Image Retrieval and Reversible Illumination Normalization

Longin Jan Latecki[a], Venugopal Rajagopal[a], Ari Gross[b]

[a]CIS Dept., Temple University, Philadelphia, PA, USA, latecki@temple.edu
[b]Queens College, CUNY, Flushing, NY, USA, ari@vision.cs.qc.edu

ABSTRACT

We propose a novel approach to retrieve similar images from image databases that works in the presence of significant illumination variations. The most common method to compensate for illumination changes is to perform color normalization. The existing approaches to color normalization tend to destroy image content in that they map distinct color values to identical color values in the transformed color space. From the mathematical point of view, the normalization transformation is not reversible.

In this paper we propose to use a reversible illumination normalization transformation. Thus, we are able to compensate for illumination changes without any reduction of content information. Since natural illumination changes affect different parts of images in different amounts, we apply our transformation locally to sub-images. Basic idea is to divide an image into sub-images, normalize each one separately, and then project it to an *n*-dimensional reduced space using principal component analysis. This process yields a normalized texture representation as a set of *n*-vectors. Finding similar images is now reduced to computing distances between sets of *n*-vectors. Results were compared with a leading image retrieval system.

Keywords: CBIR, Image Retrieval, Image Similarity, Color Normalization, Brightness Normalization.

1. INTRODUCTION

In recent years there had been a significant growth in research into methods for organizing and retrieving similar images from the databases. Some of the techniques include text labeling and Content Based Image Retrieval (CBIR). In text labeling each image is indexed by keywords. The keywords are chosen in a way that best represent the image that is stored, so reducing the search to merely a keyword search, which is done using some string matching technique. One particular disadvantage of this approach is its supervisory nature. In CBIR images are retrieved automatically based on the color, texture and shape features of the image [1]. There are a lot of commercial software's, which implement CBIR techniques and its feasibility is still a hot research area topic.

An overview of popular systems using CBIR can be found in [2]. Even though there are abundant systems available out in the market, the problem of image similarity is far from being solved. Apart from the main problem of recognizing similar images based on their semantic meaning [3], there still exist serious problems in recognizing images that are nearly identical to humans. For example, most existing systems still are not able to recognize as similar images that differ only by brightness. This is due to the fact that brightness variations that arise under natural conditions (different light conditions or different camera) affect different parts of an image differently. Therefore, the commonly used techniques of global color normalization [21] (to compensate for brightness changes) cannot provide a reliable solution to the problem of varying brightness. The color normalization techniques introduce, however, a more serious problem of content destruction, since the normalization is not a reversible transformation, i.e., the original image cannot be reconstructed from the brightness-normalized image. The reason for this is very simple: all existing brightness normalization transformations are not injective, i.e., they may map distinct pixel values to the same value, which makes the content information of these pixels disappear. One of most commonly used formulas for brightness normalization is as follows, e.g., [2]:

$$c_4(R,G,B) = \frac{R-G}{R+B}$$

$$c_5(R,G,B) = \frac{R-B}{R+B}$$

$$c_6(R,G,B) = \frac{G-B}{G+B}$$

where R, G, B are classical color values of red, green, and blue color components. Due to the subtraction in the numerators, it is obvious that many different R, G, B values are mapped to the same values, i.e., the mapping function is not reversible. This information loss may cause serious difficulties in recognizing similar images. Example images demonstrating this fact are shown in Figures 2.1 and 2.2 in Section 2.

Based on the above discussion, we arrive at the following two main requirements for any useful technique for brightness normalization:
1) it must be local, and
2) it must be reversible.

In this paper we present an image representation that satisfies both requirements. It uses Principal Component Analysis (PCA) [14] to store and retrieve images from the database. However, in contrast to standard approaches such as eigenfaces or eigenimages [18, 19, 20], we apply PCA locally. Thus, in our approach an image representation is not a single point in a high dimensional features space, but a set of points.

There are numerous image retrieval systems available. All of them seem to either use global brightness normalization or simply do color quantization without any normalization, e.g., histogram binning. They include, *Blobworld* [5] that uses histogram binning to describe color features, *C-bird* [6] that uses normalization of each RGB channel before computing its chromaticity feature descriptor, *ImageRETRO* [7] that uses a 15 bin hue histogram as one of the color features, *MARS* [8] that uses a 2D histogram over the HS coordinates of the HSV space to represent color features, and two histograms one measuring coarseness other the directionality of the image to represent texture features, *QBIC* [10] uses a 256 dimensional RGB color histogram as one of the color features, *Surfimage* [11] uses RGB color histogram, edge orientation histogram as one of the low level features, *WISE* [12] in the preprocessing step rescales images and converts from RGB color space to another color space using normalization.

We decided to compare our results with PicToSeek, because of its good performance, and demonstrate the effects of normalization. *PicToSeek* [9] defines a new model $c_4c_5c_6$ and a set of color invariants $l_4l_5l_6$ over RGB using color normalization, which is briefly explained in Section II.

Recently also machine learning techniques have been used to index and classify images. For example, vector quantization is used in [22] to obtain a codebook of semantically meaningful image categories like sky, grass. However, it turned out that brightness strongly influence the codebook images, which simply means that similar images of grass that are darker then the codebook images will not be recognize as grass.

A promising machine learning technique is nonlinear dimensionality reduction applied to geodesic distances on surface of image vectors [24]. This technique was able to perform dimensionality reduction that abstracts from image brightness. However, as for now this fact is only demonstrated on a set of very similar face images, and it is unclear whether it will work on a large set of strongly varying images.

The proposed approach is also based on dimensionality reduction. However, we apply linear dimensionality reduction locally to image parts, i.e., in [24] the whole images are points in the feature space while in our approach image parts are points in the feature space. This gives our approach a major advantage for images with locally varying brightness, since we perform gray values' normalization locally, for each image part independently, while the approach in [24] performs global normalization of gray values for the whole images. Consequently, approach in [24] cannot cope with locally varying brightness, which is a normal condition for most natural (indoor and outdoor images).

The rest of this paper is organized as follows. In Section 2 we give some background information on Principal Component Analysis, PicToSeek, and our framework; Section 3 describes methodology behind our technique; Section 4 analyzes the results from the experiments; Section 5 explains background learning one particular advantage in using our approach; Section 6 summarizes our conclusion.

2. BACKGROUND

2.1. Principal Component Analysis

Principal component analysis (PCA) involves a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible [13]. For more in depth explanation please refer [14]. Some of the applications of PCA are identifying patterns in a data, dimensionality reduction etc.

2.2. PicToSeek

PicToSeek combines the color and shape invariant information for retrieving images. Color invariant information is derived by transforming RGB color space of a given image into $c_4c_5c_6$ and $l_4l_5l_6$ space respectively. We refer these as *c* and *l* images. The first step of computing the *c* and *l* images is done using the formulas presented in Section I and the following formulas:

$$l_4(R,G,B) = \frac{|R-G|}{|R-G|+|B-R|+|G-B|}$$
$$l_5(R,G,B) = \frac{|R-B|}{|R-G|+|B-R|+|G-B|}$$
$$l_6(R,G,B) = \frac{|G-B|}{|R-G|+|B-R|+|G-B|}$$

The second step involves computing normalized histograms based on the above color invariants (*c* and *l* images). Based on the transformed color spaces $c_4c_5c_6$ and $l_4l_5l_6$, three-dimensional histograms H_A, H_B are constructed. The distribution of edges computed in $l_4l_5l_6$ color space is represented by the one-dimensional histogram H_C. These three histograms represent the color invariant information of a given image. Using $l_4l_5l_6$-based color invariant edges as feature points one-dimensional histogram H_D is constructed on the angle axis expressing the distribution of angles, and one dimensional histogram H_E is constructed on the cross ratio axis expressing the distribution of cross ratios. These two histograms represent the shape invariant information of a given image. A four-dimensional histogram $H_F(i,j,k,l)$ is created counting the number of color invariant edge triples (i,j,k) generating an angle *l*. This histogram represents the composite color and shape invariant information of a given image. Matching two images is expressed as histogram intersection that involves all histograms H_j for $j \in \{A,B,C,D,E,F\}$. For more in depth explanation on PicToSeek please refer [9].

The main problem with this and similar approaches is the fact that they fail to find even an identical image that differs in brightness. This is due to the fact that employed brightness normalization maps different color values to the same values. Consequently, the edge images computed after the normalization will be different. Figures 2.1 and 2.2 illustrate this fact. They show two versions of the same image. The normalizations of both versions to $c_4c_5c_6$ and $l_4l_5l_6$ spaces respectively are shown in the first rows. In the second rows, we see the edges obtained by Canny edge detector (We used Canny edge detector included in Matlab, Image Processing ToolBox, with standard settings). By comparing the corresponding edge images in both figures, one can draw the following conclusion. Although the edge images of the input images are already different, the color normalizations make the differences more significant. Consequently, it is impossible to identify the two input images as similar using the derived edge images.

This example illustrates why PicToSeek and similar systems, e.g., listed in the introduction, lead to wrong retrieval results for very similar images. As brightness normalization is performed before constructing the histograms and feature matching, the loss of content information causes similar images to have very different representations. This is illustrated in Section 4 for PicToSeek using one of our small image databases.

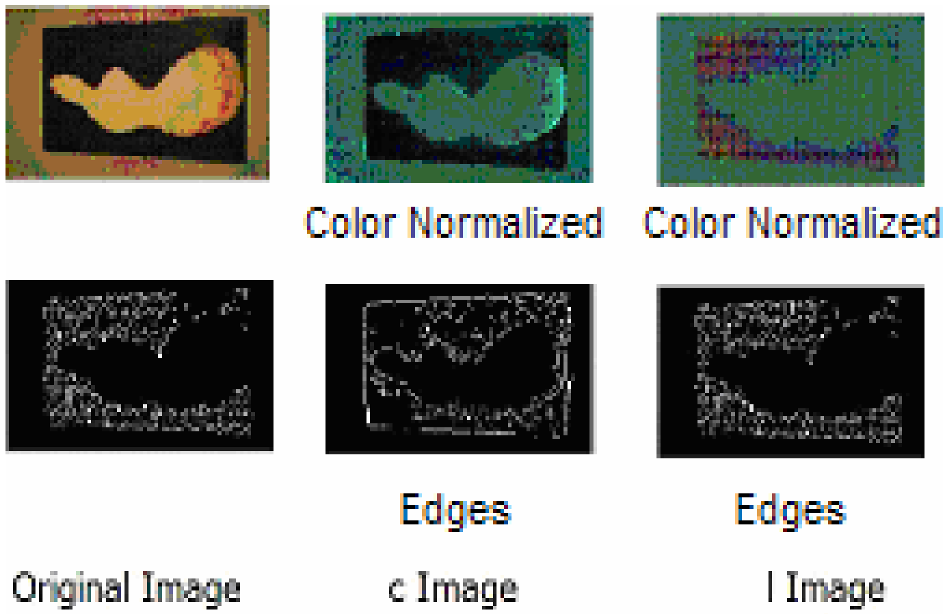

Figure 2.1 Edge images differ significantly from the edge images of the very similar image in Figure 2.2.

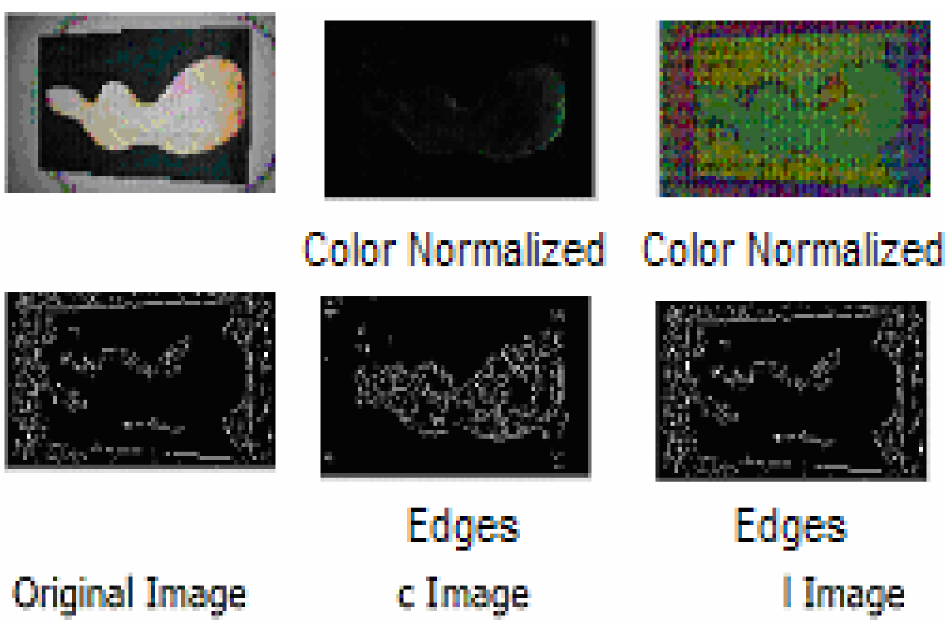

Figure 2.2 Edge images differ significantly from the edge images of the very similar image in Figure 2.1.

2.3. Framework

The framework depicted in Figure 2.3 can conceptually describe many of the image retrieval systems.

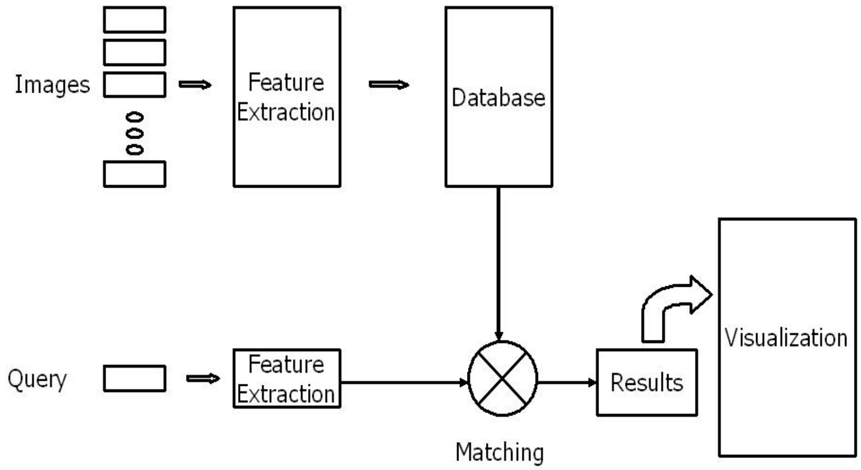

Figure 2.3 Framework

There are three phases involved in any image retrieval framework. The first phase is the feature extraction phase for the images in the database and the query image. This is done using some CBIR technique and these extracted features represent the images. The second phase is to match the features of the query image with the features of all the images to find the most similar image. The third phase is the result visualization where all the similar images are displayed based on their ranking.

3. METHODOLOGY

3.1. Illumination Normalization

Since our goal is to define local texture representation, we work only with gray level images. Therefore, color images are first converted to gray level. Let $\Omega = \{A_1, A_2, \ldots A_m\}$ be the images in the database. Given an image A_i of height H and width W, we divide it into n sub-images (non-overlapping blocks) each of size h and w (height h and width w) where $h<H$ and $w<W$, which is illustrated in Figure 3.1. Every block is normalized independent from other blocks by subtracting the mean of the block from each pixel. This local normalization of gray level values makes the proposed approach more robust to varying light under real conditions. We stress that this fact is ignored in most existing brightness normalization approaches. In all presented experiments we used block sizes of $h = 32$ and $w = 32$. A set of blocks obtained from a group of training images is used to compute the principal components. By selecting first n principal components, we obtain a projection matrix EV to project each block to an n dimensional vector that constitutes its texture representation.

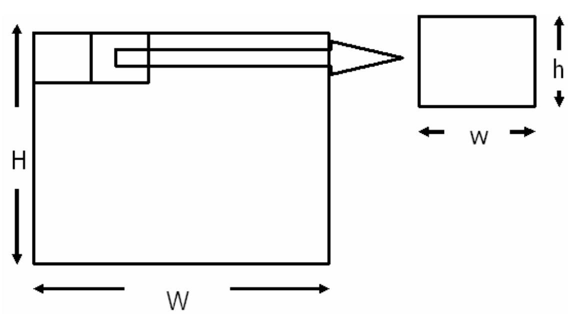

Figure 3.1 Image sub-divided into blocks

Every block is then projected to an n-dimensional subspace spanned by the principal components, which is illustrated in Figure 3.2. The right hand side of Figure 3.2 shows the plot of the image features along the first three dimensions.

The obtained *n*-vectors provide a compact texture representation of the corresponding blocks. Each image is represented by a set of *n*-vectors representing its blocks. Hence each image a_i is represented as $r(a_i) = \{a_{i1}, a_{i2}, ... a_{in}\}$.

The obtained vector $r(a_i)$ provides a representation of image a_i that is illumination invariant and is reversible if $n=h*w$. However, this level of detail is not useful for image similarity. Therefore, *n* is usually significantly smaller than $h*w$, in which case the obtained illumination-normalized texture representation is approximately reversible. This means that reconstructed image using the inverse PCA transformation approximates the original image. The higher is n the more accurate is this approximation. In our experiments we used a projection matrix composed of the first *10* principal components, i.e., the size of the learned projection matrix in our experiments is *1024x10*, and each texture representation is an *n*-vector. The reconstructed images from first 10 PCA components are shown in Fig. 3.3. The input images differ by brightness and contain (painting frame).

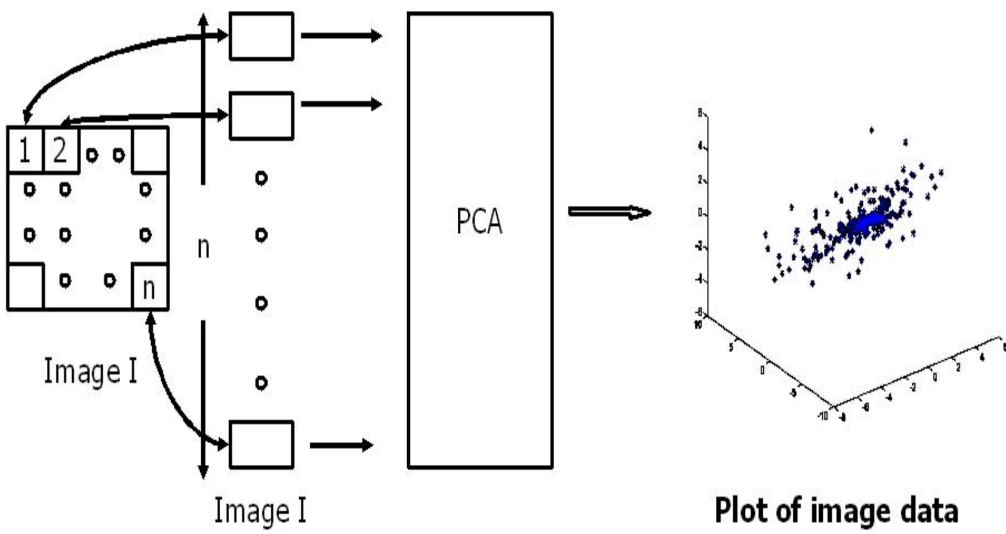

Figure 3.2 Flow of image features' extraction

3.2. Features Matching

Given a query image it is compared block-wise with each of the images in the database. Each image is represented as a set of points in an *n* dimensional space. Hausdorff distance [15] is used to find the closeness between two sets. Given two sets *A* and *B*, Hausdorff distance is defined as

$$H(A,B) = \max\left(h(A,B), h(B,A)\right)$$

$$\text{where } h(A,B) = \max_{a \in A}\left[\min_{b \in B}\left[d(a,b)\right]\right]$$

and viceversa. Euclidean distance was used as the distance metric for *d*. If *A* is the query image and *B* is the image under consideration, each block in *A* will be associated with a value *c*, which represents the distance of the closest block in *B*. Figure 3.4 shows the one-dimensional plot, of sorted *c* values in descending order with the value of the *c* in y axis between A and B. First 10 percent of the data were discarded before computing *h(A,B)* and *h(B,A)* to handle outliers. The similarity measure obtained by *H(A,B)* is used to rank all the images in the database for a given query image *A*.

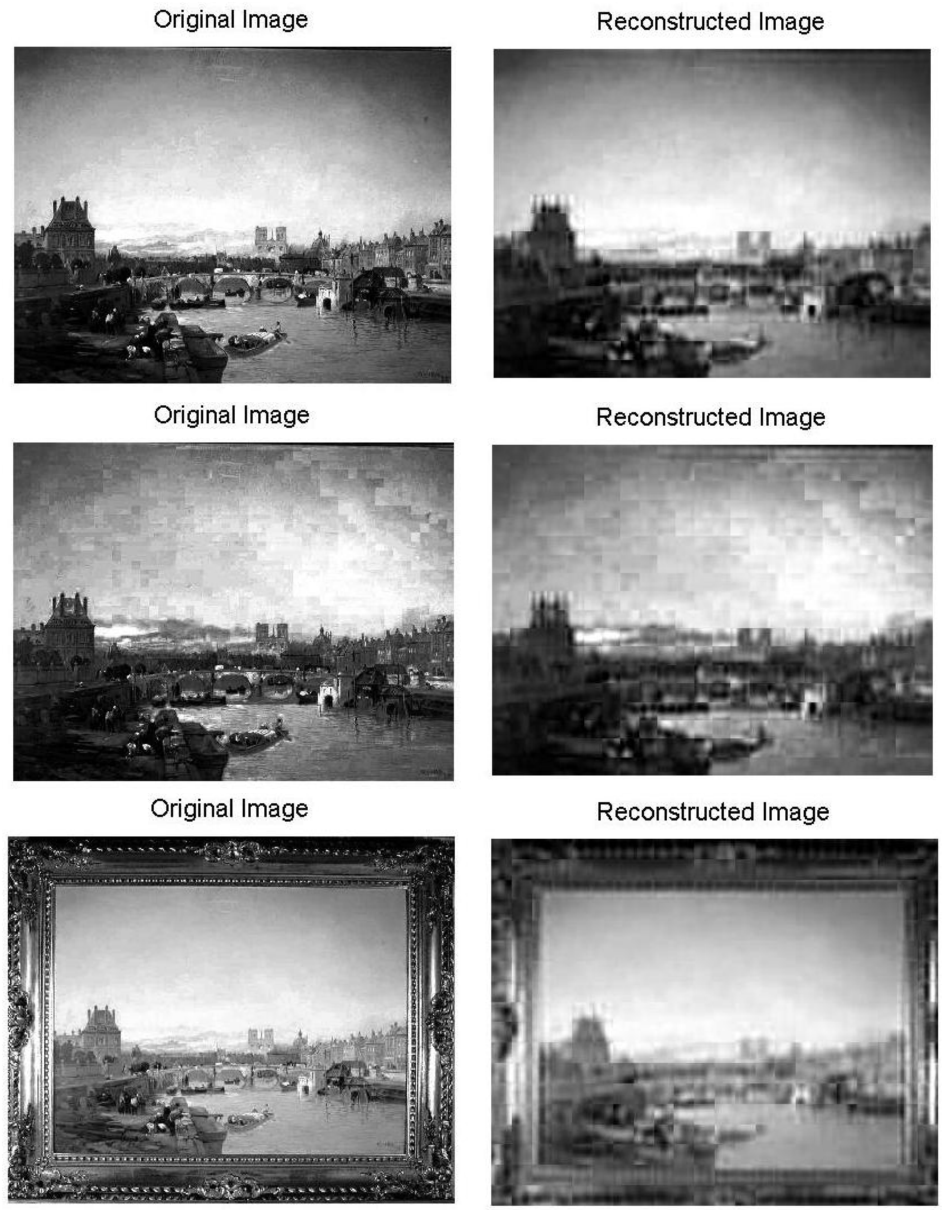

Figure 3.3. Reconstructed images from first 10 PCA components. The input images differ by brightness and contain (painting frame).

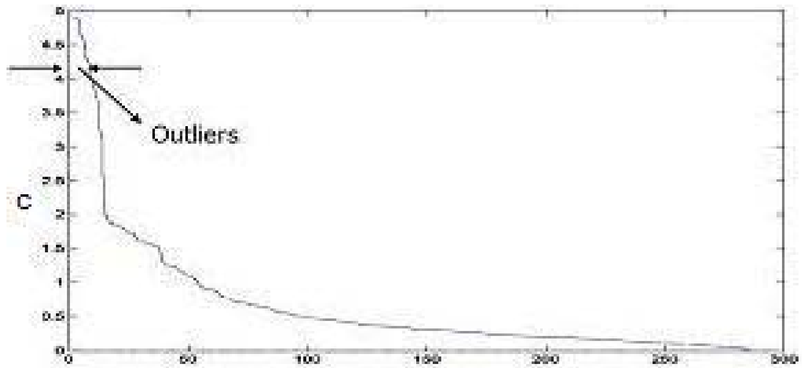

Figure 3.4 Plot of sorted block distances between two images.

4. IMAGE RETRIEVAL EXPERIMENTS

Experiments were conducted on two types of image datasets one consisting of 8 photographs shown in Figure 4.1 and the other consisting of painting images. For all the images in the database at least one extra image is available that taken under different illumination conditions. It was used as query image.

Figure 4.1 Our small database consists of 8 images.

In the first experiment, we gave the image from Figure 2.2 as the query image to our test database shown in Figure 4.1 using our implementation of PicToSeek system. Histograms were constructed for these transformed images and histogram intersection were used, as explained in Section 2, to find the closest match with respect to the query image. The images retrieved by PicToSeek are shown in Figure 4.2. The system failed to retrieve the second variant of the query image among the most similar images. Since the other retrieved images are completely different, this experiment clearly confirms our conclusions in Section 2 on the effects normalization can have on image similarity.

Figure 4.2 Output with respect to query image from Fig. 2.2.

We then performed similar kind of experiments in our system. We queried the database with the same image as in Figure 4.2. The first four images retrieved by our system are shown in Figure 4.3. We were successful in retrieving the similar images based on this query image. This example demonstrates that our approach does not suffer the consequences of normalization. We further continued experimenting with a different dataset. An example retrieval result is shown in Figure 4.4. This example illustrates that the proposed approach is not only robust in the presence of strong brightness variations but also in the presence of moderate content changes. The third retrieved image (Output2) shows the same painting as the query image but it shows additionally a painting frame. Output1 differs from the query in brightness and color distributions.

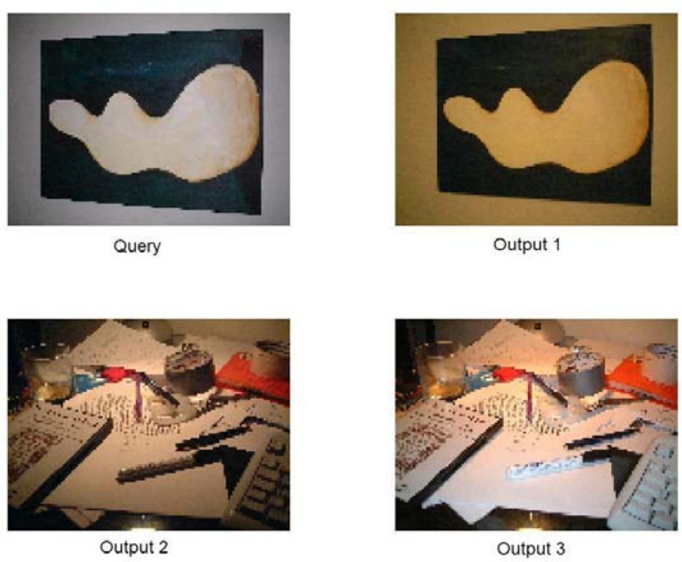

Figure 4.3 Output from our approach for the same database and query image as in Figure 4.2.

Figure 4.4 Examples of retrieved images

5. BACKGROUND LEARNING

One of the main advantages of the presented local texture representation is the fact that we learn the projection matrix *EV* from a set of image blocks. We recall that classical eigenfaces or eigenimages approaches require a set of images to learn *EV*. This fact allows us to normalize each block separately to compensate for brightness changes, as we stated above. In this section we illustrate a different advantage of this fact: we can train the *EV* matrix using a single image, which can be useful in background learning.

We employ the following background learning procedure: After constructing the *EV* matrix for a given image, we project all image blocks to their texture representation as n dimensional vectors in the PCA space. Then we identify the most compact cluster of the obtained n-vectors. The background blocks correspond to the n-vectors in this. An example image with its background identified by this method is shown in Figure 5.1. The original input image is shown in Figure 5.2. In the context of document image analysis, the background learning can be used to exclude background blocks from further processing, e.g., to have OCR software to analyze only the foreground.

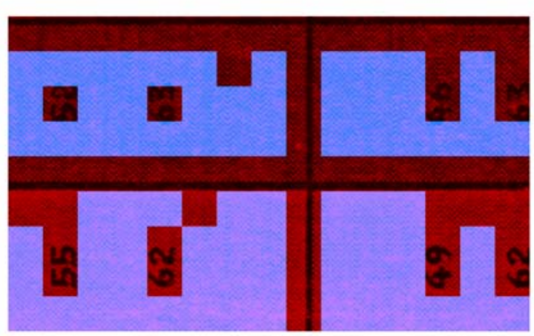

Figure 5.1 The foreground is shown highlighted (in red).

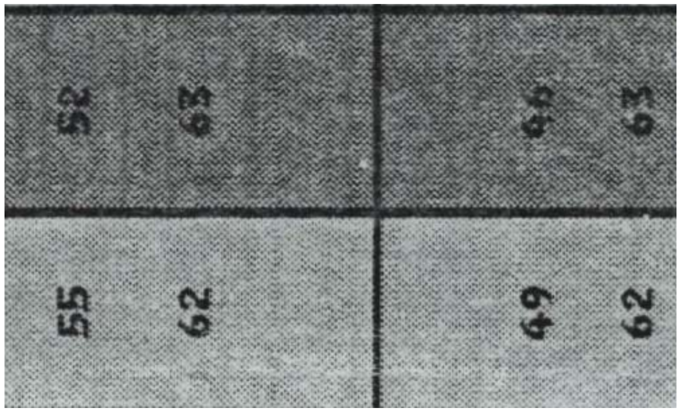

Figure 5.2 The original input image for the image shown in Fig. 5.1.

6. CONCLUSIONS

The system described in this paper looks at content-based image retrieval from a different perspective than usual. The main strength of our system is that it does not suffer the consequences of brightness or color normalization. Therefore, our results are robust to varying light conditions. We further extended our experiments to illustrate the advantages of our technique by conducting background-learning experiments and the results are encouraging.

The local texture representation proposed in this paper allows us to perform local brightness normalization using a reversible function. A given image it is divided into sub-images and its sub-image projected into *n*-dimensional texture vector in a space spanned by the principal components. The set of the obtained *n*-vectors forms a representation of each image. Hausdorff distance is then used to match these representations. Thus, we also discard classical approaches of representing images as histograms and histogram matching. Experimental results show that the proposed approach outperforms the classical image retrieval systems.

7. ACKNOWLEDGMENTS

We would like to thank Kishore Kulkarni for conducting some of our experimental results.

8. REFERENCES

1. J. Eakins, M. Graham, Content-based Image Retrieval: A report to the JISC Technology Applications Programme.
2. Remco C. Veltkamp, Mirela tanase, Danielle Sent, “Features in Content-based image retrieval systems: A Survey”, In Remco C. VeltKamp, Hans Burkhardt and Hans-Peter Kriegel [17].
3. J. P. Eakins. Towards intelligent image retrieval. Pattern Recognition 35, p. 3-14, 2002. (Special Issue on Shape Representation and Similarity for Image Databases, editors. L. J. Latecki, A. Gross, and R. Melter.)
4. M. J. Swain and D. H. Ballard, Color indexing, Int. J. on Computer Vision, vol. 7, no. 1, pp. 11-32, 1991.
5. Chad Carson, Megan Thomas, Sege Belongie, Joseph M. Hellerstein, and Jitendra Malik, “BlobWorld: A system for region-based image indexing and retrieval”, In Huijsmans and Smeulders [16].
6. Z.N. Li, O.R. Zaïane, and Z. Tauber. Illumination invariance and object model in content-based image and video retrieval. *Journal of Visual Communication and Image Representation*, 10(3):219-244, 1999.
7. J. Vendrig. *Filter image browsing: a study to image retrieval in large pictorial databases.* Master’s thesis, Dept. Computer Science, University of Amsterdam, http://carol.wins.uva.nl/~vendrig/thesis/
8. M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. S. Huang. Supporting similarity queries in MARS. In *Proc. of the 5th ACM Int. Multimedia Conf.*, Seattle, Washington, p. 403-413, 1997.
9. T. Gevers, A. W.M. Smeulders, “PicToSeek: Combining Color and Shape Invariant Features for Image Retrieval”, *IEEE Trans. on Image Processing*, Vol 9, No.1, Jan 2000.
10. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content using color, texture, and shape. In *Proc. of the SPIE Conf. on Storage and Retrieval for Image and Video Databases*, San Jose, CA, p. 173-187, 1993.

11. Chahab Nastar, Matthias Mitschke, Christophe Meilhac, and Nozha Boujemaa. Surfimage: a flexible content-based image retrieval system. In *Proc. of the ACM Int. Multimedia Conf.*, Bristol, England, p. 339-344, 1998.
12. J.Z. Wang, G. Wiederhold, O. Firschein, and S.X. Wei. Wavelet-based image indexing techniques with partial sketch retrieval capability. In *Proc. of the Fourth Forum on Research and Technology Advances in Digital Libraries*, Washington D.C., p. 13-24, 1997.
13. http://www.fon.hum.uva.nl/praat/manual/Principal_component_analysis.html.
14. Lindsay, I. Smith, “A Tutorial on Principal Component Analysis”.
15. D.P. Huttenlocher, G.A. Klanderman, W.J. Rucklidge, Comparing Images Using the Hausdorff Distance. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 15(9), p. 850-863, 1993.
16. D.P. Huijsmans and A.W. M. Smeulders, editors. *Proc. of the Third International Conf. on Visual Information and Information Systems (VISUAL)*, Amsterdam, Lecture Notes in Computer Science 1614. Springer, 1999.
17. Remco C. Veltkamp, Hans Burkhardt and Hans-Peter Kriegel, editors. *State of the Art in Content Based Image and Video Retrieval*, Vol. 22, pages 97-124, Kluwer Academic Publishers, 2001.
18. L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4:519-524, 1987.
19. M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Trans. on Pattern analysis and Machine Intelligence* 12(1):103-108, 1990.
20. M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.
21. J.-M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, and Hugo Geerts. Color Invariance. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23(12), p. 1338 - 1350, 2001.
22. A. Mustafa and I. K. Sethi. Creating Agents for Locating Images of Specific Categories. *Proc. of the IS&T/SPIE Conference Internet Imaging V*, Santa Clara, January 2004.
23. L.J. Latecki: Image Similarity Measures for Video Analysis. *Proc. of the IS&T/SPIE Conference Internet Imaging IV*, Vol. 5018, Santa Clara, pp. 218-227, January 2003.
24. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290, pp. 2319-2323, December 2000.