Edge-suppressed color image indexing and retrieval using angle-distance measurement in the scaled-space of principal components

A Thesis Presented to

The Faculty of the

Fritz J. and Dolores H. Russ
College of Engineering and Technology
Ohio University
In Partial Fulfillment
Of the Requirement for the Degree
Master of Science

by
Sergei Bobik

June, 2000
THIS THESIS ENTITLED

"EDGE-SUPPRESSED COLOR IMAGE INDEXING AND RETRIEVAL USING ANGLE-DISTANCE MEASUREMENT IN THE SCALED-SPACE OF PRINCIPAL COMPONENTS"

by Sergei Bobik

has been approved

for the School of Electrical Engineering and Computer Science

and the Russ College of Engineering and Technology

Mehmet Celenk, Ph. D.
Associate Professor of Electrical Engineering

Warren K. Wray, Dean
Fritz J. and Dolores H. Russ College of Engineering and Technology
# Table of Contents

Chapter 1  INTRODUCTION  
1.1 Image retrieval model  
1.2 Past work  

Chapter 2  PRINCIPAL COMPONENT ANALYSIS  
2.1 Applications of principal component analysis  
2.2 Principal component analysis of color images  

Chapter 3  EDGE DETECTION  

Chapter 4  EDGE-SUPPRESSED COLOR IMAGE INDEXING AND RETRIEVAL USING ANGLE-DISTANCE MEASUREMENT IN THE SCALED EIGEN-SPACE  
4.1 Edge detection and suppression in color images  
4.2 Principal component representation of edge-suppressed color images  
4.3 Rotation, translation, and scaling (RTS) invariance property of principal component representation  
4.4 Similarity measure in the RGB space  

Chapter 5  COMPUTER IMPLEMENTATION AND EXPERIMENTAL RESULTS  
5.1 RTS invariance testing  
5.2 The search engine approach  
5.3 Comparative study of approaches for color image indexing  

Chapter 6  CONCLUSIONS  

References  
Appendices
List of Tables

Table I  Retrieval efficiency vs. threshold values for the new similarity measure

Table II  Retrieval efficiency vs. list length values for different retrieval methods
List of Figures

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Image archival and retrieval system (from [9]).</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>An indexing and retrieval system model with user feedback [17].</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>HSV cone depicting black, white, bright chromatic, and achromatic regions [13].</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>Ideal vs. real edge.</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>Erroneous outputs produced by an edge operator.</td>
<td>21</td>
</tr>
<tr>
<td>3.3</td>
<td>(a) One-dimensional step edge, (b) its first derivative, and (c) its second derivative.</td>
<td>22</td>
</tr>
<tr>
<td>3.4</td>
<td>Cross-section of continuous domain Laplacian of Gaussian impulse response.</td>
<td>23</td>
</tr>
<tr>
<td>3.5</td>
<td>An 11x11 Laplacian of Gaussian kernel for $\sigma=1.4$.</td>
<td>24</td>
</tr>
<tr>
<td>3.6</td>
<td>Results of the Laplacian of Gaussian edge detector for different threshold values $t$ ($\sigma=2$). Original image is of size 159x153 with 24 bpp color specification.</td>
<td>25</td>
</tr>
<tr>
<td>3.7</td>
<td>Results of the Laplacian of Gaussian edge detector for different values of $\sigma$ ($t=0.5$). Original image is of size 159x153 with 24 bpp color specification.</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>Cluster separation by edge removal.</td>
<td>28</td>
</tr>
<tr>
<td>4.2</td>
<td>Effect of edge suppression on the image color distribution.</td>
<td>28</td>
</tr>
<tr>
<td>4.3</td>
<td>Similarity measurement between two 3-D orthogonal sub-spaces in RGB system.</td>
<td>32</td>
</tr>
<tr>
<td>5.1</td>
<td>Retrieval efficiency vs. list length values for different retrieval methods. (Legend: - : new method; ... : distance method; ◊ : color reference table method; * : histogram intersection; ▲ : Tanimoto measure; ◐ : relative entropy)</td>
<td>40</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

Continuous growth of the Internet during the past years has created a vast variety of information sources each containing a broad spectrum of data types such as text, graphics, still images, movie, video, and music clips. One of the biggest challenges in the manipulation of these multimedia data is to index their contents for search and retrieval. In order to be useful, data must be easily accessible by means of some keys or a set of features, depending on the type of information that we want to obtain. Text files or documents are the easiest to deal with since the search is based on simple word matching. This kind of search procedure is particularly useful for the search engines available on the Internet such as Yahoo©, InfoSeek©, and AltaVista©. More challenging problems arise when there is a need for a search of images or movie clips matching certain criteria; e.g., color content, shape, texture information, or all of them combined. When we search for a certain sentence or a list of keywords we expect the system to return a set of documents that contain as much of the provided words as possible. In this case, all that is required from a computer system is basic bit matching. The difficulties with image searching become obvious when we start to think about possible ways of describing an image content to a computer system. As far as the processor, an image is simply a collection of data which has no meaning whatsoever unless it is properly interpreted. Currently, the
main goal and hope of the researchers working in this direction is to find the best mechanism that will allow a computer system to “understand” image content to a certain degree that will be enough to give good results when asked to do an image search. Since the actual image understanding by a computer system is yet to be achieved, simpler ways of performing image searches have emerged. These are based on the concept of similarity between two entities rather than perfect matching, which is a valid practice since human beings tend to describe images in general terms as well. We would never say that the sky in a picture occupies a rectangular region defined by points (0,0) and (50,200) with color (10,20,255). This suggests that we should follow the same direction by using more general, high-level and abstract descriptions of images instead of raw bit matching.

Although a variety of methods have been proposed to deal with this problem [2, 5-7, 9, 16], none of the proposed algorithms are sufficiently general. Each method approaches the problem from a different perspective with a certain degree of success. We can distinguish three major ways of generating image features for indexing and retrieval. These are based on the use of shape information [2], color distribution [4, 5, 9], or a combination of both.

In this research, we propose an edge-suppressed indexing and retrieval method for color images based on the scaled-space principal component analysis (PCA). The proposed method differs from the existing approaches (e.g., histogram intersection, distance method, reference color table method, and others [4-7,9,14-15]) in the sense that the low and high frequency contents of an image are separated before
the characteristic value analysis. This helps determining the high and low activity areas in the spatial plane of a given image so that the PCA analysis can be directed toward the spectrally rich and descriptive portions of object, background, and/or foreground regions of interest.

1.1 Image retrieval model

Figure 1.1 shows a block diagram of a generic image archival and retrieval model [9]. Image matching and retrieval is based on some set of characteristic values or features, characterizing a given image or a class of images.

![Image archival and retrieval system diagram](from [9]).

In the archival phase, image database is formed in the following way. A continuous input image or picture is scanned into an uncompressed digital color image. Image analysis is then performed to extract information that can be used as features such as shape orientation, texture type, and color distribution. Raw color
data is then compressed using one of the well-known algorithms (e.g., GIF, JPEG [12, 17], etc.) and stored in the database along with the computed features.

Image retrieval process begins by submitting a query image to the search engine. The engine extracts the needed features from the input image for future use in matching. Matching is performed image-by-image using the computed features and some kind of similarity measure. A set of closely matching images is brought out as the result of search output.

An important criterion for testing the efficacy of the search and retrieval is that the output must include all the similar images. The list may have other images as well but that is not very important. The important thing is that the similar ones should not be missed in the search process. This is due to the fact that automated image retrieval systems usually have a human operator who makes the final decision. Figure 1.2 [17] gives an indexing and retrieval system model with user feedback for interactive operation of the generic block diagram in Fig. 1.1. In this case, there is a feedback loop that enables the user to refine his/her search query. Notice also the ability to browse the list of the retrieved images. This browsing may entail several goals. One of them is to give user a better interface to the system, which simplifies the whole querying process. The second one, which is possibly the most important, is to use this browsing to improve system’s retrieval capabilities. Feeding the user’s choices back to the search engine allows the system to fine-tune itself and provide better services the next time the same query is executed.
Fig. 1.2 An indexing and retrieval system model with user feedback [17].

1.2 Past work

Recently there has been an increased interest in research on color-based retrieval. Swain and Ballard have proposed a color matching method called histogram intersection [4]. It is based on matching of color histograms and the main idea in their algorithm is to compute

\[
H(I,M) = \frac{\sum_{j=1}^{n} \min(I_j,M_j)}{\sum_{j=1}^{n} M_j}
\] (1.1)
where \( H(I,M) \) is the match value and \( I \) and \( M \) are the query image and model (an image in the database) histograms, respectively, each containing \( n \) bins. The match value \( H(I,M) \) is computed for every model histogram in the database. A model image is selected as similar to the query image if the corresponding \( H(I,M) \) is closer to 1. In this approach, the feature vector \( f \) used to represent the color information of an image is the 3-D color histogram \( h(x, y, z) \) and the similarity measure is given by equation (1.1). The major drawback of this method is the extensive numerical computation. The computational complexity is given by \( O(nm) \), where \( m \) is the total number of model images in the database and \( n \) is the histogram size. For a typical 24 bits per pixel color image, since the histogram size is 16 megabytes and \( m \) is fixed for a given application, complexity can be reduced by under-sampling of the color space.

In the distance method, the feature used for capturing the color information is the mean value, \( \mu \), of the 1-D histograms (normalized by the total number of pixels) of each of the three color components of the image [5]. The feature vector \( f \) for identifying an RGB image is given by

\[
f = (\mu_R, \mu_G, \mu_B)
\]  

(1.2)

Two distance measures are defined for similarity matching as follows:

\[
D^M_{q,i} = |f_q - f_i| = \sum_{R,G,B} |\mu_q - \mu_i|
\]  

(1.3)

\[
D^E_{q,i} = \sqrt{|f_q - f_i|^2} = \sqrt{\sum_{R,G,B} (\mu_q - \mu_i)^2}
\]  

(1.4)

where \( D^M_{q,i} \) is the Manhattan distance and \( D^E_{q,i} \) is the Euclidian distance between the query image and a database image. Here, \( f_q \) is the color feature vector of the query
image and \( f_i \) is the color feature vector of the database image. Although this method is fast, it may sometimes give inappropriate results since smaller (larger) distances in the RGB space do not always correspond to perceptual similarities (differences) between corresponding images.

In the reference color table method, a number of reference colors is defined by selecting all the perceived color samples in an application. Each image pixel undergoes a color reassignment process in conjunction with the reference table. Each color is assigned the closest possible color from the table. This implies that if the selected color table is rich, then, the resultant image will perceptually be close to the original. Notice, however, that the considerable computation is needed to go through each image pixel and assign a new color to it. Therefore, the color feature \( f \) for this method is computed on the reduced color histogram based on the colors from the reference table. It is given by

\[
f = (\lambda_1, \lambda_2, \ldots, \lambda_n)
\]

where \( \lambda_i \) is the relative pixel frequency (with respect to the total number of pixels) for the \( i \)th reference table color in the image and \( n \) is the size of the reference color table. A weighted Euclidian distance measure is used to compare two feature vectors in this method [5]. Since the retrieval accuracy of the reference table method is very high, it may serve as an etalon for other newly developed methods.

Cluster-based color matching method [9] is oriented towards artificially created images such as trademarks, logos, or natural images with only a few dominant colors. The chroma of synthesized images shows continuity in the color space. Since
the number of objects in an image is usually limited, the color space is populated with clusters each corresponding to a certain object in the scene. Each cluster is defined by

\[ C_i = (R_i, G_i, B_i, \lambda_i) \]  \hspace{1cm} (1.6)

where \( i = 1,2,\ldots,m \). In this formulation, \((R_i, G_i, B_i)\) is the representative color mean vector of the cluster and \( \lambda_i \) is the fraction of the pixels in that cluster compared with the total number of pixels. The feature vector is then defined as

\[ f = (C_i | i = 1,2,3,\ldots,m) \]  \hspace{1cm} (1.7)

where \( m \) is the total number of clusters. Since it is not necessarily true that two images have the same number of colors, algorithm first computes permutation function which maps clusters in the query image to the closest clusters in the model image. After this step, the similarity value may be calculated by using the Euclidian distance or another method. Although not tested by the authors, degraded performance may be expected when dealing with real world images rather than synthetic ones. Otherwise, this method gives highly accurate results.

Another approach uses hue-saturation-value (HSV) color system to represent image data for clustering [13]. HSV space is defined by converting RGB values according to the following equations:

\[ H_1 = \cos^{-1}\left(\frac{1}{2}\left[\frac{(R-G)+(-B)}{\sqrt{(R-G)^2+(-B)(G-B)}}\right]\right) \]  \hspace{1cm} (1.8)

where \( H = H_1 \) if \( B \leq G \); otherwise \( H = 360 - H_1 \).
\[ S = \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)} \quad (1.9) \]

\[ V = \frac{\max(R,G,B)}{255} \quad (1.10) \]

In this conversion, R, G, and B are the red, green, and blue component values in the range [0, 255]. The first step in this method is to build a hue histogram for all the bright chromatic pixels. It has been found experimentally that these tend to be colors that have value >75% and saturation ≥20%. Once the pixels satisfying this criterion are identified, the hue histogram is built and thresholded into \( m \) bright colors, where \( m \) is an image-dependent quantity determined by the number of peaks in the hue histogram. From the remaining image pixels, saturation and value components are used to determine which regions of the image are achromatic. Specifically (see Fig. 1.3), it has been found

Fig. 1.3 HSV cone depicting black, white, bright chromatic, and achromatic regions.
experimentally [13] that colors with value <25% can be classified as black, and that colors with saturation <20% and value >75% can be classified as white. Although the remaining pixels fall in the chromatic region, there may be a wide range of saturation values. Multi-modal saturation histogram is thresholded and for each of the $p$ peaks that it exhibits the hue histogram is found. The resulting hue histograms are thresholded to obtain $n_p$ colors. Thus for each image, the algorithm extracts $c$ distinct colors:

$$c = \sum_p n_p + m$$

(1.11)

It is important to note that although clustering is performed in the HSV space, the retrieval process is done by using distance measures in the RGB domain. A similarity measure composed of an angle and magnitude component is used for retrieval. This is similar to the one that will be introduced in this research (Eq. (4.6)) in the sense that we use a combined angle-distance measure to compare the discrete Karhunen-Loeve (or PCA) based feature vectors of images. The main difference is in the angle measurement part. We use three angles between three eigenvectors of a color distribution, and each of them is scaled according to the formula given by equation (4.7). Moreover, in our case angles are not limited to the first quadrant of the Cartesian space, which, supposedly, gives more flexibility and power to the measure. The measure used by the authors is given by

$$\delta(x_i, x_j) = 1 - \left[1 - \frac{2}{\pi} \cos^{-1} \left( \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|} \right) \right] [1 - \frac{|x_i - x_j|}{\sqrt{3(255)^2}}]$$

(1.12)
where $x_i$ and $x_j$ are three-dimensional color mean vectors. Notice that RGB vectors are limited to the first quadrant of the Cartesian space and a normalization factor of $\frac{2}{\pi}$ is required. Another factor $\sqrt{3(255)^2}$ is needed to normalize distances in the RGB space. As a result, $\delta(x_i, x_j)$ takes on values in the $[0,1]$ range. A number of other distance measures has been explored as well such as the angular distance, including the generalized Minkowski distance and the Canberra distance [13].

During the query process, for each user-specified query color, a distance is calculated using (1.12) for each representative vector in a given database index. For multiple color queries, a distance to each representative vector is calculated for each query color. The minimum of these distances is taken and a multidimensional query distance vector $D$ is defined as

$$D(d_1, \ldots, d_n) = (\min(\delta(q_1, i_1), \ldots, \delta(q_1, i_c)), \ldots, \delta(q_n, i_1), \ldots, \delta(q_n, i_c))) \quad (1.13)$$

where $q_n$ are the $n^{th}$ query colors and $i_c$ are the $c$ indexed representative colors of each database image. Notice that the computation time grows as the number of representative or query colors increases. This problem is one of the current research directions being pursued by the authors. The method was tested by submitting one or two query colors in the (R, G, B) format (e.g., (15, 90, 255)) to the search engine. Although the resulting outputs are in strong correlation with the desired colors, a real world image query would have been more appropriate to test the search engine performance.
The methods presented here tackle the color-based indexing problem with a certain degree of success by observing it from different viewpoints. Histogram intersection method is one of the first methods developed to address this problem. It requires immense computational and storage resource without proper modification. Distance method is simple and fast, producing reasonably good retrieval results at low computational and storage cost. Color reference table is the most precise method developed based on histograms. It gives highly accurate results demanding, however, an application specific color table. Lack of generality is the most unattractive feature of this method. Clustering techniques and algorithms based on analysis of color distributions in various color spaces are the most promising directions of research. Most of the developed methods are superior or comparable to the color reference table methods and other histogram based approaches [9, 13]. In this respect, we propose a new edge-suppressed indexing and retrieval method for color images based on the scaled-space principal component analysis (PCA). The low and high frequency contents of an image are separated before the characteristic value analysis. This helps determining the high and low activity areas in the spatial plane of a given image so that the PCA analysis, which gives the best approximation for the three principal directions through which the image color distribution is governed, can be directed toward the spectrally rich and descriptive portions of object, background, and/or foreground regions of interest. Therefore, it is expected that the PCA combined with powerful distance-angle measurements provides a unique tool for color image indexing and retrieval.
The rest of this thesis is organized as follows. Chapter 2 introduces concepts of the principal component analysis (PCA). Chapter 3 discusses the Laplacian of Gaussian edge detector, focusing also on common problems of edge operators. Chapter 4 presents the new edge-suppressed color image indexing and retrieval method. Chapter 5 includes details of computer implementation, experimental results, and the comparative study of different methods. Conclusions and further research topics are provided at the end.
Chapter 2

PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) is a well-known method, which has been widely used in statistical data analysis [11]. Suppose we have an n-dimensional random vector population \( \mathbf{x} \), given by

\[
\mathbf{x} = (x_1, \ldots, x_n)^T
\]  

(2.1)

where \( T \) denotes the matrix transposition. Mean of this population is defined by

\[
\mathbf{m}_x = \mathbb{E}\{\mathbf{x}\}
\]  

(2.2)

and its covariance matrix is computed as

\[
C_x = \mathbb{E}\{ (\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^T \}
\]  

(2.3)

Here, \( \mathbb{E}\{\} \) is the expected value operator. The components of \( C_x \), denoted by \( c_{ij} \), represent the covariance between the random variable components \( x_i \) and \( x_j \). The component \( c_{ii} \) is the variance of \( x_i \). The variance of a component indicates the spread of the component around its mean value. If two components \( x_i \) and \( x_j \) are uncorrelated, their covariance is zero (\( c_{ij} = c_{ji} = 0 \)). The covariance matrix is, by definition, always symmetric.

From a sample of vectors \( \mathbf{x}_1, \ldots, \mathbf{x}_n \), we can calculate the sample mean and the sample covariance matrix as the estimates of the mean and the covariance matrix as follows:
From the covariance matrix, we can calculate an \( n \times n \) orthogonal basis by finding its eigenvalues and eigenvectors. The eigenvectors \( e_i \) and the corresponding eigenvalues \( \lambda_i \) are the solutions of the matrix equation

\[
C_x e_i = \lambda_i e_i, \quad i = 1, \ldots, n
\]  

(2.6)

This leads to the characteristic equation of the eigenvalues in the form

\[
|C_x - \lambda I| = 0
\]  

(2.7)

where \( I \) is the identity matrix having the same order as \( C_x \) and the \( |\cdot| \) denotes the determinant of a matrix [11]. If the data vector has \( n \) components, the characteristic equation becomes of order \( n \). This is easy to solve only if \( n \) is small. For large values of \( n \), solving eigenvalues and corresponding eigenvectors is a non-trivial task. By ordering the eigenvectors in the order of descending eigenvalues, we can create an ordered orthogonal basis with the first eigenvector having the direction of the largest variance or entropy of the data. In this way, we can find directions in which the data set has the most energy.

Suppose we have a data set of which the sample mean and the covariance matrix have been calculated. Let \( A \) be a matrix having the normalized eigenvectors of \( C_x \) as its columns. By transforming the data vector \( x \) using \( A \), we get

\[
y = A^T (x - m_x)
\]  

(2.8)
which is a vector in the orthogonal coordinate system defined by the eigenvectors of $C_x$. Components of $y$ are the coordinates in the orthogonal base. We can reconstruct the original vector $x$ from $y$ by

$$x = Ay + m_x \quad (2.9)$$

using the property of the orthonormal matrix that $A^{-1} = A^T$. Here, the $A^T$ is the transpose of matrix $A$ and $A^{-1}$ is its inverse.

Instead of using all the eigenvectors of the covariance matrix, we can represent the data in terms of only a few basis vectors of the orthogonal set. If we denote the matrix having only $K$ eigenvectors corresponding to $K$ largest eigenvalues as $A_K$, the transformation equation (2.8) can be written as

$$y' = A_K^T (x - m_x) \quad (2.10)$$

and

$$x' = A_K y' + m_x \quad (2.11)$$

This means that we project the original data vector onto the coordinate axes having the dimension $K$ and transforming the vector back by a linear combination of basis vectors. The mean-square error (MSE) between the original data $x$ and the approximated representation $x'$ is defined as follows:

$$E \left[ \| x - x' \|^2 \right] = E \left[ \sum_{i=k}^{N-1} y(i) a_i \|^2 \right] = \sum_{i=m}^{N-1} \lambda_i \quad (2.12)$$

where $a_i$ are the orthonormal eigenvectors and $N$ is the total number of eigenvectors [3, 8]. As can be concluded from equation (2.12), the MSE is the sum of the eigenvalues corresponding to eigenvectors that are not used in the transformation.
2.1 Applications of principal component analysis

The concept of principal eigenvectors subspace has been exploited as a classifier [3]. First, the sample mean of the whole training set is subtracted from the feature vectors. For each class, \( \omega_i \), the correlation matrix \( R_i \) is estimated and the principal \( m \) eigenvectors (corresponding to the \( m \) largest eigenvalues) are computed. A matrix \( A_i \) is then formed using the respective eigenvectors as columns. An unknown feature vector \( x \) is then classified in the class \( \omega_j \) for which

\[
|| A_i^T x || > || A_j^T x ||, \quad \forall i \neq j \tag{2.13}
\]

In other words, the best classification corresponds to the maximum norm subspace projection of \( x \). This method can be directly applied to the color image indexing and retrieval task. In this case, input vector \( x \) would represent the color information presented by the image and the feature vector would contain only eigenvector information. The best match in the search process would then be picked based on the largest value of the norm found in the whole database. One of the drawbacks of this method may be the large number of matrix operations involved in the matching process since each image has to be projected onto some orthonormal basis. This approach has not been addressed neither in this research nor in any others in the literature. It may be one of the future research topics.

Another important use of the principal component analysis is the dimensionality reduction of feature vectors. The basic concept of dimensionality reduction is to transform a given set of measurements to a new set of features [3]. If
the transform is suitably chosen, transform domain features may exhibit high "information packing" properties compared with the original input samples. This means that the most of the classification-related information is "squeezed" in a relatively small number of features, leading to a reduction of the necessary feature space dimension.

The basic reasoning behind transform-based features is that an appropriately chosen transform can exploit and remove information redundancies, which usually exist in the set of samples obtained by the measuring devices. Specifically, if the PCA is used on a set of input vectors which are then transformed by using equation (2.8) to a new orthogonal space, then, only values resulting from the projection of $x$ onto the eigenvectors corresponding to the largest eigenvalues can be used to represent the original data set. According to [3], this will introduce the minimum possible amount of error in the mean-square error sense. The value of this error may be computed using equation (2.12). A good example of how dimensionality reduction can be done is given in [2].

Although simple, PCA possesses an extremely important property of producing mutually uncorrelated features by transforming the data into a new orthogonal subspace. This property is highly valuable in pattern recognition applications. Since matrix A is computed on the data provided by the correlation matrix, the transform is highly input dependent and does not have a unique kernel as in the case of many transforms (e.g., Fourier transform, Cosine transform, etc.). For high-dimensional input vectors, computation of the matrix A may become a
bottleneck because of the need to solve a large number of linear equations (see Eq. (2.6)).

2.2 Principal component analysis of color images

An NxM color image consists of three components corresponding to the red, green, and blue color planes, which can be viewed as random variables with (NM) samples. Therefore, equations (2.4)-(2.7) can be applied directly in order to obtain a set of orthogonal eigenvectors e_i, eigenvalues λ_i, and the mean vector m_x, where i=1,2,3.

In this research, we explore the ability of PCA to produce mutually orthogonal eigenvectors together with a set of eigenvalues which provide valuable information about the relative significance of each of the eigenvectors. Specifically, for a given color image with three random components (R, G, and B) we center the orthogonal eigen-sub-space at the mean vector of a given color distribution and then use a new similarity measure, given by equation (4.6) for matching. This concept will be discussed in detail in Chapter 4.
Chapter 3

EDGE DETECTION

Edge detection is a process of locating points in an image where image function (e.g., brightness) has abrupt changes [12]. Two types of edges can be defined depending on the type of image that is being processed. Ideal edges are very rare and can only be encountered in computer-generated images comprised of basic geometric objects such as triangles, squares, etc. These are easy to detect since intensity change between object and background is instantaneous. Real edges are common to photographic images or images of physical objects and natural scenes. In this case, change in intensity is not instantaneous and harder to detect. Figure 3.1 illustrates the differences between ideal and real edges.

![Fig. 3.1 Ideal vs. real edge.](image)

A number of different edge detection operators exist and are already in use including Roberts, Sobel, Prewitt, Kirsch, Robinson, and Laplacian of Gaussian
operator [1,12]. For edge detection in color images, several approaches have been developed and studied [10, 19-21]. Most common approaches include edge detection on an equivalent intensity image and on each of color planes, respectively. Color representation is not limited to the RGB space and may be performed in such others as L*a*b* and YIQ representations [12]. Most common problems with color edge detectors include missed edge points, noise misclassified as edge points, and smeared edges.

![Original image](image1)

![Broken edges](image2)

![False edges caused by noise](image3)

![Smeared edges](image4)

**Fig. 3.2** Erroneous outputs produced by an edge operator.

These types of behavior can be observed for most of the existing edge operators. Depending on the task at hand, these problems may or may not be of big importance. Figure 3.2 shows possible outputs after edge detection is complete.

In this research we use a typical method of performing edge detection in a color image discussed in [1]. In this approach, edge detection is performed on each
of the R, G, and B color planes individually. The three resulting edge maps $M_R$, $M_G$, and $M_B$ are then combined into a single one ($M$) by using the logical OR operation

$$M = M_R | M_G | M_B$$  \hspace{1cm} (3.1)$$

where $|$ denotes the logical OR operation (see also Eq. (4.1)).

There is a class of edge operators called zero-crossing edge detectors [1]. The way they work can be easily illustrated by the one-dimensional step edge example shown in Fig. 3.3. The place where image function (e.g., brightness) has an abrupt change is exactly the place where the second derivative has a zero crossing.

Mathematically it can be written as

$$\left. \frac{d^2 f}{dx^2} \right|_{x=x_z} = 0$$

$\begin{align*}
\frac{d^2 f}{dx^2} &> 0 \quad (\leq 0), x \leq x_z \\
\frac{d^2 f}{dx^2} &< 0 \quad (> 0), x \geq x_z
\end{align*}$  \hspace{1cm} (3.2)
The isotropic generalization of the second derivative to two dimensions is the Laplacian. The Laplacian of a function $I(r,c)$ is defined by

$$\nabla^2 I = \left( \frac{\partial^2}{\partial r^2} + \frac{\partial^2}{\partial c^2} \right) I = \frac{\partial^2 I}{\partial r^2} + \frac{\partial^2 I}{\partial c^2}$$

(3.3)

where $r$ and $c$ are the spatial coordinates. The differencing entailed by taking first or second order derivatives needs to be stabilized by some kind of smoothing or averaging. Marr and Hildreth suggest using a Gaussian smoother [1]. The resulting operator is called the Laplacian of Gaussian (LOG) edge operator which is given by

$$LOG(r,c) = \left( \frac{\partial^2}{\partial r^2} + \frac{\partial^2}{\partial c^2} \right) \frac{1}{2\cdot\pi\cdot\sigma^2} e^{-\frac{1}{2} \left( \frac{r^2 + c^2}{\sigma^2} \right)} = \frac{-1}{2\cdot\pi\cdot\sigma^4} (2 - \frac{r^2 + c^2}{\sigma^2}) \cdot e^{-\frac{1}{2} \left( \frac{r^2 + c^2}{\sigma^2} \right)}$$

(3.4)

where $r$ and $c$ are the respective row and column indices and $\sigma^2$ is the variance of the Gaussian kernel. Figure 3.4 shows a 1-D graphical representation of equation (3.4).

![Fig. 3.4 Cross-section of continuous domain Laplacian of Gaussian impulse response.](image)

It has been shown that for the ideal step and ramp edges, the location of the edge is exactly where one would expect it to be. However, for the finite-width step and the staircase step edges, the edge location may be shifted. This shift is a function of the standard deviation of the Gaussian [1].

The central area of the kernel is a disk of radius $\sqrt{2}\sigma$. The domain of the LOG kernel must be at least as large as a disk of radius $3\sqrt{2}\sigma$ (see, for example, Fig.
3.5). In actual practice, since only a zero crossing is being looked for, the LOG kernel is multiplied by some constant, and the resulting values are quantized to integers, with some care being taken to do the quantization so that the sum of the positive entries equals the absolute value of the sum of the negative values. One way of accomplishing this is to define

\[ LOG(r,c) = \text{truncate}[A(1-k^2 + c^2 / \sigma^2)] \]

where \( k \) is defined to be the value that makes

\[ \sum_{r=-N}^{N} \sum_{c=-N}^{N} LOG(r,c) = 0, \text{where } N = \left\lfloor \frac{3\sqrt{2}\sigma}{2} \right\rfloor \]

and \( A \) is chosen to be just less than the largest value of \( A \) that would make \( LOG(N,N) = 0 \). Figure 3.5 shows an example of an 11x11 Laplacian of Gaussian kernel for \( \sigma = 1.4 \).

Once the image is convolved with the LOG kernel, the zero crossings can be detected in the following way. A pixel is declared to have a zero crossing if it is less than \(-t\) and one of its eight neighbors is greater than \(t\), or if it is greater than \(t\) and one of its neighbors is less than \(-t\), for some fixed threshold \( t \), or
zero-cross if

\[
\begin{cases}
    f^c(r,c) > t, \text{and } f^c(r \pm 1, c \pm 1) < -t \\
    f^c(r,c) < -t, \text{and } f^c(r \pm 1, c \pm 1) > t
\end{cases}
\]

(3.7)

where \( f^c(r,c) \) specifies the image function after the convolution, and \( r \) and \( c \) are the row and column indices, respectively.

Fig. 3.6 Results of the Laplacian of Gaussian edge detector for different threshold values \( t (\sigma = 2) \). Original image is of size 159x153 with 24 bpp color specification.

Fig. 3.7 Results of the Laplacian of Gaussian edge detector for different values of \( \sigma (t=0.5) \). Original image is of size 159x153 with 24 bpp color specification.

Figure 3.6 shows a sequence of edge-detected images for various values of the threshold \( t \). Figure 3.7 illustrates the dependence on the value of \( \sigma \). Here, edge
detection was performed on each of the RGB planes of the sample color image and the composite edge map was formed by OR-ing (Eq. 3.1) the individual edge maps for the R, G, and B components respectively [10]. Notice that the higher the value of the threshold or \( \sigma \) the better the edge detection result is. In the former case, improvement is due to the increased value of \( t \); i.e., the higher the \( t \) the less noise-triggered edges we get. In the latter case, the increased value of \( \sigma \) results in better averaging due to the larger kernel size and, therefore, a noise free edge-detected image is obtained. Increase in \( \sigma \), however, comes at a price of heavier computation; i.e., increased kernel size adds some extra computation time to perform convolution between the original image and the discrete kernel.

The Laplacian of Gaussian edge detector is highly efficient and reliable. It produces continuous edges with minimum amount of loss; i.e., it rarely misses a valid edge. Edge detection quality can also be controlled by two parameters – the threshold value \( t \) and \( \sigma \). High performance and flexibility make the LOG edge detector a unique tool for performing edge detection for such an unusual purpose - color cluster separation in the RGB tri-stimulus space.
Chapter 4

EDGE-SUPPRESSED COLOR IMAGE INDEXING AND RETRIEVAL USING ANGLE-DISTANCE MEASUREMENT IN THE SCALED EIGEN-SPACE

In this research, we have developed an edge-suppressed indexing and retrieval method for color images based on scaled-space eigenvector analysis. The proposed method differs from the existing approaches (e.g., histogram intersection, distance method, reference color table method, and others [4-7, 9, 13, 14]) in the sense that the low and high frequency contents of an image are separated before the characteristic value analysis. This helps determining the high and low activity areas in the spatial plane of a given image so that the PCA analysis can be directed toward the spectrally rich and descriptive portions of object, background, and/or foreground regions of interest. High-frequency content of an image is usually due to the presence of edges or noise. Edge colors are by nature transitional ones which do not carry significant information about the spectral contents of objects within an image. Edges simply separate objects from the background and/or foreground. Removing such transition points from an image will help forming more compact color clusters in the 3-D RGB space than the sensory data alone [26, 27].
Figure 4.1 gives a simplified 2-D graphical representation of this idea and Fig. 4.2 shows the effect of edge suppression in a color image. Note that in the latter case the four color clusters are more distinct from the edge-suppressed color distribution than the original data alone.
4.1 Edge detection and suppression in color images

In this research, we have used the Laplacian of Gaussian (LOG) edge detector given by equation (3.5). Edge detection is performed using a discrete kernel (see, for example, Fig. 3.5) obtained by using equations (3.5)-(3.6) on each of the MxN R, G, and B color planes denoted as $X_R$, $X_G$, and $X_B$ resulting in three binary edge maps $M_R$, $M_G$, and $M_B$ of size MxN. A composite edge map $M$ is then formed by

$$M = M_R | M_G | M_B$$

(4.1)

where $|$ denotes the logical OR operation. Note that $M$ is a matrix consisting of only 0 and 1 values with ones representing the detected edges and zeros denoting the non-edge points. An edge-suppressed image is then obtained as

$$X_R' = X_R - M \cdot X_R; \quad X_G' = X_G - M \cdot X_G; \quad X_B' = X_B - M \cdot X_B;$$

(4.2)

This new image is composed of the color planes $X_R'$, $X_G'$, and $X_B'$, each of which is of the same size as the original input image. However, it has a denser color distribution (see, for example, Fig. 4.2) than the sensory data. This is of great importance for finding the three principal directions through which the image color distribution is governed as discussed in the next section.

4.2 Principal component representation of edge-suppressed color images

PCA analysis introduced in Chapter 2 is applied to the edge-suppressed color planes $(X_R', X_G', X_B')$ as follows. Let us denote the number of suppressed points in
Xi (i=R, G, B) as S. Vectors \( \mathbf{x}_i = [x_r, x_g, x_b]^T \) are random variables each with \((M \cdot N - S)\) samples. According to equations (2.4)-(2.5) covariance matrix \( C_x \) and mean vector \( \mathbf{m}_x \) are calculated. Using equations (2.6)-(2.7) we solve for eigenvalues \( \lambda_i \) and eigenvectors \( \mathbf{e}_i \) of the color covariance matrix \( C_x \).

Here, we explore the unique characteristics of the principle components of \( C_x \) to represent each model image as a 3-D orthogonal subspace \((\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)\) of the RGB tri-stimulus space. In a sense, this subspace, when centered at the mean vector location of an image color distribution, is the best approximation for the three principal directions through which the image color distribution is governed. Hence, this suggests that we can use this scaled space as a key or index for retrieval purposes.

4.3 Rotation, translation, and scaling (RTS) invariance property of principal component representation

The proposed feature space defined by \( \mathbf{m}_x \) and \( \mathbf{e}_i \) is also rotation, translation, and scaling invariant since it is based on the color information provided by the image. Rotation and translation invariance are obvious since neither changes the spectral contents of an image. Thus, \( \mathbf{m}_x \) and \( C_x \) remain the same as well as the eigen-data \((\lambda_i \text{ and } \mathbf{e}_i)\). Scaling invariance is proven in the following way.

Suppose the original image is scaled by factors of \( \alpha \) and \( \beta \). This means that size of the color planes \( X_R, X_G, \) and \( X_B \) (originally \( M \times N \)) is changed by \( \alpha \) and \( \beta \), which results in a modified size of \((\alpha M) \times (\beta N)\). Let us now assume that we have \( W \)
differently colored square areas with $Q_i$ pixels in each of them ($i=1, 2, \ldots, W$). This assumption is valid since each image can always be approximated by a number of uniformly colored squares. The minimum possible square is a single pixel. Before scaling, each area has a certain probability of occurrence calculated as

$$P_i = \frac{Q_i}{KN}, i = 1, 2, \ldots, W$$  \hfill (4.3)

After scaling the same probability can be expressed as

$$P_i = \frac{\alpha \beta Q_i}{\alpha \beta KN} = \frac{Q_i}{KN}, i = 1, 2, \ldots, W$$  \hfill (4.4)

Since each of the squares is scaled by the same factor $\alpha \beta$, the resulting probability of occurrence $P_i$ does not change. This proves that scaling will not change any statistical data used to obtain $m_x$, $\lambda_i$, and $e_i$.

**4.4 Similarity measure in the RGB space**

The similarity measure proposed in this research is based on the following observation. Suppose that we have two 3-D orthogonal sub-spaces given within the RGB tri-stimulus space by $m_x^j, \lambda_i^j, e_i^j$ ($i=1, 2, 3$ and $j=1, 2$) where $m_x$ is the mean vector of the $j$th sub-space, and $\lambda_i^j$ and $e_i^j$ are the respective eigenvalues and eigenvectors. Eigenvalues, eigenvectors, and mean values are all calculated using edge suppressed image data. Using edge suppression helps maximize cluster separability further improving the PCA performance. Each sub-space specified by the three orthogonal eigenvectors is centered at the mean vector $m_x$ (see Fig. 4.3).
It is clear that two classes defined by the sub-spaces are more similar if the distance between mean vectors and angles between the respective eigenvectors are minimum. Based on this observation, we propose the following similarity measure:

\[
s(f^q, f^m) = (1 - \frac{|\mathbf{m}_x^q - \mathbf{m}_x^m|}{255\sqrt{3}})(1 - (k_1\alpha_1 + k_2\alpha_2 + k_3\alpha_3)) \quad (4.6)
\]

where \( f^q = [\mathbf{m}_x^q, \mathbf{e}_j^q, \lambda_j^q]^T \) and \( f^m = [\mathbf{m}_x^m, \mathbf{e}_j^m, \lambda_j^m]^T \) are the respective feature vectors for the query and model images, \(||\) is the Euclidian distance between two points, \( \alpha_i \)'s are the normalized angles between corresponding eigenvectors, and \( k_j \)'s satisfy the following equality:

![Fig. 4.3. Similarity measurement between two 3-D orthogonal sub-spaces in RGB system.](image-url)
\[ k_1 + k_2 + k_3 = 1 \]  \hspace{1cm} (4.7)

Division by \( \frac{255}{\sqrt{3}} \) is needed to normalize all the distances in the RGB space. Values for \( k_j \) are calculated in the following way:

\[ k_j = \frac{\lambda_j^q + \lambda_j^m}{\lambda_1^q + \lambda_1^m + \lambda_2^q + \lambda_2^m + \lambda_3^q + \lambda_3^m} \]  \hspace{1cm} (4.8)

where \( \lambda_k^j \) are the eigenvalues of corresponding image color distributions and the superscripts \( q \) and \( m \) are for the query and model (database) images, respectively.

Values for \( \alpha_i \) are given by:

\[ \alpha_i = \frac{1}{2\pi} \cos^{-1} \left( \frac{e_i^q \cdot e_i^m}{|e_i^q||e_i^m|} \right) \]  \hspace{1cm} (4.9)

where \( e_i^j \) are the eigenvectors of image color distributions. It is assumed that all angles resulting from the computation of \( \cos^{-1} \) are normalized to be in the \([0, 2\pi]\) range.

Since both the distance and the angle measurements are normalized to be within \([0, 1]\) range, the new measure, which is essentially the product of them, takes values in the same range.

Researchers in \([13]\) suggested a similar measurement given by equation (1.12) for their cluster-based approach. As mentioned in Chapter 1, the proposed measure uses the identical distance measurement in the RGB space involving different angle measurements. Specifically, the measure proposed in this thesis uses all the three angles between the corresponding eigenvectors after scaling according to the respective eigenvalues, while the angles used in \([13]\) are limited to the projection
angles between the color mean vectors of the clusters detected. This, supposedly, makes the measure more powerful since orientations of principal vectors are being compared with each other with appropriate scaling based on the values of the respective eigenvalues.
Chapter 5

COMPUTER IMPLEMENTATION
AND
EXPERIMENTAL RESULTS

The proposed method for color image indexing and retrieval has been implemented on a PC system using MATLAB© and its image processing toolbox. The test prototype database that we have used contains over 180 true color (24 bits per pixel) images. Some of the images are related by rotation, translation, and scaling. The database contains approximately equal numbers of synthetic (computer generated) and natural images as well as combined ones. Image sizes vary from 64x64 to 800x600 increasing the complexity of the task by requiring scaling invariance from the method.

For database query, an image is submitted to the search engine and the search begins by calculating the feature vector $f^q$ of the query image using equation (4.6). The search then continues by comparing $f^q$ against those ($f^m$) of model images stored in the database already.

In this work, performance of the proposed algorithm is also evaluated by using the retrieval efficiency $\eta_T$, which is defined as

$$\eta_T = \begin{cases} \frac{k}{N}, & \text{if } N \leq T \\ \frac{k}{T}, & \text{if } N > T \end{cases} \quad (5.1)$$
where \( k \) is the number of similar images correctly retrieved in the short list with length \( T \) and \( N \) is the total number of similar images in the database [5]. Here, the subscript \( T \) indicates dependency of \( \eta_T \) on the list length selected during a retrieval session. Suppose that we have \( N=8 \) similar images in the database and we want to calculate \( \eta_T \) for different values of \( T \). We first set \( T \) to some value less than the number of similar images, for example 4. Suppose that the search engine returns \( k=3 \) similar images among the 4 that we have requested. In this case \( \eta_T = 3/4 \). If the value of \( T \) is now set to 10 and the engine returns \( k=6 \) similar images among those 10, then \( \eta_T = 6/8 \). Finally, if \( T=20 \) and the engine returns \( k=8 \) similar images, then \( \eta_T = 8/8 = 1 \) implying that we have 100% retrieval efficiency. No other increase in the value of \( T \) will increase \( \eta_T \). In other words, retrieval efficiency \( \eta_T \) saturates and stays at 1 after all similar images have been retrieved.

5.1 RTS invariance testing

We start by performing several experiments to probe method’s immunity to rotation, translation, and scaling. We also summarize the retrieval accuracy for different values of \( T \) in Table I by averaging search results over ten queries for each value of \( T \). Results in Table I are based on similarity measure proposed in this thesis.
Table I. Retrieval efficiency vs. threshold values for the new similarity measure.

<table>
<thead>
<tr>
<th>T</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>η_T</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

As can be concluded from the table, the method is indeed invariant to rotation, translation, and scaling of the query images. No matter how short or how long the requested list is, the method always brings RTS-related images before other similar images. This is easily explained by the fact that RTS-related images have almost identical distributions in the RGB tri-stimulus space due to similar probability distributions (see also section 4.3). As a result, equation (4.6) is very close to unity for such images. Appendix A shows images output by the search engine as a result of a query with an image having 3 similar images related to it by rotation, translation, and scaling.

5.2 The search engine approach

A more interesting and practically important way of testing the system is to build a search engine based on the proposed approach and determine its performance. An experimental search engine program has been developed using MATLAB©. The system has two modes of operation. The first mode allows searching for similar images within the database; i.e., query image is one of the model images. In this case, we are guaranteed to have at least one perfect match (similarity of an image
with itself is 1). The second mode is based on the idea introduced in Fig. 1.1, where a query image is submitted to the search engine from an external source. The engine then performs analysis and creates a feature vector \( f^e \) which is then matched against pre-calculated feature vectors \( f^m \) in the database. In a search engine there is no predefined threshold. After similarity measure between \( f^e \) and \( f^m \) is computed for each image, a list is created. This list contains all similarity values in descending order with corresponding image file names. A user is then presented with a list of images most closely matching the query image. The user, who may wish to get only \( N \) most closely matching images, can limit the length of the list. This kind of approach is very suitable for on-line or Internet implementation where a set of web pages may be generated as a result of a query. It is then up to the user to decide which images he/she wants.

5.3 Comparative study of approaches for color image indexing

In this section we compare the performance of several different methods introduced in Chapter 1 such as the histogram intersection method, the distance method, the color reference table method, Tanimoto similarity measure, and the relative entropy concept with the proposed edge-suppressed color image indexing and retrieval method. We use equation 5.1 to determine the retrieval efficiency \( \eta_T \) for different values of \( T \). Here, we use four sets of images containing natural scenes. Appendix H gives a list of the types of images used in testing.
Tanimoto similarity measure [2] and the relative entropy measure [18] formulas are given by

\[
s(f^i, f^m) = \frac{f^i \mathbf{f}^m}{f^i \mathbf{f}^i + f^m \mathbf{f}^m - f^i \mathbf{f}^m} \quad (5.2)
\]

and

\[
E_{12} + E_{21} = \sum_{j=1}^{N} \begin{cases} 
(p_j^i - p_j^m) \log \left( \frac{p_j^i}{p_j^m} \right), & \text{if } p_j^i, p_j^m \neq 0 \\
0, & \text{otherwise}
\end{cases} \quad (5.3)
\]

where \( f^i, f^m \) are the input and the model feature vectors, and \( p_i^i, p^m \) are the probability distribution functions with \( N \) bins for the input and the model images, respectively.

Tanimoto similarity measure assumes values in the range \([-1, 1]\) with 1 indicating a perfect match and -1 implying a complete mismatch. In these experiments, a scaled version of eigenvectors is used in order to give more significance to directions in image color distribution that carry the largest amounts of energy. The feature vector used with the Tanimoto measure in this research is given by

\[
f = (m_x + \lambda_1 e_1, m_x + \lambda_2 e_2, m_x + \lambda_3 e_3) \quad (5.4)
\]

where \( m_x \) is the color mean vector of a given image, and \( \lambda_i \) and \( e_i \) are the respective eigenvalues and eigenvectors of the color covariance matrix \( C_x \).

The relative entropy formula gives values in the range \([0, \infty]\). In this case, zero means a perfect match and any deviation from zero implies differences in the probability density functions of the distributions being compared, and, respectively, in the images themselves. Formulas for feature and similarity computations for other methods were given in Chapter 1.
We summarize the results of our experiments in Table II and Fig. 5.1. For histogram intersection and relative entropy, the RGB space was quantized into 8x8x8 regions to minimize the histogram size and, respectively, the computation time. Sample search queries and the resulting outputs are given in appendices B through G.

Table II. Retrieval efficiency vs. list length values for different retrieval methods.

<table>
<thead>
<tr>
<th></th>
<th>List length T</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4  8  10  15</td>
</tr>
<tr>
<td>Proposed method, $\eta_T$</td>
<td>0.8 0.84 0.93 1</td>
</tr>
<tr>
<td>Tanimoto measure, $\eta_T$</td>
<td>0.71 0.5 0.52 0.55</td>
</tr>
<tr>
<td>Histogram intersection (8,8,8 linear quantization for RGB), $\eta_T$</td>
<td>0.56 0.65 0.73 0.97</td>
</tr>
<tr>
<td>Distance method, $\eta_T$</td>
<td>0.63 0.78 0.85 0.96</td>
</tr>
<tr>
<td>Color reference table method, $\eta_T$</td>
<td>0.92 0.8 0.8 0.84</td>
</tr>
<tr>
<td>Relative entropy (8,8,8 linear quantization for RGB), $\eta_T$</td>
<td>0.25 0.19 0.19 0.31</td>
</tr>
</tbody>
</table>

Fig. 5.1 Retrieval efficiency vs. list length values for different retrieval methods.

(Legend:
- : new method;
... : distance method;
◇ : color reference table method;
* : histogram intersection;
▷ : Tanimoto measure;
▵ : relative entropy)
As can be concluded from the table and the corresponding graph, the proposed method falls in the same performance category as the color reference table, distance method, and histogram intersection method. This suggests that it can be used together with or instead of all these methods providing better or equivalent performance.
Chapter 6

CONCLUSIONS AND FURTHER RESEARCH

A new edge-suppressed color image indexing and retrieval method has been developed. Advantages of this method include computational efficiency, RTS invariance, and high performance. It is proved experimentally that it is comparable to the existing color-based indexing and retrieval methods such as the color reference table, histogram intersection, and the distance method. The key feature of the method is the principal component analysis approach that allows computationally fast and accurate representation of random color distributions in the RGB space. A new measure has been developed to allow fast calculation of similarity between two 3-D orthogonal spaces and, respectively, between corresponding color images. The measure is designed to operate in the RGB tri-stimulus space. It uses a combination of angle and distance measurements to compare two distributions and, importantly, it is also dynamically scaled according to eigenvalues of the distributions being compared. Test results in Chapter 5 and visual examination of output images as a result to a certain query (see appendices) show that the measure is capable of identifying perceptually similar images or those that would be selected by a human operator as similar. This behavior and high performance may be attributed to the fact that we use a unique spectrum of high performance mathematical tools and engineering methods (such as the PCA, the LOG edge detector, and the angular-
distance measure) as well as years of experience provided by other researchers in this area.

Although this method can be used as a single tool for color image indexing and retrieval, a better approach may be devised by combining it with some shape or texture similarity measure. One of the ways to further enhance the performance of the proposed method is to introduce a second step of PCA analysis and a cluster-merging algorithm. In this work, edge suppression is followed directly by the principal component analysis step and the similarity measure calculation. Figure 4.2 shows that edge suppression is a good tool to perform cluster separation in the color space. Therefore, by performing region merging on the edge-suppressed image we obtain even better cluster separation with close clusters combined into single entities. The principal component analysis can be performed on each of the detected clusters resulting in seven characteristic features for each cluster; i.e., three eigenvalues, three eigenvectors, and a single mean vector. This information can be used to search for images that have similar cluster distribution characteristics. A two-step matching process may be executed in a hierarchical manner. First, only the global eigendata is used to select the best candidates. Eigendata for each of the clusters is then used to find a better match among the preselected images. If this methodology is to be used then a different similarity measure must be developed in addition to the one given by equation (4.6). Since, in general, there are several clusters in a color distribution of a given image this measure should perform matching for each of the clusters and then
find the best match based on some criteria. This is, however, a non-trivial task and it can serve as a separate very promising research topic.
References


Appendix A

Rotation, translation, and scaling invariance test results.

1. C17.jpg [159x153]  
   Similarity 1 (Query)

2. C18.jpg [153x159]  
   Similarity 1

3. C19.jpg [111x107]  
   Similarity 0.99

4. C20.jpg [159x153]  
   Similarity 0.96
Appendix B
A set of images retrieved by using the new similarity measure.

1. C171.jpg [375x500]  
Similarity 1 (Query)

2. C174.jpg [375x500]  
Similarity 0.98

3. C172.jpg [375x500]  
Similarity 0.97

4. C173.jpg [375x500]  
Similarity 0.97

5. C175.jpg [375x500]  
Similarity 0.97
Appendix C
A set of images retrieved by using the Tanimoto measure.

1. C171.jpg [375x500]
   Similarity 1 (Query)

2. C108.jpg [600x800]
   Similarity 0.99

3. C116.jpg [198x359]
   Similarity 0.99

4. C149.jpg [333x400]
   Similarity 0.99

5. C48.jpg [200x169]
   Similarity 0.98
Appendix D
A set of images retrieved by using the distance method.

1. C171.jpg [375x500]
   Similarity 1 (Query)

2. C174.jpg [375x500]
   Similarity 0.99

3. C148.jpg [332x400]
   Similarity 0.98

4. C149.jpg [332x400]
   Similarity 0.98

5. C150.jpg [294x400]
   Similarity 0.98
Appendix E
A set of images retrieved by using the histogram intersection method.

1. C171.jpg [375x500]
   Similarity 1 (Query)

2. C126.jpg [600x800]
   Similarity 0.17

3. C139.jpg [364x462]
   Similarity 0.15

4. C155.jpg [500x400]
   Similarity 0.15

5. C154.jpg [500x400]
   Similarity 0.14
Appendix F

A set of images retrieved by using the color reference table method.

1. C171.jpg [375x500]
   Similarity 1 (Query)

2. C108.jpg [600x800]
   Similarity 0.96

3. C172.jpg [332x400]
   Similarity 0.96

4. C172.jpg [375x500]
   Similarity 0.96

5. C173.jpg [375x500]
   Similarity 0.96
Appendix G
A set of images retrieved by using the relative entropy measure.

1. C171.jpg [375x500]  
Similarity 1 (Query)

2. C64.jpg [150x150]  
Similarity 1

3. C172.jpg [600x800]  
Similarity 1

4. C129.jpg [600x800]  
Similarity 1

5. C160.jpg [396x600]  
Similarity 0.99
Appendix H
Representative members of image sets used in testing.

1. C131.jpg [360x244]
   Similar images in the database - 6

2. C161.jpg [375x500]
   Similar images in the database - 8

3. C180.jpg [375x500]
   Similar images in the database - 8

4. C169.jpg [375x500]
   Similar images in the database - 8