Texture Segmentation Based on Features in Wavelet Domain for Image Retrieval

Ying Liu[a], Si Wu[b], Xiaofang Zhou[a]

[a]School of Information Technology and Electrical Engineering,
The University of Queensland, Queensland, 4072, Australia
[b] Institute of Information Research, 21 Heng Mui Keng Terrace, Singapore, 119613

ABSTRACT

Texture is a fundamental feature which provides significant information for image classification, and is an important content used in content-based image retrieval (CBIR) system. To implement texture-based image database retrieval, texture segmentation techniques are need to segment textured regions from arbitrary images in the database. Texture segmentation has been recognized as a difficult problem in image analysis. This paper proposed a block-wise automatic texture segmentation algorithm based on texture features in wavelet domain. In this algorithm, texture features of each block are extracted and L2 distance between blocks are calculated; a pre-defined threshold is used to determine if two blocks should be classified into same class, hence belong to same textured region. Results show that the proposed algorithm can efficiently catch the texture mosaics of arbitrary images. In addition, features of each textured region can be obtained directly and used for image retrieval. Applying various thresholds instead of uniform threshold to different blocks according to their homogeneity property, texture segmentation performance can be further improved. Applied to image database, the proposed algorithm shows promising retrieval performance based on texture features.

Keywords: Image database, Content-based image retrieval, texture, texture segmentation, discrete wavelet transform, texture feature, textured-based image retrieval, retrieval performance

1. INTRODUCTION

Large image collections are available nowadays in various areas such as digital broadcasting, digital library, entertainment, education, and multimedia communication. With this vast amount of image, more efficient storage, indexing and retrieval of visual information are strongly required [1]. The traditional approach for searching digital image databases requires a considerable level of human effort, and is often subjective. These problems leads to content based image retrieval (CBIR) system which retrieves images on the basis of automatically-derived features such as colour, texture and shape [2]. CBIR not only eliminates manual processing for image indexing but also provides automatic indexing according to image contents.
Among contents based features, texture is a fundamental feature which provides significant information for image classification [3]. Though no precise definition so far, textures often refer to homogeneous patterns or spatial arrangements of pixels that regional intensity or color alone does not sufficiently describe. Texture describes the content of many real-world images, such as fabrics, clouds, trees, bricks, etc [4].

Each image in database can be considered as mosaics of textured regions, and features of each textured region can be used to index the whole database for retrieval purpose. To implement such texture-based image retrieval, the first task is to segment textured regions from arbitrary images.

Texture segmentation plays an important role in both computer vision and image analysis. It consists of partitioning the input image into connected regions which are homogeneous with respect to a texture property. Texture segmentation has been recognized as a difficult problem and has been attempted in numerous ways. Roughly, these methods can be classified into feature-based methods, model-based methods, and structure-based methods [5]. Structure-based methods partition images under the assumption that the textures in the image have detectable primitive elements, arranged according to placement rules. In feature-based methods, regions with relatively constant texture characteristics are sought. Model-based methods hypothesize underlying processes for textures and segments using certain parameters of these processes. Model-based methods can be considered as a subclass of feature-based methods since model parameters are used as texture features.

From another point of view, texture segmentation can be classified into supervised and unsupervised texture segmentation. Supervised texture segmentation is restricted to a set of known textures. Unsupervised texture segmentation is a challenging task because of the lack of prior knowledge about the texture in the image, such as how many different textures exist, what types of texture they are and where they are located. Many techniques for texture segmentation were proposed in the last two decades [6][7][8]. Most approaches first map the differences in spatial structures, either stochastic or geometric, into the differences in feature space. The performance of features is closely related to the segmentation result and much research has been done in this [9][10]. Then, a segmentation algorithm is applied to the feature space to extract the homogeneous regions.

According to the segmentation methods, texture segmentation can be grouped into two classes: region-based and boundary-based. Boundary-based approach seeks to detect the differences of texture in adjacent regions while region-based approach tries to identify regions with uniform texture.

In the early 1990's, Bovik et. al. demonstrated the reliability of texture segmentation schemes based on multi-channel approach using 2-D Gabor functions as localized spatial filter [11]. Gabor filters are designed to respond to different spatial frequencies and have been applied to texture segmentation [12]. However, a large combination of parameters makes texture segmentation using Gabor filters computationally expensive. Wavelet has been proved to be a promising alternative and has several potential advantages over Gabor filters [13]. Inspired by work in [14], the authors in [15] combined the quad-tree data structure with wavelet sub-band representation to perform image segmentation. The quad-tree is grown by iteratively testing conditions for splitting parent blocks based on texture content of children blocks.

Texture segmentation can be pixel-wise or block-wise. Pixel-wise segmentation schemes evaluate the texture features in a neighbourhood surrounding each pixel in the image. The advantage of pixel-wise segmentation over block-wise segmentation lies in the removal of blockyness at region boundaries. However, the computation load is heavier. As image retrieval system does not require exact boundary of the segmented regions, block-wised segmentation is often chosen since it is much faster [16].

This paper proposed a block-wise texture segmentation algorithm, based on features derived from wavelet coefficients, which have been proved to be efficient for texture description. The algorithm can effectively capture the textured regions of any image and features of each region can be obtained, which can be used directly to index image database for retrieval purpose. Simulation results prove that the proposed algorithm can effectively segment images into textured regions, and based on features of these regions, promising retrieval performance is shown.

2. TEXTURE SEGMENTATION

2.1 2-D Discrete Wavelet Transform (DWT)

The 2-D DWT represents an image in terms of scaling function ϕ^{LL} and a set of shifted and dilated wavelet functions $\{\psi^{LH}, \psi^{HL}, \psi^{HH}\}$ that form an orthogonal basis for $L^2(R^2)$. With a *J*-scale DWT, an image *f(x,y)* of size

$$S_0 = M_0 * N_0 \tag{1}$$

is decomposed as:

$$f(x,y) = \sum_{t=0}^{N_J-1} \sum_{s=0}^{M_J-1} a_{J,s,t} \phi_{J,s,t}^{LL}(x,y) + \sum_{b \in B} \sum_{j=1}^{J} \sum_{t=0}^{N_j-1} \sum_{s=0}^{M_j-1} d_{j,s,t}^{b} \psi_{j,s,t}^{b}(x,y) \tag{2}$$

with $$\phi^{LL}_{j,s,t}(x,y) = 2^{-j/2}\phi(2^{-j/2}x - s, 2^{-j/2}y - t),$$ (3)

$$\psi^{b}_{j,s,t}(x,y) = 2^{-j/2}\psi^{b}(2^{-j/2}x - s, 2^{-j/2}y - t),\ b \in B, B = \{LH, HL, HH\}$$ and

$$N_j = N_0/2^j, M_j = M_0/2^j,\ j = 1,\ldots, J$$ (4)

$a_{J,s,t}$ is the scaling coefficient, and $d^{b}_{j,s,t}$ is the $(s,t)_{th}$ wavelet coefficient in scale *j* and sub-band *b*, $d_j = \sum_{b\in B} d^b_j$ stands for all coefficients in the three sub-bands in scale j. In some literatures, the term ‘level’ is used in stead of ‘scale’.

Fast implementation of DWT using Quad-Mirror Filters(QMF) is proposed by S. Mallat [17]. The diagram of a 1-level 2-D DWT is shown below in Figure 1. h', g' are the QMF which are designed as low-pass and high-pass filters respectively. $a_1, d_1^{LH}, d_1^{HL}, d_1^{HH}$ are the coefficients of sub-band *LL*, sub-bands *LH*, *HL* and *HH* in scale 1, respectively. Applying the same decomposition to *LL* sub-band coefficients (a_1), a 2-level DWT can be obtained. Repeating the above procedure, we can obtain a *J*-scale DWT decomposition of the image *f(x,y).*

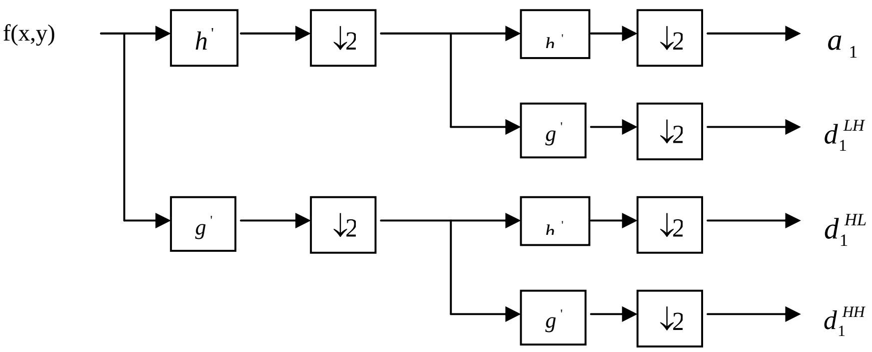


Figure 1 Diagram of 1-level 2-D DWT

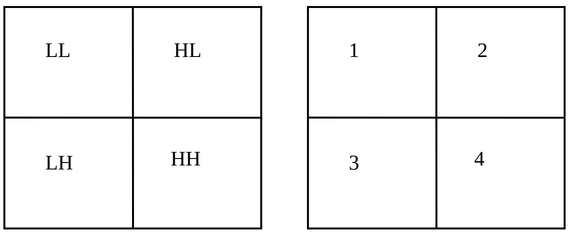


Figure 2 4 sub-bands in 1-level wavelet transform

2.2 The proposed texture segmentation algorithm

The system partitions an image into blocks of 4*4 pixels, then apply wavelet transform to each block and extracts 4 features from the transform coefficients. Image retrieval system does not require exact boundary of the segmented regions [15], hence block-wise segmentation is chosen since it is much faster than pixel-wise segmentation. Experimental results show that block size of 4*4 is proper.

Features in wavelet domain have been shown to be effective for texture representation [18]. After experiments, we choose the following features for texture segmentation.

A 1-level wavelet transform using 4-tap Daubechies filters is applied to each block, this produces 4 frequency sub-bands as shown in Figure 2. LL is the low frequency sub-band, HL, LH, HH are three high frequency sub-bands. These 4 sub-bands are denoted as sub-band 1,2,3, and 4.

Then, 4 features describing the energy of each sub-band are calculated. For example, suppose the 4 coefficients in LL sub-band are $\{C_{LL(0,0)}, C_{LL(0,1)}, C_{LL(1,0)}, C_{LL(1,1)}\}$, the feature of this sub-band is calculated as:

$$F_{LL} = \sqrt{\left(\sum_{i=0}^{1}\sum_{j=0}^{1} C_{LL(i,j)}^{2}\right)/4} \tag{5}$$

With features of all blocks available, we can classify all blocks into different classes.

Starting from the first block at the top-left corner, which belongs to the 1st class, we scan all blocks one by one. For block *Bc*, suppose the blocks previously scanned belong to *m* different classes, we first determine if *Bc* belongs to any of the *m* classes. If yes, update the feature of this class; if not, add a new class as the $(m+1)_{th}$ class, and feature of *Bc* is its initial feature.

Each class corresponds to a textured region, and 4 features are used to represent it. The initial feature of a class is the feature of the first block classified into this class. How to update the feature of this class when more blocks are classified into it? Assume that a class has *k* blocks already, and feature of this class is $(f_{LL}^{k}, f_{LH}^{k}, f_{HL}^{k}, f_{HH}^{k})$. In addition, feature of the $(k+1)_{th}$ block classified into this class is $(F_{LL}, F_{LH}, F_{HL}, F_{HH})$. The updated feature of this class is calculated as the average of features of all the blocks it contains, as in equation (6)

$$\left(f_{LL}^{k+1} = \frac{k*f_{LL}^{k}+F_{LL}}{k+1},\; f_{LH}^{k+1} = \frac{k*f_{LL}^{k}+F_{LH}}{k+1},\; f_{HL}^{k+1} = \frac{k*f_{HL}^{k}+F_{HL}}{k+1},\; f_{HH}^{k} = \frac{k*f_{HH}^{k}+F_{HH}}{k+1}\right) \tag{6}$$

How to determine whether *Bc* belongs to any of the *m* classes or not? We compute its distance to each of the *m* classes, if the smallest distance d_i falls below threshold *Thr*, then *Bc* belongs to class *i*. If all distances are above *Thr*, *Bc* belongs to a new class.

Finally we obtain *N* textured regions each containing N_i blocks. In addition, we get the 4 features of each region, and these features can be used for indexing purpose later.

Results show that the above algorithm is effective in capturing textured regions of arbitrary images. Examples are given in Figure 3. Shown at the left side are the original images, the segmentation results are at the right side. With *Thr* as 30.0, the image in Figure 3(a) is segmented into 13 classes. For the image in Figure 3(b), with *Thr* as 50.0, 4 classes are obtained. Note that blocks in same class might be disconnected.
We did not apply post-processing to remove those tiny isolated regions, as they do not affect retrieval performance.

3. ADJUSTING PARAMETERS

From description above, obviously we can see that parameter *Thr* affects the segmentation results. The higher *Thr* is, the smaller the number of textured regions obtained. If *Thr* is too low, one textured region we are interested in might be classified into different parts; if *Thr* is too high, different textured regions might be classified into same class. Figure 4 shows an example, note that the tree leaves in the image are classified into several regions.

The more the number of regions, the more homogeneous the content is; however, more regions means higher feature dimensions which will make indexing more difficult, considering the 'curse of dimensionality' problem in high-dimensional indexing [19]. A trade-off shall be made depending on system requirement. Simulation shows that *Thr* value between 30 and 50 is proper for many images.

Different textured regions in an arbitrary image could have different homogeneity property. The more homogeneous a textured region is, the more close the features of different blocks in this region are. Hence, in stead of using uniform *Thr* for all blocks, we vary *Thr* according to the content. This can not only reduce the number of segmented regions and hence reduce the number of features extracted, but also relieve the problem of one textured region being classified into different parts.

High frequency coefficients in wavelet domain correspond to sharp intensity changes in the image, like edges and boundaries. Sub-bands 2, 3, 4 in Figure 2 are the three high frequency sub-bands. We define a parameter *Eng_High*, which is the percentage of energy of coefficients in sub-bands 2,3,4, relative to the total coefficient energy across all 4 sub-bands.

$$Eng\_High = (\sum C_2^2 + \sum C_3^2 + \sum C_4^2) / \sum_{i=1}^{4} \sum C_i^2 \qquad (7)$$

where C_2, C_3 , C_4 are the wavelet coefficients in sub-bands 2,3,4; $\sum C_2^2$, $\sum C_3^2$, $\sum C_4^2$ are the energy of coefficients in these three sub-bands, respectively; $\sum_{i=1}^{4} \sum C_i^2$ is the total energy of coefficients in all 4 sub-bands.

According to texture property, different regions in an image can be classified as 'flat' regions which contain little intensity changes and hence with low *Eng_High* values, 'textured' regions which are composed of repetitive patterns and hence with high *Eng_High* values, and 'non-textured' regions which are not so homogeneous as 'textured' regions.

With 1-level QMF wavelet transform, most energy concentrates in the low frequency sub-bands. For example, for the image in Figure 3(a), 81.9% of all the 4*4 blocks have *Eng_High*<0.005. In our algorithm, experimentally, different *Thr* values are applied to blocks with different *Eng_High* as below:

if (*Eng_High*<0.005) Thr=*Thr*0; // 'flat' regions, very little high frequency information
else if(Eng_High<0.02) Thr=*Thr*0*C1; // 'non-textured' regions, less high frequency information
else Thr=*Thr*0*C2; // 'textured' regions, with more high frequency information
where C1 and C2 are two constants above 1.0, with c2>c1, *Thr*0 is a given initial threshold.

With higher threshold for blocks with more high-frequency information, the number of classes is reduced, and regions that are not very homogeneous but can be regarded as one texture are recognized. In addition, the computational load is reduced. Figure 4 shows an example. With uniform *Thr* 20.0, the original image is segmented into 37 classes; with *Thr* being 20 for regions as the lake, sky, and *Thr* being 30 for regions as the tree leaves, the original image is segmented into 9 classes.

4. IMAGE RETRIEVAL BASED ON TEXTURE CONTENT

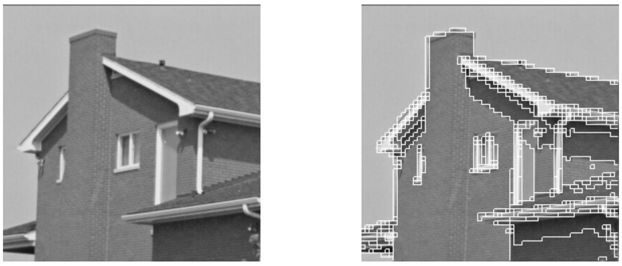
With a collection of 100 arbitrary images of different sizes and contents as the test database, texture segmentation is applied to each image in the database, and features of each segmented region are extracted.

'Qury-by-sample' is a typical query in texture-based image retrieval system, i.e., to select from the database all images containing textured region similar to the given query sample. In our experiments, given different query texture patterns, images containing regions similar to the query were selected from the 100 arbitrary images.

Similarity is measured by L2 distance between the feature of the query texture, and feature of each textured regions in an image. Retrieval rate is defined as

$$\text{Retrieval Rate} = \text{Relevant images selected} / \text{Relevant images to be found from the database} \quad (8)$$

Promising retrieval performance is shown by simulation results. Figure 5 gives 2 examples, in which all images selected when retrieval rate is 100% are listed.



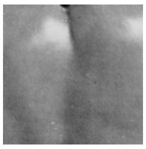
(a) *Thr*=30 (*N*=13)

(b) *Thr*=50(*N*=4)

Figure 3 Texture segmentation results

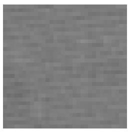
(a) original (b)uniform *Thr*=30 (*N*=16) (c) varying *Thr* (*N*=6)

Figure 4 Texture segmentation with uniform or various threshold



Query texture pattern

Retrieval results



Query texture pattern

Retrieval results

Figure 5 Retrieval results of query on texture pattern

5. CONCLUSIONS

For texture-based image retrieval, texture segmentation is required to segment textured regions from arbitrary images in the database, and features of these regions can be used as index for retrieval purpose.

This paper presents a block-wise automatic texture segmentation algorithm, which can segment any arbitrary images into textured regions. Meanwhile, features of each textured regions are obtained. Simulation results show that the proposed algorithm is effective in texture segmentation and it allows efficient image retrieval on the bases of texture content. The proposed algorithm can be applied to large real-world image database for efficient textured-based image indexing.

In our algorithm, features of the segmented regions are calculated as the mean of features of all blocks it contains, features more efficient for indexing purpose can be found later.

6. ACKNOWLEDGEMENT

This work is partly done when the author was attached to School of Computing, National University of Singapore, under project R-252-000-015-112/303.

7. REFERENCES

1. MPEG-7, "*Context and Objective Document*", N2460, 1998.

2. John P Eakins, Margaret E Graham, "*Content-based image retrieval*", Technical report, Institute for Image Data Research, University of Northumbria at Newcastle, Jan. 1999.

3. Yong Man Ro, "Matching pursuit: contents featuring and image indexing", *Proc. of SPIE, Symposium on voice, video and data communication: Multimedia storage and archiving system III, vol. 3527*, 1998, pp89-100.

4. John R. Smith, "*Integrated Spatial Feature Image Systems: Retrieval, Analysis and Compression*", Ph.D. thesis, Graduate School of Arts and Sciences, Columbia University, February, 1997.

5. Todd R.Reed and J.M.Hans Du Buf, 'A review of texture segmentation and feature extraction techniques', *CVGIP:Image Understanding, vol.57(3)*, 1993, pp359-372.

6. Hao Ye, Yan Qiu Chen, "Unsupervised texture segmentation using resonance algorithm for natural scenes", *Pattern Recognition Letters, vol.21(9),* 2000, pp741-757.

7. Andrey P., Tarroux, P., "Unsupervised segmentation of markov random-field modeled textured images using selectionist relaxation", *IEEE Trans. Pattern Analysis Machine Intelligence, 20(3),* 1998, pp252-262.

8. Jain, A., Farrokhnia, F., "Unsupervised texture segmentation using gabor filters", *Pattern Recognition, 24(12),* 1991, pp1167-1186.

9. Buf, J.D.,Spunn, M. Kardan, M., "Texture feature performance for image segmentation", *Pattern Recognition 23(2),* 1990, pp291-309.

10. Chen Y.Q., Nixon, M.Thomas D., 'Statistical geometrical features for texture classification", *Pattern Recognition, 28(4),* 1995, pp537-552.

11. Bovik, A.C., Clark M. and Geisler, W.S., "Multichannel texture analysis using localized spatial filters", *IEEE Trans. On Pattern Recognition and Machine Intelligence, vol. 12, No. 1*, 1990, pp55-73.

12. Dunn, D. and Higgins, W.E., "Optimal Gabor filters for texture segmentation", *IEEE Trans. on Image Processing, Vol.4., No.7,* 1995, pp947-964.

13. Mausumi Acharyya and Malay K. Kundu, "Adaptive basis selection for multi texture segmentation by M-band wavelet packet frames", *Inter. Conf. on Image Processing*, 2001, pp.622-625.

14. M.Spann and R. Wilson, "A quad-tree approach to image segmentation which combines statistical and spatial information", *Pattern Recognition vol. 18*, 1985, pp257-269.

15. J.R.Smith and Shih-Fu Chang, "Quad-tree segmentation for texture-based image query", *Proc. of 2nd Annual ACM Multimedia Conference*, Oct. 1994, pp278-286.

16. Jia Li, James Ze Wang, and Gio Wiederhold, "Classification of textured and non-textured images using region segmentation", *Inter. Conf. on Image Processing*, Sep.2001, pp 754-757.

17. Stephene Mallat, "*A wavelet tour of signal processing*", Academic Press, San Diego, 1998.

18. M.Unser, "Texture Classification and Segmentation Using Wavelet Frames," *IEEE. Trans. Image Processing, vol. 4, No. 11*, Nov. 1995, pp. 1549-1560.

19. Eamonn Keogh, Kaushik Chakrabarti,Michael Passani, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and Information Systems*,vol.3, 2001, pp.263-286.