1 Literature Overview and Motivation

The initial work on texture discrimination using gray level spatial dependencies was done in early seventies. In some early comparative studies, researchers have observed that gray level spatial dependency matrices were very successful in disriminating images with relatively homogeneous textures. Although there are a few counter examples [32], an important amount of human texture discrimination is also shown to be dependent on second-order statistics.

Julesz [32] was the first to conduct experiments to determine the effects of high-order spatial dependencies on the visual perception of synthetic textures. He showed that, although with few exceptions, textures with different first- and second-order probability distributions can be easily discriminated but differences in the third- or higher-order statistics are irrelevant [33].

One of the early approaches that use spatial relationships of gray levels in texture discrimination is [25], where Haralick used features like the angular second moment, angular second moment difference, angular second moment inverse difference, and correlation, computed from the co-occurrence matrices for automatic scene identification of remote sensing images and achieved 70% accuracy.

In [27], Haralick *et al.* again used features computed from the co-occurrence matrices to classify sandstone photomicrographs, panchromatic aerial photographs, and ERTS multispectral satellite images. The features they computed are angular second moment, contrast, correlation, sum of squares, inverse difference moment, sum average, sum variance, sum entropy, entropy, correlation and maximal correlation coefficient, which relate to specific textural characteristics of an image such as homogeneity, contrast and the presence of organized structure. They obtained accuracies between 80-90% for different datasets they did tests on. Although they used only some of the features they defined and did not use the same classification algorithm for different datasets in their tests, it can be concluded that features they compute

from co-occurrence matrices performed well in distinguishing between different texture classes in many kinds of image data.

Weszka *et al.* [59] made a comparative study of four texture classification algorithms; Fourier power spectrum, co-occurrence matrices, gray level difference statistics and gray level run length statistics, to classify aerial photographic terrain samples and also LANDSAT images. They obtained results similar to Haralick's [27] and concluded that features computed from co-occurrence matrices perform as well as or better than other algorithms.

Another comparative study was done by Conners and Harlow [16]. They used Markov-generated images to evaluate the performances of different texture analysis algorithms for automatic texture discrimination and concluded that the spatial gray level dependencies method performed better than the gray level run length method, power spectrum method and gray level difference method. In [18], they used theoretical comparison methodologies to evaluate the performances of these algorithms. They again used Markov-generated images and concluded that spatial gray level dependencies method performed better than the other three. These theoretical conclusions are consistent with the experimental results of Weszka *et al.* [59]. Specifically for the spatial gray level dependencies method, they concluded that using multiple distances improve performance but the commonly used measures of inertia, energy, entropy, correlation and local homogeneity do not capture all of the important texture information contained in the spatial gray level dependency matrices.

A more recent study that compares four textural features was done by Ohanian and Dubes [46]. They evaluated the performance of Markov Random Field features, Gabor filter features, co-occurrence features and fractal features in terms of their ability to classify single-texture images. The criteria used for performance was based on the probability of misclassification using a k-nearest neighbor decision rule with the leave-one-out method. Whitney's [60] forward selection algorithm was used for feature selection. Experiments conducted on synthetic images generated by fractal

methods and Gaussian Markov Random Fields and natural images of different types of leather and painted surfaces showed that co-occurrence features again performed the best, followed by the fractal features, for this dataset with 32×32 samples from each type of images.

From these experiments, it can be concluded that spatial gray level dependency matrices carry a significant amount of texture information in images with some homogeneously textured regions and perform better than many other texture extraction algorithms, that were listed above, in the micro-texture level. This seems to be a good choice for our application of finding images having similar sections.

3.2 Gray Level Co-occurrence

Co-occurrence, in general form [20, 26], can be specified in a matrix of relative frequencies $P(i, j; d, \theta)$ with which two neighboring texture elements separated by distance d at orientation θ occur in the image, one with property i and the other with property j. In gray level co-occurrence, as a special case, texture elements are pixels and properties are gray levels. For example, for a 0° angular relationship, $P(i, j; d, 0^\circ)$ averages the probability of a left-right transition of gray level i to gray level j at a distance d.

In the derivations below, origin of the image is defined as the upper-left corner pixel. Let $L_r = \{0, 1, \ldots, N_r - 1\}$ and $L_c = \{0, 1, \ldots, N_c - 1\}$ be the spatial domains of row and column dimensions, and $G = \{0, 1, \ldots, N_g - 1\}$ be the domain of gray levels. The image I can be represented as a function which assigns a gray level to each pixel in the domain of the image; $I : L_r \times L_c \rightarrow G$. Then, for the orientations

shown in Figure 3.1, gray level co-occurrence matrices can be defined as

$$\begin{aligned}
P(i,j;d,0^\circ) &= \#\{((r,c),(r',c')) \in (L_r \times L_c) \times (L_r \times L_c) | \\
&\qquad r'-r=0, |c'-c|=d, I(r,c)=i, I(r',c')=j\} \\
P(i,j;d,45^\circ) &= \#\{((r,c),(r',c')) \in (L_r \times L_c) \times (L_r \times L_c) | \\
&\qquad (r'-r=d, c'-c=d) \text{ or } (r'-r=-d, c'-c=-d), \\
&\qquad I(r,c)=i, I(r',c')=j\} \\
P(i,j;d,90^\circ) &= \#\{((r,c),(r',c')) \in (L_r \times L_c) \times (L_r \times L_c) | \\
&\qquad |r'-r|=d, c'-c=0, I(r,c)=i, I(r',c')=j\} \\
P(i,j;d,135^\circ) &= \#\{((r,c),(r',c')) \in (L_r \times L_c) \times (L_r \times L_c) | \\
&\qquad (r'-r=d, c'-c=-d) \text{ or } (r'-r=-d, c'-c=d), \\
&\qquad I(r,c)=i, I(r',c')=j\}.
\end{aligned} \tag{3.1}$$

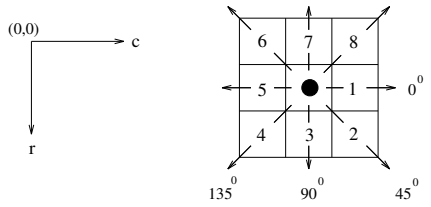


Figure 3.1: Spatial arrangements of pixels.

Resulting matrices are symmetric. The distance metric used in equation (3.1) can be explicitly defined as

$$\rho((r,c),(r',c')) = \max\{|r-r'|, |c-c'|\}. \tag{3.2}$$

Figure 3.2 shows an example for co-occurrence matrix computation from [27] for a 4×4 image with gray levels between 0 and 3.

These matrices can be normalized by dividing each entry in a matrix by the number of neighboring pixels used in computing that matrix. Given distance d,

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

(a)
4x4 image
with gray levels
0-3.

		Gray Level			
		0	1	2	3
Gray Level	0	#(0,0)	#(0,1)	#(0,2)	#(0,3)
	1	#(1,0)	#(1,1)	#(1,2)	#(1,3)
	2	#(2,0)	#(2,1)	#(2,2)	#(2,3)
	3	#(3,0)	#(3,1)	#(3,2)	#(3,3)

(b) General form of co-occurrence matrices $P(i,j;d,\theta)$ for gray levels 0-3 where $\#(i,j)$ stands for number of times gray levels i and j have been neighbors.

$$P(i,j;1,0^\circ) = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

(c) $(d,\theta) = (1,0^\circ)$

$$P(i,j;1,45^\circ) = \begin{pmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

(d) $(d,\theta) = (1,45^\circ)$

$$P(i,j;1,90^\circ) = \begin{pmatrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

(e) $(d,\theta) = (1,90^\circ)$

$$P(i,j;1,135^\circ) = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

(f) $(d,\theta) = (1,135^\circ)$

Figure 3.2: Examples for co-occurrence matrix computation from Haralick [27].

this number is $2N_r(N_c - d)$ for 0° orientation, $2(N_r - d)(N_c - d)$ for 45° and 135° orientations, and $2(N_r - d)N_c$ for 90° orientation.

Weszka *et al.* [59] discussed that if a texture is coarse and the distance d used to compute the co-occurrence matrix is small compared to the sizes of the texture elements, pairs of pixels at separation d should usually have similar gray levels. This means that high values in the matrix $P(i, j; d, \theta)$ should be concentrated on or near its main diagonal. Conversely, for a fine texture, if d is comparable to the texture element size, then the gray levels of points separated by d should often be quite different, so that values in $P(i, j; d, \theta)$ should be spread out relatively uniformly. Similarly, if a texture is directional, i.e. coarser in one direction than another, the degree of spread of the values about the main diagonal in $P(i, j; d, \theta)$ should vary with the orientation θ. Thus texture directionality can be analyzed by comparing spread measures of the $P(i, j; d, \theta)$ for various orientations.

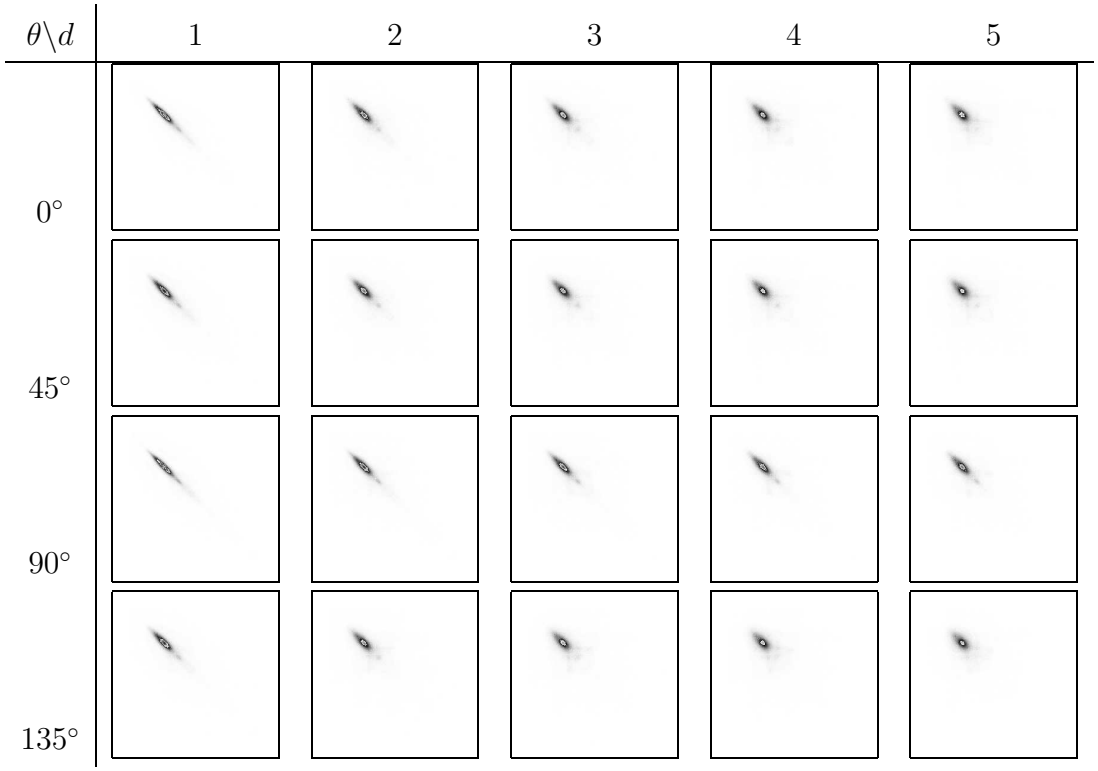
Example co-occurrence matrices are given in Figures 3.3 and 3.4. In Figure 3.3, the grayscale image has small amount of local spatial variations so the co-occurrence values are concentrated near the main diagonals. On the other hand, in Figure 3.4, gray levels have larger amount of local spatial variations so co-occurrence matrices are more sparse.

3.3 Pre-processing

Before computing co-occurrence matrices, a common approach is to apply Equal Probability Quantization as a pre-processing step [27, 49, 59, 16, 56, 17, 19, 63]. The idea is to overcome the effects of monotonic transformations of the true image gray levels caused by the variations in lighting, lens, film, developer and digitizers. Equal probability quantization guarantees that images which are monotonic transformations of each other produce the same results (please refer to [27] for a proof).

Conners and Harlow [17] examined the effects of distortions caused by the dif-

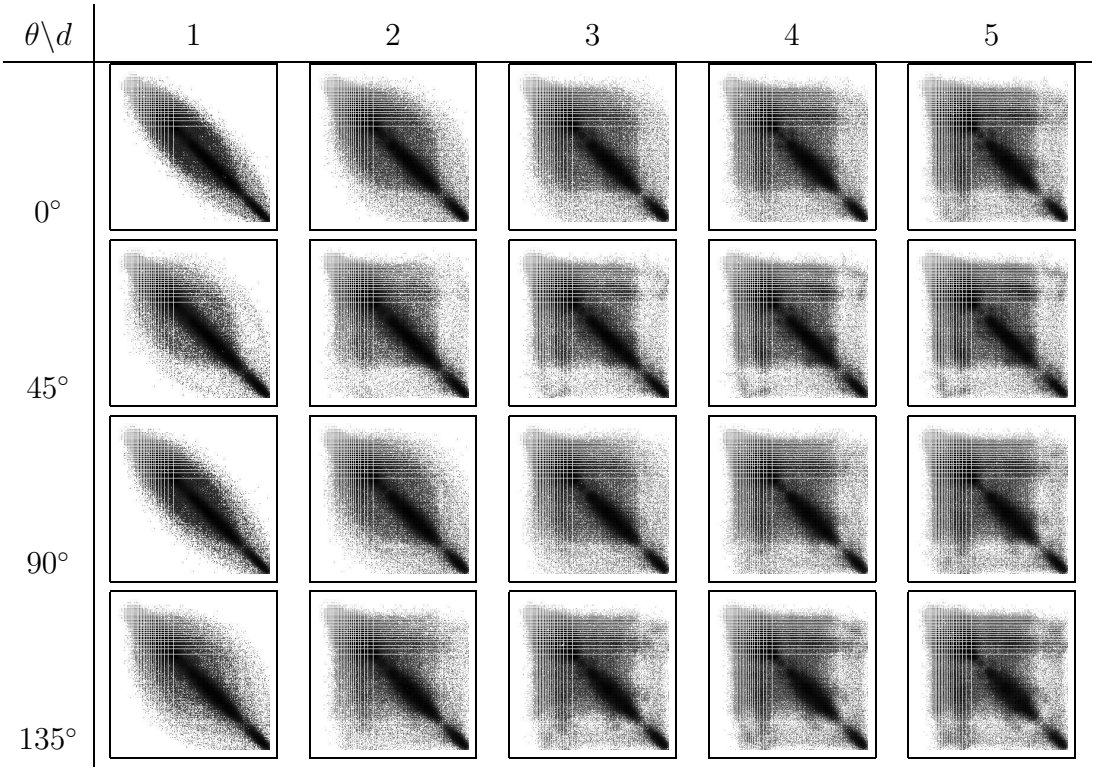
(a) Grayscale image.



(b) Co-occurrence matrices for different (d, θ).

Figure 3.3: Co-occurrence matrices for an image with a small amount of local spatial variations.

(a) Grayscale image.



(b) Co-occurrence matrices for different (d, θ).

Figure 3.4: Co-occurrence matrices for an image with a large amount of local spatial variations.

ferences in exposure time, development temperature, development time and scanner settings on radiographic images and showed that equal probability quantization normalizes the image contrast, provides a near-optimal way to reduce the number of gray levels in an image while retaining an accurate representation, and makes spatial gray level dependency matrices invariant to the distortions mentioned above.

We use the algorithm in [27] which iteratively tries to quantize the remaining unquantized gray levels into the remaining number of levels. The algorithm can be summarized as follows:

- Let $\mathbf{x}$ be a non-negative random variable with a cumulative probability distribution function $F_{\mathbf{x}}$.

- Let $G_{\mathbf{x}}$, the K–level equal probability quantization function for $\mathbf{x}$, be defined as $G_{\mathbf{x}} = k$ if and only if $g_{k-1} \leq x < g_k$, where g_{k-1} and g_k are the end points of the k'th quantization level and $k = 1, \ldots, K$.

- Iterate to find the quantization levels g_k's as follows:

  - Let $g_0 = 0$.
  - Assume g_{k-1} is defined.
  - Then g_k is the smallest number such that

$$\left| \frac{1 - F_{\mathbf{x}}(g_{k-1})}{K - (k-1)} - (F_{\mathbf{x}}(g_k) - F_{\mathbf{x}}(g_{k-1})) \right| \leq \left| \frac{1 - F_{\mathbf{x}}(g_{k-1})}{K - (k-1)} - (F_{\mathbf{x}}(g) - F_{\mathbf{x}}(g_{k-1})) \right|, \quad \forall g. \tag{3.3}$$

Examples are given in Figures 3.5 and 3.6. Although the original image and its monotonically transformed images in Figure 3.5 have significantly different co-occurrence matrices, equal probability quantized versions of them result in approximately the same co-occurrence distributions of gray levels in Figure 3.6. This fact

will make the features, therefore the similarity computation, invariant to distortions resulting in monotonic gray level transformations.

In the experiments that will be described in Chapter 7, we use 64 quantization levels because it performed the best among 16, 32 and 64 levels in terms of "total cost" that will be defined in Chapter 5. In the literature, usually small number of levels were used because the images under consideration usually contained homogeneous textures, but our images, that will be presented in Section 7.1, are much more complex than those images and small number of levels cause significant information loss.

3.4 Textural Features

In order to use the information contained in the gray level co-occurrence matrices, Haralick *et al.* [27] defined 14 statistical measures which measure textural characteristics like homogeneity, contrast, organized structure, complexity, and nature of gray level transitions. Since many distances and orientations result in a very large number of values, computation of co-occurrence matrices and extraction of textural features from them become infeasible for an image retrieval application which requires fast computation of features. We decided to use only the variance

$$v(d,\theta) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i-j)^2 P(i,j;d,\theta) \tag{3.4}$$

which is a difference moment of P and measures the contrast in the image. Rosenfeld and Troy [50] called this feature the moment of inertia. It will have a large value for images which have a large amount of local spatial variation in gray levels and a smaller value for images with spatially uniform gray level distributions.

Conners and Harlow [19] used a tiling model for texture which is composed of parallelogram unit patterns as primitives and used the inertia feature for periodicity detection. They showed that the local minima of the inertia feature computed at different distances at a given orientation are candidate points for periodicity at that

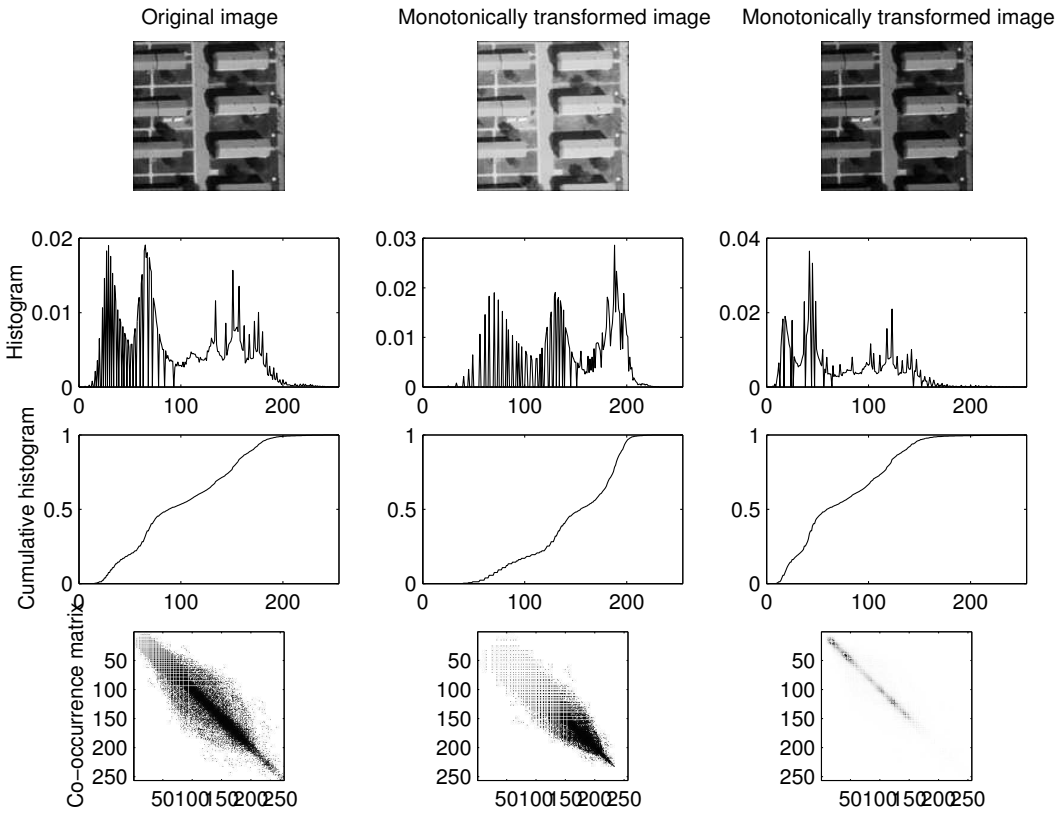


Figure 3.5: Example images motivating the Equal Probability Quantization Algorithm. First column shows the original grayscale image, second column shows an image that is made brighter and third column shows an image that is made darker by monotonic transforms on gray levels. Corresponding gray level histograms, cumulative histograms and co-occurrence matrices are plotted in the second, third and fourth rows respectively.

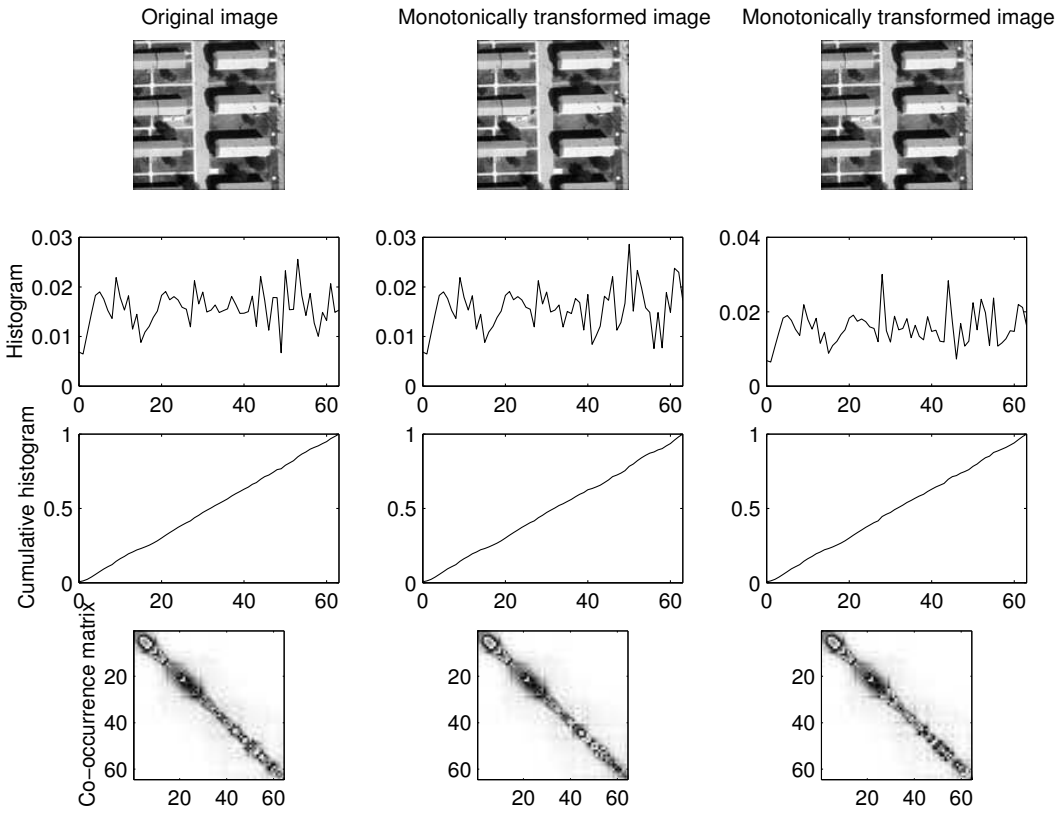


Figure 3.6: Results of the Equal Probability Quantization Algorithm performed on the images in Figure 3.5 using 64 levels. First column shows the quantized version of the original grayscale image, second column shows the quantized version of the brightened image and third column shows the quantized version of the darkened image. Corresponding gray level histograms, cumulative histograms and co-occurrence matrices are plotted in the second, third and fourth rows respectively. Note that the co-occurrence matrices are almost the same for all three images.

orientation because an ideal co-occurrence matrix should be a diagonal matrix which results in zero variance for that distance and orientation. Other useful properties of the inertia measure was listed as defining the shape, size and orientation of parallelogram shaped tiles to represent periodic textures, embodying all the information contained in power spectrum, and being insensitive to the arbitrary selection of unit patterns.

Gotlieb and Kreyszig [23] used heuristic selection methods to select the best subset of features that can be computed from co-occurrence matrices. The heuristics they used are based on the idea that, when multiple texture labels are assigned to images in decreasing order of assignment probabilities, the correct texture label should be ranked at the top most of the times. They performed tests using small and homogeneous texture images and found that the variance feature performed the best, followed by the inverse difference moment and the entropy features.

3.5 Feature Selection

Here a problem arises as deciding on which distances to use to compute the co-occurrence matrices. Researchers tried to develop methods to select the co-occurrence matrices that reflect the greatest amount of texture information from a set of candidate matrices obtained by using different spatial relationships. In [56], Tou and Chang used an eigenvector-based approach and Karhunen-Loeve expansion to eliminate dependent features. Zucker and Terzopoulos [63] interpreted intensity pairs in an image as samples obtained from a two-dimensional random process and defined a chi-square test to determine whether their observed frequencies of occurrences appear to have been drawn from a distribution where two intensities are independent of each other.

We use the methods that will be described in Chapter 5 to select the distances that perform the best among distances of 1 to 20 pixels, according to our statistical

measures. Results of these tests are presented in Section 7.2.2.

3.6 Summary

The pre-processing and feature extraction steps for the variances of gray level spatial dependencies method are summarized in the object/process diagram in Figure 3.7. Given a sub-image, the features are computed using the following steps:

- Perform equal probability quantization.
- Compute gray level spatial dependency matrices for different distances and orientations.
- Compute variances of these matrices to form the feature vector.

Portions of this work was published in [4].

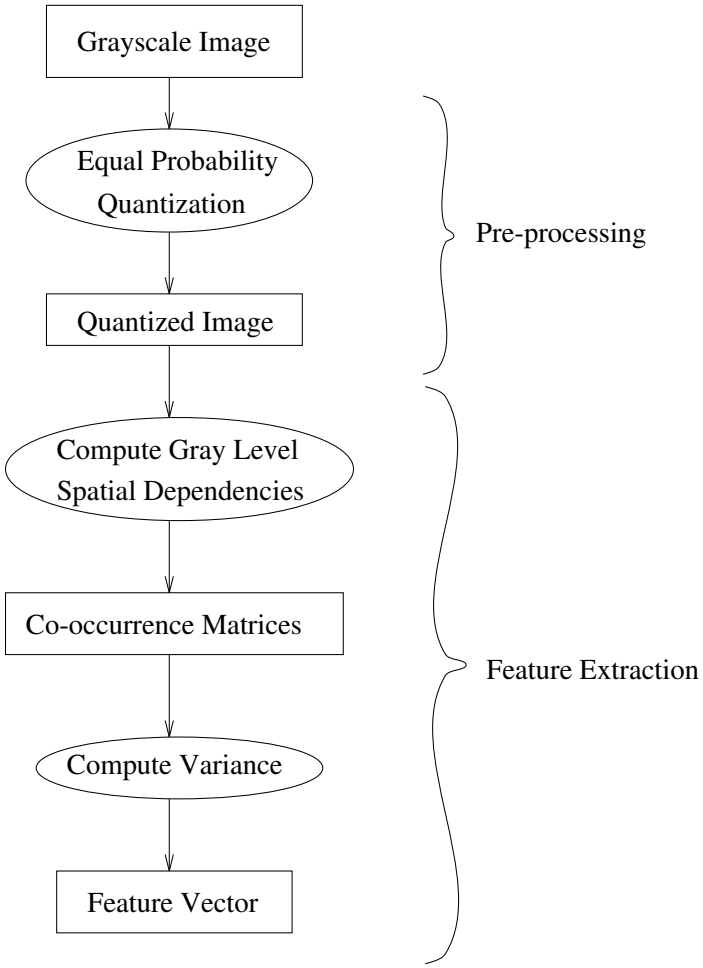


Figure 3.7: Object/process diagram for the pre-processing and feature extraction steps of the Variances of Gray Level Spatial Dependencies method.

Chapter 4

MULTI-SCALE TEXTURE ANALYSIS

In Chapters 2 and 3 we described how we assign a feature vector to each of the sub-images in the database using line-angle-ratio statistics and co-occurrence variances. In this chapter we will address the problem of combining these features in order to make use of their different advantages. A third feature extraction algorithm will also be introduced in order to capture the texture information that cannot be described by the first two set of features.

4.1 Motivation

Line-angle-ratio features that were described in Chapter 2 capture the global spatial organization in an image by using relative orientations of lines extracted from it, therefore they can be regarded as a macro-texture measure. Unfortunately, these features are not effective if the image does not have any line content. On the other hand, co-occurrence variances that were described in Chapter 3 capture local spatial variations of gray levels in the image. Therefore, these features are effective if the image is dominated by a fine, coarse, directional, or repetitive texture and can be regarded as a micro-texture measure.

Another important difference is that line-angle-ratio features are invariant to rotation because they use relative orientations. On the contrary, co-occurrence variances are not rotation invariant because they are angularly dependent. We can argue whether we want rotation invariance in a content-based retrieval system or not. One can say that a rotated image is not the same as the original image. For example,

people standing up and people lying down can be regarded as two different situations so these images can be perceived as quite different. Thus, rotation invariance may not be desirable. On the other hand, in a military target database we do not want to miss a tank when it is in a different orientation in an image in our database than its orientation in the query image. This dilemma is also present in object-based queries. In this work, we use the co-occurrence feature vector described in Chapter 3 which is rotation variant. If one wants rotation invariance for these features too, the feature vector can be modified as discussed in [27]. This modification procedure involves using mean, range and deviation of features for each distance over the four orientations as new features.

4.2 Orthogonal Differencing Kernels

When a large database contains different types of complex images, a multi-scale analysis is crucial for a general and compact representation. The sub-images in our database are of size 256×256. We assume that textures at a scale between micro and macro scales correspond to blob-like structures with sizes around 16×16.

If an image contains blob-like 16×16 structures, this information can be extracted using differencing kernels. We define two 1-D differencing kernels as

$$g_1 = [\ 1 \quad 1 \quad -1 \quad -1\ ], \qquad g_2 = [\ -1 \quad 1 \quad 1 \quad -1\ ]. \tag{4.1}$$

These kernels are orthogonal to each other and can be used to construct four 2-D

orthogonal kernels

$$k_1 = g_1' g_1 = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix}, \qquad k_2 = g_2' g_1 = \begin{bmatrix} -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \quad (4.2)$$

$$k_3 = g_1' g_2 = \begin{bmatrix} -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, \qquad k_4 = g_2' g_2 = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (4.3)$$

To detect 16×16 blobs using these 2-D kernels, first the image I is smoothed and sampled at 8×8 non-overlapping blocks. The resulting image is 32×32. Then, this image is filtered and sampled at 4×4 non-overlapping blocks using the kernels k_1, k_2, k_3 and k_4. We call the resulting 8×8 images I_1, I_2, I_3 and I_4 respectively. The algorithm is summarized in Figure 4.1 as stages composed of sequential filtering operations. Finally, we compute the sum of absolute values as a single feature f for the image I as

$$f = \sum_{i=1}^{4} \sum_{\substack{(r,c)\in \\ \{0\ldots7\}\times\{0\ldots7\}}} |I_i(r,c)|. \quad (4.4)$$

4.3 Combined Features

In this work, we append the line-angle-ratio features, co-occurrence features and differencing kernel features to obtain a single combined feature vector for each sub-image. Weighted combinations or even polynomial combinations [10] can also be used. In the rest of the thesis we will denote the size of a feature vector by Q, whether it is computed from line-angle-ratio statistics, co-occurrence variances, differencing kernels or it is the combined feature vector.

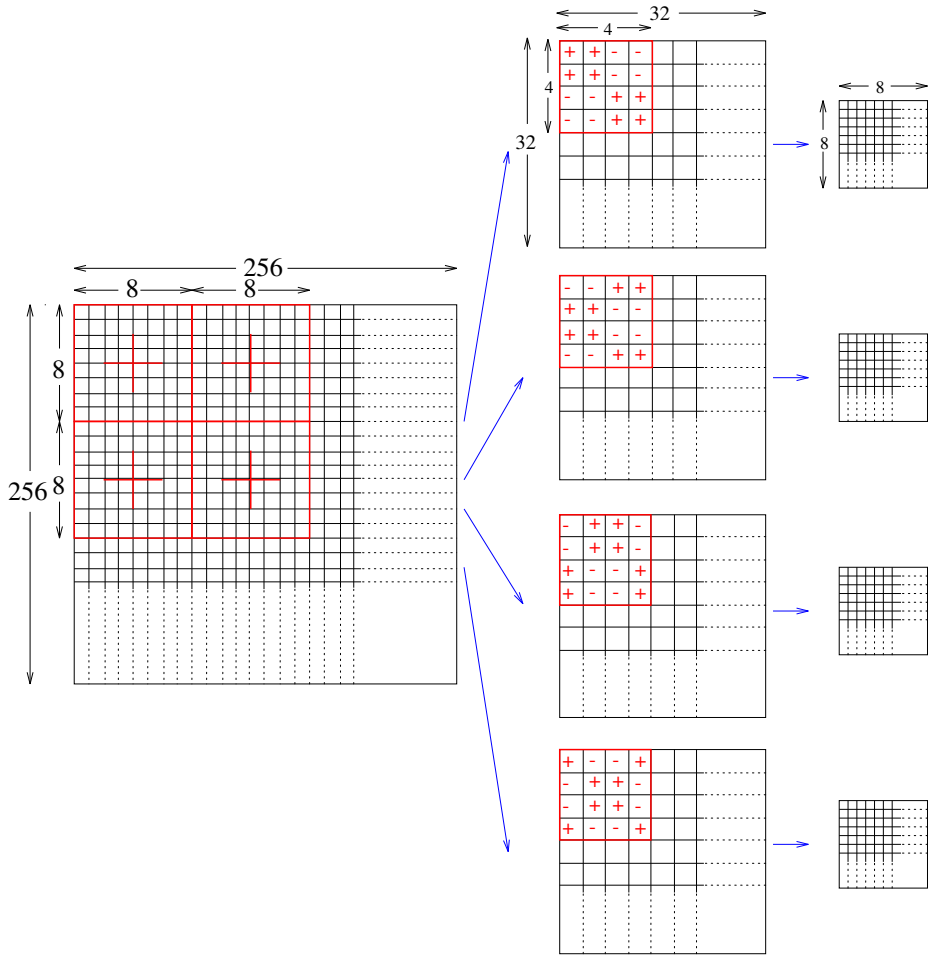


Figure 4.1: Orthogonal differencing kernels. From left to right, the first stage shows smoothing by 8×8 non-overlapping blocks. The second stage shows filtering by 4×4 non-overlapping blocks using the kernels k_1, k_2, k_3 and k_4. Final stage shows the resulting 8×8 images.

In the following section we describe how the individual features can be normalized to equalize their effects in the combined feature vector.

4.4 Feature Normalization

Not all features have the same range. In order to approximately equalize their ranges and make them have approximately the same effect in the computation of similarity, we have to normalize them. Given a Q-dimensional feature vector $f = [f_1 f_2 \cdots f_{\mathrm{Q}}]$, a set of lower bounds $l = [l_1 l_2 \cdots l_{\mathrm{Q}}]$ and a set of upper bounds $u = [u_1 u_2 \cdots u_{\mathrm{Q}}]$ for the components of f, we can obtain a normalized feature vector f' as

$$f'_q = \frac{f_q - l_q}{u_q - l_q}\,, \quad q = 1, \ldots, \mathrm{Q}. \tag{4.5}$$

This procedure results in features all being in the range $[0, 1]$. During database update, if any new feature f_q is out of the range $[l_q, u_q]$, the bounds for that q'th feature are updated as

$$\begin{aligned} l'_q &= f_q \quad \text{if } f_q < l_q \\ u'_q &= f_q \quad \text{if } f_q > u_q \end{aligned} \tag{4.6}$$

where l'_q and u'_q are the new bounds. We use this method to normalize the features used in the experiments in Chapter 7.

Another normalization procedure is to treat each component f_q as a random variable and transform it to another random variable with zero mean and unit variance as

$$f'_q = \frac{f_q - \mu_q}{\sigma_q}\,, \quad q = 1, \ldots, \mathrm{Q} \tag{4.7}$$

where μ_q and σ_q are the sample mean and the sample variance of the q'th component.

A third normalization procedure is to use the fact that $f'_q = F_{f_q}(f_q)$, where $F_{f_q}(\cdot)$ is the cumulative distribution function of f_q, makes f'_q a random variable uniformly distributed in the interval (0,1) (equal probability quantization).

4.5 Summary

In this chapter, we described how we combine the line-angle-ratio features of Chapter 2 and the co-occurrence features of Chapter 3 to make use of their different advantages. A third feature extraction algorithm was also described in order to capture the texture information that cannot be described by the first two features. A normalization scheme was used to equalize the effects of different features in the combined feature vector that was constructed by appending individual feature vectors. Portions of this work was published in [5].

Chapter 5

FEATURE SELECTION

In a content-based retrieval system, features that are used to represent images should have close values for similar images and significantly different values for dissimilar ones. In Chapter 1 we reviewed some of the features that have been used in content-based retrieval applications. These complex algorithms often require many parameters to be adjusted. Most of the times this feature selection process is done heuristically.

In this chapter we start with discussing some of the previous approaches to feature selection. This is followed by the description of a statistical framework to obtain the most useful subset of features among a larger set of possible ones that can be extracted using the algorithms described in Chapters 2, 3 and 4. Experiments performed using these feature extraction algorithms will be presented in Chapter 7. The ultimate goal of this work is to achive the best success rate while avoiding expensive and redundant computation.

5.1 Literature Overview and Motivation

In computer vision and pattern classification, researchers mostly concentrated on designing optimal classification procedures after feature extraction [15]. They have assumed that the selection of features is completely pre-determined by the designer. One of the main reasons why a smaller but more effective subset of features is not sought is that formulating a statistical feature selection problem is often impossible because the probability distributions of the features may not be known or an opti-

mization problem involving "goodness" of features as an objective function is hard to formulate. It is also possible that the available feature set is already small and reducing the dimension is not practical at all.

In many complex feature extraction algorithms, there are many parameters that, when varied, result in a large number of possible feature measurements. These high dimensional feature spaces may cause a problem of having less significant or even redundant features that contribute very little in the decision process.

Early works on statistical pattern recognition include many examples of algorithms for feature selection. Researchers tried to form a new set of features from a set of available ones either by selecting a subset or by combining them into new features. To solve the problem of selecting the "best" subset of features, Chien [15] proposed a sequential selection algorithm that successively selects one of the finite number of pre-determined feature sets in each iteration according to the previous classification results. He proved that this procedure converges to the best performing subset as a limit but the success of his algorithm depends on the selection of initial subsets. To reduce the number of subsets evaluated, Whitney [60] used a suboptimum search procedure that first selects the best single measurement, then selects the best pair that includes the best single measurement that was already selected, and continues by adding a single measurement that appears to be the best when combined with the previously selected subset of measurements. Another approach is Narendra and Fukunaga's [45] branch and bound algorithm. They described an algorithm that selects the best subset of a feature set with guaranteed global optimality of any criterion that satisfies monotonicity, without extensive search. They also discussed suboptimal variants of the algorithm that are easier to compute by compromising optimality. Jain and Dubes [30] defined the goal as to generate a set of weakly correlated primitive features which discriminate well among the pattern classes and described a two phase algorithm that first creates subsets of potential features by clustering and then reduces each subset into a single primitive feature by cluster

compression.

Only a few researchers presented feature selection algorithms in their papers on database retrieval. Among the ones reviewed in Chapter 1, Manjunath and Ma [42] used total difference energy within the spectral coverage to select among many possible Gabor filters and Carson *et al.* [13] used the minimum description length principle to select the number of Gaussians that best model the feature space. Other works that presented some kind of feature selection are [47, 29, 39].

In our work, in order to find statistical measures of how well some of the features perform better than others, we use a two-class pattern classification approach. In doing so, we define two classes, namely the relevance class $\mathcal{A}$ and the irrelevance class $\mathcal{B}$, in order to classify image pairs as similar or dissimilar. Given a pair of images, if these images are similar, they should be assigned to the relevance class, if not, they should be assigned to the irrelevance class.

In the following sections, first we describe a protocol to automatically construct groundtruth image pairs and then we discuss the decision rule and an experimental procedure for classification.

5.2 Automatic Groundtruth Construction

The protocol for constructing the groundtruth to determine the parameters of the relevance and irrelevance classes involves making up two different sets of sub-images for each image i, $i = 1, \ldots, \mathrm{I}$, in the database. The first set of sub-images begins in row 0 column 0 and partitions each image i into M_i $\mathrm{K} \times \mathrm{K}$ sub-images. These sub-images are partitioned such that they overlap by half the area. We ignore the partial sub-images on the last group of columns and last group of rows which cannot make up the $\mathrm{K} \times \mathrm{K}$ sub-images. This set of sub-images will be referred as the *main database* in the rest of the thesis.

The second set of sub-images are shifted versions of the ones in the main database.

They begin in row K/4 and column K/4 and partition the image i into N_i K $\times$ K sub-images. We again ignore the partial sub-images on the last group of columns and last group of rows which cannot make K $\times$ K sub-images. This second set of sub-images will be referred as the *test database* in the rest of the thesis.

To construct the groundtruth to determine the parameters, we record the relationships of the shifted sub-images in the test database with the sub-images in the main database that were computed from the same image. The feature vector for each sub-image in the test database is strongly related to four feature vectors in the main database in which the sub-image overlap is 9/16 of the sub-image area. From these relationships, we establish a *strongly related sub-images* set $R_s(n)$ for each sub-image n where $n = 1, \ldots, \mathrm{N}_i$.

We assume that, in an image, two sub-images that do not overlap are usually not relevant. From this assumption, we randomly select four sub-images that have no overlap with the sub-image n. These four sub-images form the *other sub-images* set $R_o(n)$.

These groundtruth sub-image pairs constitute the relevance class $\mathcal{A}_i$,

$$\mathcal{A}_i = \{(n, m) |\, m \in R_s(n)\,,\, n = 1, \ldots, \mathrm{N}_i\}, \tag{5.1}$$

and the irrelevance class $\mathcal{B}_i$,

$$\mathcal{B}_i = \{(n, m) |\, m \in R_o(n)\,,\, n = 1, \ldots, \mathrm{N}_i\} \tag{5.2}$$

for each image i. Then, the overall relevance class becomes

$$\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \cdots \cup \mathcal{A}_{\mathrm{I}} \tag{5.3}$$

and the overall irrelevance class becomes

$$\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_{\mathrm{I}}. \tag{5.4}$$

An example for the overlapping concept is given in Figure 5.1 where the shaded region shows the 9/16 overlapping. For K = 128, sub-images with upper-left corners

at (0,0), (0,64), (64,0), (64,64) and (192,256) are examples from the main database. The sub-image with upper-left corner at (32,32) is a sub-image in the test database. For this sub-image, R_s will consist of the sub-images at (0,0), (0,64), (64,0), and (64,64) because they overlap by the required amount. On the other hand, R_o will consist of four randomly selected sub-images, one being the sub-image at (192,256) for example, which are not in R_s and have no overlap with the test sub-image. The pairs formed by the test sub-image and the ones in R_s and R_o form the groundtruth for the relevance class $\mathcal{A}$ and the irrelevance class $\mathcal{B}$ respectively. Note that for any sub-image which is not one of the sub-images shifted by (K/4,K/4), there is a sub-image in the main database which has an overlap of more than half the area so this 9/16 overlap is the worst case. Same concepts are also illustrated on an image in Figure 5.2.

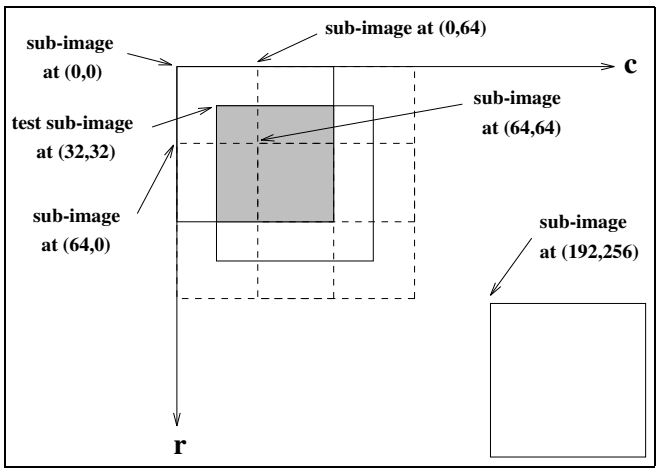


Figure 5.1: The shaded region shows the 9/16 overlapping between two sub-images. For K = 128, sub-images with upper-left corners at (0,0), (0,64), (64,0) and (64,64) form the set of *strongly related sub-images*, R_s, for the test sub-image at (32,32). *Other sub-images* set consists of four randomly selected sub-images which may include the one at (192,256).

In order to estimate the distribution of the relevance class, we first compute the

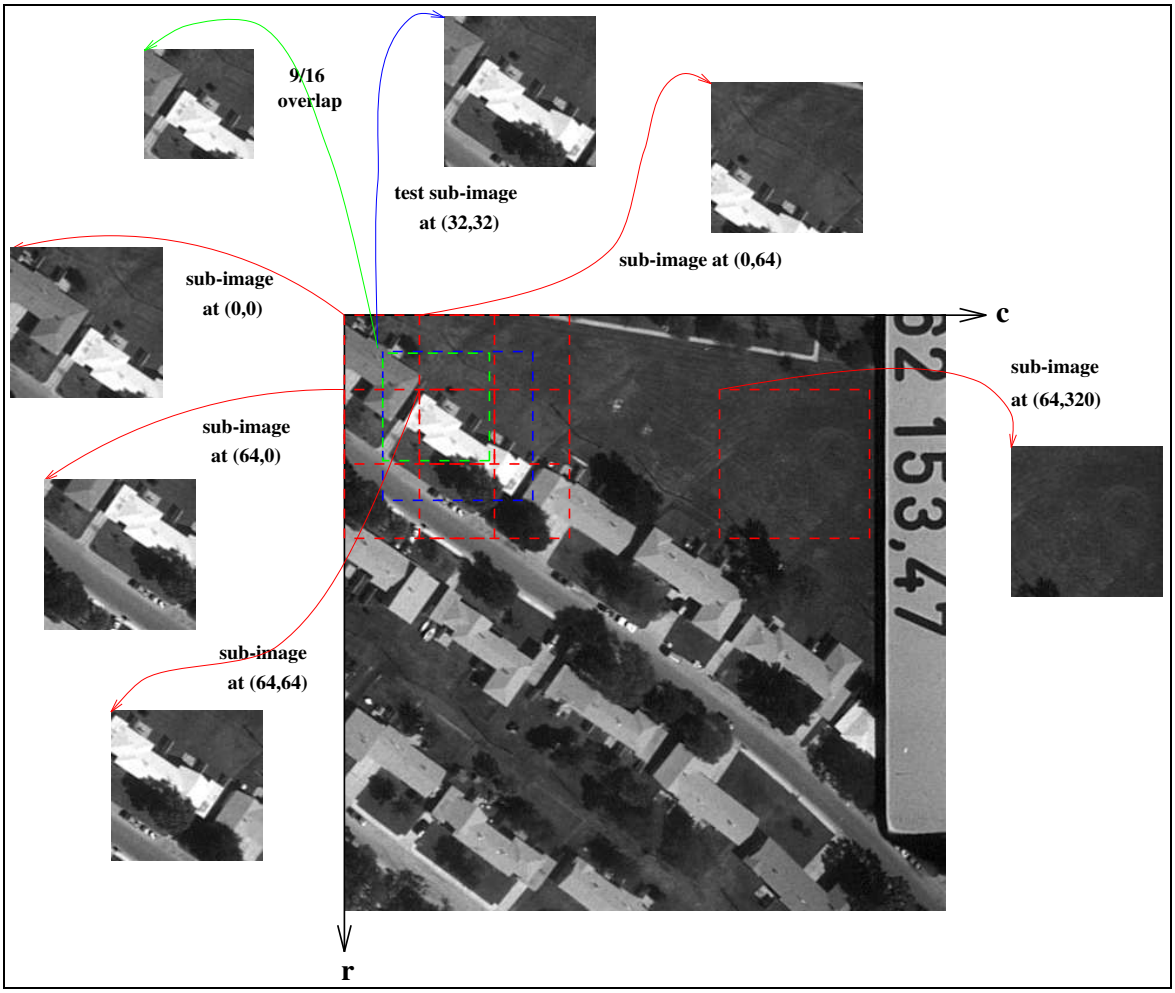


Figure 5.2: Overlapping sub-images are illustrated on an image. Sub-images at (0,0), (0,64), (64,0) and (64,64) form the set of *strongly related sub-images* for the test sub-image at (32,32). The 9/16 overlap between the ones at (0,0) and (32,32) is also illustrated.

differences d,

$$d = x^{(n)} - y^{(m)}, \quad (n,m) \in \mathcal{A},\ x^{(n)}, y^{(m)} \in \mathbf{R}^{\mathrm{Q}} \tag{5.5}$$

where Q is the number of features, and $x^{(n)}$ and $y^{(m)}$ are the feature vectors of sub-images n and m respectively. Since components of $x^{(n)}$ and $y^{(m)}$ are in the range [0,1], components of d will be in the range [-1,1]. Then, we compute the sample mean, $\mu_{\mathcal{A}}$, and the sample covariance, $\Sigma_{\mathcal{A}}$, of these differences. We assume that the differences for the relevance class have a normal distribution with mean $\mu_{\mathcal{A}}$, and covariance matrix $\Sigma_{\mathcal{A}}$,

$$f(d|\mu_{\mathcal{A}}, \Sigma_{\mathcal{A}}) = \frac{1}{(2\pi)^{\mathrm{Q}/2}|\Sigma_{\mathcal{A}}|^{1/2}}\, \mathrm{e}^{-(\mathrm{d}-\mu_{\mathcal{A}})'\Sigma_{\mathcal{A}}^{-1}(\mathrm{d}-\mu_{\mathcal{A}})/2}. \tag{5.6}$$

Similarly, we compute the differences d,

$$d = x^{(n)} - y^{(m)}, \quad (n,m) \in \mathcal{B},\ x^{(n)}, y^{(m)} \in \mathbf{R}^{\mathrm{Q}}, \tag{5.7}$$

then the sample mean, $\mu_{\mathcal{B}}$, and the sample covariance matrix, $\Sigma_{\mathcal{B}}$, for the irrelevance class. Then, the density for this class becomes

$$f(d|\mu_{\mathcal{B}}, \Sigma_{\mathcal{B}}) = \frac{1}{(2\pi)^{\mathrm{Q}/2}|\Sigma_{\mathcal{B}}|^{1/2}}\, \mathrm{e}^{-(\mathrm{d}-\mu_{\mathcal{B}})'\Sigma_{\mathcal{B}}^{-1}(\mathrm{d}-\mu_{\mathcal{B}})/2}. \tag{5.8}$$

The automatic groundtruth construction protocol is summarized in the object/process diagram in Figure 5.3.

5.3 Classification Tests

In the previous section we constructed groundtruth image pairs for the relevance and irrelevance classes. Since we know which non-shifted sub-images and shifted sub-images overlap, we also know which sub-image pairs should be assigned to the relevance class $\mathcal{A}$ and which to the irrelevance class $\mathcal{B}$. So, to test the classification effectiveness of the features, we check whether each pair that should be classified into class $\mathcal{A}$ or $\mathcal{B}$ is classified into class $\mathcal{A}$ or $\mathcal{B}$ correctly.

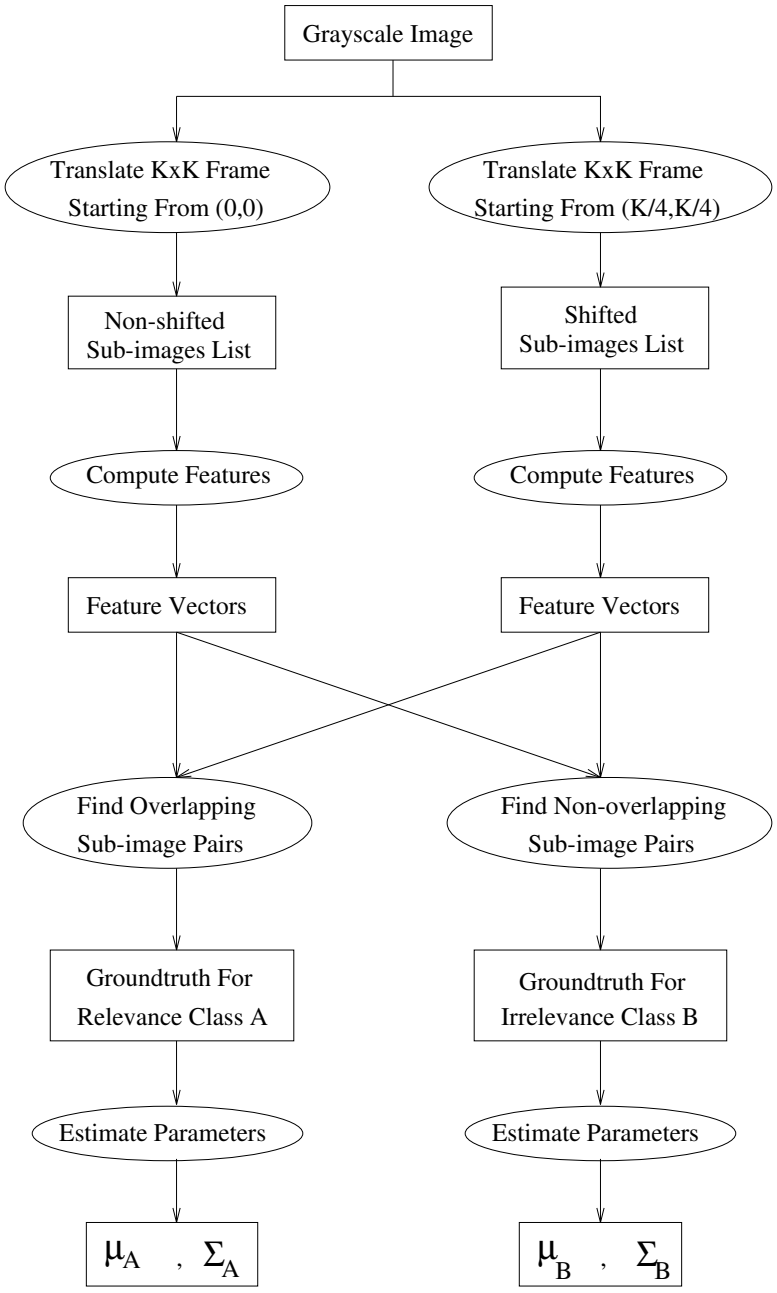


Figure 5.3: Object/process diagram for the automatic groundtruth construction protocol.

As the classifier, the Bayesian decision rule is used. In the following sections, we describe first the decision rule and then the experimental set-up. In the derivations below we assume the differences of features in the Q-dimensional feature vectors are independent and prior probabilities for both of the classes are equal.

5.3.1 Decision Rule

Given a groundtruth sub-image pair (n,m) as in equations (5.1) or (5.2) and their Q-dimensional feature vectors $x^{(n)}$ and $y^{(m)}$ respectively, first the difference

$$d = x^{(n)} - y^{(m)}\,, \quad x^{(n)}, y^{(m)} \in \mathbf{R}^{\mathrm{Q}} \tag{5.9}$$

is computed. The probability that these sub-images are relevant is

$$P(\mathcal{A}|d) = P(d|\mathcal{A})P(\mathcal{A})/P(d) \tag{5.10}$$

and that they are irrelevant is

$$P(\mathcal{B}|d) = P(d|\mathcal{B})P(\mathcal{B})/P(d). \tag{5.11}$$

The sub-image pair is assigned to the relevance class if $P(\mathcal{A}|d) > P(\mathcal{B}|d)$, and to the irrelevance class otherwise. This can be written as a ratio

$$\frac{P(\mathcal{A}|d)}{P(\mathcal{B}|d)} > 1. \tag{5.12}$$

Since we assume two classes are equally likely, (5.12) becomes the likelihood ratio

$$\begin{aligned} \frac{P(d|\mathcal{A})}{P(d|\mathcal{B})} &= \frac{P(d|\mu_{\mathcal{A}}, \Sigma_{\mathcal{A}})}{\mathrm{P(d}|\mu_{\mathcal{B}}, \Sigma_{\mathcal{B}})} \\ &= \frac{\frac{1}{(2\pi)^{\mathrm{Q}/2}|\Sigma_{\mathcal{A}}|^{1/2}} e^{-(d-\mu_{\mathcal{A}})'\Sigma_{\mathcal{A}}^{-1}(\mathrm{d}-\mu_{\mathcal{A}})/2}}{\frac{1}{(2\pi)^{\mathrm{Q}/2}|\Sigma_{\mathcal{B}}|^{1/2}} e^{-(d-\mu_{\mathcal{B}})'\Sigma_{\mathcal{B}}^{-1}(\mathrm{d}-\mu_{\mathcal{B}})/2}} \\ &> 1. \end{aligned} \tag{5.13}$$

After taking the natural logarithm of (5.13) as

$$\ln \frac{1}{|\Sigma_{\mathcal{A}}|^{1/2}} - (d-\mu_{\mathcal{A}})'\Sigma_{\mathcal{A}}^{-1}(\mathrm{d}-\mu_{\mathcal{A}})/2 - \ln \frac{1}{|\Sigma_{\mathcal{B}}|^{1/2}} + (\mathrm{d}-\mu_{\mathcal{B}})'\Sigma_{\mathcal{B}}^{-1}(\mathrm{d}-\mu_{\mathcal{B}})/2 > 0, \tag{5.14}$$

we obtain

$$(d-\mu_{\mathcal{A}})'\Sigma_{\mathcal{A}}^{-1}(\mathrm{d}-\mu_{\mathcal{A}})/2 < (\mathrm{d}-\mu_{\mathcal{B}})'\Sigma_{\mathcal{B}}^{-1}(\mathrm{d}-\mu_{\mathcal{B}})/2 + \ln\frac{|\Sigma_{\mathcal{B}}|^{1/2}}{|\Sigma_{\mathcal{A}}|^{1/2}} \tag{5.15}$$

Since we assumed that the features are independent, the covariance matrices $\Sigma_{\mathcal{A}}$ and $\Sigma_{\mathcal{B}}$ contain only the variances

$$\Sigma_{\mathcal{A}} = \begin{pmatrix} \sigma^2_{\mathcal{A}_1} & 0 & \cdots & 0 \\ 0 & \sigma^2_{\mathcal{A}_2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma^2_{\mathcal{A}_{\mathrm{Q}}} \end{pmatrix}, \quad \Sigma_{\mathcal{B}} = \begin{pmatrix} \sigma^2_{\mathcal{B}_1} & 0 & \cdots & 0 \\ 0 & \sigma^2_{\mathcal{B}_2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma^2_{\mathcal{B}_{\mathrm{Q}}} \end{pmatrix}. \tag{5.16}$$

Then, (5.15) can be simplified as

$$\sum_{q=1}^{\mathrm{Q}} \left(\frac{d_q-\mu_{\mathcal{A}q}}{\sigma_{\mathcal{A}_q}}\right)^2 < \sum_{q=1}^{\mathrm{Q}} \left(\frac{d_q-\mu_{\mathcal{B}q}}{\sigma_{\mathcal{B}_q}}\right)^2 + 2\sum_{q=1}^{\mathrm{Q}} \ln\frac{\sigma_{\mathcal{B}_q}}{\sigma_{\mathcal{A}_q}}. \tag{5.17}$$

As a result, if the difference d of the feature vectors of two sub-images satisfy the inequality in (5.17), this sub-image pair is assigned to the relevance class, otherwise it is assigned to the irrelevance class.

To determine relative effectiveness of different features, a subset U of Q features can be used for classification. Then, the decision rule becomes

$$\sum_{q\in\mathrm{U}} \left(\frac{d_q-\mu_{\mathcal{A}q}}{\sigma_{\mathcal{A}_q}}\right)^2 < \sum_{q\in\mathrm{U}} \left(\frac{d_q-\mu_{\mathcal{B}q}}{\sigma_{\mathcal{B}_q}}\right)^2 + 2\sum_{q\in\mathrm{U}} \ln\frac{\sigma_{\mathcal{B}_q}}{\sigma_{\mathcal{A}_q}}. \tag{5.18}$$

5.3.2 Experimental Set-up

To check the effectiveness of the features, groundtruth sub-image pairs that were constructed in Section 5.2 are classified into the relevance or irrelevance classes using the decision rule in (5.18). Suitable measures for the classification results are misdetection, false alarm, total cost and total success. In content-based retrieval we are more concerned with misdetection because we want to retrieve all the images similar to the query image. But false alarm rate is also important because the purpose of

querying a database is to retrieve similar images only, not all of them. In our tests total cost is defined as 3 misdetection and 2 false alarm and is used as the criterion for "goodness", i.e. if a subset of features has a small total cost compared to others, it is called "good".

If the dimension of the feature space is large, it is computationally too expensive to do classification tests using all possible subsets of these features. In our work, first, we do tests using only one of the features at a time. Although combinations of features carry more information than individual features, these tests will help us see which features are significantly better or worse than others.

The second test is done as follows. First, the total cost using all Q features is computed. The feature with the worst total cost, compared to the cost using all Q, is discarded and the total cost using the remaining Q-1 features is computed. Then, the worst feature among the remaining Q-1 features is discarded and the total cost using Q-2 features is computed. This procedure continues until one feature is left.

A third test is done by starting with the total cost for each individual feature. First, the best one is selected. Given the best one, pairs of features are formed using one of the remaining features and this best feature. Total cost is computed for each pair and the one having the smallest cost is selected. Given the best two features, next, triplets of features are formed using one of the remaining features and these two best features. Total cost is computed for each triplet and the one having the smallest cost is selected. This procedure continues until all or a preselected number of features are used [60].

These tests do not guarantee the optimal subset of features but allow us to select a suboptimal subset without doing an exhaustive search. Results of these tests for each of the feature extraction algorithms will be presented after database population in Chapter 7.

5.4 Validation of Labels in Automatically Constructed Groundtruth

In Section 5.2, we described an automatic groundtruth construction protocol that assigns labels to sub-image pairs according to the assumptions that overlapping sub-images are relevant and non-overlapping ones are irrelevant. However, we cannot expect that these assumptions will always hold, especially when the images in the database are very complex. Since the relevance and irrelevance class parameters as well as the success and error rates in the classification tests depend heavily on the selection of these groundtruth sub-image pairs, evaluation of the results require compensation of the effect of "learning from an imperfect teacher" [3, 52, 40].

A statistical framework to validate the labels that are assigned to sub-image pairs by the automatic groundtruth construction protocol in Section 5.2 and to estimate the true results of the Gaussian classifier in Section 5.3.1 can be defined as follows. Let the relevance class be $\mathcal{A}$ and the irrelevance class be $\mathcal{B}$. Let the true label for a pair of sub-images be $\mathbf{T}$. Let the label that is assigned by the automatic groundtruth construction protocol be $\mathbf{L}$ and the label that is assigned by the Gaussian classifier be $\mathbf{C}$. In the previous section, success and error rates were computed using the probabilities $P(\mathbf{L} = \mathcal{A}, \mathbf{C} = \mathcal{A})$, $P(\mathbf{L} = \mathcal{A}, \mathbf{C} = \mathcal{B})$, $P(\mathbf{L} = \mathcal{B}, \mathbf{C} = \mathcal{A})$ and $P(\mathbf{L} = \mathcal{B}, \mathbf{C} = \mathcal{B})$ as shown in Table 5.1, in other words, labels assigned by the classifier were compared to the labels of the automatically constructed groundtruths. We want to know the true success and error rates so we want to estimate the probabilities $P(\mathbf{T} = \mathcal{A}, \mathbf{C} = \mathcal{A})$, $P(\mathbf{T} = \mathcal{A}, \mathbf{C} = \mathcal{B})$, $P(\mathbf{T} = \mathcal{B}, \mathbf{C} = \mathcal{A})$ and $P(\mathbf{T} = \mathcal{B}, \mathbf{C} = \mathcal{B})$, in other words, we want to compare classifier labels to the true labels.

Let's consider $P(\mathbf{T} = \mathcal{A}, \mathbf{C} = \mathcal{A})$, which is the probability that actually relevant $(\mathbf{T} = \mathcal{A})$ pairs of sub-images being assigned to the relevance class by the Gaussian classifier $(\mathbf{C} = \mathcal{A})$. This can be written as

$$P(\mathbf{T} = \mathcal{A}, \mathbf{C} = \mathcal{A}) = P(\mathbf{T} = \mathcal{A}, \mathbf{L} = \mathcal{A}, \mathbf{C} = \mathcal{A}) + P(\mathbf{T} = \mathcal{A}, \mathbf{L} = \mathcal{B}, \mathbf{C} = \mathcal{A}). \tag{5.19}$$

Table 5.1: Confusion matrix for the classification tests using the Gaussian classifier.

	Assigned to $\mathcal{A}$	Assigned to $\mathcal{B}$
G.truth $\mathcal{A}$	$P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{A})$	$P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{B})$
G.truth $\mathcal{B}$	$P(\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{A})$	$P(\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{B})$

But $P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{A})$ is equivalent to

$$P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{A}) = P(\mathbf{T}=\mathcal{A},\mathbf{C}=\mathcal{A}|\mathbf{L}=\mathcal{A})P(\mathbf{L}=\mathcal{A}). \tag{5.20}$$

We assume that $P(\mathbf{T}=\mathcal{A}|\mathbf{L}=\mathcal{A})$ and $P(\mathbf{C}=\mathcal{A}|\mathbf{L}=\mathcal{A})$ are independent because former is effected by the mislabeling of the automatic groundtruth construction protocol while latter is effected by the errors in the classifier. Then,

$$\begin{aligned} P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{A}) &= P(\mathbf{T}=\mathcal{A}|\mathbf{L}=\mathcal{A})P(\mathbf{C}=\mathcal{A}|\mathbf{L}=\mathcal{A})P(\mathbf{L}=\mathcal{A}) \\ &= \frac{P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{A})}{P(\mathbf{L}=\mathcal{A})}\frac{P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{A})}{P(\mathbf{L}=\mathcal{A})}P(\mathbf{L}=\mathcal{A}) \\ &= \frac{P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{A})P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{A})}{P(\mathbf{L}=\mathcal{A})}. \end{aligned} \tag{5.21}$$

After making similar derivations for $P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{A})$, we obtain

$$\begin{aligned} P(\mathbf{T}=\mathcal{A},\mathbf{C}=\mathcal{A}) =& \frac{P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{A})P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{A})}{P(\mathbf{L}=\mathcal{A})} + \\ & \frac{P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{B})P(\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{A})}{P(\mathbf{L}=\mathcal{B})}. \end{aligned} \tag{5.22}$$

Similarly,

$$\begin{aligned} P(\mathbf{T}=\mathcal{A},\mathbf{C}=\mathcal{B}) =& \frac{P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{A})P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{B})}{P(\mathbf{L}=\mathcal{A})} + \\ & \frac{P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{B})P(\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{B})}{P(\mathbf{L}=\mathcal{B})}, \end{aligned} \tag{5.23}$$

$$\begin{aligned} P(\mathbf{T}=\mathcal{B},\mathbf{C}=\mathcal{A}) =& \frac{P(\mathbf{T}=\mathcal{B},\mathbf{L}=\mathcal{A})P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{A})}{P(\mathbf{L}=\mathcal{A})} + \\ & \frac{P(\mathbf{T}=\mathcal{B},\mathbf{L}=\mathcal{B})P(\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{A})}{P(\mathbf{L}=\mathcal{B})}, \end{aligned} \tag{5.24}$$

$$P(\mathbf{T}=\mathcal{B},\mathbf{C}=\mathcal{B}) = \frac{P(\mathbf{T}=\mathcal{B},\mathbf{L}=\mathcal{A})P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{B})}{P(\mathbf{L}=\mathcal{A})} + \frac{P(\mathbf{T}=\mathcal{B},\mathbf{L}=\mathcal{B})P(\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{B})}{P(\mathbf{L}=\mathcal{B})}. \tag{5.25}$$

The probabilities $P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{A})$, $P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{B})$, $P(\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{A})$ and $P(\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{B})$ can be observed from the classification experiments as shown in Table 5.1. Similarly $P(\mathbf{L}=\mathcal{A})$ and $P(\mathbf{L}=\mathcal{B})$ can be observed as

$$P(\mathbf{L}=\mathcal{A}) = P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{A}) + P(\mathbf{L}=\mathcal{A},\mathbf{C}=\mathcal{B}) \tag{5.26}$$

$$P(\mathbf{L}=\mathcal{B}) = P(\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{A}) + P(\mathbf{L}=\mathcal{B},\mathbf{C}=\mathcal{B}). \tag{5.27}$$

To determine the probabilities $P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{A})$, $P(\mathbf{T}=\mathcal{A},\mathbf{L}=\mathcal{B})$, $P(\mathbf{T}=\mathcal{B},\mathbf{L}=\mathcal{A})$ and $P(\mathbf{T}=\mathcal{B},\mathbf{L}=\mathcal{B})$ which correspond to the mislabeling and correct labeling probabilities of the automatic groundtruth construction protocol, we can manually sample some of the sub-image pairs and estimate these probabilities using frequencies of the corresponding cases. As a result, $P(\mathbf{T}=\mathcal{A},\mathbf{C}=\mathcal{A})$, $P(\mathbf{T}=\mathcal{A},\mathbf{C}=\mathcal{B})$, $P(\mathbf{T}=\mathcal{B},\mathbf{C}=\mathcal{A})$ and $P(\mathbf{T}=\mathcal{B},\mathbf{C}=\mathcal{B})$ give an estimate of the true classification results, which correspond to the comparison of the labels assigned by the classifier to the true labels.

5.5 Summary

In this chapter, we addressed the problem of selecting the features that perform the best according to the criteria that "good" features should result in small classification errors. First, we defined two classes, the relevance class and the irrelevance class. Then, we described a protocol that translates a frame throught every image and constructs groundtruth sub-image pairs using the assumptions that overlapping sub-images are relevant and non-overlapping ones are irrelevant. We defined a Gaussian classifier that assigns a pair of images to the relevance class if they are similar and to the irrelevance class if they are not. The classification error was defined as 3

misdetections and 2 false alarms for this two-class classification problem. Finally, we discussed a statistical framework that estimates the correct classification results from the decisions made by the classifier and the mislabeling probabilities of the automatic groundtruth construction protocol. This is often referred to as "learning from an imperfect teacher" in the pattern recognition literature.

Chapter 6

DECISION METHODS

After computing the feature vectors for all images in the database, given a query image, we have to decide which images in the database are relevant to it and we have to retrieve the most relevant ones as the results of the query. A similarity measure for content-based retrieval should be efficient enough to match similar images as well as being able to discriminate dissimilar ones.

Here we want to note a problem that may arise when we are given the unnormalized feature vector $f = [f_1 f_2 \cdots f_{\mathrm{Q}}]$ of a query image. Given the lower and upper bounds $[l_q, u_q]$, each component f_q of the query feature vector is normalized as

$$f'_q = \begin{cases} 0 & \text{if } f_q < l_q \\ 1 & \text{if } f_q > u_q \\ \frac{f_q - l_q}{u_q - l_q} & \text{otherwise.} \end{cases} \tag{6.1}$$

In our experiments we use two different types of decision methods; a likelihood ratio which is a Gaussian classifier, and a nearest neighbor classifier. In the following sections, first we give a brief review of the decision methods used in the database retrieval literature, then we discuss the decision methods that we use.

6.1 Literature Overview

Nearest neighbor rule has been the most widely used decision method in content-based retrieval literature. The main reason for this is the non-parametric nature of this method. We do not need to make any assumptions about the probability

distribution of the features, which is especially hard when the feature space is high-dimensional.

In the nearest neighbor rule, the Euclidean distance was usually used as the distance measure. Among the image retrieval systems that were reviewed in Chapter 1, [37] used the Euclidean distance and [22, 47, 13] used a weighted Euclidean distance. [34] used the difference between two sets of sums of weighted Gaussians, [8, 42] used sum of absolute distances normalized using standard deviations, [29] compared how many wavelet coefficients that two images have in common and [54] used different distance measures like the L^1 norm, Euclidean distance, quadratic distances and binary set intersections.

6.2 Likelihood Ratio - Gaussian Classifier

In Section 5.3 we developed a Gaussian classifier which tries to classify a pair of images into the relevance and irrelevance classes according to a likelihood ratio. Suppose for the moment that the user query is a $\mathrm{K} \times \mathrm{K}$ image. First, its feature vector x is determined. Then, the search goes through all the feature vectors $y^{(m)}$ in the main database where $m = 1, \ldots, (\sum_{i=1}^{\mathrm{I}} \mathrm{M}_i)$, M_i being the number of sub-images in the i'th image and I being the total number of images. For each feature vector pair (x, $y^{(m)}$) where $x, y^{(m)} \in \mathbf{R}^{\mathrm{Q}}$, the difference $d^{(m)} = x - y^{(m)}$ is computed.

The likelihood ratio to classify image pairs to relevance or irrelevance classes was given in (5.13). In order to rank the sub-images in the database according to the likelihood ratio, we use (5.14) as

$$r(d) = (d - \mu_{\mathcal{A}})' \Sigma_{\mathcal{A}}^{-1} (\mathrm{d} - \mu_{\mathcal{A}}) - (\mathrm{d} - \mu_{\mathcal{B}})' \Sigma_{\mathcal{B}}^{-1} (\mathrm{d} - \mu_{\mathcal{B}}) \tag{6.2}$$

after eliminating the constant terms. Note that here we do not make the assumption that feature differences are independent and use the covariances instead of using only the variances. To find the sub-images that are relevant to the input query, $r(d^{(m)})$ in (6.2) is computed for all sub-images in the database and they are ranked in ascending

order using these values. This ranking is equivalent to ranking in descending order using the likelihood ratios in (5.13) but is more efficient to compute. At the end, the top k sub-images are retrieved as the most relevant ones, where k being a user-selected parameter.

6.3 Nearest Neighbor Rule With Modified Distance

6.3.1 Nearest Neighbor Rule

In the nearest neighbor decision rule, each sub-image m in the database is assumed to be represented by its feature vector $y^{(m)}$ in the Q-dimensional feature space. Given the feature vector x for the input query, we want to find the y's which are the closest neighbors of x according to a distance measure. Then, the k-nearest neighbors of x will be retrieved as the most relevant ones.

The problem of finding the k-nearest neighbors can be formulated as follows. Given the set $Y = \{y^{(m)} | y^{(m)} \in \mathbf{R}^{\mathrm{Q}}, m = 1, \ldots, \mathrm{M}\}$ and feature vector $x \in \mathbf{R}^{\mathrm{Q}}$, find the set of sub-images $P \subseteq \{1, \ldots, \mathrm{M}\}$ such that $\#P = k$ and

$$\rho(x, y^{(p)}) \leq \rho(x, y^{(r)}) \ , \quad \forall p \in P, r \in \{1, \ldots, \mathrm{M}\} \backslash P \tag{6.3}$$

where $\mathrm{M} = \sum_{i=1}^{\mathrm{I}} \mathrm{M}_i$, M_i being the number of sub-images in the i'th image and I being the total number of images. Then, images with id.'s in the set P are retrieved as the results of the query.

For the distance metric ρ, we use the Minkowsky L^1 norm

$$\rho(x, y) = \sum_{q=1}^{\mathrm{Q}} |x_q - y_q|, \tag{6.4}$$

the Euclidean distance (Minkowsky L^2 norm)

$$\begin{aligned} \rho(x, y) &= \|x - y\| \\ &= \sqrt{\sum_{q=1}^{\mathrm{Q}} (x_q - y_q)^2} \end{aligned} \tag{6.5}$$

or the infinity norm

$$\rho(x, y) = \max_{q=1,\ldots,\mathrm{Q}} |x_q - y_q| \tag{6.6}$$

where $x, y \in \mathbf{R}^{\mathrm{Q}}$ and x_q and y_q are the q'th components of the feature vectors x and y respectively.

6.3.2 Weighted Features

Although ranges of the features are normalized using the procedure discussed in Section 4.4, we may also want to weight the features for similarity computation. Given two feature vectors x and y, the weighted L^1 norm can be defined as

$$\rho'(x, y; w) = \sum_{q=1}^{\mathrm{Q}} |\,(x_q - y_q)\, w_q|, \tag{6.7}$$

the weighted Euclidean distance can be defined as

$$\begin{aligned} \rho'(x, y; w) &= \|(x - y)'w\| \\ &= \sqrt{\sum_{q=1}^{\mathrm{Q}} \left[(x_q - y_q)\, w_q\right]^2} \end{aligned} \tag{6.8}$$

and the weighted infinity norm can be defined as

$$\rho'(x, y; w) = \max_{q=1,\ldots,\mathrm{Q}} |\,(x_q - y_q)\, w_q| \tag{6.9}$$

where $x, y, w \in \mathbf{R}^{\mathrm{Q}}$.

One possible weighting method is to allow the user to adjust the weights for different features. Most of the time the user is not fully aware of the effective ranges and also meanings of the features so an alternative is to use weights automatically trained from the images in the database. In pattern classification, a feature can be called a "good" feature if its within-class variance is small and between-class variance is large. In Section 5.3 a maximum likelihood classifier is designed for the relevance and the irrelevance classes. Variances that were trained for these classes can be used as

weights to take into account the "goodness" of the features in similarity computation. Particularly, if we assume means of the classes are zero [1], (5.17) can be rewritten as

$$\sum_{q=1}^{\mathrm{Q}} \left(\frac{d_q}{\sigma_{\mathcal{A}_q}} \right)^2 < \sum_{q=1}^{\mathrm{Q}} \left(\frac{d_q}{\sigma_{\mathcal{B}_q}} \right)^2 + \text{constant} \tag{6.10}$$

and then as

$$\sum_{q=1}^{\mathrm{Q}} \left[d_q \left(\frac{1}{\sigma_{\mathcal{A}_q}^2} - \frac{1}{\sigma_{\mathcal{B}_q}^2} \right)^{1/2} \right]^2 < \text{constant}. \tag{6.11}$$

Then the weights w_q in equations (6.8) and (6.9) can be chosen as

$$w_q = \left(\frac{1}{\sigma_{\mathcal{A}_q}^2} - \frac{1}{\sigma_{\mathcal{B}_q}^2} \right)^{1/2}. \tag{6.12}$$

We should be careful with the term inside the paranthesis in equation (6.12) because it can be negative if $\sigma_{\mathcal{A}_q}$ is greater than $\sigma_{\mathcal{B}_q}$. But note that with classes having the same means, a feature with a within-class variance greater than its between-class variance will be of no use so it will not be used at all.

Another way of choosing the weights is to formulate the weight selection process as a regression problem. Given N groundtruth sub-image pairs with their feature vectors $(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})$ and their labels $c^{(1)}, \ldots, c^{(N)}$ where

$$c^{(i)} = \begin{cases} +1 & \text{if pair } i \text{ is from the relevance class} \\ -1 & \text{if pair } i \text{ is from the irrelevance class,} \end{cases} \tag{6.13}$$

first, the differences $d^{(1)}, \ldots, d^{(N)}$ are computed as

$$d^{(i)} = x^{(i)} - y^{(i)}, \quad x^{(i)}, y^{(i)}, d^{(i)} \in R^{\mathrm{Q}},\ i = 1, \ldots, N. \tag{6.14}$$

We can then compute the Q-dimensional weight vector W as

$$\underbrace{\left(d^{(1)} \ \cdots \ d^{(N)}\right)'}_{D^{(N\times \mathrm{Q})}} \quad W \quad = \underbrace{\left(c^{(1)} \ \cdots \ c^{(N)}\right)'}_{C^{(N\times 1)}}, \tag{6.15}$$

$$D^{(N\times \mathrm{Q})} \qquad W^{(\mathrm{Q}\times 1)} = \qquad C^{(N\times 1)}. \tag{6.16}$$

[1] The zero mean assumption is very reasonable because the differences are in the range $[-1, 1]$ and are almost symmetric around 0.

The solution for this regression problem is

$$W = (D'D)^{-1}D'C. \tag{6.17}$$

6.3.3 Modified Distance Measures

Note that establishing similarity using these distances require two images to be similar with respect to all components (features) in their feature vectors. If we consider similarity as two images being similar with respect to at least t features, we can define the following modified distance measures.

Let $\mathrm{U} \subseteq \{1, \ldots, \mathrm{Q}\}$ is a set of features such that $\#\mathrm{U} = t$ and

$$|x_u - y_u| \leq |x_v - y_v|, \quad \forall u \in \mathrm{U}, v \in \{1, \ldots, \mathrm{Q}\} \backslash \mathrm{U} \tag{6.18}$$

where $t \in \{1, \ldots, \mathrm{Q}\}$, $x, y \in R^{\mathrm{Q}}$ and x_u and y_u are the u'th components of x and y respectively. Then, the modified distance measures are defined as

$$\rho''(x, y) = \sum_{u \in \mathrm{U}} |x_u - y_u| \tag{6.19}$$

for the L^1 norm,

$$\rho''(x, y) = \sqrt{\sum_{u \in \mathrm{U}} (x_u - y_u)^2} \tag{6.20}$$

for the Euclidean distance and

$$\rho''(x, y) = \max_{u \in \mathrm{U}} |x_u - y_u| \tag{6.21}$$

for the infinity norm. These distances can be modified to be used with the weighted distances in a similar way.

6.4 Summary

The decision making process is summarized in the object/process diagram in Figure 6.1. Given the feature vector of the query image, feature vectors of the sub-images

in the database, and a distance measure, sub-images in the database are ranked in ascending order of their distances to the query image. Then, the database images which contain the sub-images that best match to the query image are retrieved as the results of the query. Results of the experiments using the decision methods described in this chapter are presented in the next chapter.

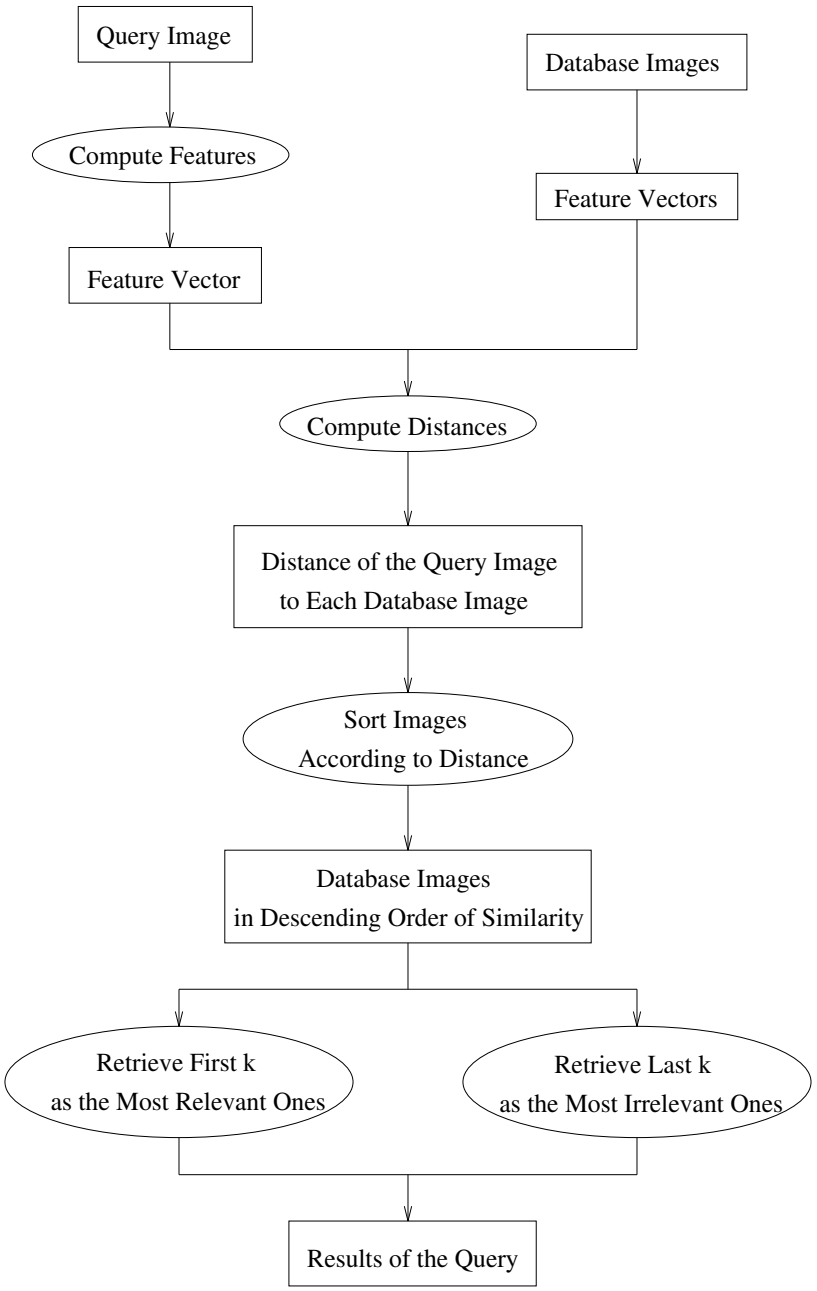


Figure 6.1: Object/process diagram for the decision making process.

Chapter 7

EXPERIMENTS AND RESULTS

Testing content-based retrieval systems and comparing their performances is an open question. In most of the content-based retrieval literature, researchers presented example queries to visually evaluate the performance of their systems. In order to compare two content-based retrieval systems, we need experiments based on groundtruth data. There are measures like precision, recall, misdetection rate and false alarm rate that were already proposed in information retrieval and pattern recognition literature to evaluate the results of such experiments. After computing these measures for each content-based retrieval system, we will be able to compare them based on how well they perform on this groundtruth data.

In this work, two datasets, an Aerial Image Dataset and the COREL Photo Stock Library, are used as databases. The features used include the line-angle-ratio statistics as described in Chapter 2, the co-occurrence variances as described in Chapter 3 and their combinations as described in Chapter 4 (The features that were developed in Section 4.2 are not used here.). The rest of the chapter is organized as follows. First, the database population including sub-image-level and image-level groundtruth construction is described. Then, results of the feature selection tests are presented. After finalizing the parameters, experimental procedures for "classification effectiveness" and "retrieval performance" are given. Next, experimental results accompanied with example queries are discussed. The chapter concludes with the analysis of error pictures.

7.1 Database Population

7.1.1 Aerial Image Database

To populate the first database we used the Fort Hood Data [2], supplied for the RADIUS Project by the Digital Mapping Laboratory at the Carnegie Mellon University. These aerial images consist of visible light images of the Fort Hood area at Texas. We used the images `fhn711`, `fhn713`, `fhn715`, `fhn717` and `fhn719`, and divided them into a total of 1,000 512×512 images.

The second source for this database is the Remote Sensing image collection from the LANDSAT and Defense Meteorological Satellite Program (DMSP) Satellites. These include images of USA (800×720), Chernobyl (512×512), and North Pole (608×896), making a total of 90 images. Samples from the Aerial Image Database are shown in Figure 7.3.

After the database was constructed, we carried out the approach described in Section 5.2 which involved translating a $\mathrm{K} \times \mathrm{K}$ frame throughout every image and extracted the desired features for all sub-images. As described in that section, first sub-image database (*main database*) contains a unique sub-image i.d., bounding box, and the feature vector for each sub-image $m = 1, \ldots, \mathrm{M}_i$ and $i = 1, \ldots, \mathrm{I}$ where M_i is the number of non-shifted sub-images in image i and I is the total number of images in the database. The second sub-image database (*test database*) contains a unique sub-image i.d., bounding box, overlapping sub-images set $R_s(n)$, randomly selected non-overlapping sub-images set $R_o(n)$, and the feature vector for each sub-image $n = 1, \ldots, \mathrm{N}_i$ and $i = 1, \ldots, \mathrm{I}$ where N_i is the number of shifted sub-images in image i and I is again the total number of images in the database. Schemas for the *main database* and the *test database* are given in Figures 7.1 and 7.2 respectively. Sub-image size K is selected to be 256. M_i and N_i vary according to the corresponding image sizes. Feature vector size Q will be determined in Section 7.2.

Final sub-image databases consist of 10,410 256×256 sub-images for the *main*

key		bounding box				feature vector			
imagename	i.d.	ulr	ulc	nr	nc	f_1	$\cdots$	$\cdots$	f_{Q}
image_1	1	0	0	K	K	•	$\cdots$	$\cdots$	•
$\vdots$			$\vdots$				$\vdots$		
image_1	M_1	•	•	K	K	•	$\cdots$	$\cdots$	•
$\vdots$			$\vdots$				$\vdots$		
$\vdots$			$\vdots$				$\vdots$		
$\mathrm{image}_{\mathrm{I}}$	1	0	0	K	K	•	$\cdots$	$\cdots$	•
$\vdots$			$\vdots$				$\vdots$		
$\mathrm{image}_{\mathrm{I}}$	M_{I}	•	•	K	K	•	$\cdots$	$\cdots$	•

Figure 7.1: Schema for the *main database*. (ulr,ulc) is the upper-left row and column locations, nr and nc are the height (number of rows) and width (number of columns) of the sub-image bounding box respectively. f_i is the i'th component of the Q-dimensional feature vector.

key		bounding box				R_s(key)	R_o(key)	feature vector
imagename	i.d.	ulr	ulc	nr	nc	$s_1 \cdots s_4$	$o_1 \cdots o_4$	$f_1 \cdots \cdots f_Q$
$image_1$	1	K/4	K/4	K	K	$s_1 \cdots s_4$	$o_1 \cdots o_4$	• ⋯ ⋯ •
⋮			⋮			⋮	⋮	⋮
$image_1$	N_1	•	•	K	K	$s_1 \cdots s_4$	$o_1 \cdots o_4$	• ⋯ ⋯ •
⋮			⋮			⋮	⋮	⋮
⋮			⋮			⋮	⋮	⋮
$image_I$	1	K/4	K/4	K	K	$s_1 \cdots s_4$	$o_1 \cdots o_4$	• ⋯ ⋯ •
⋮			⋮			⋮	⋮	⋮
$image_I$	N_I	•	•	K	K	$s_1 \cdots s_4$	$o_1 \cdots o_4$	• ⋯ ⋯ •

Figure 7.2: Schema for the *test database*. In addition to the attributes in the *main database*, this database also contains the overlapping and non-overlapping sub-images sets for each sub-image entry.

database and 4,780 256×256 sub-images for the *test database*. There are 4 relevant and 4 irrelevant *main database* sub-images for each of the *test database* sub-images, which make a total of 38,240 groundtruth sub-image pairs. We also manually grouped the images into 10 groups; parking lots, large buildings, small buildings, residential areas 1, residential areas 2, roads, landscapes, LANDSAT USA, DMSP North Pole and LANDSAT Chernobyl. The rows in Figure 7.3 show sample images from each group.

7.1.2 COREL Database

To populate the second database, we used the categories Sunsets and Sunrises, Air Shows, Bears, Elephants, Tigers, Patterns, Arabian Horses, Flowers, Flowers2, American Gardens, Cheetahs, Bald Eagles, Textures, Autumn, Land of the Pyramids, Polar Bears, Ice and Icebergs, Horses, Mountains of America, Fruits and Vegetables,

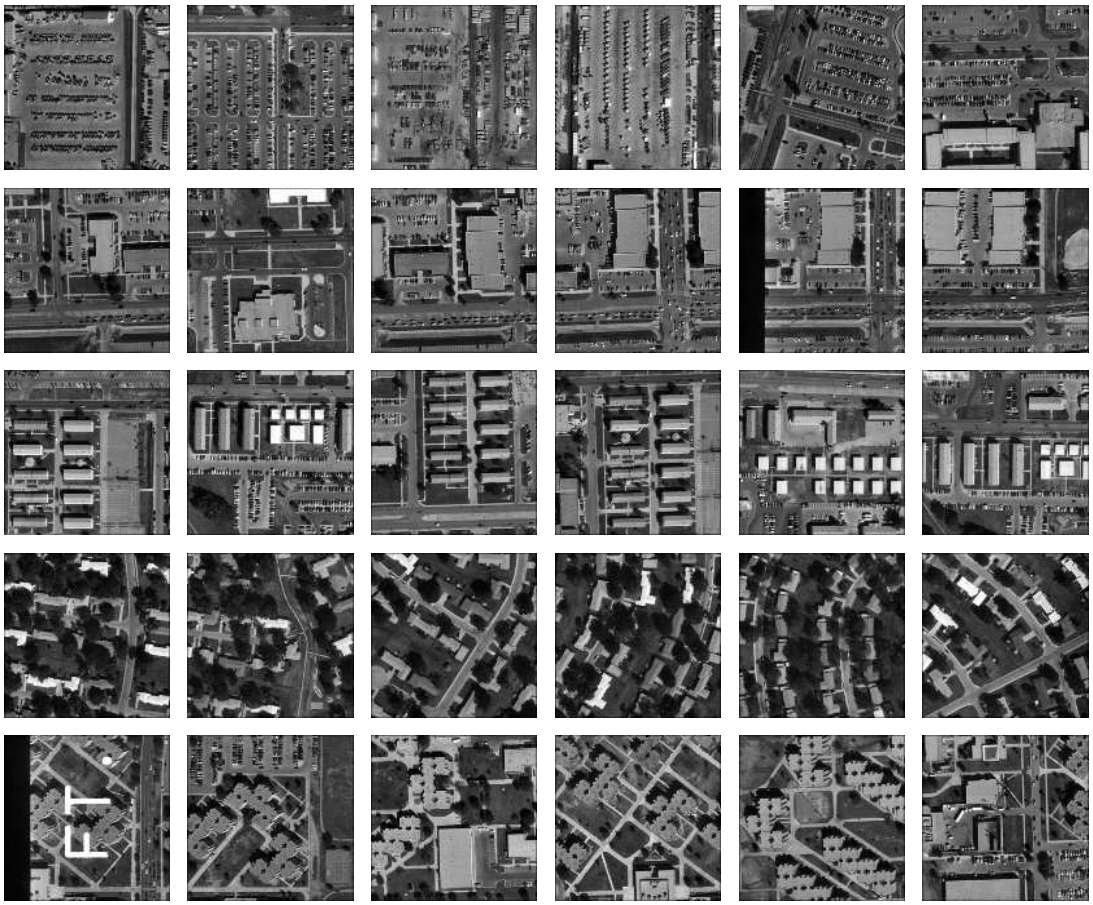


Figure 7.3: Sample images from the Aerial Image Database. The rows show sample images from the groundtruth groups parking lots, large buildings, small buildings, residential areas 1 and residential areas 2 in top-down order.

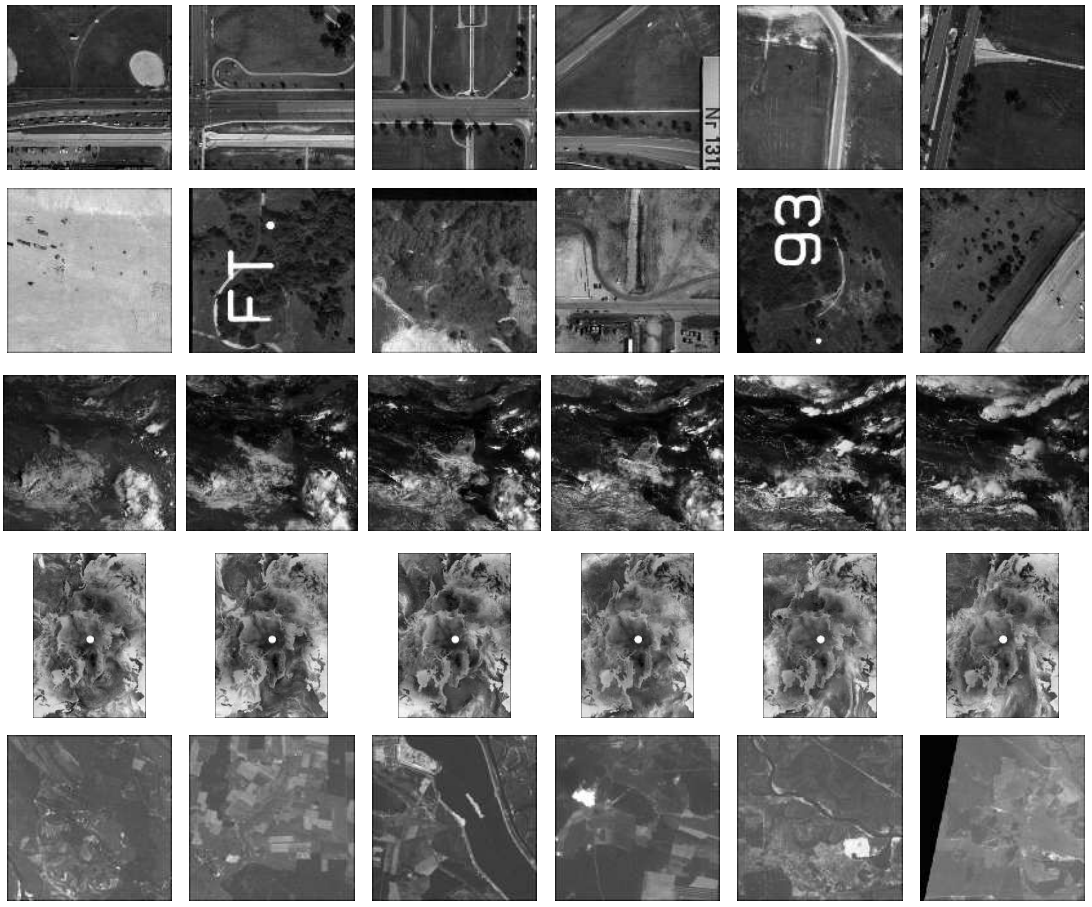


Figure 7.3: Sample images from the Aerial Image Database (cont.). The rows show sample images from the groundtruth groups roads, landscapes, LANDSAT USA, North Pole and Chernobyl in top-down order.

Tulips, Fields, Deserts, Death Valley, Fireworks, Coasts, Doors of San Francisco, Owls, Roses, Food, and Candy Backgrounds in the COREL Photo Stock Library 1 [1]. Each category consists of 100 images with sizes 256×384 or 384×256. Since we do not use color information, only grayscale versions of the images are included.

Since image sizes are already small compared to the sub-image sizes, whole images are used as entries to the database. Because of this, only the *main database* is constructed for this dataset. Image-level groundtruth for the COREL Database is already available in the COREL Library, as each CD is a category containing 100 images. Carson *et al.* [13] discarded some of the images from each category because they are visually inconsistent with the rest of the category. We use all of the images, which constitute a database of 3,100 images in 31 categories in our experiments but we argue their visual consistencies in Section 7.6. The rows in Figure 7.4 show sample images from some of the categories.

As a result, the Aerial Image Database contains 10,410 256×256 sub-images and the COREL Database contains 3,100 256×384 or 384×256 images. Experimental procedures and their results are presented in the following sections.

7.2 Feature Selection

In the feature selection experiments only the Aerial Image Database is used. The following sections discuss the parameter selection for line-angle-ratio statistics and co-occurrence variances. Experiments for each parameter combination used consist of classifying approximately 38,000 sub-image pairs into the relevance or irrelevance classes.

7.2.1 Line-Angle-Ratio Statistics

The goal of the feature selection tests for line-angle-ratio statistics is to select the quantizer that performs the best, according to the criteria developed in Section 5.3,

Figure 7.4: Sample images from the COREL Database. The rows show sample images from the groundtruth groups candies, doors, sunsets, air shows, fireworks, roses and bears in top-down order.

Figure 7.4: Sample images from the COREL Database (cont.). The rows show sample images from the groundtruth groups bald eagles, owls, fruits and veggies, pyramids, coasts, fields and mountains in top-down order.

among a set of quantizers with 15, 20 and 25 quantization cells.

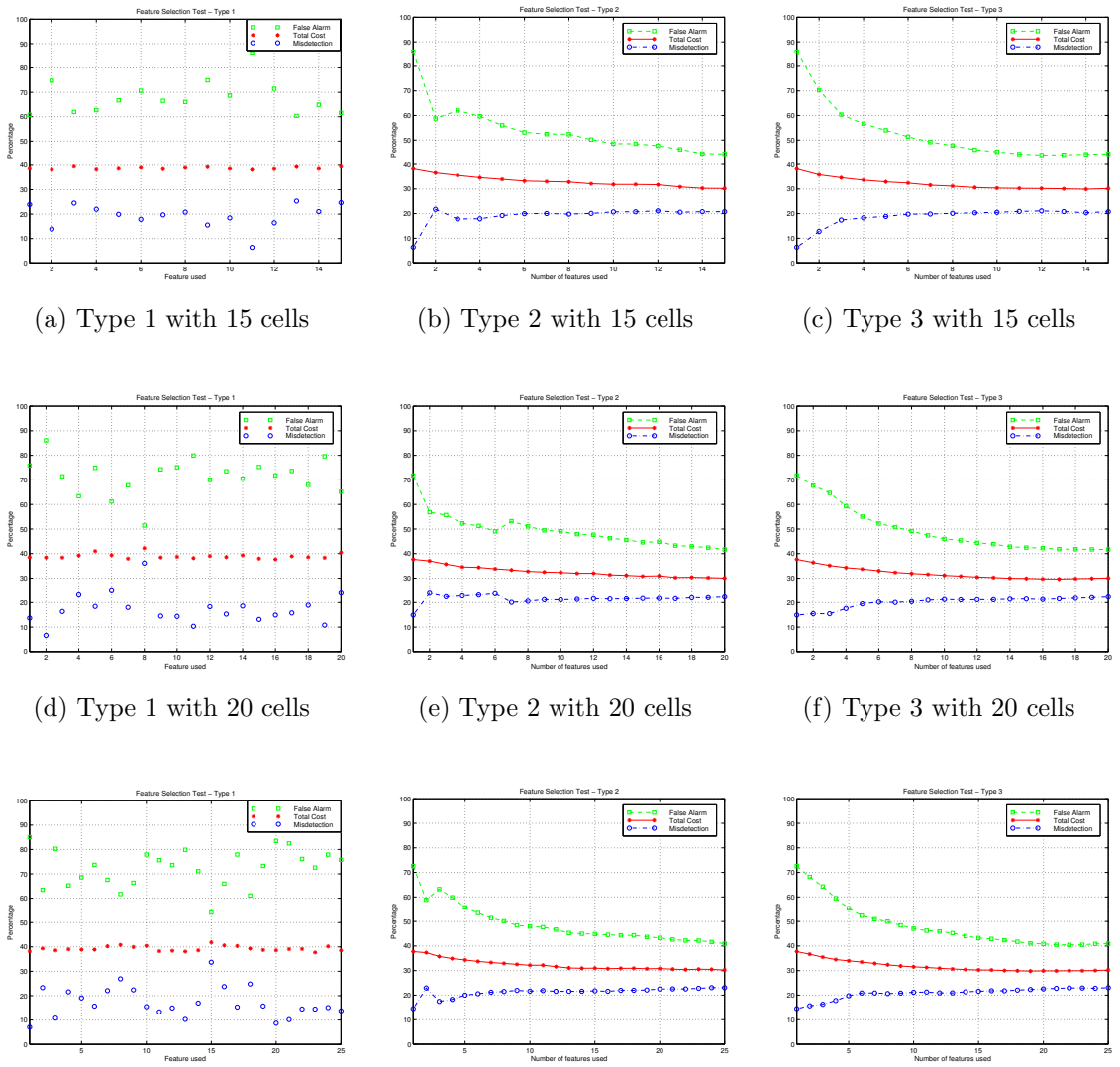
Distribution of the training line-angle-ratio samples used for the quantizer were given in Figure 2.4(a). Centroids of the resulting partitions using 15, 20 and 25 quantization cells were given in Figures 2.4(b), 2.4(c) and 2.4(d) respectively. During training, 6,844 intersecting/near-intersecting line pairs that were extracted from 1,000 randomly selected 256×256 sub-images were used to train the quantizers. Another set of 8,383 intersecting/near-intersecting line pairs were extracted to test the quantizers and were quantized into 15 cells with a mean square error of 0.005668, into 20 cells with a mean square error of 0.004031 and into 25 cells with a mean square error of 0.003377.

Results of the feature selection tests are given in Figure 7.5. Three types of experiments described in Section 5.3 were performed on the Aerial Image Database. In all of the cases the false alarm rate was higher than the misdetection rate. Since the main concern is to select among 15, 20 and 25 quantization cells, only the total costs using all of the features were considered. The quantizer with 15 quantization cells resulted in a total cost of 30.20% whereas quantizers with 20 and 25 cells had 30.05% and 30.22% total costs respectively. As a result, we decided to use the quantizer with 20 cells, which results in a 20-dimensional feature vector in the rest of the experiments that use line-angle-ratio statistics.

7.2.2 Co-occurrence Variances

The goal of the feature selection tests for co-occurrence variances is to select the set of distances, among distances of 1 to 20 pixels, that perform the best according to the classification criteria. Each distance considered here is a combination of four features which correspond to variances computed at 0, 45, 90 and 135 degree orientations for that distance.

Results of the feature selection tests are given in Figure 7.6. In all of the cases the false alarm rate was higher than the misdetection rate. Type 3 tests which involve



(a) Type 1 with 15 cells (b) Type 2 with 15 cells (c) Type 3 with 15 cells

(d) Type 1 with 20 cells (e) Type 2 with 20 cells (f) Type 3 with 20 cells

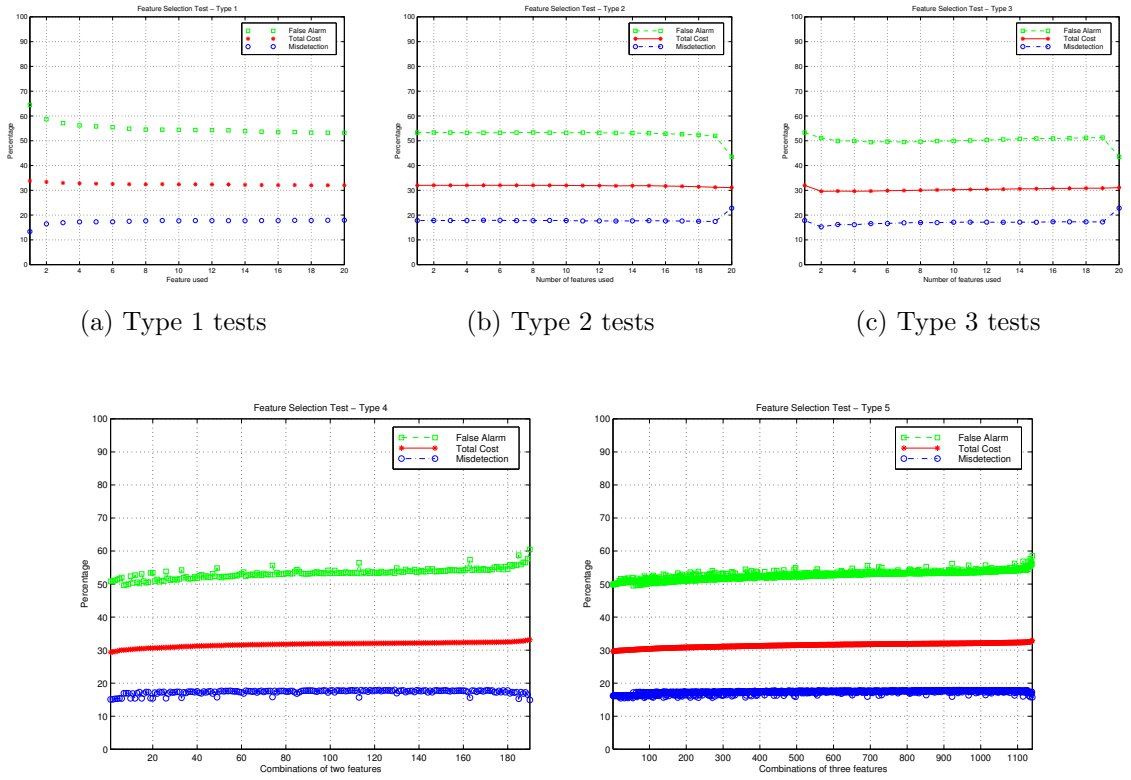
(g) Type 1 with 25 cells (h) Type 2 with 25 cells (i) Type 3 with 25 cells

Figure 7.5: Feature selection tests for Line-Angle-Ratio Statistics using the Aerial Image Database. Type 1 tests are done using only one feature at a time, type 2 tests are done using all features first and discarding the worst features one by one, type 3 tests are done using the best feature first and adding the next best features one by one. The criterion for "goodness" is the total cost which is plotted in red.

using the best feature first and adding the next best features one by one converged to an overall total cost faster than the type 2 tests which involve using all features first and discarding the worst features one by one. In other words, building up feature sets decreased the total cost faster than shrinking down the set of all features. Another observation was that after using approximately 2 or 3 distances, total cost did not decrease much, so we decided to try all possible combinations of 2 and 3 distances. In type 4 and type 5 tests, where all possible combinations of 2 and 3 distances are used respectively, the minimum total cost of 29.36% was obtained using the distances 1 and 20 together.

When we consider Remote Sensing Dataset and Ft. Hood Dataset separately, smaller distances resulted in smaller total costs for Remote Sensing images, while smaller total costs were obtained using large distances for Ft. Hood images. This is consistent with the results of Weszka *et al.* [59] who stated that co-occurrence matrices computed for small distances performed better for a LANDSAT dataset. We believe that the reason for this is the strong micro-scale texture information in Remote Sensing images in contrast to larger structures in the Ft. Hood images.

As a result, we decided to use the distances 1 and 20, which result in an 8-dimensional feature vector in the rest of the experiments that use co-occurrence variances. Although these feature selection tests do not guarantee an optimal solution, they resulted in a suboptimal one in 1,560 classification tests without using exhaustive search which would then require $2^{20} - 1$ classification tests. After selecting the parameters, performance of the system is evaluated using two types of experiments, namely the classification effectiveness and the retrieval performance, as described in the following sections.



(a) Type 1 tests (b) Type 2 tests (c) Type 3 tests

(d) Type 4 tests (e) Type 5 tests

Figure 7.6: Feature selection tests for Co-occurrence Variances using the Aerial Image Database. Type 1 tests are done using only one feature at a time, type 2 tests are done using all features first and discarding the worst features one by one, type 3 tests are done using the best feature first and adding the next best features one by one, type 4 tests are done using all possible combinations of 2 features, type 5 tests are done using all possible combinations of 3 features. The criterion for "goodness" is the total cost which is plotted in red.

7.3 Classification Effectiveness

7.3.1 Experimental Set-up

To test the classification effectiveness of the features, we use a Gaussian classifier and a nearest neighbor classifier. The classification algorithm for the Gaussian classifier was defined in (5.15) and was described in Section 5.3. Here we do not assume that the features are independent. Since we know which non-shifted sub-images and shifted sub-images overlap, we also know which sub-image pairs should be assigned to the relevance class $\mathcal{A}$ and which to the irrelevance class $\mathcal{B}$.

To train the nearest neighbor classifier to classify sub-image pairs, we use the following procedure. Given groundtruth sub-image pairs (n,m) with feature vectors x and y respectively, first, their distances $\rho(x, y)$ are computed using the L^1 norm, the Euclidean distance and the infinity norm that were described in Section 6.3. Then, from the histograms of ρ for pairs from the relevance class $\mathcal{A}$ groundtruth and the irrelevance class $\mathcal{B}$ groundtruth, a threshold θ is selected to distinguish between two classes. The best threshold is defined as the one that minimizes the total cost which was defined as 3 misdetections and 2 false alarms in Section 5.3. In the testing phase, given a sub-image pair (n, m) with feature vectors x and y, the classifier decides as

$$\text{assign } (n,m) \text{ to } \begin{cases} \text{class } \mathcal{A} & \text{if } \rho(x,y) < \theta \\ \text{class } \mathcal{B} & \text{if } \rho(x,y) \geq \theta. \end{cases} \tag{7.1}$$

As a result, to test our approach, we check whether each sub-image pair that should be classified into class $\mathcal{A}$ or $\mathcal{B}$ is classified into class $\mathcal{A}$ or $\mathcal{B}$ correctly. This accounts to checking pairs as many as eight times the total number of shifted sub-images, which makes approximately 38,000 pairs in each of the experiments for each feature and for each classifier. We evaluate the results by comparing misdetection and false alarm rates and the total costs that are computed from the confusion matrices. In these experiments only the Aerial Image Database is used. Results are given in

the next section.

7.3.2 Results

Likelihood Ratio – Gaussian Classifier:

As can be seen in Tables 7.1 – 7.3, approximately 80% of the relevance class $\mathcal{A}$ groundtruth pairs were assigned to class $\mathcal{A}$ correctly. Line-angle-ratio features had a total cost of 30.44%, co-occurrence features had a total cost of 27.20% and combined features had a total cost of 23.50%. Co-occurrence features performed better than line-angle-ratio features, while combined features were much better than both.

Nearest Neighbor Classifier:

Distributions of distances for both the relevance and the irrelevance class groundtruth sub-image pairs are given in Figures 7.7 – 7.9. Results of the classification tests using the thresholds that minimize the corresponding total costs are given in Tables 7.4 – 7.12. Approximately 85% of the relevance class $\mathcal{A}$ groundtruth pairs were assigned to class $\mathcal{A}$ correctly. Among the features used, line-angle-ratio features and co-occurrence features performed almost equally, while combined features were better than both. In the case of distance measures, the L^1 norm performed slightly better than the Euclidean distance and the infinity norm was the worst of all. Also the infinity norm seems to be dominated by the worst performing features. We believe that this problem can be solved using the modified distance measures proposed in Section 6.3.3. Further research is required on this issue. Note that results of the nearest neighbor classifier are worse than those of the Gaussian classifier because the latter also considers the variances of the two classes and is effected less from the overlap between the class distributions.

We can say that most of the relevance class $\mathcal{A}$ groundtruth pairs were assigned to class $\mathcal{A}$ correctly but the irrelevance class $\mathcal{B}$ groundtruth pairs seem to be split

Table 7.1: Classification effectiveness test for line-angle-ratio statistics using the Gaussian classifier (total cost is 30.44%).

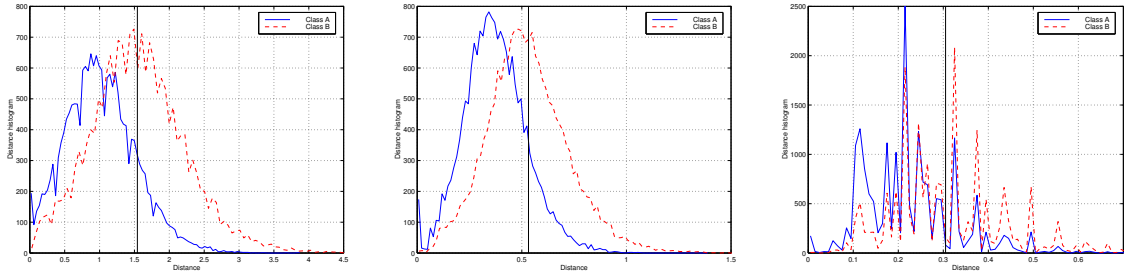
	Assigned $\mathcal{A}$	Assigned $\mathcal{B}$	Success (%)
G.truth $\mathcal{A}$	14,513	4,254	77.33
G.truth $\mathcal{B}$	7,873	10,828	57.90
Overall	22,386	15,082	67.63

Table 7.2: Classification effectiveness test for co-occurrence variances using the Gaussian classifier (total cost is 27.20%).

	Assigned $\mathcal{A}$	Assigned $\mathcal{B}$	Success (%)
G.truth $\mathcal{A}$	16,229	2,868	84.98
G.truth $\mathcal{B}$	8,678	10,403	54.52
Overall	24,907	13,271	69.76

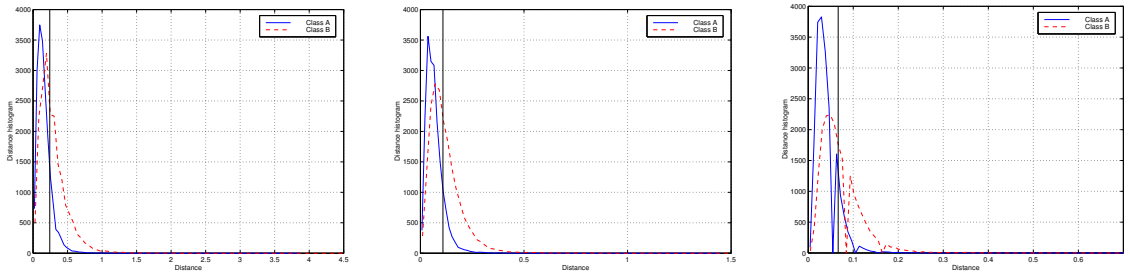
Table 7.3: Classification effectiveness test for combined features using the Gaussian classifier (total cost is 23.50%).

	Assigned $\mathcal{A}$	Assigned $\mathcal{B}$	Success (%)
G.truth $\mathcal{A}$	15,904	3,197	83.26
G.truth $\mathcal{B}$	6,422	12,665	66.35
Overall	22,326	15,862	74.81



(a) L^1 norm (b) Euclidean distance (c) Infinity norm

Figure 7.7: Distance histograms for line-angle-ratio statistics. Distances for the groundtruth pairs in the relevance class and the irrelevance class are shown in blue and red respectively. The decision thresholds are selected to be 1.5419 for the L^1 norm, 0.5320 for the Euclidean distance and 0.305 for the infinity norm.



(a) L^1 norm (b) Euclidean distance (c) Infinity norm

Figure 7.8: Distance histograms for co-occurrence variances. Distances for the groundtruth pairs in the relevance class and the irrelevance class are shown in blue and red respectively. The decision thresholds are selected to be 0.2417 for the L^1 norm, 0.1085 for the Euclidean distance and 0.0665 for the infinity norm.

Table 7.4: Classification effectiveness test for line-angle-ratio statistics using the L^1 norm (total cost is 29.33%).

	Assign $\mathcal{A}$	Assign $\mathcal{B}$	Success(%)
G.truth $\mathcal{A}$	16,333	2,787	85.42
G.truth $\mathcal{B}$	9,819	9,258	48.53
Overall	26,152	12,045	66.98

Table 7.5: Classification effectiveness test for line-angle-ratio statistics using the Euclidean distance (total cost is 29.65%).

	Assign $\mathcal{A}$	Assign $\mathcal{B}$	Success(%)
G.truth $\mathcal{A}$	16,492	2,628	86.26
G.truth $\mathcal{B}$	10,209	8,868	46.49
Overall	26,701	11,496	66.37

Table 7.6: Classification effectiveness test for line-angle-ratio statistics using the infinity norm (total cost is 33.13%).

	Assign $\mathcal{A}$	Assign $\mathcal{B}$	Success(%)
G.truth $\mathcal{A}$	15,364	3,756	80.36
G.truth $\mathcal{B}$	10,178	8,899	46.65
Overall	25,542	12,655	63.50

Table 7.7: Classification effectiveness test for co-occurrence variances using the L^1 norm (total cost is 30.33%).

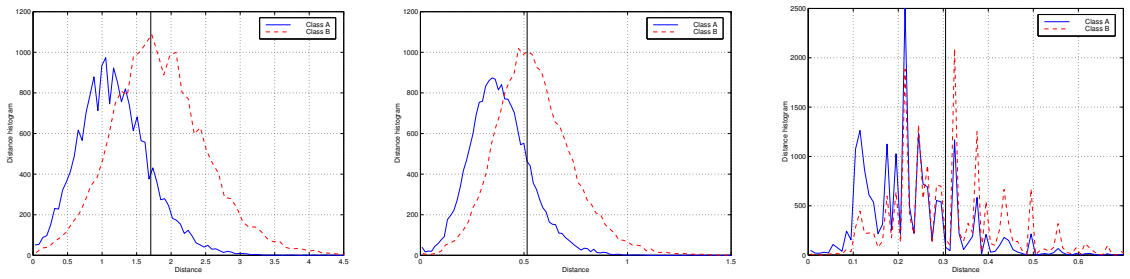
	Assign $\mathcal{A}$	Assign $\mathcal{B}$	Success(%)
G.truth $\mathcal{A}$	15,919	3,201	83.26
G.truth $\mathcal{B}$	9,694	9,423	49.29
Overall	25,613	12,624	66.27

Table 7.8: Classification effectiveness test for co-occurrence variances using the Euclidean distance (total cost is 29.58%).

	Assign $\mathcal{A}$	Assign $\mathcal{B}$	Success(%)
G.truth $\mathcal{A}$	16,584	2,536	86.74
G.truth $\mathcal{B}$	10,333	8,784	45.95
Overall	26,917	11,320	66.34

Table 7.9: Classification effectiveness test for co-occurrence variances using the infinity norm (total cost is 29.28%).

	Assign $\mathcal{A}$	Assign $\mathcal{B}$	Success(%)
G.truth $\mathcal{A}$	16,640	2,480	87.03
G.truth $\mathcal{B}$	10,275	8,842	46.25
Overall	26,915	11,322	66.64



(a) L^1 norm (b) Euclidean distance (c) Infinity norm

Figure 7.9: Distance histograms for combined features. Distances for the groundtruth pairs in the relevance class and the irrelevance class are shown in blue and red respectively. The decision thresholds are selected to be 1.7053 for the L^1 norm, 0.5151 for the Euclidean distance and 0.305 for the infinity norm.

Table 7.10: Classification effectiveness test for combined features using the L^1 norm (total cost is 26.69%).

	Assign $\mathcal{A}$	Assign $\mathcal{B}$	Success(%)
G.truth $\mathcal{A}$	16,295	2,825	85.22
G.truth $\mathcal{B}$	8,519	10,601	55.44
Overall	24,814	13,426	70.33

Table 7.11: Classification effectiveness test for combined features using the Euclidean distance (total cost is 28.55%).

	Assign $\mathcal{A}$	Assign $\mathcal{B}$	Success(%)
G.truth $\mathcal{A}$	15,825	3,295	82.77
G.truth $\mathcal{B}$	8,704	10,416	54.48
Overall	24,529	13,711	68.62

Table 7.12: Classification effectiveness test for combined features using the infinity norm (total cost is 33.05%).

	Assign $\mathcal{A}$	Assign $\mathcal{B}$	Success(%)
G.truth $\mathcal{A}$	15,355	3,765	80.31
G.truth $\mathcal{B}$	10,149	8,971	46.92
Overall	25,504	12,736	63.61

between being assigned to $\mathcal{A}$ or $\mathcal{B}$. The cause of this problem can be explained as follows. Although the assumption that overlapping sub-images are relevant almost always holds, we cannot always guarantee that non-overlapping sub-images are irrelevant. Obvious examples are images which are dominated by a single texture, or have an almost constant gray value, or have the same object pattern at more than one location. This is not uncommon among the complex images in our database. Illustration of this fact can be found in [6], where we manually eliminated some images with large regions of constant gray values from the Fort Hood Dataset and obtained a 42% decrease in the false alarm rate with a misdetection rate remaining approximately the same. Hence, some of the assignments which we count as incorrect are not in fact incorrect. This strengthens our claim that the low classification results are mainly due to the mislabeling problems of the automatic groundtruth construction protocol. Further research is required to empirically determine the mislabeling probabilities and to estimate the true classification results as described in Section 5.4.

7.4 Retrieval Performance

7.4.1 Experimental Set-up

To evaluate the retrieval performance of the features and the decision methods, we use two types of tests; pair retrieval tests and precision–recall tests. In the pair retrieval tests the database is queried with each of the sub-images in the groundtruth sub-image pairs. Given a query sub-image, first, images in the *main database* are retrieved in ascending order of their distances, that were described in Chapter 6, to the query. Then, if the image that the query sub-image belongs to is retrieved as one of the k best matches, it is considered a success. Average rank for these correct images is also computed for each value of k. When the query is a non-shifted sub-image in the *main database*, this is considered as the best case analysis. On the other hand, if a shifted sub-image in the *test database* is used as the query, this is

considered as the worst case because the shifted sub-images overlap a sub-image in the *main database* by only half the area while all other possible sub-images have a corresponding sub-image which they overlap by more than this amount.

Two traditional measures for retrieval performance in the information retrieval literature are precision and recall. Precision is defined as the percentage of retrieved images that are actually relevant $\left(\frac{\text{retrieved and relevant}}{\text{total \# of retrieved}}\right)$ and recall is defined as the percentage of relevant images that are retrieved $\left(\frac{\text{retrieved and relevant}}{\text{total \# of relevant}}\right)$ [51]. Given a query, high precision implies that very little irrelevant images have been retrieved and high recall implies that much of what is relevant in the database have been retrieved. Lack of precision can be compared to a type 2 error (false alarm) and deficiency in recall for a given search is comparable to type 1 error (misdetection). For performance evaluation, one can plot precision and recall as a function of the number of images retrieved as well as the precision versus recall curves for different numbers of images retrieved.

To evaluate the overall retrieval performance (precision and recall), we use the image-level groundtruth constructed for both the Aerial Image Database and the COREL Database. First, the database is queried with each of the images in each of the groundtruth groups shown in Figures 7.3 and 7.4, then average precision and recall percentages are computed for each group as well as for the entire database. To rank-order the database images, decision methods described in Chapter 6 are used. Specifically for the results presented below, likelihood ratio, L^1 norm, Euclidean distance and infinity norm were used for the Aerial Image Database, L^1 norm, Euclidean distance and infinity norm were used for the COREL Database.

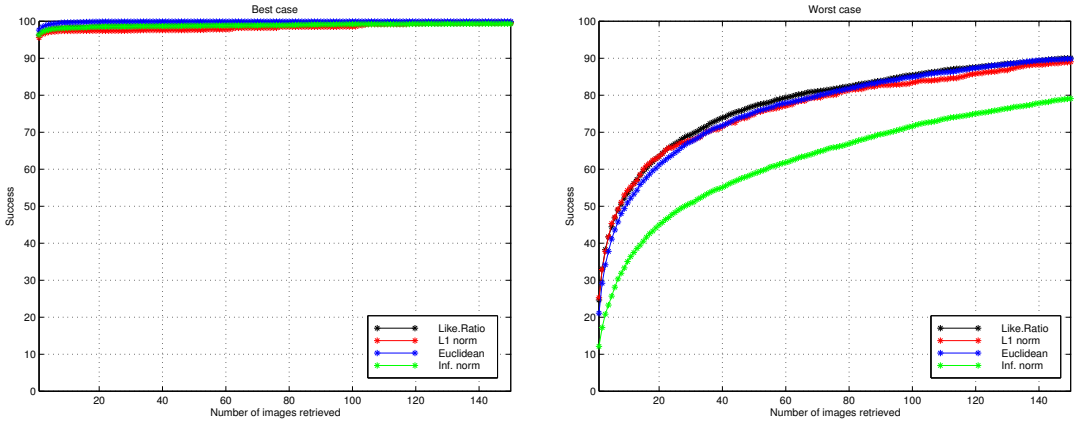
7.4.2 Results

Pair Retrieval Performance:

Results of the pair retrieval performance tests are given in Figures 7.10–7.12. In these experiments, the best case analysis consists of 10,410 queries to a database of 10,410 sub-images for each feature and for each distance measure. The worst case analysis consists of 4,780 queries for the same database. In the best case analysis, all features performed perfectly. On the other hand, in the worst case analysis, co-occurrence features had a higher success rate and a smaller average rank than those of the line-angle-ratio features, while combined features outperformed both. When the distance measures are concerned, the results were similar to the nearest neighbor classification results in Section 7.3.2. The likelihood ratio outperformed the rest by retrieving the correct image as one of the 60 best matches 100 percent of the time with an average rank of 4 among a total of 10,410 images. Among the rest, the L^1 norm performed better than the Euclidean distance, while the infinity norm was the worst. The problem that the infinity norm having a tendency of being dominated by the worst performing features is also present here.

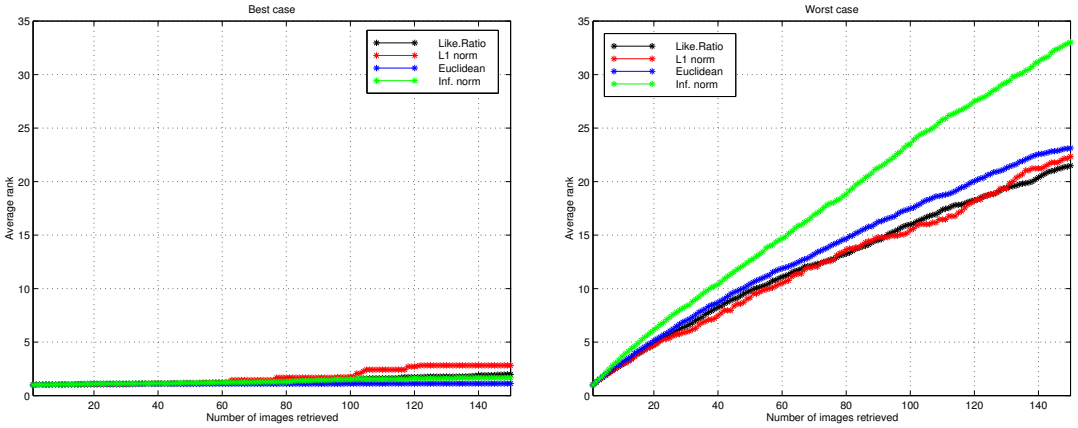
Precision and Recall: Aerial Image Database

Results of the precision–recall tests for the Aerial Image Database are given in Figures 7.13 – 7.18. These experiments consist of 10,410 queries to a database of 10,410 sub-images from 1,090 images for each feature and for each distance measure. For the Ft. Hood Dataset, where all Ft. Hood images are considered as a single group here, all features and all distance measures performed almost perfectly. The recall curves seem to be low but indeed they are very close to the ideal case because there are 1,000 images in the Ft. Hood Dataset. We do not have complete groundtruth for the Ft. Hood Dataset now so no results for individual groups shown in first 7 rows of Figure 7.3 are presented. The results given here show that the features successfully



(a) Best case success rate

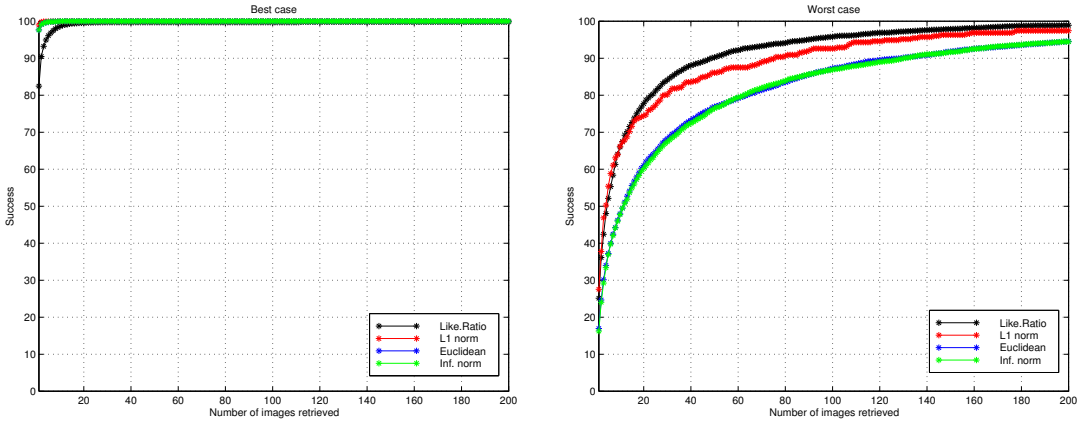
(b) Worst case success rate



(c) Best case average rank

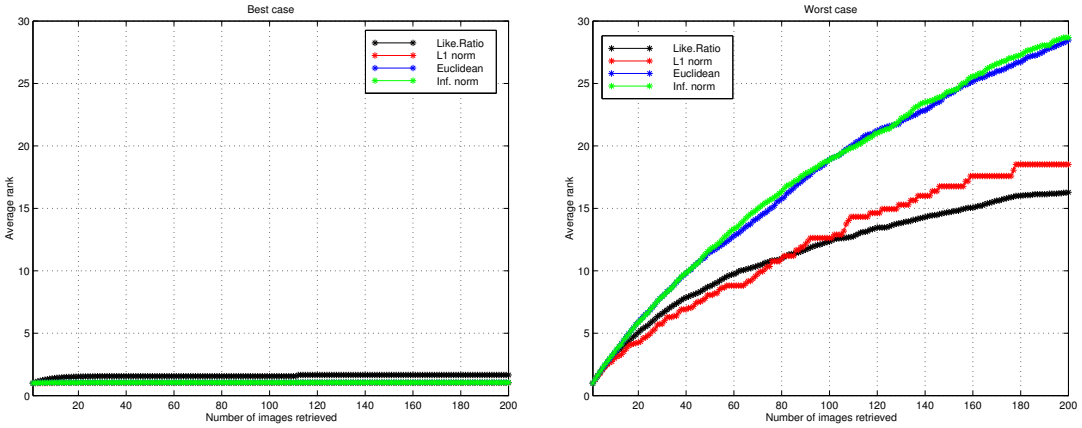
(d) Worst case average rank

Figure 7.10: Pair retrieval performance tests for line-angle-ratio statistics. Both the best case and the worst case of the success rate and the average rank are plotted as a function of the number of images retrieved using likelihood ratio, L^1 norm, Euclidean distance and infinity norm.



(a) Best case success rate

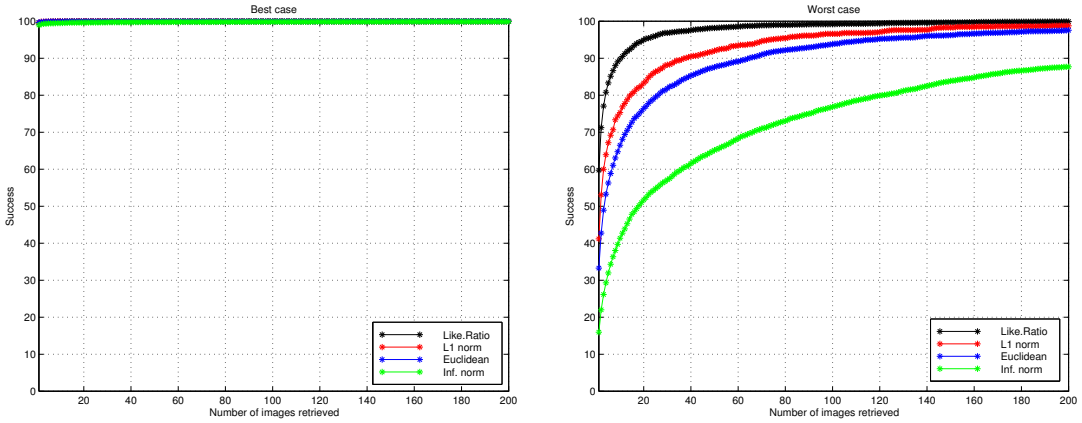
(b) Worst case success rate



(c) Best case average rank

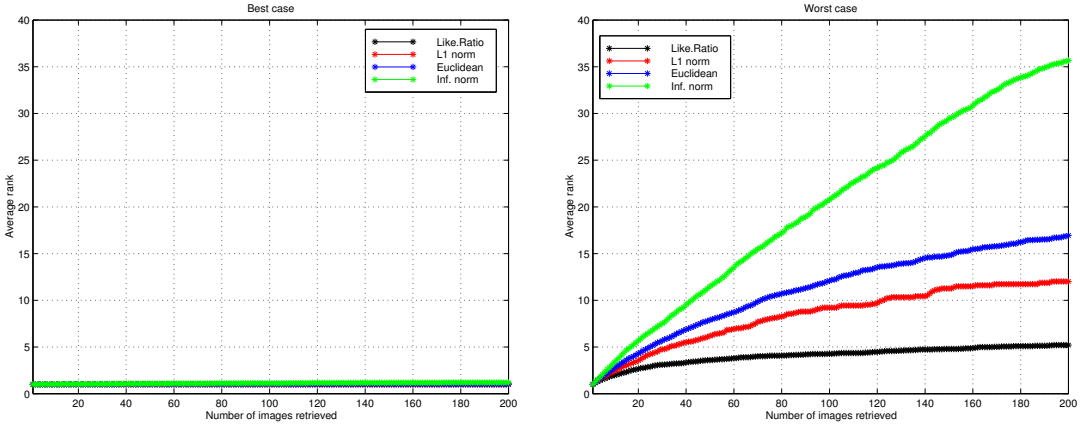
(d) Worst case average rank

Figure 7.11: Pair retrieval performance tests for co-occurrence variances. Both the best case and the worst case of the success rate and the average rank are plotted as a function of the number of images retrieved using likelihood ratio, L^1 norm, Euclidean distance and infinity norm.



(a) Best case success rate

(b) Worst case success rate



(c) Best case average rank

(d) Worst case average rank

Figure 7.12: Pair retrieval performance tests for combined features. Both the best case and the worst case of the success rate and the average rank are plotted as a function of the number of images retrieved using likelihood ratio, L^1 norm, Euclidean distance and infinity norm.

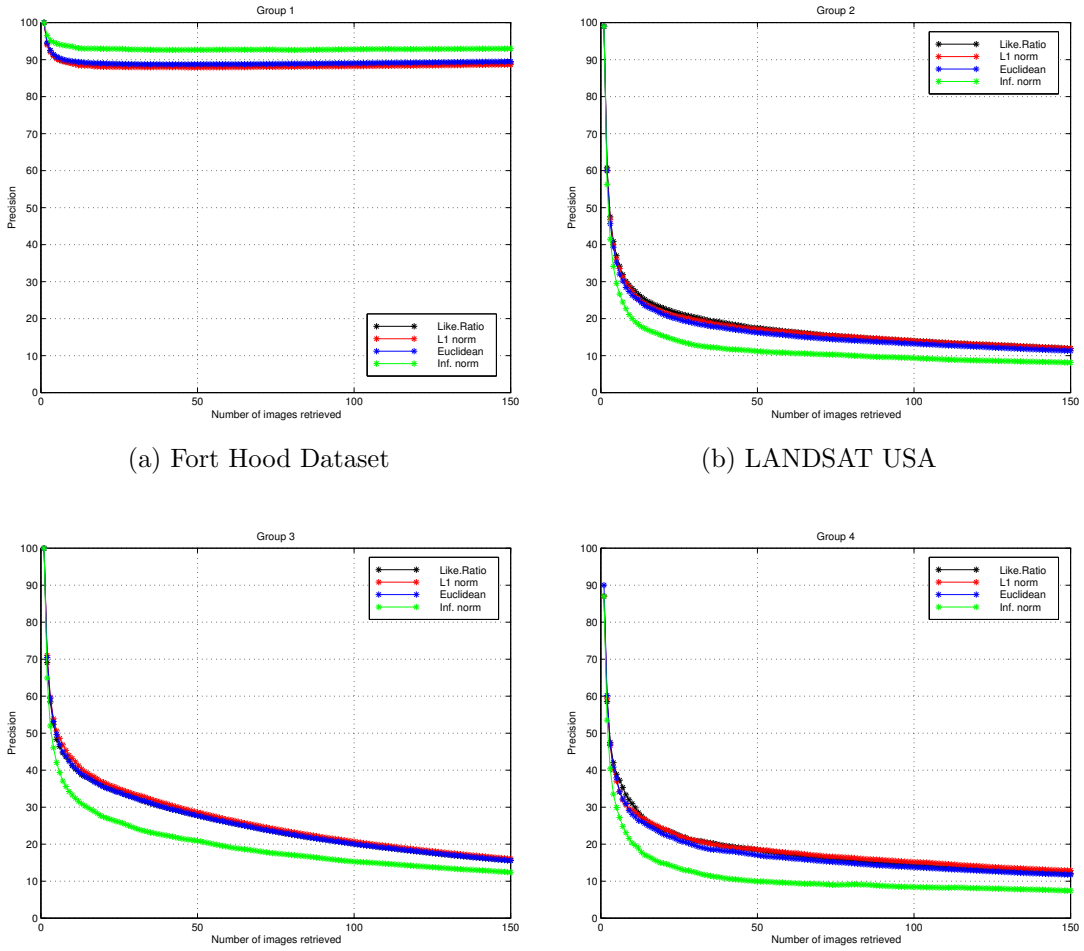
distinguished the Ft. Hood images from the Remote Sensing images.

For the Remote Sensing Dataset, which includes 1,410 of the 10,410 sub-images in the Aerial Image Database, co-occurrence features and combined features outperformed line-angle-ratio features. This is due to the significant micro-scale texture characteristics of these images. As the distance measures are concerned, the likelihood ratio was again the best performing distance. The L^1 norm performed at least as good as and sometimes better than the Euclidean distance. The infinity norm was again the worst performing one.

Precision and Recall: COREL Database

Results of the precision–recall tests for the COREL Database are given in Figures 7.19 – 7.24. These experiments consist of 3,100 queries to a database of 3,100 images for each feature and for each distance measure. Similar to the Aerial Image Database, different features performed differently for different groundtruth groups. For groups like "doors" where there is significant line information, line-angle-ratio features performed much better than co-occurrence features. On the contrary, for groups like "owls" and "fireworks" where a micro-texture is dominant, co-occurrence features outperformed line-angle-ratio features. Combined features again combined the advantages of individual features and performed significantly better than the individual cases with the exception of the infinity norm that was dominated by the worst performing feature for the groups "owls" and "roses". The rest of the distance measures performed equally well when individual features are used. The Euclidean distance and the L^1 norm performed better than the infinity norm for combined features case due to the reasons mentioned above.

As can be seen from the precision curves, most of the first 10 images retrieved belong to the same group as the query image. After approximately 10 images, precision decreases more rapidly. This is mainly due to the visual inconsistencies in the groundtruth groups. More discussion on this issue will be made in Section 7.6.



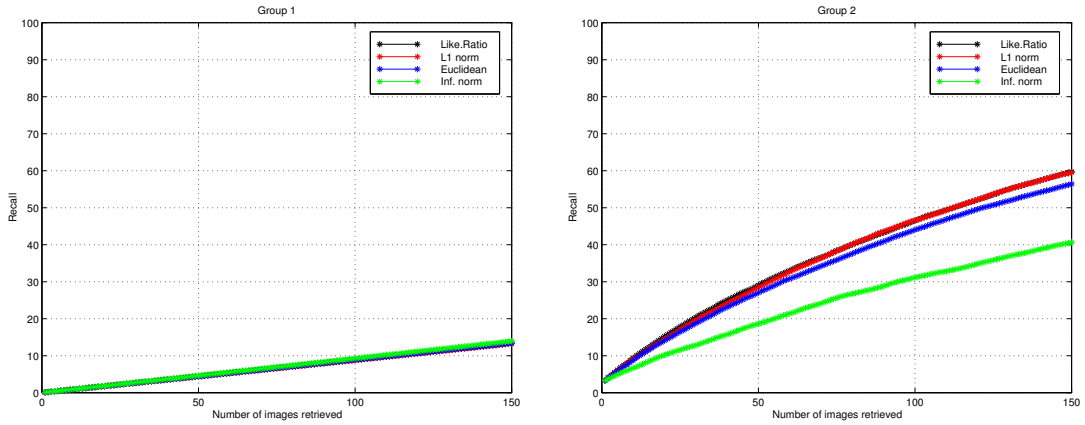
(a) Fort Hood Dataset

(b) LANDSAT USA

(c) DMSP North Pole

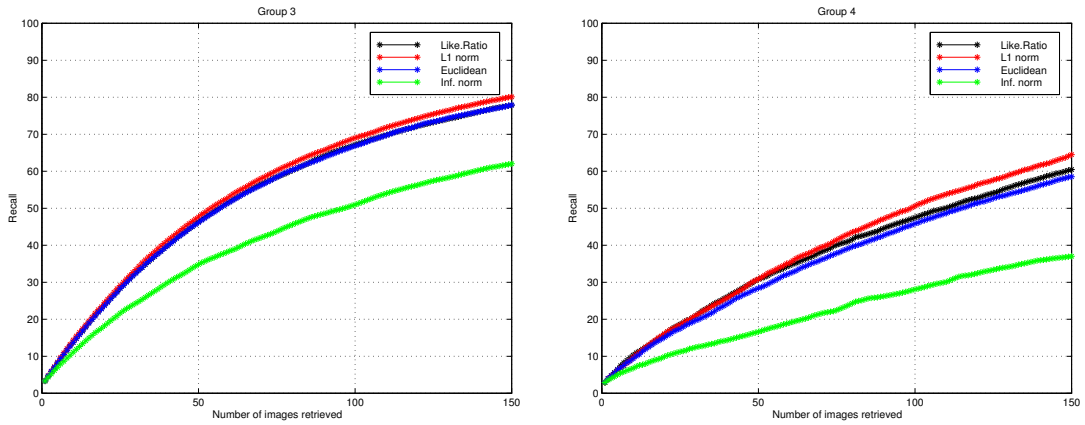
(d) LANDSAT Chernobyl

Figure 7.13: Precision performance tests for the Aerial Image Database using line-angle-ratio features and different distance measures.



(a) Fort Hood Dataset

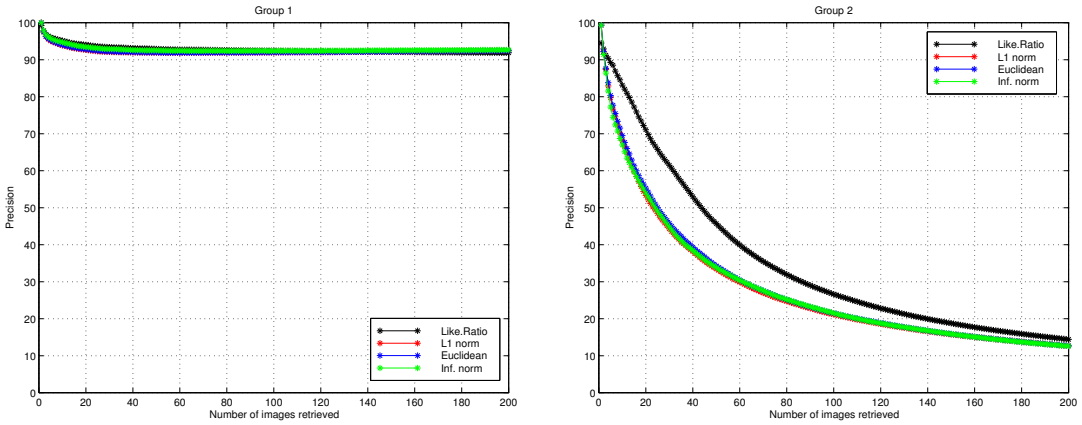
(b) LANDSAT USA



(c) DMSP North Pole

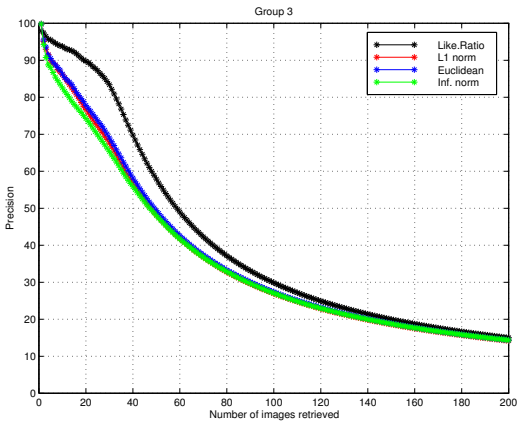
(d) LANDSAT Chernobyl

Figure 7.14: Recall performance tests for the Aerial Image Database using line-angle-ratio features and different distance measures.

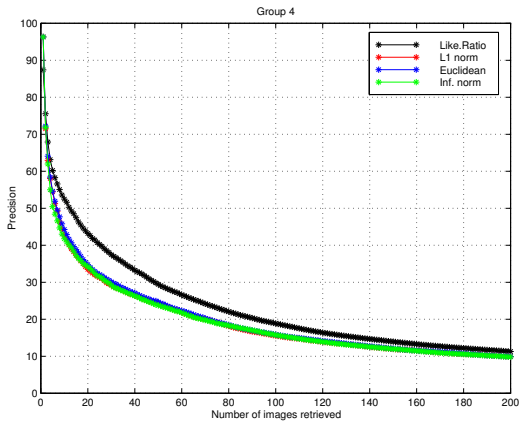


(a) Fort Hood Dataset

(b) LANDSAT USA

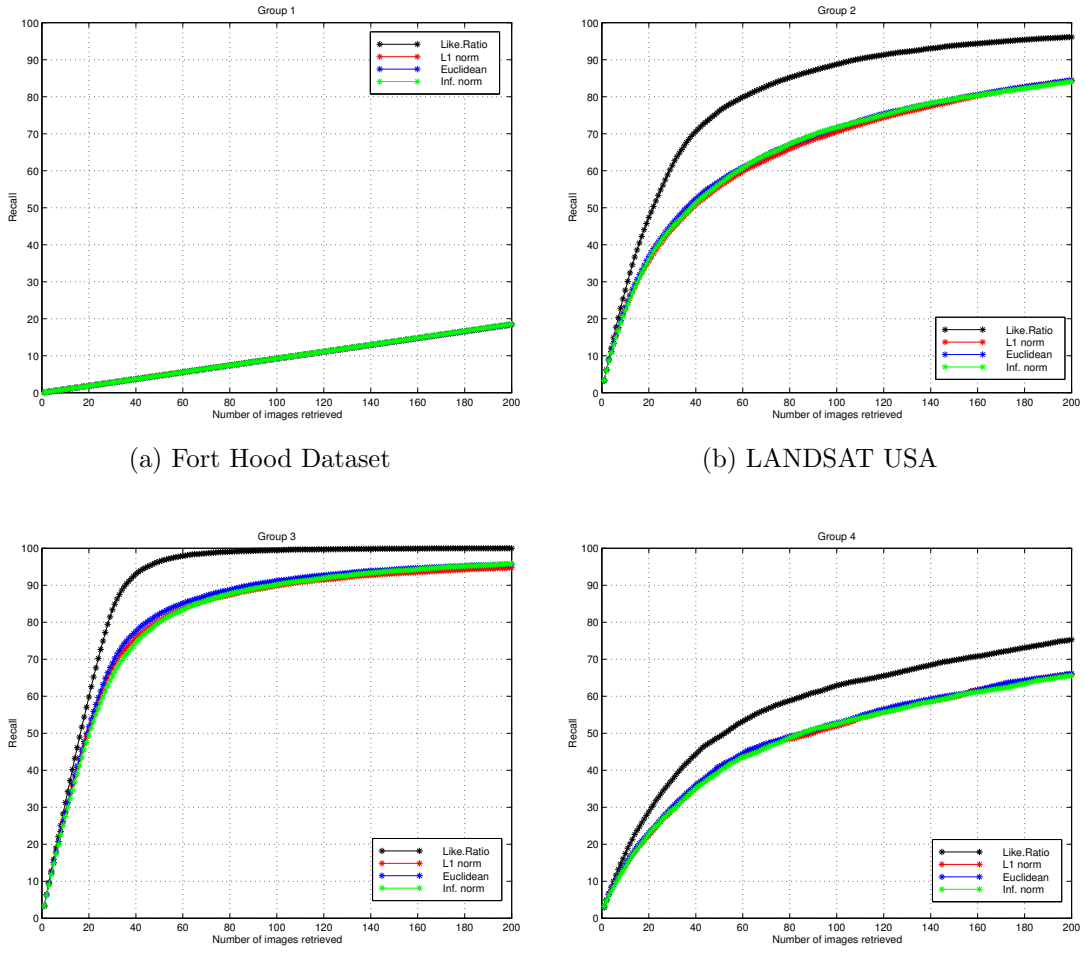


(c) DMSP North Pole



(d) LANDSAT Chernobyl

Figure 7.15: Precision performance tests for the Aerial Image Database using co-occurrence features and different distance measures.



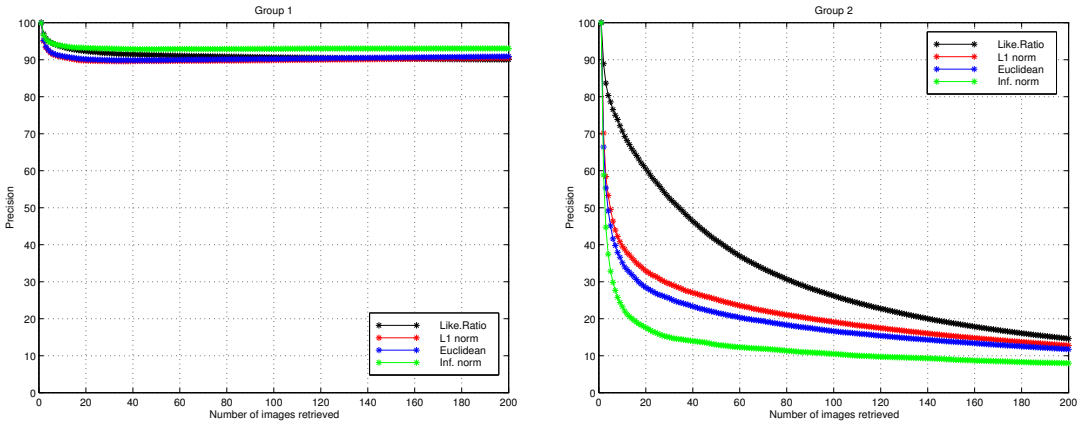
(a) Fort Hood Dataset

(b) LANDSAT USA

(c) DMSP North Pole

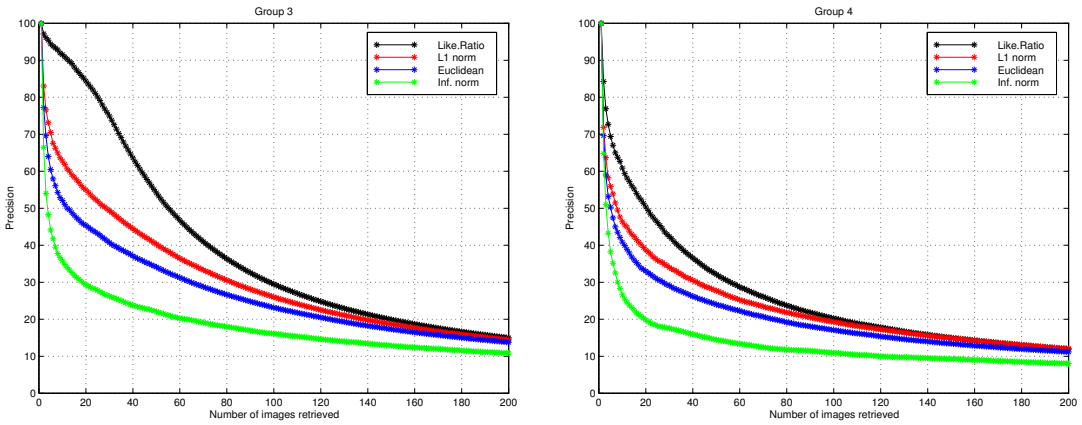
(d) LANDSAT Chernobyl

Figure 7.16: Recall performance tests for the Aerial Image Database using co-occurrence features and different distance measures.



(a) Fort Hood Dataset

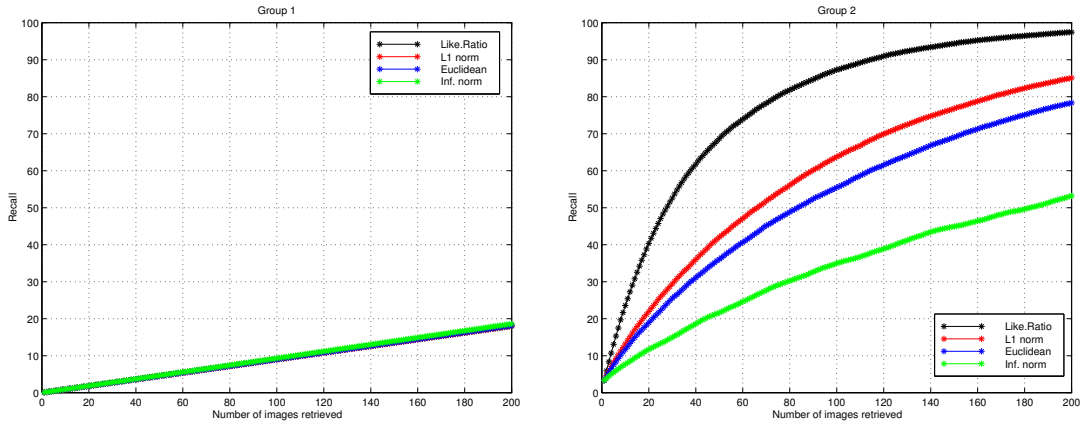
(b) LANDSAT USA



(c) DMSP North Pole

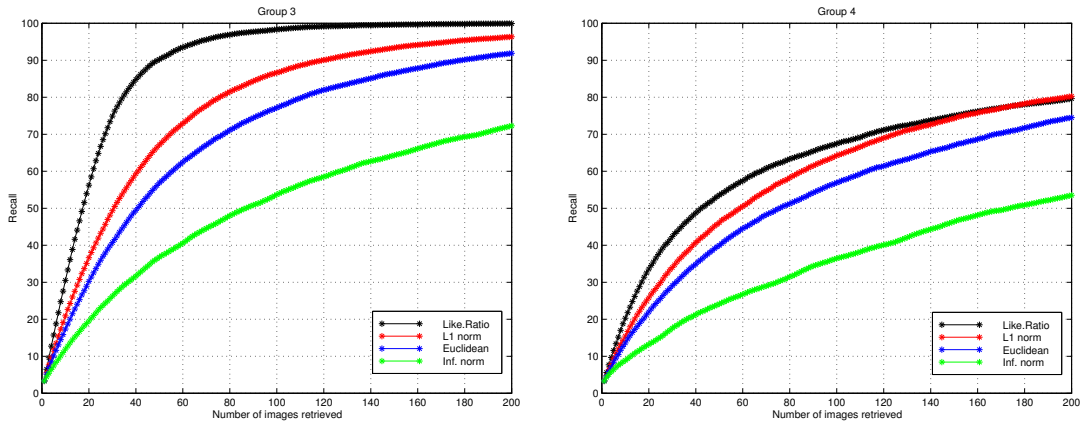
(d) LANDSAT Chernobyl

Figure 7.17: Precision performance tests for the Aerial Image Database using combined features and different distance measures.



(a) Fort Hood Dataset

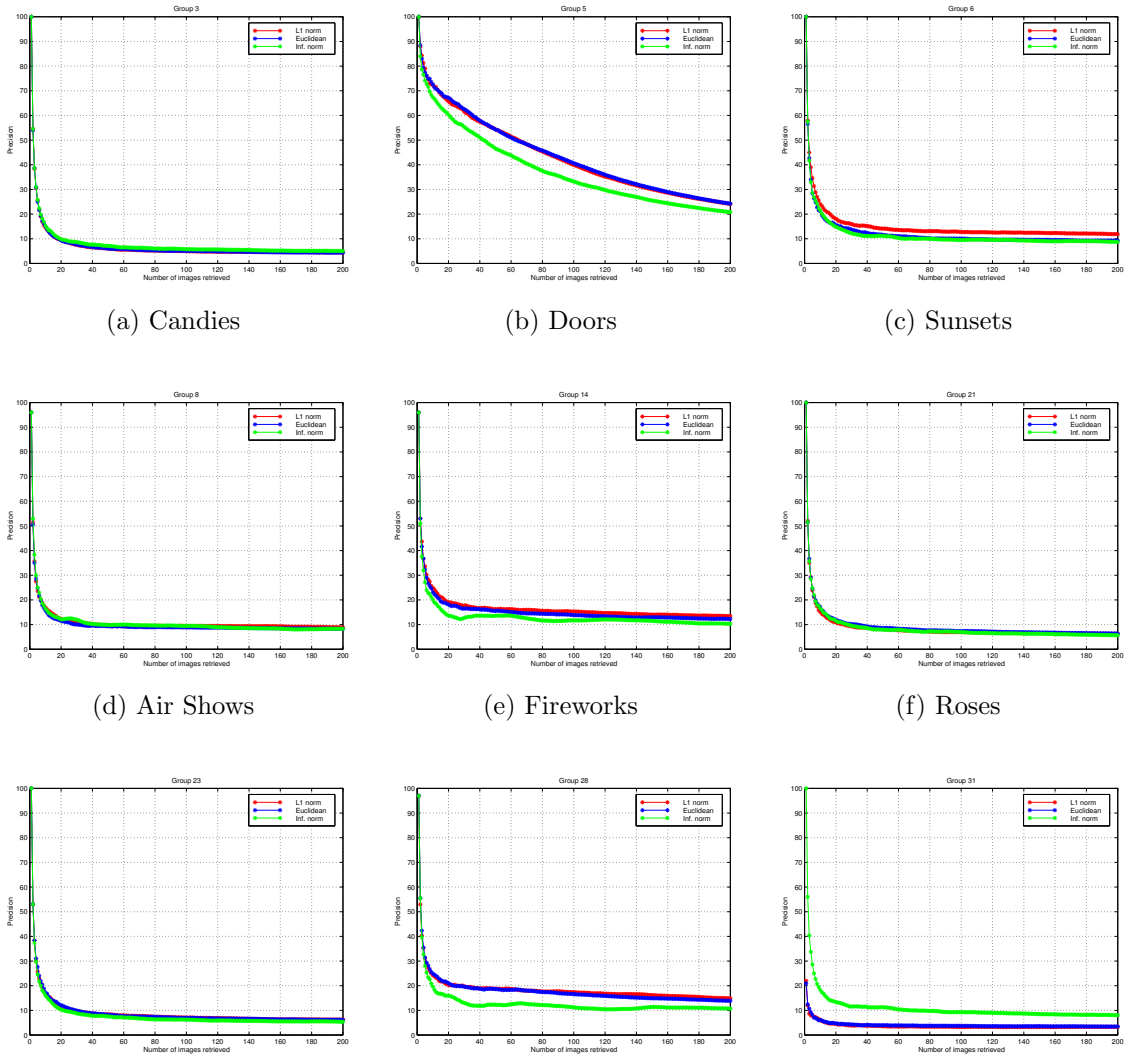
(b) LANDSAT USA



(c) DMSP North Pole

(d) LANDSAT Chernobyl

Figure 7.18: Recall performance tests for the Aerial Image Database using combined features and different distance measures.

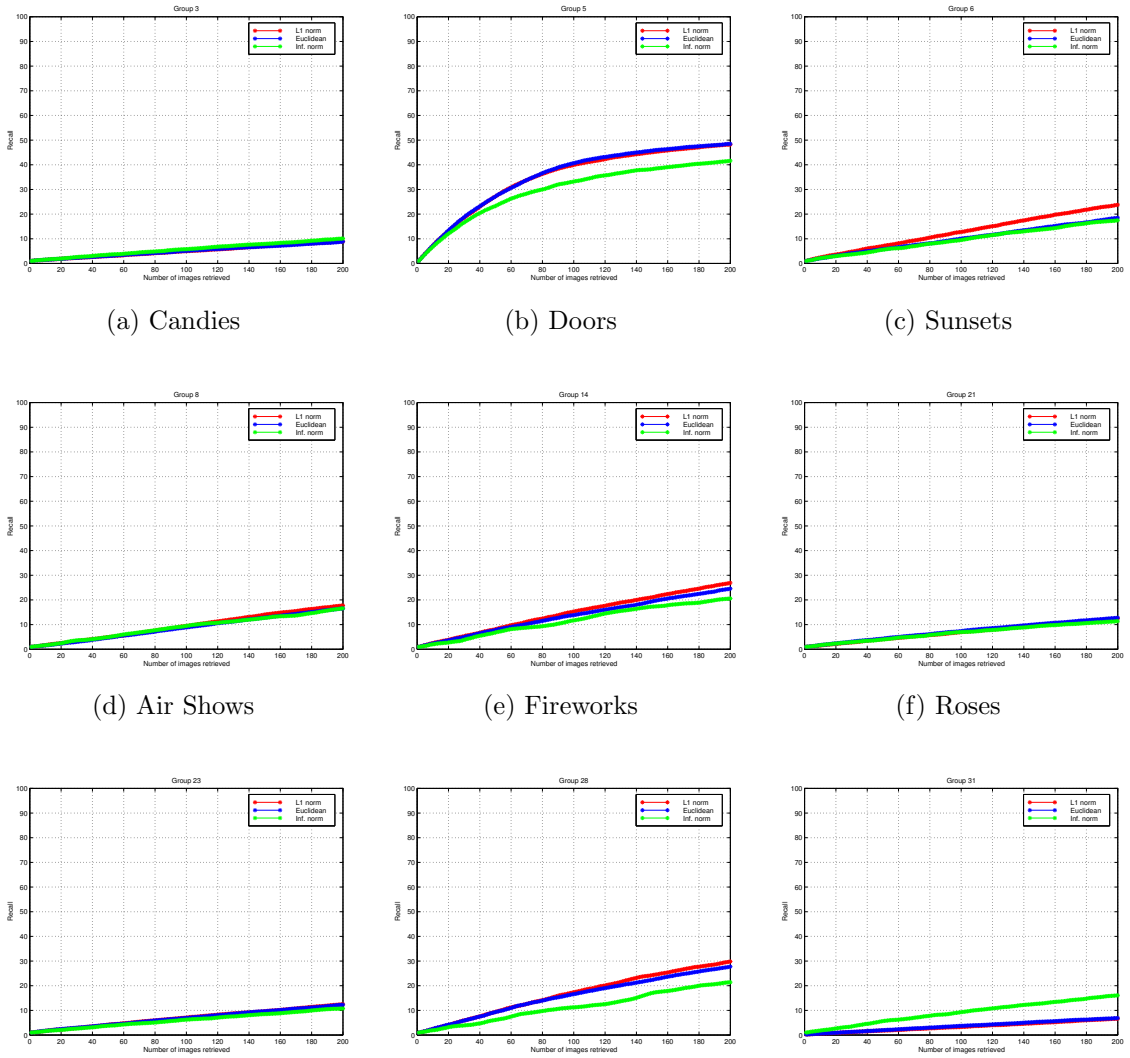


(a) Candies (b) Doors (c) Sunsets

(d) Air Shows (e) Fireworks (f) Roses

(g) Bears (h) Bald Eagles (i) Owls

Figure 7.19: Precision performance tests for the COREL Database using line-angle-ratio features and different distance measures.

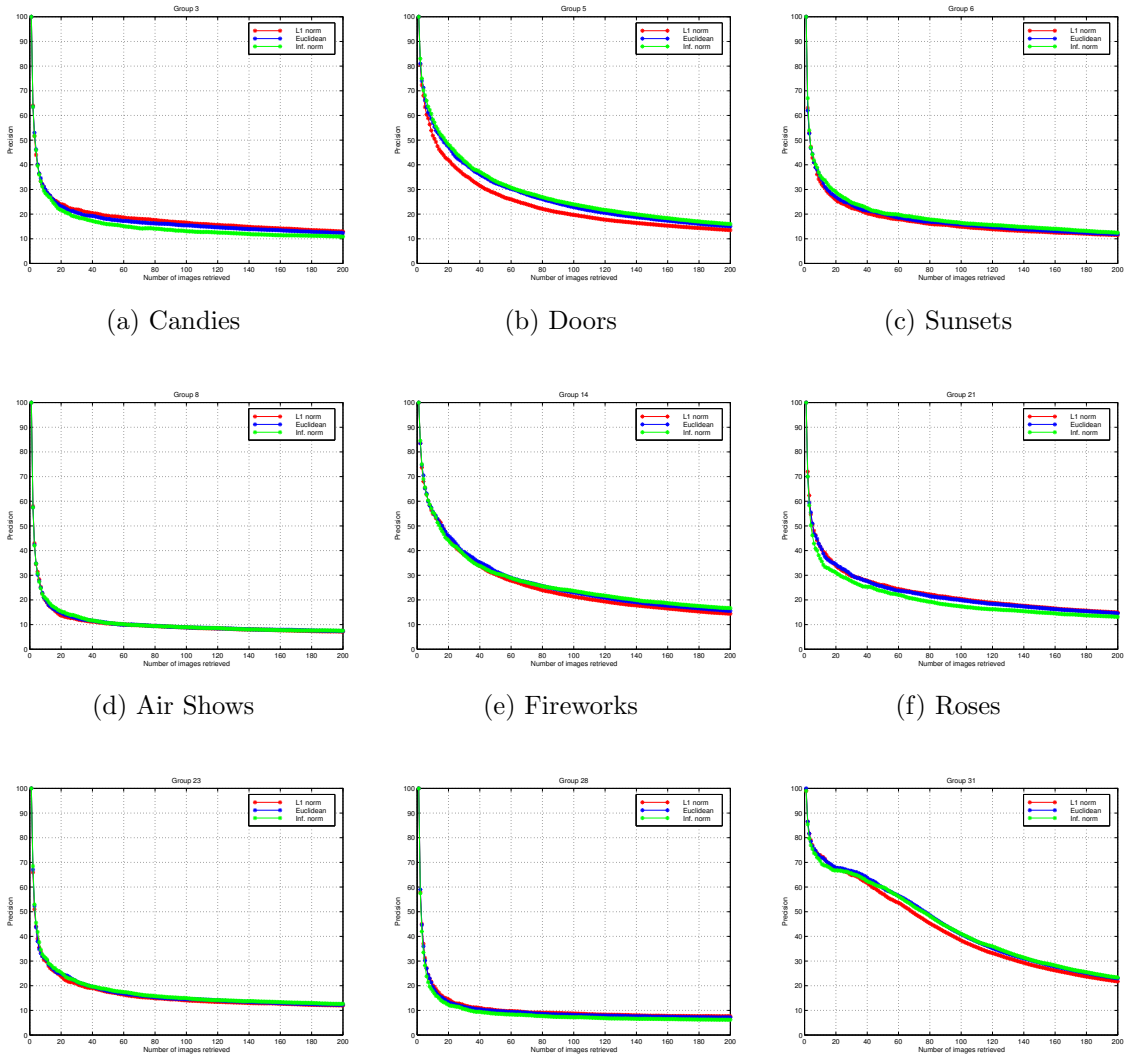


(a) Candies (b) Doors (c) Sunsets

(d) Air Shows (e) Fireworks (f) Roses

(g) Bears (h) Bald Eagles (i) Owls

Figure 7.20: Recall performance tests for the COREL Database using line-angle-ratio features and different distance measures.

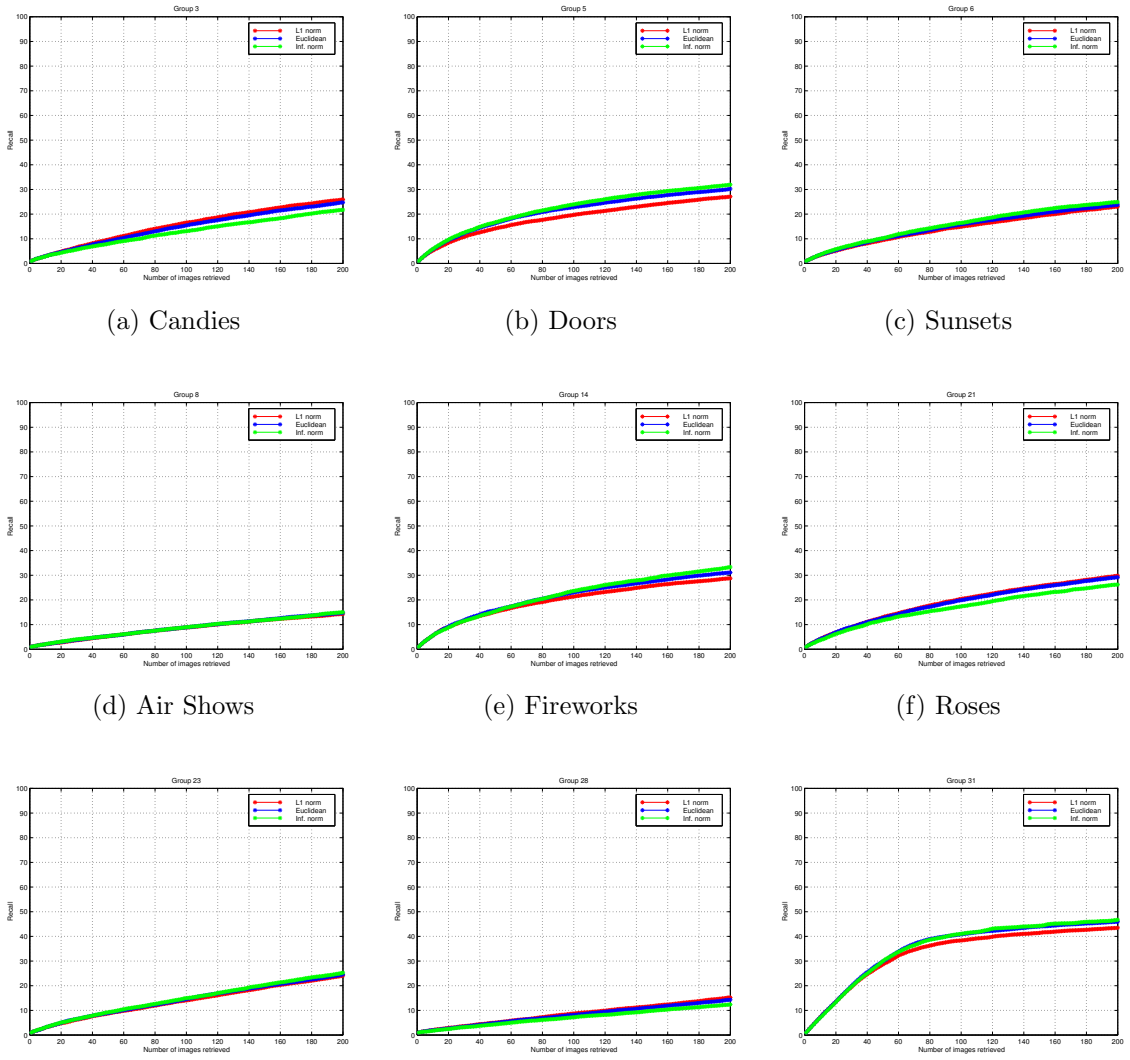


(a) Candies (b) Doors (c) Sunsets

(d) Air Shows (e) Fireworks (f) Roses

(g) Bears (h) Bald Eagles (i) Owls

Figure 7.21: Precision performance tests for the COREL Database using co-occurrence features and different distance measures.

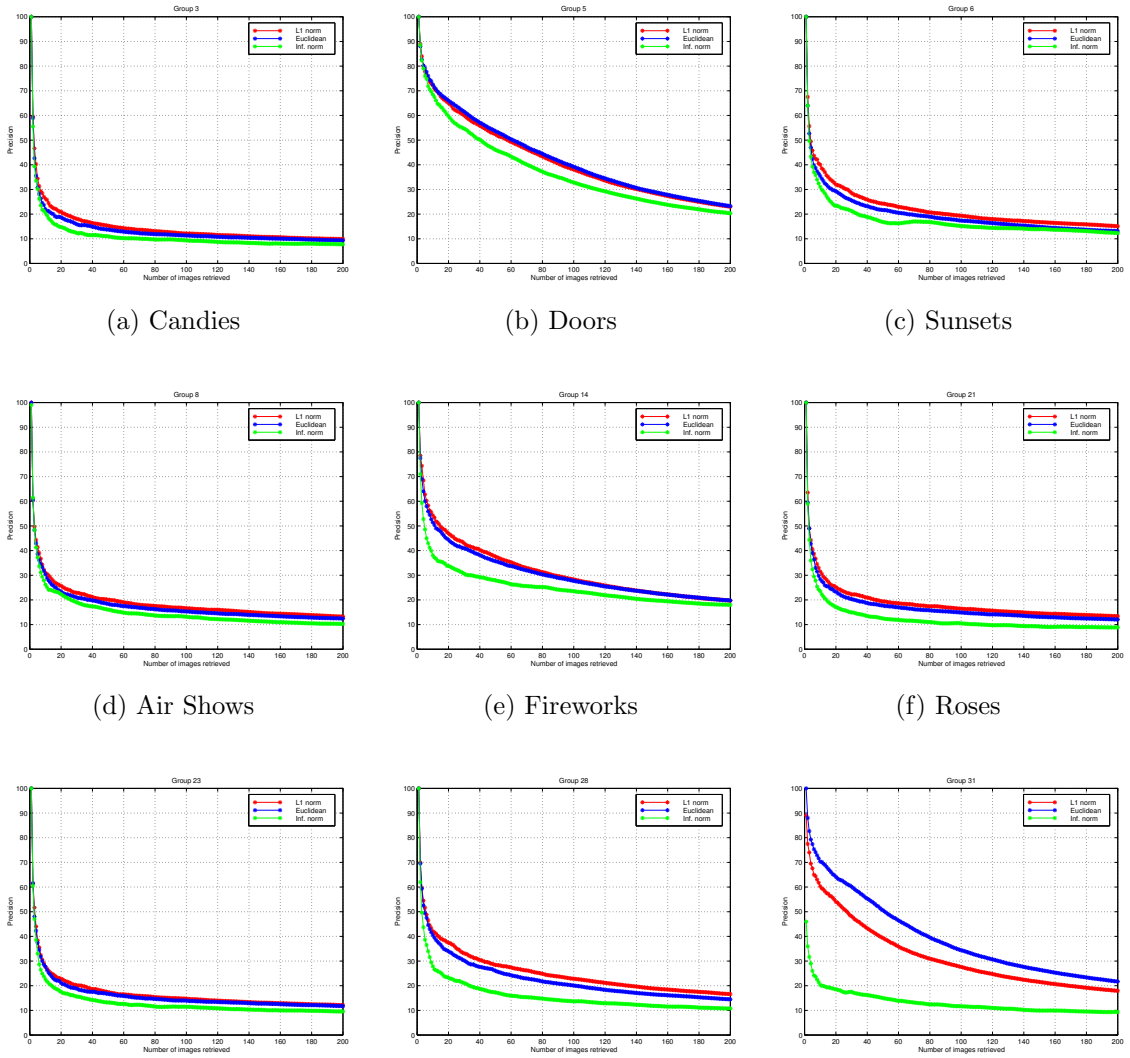


(a) Candies (b) Doors (c) Sunsets

(d) Air Shows (e) Fireworks (f) Roses

(g) Bears (h) Bald Eagles (i) Owls

Figure 7.22: Recall performance tests for the COREL Database using co-occurrence features and different distance measures.

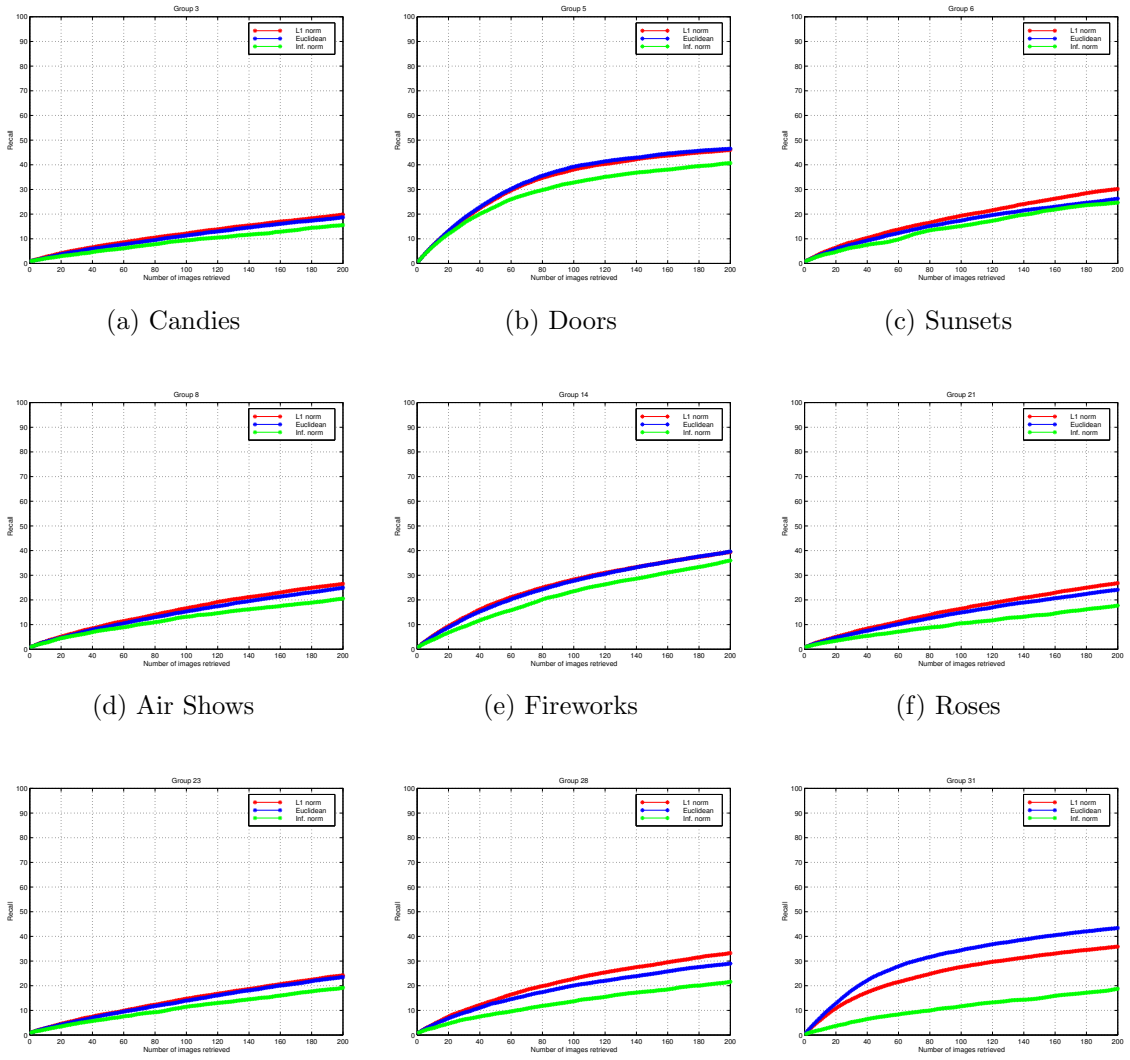


(a) Candies (b) Doors (c) Sunsets

(d) Air Shows (e) Fireworks (f) Roses

(g) Bears (h) Bald Eagles (i) Owls

Figure 7.23: Precision performance tests for the COREL Database using combined features and different distance measures.



(a) Candies (b) Doors (c) Sunsets

(d) Air Shows (e) Fireworks (f) Roses

(g) Bears (h) Bald Eagles (i) Owls

Figure 7.24: Recall performance tests for the COREL Database using combined features and different distance measures.

7.5 Example Queries

"Similarity" is a very abstract concept. Sometimes the user is not exactly sure about what kind of similarity he/she is looking for. Because of this, our system also retrieves the most irrelevant images to help the user understand how the system decides what is relevant and what is not. By looking at these irrelevant images and comparing them with the relevant ones, the user can refine his/her query in a more effective way. A possible extension to this idea is to allow queries like "retrieve images similar to image A and dissimilar to image B".

Some example queries from the Aerial Image Database are given in Figures 7.25–7.32 and example queries from the COREL Database are given in Figures 7.33–7.40. The user interface was written in MATLAB and works both as a parameter adjustment tool for the feature extraction and database retrieval code written in C under UNIX, and also as a display tool for both the query image and the retrieved images. Different feature extraction methods; "line-angle-ratio statistics", "co-occurrence variances" and "combined features" can be selected. Available decision methods are "likelihood ratio", "L^1 norm", "Euclidean distance", "infinity norm", "modified L^1 norm", "modified Euclidean distance" and "modified infinity norm". There are also selections for the number of effective features in the modified distance measures and the number of images to be retrieved as the results of the query. In the display, the upper left image is the query image. Among the retrieved images, first three rows show the most relevant images in descending order of similarity and the last row shows the most irrelevant images in descending order of dissimilarity. The sub-image that best matches the query image is also marked with a white square in each image retrieved.

The rotation invariance characteristics of the line-angle-ratio features can be observed in Figure 7.28, while the angular dependency of the co-occurrence features is visible in Figure 7.25. Note that the individual query results look much better than

the precision and recall results of the previous section that were averaged over all of the images in each group. The reasons for this are the difficulties encountered during assigning complex aerial images into single categories, as well as the inconsistencies, in terms of texture similarity, found in the groundtruth groupings in the COREL Library. These images were grouped in terms of domain similarity. For example, the groundtruth thinks an image of "a man with a parachute" and "an airplane" are similar because they are in the "air shows" group. Similarly, "beans" and "fish" are considered similar because they both belong to the "food" group. The most feasible way of retrieving images using domain information is to integrate keyword matches to the content-based feature similarities. The error pictures are discussed in the next section.

7.6 Analysis of Error Pictures

As noted in the previous section, we believe that most of the errors in the retrieval performance measurements are caused by the problems in groupings for the Aerial and COREL databases. From the example queries and the sample images in Figure 7.3 for the Aerial Image Database, we can see that it is very hard to assign most of the images in the database to specific groups. For example, the image in Figure 7.41(a) contains both "houses" and "landscape" so it can be visually assigned to both of the groups. Therefore, if we assign it to the group "residential", it will be considered as an error picture for queries from the "landscape" group. Other examples are given in Figures 7.41(b) and 7.41(c).

Similarly in the COREL Database, the groupings that are already available as categories in the COREL Library are not exactly consistent with the groupings that can be made visually according to texture similarity. For example, queries using two images that are in the "food" group are given in Figure 7.42. Although the retrieved images "look" quite "similar" to the query images, only the first two of them belong

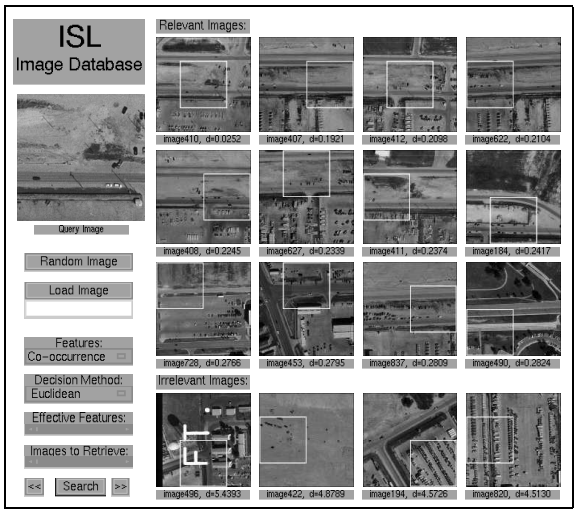


Figure 7.25: Query using co-occurrence variances and Euclidean distance.

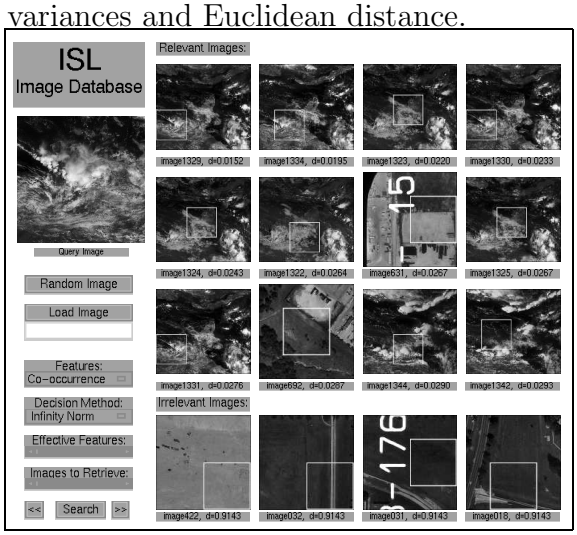


Figure 7.26: Query using co-occurrence variances and infinity norm.

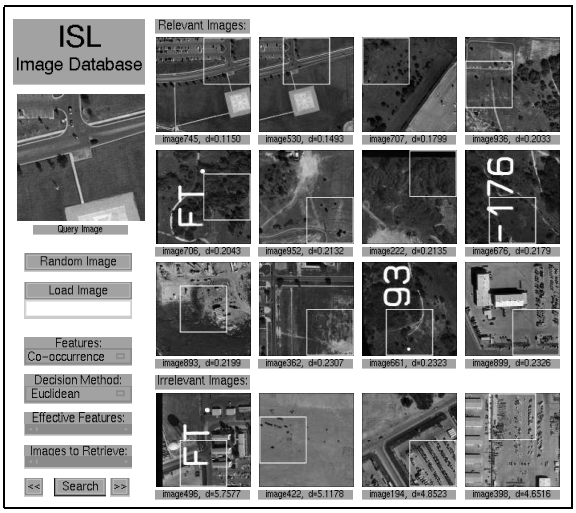


Figure 7.27: Query using co-occurrence variances and Euclidean distance.

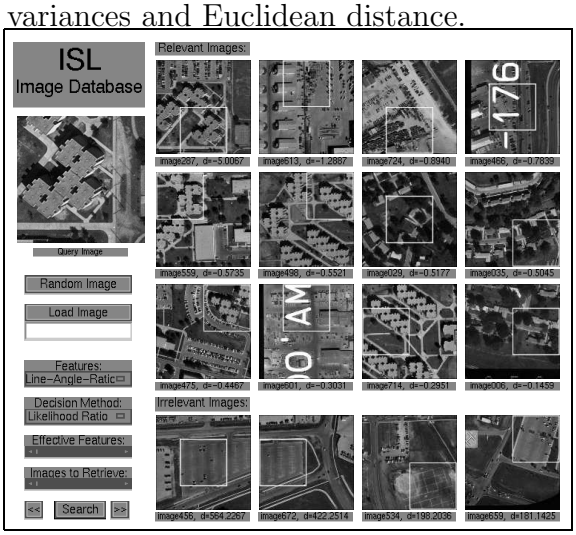


Figure 7.28: Query using line-angle-ratio statistics and likelihood ratio.

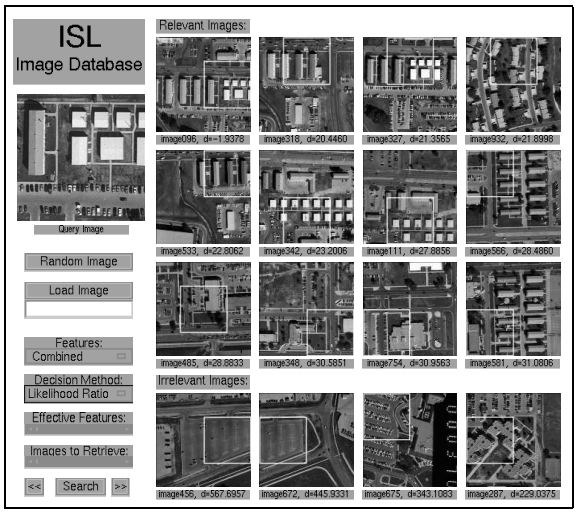


Figure 7.29: Query using combined features and likelihood ratio.

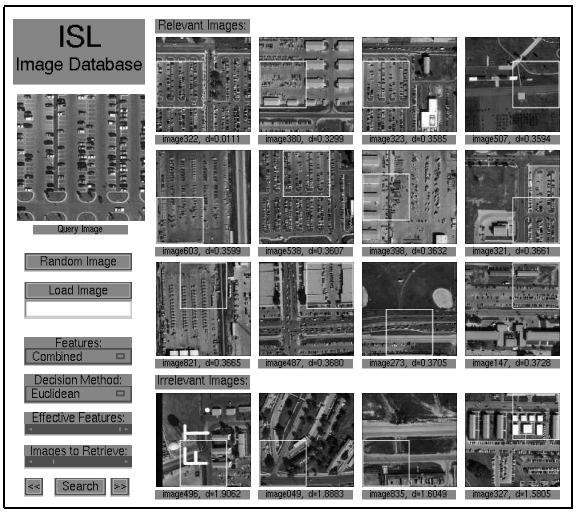


Figure 7.31: Query using combined features and Euclidean distance.

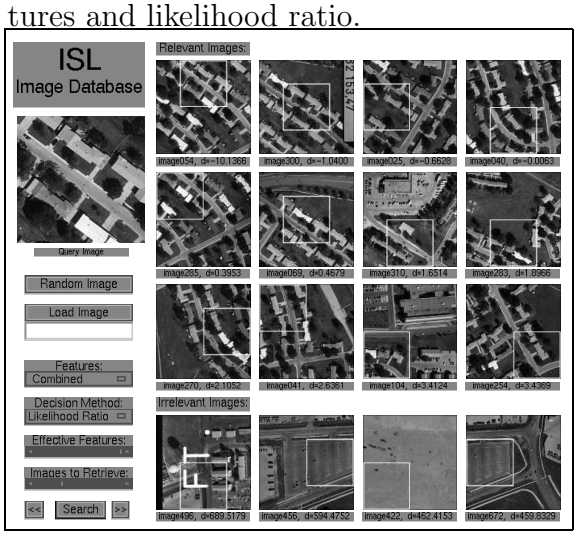


Figure 7.30: Query using combined features and likelihood ratio.

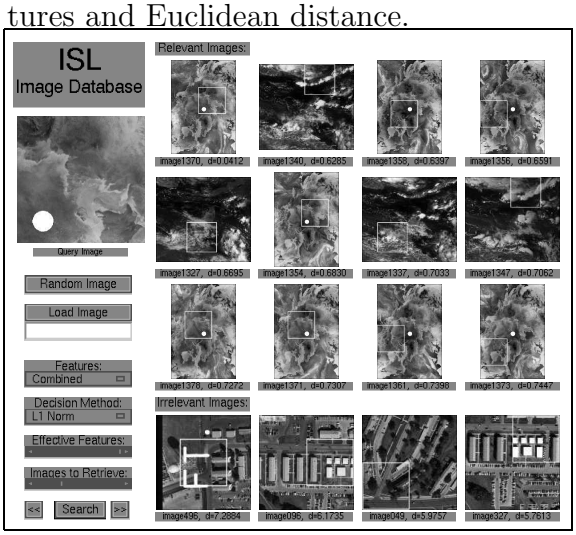


Figure 7.32: Query using combined features and L^1 norm.

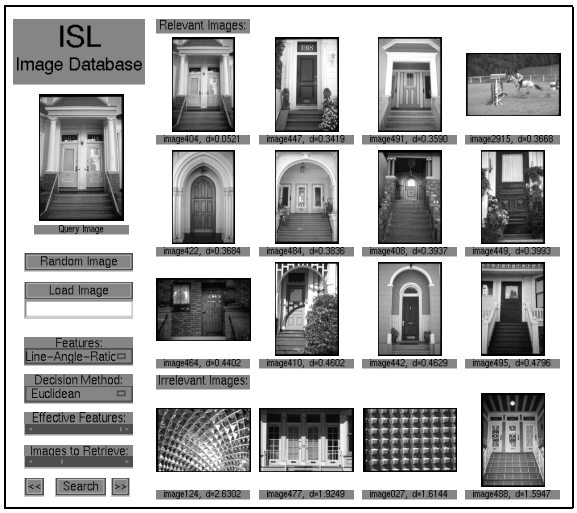


Figure 7.33: Query using line-angle-ratio statistics and Euclidean distance.

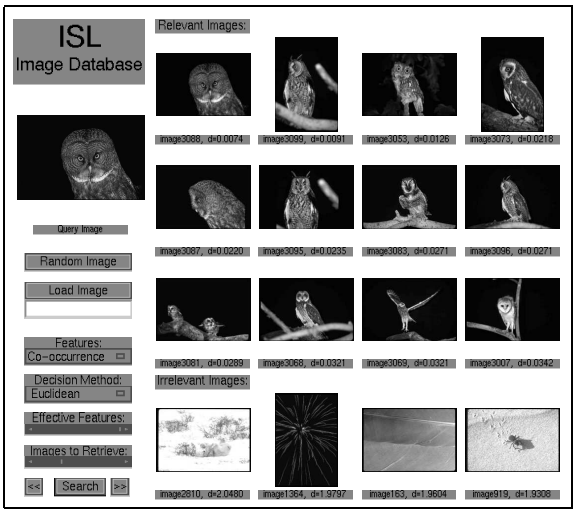


Figure 7.35: Query using co-occurrence variances and Euclidean distance.

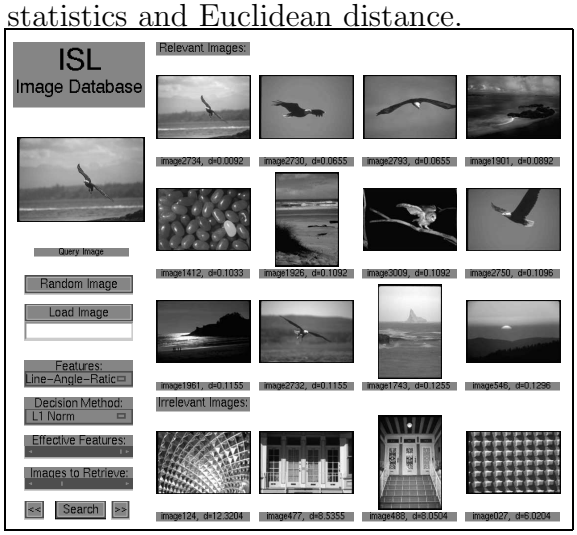


Figure 7.34: Query using line-angle-ratio statistics and L^1 norm.

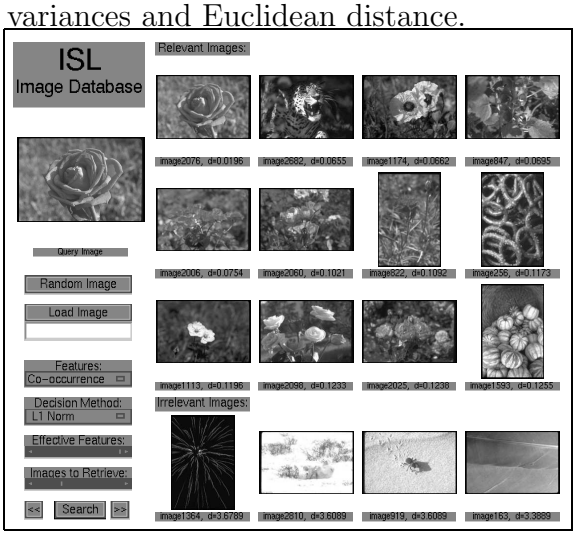


Figure 7.36: Query using co-occurrence variances and L^1 norm.

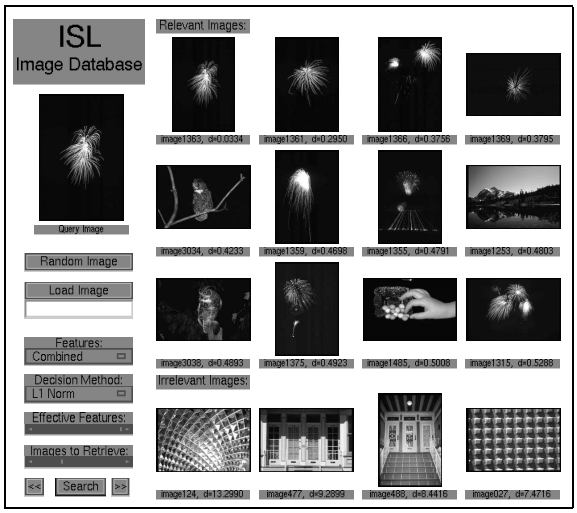


Figure 7.37: Query using combined features and L^1 norm.

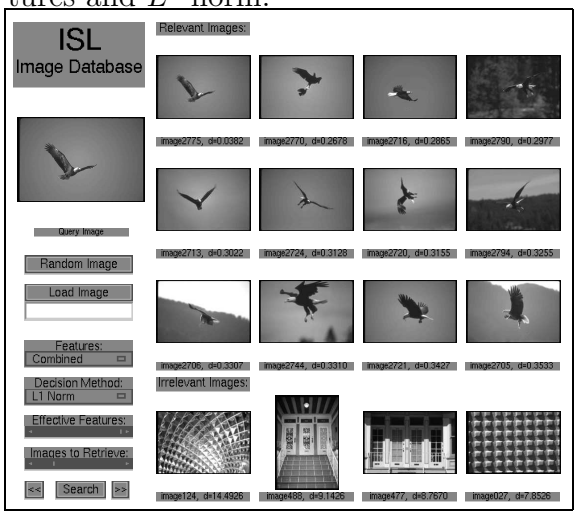


Figure 7.38: Query using combined features and L^1 norm.

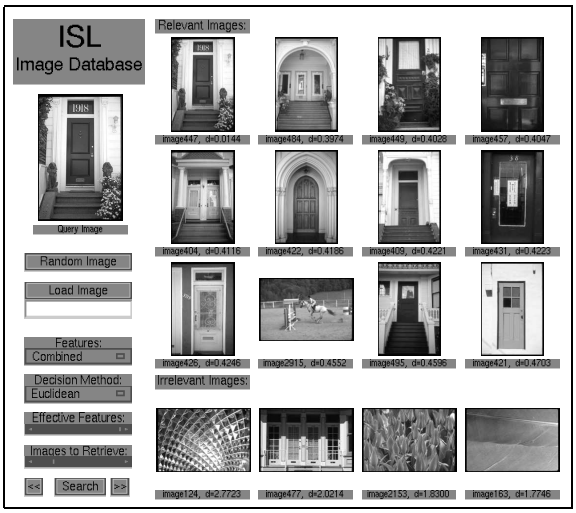


Figure 7.39: Query using combined features and Euclidean distance.

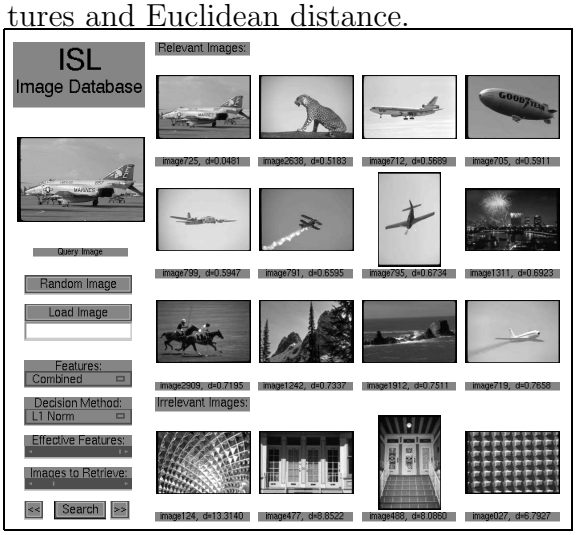


Figure 7.40: Query using combined features and L^1 norm.

to the same group as the query image so the rest are classified as error pictures and make precision decrease drastically in the performance evaluation tests.

We also want to note that, specifically for images like the ones in the COREL Database, color seems to be an important cue in distinguishing images but performance will again be effected by the inconsistencies if we use these groupings. Another important observation is that some images that are quite irrelevant to the query image are retrieved simply because they are close to the query image in the feature space. Further research is required to investigate the distributions of the feature vectors in the high dimensional feature space [31]. Also, new distance measures that resemble the human measure of similarity more closely need to be found. We will address this problem further in Section 8.2.

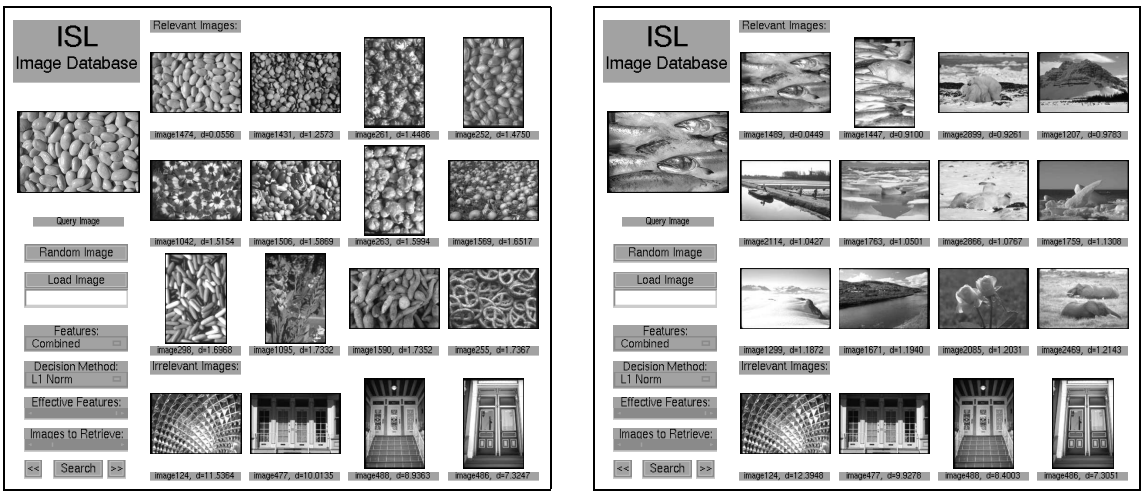
(a) Residential area and landscape.

(b) Buildings and landscape.

(c) Parking lots, buildings and roads.

Figure 7.41: Example Ft. Hood images that can be assigned to more than one group.

These are probably the main reasons why researchers preferred to present only example queries instead of evaluating the performance using objective numerical measures. Only some of them presented experiments using complex and unconstrained images. Among the systems that we reviewed in Chapter 1, [13] presented precision results for a COREL dataset, [8] presented precision–recall as a function of a pa-



(a) Query using a bean image.

(b) Query using a fish image.

Figure 7.42: Queries using two images that are in the "food" group in the COREL Database. Although the retrieved images "look similar" to the query image, most of them are counted as error pictures because they are not in the same group as the query image.

rameter for a LANDSAT dataset, tests in [37] include precision–recall measurements on a small satellite image set as well as a Brodatz texture set, [29] evaluated the performance in terms of the rank of the correct image when the database is queried by the scanned or user sketched version of that image, on the other hand, [41] presented only example queries from a COREL dataset, [47] presented examples from a Brodatz dataset, [22] presented examples from a photo clip collection, examples in [34] are from a small medical dataset and [35] include examples from a small LANDSAT dataset. A common database that includes different kinds of images, that is objectively groundtruthed, and that is publicly available needs to be constructed to evaluate and compare different content-based retrieval systems.

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

In this thesis we discussed a system that allows a user to input an image or a section of an image and retrieves all images from a database having some section similar to the user input image. We defined the problem as developing efficient features for image representation and finding effective measures that use these representations, individually or as a combination, to establish similarity between images. The features and the similarity metrics should be efficient enough to match similar images and also should be able to discriminate dissimilar ones.

The first set of features were computed from the line-angle-ratio statistics which was a texture histogram method that used the spatial relationships between lines as well as the properties of their surroundings as features. Spatial relationships were represented by the angles between intersecting/near-intersecting lines and the properties of their surroundings were represented by the ratio of the mean gray levels inside and outside the regions spanned by those angles. We also developed a line selection algorithm using hypothesis testing to eliminate lines that were not significant enough.

The second set of features were the variances of gray level spatial dependencies and were computed from the co-occurrence matrices which have been demonstrated to be an important texture measure in the micro-texture level by defining texture as specified by the statistical distribution of the spatial relationships of gray level properties. Equal probability quantization was used as a pre-processing step before

computing the co-occurrence matrices.

We also combined these features to make use of their different advantages. A multi-scale texture analysis is crucial for a compact representation, especially for large databases containing different types of complex images.

Statistical feature selection methods were used to select the parameters of the feature extraction algorithms, in order to avoid heuristic selections and redundant features. We defined two classes, the relevance class and the irrelevance class, and designed an automatic groundtruth construction protocol that translated a frame throughout every image and grouped sub-image pairs as relevant or irrelevant according to the assumption that overlapping sub-images are relevant and non-overlapping ones are irrelevant. The criteria for "goodness" of the features was based on the total classification error that was computed using a Gaussian classifier that associated sub-image pairs with either the relevance or the irrelevance classes. Using feature selection methods that both shrink down and build up feature sets, we constructed suboptimal subsets of features that had small classification errors. As a result, the line-angle-ratio feature space was partitioned into 20 cells, and the co-occurrence matrices were computed for 1 and 20 pixel distances and 0, 45, 90, 135 degree orientations.

Given the features for the query image and the images in the database, we used two approaches to rank-order the database images according to their similarities to the query image. First approach used a likelihood ratio that measured the relevancy of two images so that image pairs which had a high likelihood ratio were classified as relevant and the ones which had a lower likelihood ratio were classified as irrelevant. The second approach used the k-nearest neighbor rule and the L^1 norm, Euclidean distance and the infinity norm as the distance measures to find and retrieve the images that were the closest to the query image in the high dimensional feature space. We also modified these distance measures to allow two images to be similar with respect to at least some of the features instead of all of them. The distances can include weights for each feature. The weights were selected using the within-class

and between-class variances of the individual features. The same problem was also defined as a regression problem and the groundtruth sub-image pairs were again used to adjust the weights.

To evaluate the performance, we used two types of experiments, namely the classification effectiveness and the retrieval performance. The classification effectiveness tests used both a Gaussian classifier and a nearest neighbor classifier. We checked whether each groundtruth sub-image pair that should be classified into the relevance or irrelevance classes were classified into the relevance or irrelevance classes correctly. The evaluation criteria was the "total cost" which was defined as 3 misdetections and 2 false alarms. More than 450,000 sub-image pair classifications showed that approximately 80% of the relevance class groundtruth pairs were assigned to the relevance class correctly. Using combined features was proved to be very effective by decreasing the total cost from 30% to 24%. The infinity norm seemed to be dominated by the worst performing features. We believe that this problem can be solved using the modified distance measures. Results for the nearest neighbor classifier were worse than those of the Gaussian classifier because the latter also considered the variances of the two classes and was effected less from the overlap between the class distributions. We observed that although the assumption that overlapping sub-images were relevant almost always held, we could not always guarantee that non-overlapping sub-images were irrelevant. To compansate the effects of "learning from an imperfect teacher", we developed a statistical framework that estimated the correct classification results from the assignments made by the classifier and the mislabeling probabilities of the automatic groundtruth construction protocol. Hence, some of the assignments which we counted as incorrect were not in fact incorrect. Thus the approximate 80% correct relevant pair rate was a lower bound.

In the pair retrieval tests, first the database was queried with each of the sub-images in the groundtruth sub-image pairs, then the success rate and the average rank were computed, where success was defined as retrieving the correct image among a

predefined number of best matches. In more than 180,000 queries to a database of 10,410 sub-images, combined features were again the most successful. The likelihood ratio distance measure we defined was more successful than the other distance measures.

In the precision–recall tests, from approximately 125,000 queries to the Aerial Image Database which consists of 10,410 sub-images, we observed that the line-angle-ratio features performed better for the images from the Ft. Hood Dataset, whereas the co-occurrence features performed better for the images from the Remote Sensing Dataset. From approximately 28,000 queries to the COREL database which consists of 3,100 images, we again observed that the line-angle-ratio features were more successful for the images with a dominant line information and the co-occurrence features performed better when the micro-texture characteristics were dominant. Combining these two feature sets always outperformed the results for individual features.

The major problem we had in these experiments was in both the sub-image-level and the image-level groundtruths. As noted before, a widely accepted database needs to be generated to evaluate and compare different content-based retrieval systems.

We believe that our textural features and decision methods played a significant role in capturing similarities between images and retrieving the ones that are similar to the query image. We successfully showed that low-level textural features can help in grouping images into semantically meaningful categories. All of the experiments done on a large database with many different kinds of complex images showed that combined features outperformed individual features, which leads to the conclusion that a significant improvement in both classification effectiveness and retrieval performance was obtained using a multi-scale texture representation. Also the likelihood ratio measure we developed increased the retrieval efficiency compared to the commonly used methods like Euclidean distance and L^1 norm. To further improve the performance, some future research directions are suggested in the next section.

8.2 Future Work

In order to further improve the performance and to overcome the incapabilities of our features in representing some of the images that do not contain significant line information and that cannot be distinguished at the micro–texture level, we will try to add, with appropriate weightings, new features that capture texture information at other scales. Color is also another important cue for visual similarity. Integrating color features with the textural features developed here will definitely improve the performance for the COREL Database. Note that we used neither the weighted distance measures nor the modified distance measures in the experiments. The weights are still have to be estimated. The number of features to be used in the modified distance measures can be selected by first running the nearest neighbor classification tests for different numbers of features and then choosing the best performing subset size. An alternative is to try all possible numbers of features but this is computationally too expensive. Also the mislabeling probabilities of the automatic groundtruth construction protocol still need to be estimated in order to be able to use the framework developed to estimate the correct classification results. Finally, the features developed here should be compared to other techniques to observe the actual degree of improvement made.

One final observation is that sometimes images that are quite irrelevant to the query image are also retrieved simply because they are close to the query image in the feature space. Further research is required to find features and distance measures that resemble the human visual system more closely. One decision method that is worth trying is to use a graph-theoretic approach. Let's assume we query the database and get back the best N matches. Then, for each of these N matches we can do a query and get back the best N matches again. Now we have a minimum of N and a maximum of N^2 retrieved images. Let's define S as the set of images that are retrieved as the results of these queries. Then, we can construct a graph with

the images in S as the nodes. We can draw arcs between the N query images and their corresponding N matches. These will be the edges of the graph. We define this as the set R where $R \subseteq S \times S$. These edges can also have the distances between images as weights. We want to find the connected clusters of this graph because they correspond to similar images. The problem now becomes finding the maximal P, where $P \subseteq S$ such that $P \times P \subseteq R$. This is also called the maximal clique of this graph. To increase the speed, we can use the algorithm developed by Shapiro and Haralick [53] that finds "near-cliques" by using dense regions instead of the maximally connected ones in the graph. The clusters of interest are the ones that include the original query image. If two clusters contain the same number of nodes (images), we can compute the sum of the weights of the edges in each of the clusters and select the one that has the minimum total distance. This method increases the chance of retrieving similar images by not only ensuring that the retrieved images are close to the query image, but also adding another constraint that they should be close to each other in the feature space.

BIBLIOGRAPHY

[1] COREL Photo Stock Library 1. Online information: http://wp.novell.com/products/clipartandphotos/photos/.

[2] Fort Hood Datasets. Online information: http://www.mbvlab.wpafb.af.mil/public/sdms/datasets/fthood/index.htm.

[3] A. K. Agrawala. Learning with a probabilistic teacher. *IEEE Transactions on Information Theory*, 16(4):373–379, July 1970.

[4] S. Aksoy and R. M. Haralick. Content-based image database retrieval using variances of gray level spatial dependencies. In *Proceedings of IAPR International Workshop on Multimedia Information Analysis and Retrieval*, Hong Kong, August 1998.

[5] S. Aksoy and R. M. Haralick. Textural features for image database retrieval. In *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries, in conjunction with CVPR'98*, Santa Barbara, CA, June 1998.

[6] S. Aksoy, M. L. Schauf, and R. M. Haralick. Content-based image database retrieval based on line-angle-ratio statistics. Technical Report ISL-TR, Intelligent Systems Lab., University of Washington, Seattle, WA, November 1997.

[7] J. Ashley, R. Barber, M. Flickner, J. Hafner, D. Lee, W. Niblack, and D. Petkovic. Automatic and semi-automatic methods for image annotation and retrieval in QBIC. In *SPIE Storage and Retrieval of Image and Video Databases*, volume 2420, pages 24–35, 1995.

[8] J. Barros, J. French, W. Martin, and P. Kelly. System for indexing multi-spectral satellite images for efficient content based retrieval. In *SPIE Storage and Retrieval of Image and Video Databases III*, February 1995.

[9] J. Barros, J. French, W. Martin, P. Kelly, and M. Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. In *SPIE Storage and Retrieval of Image and Video Databases IV*, January 1996.

[10] A. Berman and L. G. Shapiro. Efficient image retrieval with multiple distance measures. In *SPIE Storage and Retrieval of Image and Video Databases*, pages 12–21, February 1997.

[11] A. Berman and L. G. Shapiro. Selecting good keys for triangle-inequality-based pruning algorithms. In *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Databases*, pages 12–18, January 1998.

[12] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[13] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Region-based image querying. In *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997.

[14] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury Press, California, 1990.

[15] Y. T. Chien. A sequential decision model for selecting feature subsets in pattern recognition. *IEEE Transactions on Computers*, C-20(3):282–290, March 1971.

[16] R. W. Conners and C. A. Harlow. Some theoretical considerations concerning texture analysis of radiographic images. In *Proceedings of the 1976 IEEE Conference on Decision and Control*, pages 162–167, 1976.

[17] R. W. Conners and C. A. Harlow. Equal probability quantizing and texture analysis of radiographic images. *Computer Graphics and Image Processing*, 8:447–463, 1978.

[18] R. W. Conners and C. A. Harlow. A theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(3):25–39, May 1980.

[19] R. W. Conners and C. A. Harlow. Toward a structural textural analyzer based on statistical methods. *Computer Graphics and Image Processing*, 12:224–256, March 1980.

[20] L. Davis, S. Johns, and J. K. Aggarwal. Texture analysis using generalized co-occurrence matrices. In *Proceedings of 1978 IEEE Conference on Pattern Recognition and Image Processing*, pages 313–318, Chicago, IL, May 1978.

[21] A. Etemadi. Robust segmentation of edge data. In *IEE Image Processing Conference*, 1992.

[22] 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. The QBIC project: Querying images by content using color, texture and shape. In *SPIE Storage and Retrieval of Image and Video Databases*, pages 173–181, 1993.

[23] C. C. Gotlieb and H. E. Kreyszig. Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics, and Image Processing*, 51(1):70–86, July 1990.

[24] V. N. Gudivada and V. V. Raghavan. Content-based image retrieval systems. *IEEE Computer Magazine*, 28(9):18–22, September 1995.

[25] R. M. Haralick. A texture-context feature extraction algorithm for remotely sensed imagery. In *Proceedings of the 1971 IEEE Conference on Decision and Control*, pages 650–657, Gainesville, FL, December 1971.

[26] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.

[27] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, November 1973.

[28] R. M. Haralick and L. G. Shapiro. Glossary of computer vision terms. *Pattern Recognition*, 24(1):69–93, January 1991.

[29] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *Proceedings of SIGGRAPH'95*, pages 277–285, Los Angeles, CA, August 1995.

[30] A. K. Jain and R. Dubes. Feature definition in pattern recognition with small sample size. *Pattern Recognition*, 10(2):85–97, 1978.

[31] J. W. Sammon Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, May 1969.

[32] B. Julesz. Visual pattern discrimination. *IRE Transactions on Information Theory*, pages 84–92, February 1962.

[33] B. Julesz. Experiments in the visual perception of texture. *Scientific American*, pages 34–43, April 1975.

[34] P. M. Kelly and T. M. Cannon. CANDID: Comparison algorithm for navigating digital image databases. In *Proceedings of the Seventh International Working Conference on Scientific and Statistical Database Management*, pages 252–258, September 1994.

[35] P. M. Kelly, T. M. Cannon, and D. R. Hush. Query by image example: The CANDID approach. In *SPIE Storage and Retrieval of Image and Video Databases III*, pages 238–248, 1995.

[36] K. I. Laws. Rapid texture classification. In *SPIE Image Processing for Missile Guidance*, volume 238, pages 376–380, 1980.

[37] C. S. Li and V. Castelli. Deriving texture set for content based retrieval of satellite image database. Technical Report RC20727, IBM T.J. Watson Research Center, Yorktown Heights, NY, February 1997.

[38] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28:84–95, January 1980.

[39] F. Liu and R. W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):722–733, July 1996.

[40] G. Lugosi. Learning with an unreliable teacher. *Pattern Recognition*, 25(1):79–87, January 1992.

[41] W. Y. Ma and B. S. Manjunath. NETRA: A toolbox for navigating large image databases. In *Proceedings of IEEE International Conference on Image Processing*, 1997.

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

[43] J. Mao and A. K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188, 1992.

[44] T. P. Minka and R. W. Picard. Interactive learning using a "society of models". *Pattern Recognition Special Issue on Image Databases: Classification and Retrieval*, 1996. also appears as MIT Media Lab. Perceptual Computing Section Tech. Rep. No: 349, 1995.

[45] P. M. Narendra and K. Fukunage. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, C-26(9):917–922, September 1977.

[46] P. P. Ohanian and R. C. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25(8):819–833, August 1992.

[47] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. In *SPIE Storage and Retrieval of Image and Video Databases II*, pages 34–47, February 1994.

[48] R. W. Picard and A. P. Pentland. Introduction to the special section on digital libraries: Representation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):769–770, August 1996.

[49] A. Rosenfeld. Visual texture analysis: An overview. Technical Report TR-406, University of Maryland, College Park, MD, August 1975.

[50] A. Rosenfeld and E. B. Troy. Visual texture analysis. In *Conference Record for Symposium on Feature Extraction and Selection in Pattern Recognition*, pages 115–124, Argonne, IL, October 1970. IEEE Publication: 70C-51C.

[51] G. Salton. *Automatic Information Organization and Retrieval*. McGraw-Hill, 1968.

[52] K. Shanmugam and A. M. Breipohl. An error correction procedure for learning with an imperfect teacher. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3):223–229, July 1971.

[53] L. G. Shapiro and R. M. Haralick. Decomposition of two-dimensional shapes by graph-theoretic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(1):10–20, January 1979.

[54] J. R. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression*. PhD thesis, Columbia University, 1997.

[55] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-8(6):460–473, June 1978.

[56] J. T. Tou and Y. S. Chang. Picture understanding by machine via textural feature extraction. In *Proceedings of 1977 IEEE Conference on Pattern Recognition and Image Processing*, pages 392–399, Troy, NY, June 1977.

[57] M. Tuceryan and A. K. Jain. *Handbook of Pattern Recognition and Computer Vision*, chapter Texture Analysis, pages 235–276. World Scientific Publishing Company, River Edge, NJ, 1993.

[58] A. Vailaya, A. Jain, and H. J. Zhang. On image classification: City vs. landscape. In *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries, in conjunction with CVPR'98*, Santa Barbara, CA, June 1998.

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

[60] A. Whitney. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 20(9):1100–1103, September 1971.

[61] Y. Yakimovsky. Boundary and object detection in real world images. *Journal of the ACM*, 23(4):599–618, October 1976.

[62] Q. Zhou. Line drawing symbol recognition. Master's thesis, University of Washington, Seattle, WA, 1996.

[63] S. W. Zucker and D. Terzopoulos. Finding structure in co-occurrence matrices for texture analysis. *Computer Graphics and Image Processing*, 12:286–308, March 1980.

Appendix A

LINE AND ANGLE PRELIMINARIES

In this appendix we present some definitions that are used in the derivations in Chapter 2, Line-Angle-Ratio Statistics.

Definition 1 *Parametric line equation*

Given two n-dimensional end points of a line segment, P_1 and P_2, the equation of it can be written as

$$P = P_1 + \lambda(P_2 - P_1) \tag{A.1}$$

where

$$\lambda = \frac{||PP_1||}{||P_2P_1||}, \tag{A.2}$$

a real constant between 0 and 1.

In 2-D, $P = \begin{bmatrix} r \\ c \end{bmatrix}$, so for a line segment with end points $\begin{bmatrix} r_1 \\ c_1 \end{bmatrix}$ and $\begin{bmatrix} r_2 \\ c_2 \end{bmatrix}$, equation (A.1) becomes

$$\begin{bmatrix} r \\ c \end{bmatrix} = \begin{bmatrix} r_1 \\ c_1 \end{bmatrix} + \lambda \begin{bmatrix} r_2 - r_1 \\ c_2 - c_1 \end{bmatrix}. \tag{A.3}$$

Definition 2 *Angle between two lines*

The angle θ between two directed lines $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_1P_3}$ as in Figure A.1 is given by

$$\theta = \frac{180}{\pi} \arccos\left(\frac{(P_2 - P_1) \cdot (P_3 - P_1)}{\|P_2 - P_1\|\|P_3 - P_1\|}\right). \tag{A.4}$$

Resulting θ is always in the range $[0°, 180°]$.

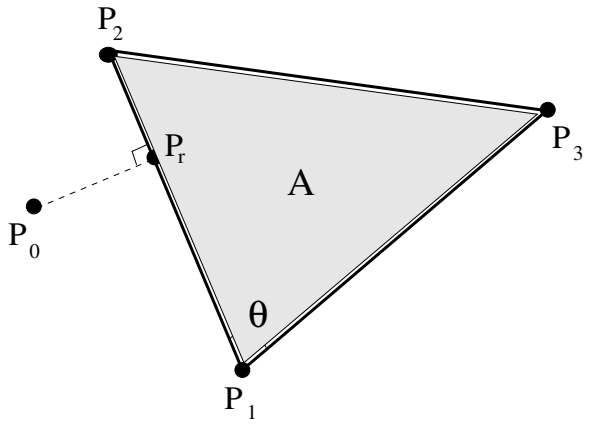


Figure A.1: Lines and points in 2-D space.

Definition 3 *Distance between a point and a line*

Given the line P_1P_2 with equation as in (A.1) and a point P_0 as illustrated in 2-D in Figure A.1, λ_0 that minimizes the distance between the point and the line is given by

$$\lambda_0 = \frac{(P_0 - P_1)'(P_2 - P_1)}{\|P_2 - P_1\|^2}. \tag{A.5}$$

Then, P_r, which is the point on the line that is closest to P_0, can be found as

$$P_r = P_1 + \lambda_0(P_2 - P_1). \tag{A.6}$$

Finally, the smallest distance can be computed as

$$d = \|P_r - P_0\|. \tag{A.7}$$

Definition 4 *Area of a triangle*

Given three vertices P_1, P_2, P_3 of a triangle in 2-D space as in Figure A.1, its area can be computed as

$$A = \frac{1}{2}[P_{1r}P_{2c} - P_{1c}P_{2r} + P_{1c}P_{3r} - P_{1r}P_{3c} + P_{2r}P_{3c} - P_{2c}P_{3r}]. \tag{A.8}$$

Note that this is a signed area. If the vertices P_1, P_2, P_3 form a counterclockwise circuit, the area is positive, if they form a clockwise circuit, the area is negative.

Definition 5 *Location of a point with respect to a line*

To determine which side of a directed line $\overrightarrow{P_1P_2}$ a point P_0 lies, first the area of the triangle $P_0P_1P_2$ is computed using equation (A.8). If this area is positive, the point P_0 lies on the left of the line $\overrightarrow{P_1P_2}$, otherwise it lies on the right.

Appendix B

REGION CONVENTION FOR MEAN COMPUTATION

In this appendix we describe the procedure to find the regions defined as *in* and *out* regions in Section 2.2.2. Given end points of two lines L_1 and L_2 as in Figure B.1, we want to find the regions to calculate mean gray levels. After computing the unit vector $\hat{n}_1 = \begin{bmatrix} n_{1r} \\ n_{1c} \end{bmatrix}$ along L_1 and the unit vector $\hat{n}_2 = \begin{bmatrix} n_{2r} \\ n_{2c} \end{bmatrix}$ along L_2, we can find their intersection point $P = \begin{bmatrix} P_r \\ P_c \end{bmatrix}$ using the approach described in Section 2.1.2. To find the region boundaries, first we find the angle bisector line L_3 between L_1 and L_2. Equation of L_3 is

$$L_3 = P + \lambda_3 \hat{n}_3 \tag{B.1}$$

where $\hat{n}_3$ is the unit vector along L_3 and λ_3 is a real constant. Note that this equation is slightly different than the one in equation (A.1) because in (B.1) $\hat{n}_3$ is the unit vector. To find the unit vector $\hat{n}_3$ we use the constraints

$$\hat{n}_3 \cdot \hat{n}_1 = \hat{n}_3 \cdot \hat{n}_2 \tag{B.2}$$

and

$$\|\hat{n}_3\| = 1. \tag{B.3}$$

These result to

$$n_{3r} n_{1r} + n_{3c} n_{1c} = n_{3r} n_{2r} + n_{3c} n_{2c} \tag{B.4}$$

and

$$n_{3r}^2 + n_{3c}^2 = 1. \tag{B.5}$$

Then, we can derive n_{3c} as

$$n_{3c} = \pm\sqrt{\frac{(n_{2r} - n_{1r})^2}{(n_{2c} - n_{1c})^2 + (n_{2r} - n_{1r})^2}} \tag{B.6}$$

and n_{3r} as

$$n_{3r} = -n_{3c}\frac{n_{2c} - n_{1c}}{n_{2r} - n_{1r}}. \tag{B.7}$$

To select n_{3c} with the correct sign we check the angle between lines L_1 and L_3 and select $\begin{bmatrix} n_{3r} \\ n_{3c} \end{bmatrix}$ which makes this angle less than 90 degrees because the angle between L_1 and L_2 is always less than 180 degrees.

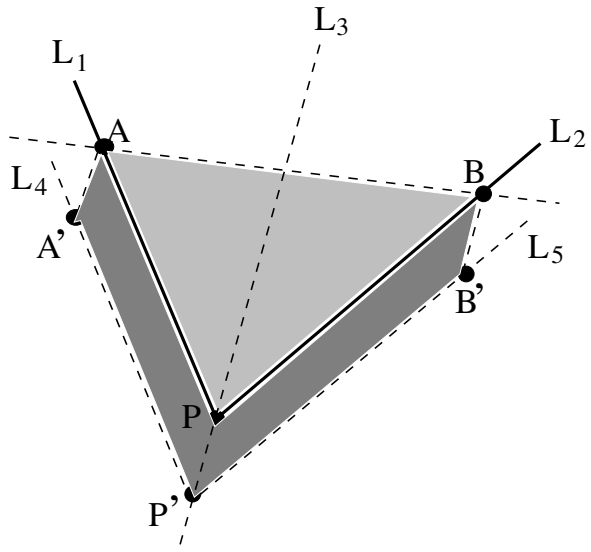


Figure B.1: Regions for mean calculation.

Then, we can find the point P', which lies on the line L_3 and is at a distance of λ' pixels to the point P, as

$$P' = P + \lambda'\hat{n}_3 \tag{B.8}$$

where λ' is a given negative constant. Now, we know the equations of the lines L_4

$$L_4 = P' + \lambda_4 \hat{n}_1 \tag{B.9}$$

and L_5

$$L_5 = P' + \lambda_5 \hat{n}_2. \tag{B.10}$$

We can find locations of the points A' and B' as

$$A' = P' + \|AP\| \hat{n}_1 \tag{B.11}$$

and

$$B' = P' + \|BP\| \hat{n}_2 \tag{B.12}$$

where points A and B are located away from P at a distance of 80 percent of the lengths of lines L_1 and L_2 respectively.

Given the points A, A', B, B', P and P', we can find whether a point $X = \begin{bmatrix} r \\ c \end{bmatrix}$ lies inside or outside the region spanned by the lines L_1 and L_2. We call the *in* region as the one bounded by the lines AP, PB, BA, and the *out* region as the one bounded by the lines AA', $A'P'$, $P'B'$, $B'B$, BP, PA.