MPEG-7 Feature Visualization for CBIR Systems

Laila Shoukry, Sebastian Klenk, Gunther Heidemann
Stuttgart University, Intelligent Systems Department,
Universitaetsstrasse 38, D-70569 Stuttgart, Germany

Abstract—Content-Based Image Retrieval techniques are developed in an attempt to make fast and efficient exploration of visual content possible. In addition, standard MPEG-7 visual features should offer interoperability across different multimedia database systems. This paper presents an attempt to bridge the gap between the target semantic concepts of multimedia data and the available low-level visual descriptors by providing an intermediate feedback to the user of a CBIR system where the features selected in his query are visualized according to his input image. Two MPEG-7 visual features, a color feature (Color Layout Descriptor - CLD) and a texture feature (Edge Histogram Descriptor - EHD), were chosen for this purpose.

Index Terms—Information Retrieval, Machine Vision, Pattern Recognition.

I. INTRODUCTION

THE overwhelming growth of multimedia content collections in recent years and the wide access to these huge quantities through the internet made it essential to find fast and efficient ways of exploring digital data to locate and retrieve information matching our needs. This process is especially challenging for visual content where text-based indexing and retrieval is no longer practical and where a desired automated imitation of the intelligent human visual perception mechanism is extremely sophisticated. This is the reason why the field of Content-Based Retrieval, where indexing is automatically performed on the basis of media content, has been for years subject to active research. A lot of techniques have been proposed for the processes of feature extraction, similarity matching and retrieval. The need for a standard metadata representation of multimedia content allowing more interoperability across different multimedia database systems has led to the development of MPEG-7 [1] as a standard for multimedia content description. Lots of research has been conducted on making use of MPEG-7 visual descriptors [2] in the field of Computer Vision, especially in Content-Based Image Retrieval (CBIR) Systems (see [3], [4]).

Unfortunately, bridging the gap between the target semantic concepts of multimedia data and the available low-level visual descriptors is still an unsolved problem. Users of CBIR Systems are frequently frustrated by the images retrieved as results for their queries as they fail to understand the mechanism by which these systems are operating. To address this issue, a new approach was suggested in [5] which should help the user get an idea about how his input image is interpreted by the system to guide him in his future searches. This should be done through a kind of intermediate feedback where the features selected by the user are visualized according to his input image. The suggestion is implemented in our project, where two MPEG-7 visual features, a color feature (CLD) and a texture feature (EHD), are visualized.

II. CLD VISUALIZATION

A. Implementation

The CLD [6] represents the spatial distribution of colors in an image. To extract this feature, the input image array is first partitioned into $8 \times 8$ blocks. Representative colors are then selected and expressed in YCbCr color space and each of the three components (Y, Cb and Cr) is transformed by $8 \times 8$ DCT (Discrete Cosine Transform). The resulting sets of DCT coefficients are zigzag-scanned and the first few coefficients (6 from the Y-DCT-matrix and 3 from each DCT matrix of the two chrominance components) are non-linearly quantized to form the descriptor. The Descriptor is saved as an array of 12 values. For further details, please refer to [6].

To visualize the CLD descriptor, the above process should be reversed using only the information remaining in the final descriptor. The following steps were involved in the reconstruction process:

1) From the descriptor, the DCT coefficients of each color component were separated, repositioned into their zigzag indices in the matrix and the rest of each matrix was filled with zeros to replace those values which were lost in the formation of the descriptor.

2) The three DCT matrices were now transformed back into the spatial domain using the $8 \times 8$ IDCT (Inverse Discrete Cosine Transform).

3) The spatial domain matrices were converted from the YCbCr color space back to the RGB color space.

4) The visualization image of the reconstructed RGB $8 \times 8$ matrices is created which has the size $\text{blockwidth} \times \text{blockheight} \times 8$.

It should be noted that the reconstructed image could be smaller than the original image if the original dimensions were not divisible by 8.

B. Results

It is clear from the example in Fig.1 how much information is left in the CLD descriptor about the color layout of the image. By comparing the original visualization of the representative colors of the 8x8 image blocks and the visualization of the image reconstructed using only the information available in the CLD descriptor, one can see how much information was lost during the process of the descriptor formation, particularly by discarding lots of DC coefficients from the DCT matrices.
III. EHD VISUALIZATION

A. Implementation

The EHD [8] represents the spatial distribution of edges in an image. The extraction process of the EHD consists of the following stages:

1) The image array is divided into 4x4 subimages.
2) Each subimage is further partitioned into non-overlapping square image-blocks whose size depends on the resolution of the input image.
3) The edges in each image-block are categorized into one of the following six types: vertical, horizontal, 45° diagonal, 135° diagonal, nondirectional edge and no-edge. The first five edge types are detected using the filter coefficients illustrated in Fig. 2. Information about no-edge blocks can be automatically obtained after normalization.
4) Now a 5-bin edge histogram of each subimage can be obtained.
5) Each bin value is normalized by the total number of image-blocks in the subimage.
6) The normalized bin values are nonlinearly quantized to limit the number of bits needed for the descriptor.

The EHD visualization process consisted of the following steps:

1) Inverting the non-linear quantization of the extracted EHD descriptor. During this decompression, some edge information gets lost.
2) Inverting the normalization of the extracted EHD descriptor to regain the absolute number of occurrence of each edge type.
3) Creating a 2D vector whose first dimension specifies the subimage and whose second dimension specifies the edge types contained in each subimage and their absolute number of occurrences (they were repeated with a frequency relative to their frequency of occurrence). This vector represents an edge histogram for the image which the visualization methods can utilize. Many types of edge arrays can then be extracted from this vector for different visualization purposes.
4) Building an array with the edges being randomly ordered in each subimage.
5) Building an array with the edges being linearly ordered in each subimage (starting with vertical edges and ending with no-edge image-blocks).
6) Direct visualization of both arrays.
7) Enhanced visualization of both arrays.

The direct visualization was performed by specifying the visualizations for the different edge types as seen in Fig. 3:

These visualizations were then applied while passing through the edge arrays (linear or random, respectively).

The drawback of this visualization technique is that the borders of the subimage are visualized as thick black edges whose thickness depends on the number of pixels being discarded due to dimension indivisibility during the EHD extraction. Another drawback is that the visualization image appears pixelated due to the sub-blocks being filled with one and the same color, as four such sub-blocks constitute one edge. This is why an enhanced visualization method was needed. To overcome the pixelation problem, finer patterns were used for the visualization of edge blocks as seen in Fig. 4 instead of those of the first method.

Additionally, the edges which resided on the border to the next subimage were elongated so as to cover the discarded pixels between the subimages and produce a more natural undivided image.
B. Results

The resulting visualizations look like the examples in Fig. 5.

As seen in the figure, this visualization eliminated the drawbacks of the first visualization method. The pixelation as well as the appearance of thick edges separating the subimages were covered.

Although in a linear visualization, one can recognize the share of each edge type in each subimage better, for most images the random visualization gives a better result than the linear visualization, as the latter sometimes destroys the overall shape of the image.

By comparing the results in Fig. 6, it can be noted how similar their visualizations are due to the similarity of their edge distributions.

To further illustrate how the visualization performs, Fig. 7 shows a rectangle as a regular simple shape.

IV. Combined Visualization

A CBIR system generally depends in its matching mechanism not only on one descriptor, but on a combination of different descriptors to increase retrieval accuracy. A lot of systems also give the user the choice of the descriptor combination and their respective weights.

In [10] the visual features of each sub-image were characterized by the representative colors of the CLD as well as the edge histogram of the EHD at that sub-image using weighing factors. The Query-By-layout project (QBL) allowed the user to directly input information about edges and colors using sliding bars for each subimage separately. As a result,
it created a visualization of the edge popularity of the EHD (indicated by percentages) and the luminance of the CLD (using gray levels) in all sub-images.

The reason why the developer of [10] decided to combine especially the CLD descriptor and the EHD descriptor into one is that in both descriptors image features, color and edge distribution respectively, can be localized in separate $4 \times 4$ sub-images which makes them suitable for a combined representation.

"[...]Since the EHD and the IDCT coefficients of the CLD are based on $4 \times 4$ and $8 \times 8$ grids, respectively, there exists a spatial one-to-one correspondence between the sub-image of the EHD and 2x2 IDCT values of the CLD. That is, the adjacent 2 $\times$ 2 IDCT values correspond to one sub-image of the EHD. So, the visual features of each subimage can be characterized by the 2x2 representative colors of the IDCT of the CLD as well as the edge histogram of the EHD at that sub-image.[...][10]

Fig. 8 shows how both descriptors where concatenated for retrieval in the QBL system.

![Figure 8. Combining EHD and CLD as suggested in [10]](image)

It is interesting to note that in [11] other MPEG-7 descriptors were also combined together into one representation, namely Scalable Color Descriptor (SCD) for color and Homogeneous Texture Descriptor (HTD) for texture representation, respectively. Also, in [12] two color descriptors, Color Structure and Dominant Color, were combined together into a suggested "Dominant Color Structure Descriptor". Finally, in [13] all MPEG-7 visual descriptors were "fused" into one descriptor.

This is why we have chosen to additionally offer a joint visualization of both descriptors, EHD and CLD.

The edges of the EHD were drawn with a color a little darker or a little lighter than the background color which represented the CLD (See Fig.9).

For images with lighter colors, the first version would be more suitable, whereas for images with darker colors, the second version could be preferred. The brightness of the color can also be specified by the user.

V. Evaluation

Fig. 10 shows some intermediate visualizations as well as the final visualization for an image with a circle as a simple shape.

![Figure 10. Intermediate visualizations for an input image](image)

As can be seen in the figure, the structure of a circle is still recognizable, although we have a random distribution of edges in each subimage. From the visualization alone it can be concluded that using only our two descriptors, a CBIR system can retrieve all images with a blue circle on a white background. Hence the visualization can give us more insight on whether the descriptors used were convenient for the particular input image or not.

![Figure 11. Visualization results for two texture images (input image source:[14])](image)

Finally, Fig. 11 shows the visualization results for two images showing different textures. As in these images, the number of different edge types are considerably less than the previous -more complicated- drawings, the visualizations here show the predominant edge types in the images which give a good idea about how the original textures might look like.
VI. CONCLUSION

This paper presents an attempt to visualize two MPEG-7 visual descriptors, the Color Layout Descriptor and the Edge Histogram Descriptor. A possible joint visualization of both descriptors was also implemented. This was done through two stages: Feature Extraction and Image Reconstruction. In the first stage, features were extracted according to the guidelines of the MPEG-7 eXperimentation Model (XM). In the second stage, synthetic images were constructed using only the information available in the final form of the descriptors. This should serve as an authentic representation of what the descriptors capture from the provided input images. The developed visualizations should guide an ordinary user of an MPEG-7 compliant Content-Based Image Retrieval (CBIR) system which uses the stated descriptors in adjusting either his seed images or the descriptors used for his query accordingly in order to get the desired results. Results showed that the descriptors capture much less than the user would expect and knowing how much information is actually left in the final form of the descriptors might guide him in choosing the proper descriptor (or combination of descriptors) for his query. Therefore, it is recommended that similar visualizations be developed for the rest of the MPEG-7 descriptors. For systems which use weighted combinations of different descriptors, more combined visualizations should also be implemented. Moreover, the visualization for the Edge Histogram Descriptor which was implemented in our project could be further enhanced to give a better representation of the descriptor. Visualizations for all MPEG-7 descriptors could then be incorporated into a fully-functional MPEG-7 compliant CBIR system in such a way as to provide a feedback for the user according to his query and guide him in his future decisions.

REFERENCES