CLASSIFICATION OF TEXTURED AND NON-TEXTURED IMAGES USING REGION SEGMENTATION

Jia Li, James Ze Wang, and Gio Wiederhold

Department of Computer Science, Stanford University

ABSTRACT

The classification of general-purpose photographs into textured and non-textured images is critical for developing accurate content-based image retrieval systems for large-scale image databases. With the accurate detection of textured images, we may retrieve images based on features tailored for the corresponding image type. In this paper, we present an algorithm to classify a photographic image as textured or non-textured using region segmentation and statistical testing. The application of the system to a database of about $60,000$ general-purpose images shows much improved accuracy in retrieval.

1. INTRODUCTION

Effective indexing and searching of large-scale image databases is a challenging problem in image processing and computer vision. The automatic derivation of semantics from the content of an image is the focus of interest for most research on image databases. Classifying a photographic image as textured or non-textured is important for the task of content-based image retrieval for large-scale image databases. With the accurate classification of the two image types, features for retrieval can be tailored for textured or non-textured images.

By *textured* images, we refer to images that are composed of repeated patterns and appear like a unique texture surface, as shown in Figure 1. As textured images do not contain clustered objects, the perception of such images focuses on color and texture, but not shape, which is critical for understanding *non-textured* images. Thus an efficient retrieval system should use different features to depict these two types of images. To our knowledge, the problem of distinguishing textured images and non-textured images has not been explored in the image retrieval literature. In this paper, we describe an algorithm to detect textured images based on segmentation results.

THIS WORK WAS SUPPORTED IN PART BY THE NATIONAL SCIENCE FOUNDATION. WE WOULD LIKE TO THANK OSCAR FIRSCHEIN OF STANFORD UNIVERSITY FOR VALUABLE DISCUSSIONS.

Also of Department of Electrical Engineering. Currently with Xerox Palo Alto Research Center. Email: `jiali@db.stanford.edu`

Also of Department of Medical Informatics. Email: `wangz@cs.stanford.edu`

Also of Department of Electrical Engineering and Department of Medical Informatics. Email: `gio@cs.stanford.edu`

2. IMAGE SEGMENTATION

This section describes our image segmentation procedure based on color and frequency features using the k-means algorithm [2]. For general-purpose images such as the images in a photo library or the images on the World-Wide Web (WWW), automatic image segmentation is almost as difficult as automatic image semantic understanding. Currently there is no existing non-stereo image segmentation algorithm that can perform at the level of the HVS. The segmentation accuracy of our system is not crucial for image search because we use a robust integrated region-matching (IRM) scheme to compare images, which is insensitive to inaccurate segmentation.

To segment an image, the system partitions the image into blocks with 4×4 pixels and extracts a feature vector for each block. The k-means algorithm is used to cluster the feature vectors into several classes with every class corresponding to one region in the segmented image. An alternative to the block-wise segmentation is a pixel-wise segmentation by forming a window centered around every pixel. A feature vector for a pixel is then extracted from the windowed block. The advantage of pixel-wise segmentation over block-wise segmentation is the removal of blockyness at boundaries between regions. Since we use rather small block size and boundary blockyness has little effect on retrieval, we choose block-wise segmentation with the benefit of 16 times faster segmentation.

The k-means algorithm is a well-known statistical classification algorithm [2]. Six features are used for segmentation. Three of them are the average color components in a 4×4 block. The other three represent energy in high frequency bands of wavelet transforms [1, 5], that is, the square root of the second order moment of wavelet coefficients in high frequency bands. We use the well-known LUV color space, where L encodes luminance, U and V encode color information (chrominance). To obtain the other

Fig. 1. Sample textured images.

three features, the Haar wavelet transform is applied to the L component of the image. After a one-level wavelet transform, a 4×4 block is decomposed into four frequency bands as shown in Figure 2. Each band contains 2×2 coefficients. Without loss of generality, suppose the coefficients in the HL band are $\{c_{k,l}, c_{k,l+1}, c_{k+1,l}, c_{k+1,l+1}\}$. One feature is then computed as

$$f = \left(\frac{1}{4} \sum_{i=0}^{1} \sum_{j=0}^{1} c_{k+i,l+j}^2 \right)^{\frac{1}{2}} .$$

The other two features are computed similarly from the LH and HH bands. The motivation for using the features extracted from high frequency bands is that they reflect texture properties. Moments of wavelet coefficients in various frequency bands have been shown to be effective for representing texture [6]. The intuition behind this is that coefficients in different frequency bands show variations in different directions. For example, the HL band shows activities in the horizontal direction. An image with vertical strips thus has high energy in the HL band and low energy in the LH band.

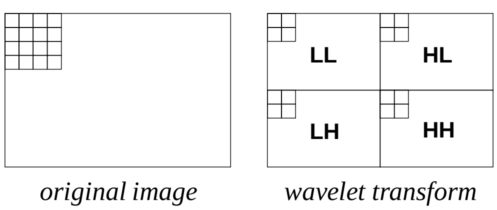


Fig. 2. Decomposition of images into frequency bands by wavelet transforms.

Examples of segmentation results for both textured and non-textured images are shown in Figure 3. Segmented regions are shown in their representative colors. It takes about one second on average to segment a 384×256 image on a Pentium Pro 430MHz PC using the Linux operating system. We do not apply post-processing techniques to smooth region boundaries or to delete small isolated regions because these errors are often less significant. Since our retrieval system is designed to tolerate inaccurate segmentation, cleaning the segmentation results by post-processing (at the cost of speed) is unnecessary.

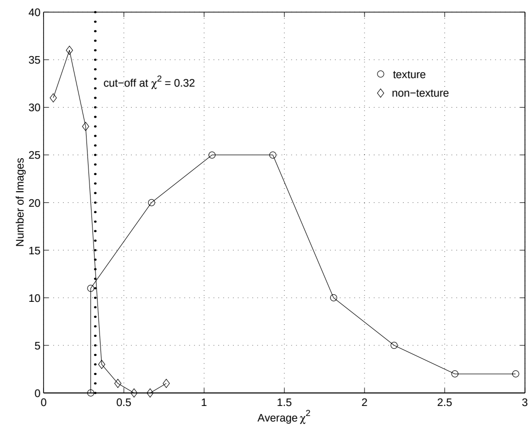


Fig. 4. The histograms of average χ^2's over 100 textured images and 100 non-textured images.

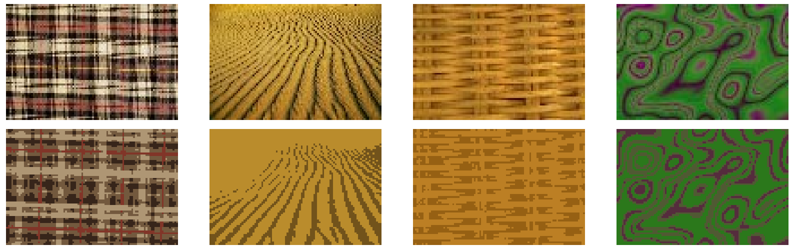
3. CLASSIFICATION OF TEXTURED AND NON-TEXTURED IMAGES

In this section we describe the algorithm to classify images into the semantic classes *textured* or *non-textured*. As shown by the segmentation results in Figure 3, regions in textured images tend to scatter in the entire image, whereas non-textured images are usually partitioned into clumped regions. A mathematical description of how evenly a region scatters in an image is the goodness of match between the distribution of the region and a uniform distribution. The goodness of fit is measured by the χ^2 statistics.

We partition an image evenly into 16 zones, denoted by $\{Z_1, Z_2, ..., Z_{16}\}$. Suppose the image is segmented into regions $\{r_i : i = 1, ..., m\}$. For each region r_i, its percentage in zone Z_j is $p_{i,j}$, $\sum_{j=1}^{16} p_{i,j} = 1$, $i = 1, ..., m$. The uniform distribution over the zones should have probability mass function $q_j = 1/16$, $j = 1, ..., 16$. The χ^2 statistics for region i, χ_i^2, is computed by

$$\chi_i^2 = \sum_{j=1}^{16} \frac{(p_{i,j} - q_j)^2}{q_j} = \sum_{j=1}^{16} 16 (p_{i,j} - \frac{1}{16})^2 .$$

Textured and non-textured images are classified by thresholding the average χ^2 statistics for all the regions in the image, $\bar{\chi}^2 = \frac{1}{m} \sum_{i=1}^{m} \chi_i^2$. If $\bar{\chi}^2 < 0.32$, the image is labeled as textured; otherwise, non-textured. We randomly



#reg=4	2	2	3
$\bar{\chi}^2 = 0.13$	0.21	0.01	0.07

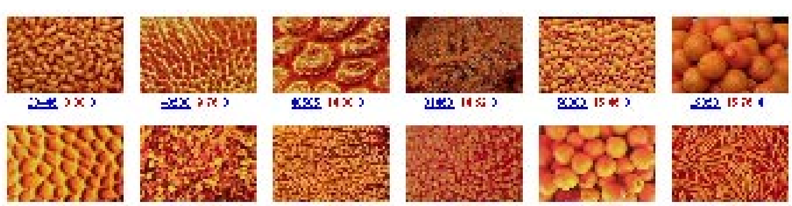
(a)

#reg=2	4	4	12
$\bar{\chi}^2 = 0.69$	1.61	1.45	1.25

(b)

Fig. 3. Segmentation results by the k-means clustering algorithm: (a) Original texture images and the region segmentation results, (b) Original non-textured images and the region segmentation results.

querying on a non-textured image



querying on a textured image

Fig. 5. Content-based image retrieval using textured and non-textured image classification.

chose 100 textured images and 100 non-textured images and computed $\bar{\chi}^2$ for them. The histograms of $\bar{\chi}^2$ for the two types of images are shown in Figure 4. It is shown that the two histograms separate significantly around the decision threshold 0.32. The sensitivity and specificity of the classification are both above 95%.

4. EXPERIMENTS

The algorithm has been implemented on a Pentium Pro 430MHz PC using the Linux operating system. On average, one second is needed to segment an image and to compute the features of all regions.

We tested this algorithm on a general-purpose image database including about $60,000$ pictures, which are stored in JPEG format with size 384×256 or 256×384. These images were segmented and classified into textured and non-textured types. According to the system, there are 3772 textured images in the database, about 6% of the total collection.

Figure 5 shows two example query results obtained from the SIMPLIcity system [4], a content-based image retrieval system using the classification of textured and non-textured images. An on-line demo for the system is provided at URL: `http://WWW-DB.Stanford.EDU/IMAGE`.

5. CONCLUSION

A method for classifying textured and non-textured images using statistical testing has been developed. The integration of the classifier into an image database retrieval system has demonstrated much improved results.

6. REFERENCES

[1] I. Daubechies, *Ten Lectures on Wavelets*, Capital City Press, 1992.

[2] J. A. Hartigan and M. A. Wong, "Algorithm AS136: a k-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100-108, 1979.

[3] J. Li and R. M. Gray, "Context based multiscale classification of images," *Int. Conf. Image Processing*, Chicago, Oct. 1998.

[4] J. Li, J. Z. Wang, G. Wiederhold, "SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries," submitted for journal publication, September 1999.

[5] Y. Meyer, *Wavelets Algorithms and Applications*, SIAM, Philadelphia, 1993.

[6] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Processing*, vol. 4, no. 11, pp. 1549-1560, Nov. 1995.