

HYPertext And MULTIMEDIA

Final Project

Content Based Image Retrieval System

Amine Ben Khalifa

Faezeh Tafazzoli

December 2012

Table of Contents

1	Abstract.....	1
2	Introduction.....	1
3	Image Features	1
3.1	Global vs. Local.....	1
3.2	Color	2
3.2.1	Color Descriptors.....	2
3.2.1.1	Color Histogram.....	2
3.2.1.2	Color Structure Descriptor (CSD)	2
3.2.1.3	Color Sketch.....	2
3.2.2	Color Space.....	2
3.3	Texture	3
3.3.1	Tamura Features.....	3
3.3.2	Gray Level Co-occurrence Matrix (GLCM).....	4
3.3.3	Wavelet-based Texture Features	4
3.3.4	Edge Histogram Descriptor (EHD)	4
3.4	SIFT	4
4	Query Matching	5
4.1	Dissimilarity Measurements.....	5
5	Performance Evaluation	5
6	Experimental Results.....	5
7	Conclusion	10
8	References	10

List Of Figures

Figure 1. System Framework 1

Figure 2. System user interface..... 6

Figure 3. Precision of local features 9

Figure 4. Recall of local features 9

Figure 5. F-measure of local features10

1 Abstract

Due to the enormous increase in image database sizes, as well as its vast deployment in various applications, many Content Based Image Retrieval (CBIR) systems have been developed. The challenge, however, is in designing a system with the ability to retrieve best matches in case of having all kind of images as query. The power of such system highly depends on the features it employs to perform the matching process.

In this project a CBIR system has been designed and developed. Firstly, this report outlines a description of some primitive features of image, which have been utilized in the presented system. These features are extracted and used as the basis for a similarity check between images. The algorithms used to calculate the similarity between extracted features, are then explained.

Final result was a Matlab built software application, with an image database, that utilized different feature of the images in the database as the basis of comparison and retrieval. The structure of the final software application is illustrated. Furthermore, the results of its performance are illustrated.

2 Introduction

Content-based image retrieval (CBIR) systems experience the challenge of semantic gap between the low-level visual features and the high-level semantic concepts. It would be advantageous to build CBIR systems which support high-level semantic query. The main idea is to integrate the strengths of content- and keyword-based image indexing and retrieval algorithms while alleviating their respective difficulties.

The CBIR basically performs two main tasks; firstly feature extraction, which is extracting feature set from the query image which is generally known as feature vectors which represents the content of each image in the database.

The second task is similarity measurement, which basically measures the distance between the query image and each image in database using the computed feature vectors and thus retrieves the closest match/matches.

We have implemented a CBIR system the details of which will be illustrated and analyzed in this report. The general framework of system is displayed in Figure. 1.

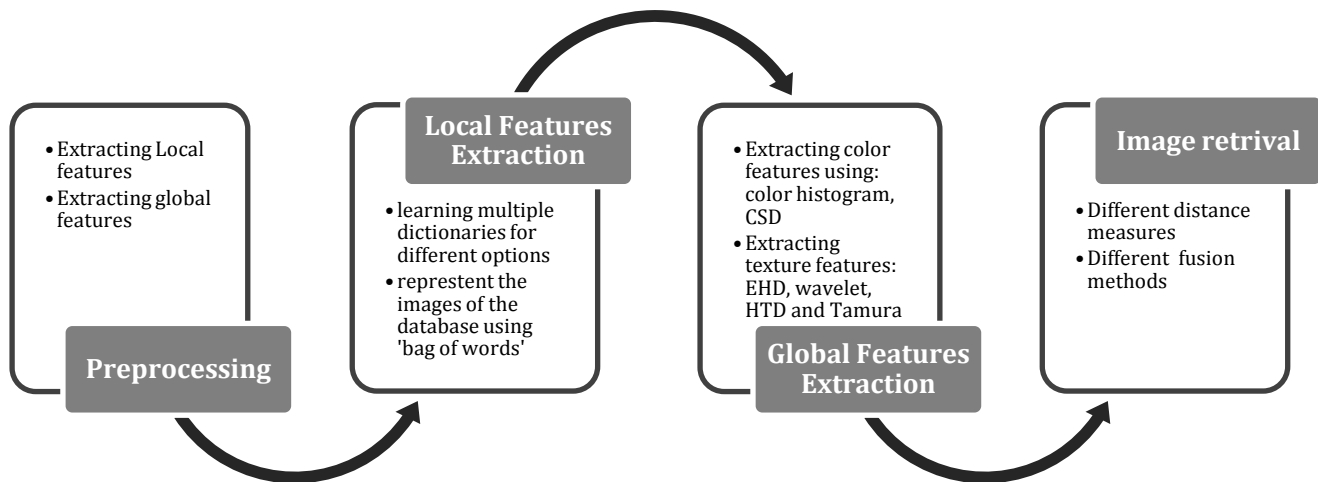


Figure 1. System Framework

3 Image Features

3.1 Global vs. Local

The global image descriptor is composed by color and texture features being computed on the entire image. The texture features are not always an accurate description of the image because they are computed on the whole image.

The local image description is founded on the premise that images can be characterized by attributes computed on regions of the image. Bag-Of-Visual-Words (BOVW) is a representation of images which is built using a large set of local features. They are inspired by the bag-of-words models in text retrieval, where a document is represented by a set of distinct keywords. Analogously, in BOVW models, an image is represented by a set of distinct visual words derived from local features [1].

3.2 Color

Color is an important cue for image retrieval. The image retrieval based on color features has proved effective for a large image database.

3.2.1 Color Descriptors

Color descriptors of images and video can be global and local. Global descriptors specify the overall color content of the image but with no information about the spatial distribution of these colors. Local descriptors relate to particular image regions and, in conjunction with geometric properties of these latter, describe also the spatial arrangement of the colors.

3.2.1.1 Color Histogram

A color histogram describes the distribution of colors within a whole image or video scene or within a specified region. As a pixel-wise characteristic, the histogram is invariant to rotation, translation, and scaling of an object. At the same time, the histogram does not capture semantic information, and two images with similar color histograms can possess totally different contents. A quantized HSI (or RGB) color space is typically used to represent the color in order to make the search partially invariant to irrelevant constraints such as illumination and object viewpoints.

A color histogram is a K -dimensional vector such that each component represents the relative number of pixels of color C_k in the image, that is, the fraction of pixels that are most similar to the corresponding representative color. To build the color histogram, the image colors should be transformed to an appropriate color space and quantized according to a particular codebook of the size K . By extracting the color histograms for image regions such as shown below, the spatial distribution of colors can be taken into account at least roughly because the dissimilarity of image colors is now measured by the weighted sum of the individual color dissimilarities between the corresponding regions.

3.2.1.2 Color Structure Descriptor (CSD)

The CSD uses the HMMD color space and an $m \times m$ structuring element to represent local color structure in an image by counting the number of times a particular color appears within the structuring element while the latter scans the image. Let C_0, C_1, \dots, C_{K-1} denote the K quantized colors. A *color structure histogram* has in each bin k the number of structuring elements in the image containing one or more pixels with color C_k . The bin values of the CSD are normalized by the number of locations of the structuring element and lie in the range $[0.0, 1.0]$. The normalized values are then nonlinearly quantized to 8 bits per bin.

3.2.1.3 Color Sketch

This feature segments each image to a $m \times m$ grid each cell of which will be analyzed separately. A normal Color Histogram is applied on each sub-image created by dividing the original image. The most frequent color of each cell will constitute a part of a 1 by m dimensional feature vector.

3.2.2 Color Space

Human color perception is quite subjective as regarding perceptual similarity. To design formal color descriptors, one should specify a color space, its partitioning, and how to measure similarity between colors.

A Color space is a multidimensional space of color components.

- **RGB**

Human color perception combines the three primary colors: red (R) with the wavelength $\lambda=700$ nm, green (G) with the wavelength $\lambda=546.1$ nm and blue (B) with the wavelength $\lambda=435.8$ nm. Any visible wavelength L is sensed as a color obtained by a linear combination of the three primary colors (R, G, B) with the particular weights $c_R(L)$, $c_G(L)$, $c_B(L)$:

$$F(L) = R c_R(L) + G c_G(L) + B c_B(L)$$

- **HSV**

HSV (hue - saturation - value) space is obtained by non-linear transformation of the RGB space. The HSV representation uses the brightness (or intensity) value $I = (R + G + B)/3$ as the main axis orthogonal to the chrominance plane. The saturation S and the hue H are the radius and the angle, respectively, of the polar coordinates in the chrominance plane with the origin in the trace of the value axis (with R corresponding to 0°). This representation is approximately perceptually uniform and is closely related to the way the human vision perceives color images. Because of invariance to the object orientation with respect to illumination and camera viewing direction, the hue is more suitable for object retrieval.

3.3 Texture

Texture is that innate property of all surfaces that describes visual patterns, each having properties of homogeneity. It contains important information about the structural arrangement of the surface. It also describes the relationship of the surface to the surrounding environment.

Texture features are usually compared on the basis of dissimilarity between the two feature vectors. The dissimilarity is given by the Euclidean, Mahalanobis, or city-block distance. In some cases, the weighted distances are used where the weight of each vector component is inversely proportional to the standard deviation of this feature in the database.

The texture features used in the presented system are briefly described as follows.

3.3.1 Tamura Features

Tamura features proposed in [2], are six texture features corresponding to human visual perception: coarseness, contrast, directionality, line-likeness, regularity, and roughness. In [2], experiments performed to test the significance of the features showed that the first three features, which will be described as follows, are very important.

- **Coarseness**

The coarseness gives information about the size of the texture elements. The higher the coarseness value is, the rougher is the texture. The essence of calculating the coarseness value is to use operators of various sizes. The coarseness measure is calculated as follows

1. For every pixel the average over neighborhoods, with size of powers of two, is calculated:

$$A_k(n_0, n_1) = \frac{1}{2^{2k}} \sum_{i=1}^{2^{2k}} \sum_{j=1}^{2^{2k}} X(n_0 - 2^{k-1} + i, n_1 - 2^{k-1} + j)$$

2. For every pixel the differences between the not overlapping neighborhoods on opposite sides of the point in horizontal and vertical direction will be calculated:

$$\begin{aligned} E_k^h(n_0, n_1) &= |A_k(n_0 + 2^{k-1}, n_1) - A_k(n_0 - 2^{k-1}, n_1)| \\ E_k^v(n_0, n_1) &= |A_k(n_0, n_1 + 2^{k-1}) - A_k(n_0, n_1 - 2^{k-1})| \end{aligned}$$

3. At each point, the size leading to the highest difference value will be selected:

$$S(n_0, n_1) = \arg\max_{k=1..5} \max_{d=h,v} E_k^d(n_0, n_1)$$

4. Finally take the average over 2^S will be considered as the coarseness measure for the image:

$$F_{crs} = \frac{1}{N_0 N_1} \sum_{n_0=1}^{N_0} \sum_{n_1=1}^{N_1} 2^{S(n_0, n_1)}$$

- **Contrast**

In the narrow sense, contrast stands for picture quality. More detailed, contrast can be considered to be influenced by the following four factors: dynamic range of gray-levels, polarization of the distribution of black and white on the gray-level histogram, sharpness of edges and period of repeating patterns. The contrast of an image is calculated by:

$$F_{con} = \frac{\sigma}{\alpha_z^4} \quad \text{with} \quad \alpha_4 = \frac{\mu_4}{\sigma^4}$$

where μ_4 is the fourth moment about the mean, and z has experimentally been determined to be 0.25.

- **Directionality**

Not the orientation itself but presence of orientation in the texture is relevant here. That is, two textures differing only in the orientation are considered to have the same directionality. To calculate the directionality the horizontal and vertical derivatives Δ_H and Δ_V are calculated, and then for every pixel (n_0, n_1) the following formula will be calculated.

$$\theta = \frac{\pi}{2} + \tan^{-1} \frac{\Delta_V(n_0, n_1)}{\Delta_H(n_0, n_1)}$$

These values are then histogramized in a 16-bin histogram and the directionality can be calculated as the sum of second moments around each peak from valley to valley.

3.3.2 Gray Level Co-occurrence Matrix (GLCM)

The identification of specific textures in an image is achieved primarily by modeling texture as a two-dimensional gray level variation. This two dimensional array is called as Gray Level Co-occurrence Matrix (GLCM). The GLCM is computed in four directions for $\delta=00, \delta=45, \delta=90, \delta=135$. Based on the GLCM four statistical parameters energy, contrast, entropy and correlation are computed. Finally a feature vector is computed using the means and variances of all the parameters [3].

3.3.3 Wavelet-based Texture Features

Discrete wavelet transformation (DWT) is used to transform an image from spatial domain into frequency domain. Wavelet transforms extract information from signal at different scales by passing the signal through low pass and high pass filters. Wavelets are robust with respect to color intensity shifts and can capture both texture and shape information efficiently. The wavelet transforms can be computed linearly with time and thus allowing for very fast algorithms.

The wavelet transform computation of a two-dimensional applies recursive filtering and sub-sampling. At each level (scale), the image is decomposed into four frequency sub-bands, LL, LH, HL, and HH where L denotes low frequency and H denotes high frequency [4].

3.3.4 Edge Histogram Descriptor (EHD)

The edge histogram descriptor resembles the color layout descriptor (CLD) in its principle of capturing the spatial distribution of edges which is useful in image matching even if the texture itself is not homogeneous. An image is partitioned into $4 \times 4 = 16$ sub-images, and 5-bin local edge histograms are computed for these sub-images, each histogram representing five broad categories of vertical, horizontal, 45°-diagonal, 135°-diagonal, and isotropic (non-orientation specific) edges. The resulting scale-invariant descriptor is of size 240 bits, i.e. $16 \times 5 = 80$ bins and supports both rotation-sensitive and rotation-invariant matching.

The edge histograms are computed by subdividing each of the 16 sub-images into a fixed number of blocks. The size of these blocks depends on the image size and is assumed to be a power of 2. To have the constant number of blocks per sub-image, their sizes are scaled in accord with the original image dimensions. Each block is then treated as 2×2 pixel image (by averaging each of the 2×2 partitions), and a simple edge detector is applied to these average values. The detector consists of four directional filters and one isotropic filter.

Five edge strengths, one for each of the five filters, are computed for each image block. If the maximum of these strengths exceeds a certain preset threshold, the corresponding image block is an edge block contributing to the edge histogram bins. The bin values are normalized to the range [0.0, 1.0].

3.4 SIFT

Scale Invariant Features Transform (SIFT) [5] is a powerful framework to recognize/retrieve objects. This approach can be viewed as a texture descriptor composed by four major stages:

- Scale-space extrema detection: identifies locations and scales that can be repeatedly assigned under differing views of the same object. Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space.
- Keypoint localization: each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below. It is selected only if it is larger than all of these neighbors or smaller than all of them.

- Orientation assignment: By assigning a consistent orientation to each keypoint based on local image properties, its feature vector can be represented relative to this orientation and therefore achieve invariance to image rotation.
- Keypoint description.

4 Query Matching

To retrieve desired images, user has to provide a query image. The system then performs certain feature extraction procedures on it and represents it in the form of feature vectors. The similarity distances between the feature vectors of the query image and those of the images in the database are then calculated and retrieval is performed with the help of indexing schemes. The indexing scheme provides an efficient way to search for the image database.

4.1 Dissimilarity Measurements

Dissimilarity measurement plays a crucial role in content-based image retrieval. There are different options for feature vector comparison or in other words, measuring histogram cross-bin distances. Dissimilarity measures are classified into three categories according to their theoretical origins [7].

Minkowsky Family:

Kullback-Leibler (K-L) Divergence:

χ^2 Statistics:

Kolmogorov-Smirnov:

$$\begin{aligned} & (\sum_{i=1}^n |v_i - w_i|^p)^{1/p} \\ & \sum_{i=1}^n v_i \log \frac{v_i}{w_i} \\ & \sum_{i=1}^n \frac{(v_i - m_i)^2}{m_i} \\ & \max_{1 \leq i \leq n} |F_v(i) - F_w(i)| \end{aligned}$$

5 Performance Evaluation

The performance of retrieval of the system can be measured in terms of its recall and precision. Recall measures the ability of the system to retrieve all the models that are relevant, while precision measures the ability of the system to retrieve only the models that are relevant. It has been reported that the histogram gives the best performance through recall and precision value. They are defined as:

$$\begin{aligned} \text{Precision} &= \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}} = \frac{A}{A + B} \\ \text{Recall} &= \frac{\text{Number of relevant images retrieved}}{\text{Total number of relevant images}} = \frac{A}{A + C} \end{aligned}$$

Where A represent the number of relevant images that are retrieved, B, the number of irrelevant items and the C, number of relevant items those were not retrieved. The number of relevant items retrieved is the number of the returned images that are similar to the query image in this case. The total number of items retrieved is the number of images that are returned by the search engine.

6 Experimental Results

One of the most important components of CBIR systems is the user interface (UI). A CBIR system should offer easy and optimized ways first to select the query image, second to pick the retrieval options, and finally it should offer a way to visualize the results in an interpretable way. Much like Google “search by image” is doing: you can drag and drop images to the search box and you can easily brows tons and tons of search results just by scrolling down the window.

For the sake of simplicity and debugging we decided to design our UI as a Matlab GUI. The main interface of the built software is illustrated in the following figure.

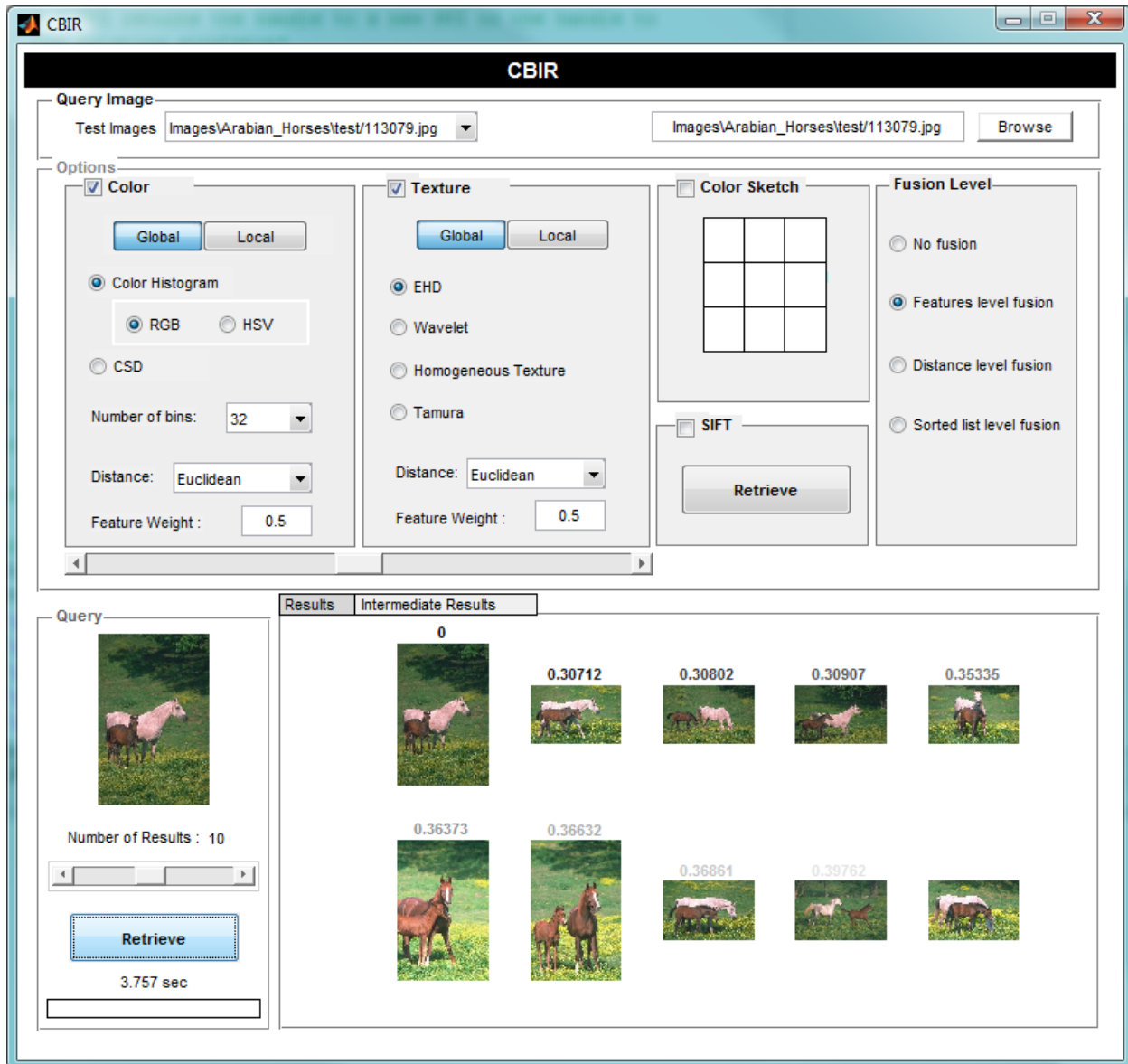





Figure 2. System user interface

The UI offers the possibility to select a testing query right from the database or to upload new –never seen- query. The UI has options to select the features, distance measures and fusion levels for retrieval. A user can use slide bars to change the weight of a given feature or simply increase the number of retrieved images. Moreover when the fusion get a bit complex –such sorted list level fusion- the user can switch between the final results and intermediate results with just a click of a button, we believe such feature help improve the user experience and give more insight into how the final list is generated. Furthermore the color sketch pad allow the user to query using custom color distribution, this is very helpful when one know what color the results should have (e.g. the user wants to retrieve more bluish images that goes perfectly with the colors of his living room for example).




In the following we present some of the best search results. Visually the results using Global features look more pure. To be sure which local features would give the best results, we generated performance measures for all the possible local options (64 options). From the F-measure presented at Figure 5 Local feature generated using the concatenated EHD and Color Histogram with 32 bins and color space RGB gave the best results. We also believe that the Global features did better because it is hard for the local method to capture all the keywords present at the 2000

images, especially we have used dictionary with only 200 dimensions. However going beyond 200 would certainly hurt distance computation and make the system very slow.




Global : Color Hist, 32, RGB, + EHD + Feature level fusion

Query	Results	Intermediate Results
 <p>Number of Results : 10</p> <p>Retrieve</p> <p>3.757 sec</p>		<p>0</p> <p>0.30712</p> <p>0.30802</p> <p>0.30907</p> <p>0.35335</p>
		<p>0.36373</p> <p>0.36632</p> <p>0.36861</p> <p>0.39762</p>

Local : Color Hist, 32, RGB, + EHD + no fusion


Query	Results	Intermediate Results
 <p>Number of Results : 10</p> <p>Retrieve</p> <p>4.539 sec</p>		<p>0</p> <p>0.23895</p> <p>0.24533</p> <p>0.25459</p> <p>0.26644</p>
		<p>0.26644</p> <p>0.27217</p> <p>0.27217</p> <p>0.27459</p>

Local : Color Hist, 32, RGB, + EHD + Feature level fusion

Query	Results	Intermediate Results
 <p>Number of Results : 10</p> <p>Retrieve</p> <p>6.386 sec</p>		<p>0</p> <p>0.1853</p> <p>0.18634</p> <p>0.1884</p> <p>0.18942</p>
		<p>0.19043</p> <p>0.19543</p> <p>0.19642</p> <p>0.19837</p>

Global: Color Hist, 32, RGB, + EHD + Distance level fusion

Query



Number of Results : 10

Retrieve


3.200 sec

Results Intermediate Results

Results	Intermediate Results
0	0.00012747 0.00015981 0.00016871 0.00017115
0.00017166 0.00017312 0.00017398 0.00018278	

Global: Color Hist, 32, RGB, + EHD + Sorted List level fusion

Query



Number of Results : 10

Retrieve

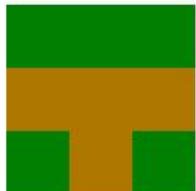
3.613 sec

Results Intermediate Results

Results	Intermediate Results
2	18 38 60 61
69 73 77 82	

Color Sketch, Global: Color Hist, 32

Query



Number of Results : 10

Retrieve

2.068 sec

Results Intermediate Results

Results	Intermediate Results
0.27065 0.27304 0.30074 0.30277 0.30758	
0.3187 0.33113 0.33473 0.33489	

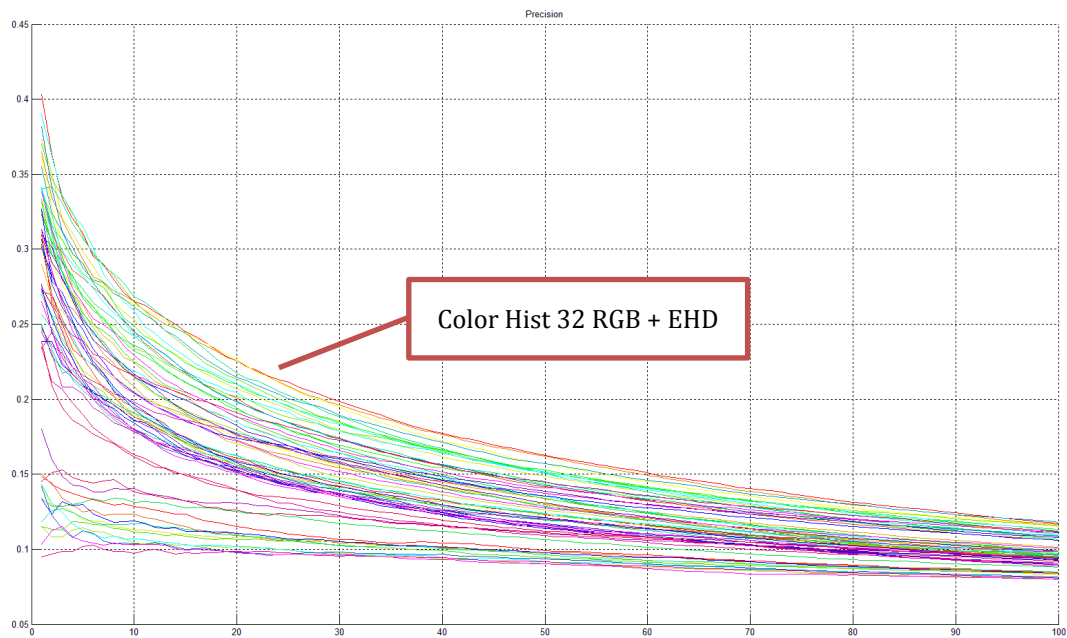


Figure 3. Precision of local features

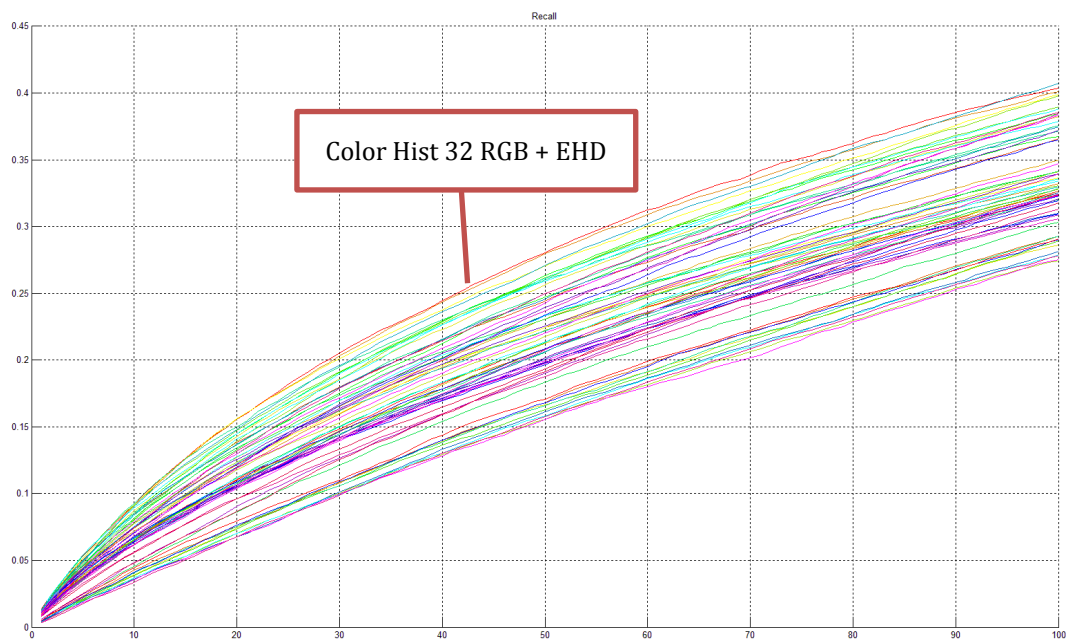


Figure 4. Recall of local features

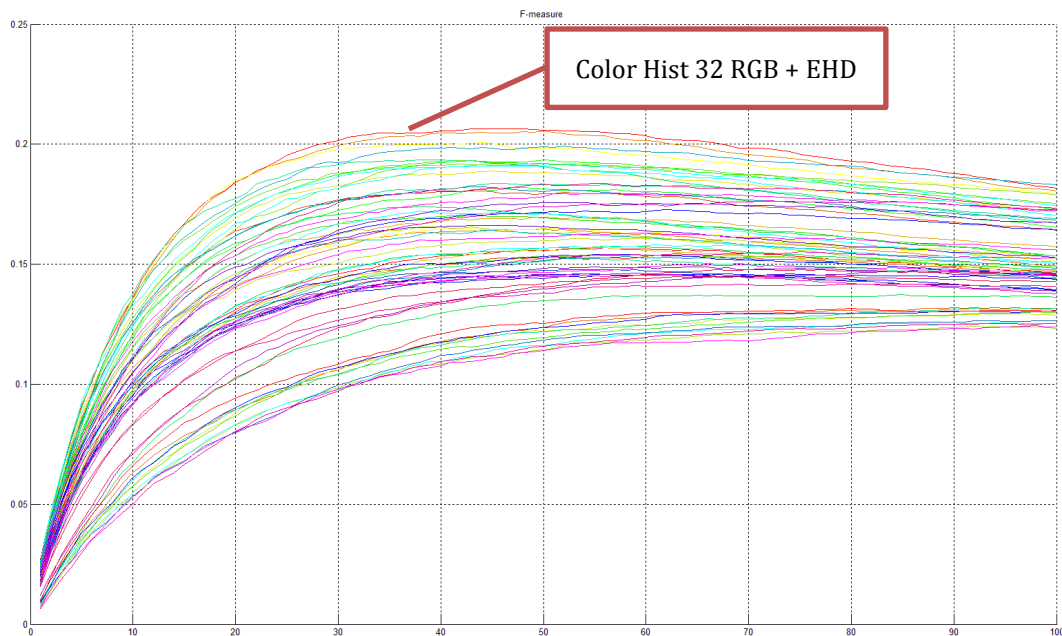


Figure 5. F-measure of local features

7 Conclusion

We have presented a system for searching and retrieving images in this project. Regarding the “huge” size of the database we can say that our system provided good results. Using more performance measures we can certainly fine tune more features and possibly provide the users with the best options of retrieval as default parameters, much like Google search by image is doing, the features, fusion options and distance measures are all hidden but set in away the search results will be as relevant as possible to the query. As future work we are planning to measure the performance of more options, and offer 3D visualization of the search results.

8 References

- [1] K. ZAGORIS, S. CHATZICHRISTOFIS, AND A. ARAMPATZIS. “BAG-OF-VISUAL-WORDS VS. GLOBAL IMAGE DESCRIPTORS ON TWO-STAGE MULTIMODAL RETRIEVAL”. 34TH INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, PP. 1251-1252 2011
- [2] H. TAMURA, S. MORI, T. YAMAWAKI. “TEXTURAL FEATURES CORRESPONDING TO VISUAL PERCEPTION”. IEEE TRANSACTION ON SYSTEMS, MAN, AND CYBERNETCS, VOL. SMC-8, NO. 6, PP. 460-472, JUNE 1978
- [3] H.B. KEKRE, S. D. THEPADE, T. K. SARODE AND V. SURYAWANSHI. “IMAGE RETRIEVAL USING TEXTURE FEATURES EXTRACTED FROM GLCM, LBG AND KPE”. INTERNATIONAL JOURNAL OF COMPUTER THEORY AND ENGINEERING, VOL. 2, NO. 5, OCTOBER, 2010
- [4] H. LIN, C. CHIU, S. YANG. “FINDING TEXTURES BY TEXTUAL DESCRIPTIONS, VISUAL EXAMPLES, AND RELEVANCE FEEDBACKS”. PATTERN RECOGNITION LETTERS, VOL. 24, NO. 14, PP. 2255-2267, JANUARY 2003
- [5] D. G. LOWE. “DISTINCTIVE IMAGE FEATURES FROM SCALE-INVARIANT KEYPOINTS”. INTERNATIONAL JOURNAL OF COMPUTER VISION, VOL. 60, NO. 2, PP. 91-110, 2004

- [6] M. SINGHA AND K. HEMACHANDRAN. "CONTENT BASED IMAGE RETRIEVAL USING COLOR AND TEXTURE". SIGNAL & IMAGE PROCESSING: AN INTERNATIONAL JOURNALS (SIPIJ), VOL. 3, NO. 1, FEBRUARY, 2012
- [7] R. HU, S. RÜGER, D. SONG, AND H. LIU. "DISSIMILARITY MEASURES FOR CONTENT-BASED IMAGE RETRIEVAL". IEEE INTERNATIONAL CONFERENCE MULTIMEDIA AND EXPO, HANNOVER, GERMANY JUNE 2008