Content-based Image Database Retrieval Using Variances of Gray Level Spatial Dependencies

Selim Aksoy and Robert M. Haralick

Intelligent Systems Laboratory
Department of Electrical Engineering
University of Washington
Seattle, WA 98195-2500
{aksoy,haralick}@@isl.ee.washington.edu
http://isl.ee.washington.edu

Abstract. In this paper, we discuss how we use variances of gray level spatial dependencies as textural features to retrieve images having some section in them that is like the user input image. Gray level co-occurrence matrices at five distances and four orientations are computed to measure texture which is defined as being specified by the statistical distribution of the spatial relationships of gray level properties. A likelihood ratio classifier and a nearest neighbor classifier are used to assign two images to the relevance class if they are similar and to the irrelevance class if they are not. A protocol that involves translating a $K \times K$ frame throughout every image to automatically construct groundtruth image pairs is proposed and performance of the algorithm is evaluated accordingly. From experiments on a database of 300 512×512 grayscale images with 9,600 groundtruth image pairs, we were able to estimate a lower bound of 80% correct classification rate of assigning sub-image pairs we were sure were relevant, to the relevance class. We also argue that some of the assignments which we counted as incorrect are not in fact incorrect.

1 Introduction

Large amount of images that are generated by various applications and the advances in computation power, storage devices, scanning, networking, image compression, desktop publishing, and the World Wide Web have made image databases increasingly popular. The advances in these areas contribute to an increase in the number, size, use, and availability of on-line image databases. 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, instead of searching only on the associated textual information.

In a typical content-based image database retrieval application, the user has an image he or she is interested in and wants to find similar images from the entire database. The image retrieval scenario we address 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 having some section in them that is like the user input image.

In content-based retrieval, the problem is first to find efficient features for image representation, then to use an effective measure to establish similarity between two images. The features and the similarity measure should be efficient enough to match similar images as well as being able to discriminate dissimilar ones. In this paper, we discuss how we use variances of gray level spatial dependencies as textural features to retrieve images from a database of grayscale images. Then, we propose a protocol to automatically construct groundtruth image pairs to evaluate the performance of the algorithm accordingly. Given these groundtruths, we find the best case and worst case classification efficiencies of the algorithm.

The paper is organized as follows. First, some of the previous approaches to texture and its use in content-based retrieval are discussed in Section 2. Second, we discuss our textural features in Section 3. Section 4 describes the decision methods for similarity measurement. Next, we present our experiments and results in Section 5. Finally, we discuss the conclusions and suggestions for future work in Section 6.

2   Background and Motivation

Texture has been one of the most important characteristics which have been used to classify and recognize objects and scenes. Texture can be characterized by the spatial distribution of gray levels in a neighborhood. 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.

In his texture survey, Haralick [6] 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 [13] 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.

Many researchers used texture in finding similarities between images in a database. In the QBIC Project, Niblack *et al.* [4] 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 extracted features based on coarseness, contrast,

and directionality. In the Photobook Project, Pentland *et al.* [12] used features based on appearance, 2-D shape and textural properties. For texture, they used 2-D Wold-based decompositions. In the CANDID Project, Kelly *et al.* [9] used Laws' texture energy maps to extract textural features and introduced a global signature based on a sum of weighted Gaussians to describe the texture. Manjunath and Ma [11] 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. Li *et al.* [10] used 21 different spatial features like gray level differences (mean, contrast, moments, directional derivatives, etc.), co-occurrence matrices, moments, autocorrelation functions, fractals and Robert's gradient on remote sensing images. Carson *et al.* [2] 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.

We define texture as being specified by the statistical distribution of the spatial relationships of gray level properties. Julesz [8] 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.

One of the early approaches that use spatial relationships of gray levels in texture discrimination is [5], 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 [7], Haralick *et al.* again used features computed from co-occurrence matrices to classify sandstone photomicrographs, panchromatic aerial photographs, and ERTS multispectral satellite images. Although they used only some of the features they defined and did not use the same classification algorithm in their tests for different data sets, 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.* [15] made a comparative study of four texture classification approaches; 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 [7] and concluded that features computed from co-occurrence matrices perform as well as or better than other algorithms.

Another comparative study is done by Conners and Harlow [3]. 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.

From the experiments on wide class of images, it can be concluded that spatial gray level dependencies carry much of the texture information [6] and they are more general and perform better than other methods [15, 3]. More information on this topic will be given in Section 3.1.

3 Feature Extraction

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 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 we describe the co-occurrence matrices and the features we compute from them.

3.1 Gray Level Co-Occurrence

Gray level co-occurrence can be specified in a matrix of relative frequencies $P(i, j; d, \theta)$ with which two neighboring pixels separated by distance d at orientation θ occur in the image, one with gray level i and the other with gray level j. For example, for a $0°$ angular relationship, $P(i, j; d, 0°)$ averages the probability of a left-right transition of gray level i to gray level j at a distance d.

In our derivations, we define the origin of the image 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 1, co-occurrence matrices can be

defined as

$$
\begin{aligned}
P(i,j;d,0°) = \#\{&((r,c),(r',c')) \in (L_r \times L_c) \times (L_r \times L_c)| \\
&r'-r=0, |c'-c|=d, I(r,c)=i, I(r',c')=j\} \\
P(i,j;d,45°) = \#\{&((r,c),(r',c')) \in (L_r \times L_c) \times (L_r \times L_c)| \\
&(r'-r=d, c'-c=d) \text{ or } (r'-r=-d, c'-c=-d), \\
&I(r,c)=i, I(r',c')=j\} \\
P(i,j;d,90°) = \#\{&((r,c),(r',c')) \in (L_r \times L_c) \times (L_r \times L_c)| \\
&|r'-r|=d, c'-c=0, I(r,c)=i, I(r',c')=j\} \\
P(i,j;d,135°) = \#\{&((r,c),(r',c')) \in (L_r \times L_c) \times (L_r \times L_c)| \\
&(r'-r=d, c'-c=-d) \text{ or } (r'-r=-d, c'-c=d), \\
&I(r,c)=i, I(r',c')=j\}.
\end{aligned}
\tag{1}
$$
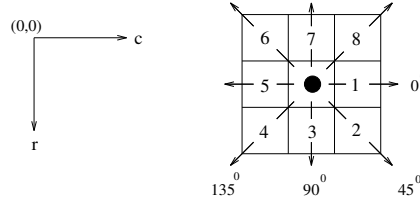
Fig. 1. Spatial arrangements of pixels

Resulting matrices are symmetric. The distance metric used in Equation (1) can be explicitly defined as

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

We can normalize these matrices by dividing each entry in a matrix by the number of neighboring pixels used in computing that matrix. Given distance d, 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.

3.2 Textural Features

In order to use the information contained in the gray level co-occurrence matrices, Haralick [7] defined 14 statistical measures which measure textural characteristics like homogeneity, contrast, organized structure, complexity, and nature of gray level transitions. Since from many distances and orientations we obtain 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

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

which is a difference moment of P that measures the contrast in the image. Rosenfeld and Troy [13] 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.

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. Zucker and Terzopoulos [16] 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. In [14], Tou and Chang used an eigenvector-based approach and Karhunen-Loeve expansion to eliminate dependent features. Currently we are developing methods to select the distances that perform the best according to some statistical measures. In this work we compute the variance feature for 1 to 5 pixel distances and four orientations. This constitutes a 20-dimensional feature vector.

Note that angularly dependent features are not invariant to rotation. 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 anymore. 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. 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 the image in our database than the orientation in our query image. This dilemma is also present in object-based queries. In this work, we will use the feature vector described above which is rotation variant. We are in the process of modifying our feature vector to include rotation invariance as discussed in [7] and are going to do experiments with the new feature vector on the same database.

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 using the protocol which will be discussed in Section 4.1. We then compute a 20-dimensional feature vector for each sub-image in the database.

4 Decision Methods

After computing the feature vectors for all images in the database, given a query, 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. In our experiments we use two different types of decision methods; a likelihood ratio approach which is a

Gaussian classifier, and a nearest neighbor rule based approach. In the following sections we discuss these two approaches.

4.1 Likelihood Ratio

In the likelihood ratio approach, we define two classes, namely the relevance class $\mathcal{A}$ and the irrelevance class $\mathcal{B}$. Given the feature vectors of 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 two sections we describe first, how to determine the parameters of the two classes, and second, how to construct the likelihood ratio.

Determining the Parameters The protocol for constructing groundtruths to determine the parameters of the likelihood ratio classifier involves making up two different sets of sub-images for each image i, $i = 1 \ldots I$, in the database. The first set of sub-images begins in row 0 column 0 and partitions each image i into M_i K $\times$ 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 K $\times$ K sub-images. This set of sub-images will be referred as the *main database* in the rest of the paper.

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 paper.

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 $n = 1 \ldots N_i$.

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 $\mathcal{A}_i$,

$$\mathcal{A}_i = \{(n, m) | \, m \in R_s(n) \, , \, n = 1 \ldots N_i\},$$

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

$$\mathcal{B}_i = \{(n, m) | \, m \in R_o(n) \, , \, n = 1 \ldots N_i\}$$

for each image i. Then, the overall relevance class becomes $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \cdots \cup \mathcal{A}_I$ and the overall irrelevance class becomes $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_I$.

An example for the overlapping concept is given in Figure 2 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 groundtruths for the relevance and irrelevance classes respectively. Note that for any sub-image which is not shifted by (K/4,K/4), there is a sub-image which it overlaps by more than half the area. We will use this property to evaluate the performance of the algorithm in Section 5.
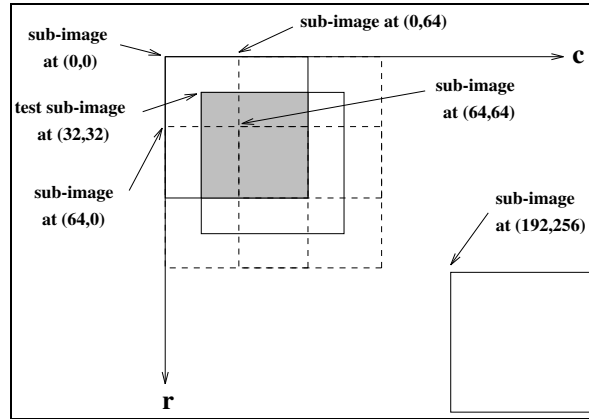
Fig. 2. The shaded region shows the 9/16 overlapping between two sub-images

As the database structure is concerned, our first sub-image database (*main database*) contains a unique sub-image i.d., bounding box, and the feature vector for each sub-image $m = 1 \dots M_i$ and $i = 1 \dots I$. The second sub-image database (*test database*) contains a unique sub-image i.d., bounding box, $R_s(n)$, $R_o(n)$, and the feature vector for each sub-image $n = 1 \dots N_i$ and $i = 1 \dots I$.

In order to estimate the distribution of the relevance class, we first compute the differences d, $d = x^{(n)} - y^{(m)}$, $(n,m) \in \mathcal{A}$, $x^{(n)}, y^{(m)} \in \mathbb{R}^Q$ where Q is 20 for our features, and $x^{(n)}$ and $y^{(m)}$ are the feature vectors of sub-images n and m respectively. Then, we compute the sample mean, $\mu_{\mathcal{A}}$, and the sample covariance, $\Sigma_{\mathcal{A}}$, of these differences. We assume that these differences for the relevance class have a normal distribution with mean $\mu_{\mathcal{A}}$, and covariance $\Sigma_{\mathcal{A}}$. Similarly, we compute the differences d, $d = x^{(n)} - y^{(m)}$, $(n,m) \in \mathcal{B}$, $x^{(n)}, y^{(m)} \in \mathbb{R}^Q$, then the sample mean, $\mu_{\mathcal{B}}$, and the sample covariance, $\Sigma_{\mathcal{B}}$, for the irrelevance class.

Making the Decision Suppose for the moment that the user query is a $K \times K$ image. First, its feature vector x is determined. Then, the search goes through all the feature vectors $y^{(m)}$ in the *main database* where $m = 1 \ldots (\sum_{i=1}^{I} M_i)$, M_i being the number of sub-images in the i'th image. For each feature vector pair $(x, y^{(m)})$, the difference $d = x - y^{(m)}$ is computed.

The probability that the input query image with feature vector x, and a sub-image in the database with feature vector $y^{(m)}$ are relevant is $P(\mathcal{A}|d) = P(d|\mathcal{A})P(\mathcal{A})/P(d)$ and that they are irrelevant is $P(\mathcal{B}|d) = P(d|\mathcal{B})P(\mathcal{B})/P(d)$. We can define the likelihood ratio as

$$r(d) = \frac{P(\mathcal{A}|d)}{P(\mathcal{B}|d)}. \tag{3}$$

If this ratio is greater than 1, the sub-image m is considered to be relevant to the input query image. If we assume two classes are equally likely, equation (3) becomes

$$\frac{P(d|\mathcal{A})}{P(d|\mathcal{B})} = \frac{P(d|\mu_\mathcal{A}, \Sigma_\mathcal{A})}{P(d|\mu_\mathcal{B}, \Sigma_\mathcal{B})}$$

$$= \frac{\frac{1}{(2\pi)^{Q/2}|\Sigma_\mathcal{A}|^{1/2}} e^{-(d-\mu_\mathcal{A})'\Sigma_\mathcal{A}^{-1}(d-\mu_\mathcal{A})/2}}{\frac{1}{(2\pi)^{Q/2}|\Sigma_\mathcal{B}|^{1/2}} e^{-(d-\mu_\mathcal{B})'\Sigma_\mathcal{B}^{-1}(d-\mu_\mathcal{B})/2}} \tag{4}$$

$$> 1.$$

After taking the natural logarithm of (4) we obtain

$$(d - \mu_\mathcal{A})'\Sigma_\mathcal{A}^{-1}(d - \mu_\mathcal{A})/2 < (d - \mu_\mathcal{B})'\Sigma_\mathcal{B}^{-1}(d - \mu_\mathcal{B})/2 + \ln \frac{|\Sigma_\mathcal{B}|^{1/2}}{|\Sigma_\mathcal{A}|^{1/2}}. \tag{5}$$

To find the sub-images that are relevant to an input query image, likelihood ratios for all sub-images in the database are computed as in (3) and the sub-images are ranked by these likelihood ratios. Among them, k sub-images having the highest r-values are retrieved as the most relevant ones.

4.2 Nearest Neighbor Rule

In the nearest neighbor approach we assume each sub-image m in the database is 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 by 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 $Y = \{y^{(m)}|y^{(m)} \in \mathbb{R}^Q, m = 1, \ldots, M\}$ and feature vector $x \in \mathbb{R}^Q$, find the set of sub-images $R \subseteq \{1, \ldots, M\}$ such that $\#R = k$ and

$$\rho(x, y^{(r)}) \le \rho(x, y^{(p)}), \quad \forall r \in R, p \in \{1, \ldots, M\}\backslash R$$

where $M = \sum_{i=1}^{I} M_i$, M_i being the number of sub-images in the i'th image.

For the distance metric ρ we use the Euclidean distance

$$\rho(x,y) = \|x - y\|$$

or the infinity norm

$$\rho(x,y) = \max_{i=1,\ldots,Q} |x_i - y_i|$$

where x_i and y_i are the i'th components of the corresponding feature vectors.

5   Experiments and Results

Testing content-based retrieval systems and comparing the performances of two different algorithms is an open question. Two traditional measures for retrieval performance are precision and recall. Precision is the percentage of retrieved images that are relevant and recall is the percentage of relevant images that are retrieved. Note that computation of these measures requires image-level goundtruthing of the database. We created two databases of sub-images according to the protocol in Section 4.1 but since these automatically generated sub-image-level groundtruths are not the ones required for precision and recall, we use modified versions of these measures to evaluate the performance of our algorithm. After manually grouping a smaller set of images in our database, we will evaluate the performance using precision and recall too.

In the following sections we describe the database population and two experimental procedures for our decision methods.

5.1   Database Population

To populate the database, we used the Fort Hood Data, supplied for the RADIUS program by the Digital Mapping Laboratory at Carnegie Mellon University. These images consist of visible light images of the Fort Hood area at Texas. We divided these aerial images into 300 512×512 images. After the database was constructed, we carried out the approach described in Section 4.1 which involved translating a 256×256 frame throughout every image and extracted the desired features for all sub-images.

5.2   Experimental Set-up

To test the classification effectiveness using the Gaussian classifier, we can apply the classification algorithm to each groundtruth pair (n,m) described in Section 4.1. Since we know which non-shifted sub-images and shifted sub-images overlap, we also know which sub-image pairs should be assigned to class $\mathcal{A}$ and which to class $\mathcal{B}$. So, to test our approach, we then 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.

To test the retrieval performance of the algorithm, we use the following procedure. Given an input query image of size K $\times$ K, we create a list of retrieved

images in descending order of likelihood ratio or ascending order of distance for nearest neighbor rule. If the correct image is retrieved as one of the k best matches, it is considered a success. This can also be stated as a nearest neighbor classification problem where the relevance class is defined to be the best k matches and the irrelevance class is the rest of the images. We also compute the average rank of the correct image among retrieved images. For this experiment, we use the non-shifted sub-images to compute the best case performance and the shifted sub-images to compute the worst case performance. We call this the worst case performance because the shifted sub-images overlap by approximately half the area of a sub-image in the database. All other possible sub-images have a sub-image in the database which they overlap by more than half the area. This experimental procedure is appropriate to our problem of retrieving images which have some section in them that is like the user input image.

5.3 Results

Classification Effectiveness In this experiment, the *main database* consists of 2,700 256 × 256 sub-images and the *test database* consists of 1,200 256 × 256 sub-images. There are 4 relevant and 4 irrelevant non-shifted sub-images for each of the 1,200 shifted sub-images, which make a total of 9,600 groundtruth sub-image pairs. As can be seen in Table 1, 79.75% of the groundtruth $\mathcal{A}$ pairs were assigned to $\mathcal{A}$ with an overall success of 62.96%.

Table 1. Confusion matrix for the classification effectiveness test.

	Assigned Relevant	Assigned Irrelevant	Success (%)
Relevant Pair G.truth	3,828	972	79.75
Irrelevant Pair G.truth	2,584	2,216	46.17
Overall	6,412	3,188	62.56

We can say that most of the groundtruth $\mathcal{A}$ pairs were assigned to $\mathcal{A}$ but the groundtruth $\mathcal{B}$ pairs seem to be split 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 can not always guarantee that non-overlapping sub-images are irrelevant. Obvious examples are images which have the same texture pattern at more than one location. Illustration of this fact can be found in [1] 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. Hence, some of the assignments which we count as incorrect are not in fact incorrect. Thus the approximate 80% correct relevant pair rate is a lower bound.

Retrieval Performance Results for the retrieval performance experiments are summarized in Table 2 as the number of tests, number of successes, and average rank of the correct image. For the best case analysis 2,700 sub-image queries were used. As explained before, these are the non-shifted sub-images in the database. For the worst case analysis, 1,200 shifted sub-images in the *test database* are used. To illustrate the bounds found in these experiments, the database was queried with 500 randomly extracted 256×256 sections from images in the database. In all of these experiments a success means the correct image is retrieved as one of the best 20 matches.

Table 2. Results for the retrieval performance test.

	Original sub-images (no of tests = 2,700)				Shifted sub-images (no of tests = 1,200)		
	Likelihood Ratio	Euclidean Distance	Infinity Norm		Likelihood Ratio	Euclidean Distance	Infinity Norm
# successes	2,536	2,683	2,684	# successes	683	701	681
% success	93.93	99.37	99.41	% success	56.92	58.42	56.75
avg. rank	4.0430	2.0078	2.0138	avg. rank	6.1830	5.5706	5.5727

	Random sub-images (no of tests = 500)		
	Likelihood Ratio	Euclidean Distance	Infinity Norm
# successes	326	330	325
% success	65.20	66.00	65.00
avg. rank	5.7301	4.2576	4.2523

As can be seen in Table 2, the algorithm successfully retrieved the correct image as one of the 20 best matches 56 percent of the time at the worst case. Euclidean distance performed slightly better than the infinity norm. Although the worst case and random query results of both likelihood ratio and nearest neighbor decision methods were almost equal, nearest neighbor method performed slightly better in the best case analysis. Also the nearest neighbor rule retrieved the correct image at a higher rank than the likelihood ratio which is 2 at the best case and 5 at the worst case on the average.

Experimenting on sub-image size showed that smaller sub-images give better results because co-occurrence features are measures of micro texture and texture tends to be more homogeneous as sub-image size gets smaller.

Some example queries are given in Figures 3(a)-3(f). In all of these figures the upper-left image is the 256×256 query image. First three rows show the best 12 matches among the 512×512 images in the database. Last row shows 4 images that are found to be the most irrelevant to the query image. Our

system also displays 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 query in a more effective way. In each retrieved image, matched 256×256 sub-images are marked with a white border. More examples can be found at http://isl.ee.washington.edu/~aksoy/research/database.shtml.

6 Conclusions

In this paper, 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 in them that is like the user input image.

To achieve this goal, texture was defined as being specified by the statistical distribution of the spatial relationships of gray level properties and variances computed from two-dimensional gray level co-occurrence matrices at 1 to 5 pixel distances and four orientations were used to extract this information.

A likelihood ratio classifier was defined to measure the relevancy of two images, one being the query image and one being a database image, 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. Also k-nearest neighbor rule was used to retrieve k images which have the closest feature vector to the feature vector of the query image in the 20-dimensional feature space.

Testing content-based retrieval systems and comparing the performance of two different algorithms is an open question. A protocol which involved translating a K×K frame throughout every image to automatically construct groundtruth image pairs for the relevance and irrelevance classes was proposed and performance of the algorithm was evaluated accordingly.

Experiments were done on a database of 300 images to check the effectiveness of the features in representing images. Results of the classification effectiveness tests showed that the algorithm assigned 79.75% of the sub-image pairs we were sure were relevant, to the relevance class correctly when the database was partitioned into 9,600 256×256 sub-image pairs even with an offset of quarter of the image size which was 64 pixels in the tests. Results of the retrieval performance tests showed that all of the decision methods retrieved correct images successfully as one of the best 20 matches, which is less than 1 percent of the total, in more than 93 percent of the 2,700 experiments for the best case analysis and in more than 56 percent of the 1,200 experiments for the worst case analysis.

An interesting study will be to examine images that are successfully retrieved with the nearest neighbor rule but missed with the likelihood ratio, and vice versa. Although being a micro texture measure, our features showed significant performance on a database of complex aerial images. We are currently adding more features that will capture the texture information at higher scales [1]. This will result in a more compact representation that is needed for large databases containing different types of complex images.
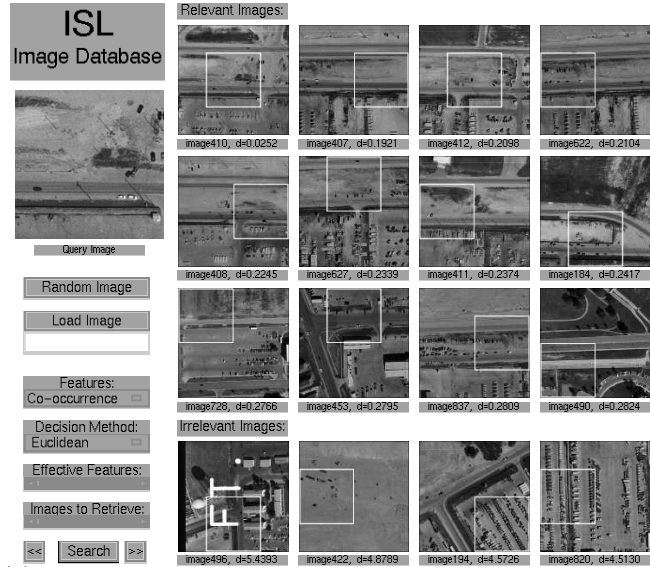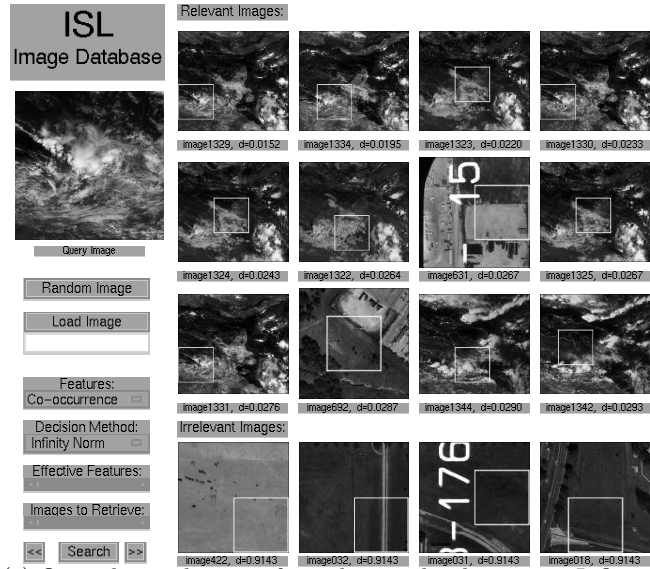
(a) Query by a sub-image from the main database using Euclidean distance.
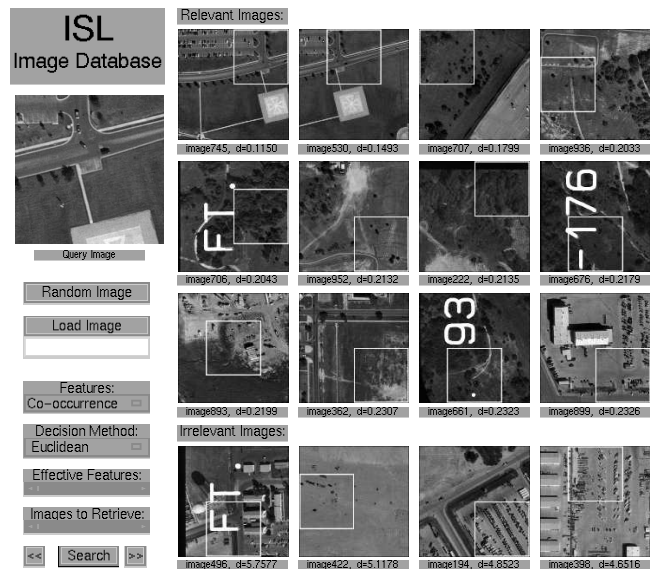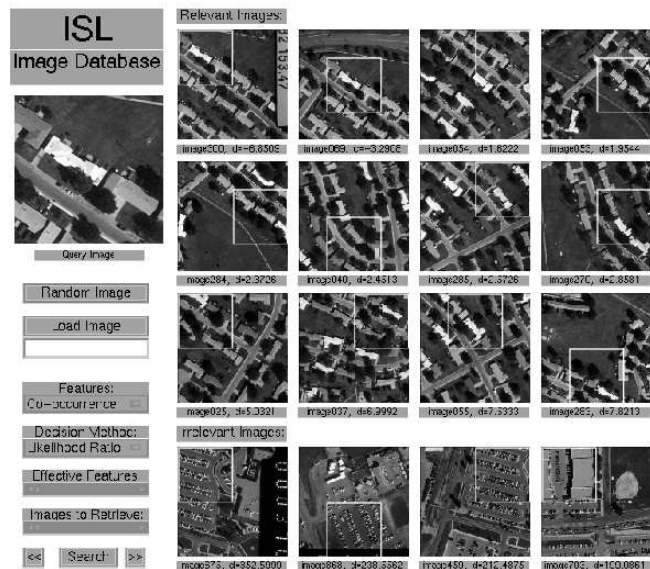

(b) Query by a sub-image from the test database using Euclidean distance.

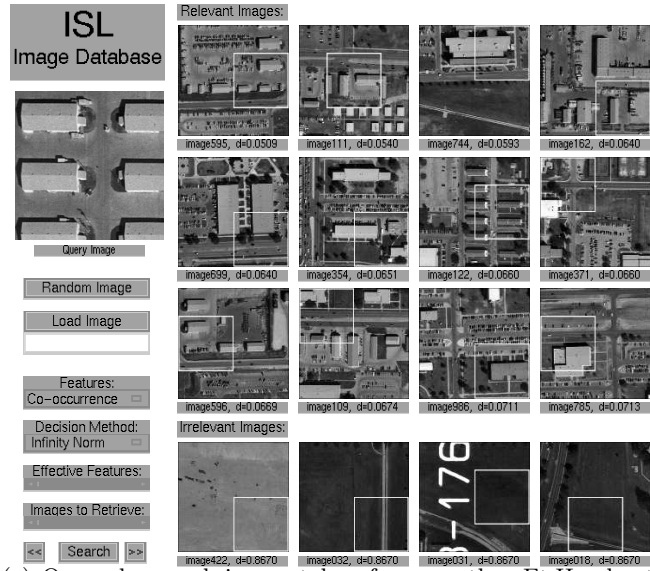**Fig. 3.** Example queries using different distance methods

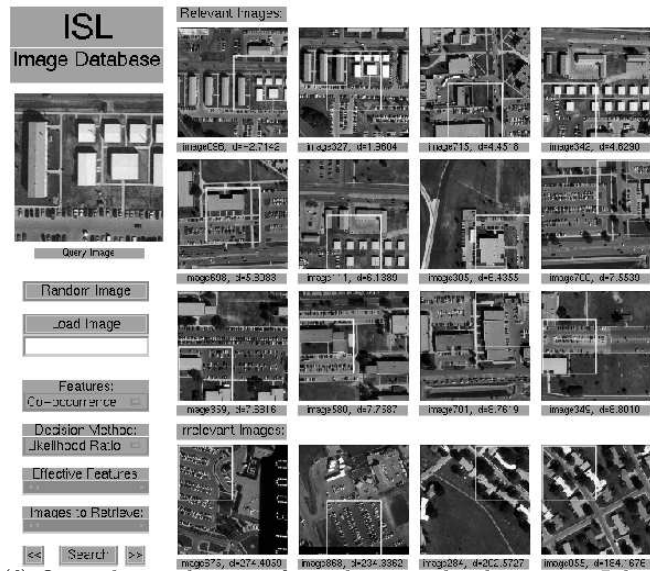(c) Query by a sub-image from the test database using Infinity norm.

(d) Query by a sub-image from the main database using Likelihood ratio.

Fig. 3. Example queries using different distance methods (cont.)

(e) Query by a sub-image taken from another Ft.Hood set using Infinity norm.

(f) Query by a sub-image from the main database using Likelihood ratio.

Fig. 3. Example queries using different distance methods (cont.)

References

1. 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.

2. 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.

3. 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.

4. 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.

5. 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.

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

7. 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.

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

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.