INFORMATION SEARCH USING TEXT AND IMAGE QUERY

Zaware Sarika Nitin[1]

[1]*Assistant Professor, Computer Engineering Department, AISSMS IOIT, Maharashtra, India,* ***sarika_k_99@yahoo.com***

Abstract

An image retrieval and re-ranking system utilizing a visual re-ranking framework which is proposed in this paper the system retrieves a dataset from the World Wide Web based on textual query submitted by the user. These results are kept as data set for information retrieval. This dataset is then re-ranked using a visual query (multiple images selected by user from the dataset) which conveys user's intention semantically. Visual descriptors (MPEG-7) which describe image with respect to low-level feature like color, texture, etc are used for calculating distances. These distances are a measure of similarity between query images and members of the dataset. Our proposed system has been assessed on different types of queries such as apples, Console, Paris, etc. It shows significant improvement on initial text-based search results.This system is well suitable for online shopping application.

Index Terms*: MPEG-7, Color Layout Descriptor (CLD), Edge Histogram Descriptor (EHD), image retrieval and re-ranking system*

----------***----------

1. INTRODUCTION

Image search engines are implemented using the "query by keyword" paradigm which index and search the associated textual information of images. Here image retrieval is based on how contents of an image or a chain of images can be represented. Conventional techniques of text data retrieval can be applied only if every image and video record is accompanied with a textual content description. But image or video content is much more versatile compared to text, and in the most cases the query topics are not reflected in the textual metadata available.
Visual reranking [1] is an integrated framework that helps to obtain effective search results. Visual reranking incorporates both textual and visual indications. A list of text-based search results is first returned by using textual information. The text-based search result provides a good baseline for the "true" ranking list which may be noisy but the text-based search result still reflect partial facts of the "true" list and are used for reranking. Then visual information is applied to reorder the initial result for refinement.

The visual cues are obtained by using a QBE or Query by Example paradigm. In a QBE framework the color, texture, shape, or other features of the query image, extracted and stored as metadata, are matched to the image metadata in the dataset of indexed images and returned results are based on matching scores.

However single query may not sufficiently represent user's intention. Hence it might be desirable to query an image dataset using more than one query images for detailed knowledge representation. A multi-query retrieval technique which searches each query individually and then merges the results of each query afterwards into a synthetic list has been proposed [2]. It returns semantically related images in different visual clusters by merging the result sets of multiple queries. For a given retrieval task, the user may pick different queries, which are all semantically related to the images the user desires.

These queries will generate different retrieval results by the same CBIR system. These different result lists can be thought of as different viewpoints regarding the retrieval task in user's mind.

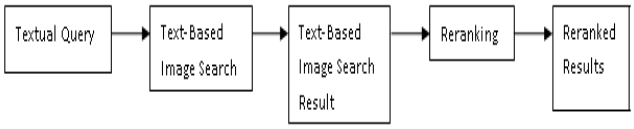

Figure 1: Visual Reranking Framework

We use MPEG7 [3] standard to generate low level visual descriptors. MPEG-7 is a standard developed by Moving Picture Experts Group (MPEG) [4]. It was formally named as Multimedia Content Description Interface. The goal of the content description (i.e., metadata) standard MPEG-7 is to enable fast and efficient searching, filtering, and adaptation of multimedia content. In MPEG-7 Visual features [5] related to semantic contents are represented by the following descriptors:

- the colour descriptors - colour space, colour quantization, dominant colours, scalable colour, colour-structure, colour layout, and group of frames / group of pictures colour descriptor;
- the texture descriptors - homogeneous texture, texture browsing, and edge histogram;
- the shape descriptors - object region-based shape, contour based shape, etc

We combine color and texture features to generate accurate results for image retrieval. We use Color Layout Descriptor (CLD) as our color and Edge Histogram Descriptor (EHD) as texture descriptor [6]. CLD represents the spatial distribution of colors in an image and EHD describes edge distribution with a histogram based on local edge distribution in an image [6]. The visual features of each sub-image can be characterized by the representative colors of the CLD as well as the edge histogram of the EHD at that sub-image using weighing factors.

2. COLOUR LAYOUT DESCRIPTOR

CLD [5] is a very compact and resolution-invariant representation of color for high-speed image retrieval and it has been designed to efficiently represent the spatial distribution of colors.

A. Extraction

The extraction process of this color descriptor consists of four stages: image partitioning, representative color detection, DCT transformation and a zigzag scanning. Moreover, as the images used during the realization of this project were defined on the RGB color space, a stage of color space conversion was added, as the standard MPEG-7 recommends to use the YCbCr color space for the CLD.

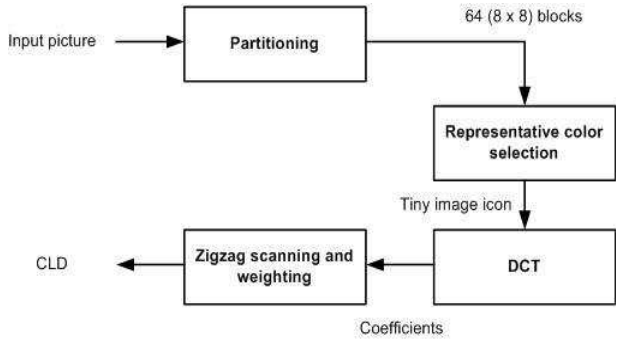

Figure 2: Color Layout Descriptor

The CLD descriptor was obtained through the following steps:

1. The image is loaded and the width and height of the image are obtained, from which the block width and block height of the *CLD* were calculated by dividing by 8. The division was done using truncation, so that if the image dimensions were not divisible by 8, the outermost pixels are not considered in the descriptor. In the image partitioning stage, the input picture (on RGB color space) is divided into 64 blocks to guarantee the invariance to resolution or scale.
2. The image partitioning stage, a single representative color is selected from each block by the use of the average of the pixel colors in a block as the corresponding representative color, which results in a tiny image icon of size 8x8.
3. Once the tiny image icon is obtained, the color space conversion between RGB and YCbCr is applied. This conversion is defined by a linear transformation of the RGB color space:

In the fourth stage, the luminance (Y) and the blue and Red chrominance (Cb and Cr) are transformed by 8x8 DCT, so three sets of 64 DCT coefficients are obtained.

Luminance: $Y = 0.29*R + 0.59*G + 0.114*B - 128$

Blue chrominance: $Cb = 0.169*R - 0.331*G + 0.5*B$

Red chrominance: $Cr = 0.5*R - 0.419*G - 0.081*B$

The general equation for a 2D (N by M image) DCT is defined by the following equation:

$$F(u,v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Lambda(i).\Lambda(j).\cos\left[\frac{\pi.u}{2.N}(2i+1)\right] \cos\left[\frac{\pi.v}{2.M}(2j+1)\right].f(i,j)$$

Where

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for} \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

a. The input image is N by M;
b. f(i,j) is the intensity of the pixel in row i and column j;
c. F (u,v) is the DCT coefficient in row k1 and column k2 of the DCT matrix.

1. A zigzag scanning is performed with these three sets of 64 DCT coefficients which groups the low frequency coefficients of the 8x8 matrix. Finally, these three set of matrices correspond to the CLD of the input image.

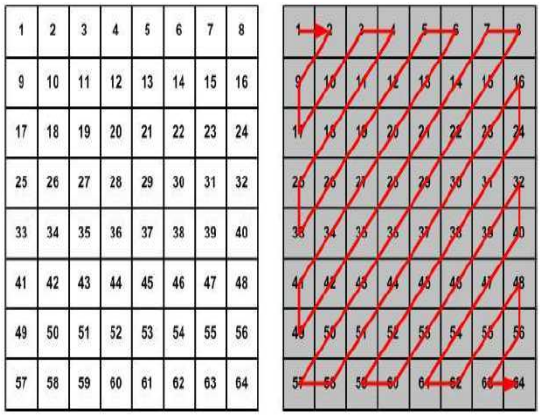

Figure 3: ZigZag Co-Efficients

B. To Calculate Distances Between Two Images:

$$D = \sqrt{\sum_i w_{yi}(DY_i - DY_i')^2} + \sqrt{\sum_i w_{bi}(DCb_i - DCb_i')^2} + \sqrt{\sum_i w_{ri}(DCr_i - DCr_i')^2}$$

Where:

a. "i" represents the zigzag-scanning order of the coefficients.

b. DY, DCb, DCr represent input image and DY', DCr', DCb' represent another image.

3. EDGE HISTOGRAM DESCRIPTOR

Edge in the image is considered an important feature to represent the content of the image. Human eyes are known to be sensitive to edge features for image perception. In MPEG-7, there is a descriptor for edge distribution in the image [7]. This edge histogram descriptor proposed for MPEG-7 consists only of local edge distribution in the image.

A. Partition of Image Space

To localize edge distribution to a certain area of the image, we divide the image space into 4x4 sub-images. Then, for each sub-image, we generate an edge histogram to represent edge distribution in the sub-image. To define different edge types, the sub-image is further divided into small square blocks called image-blocks. Regardless of the image size, we divide the sub-image into a fixed number of image-blocks i.e., the size of the image-block is proportional to the size of original image to deal with the images with different resolutions. Equations (1) and (2) show how to decide the size of the image-block for a given image with image_width*image_height. The size of image-block is assumed to be a multiple of 2. If it is not a multiple of 2, we can simply ignore some outmost pixels so that the image-block becomes a multiple of 2.

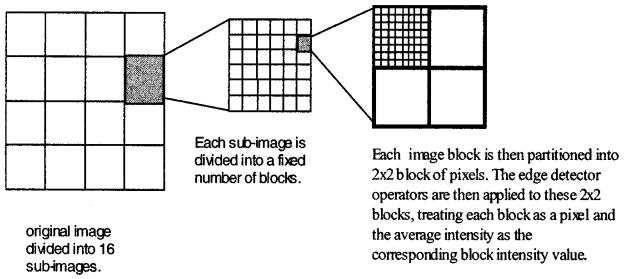

Figure 4: Partition of Image Space

$$x = \sqrt{\frac{image\_width \times image\_height}{desired\_num\_block}} \quad (1)$$

$$block\_size = \left\lfloor \frac{x}{2} \right\rfloor \times 2 \quad (2)$$

B. Edge Types

Five edge types are defined in the edge histogram descriptor. They are four directional edges and a non-directional edge. Four directional edges include vertical, horizontal, 45 degree, and 135 degree diagonal edges. These directional edges are extracted from the image-blocks. If the image-block contains an arbitrary edge without any directionality, then it is classified as a non-directional edge.

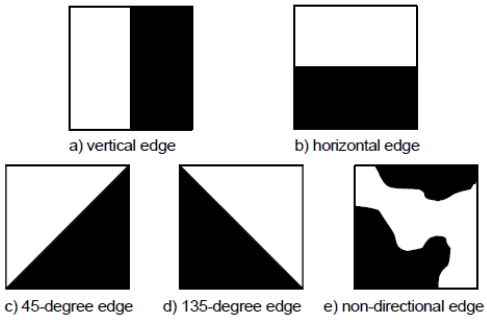

Figure 5: Five types of Edges

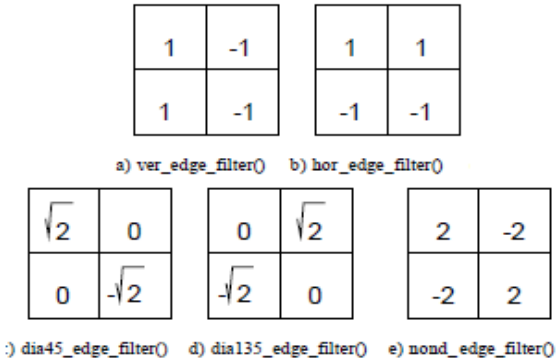

Figure 6: Filters for Edge Detection

Edge feature is extracted from the image-block. Here, the image-block is further divided into four sub-blocks. Then, the luminance mean values for the four sub-blocks are used for the edge detection. More specifically, mean values of the four sub-blocks are obtained, and they are convolved with filter coefficients to obtain edge magnitudes. For the kth (k=0,1,2,3) sub-block of the (i, j)th image block, we can calculate the average gray level .

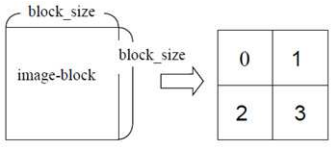

Figure 7: Image Sub-Block

By using equations (3) - (7), we can obtain directional edge strengths.

$$ver\_edge\_stg(i,j) = \sum_{k=0}^{3} \left| A_k(i,j) \times ver\_edge\_filter(k) \right| \quad (3)$$

$$hor\_edge\_stg(i,j) = \sum_{k=0}^{3} \left| A_k(i,j) \times hor\_edge\_filter(k) \right| \quad (4)$$

$$dia45\_edge\_stg(i,j) = \sum_{k=0}^{3} \left| A_k(i,j) \times dia45\_edge\_filter(k) \right| \quad (5)$$

$$dia135\_edge\_stg(i,j) = \sum_{k=0}^{3} \left| A_k(i,j) \times dia135\_edge\_filter(k) \right| \quad (6)$$

$$nond\_edge\_stg(i,j) = \sum_{k=0}^{3} \left| A_k(i,j) \times nond\_edge\_filter(k) \right| \quad (7)$$

If the maximum value among five edge strengths obtained from equations (3) to (7) is greater than a threshold (Thedge) , then the image-block is considered to have the corresponding edge in it. For our experiments, we set the total number of image-blocks at 1100 and the threshold for edge detection (Thedge) at 11.

C. Semantics of Local Edge Histogram

After the edge extraction from image-blocks, we count the total number of edges for each edge type in each sub-image. Since there are five different edges, we can define five histogram bins for each sub-image. Then, since there are 4x4=16 sub-images, we have total 16x5=80 bins for the edge histogram. The semantics of the bins are defined as in Table 1. These 80 bin values are non-linearly quantized and fixed length coded with 3 bits/bin as suggested by Won.[7].

Table 1: Bin semantics

HistogramBins	Semantics
BinCount[0]	Vertical Edge of sub-image at (0,0)
BinCount[1]	Horizontal Edge of sub-image at (0,0)
BinCount[2]	45 degree Edge of sub-image at (0,0)
BinCount[3]	135 degree Edge of sub-image at (0,0)
BinCount[4]	Non-Directional Edge of sub-image at (0,0)
:	
BinCount[75]	Horizontal Edge of sub-image at (3,3)
BinCount[76]	45 degree Edge of sub-image at (3,3)
BinCount[77]	135 degree Edge of sub-image at (3,3)
BinCount[78]	Non-Directional Edge of sub-image at (3,3)
BinCount[79]	Vertical Edge of sub-image at (3,3)

The 80 bins of the local edge histogram in Table 1 (i.e., BinCounts[i], i=0,…79) are the only normative semantics for the EHD. For the similarity matching, we calculate the distance D(A,B) of two image histograms A and B using the following measure:

$$D(A,B) = \sum_{i=0}^{79} \left| Local\_A[i] - Local\_B[i] \right|$$

Where Local_A[i] represents the reconstructed value of BinCount[i] of image A and Local_B[i] is the reconstructed value of BinCount[i] of image B.

4. COMBINING CLD AND EHD

Relevance between two images is measured as a distance [8] between the two images in multiple feature spaces i.e. color and texture.

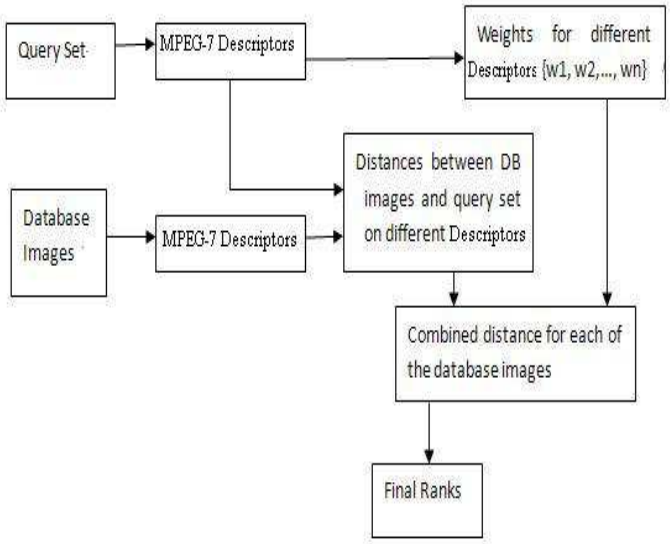

Figure 8: Retrieval procedure

Let f1 and f2 be visual features to describe the images and feature and w = (w1,w2) represents the weights[9] of the features, where Fi=1 wi =1 and Ir is the image to be compared.

1. Calculate the distances between I_r and q_i, i = 1, 2, ..., n with respect to feature f_j , j = 1, 2, ... , F

2. Combine distances with respect to individual features according to their importance, w_j , to form the overall distance d_r between I_r and Q.
   $\mathbf{d_r} = \sum \mathbf{w_j} * \mathbf{d_r^j}$ where , j = 1, 2, ... , F

In retrieval, images in the dataset, Ir, r = 1, 2,., N are ranked according to their distances, dr, to the query set Q and the top K images are output as the retrieval result.

In our experiments we set the weight of EHD distance to 0.5 and the weight of CLD to 0.5. Then we retrieve using the combined distances. This approach allows us to retain the positive points of both multiple queries and their features.

5. EXPERIMENTAL RESULTS

For our experiments we have used datasets generated by Bing Image Search. We have used queries of different types like apples (rigid object), Paris (ambiguous query) and Console (ambiguous query). Each of the original dataset has noisy images which are filtered out by our image retrieval and re-ranking system.

For each query an initial dataset created by Bing Image Search (top 100) are displayed to user. From this dataset user has to select 3 images which he feels are most relevant to his query thus capturing his intention. This dataset is then reranked according to low level visual features like color and texture. From this reranked list only top 50 images are displayed to user.

Figure 9: Apple_Original

As we can the images numbered 13, 14,17,18,19,22,23,24 are pushed below top 25 when images 1, 2, 3 are given as query set.

Figure 10: Apple_Reranked

Figure 11: Paris_Original

As we can see in Paris dataset, images 7, 10,11,12,13, 16, 17, 18, 19, 22 are pushed below top 25 when 1, 2, 6 are given as query images.

Figure 12: Paris_Reranked

Figure 13: Console_Original

As we can see in Console dataset, where the intention of user is a console table, irrelevant images are pushed below top 25 when 1, 2, 24 are given as query images

Figure 14: Console_Reranked

CONCLUSIONS

In our paper, an image retrieval and re-ranking system has been proposed. A single query image can hardly provide all the information about the image category, and the use of multiple images as the query can reflect the attributes of target image category. Hence we have proposed a multiple query based image retrieval system which can capture user's intention. Our system uses a set of low-level features called visual descriptors that have been defined by MPEG-7. Thus, the implemented tool is based on a standard, an effort to increase its reliability.

These visual descriptors have been classified depending on features such as color, texture, etc. Then individual distances for each feature for each image from query set are calculated and assigned weights based on their importance. This method takes advantage of the idea that using a set of descriptors leads to some better results than the ones obtained using an isolated descriptor. Given two images from which a visual descriptor has been obtained, this distance gives us an idea of how similar these images are according to the extracted feature. Thus, these dissimilarity measures allow obtaining a quantitative result that can be used for sorting out the target images objectively.

Thus our proposed system helps to improve the overall performance of image search engine by implementing a Visual Reranking based framework which takes into consideration the

textual query to generate initial dataset and then re-ranks images based on low level visual features.

REFERENCES

[1] X. Tian , L. Yang , J. Wang , X. Wu and X.-S. Hua "Bayesian visual reranking", IEEE Trans. Multimedia, vol. 13, no. 4, pp.639 -652, 2011

[2] X. Jin and J. C. French "Improving image retrieval effectiveness via multiple Queries" In MMDB '03: Proceedings of the 1st ACM international workshop on Multimedia databases, pages 86–93, New York, NY, USA, 2003. ACM Press

[3] T. Sikora, "The MPEG-7 visual standard for content description - an overview," IEEE Trans on Circuits and System for Video Technology, vol. 11, no. 6, June 2001

[4] http://mpeg.chiariglione.org/standards/mpeg-7

[5] B. S. Manjunath, P.Salembier, and T. Sikora, "Introduction to MPEG-7, Multimedia Content Description Interface", John Wiley and Sons, Ltd., Jun 2002

[6] B. S. Manjunath, Jens-Rainer Ohm, Vinod V. Vasudevan, Member, IEEE, and Akio Yamada "Color and Texture Descriptors" , Ieee Transactions On Circuits And Systems For Video Technology, VOL. 11, NO. 6, JUNE 2001

[7] D. K. Park, Y. S. Jeon, and C. S.Won," Efficient use of local edge histogram descriptor", ACM, 2000, pp. 51–54.

[8] Yuan Zhong, Lei Ye, Wanqing Li, Philip Ogunbona, "Perceived similarity and visual descriptions in content-based image retrieval", University of Wollongong Research Online

[9] Y. Zhong, "A weighting scheme for content-based image retrieval," Master's thesis, School of Computer Science and Software Engineering, University of Wollo

BIOGRAPHIES

SARIKA N. ZAWARE received the B.E.. degree in Computer engineering from the University of Pune, Maharashtra State, in 2000, the M.E. degree in Computer Science And Engineering from Swami Ramanand Theerth Marathwada University , Nanded , Maharashtra , in 2005. Currently, She is an Assistant Professor of Computer Engineering at University of Pune, AISSMS IOIT. Her teaching and research areas include Data Mining, Data warehousing, web mining and Cloud Computing.