Fast Texture Feature Extraction Method Based on Segmentation for Image Retrieval

Yi-Ling Chen, Tse-Wei Chen, and Shao-Yi Chien
Media IC and System Lab
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University
BL-421, No. 1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan
{yipaul, variant, sychien}@media.ee.ntu.edu.tw

Abstract—This paper describes a fast and efficient texture feature extraction method based on image segmentation in HSV color space, which is especially useful in resource-limited embedded systems. In this approach, a query image is first segmented based on its color feature, and the proposed technique is applied on the labeled image to efficiently extract texture feature. Compared with the feature extraction method using discrete wavelet transform, the proposed algorithm is 15.51 times faster than the wavelet-based method while keeping the distortion of the retrieval results within a reasonable range.

I. INTRODUCTION
The size of digital image collections has been growing dramatically not only in personal computers but also in hand-held devices, which come along with cameras, in last decades. This huge amount of information cannot be efficiently used unless it is well-organized. Since it is impossible to manually annotate all collections by text-based keywords, the technique for images being indexed by their own visual contents becomes acute. Therefore, the requirement for efficient Content-Based Image Retrieval (CBIR) systems has increased tremendously in many application areas. The key step in CBIR systems is to extract features from every image and use these features to measure the similarity across them. In addition to color, texture is also a key component to human perception. Among various texture descriptors, Discrete Wavelet Transform (DWT) for decomposing an image into orthogonal components is widely used for its better localization and computationally inexpensive properties.

However, texture feature extraction is computationally intensive, and the operating speed is very critical in CBIR systems as the response time needs to be short enough for good interactivity. Therefore, our motivation is to provide a fast and efficient texture feature extraction method for the CBIR systems in embedded systems. The proposed approach is performed on the labeled image, which is obtained from color image segmentation in a quantized HSV color space [1]. Only exclusive-or and addition operations are required in the proposed method. By comparing the texture feature extraction time between the wavelet-based method and the proposed algorithm, the proposed algorithm is 15.51 times faster than the wavelet-based method on average.

Fig. 1. Overview of the proposed algorithm.

Fig. 2. An illustration of the proposed texture extraction method, Label Wavelet Transform (LWT): (a) the original image, (b) the labeled image after segmentation, and (c)(d) three level decomposition of the LWT. Each pixel in the segmentation image is represented by the color of its cluster centroid.
II. TEXTURE FEATURE EXTRACTION IN CBIR

An overview of the proposed CBIR system is illustrated in Fig. 1. The proposed algorithm, Label Wavelet Transform (LWT), is based on color image segmentation [1], and it is an extension of DWT-based texture feature extraction method. The 2-D DWT is computed by applying separable filter banks to the gray level images. The detail images $D_{n,1}$, $D_{n,2}$, and $D_{n,3}$ are obtained by band-pass filtering in a specific direction, and they can be categorized into three frequency bands: HL, LH, HH band, respectively. Each band contains different directional information at scale $n$. The texture feature is extracted from the variance ($\sigma_{n,i}^2$) of the coefficients $c_{n,i}$ of the detail image $D_{n,1}$, $D_{n,2}$, and $D_{n,3}$ at different scale $n$. To represent the texture feature of an image $q$, the texture feature vector of DWT is defined as [2]:

$$T_{DWT}(q) = [\sigma_{1,1}^2, \sigma_{1,2}^2, \sigma_{1,3}^2, ..., \sigma_{N_{\text{max}},3}^2],$$  \hspace{1cm} (1)

where $N_{\text{max}}$ denotes the largest scale. In this work, $N_{\text{max}} = 3$ is chosen according to the size of testing images, and a 9-component texture feature vector is used to represent each image.

Although the Haar transform is the fastest transform of DWT, it requires floating point multiplications or integer multiplications with divisions, which are inefficient for the real-time processing demand. Therefore, LWT concentrates on providing a fast texture feature extraction method on the labeled image from image segmentation. An illustration of a three-level LWT is shown in Fig. 2. For a labeled image, it is meaningless to directly apply DWT on it since these labels represent the cluster label of the pixels, which only reveals whether neighboring pixels belong to the same cluster or not. In order to extract texture feature in the labeled image, low-pass and high-pass filtering are implemented by downsampling and calculating the number of neighboring label changes in the three directions (horizontal, vertical, and diagonal) in the labeled image. $H_{n,1/2/3}$ is the $n$-th level measure of the horizontal/vertical/diagonal label dissimilarity in the labeled image. It is observed that only the logic operations (exclusive-or) and additions are needed for computation in LWT, and neither multiplications nor divisions are required. The histograms of the label changes in three directions are calculated and are then treated as the texture feature to represent that texture is a property of the homogeneity and variations in a region. For the purpose of comparing the retrieval results with that of the DWT feature, the representation of LWT feature vector of an image $q$ is similar to that of DWT feature vector:

$$T_{LWT}(q) = [H_{1,1}, H_{1,2}, H_{1,3}, ..., H_{N_{\text{max}},3}],$$  \hspace{1cm} (2)

where $N_{\text{max}}$ denotes the largest scale. In this work, $N_{\text{max}} = 3$ is chosen according to the size of testing images, and a 9-component texture feature vector is used to represent each image.

The CBIR system obtains the retrieval results by using the nearest neighbor algorithm based on the distance measurement, and the database images are then retrieved according to their rankings of distance, with the one having the highest ranking being retrieved first.

<table>
<thead>
<tr>
<th>Category</th>
<th>Beach</th>
<th>Building</th>
<th>Bus</th>
<th>Flower</th>
<th>Food</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>FET (DWT/LWT)</td>
<td>15.37</td>
<td>14.27</td>
<td>11.58</td>
<td>14.55</td>
<td>21.78</td>
<td>15.51</td>
</tr>
<tr>
<td>MAP (DWT/LWT)</td>
<td>1.15</td>
<td>1.02</td>
<td>1.10</td>
<td>0.98</td>
<td>1.16</td>
<td>1.075</td>
</tr>
<tr>
<td>Total FET (DWT/LWT)</td>
<td>3.17</td>
<td>3.19</td>
<td>2.75</td>
<td>2.93</td>
<td>4.02</td>
<td>3.21</td>
</tr>
</tbody>
</table>

III. EXPERIMENTAL RESULTS

In the experiment, the COREL database [3] is adopted. The average precision comparison (ratio), which is the average precision over recall, and “FET” stands for “Feature Extraction Time” to extract the texture feature. Total “FET” considers the gray level transform time for both algorithms: for DWT, the time of gray level transform is included; for LWT, the time for image segmentation is included. From Table I, it is observed that the average texture feature extraction time (FET) of LWT is 15.51 times faster than that of DWT, which is especially suitable in hand-held devices to meet the real-time response demands. Moreover, by comparing total “FET”, the speed of LWT is still 3.31 times faster than DWT. As for the quality of the retrieval results using LWT, there is only 6.9% loss of average precision as compared with DWT. Since labeled images of segmentation contain less information than gray scale images, it is reasonable that the performance of LWT is lower than that of DWT. In summary, the experiments show that the proposed method, LWT, can greatly speed up texture feature extraction while keeping the distortion within a reasonable range. The proposed method effectively provides the trade-off between the performance and the texture feature extraction time.

IV. CONCLUSIONS

The proposed algorithm is an application of color image segmentation. Different from existing methods, the proposed approach utilizes color feature to segment the image and then apply only exclusive-or and addition operations to extract texture from the labeled image. This algorithm provides a fast and efficient way towards texture feature extraction. It is especially useful in the retrieval systems with real-time demands in embedded systems.

REFERENCES

