Efficient Use of Local Edge Histogram Descriptor

Dong Kwon Park
Dep. of Electronic Engineering,
Dongguk Univ.
26, 3-ga, Pil-dong, Jung-gu,
Seoul, 100-715, KOREA
+82-2-2260-3337
dkpark@dgu.ac.kr

Yoon Seok Jeon
Dep. of Electronic Engineering,
Dongguk Univ.
26, 3-ga, Pil-dong, Jung-gu,
Seoul, 100-715, KOREA
+82-2-2260-3337
hardjeon@dgu.ac.kr

Chee Sun Won
Dep. of Electronic Engineering,
Dongguk Univ.
26, 3-ga, Pil-dong, Jung-gu,
Seoul, 100-715, KOREA
+82-2-2260-3337
cswon@dgu.ac.kr

ABSTRACT

The purpose of this paper is to show how the edge histogram descriptor for MPEG-7 can be efficiently utilized for image matching. Since the edge histogram descriptor recommended for the MPEG-7 standard represents only local edge distribution in an image, the matching performance for image retrieval may not be satisfactory. In this paper, to increase the matching performance, we propose to use the global and semi-local edge histograms generated directly from the local histogram bins. Then, the global, semi-global, and local histograms of two images are compared to evaluate the similarity measure. Since we exploit the absolute locations of edge in the image as well as its global composition, the proposed matching method is considered to be a more image content-based retrieval. Experimental results support this claim. Experiments on test images for MPEG-7 core experiment show that the proposed method yields better retrieval performance especially for semantic similarity.

Keywords

MPEG-7, image retrieval, edge feature, histogram

1. INTRODUCTION

Histogram is the most commonly used characteristic to represent the global feature composition of an image. It is invariant to translation and rotation of the images and normalizing the histogram leads to scale invariance. Exploiting the above properties, the histogram is considered to be very useful for indexing and retrieving images [1][2].

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. This edge histogram descriptor proposed for MPEG-7 [3][4] consists only of local edge distribution in the image. That is, since it is important to keep the size of the histogram as small as possible for the efficient storage of the metadata, the normative edge histogram for MPEG-7 is designed to contain only local edge distribution with 80 bins. These 80 histogram bins are the only standardized semantics for the MPEG-7 edge histogram descriptor. However, with the local histogram bins only, it is not sufficient to represent global features of the edge distribution. Note that to improve the retrieval performance, we need global edge distribution as well. In this paper, we generate the semi-global and global edge histograms from the local histogram bins. Then, the global, semi-global, and local histogram bins are used to evaluate the similarity between images.

The structure of this paper is as follows. In section 2, the semantic of the normative edge histogram is described. At the next section, the edge histogram is introduced. Then, the global and semi-global edge histograms are generated in section 4. All of these histograms put together to test the retrieval performance at section 5. Finally, we conclude this paper at section 6.

2. SEMANTICS FOR NORMATIVE LOCAL EDGE HISTOGRAM

The normative part of the edge histogram descriptor consists of 80 local edge histogram bins [3][4]. The semantics of those histogram bins are described in the following sub-sections.

2.1 Partition of Image Space for Edge Identification and Localization

To localize edge distribution to a certain area of the image, we divide the image space into 4x4 sub-images as shown in Figure 1. 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. The size and the number of image-blocks in each sub-image will be described in section 3.

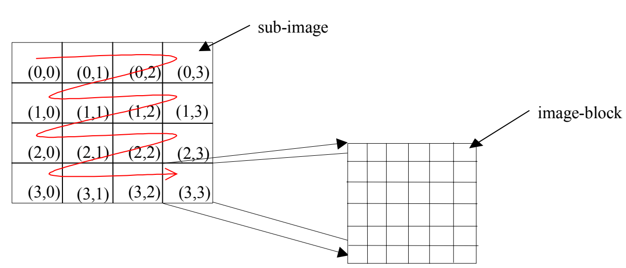


Figure 1. Definition of sub-image and image-block

4th Author : Soo-Jun Park, Electronics and Telecommunications Research Institute, 161, Kajong-dong, Yoosung, Taejon, Korea, 305-350, psj@etri.re.kr

2.2 Edge Types

As shown in Figure 2, 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. Extraction of edge information from the image block will be described in section 3.

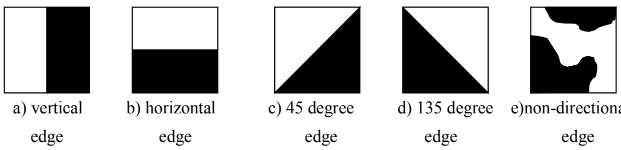


Figure 2. Five types of edges

2.3 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. By scanning sub-images according to the order shown in Figure 1, the semantics of the bins are defined as in Table 1.

Table 1. Semantics of local edge bins

Histogram bins	Semantics
Local_Edge [0]	Vertical edge of sub-image at (0,0)
Local_Edge [1]	Horizontal edge of sub-image at (0,0)
Local_Edge [2]	45degree edge of sub-image at (0,0)
Local_Edge [3]	135 degree edge of sub-image at (0,0)
Local_Edge [4]	Non-directional edge of sub-image at (0,0)
Local_Edge [5]	Vertical edge of sub-image at (0,1)
: : :	: : :
Local_Edge [74]	Non-directional edge of sub-image at (3,2)
Local_Edge [75]	Vertical edge of sub-image at (3,3)
Local_Edge [76]	Horizontal edge of sub-image at (3,3)
Local_Edge [77]	45degree edge of sub-image at (3,3)
Local_Edge [78]	135 degree edge of sub-image at (3,3)
Local_Edge [79]	Non-directional edge of sub-image at (3,3)

2.4 Normalization and Quantization of the Bins

After generating local edge histograms for all 16 sub-images, we need to normalize each bin in the histogram by dividing it with the total number of image-blocks with an edge in the corresponding sub-image. Then, each histogram bin has a value ranging from 0 to 1. To represent the normalized bin values in binary form, we need to quantize them. Since the normalized bin values are normally distributed in a small range (say, from 0 to 0.3), bin values are non-linearly quantized. The quantization tables are obtained by adopting the Lloyd-Max algorithm. Then, assigning 3 bits per bin we have total 3x80=240 bits to represent the local histogram [4].

3. EDGE EXTRACTION METHOD

To extract both directional non-directional edge features, we need to define a small square image-block. That is, we divide an image space into non-overlapping square blocks and then we can extract edge information from each block. Note that, regardless of the image size, we divide the sub-image into a fixed number of image-blocks. That is, 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.

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

$$block\_size = \left\lfloor \frac{x}{2} \right\rfloor \times 2 \tag{2}$$

Here, image_width and image_height represent the horizontal and vertical sizes of the image, respectively. The desired_num_of_block is given and fixed to cope with the various image resolutions.

Edge feature is extracted from the image-block as shown in Figure 3. 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 in Figure 4 to obtain edge magnitudes. That is, by using equations (3) ~ (7), we can obtain directional edge strengths. Among the calculated five directional edge strengths for five edge types, if the maximum of them is greater than a thresholding value (Th_{edge}), then we accept that the block has the corresponding edge type. More specifically, let us label the sub-blocks from 0 to 3 as in Figure 3. For the k^{th} (k=0,1,2,3) sub-block of the (i, j)th image-block, we can calculate the average gray level $A_k(i, j)$.

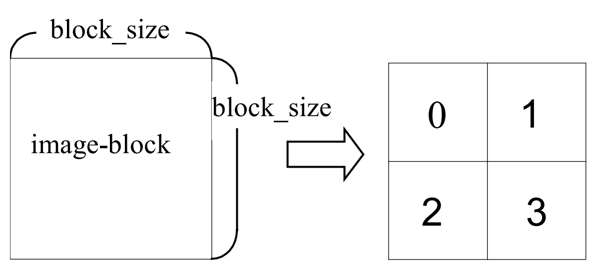


Figure 3. Sub-blocks and their labeling

Adopting the same labeling order as in Figure 3, we have the coefficients of the vertical edge filter in Figure 4-a) as follows.

ver_edge_filter (0) = 1

ver_edge_filter (1) = -1

ver_edge_filter (2) = 1

ver_edge_filter (3) = -1

Similarly, we can represent the filter coefficients for other edge filters as shown in Figure 4 – b), c), d) and e).

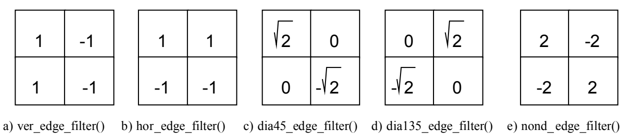


Figure 4. Filters for edge detection

Note that the edges which do not have any directionality are extracted by nond_edge_filter in Figure 4 – e). Using five edge filters in Figure 4, we can obtain five edge strengths for the image block (i, j) as follows.

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

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

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

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

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

If the maximum value among five edge strengths obtained from equations (3) to (7) is greater than a threshold (Th_{edge}) as in equation (8), then the image-block is considered to have the corresponding edge in it.

$$\max\{ver\_edge\_stg(i,j), hor\_edge\_stg(i,j), dia45\_edge\_stg(i,j), dia135\_edge\_stg(i,j), nond\_edge\_stg(i,j)\} > Th_{edge} \quad (8)$$

Finally, we also note that the edge extraction part of MPEG-7 belongs to an non-normative part. So, one can certainly adopt other edge extraction methods if they can at least extract 4 directional and one non-directional edge. The advantages of the proposed edge extraction method are its simplicity and direct applicability to MPEG-2 compressed bit stream.

4. NON-NORMATIVE GLOBAL AND SEMI-GLOBAL EDGE HISTOGRAMS

To achieve a high retrieval performance, the local histogram alone may not be enough. Rather, we may need an edge distribution information for the whole image space and some horizontal and vertical semi-global edge distributions as well. That is, beside the local histogram, we need the global and some semi-global edge histograms. The global edge histogram represents the edge distribution for the whole image space. Since there are five edge types, the global edge histogram also has five bins. For the semi-global edge histograms, we cluster four connected sub-images as shown in Figure 5. There are 13 different clusters and for each cluster we generate edge distributions for five different edge types. Consequently, we have total 80 bins(local) + 5 bins(global) + 65 bins (13x5, semi-global) = 150 bins. Note that the bin values for all global and semi-global histograms can be obtained directly from the local histogram. We also note that the 13 clusters used for the semi-global histogram in Figure 5 are supposed to represent the edge distributions in larger areas. In particular, clusters from 1 to 4 emphasize the vertical edge connectivity. Similarly, clusters from 5 to 8 are designed for the horizontal edges. The overall histogram semantics are depicted in Figure 6.

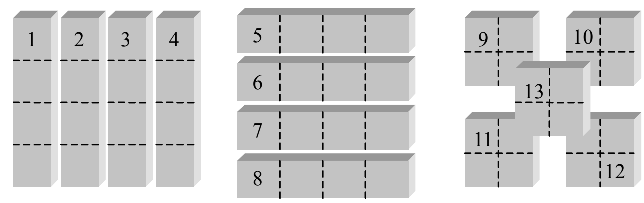


Figure 5. 13 Clusters of sub-images for semi-global histograms

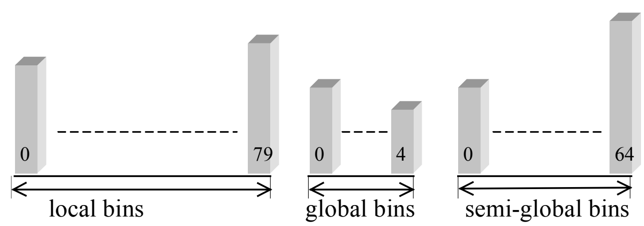


Figure 6. Overall histogram semantics

5. EXPERIMENTAL RESULTS

For our experiments, we set desired_number_block as 1100 and the threshold for edge detection (Th_{edge}) as 11. For all our experiments, we use the image data set of the MPEG-7 core experiment [5], which have 11639 images in the database. As a measure for the retrieval accuracy, we use the ANMRR(Average Normalized Modified Retrieval Rank) used also in MPEG-7 core experiments [6]. Note that lower ANMRR value means more accurate retrieval performance.

Table 2 shows the results of the retrieval accuracy for some bits-per-bin. As one can see in Table 2, the proposed method with semi-global and global histograms yield significantly better retrieval performance. As the bits-per-bin increases the ANMRR decreases. However, as shown in Figure 7, the further decrease of the ANMRR is not significant beyond 3 bits per bin. This is the reason why the standard of MPEG-7 [4] sets the bits per bin to be 3.

Figure 8~10 demonstrate retrieval results for some query images. As you can see, the proposed method retrievals semantically more similar images. In Figure 8~10, the left-upper solid-lined image is a query and 1st ranked image. Other images are displayed in a raster scan order according to the retrieval ranks.

Table 2. Retrieval Performance

	2bits/bin	3bits/bin	4bits/bin	5bits/bin
with local histogram only	0.396012	0.336060	0.317815	0.324698
with local, semi-global and global histograms (proposed)	0.363593	**0.296225**	0.285961	0.284346

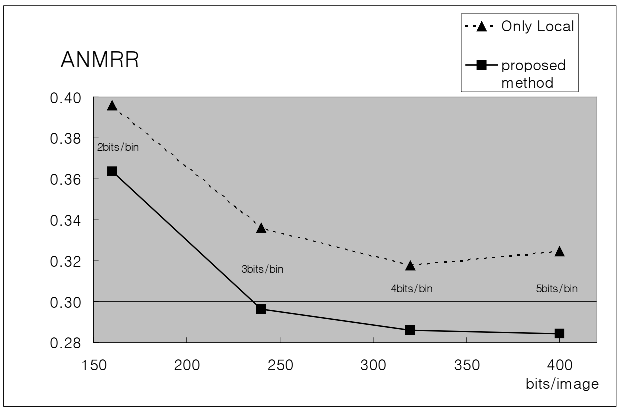


Figure 7. ANMRR

6. CONCLUSIONS

In this paper, we show how to construct global and semi-global edge histogram bins from the local histogram bins. From various possible clusters of sub-images, we used 13 patterns for the semi-global histograms. These 13 semi-global regions and the whole image space are adopted to define the semi-global and the global histograms respectively. These extra histogram information can be obtained directly from the local histogram bins without feature extraction process. Experimental results show that the semi-global and global histograms generated from the local histogram bins help to improve the retrieval performance.

7. REFERENCES

[1] M.J. Swain and D.H. Ballard, "Color indexing," *Int. J. of Computer Vision*, vol.7-1, pp. 11-32, 1991.

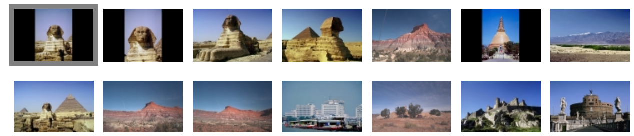
[2] A. K. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern Recognition*, Vol. 29, No. 8, pp.1233-1244, 1996.

[3] S. J. Park, D. K. Park, C. S. Won, "Core experiments on MPEG-7 edge histogram descriptor," *MPEG document M5984*, Geneva, May, 2000.

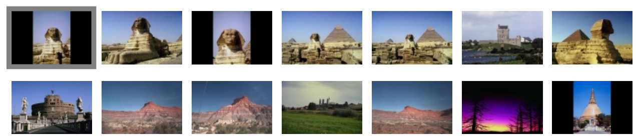
[4] ISO/IEC/JTC1/SC29/WG11 : "Core Experiment Results for Edge Histogram Descriptor (CT4)," *MPEG document M6174*, Beijing, July 2000.

[5] ISO/IEC/JTC1/SC29/WG11 : "Core Experiment Results for Spatial Intensity Descriptor (CT4)," *MPEG document M5374*, Maui, Dec. 1999.

[6] ISO/IEC/JTC1/SC29/WG11 : "Description of Core Experiments for MPEG-7 Color/Texture Descriptors," *MPEG document N2929*, Melbourne, Oct. 1999.

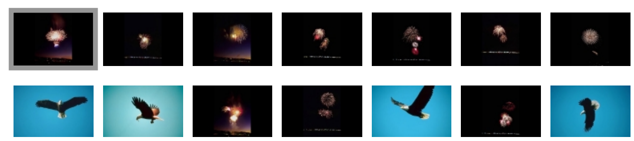


a) Retrieval results with the local histograms (3 bits/bin)

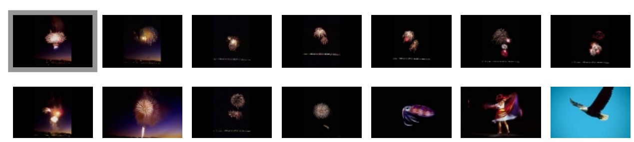


b) Retrieval results of proposed method (3 bits/bin)

Figure 8. Retrieval results of 161044.jpg image (Pyramid)

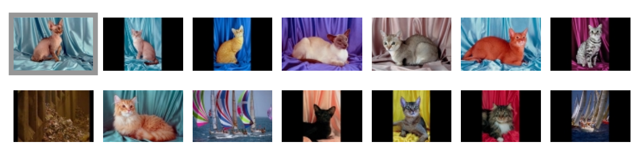


a) Retrieval results with the local histograms (3 bits/bin)

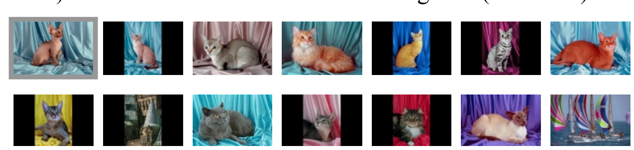


b) Retrieval results of proposed method (3 bits/bin)

Figure 9. Retrieval results of 40092.jpg image (Firework)



a) Retrieval results with the local histograms (3 bits/bin)



b) Retrieval results of proposed method (3 bits/bin)

Figure 10. Retrieval results of 458078.jpg image (Cat)