

TEXTURE CLASSIFICATION BASED ON DISCRIMINATIVE FEATURES EXTRACTED IN THE FREQUENCY DOMAIN

Antonella Di Lillo, Giovanni Motta[†], James A. Storer

{dilant, storer}@cs.brandeis.edu, Computer Science Dept., Brandeis University, Waltham, MA 02454
[†]giovanni.motta@hp.com, Hewlett-Packard Corp., 16399 W. Bernardo Dr., San Diego, CA 92130

ABSTRACT

Texture identification can be a key component in Content Based Image Retrieval systems. Although formal definitions of texture vary in the literature, it is commonly accepted that textures are naturally extracted and recognized as such by the human visual system, and that this analysis is performed in the frequency domain. In this work, a feature extraction method is presented which employs a discrete Fourier transform in the polar space, followed by a dimensionality reduction. Selected features are then processed with vector quantization for the supervised segmentation of images into uniformly textured regions. Experiments performed on a standard test suite show that this method compares favorably to the state-of-the-art and improves over previously studied frequency-domain based methods.

Index Terms— Image texture analysis, Pattern classification

1. INTRODUCTION

Texture analysis plays an important role in computer vision and many image processing applications such as texture-based image retrieval. The extraction of metadata based on the identification of texture can be used as an important tool to improve the performance of Content Based Image Retrieval (CBIR) systems. The general problem is identifying images in a database (e.g., digital library, medical, or scientific database) of similar content, given a sample image or sample image content. Manual annotation of image data is often unfeasible (because of the amount of data involved) and unreliable (since it may be impossible to predict which characteristics of the image are relevant to the search). Methods to automatically extract information from images have been developed in the past, in order to produce metadata annotation for CBIR. Most of these methods are based on statistical distributions of the gray-scale value or the color value of single pixels. These methods have achieved good performance in many circumstances, but the main drawback is that the gray-scale and color distribution values are identical for a number of dramatically differing spatial arrangements of pixels. While there is no agreement on a formal definition of texture, it is commonly accepted

that textures are naturally extracted and recognized as such by the human visual system. It is also widely believed that the visual system extracts relevant features in the frequency domain, independently of their illumination and the presence of noise. Furthermore, human experiments have revealed that the visual cortex contains orientation and scale band-pass filters for vision analysis. Unfortunately, previous attempts to classify textures in the frequency domain have shown inferior performance with respect to statistical methods. Based on these motivations, we propose a classifier that combines vector quantization (VQ) with a feature extractor operating in the frequency domain. Our method compares favorably with the state-of-the-art and improves over previously proposed frequency-based methods.

2. PREVIOUS WORK

Despite many attempts to define texture, vision researchers have not yet produced a universally accepted formal definition. Mathematical models have been proposed in an attempt to converge on a unified definition, and properties such as uniformity, density, coarseness, roughness, regularity, linearity, directionality, direction, frequency, and phase are known to play an important role in the description of a texture. Based on these properties, a variety of feature extraction techniques have been studied. Tuceryan and Jain [1] identify four major feature extraction methods in texture analysis: statistical, geometric, model-based, and signal processing based.

Randen and Husøy [2] extensively studied filter banks and compared these methods using the same system setup. The filters that have been compared are: Laws filter masks, Ring and Wedge filters, Dyadic Gabor filter bank, Wavelet transform, Discrete Cosine Transform, Quadrature Mirror filters, and the Tree-Structured Gabor filter bank. Performance was also compared to two popular methods that did not involve filtering; the first, using the co-occurrence matrix, the second, based on an autoregressive model. Randen and Husøy concluded that different filtering methods yield different results for different images, i.e. there was no single method that performed the best for all the images. Furthermore, they identified that because of the large number of features, the computational complexity is

typically very large in both feature extraction and classification. Therefore, a low feature vector dimensionality, maximized feature separation, and in some cases, a simpler classifier are highly preferable.

Randen and Husøy's results were improved by Mäenpää, Pietikäinen, and Ojala [3] using the Local Binary Pattern texture operator (LBP), introduced by Ojala, Pietikäinen, and Harwood [4]. LBP is a statistical feature extractor that has been successfully used in several classification and segmentation problems. LBP classifies textures by comparing quantized histograms with a log-likelihood measure. LBP preserves information on the texture's spatial distribution, and is invariant to all monotonic gray scale transformations. A multi-scale version of LBP has also been studied [3]; MP-LBP combines texture information extracted at different resolutions.

Recently, several authors have studied the use of VQ to classify textures [5], [6]. Direct comparison with these proposals is difficult because the results have not been assessed with a comparable methodology.

To overcome this problem, in our experiments, we have used a problem set, first used by Randen and Husøy [2] and later by Mäenpää, Pietikäinen, and Ojala [3]. This test suite is now part of *Outex* [7], a unified framework designed for the empirical evaluation of texture analysis algorithms.

3. PROPOSED METHOD

Textured images are classified here with a *supervised segmentation* approach. Texture features are extracted in the frequency domain and classified with a vector quantizer. In supervised segmentation, the classifier is first trained on texture samples, and then tested on images composed of texture compositions.

3.1. Training

Training aims at extracting a small set of significant and discriminating features from texture samples. The features extracted characterize textural properties of the training samples. The basis for the classification consists of a set of h features extracted from n training images, which are formed by textures belonging to k different classes. Feature extraction maps raw pixels to a feature space, and the performance of the feature extractor will greatly influence the correct classification rate. Features are collected on a pixel-by-pixel basis, with large textures being randomly sampled to contain memory usage and computational burden. Features for a pixel (x_i, y_i) are collected from a window W of $wsize^2$ neighbor pixels. The square window is weighed by a 2-dimensional Hamming window before the application of a 2D Fourier transform (FT), after which the magnitude of the coefficients is further transformed into polar coordinates:

$$FP_W(r, \theta) = \left| \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W(x, y) e^{-ir(x\cos\theta + y\sin\theta)} dx dy \right|$$

The use of a frequency-domain method has been preferred because it is consistent with the findings of researchers in biological vision (see for example [8-11]), and the mapping into polar coordinates has been found to improve the precision of the classifier on the test problems. We speculate that this is due to an improved directionality of the system. While the FT and the polar mapping can be combined to extract features that are invariant to translation and rotation, the way polar coordinates are being used here has proved experimentally to produce slightly better results than this more traditional combination of transforms.

Capturing textural features requires a large window and results in feature vectors having many components. Besides the obvious issues of time and space complexity, the components of these vectors are often highly correlated or not useful to the classification. We have experimented with two techniques for dimensionality reduction: the Principal Component Analysis (PCA) and the Fisher coefficients. PCA performs an orthonormal transformation on the feature space that retains only significant eigenvectors of the whole dataset, called principal components. The feature space is transformed and information is compacted into the smallest number of dimensions by discarding redundant features. PCA reduces the number of dimensions without considering their contributions to the classification.

A complementary approach, which ranks each feature's contribution to the classification without exploiting feature correlation, uses the computation of the Fisher coefficients. Fisher coefficients measure the discriminative power of each feature, so that dimensions that do not help with the classification can be safely discarded. Fisher coefficients are defined as the ratio of between-class variance to within-class variance:

$$F = \frac{D}{V} = \frac{\frac{1}{1 - \sum_{k=1}^K P_k^2} \sum_{k=1}^K \sum_{j=1}^K P_k P_j (\mu_k - \mu_j)^2}{\sum_{k=1}^K P_k V_k}$$

where D denotes the between-class scatter, V denotes within-class variance, μ_k, V_k mean and variance of class k , and P_k the probability of class k [12].

Unlike PCA, Fisher coefficients are computed independently on each dimension, so they do not provide any information on how to discard correlated features. The use of Fisher coefficients is less aggressive than PCA in reducing dimensionality, however, it has the advantage of requiring very little computation during the training and no computation at all during the testing.

Training our classifier consists of finding, for each sample class, a small set of "typical" feature vectors that can be compared to an unknown signature. For this reason, we employ a vector quantizer that determines, for each class, a small set of centroids. Centroids are computed so that they minimize the Mean Squared Error with the feature vectors collected for each sample texture.

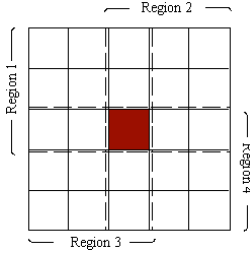


Figure 1. Overlapping quadrants in a Kuwahara filter.

3.2. Testing

In the testing, the performance of the proposed method has been assessed by classifying a number of unknown images (also called *problems*). The segmentation of the problem is done on a pixel-by-pixel basis. A window of $wsize \times wsize$ pixels is centered on the pixel being classified, and textural features are extracted as described in the previous section. The dimensions of the feature vector are reduced, and the result is compared to the centroids of each class. The class associated to the closest centroid is finally assigned to the pixel.

Performance is assessed by counting the number of pixels correctly classified. By assuming that adjacent pixels are likely to belong to the same texture, the classification performance can be improved with a post-processing filter. The simplest method uses a smoothing (low-pass) filter on the result of the classifier. The smoothing filter must preserve edges of the segmented areas, and the filter introduced by Kuwahara, et al. [13], is an example of such edge-preserving filter. As depicted in Figure 1, the Kuwahara filter divides a square neighborhood of each pixel into four overlapping quadrants, each containing the central pixel. For each quadrant, the mean and the variance of the pixel values are computed; then the value of the central pixel is replaced by the mean of the quadrant having smallest variance.

We aim at filtering segmentation maps consisting of texture indices, not natural images for which the Kuwahara filter has been designed. Indices don't have a physical meaning, so determining mean and variance of the quadrants is not appropriate. It is possible however to use the same strategy and compute the histogram of the indices, and the entropy of the values in each quadrant. Our *Kuwahara-like* filter replaces the central index with the highest-frequency index belonging to the quadrant with the smallest entropy.

4. EXPERIMENTAL RESULTS AND CONCLUSIONS

We have compared our results to the results of Randen and Husøy [2] and Mäenpää, Pietikäinen, and Ojala [3], on the same test suite they have used in their work. The test suite is based on images from three texture databases: the album of

Brodatz [14], the MIT Vision and Modeling Texture Databases¹, and the MeasTex Image Texture Database². The use of images acquired under different conditions and with different equipment increases the complexity of the task. Textures are all gray-scale, and have equalized histograms. The test images, called *problems*, combine 2 to 16 sample textures and have size 256×512 (2 textures), 256×256 (5 textures with straight and round borders), 256×640 (10 textures with simple borders), and 512×512 (16 textures with complex borders). To each texture in a problem is associated a 256×256 training sample, that is similar, but not identical, so that unbiased error estimate can be obtained.

Texture segmentation identifies regions of uniform texture. Segmentation quality depends largely on the size of the window W . While a larger window captures textural features at different scales, a small window is preferred during the test since it allows a more precise localization of the boundaries between adjacent regions. Choosing the optimal window for a given set of textures is not a trivial task. A multi-resolution approach, fully compatible with our method, could be used to entirely avoid the problem. However, for simplicity, we have tested our method on windows having a fixed radius of 20 pixels ($wsize = 41$), a size already suggested in [3].

PCA and Fisher coefficients are both effective at compacting the feature vector while retaining its discriminative power. Selecting the coefficients so that the cumulative sum of their magnitude is not greater than 98%, reduces the number of features by an order of magnitude with the PCA, and by about one third with Fisher.

Empirically we found 4 centroids per texture sample to be sufficient to achieve competitive results while containing the complexity of the training.

Figure 2 shows the classification obtained by our algorithm on problems P1, P6, P9 and P10, respectively containing 2, 5, 10, and 16 textures. As one might expect, error concentrates at the region boundaries.

Table 1 compares classification rate with the results reported in [2] and [3]. On average, our method, depending on the use of PCA or Fisher, improves respectively by 8.9% to 10.1% over Randen and Husøy (which also use spatial filtering) and by 1.4% to 2.5% over Basic LBP (which is based on statistical texture representation).

The classification rate of our method is also roughly comparable to Multi-Predicate LBP, a more complex approach that combines features collected at multiple resolutions.

The improvement over Randen and Husøy [2] is noteworthy since, they utilize more sophisticated filtering. Furthermore, unlike in their study, we use a single feature extraction method for all problems.

¹ <http://www.media.mit.edu/vismod>

² <http://www.cssip.elec.uq.edu.au/~guy/meastex/meastex.html>

P#	R. & H. [2]	Basic LBP	MP-LBP (1,3)	MP-LBP (1,5)	MP-LBP (1,2,3)	Our PCA	Our Fisher
P1	92.80	93.80	93.90	92.70	93.00	96.58	96.63
P2	81.10	81.90	85.70	85.50	86.40	83.04	83.95
P3	79.40	87.90	89.80	86.30	88.90	89.65	86.97
P4	83.20	90.00	90.90	89.50	91.20	91.38	93.38
P5	82.80	89.10	92.00	90.60	91.70	91.67	91.85
P6	65.30	83.20	84.70	84.50	84.50	81.71	81.34
P7	58.30	79.20	79.30	76.70	78.10	75.33	78.33
P8	67.70	77.20	81.90	79.70	82.40	81.08	78.04
P9	72.20	80.80	78.60	76.70	76.80	79.86	90.39
P10	99.20	99.70	99.60	99.40	99.60	99.59	99.64
P11	99.80	99.00	99.20	99.30	99.20	98.48	98.67
P12	97.50	90.10	94.70	93.20	94.70	97.89	98.86
Avg	81.62	87.66	89.14	87.84	88.88	88.85	89.84

Table 1. Comparison between the classification performances.

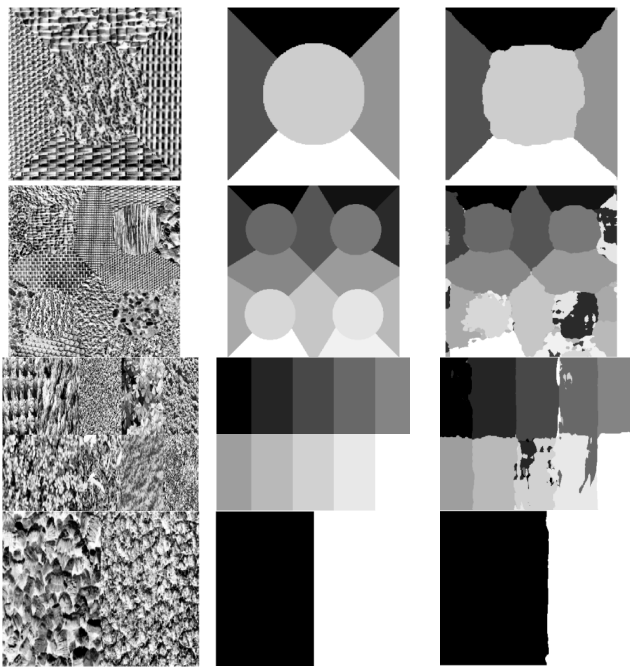


Figure 2. Problem (left), ground truth (center) and classification (right) for P1, P6, P8, P10.

As previously mentioned, our method is compatible with multi-resolution and the use of this technique would make possible to increase robustness to resolution changes and alleviate issues due to poor choice of the fixed-size window.

Feature vectors resulting from the use of Fisher coefficients are considerably bigger, however this approach has much lower complexity than PCA since it does not require any matrix multiplication to reduce the dimension of the feature vector. Despite the fact that PCA and Fisher are somewhat complementary, combining the two does not seem to bring any particular advantage.

The use of polar coordinates makes our algorithm relatively robust to variations in the rotational angle of the texture. The polar mapping typically accounts for about 4.5% of the performance, and it seems to provide a more

precise characterization of textures exhibiting a quasi-random pattern. Ongoing experimentation suggests that resilience to rotational variance and a more precise characterization of the textures allow for better performance on natural scenes as well.

5. REFERENCES

- [1] M. Tuceryan, A. K. Jain, "Texture Analysis". In: C. H. Chen, L. F. Pau, and P. S. P. Wang (eds): *The Handbook of Pattern Recognition and Computer Vision* (2nd Edition), World Scientific Publishing Co., pp. 207-248, 1998.
- [2] T. Randen, J. H. Husøy, "Filtering for Texture Classification: A Comparative Study," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:4, pp.291-310, 1999.
- [3] T. Mäenpää, M. Pietikäinen, and T. Ojala, "Texture Classification by Multi-Predicate Local Binary Pattern Operators," *ICPR*, pp. 3951-3954, 2000.
- [4] T. Ojala, M. Pietikäinen, and D. Harwood, "A Comparative Study of Texture Measures with Classification Based Feature Distributions", *Pattern Recognition*, vol. 29, pp. 51-59, 1996.
- [5] K. Pyun, C. S. Won, J. Lim, and R. M. Gray, "Texture classification based on multiple Gauss mixture vector quantizers," *Proc. of ICME 2002*, pp. 501-504, 2002.
- [6] A. Aiyer, K. Pyun, Y. Huang, D. B. O'Brien, and R. M. Gray "Lloyd Clustering of Gauss Mixture Models for Image Compression and Classification," *Signal Processing: Image Communication*, 2005.
- [7] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen, "Outex – New Framework for Empirical Evaluation of Texture Analysis Algorithms," *Proc. 16th Int. Conf. on Pattern Recognition*, Quebec, Canada, 1, pp. 701-706, 2002.
- [8] C. B. Blakemore, F. W. Campbell, "On the existence of neurons in the visual system selectivity sensitive to the orientation and size of retinal images," *J. Physiol.* 203, pp. 237-260, 1969.
- [9] F. W. Campbell, J. Nachmias, J. Jukes, "Spatial frequency discrimination in human vision," *J. Opt. Soc. Am.*, 60, pp. 555-559, 1970.
- [10] L. Maffei, A. Fiorentini, "The visual cortex as a spatial frequency analyzer," *Vision Res.* 13, pp. 1255-1267, 1973.
- [11] R. L. de Valoris, D. G. Albrecht, L. G. Thorell, "Spatial frequency selectivity of cells in macaque visual cortex," *Vision Res.*, 22, pp. 545-559, 1982.
- [12] J. Schürman, *Pattern Classification, a unified view of statistical and neural approaches*, John Wiley & Sons, 1996.
- [13] M. Kuwahara, K. Hachimura, S. Eiho, M. Kinoshita, *Digital Processing of Biomedical Images*, Plenum Press, New York, pp. 187-203, 1976.
- [14] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, New York, Dover, 1996