MULTIRESOLUTION EIGENIMAGES FOR TEXTURE CLASSIFICATION

Mehrdad Jabbarzadeh Gangeh[1], Michel Bister[2], Madasu Hanmandlu[3]
[1] Faculty of Engineering, Multimedia University, Jalan Multimedia, 63100 Cyberjaya, Selangor, Malaysia
[2] Was with Faculty of Engineering, Multimedia University. He is now with Division of Engineering, University Nottingham in Malaysia, Wisma MISC, No. 2, Jalan Conlay, 50450 Kuala Lumpur, Malaysia
[3] Was with Faculty of Engineering, Multimedia University. He is now with Dept. of Electrical Engineering, I.I.T. Delhi, Hauz Khas, New Delhi-110016, India

Mehrdad@mmu.edu.my, Michel.Bister@nottingham.edu.my, mhmandlu@ee.iitd.ernet.in,

ABSTRACT

Following an idea from [1], based on the Gaussian properties of eigenimages, this paper presents a new technique for texture classification using multiresolution eigenimages. The input image, composed of two textures from the Brodatz album, is subdivided into N sub-images of fixed size $\delta\times\delta$, which are blurred with a Gaussian and normalized. The application of Hotelling transform decomposes each sub-image into δ^2 eigenimages. The R largest resulting coefficients can be used for classification of the texture present in the sub-images. Classification is done using the fuzzy C-means (FCM) algorithm and the performance is measured with an appropriate quality factor. We discuss the successful application of this technique, as well as the influence of the different parameters of the classification process on several pairs of textures. Moreover, combination of Hotelling coefficients obtained with different values of δ is shown to improve the performance, based on the idea of analyzing the texture at different levels of resolution.

1. INTRODUCTION

There is a vast literature on texture analysis, as can be judged from the innumerable applications the texture analysis has in various fields. We now briefly discuss the categories of texture as emerged in the literature [2].

Texture analysis techniques are classified basically into four types of approaches: statistical, structural, transform-based and model-based [2]. Statistical approaches represent the texture indirectly by the non-deterministic properties that govern the distributions and relationships between the gray-levels of an image. For example, different order of moments in a localized window can be used to represent smooth, coarse, grainy textures, etc. Co-occurrence matrices are widely used in the texture analysis [3].

Structural approaches represent texture by well-defined primitives (microstructure) and a hierarchy of spatial arrangements (macrostructure) of those primitives. In these approaches, spatial structure descriptors are used to identify geometric primitives and their arrangement in an image.

The first review on texture analysis is by Haralick [4] who categorized all approaches into the statistical and structural classes. The next excellent review is by Van Gool et al. [5] who also classified all the approaches into the same two classes.

The transform-based approach is used to detect global periodicity in an image by finding high energy, narrow peaks in the spectrum like in Fourier transforms. This approach is not popular because of computational complexity. Presently, Gabor filters [6] and wavelets are popular.

Model-based texture analysis attempts to interpret an image texture by the use of regenerative image model and stochastic model [2]. This approach has evolved in recent years as can be seen in the survey paper of Reed and du Buf [7]. Some techniques using this approach are: local interaction models like autoregressive moving average (ARMA) model, Gauss-Markov random field model [8], and fractal model [9].

In recent years, scale space theory has been recognized as the vital tool for texture analysis. This is because texture displays multiscale property. Whatever may be the representation, it is applicable in different scales. Fractals which represent natural textures are derived from the decomposition of texture into multiscales.

The other types of approaches are also coming into vogue. These include entropy based approaches, neural network oriented approaches and diffusion based approaches.

Present paper originates from a discussion between the authors and Prof. Romeny, author of [1], regarding the experiment described on pp. 187-190 of [1], where the relationship between Gaussian derivatives and eigenimages is highlighted. Since eigenimages form a good basis to express an image, it seems that eigenimages for textured regions should reflect the texture somehow. Hence the idea to use these eigenimages for texture classification.

In this paper, we will first review the basic idea of eigenimages. Then we will show how the eigenimages of textured regions indeed reflect the texture present in the region. As a logical consequence, a method will be discussed to use these eigenimages to do texture classification. The method will then be extended to incorporate multiresolution property for texture classification so as to handle the problem of different grain of textures. Eventually, the sensitivity of the method to the training set will be discussed, which will lead to some important conclusions.

0-7803-8484-9/04/$20.00 ©2004 IEEE

2. EIGENIMAGE ANALYSIS

Eigenvectors E of a matrix M are defined as the vectors for which the following equation holds: $ME = \lambda E$, with λ being the corresponding eigenvalue. These eigenvectors are particularly useful in the Karhunen-Loève Transform (KLT, also called Hotelling Transform or Principal Component Analysis PCA). In this transform, the eigenvectors of the covariance matrix of a vector population are used to define a transform matrix A, the rows of which are made up of the eigenvectors weighted by decreasing magnitude of corresponding eigenvalue. Transformation of the input vectors with this transform yields a vector population, the components of which are decorrelated, i.e., the covariance matrix becomes diagonal. Applications include multispectral image analysis (whereby the vectors are composed of the different color channels of each pixel) and binary object analysis (whereby the vectors are composed of the coordinates of the pixels that are part of the object).

In the case of eigenimage analysis, the image is decomposed into N small patches of size $\delta \times \delta$. The intensities of the pixels in the patches are re-arranged as vectors of length δ^2. The covariance matrix, as well as its eigenvectors are calculated as previously. The re-arrangement of those δ^2-dimensional eigenvectors as $\delta \times \delta$ patches are called eigenimages. It has been shown that, for large values of N, those eigenimages tend to look like Gaussian derivative functions [1].

In the present paper, our aim is to use eigenimage analysis for texture classification.

3. EIGENIMAGE TEXTURE CLASSIFICATION

Since the eigenimages form a good basis in which to express the measurement vectors, it seems that the eigenimages of textured regions should reflect the features of the texture. To validate this intuition which was communicated orally by Bart M. ter Haar Romeny to the authors, the eigenimages of a number of textures from the Brodatz texture album were calculated and compared. In Figure 1, the rows correspond to different textures in the Brodatz collection, the first column is the original image, and the subsequent columns are the first 10 eigenimages, arranged in decreasing values of the corresponding eigenvalues.

From this figure, we can indeed notice a strong resemblance of the eigenimages to the textures being considered. Hence, it seems that these eigenimages could be used for texture classification. One difficulty is that the zero-order Gaussian derivative usually comes out as the first eigenimage - hence the nature of the first eigenimage could hardly be used for texture classification. Using a larger number of eigenimages as such for texture classification would drastically increase the dimensionality of the problem: each eigenimage is already of dimension δ^2, which should be reduced.

Dimensionality reduction is typically a task for KLT, since the relevance of the components of the transformed vectors is proportional to the corresponding eigenvalues, which are arranged in decreasing order of magnitude. However, applying a KLT on the eigenimages would be a double application of the KLT, since this was already applied to obtain the eigenimages themselves.

A more sensible approach would be to calculate the transform matrix A, and then to transform the measurement patches according to A and do the classification on basis of the first R components. This, however, poses the problem that the transform matrix A is calculated for individual textures. To discriminate between two textures, the training patches should be coming from both texture populations, so that the obtained eigenimages result in a transform matrix A that is ideal for discriminating between the two textures considered.

This approach was implemented using a number of pairs from the Brodatz album. We describe here the practical implementation of this texture detection scheme, and emphasize the choices and parameters it depends on.

A first choice is the selection of two textures to classify. We will call them T_1 and T_2 (e.g. Brodatz images D8 and D13). Next, a number N of $\delta \times \delta$ patches is extracted. The patches are designated as $P_i(x, y)$, $i = 0 \ldots N\text{-}1$, while the Brodatz images are $T_j(x, y)$, $j = 1, 2$. The patches $P_i(x, y)$ are selected from the images T_j with an overlap of ε pixels:

$$P_i(x, y) = T_{i\mathrm{MOD}}\left(x + ((\lfloor i/2 \rfloor))_\delta (\delta - \varepsilon),\; y + \lfloor \lfloor i/2 \rfloor / \delta \rfloor (\delta - \varepsilon)\right) \tag{1}$$

where $((x))_y$ is the modulo operator, and $\lfloor . \rfloor$ (FLOOR) returns the nearest integer value toward minus infinity. As mentioned in [1], the observation process is simulated by convolving the patches with a Gaussian with variance σ. From these patches, covariance matrix, its eigenvectors, and finally the transformation matrix A are calculated in the classical way.

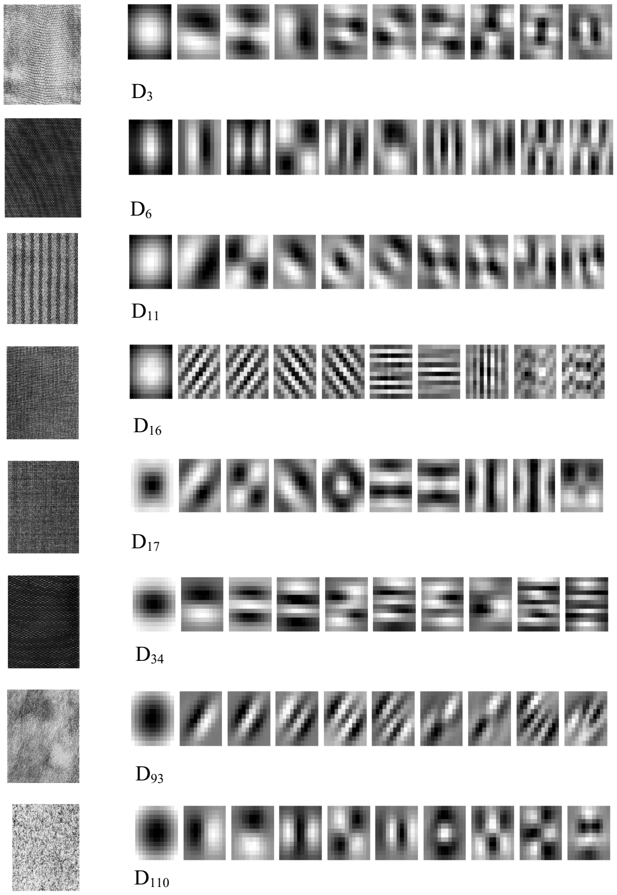

Fig. 1. Eight images from the Brodatz album, with their corresponding first ten eigenimages. Patches were taken to be of size 12x12, with 6 pixels overlap

Next, K patches are extracted at random and submitted for the application of transform. Only the first R values are kept for classification, and classification is done using fuzzy C-means classifier (FCM). As a summary, the choices are the input textures, and the parameters are: the number of training patches N, the patch size δ, the patch overlap ε, the amount of blurring σ, and the number of features used for classification R.

The extent of success is measured with a quality factor (QF), which is defined as follows:

$$QF = \left(1 - 2\times\frac{NWP}{NTP}\right)\times 100\% \tag{2}$$

where NWP is the number of wrong classified patches and NTP is the number of total classified patches.

4. MULTIRESOLUTION EIGENIMAGE TEXTURE CLASSIFICATION

Texture classification is often a resolution-related matter. Indeed, a feature size that is smaller than the grain of the texture could not possibly correctly classify a texture. On the other hand, large texture operators are computationally expensive and do not locate the texture borders well.

In our idea, we could combine operators on a number of S levels of scale by applying several operators of different sizes in parallel, and selecting the classification features as a mixture of the multi-scale features.

Our current algorithm works with S = 1, 2 or 3 (S is the number of scales). Of course, δ, ε and σ have to be made dependent on the resolution level. In our implementation, δ varies from 12 to 36 (δ_1 = 12, δ_2 = 24 and δ_3 = 36), σ from 5.5 to 17.5 (σ_1 = 5.5, σ_2 = 11.5, σ_3 = 17.5) and ε is 6 in all scales.

Results of the classification with different values of S are shown in Figure 2. From this figure, the added advantage of the multiresolution is very evident. It appears also that higher number of levels should be considered.

The influence of the other parameters is shown in figures 3-6. In each case, the value of QF is given as a function of the parameter being analyzed, and this is done for a number of Brodatz pairs as given in the legend.

From these figures, we notice that the classification is quite successful, depending on the particular texture pair to be classified. Surprisingly, there is little influence from the size of the training set N. This will be discussed further in the article. Patch size (δ) and overlap (ε) do not have a drastic influence on the performance either. This is related to the little dependence on the training set, as will be further discussed.

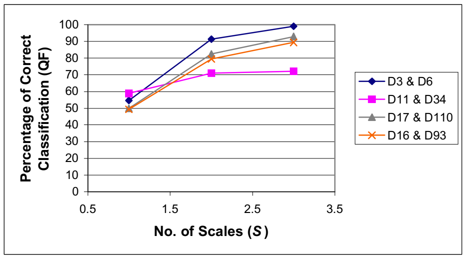

Fig. 2. Quality of eigenimage texture classification as a function of the number of multiresolution levels (S) for a number of Brodatz texture pairs. Other parameters: N = 10000, δ_1 = 12, δ_2 = 4, δ_3 = 36, ε = 6, σ_1 = 5.5, σ_2 = 11.5, σ_3 = 17.5, R = 10, 20 and 30 for 1, 2 and 3 scales respectively

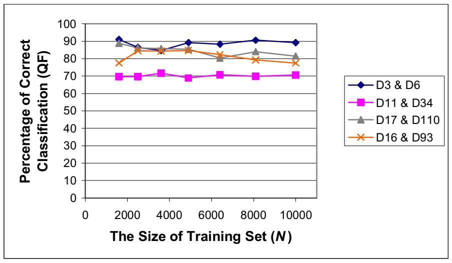

Fig. 3. Quality of eigenimage texture classification as a function of the size of the training set (N) for a number texture pairs. Other parameters: δ = 12 & 24, ε = 6, σ = 5.5 & 11.5, R = 20

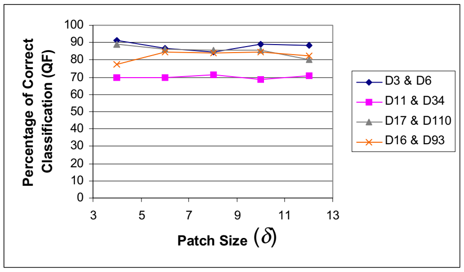

Fig. 4. Quality of eigenimage texture classification as a function of the patch size (δ) for a number of Brodatz texture pairs. The patch size shown is for 1st scale. In the 2nd scale the patch size is double. Other parameters: N = 10000, ε = 6, σ = half of patch size in 1st scale, R = 20

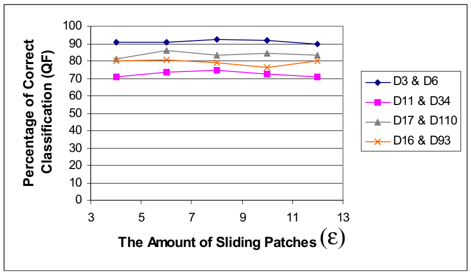

Fig. 5. Quality of eigenimage texture classification as a function of the overlap (ε) for a number of Brodatz texture pairs. Other parameters: N = 10000, δ = 12 & 24, σ = 5.5 & 11.5, R = 20

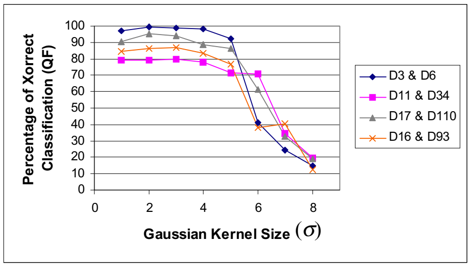

Fig. 6. Quality of eigenimage texture classification as a function of the initial blurring (σ) for a number of Brodatz texture pairs. Other parameters: N = 10000, δ = 12 & 24, ε = 6, R = 20

The Gaussian kernel size (σ) is more influential. Particularly, it should not be too big compared with the grain of the texture, and different texture pairs seem to have a different value of the optimal kernel size.

All in all, the most important parameter is the number of different scales taken into account, and to a lesser degree the amount of pre-processing (σ). However, the little amount of influence from the training set deserves further analysis.

5. SENSITIVITY TO TRAINING SET

A surprising fact can be noticed when looking at the eigenimages produced by the process developed in the previous paragraphs, as depicted in Figure 7. The eigenimages are "trained" to discriminate between two specific textures. We remember (see Figure 1) that the eigenimages for a single texture carry a lot of information about that specific texture. However, when we look at the results in Figure 7, we notice that the specificity to the particular textures being considered is almost gone! Most of the eigenimages are plain derivatives of the Gaussian, just as was observed for a non-textured image in [1].

The question arises then as to how sensitive the proposed algorithm is to the training for a specific pair of textures. To assess this sensitivity, we took the transform matrix A for the least-performing pair (D11 and D34: QF = 74%) and used that for the classification of the best-performing pair (D3 and D6: QF = 99%). If the results are dependent on the training for a specific pair, the results we get should be close to or worse than 74%. If the results only depend on the input pair and not on the training, the results should be close to 99%. In practice, the results are 99% indeed, showing that the results do not depend on the training set.

If this is the case, we could just follow the reasoning in [1] and use the first derivatives of the Gaussian as eigenimages to compose transformation matrix A. We did a test with this matrix, and the results are plotted in Figure 8, showing ever better performance than with training!

As a result, we can skip the whole (computationally expensive and time-consuming) training altogether and just work with Gaussian derivatives as characteristic features to do the texture classification.

One might wonder how the results of the un-trained process can be better than those of the trained process. The answer lies in the statistical significance of the training set. A larger training set would probably make the eigenimages converge more and more to the Gaussian derivatives. Once we know that, we can skip this step and go immediately to the results that would be produced with an infinite training set, namely the Gaussian derivatives.

6. DISCUSSION AND CONCLUSION

Humans are very efficient in discriminating textures, even at different resolution levels. Research about the human early vision system shows that basic preprocessing is done at the level of the retina by "convolving" the observed intensities with Gaussian derivatives of different sizes [1]. Human texture analysis must thus be based on uncommitted (un-trained) Gaussian-shaped texture analysis.

Our experiments show that it is possible to build a computer-based algorithm doing just the same: uncommitted (un-trained) multiresolution texture analysis by transforming the image patches according to a linear transformation matrix made up of Gaussian derivative patches. The derivatives to be used are arranged according to increasing order of the derivative, and not many derivatives are necessary to come to an efficient system - typically up to 1st or 2nd order (3 to 6 features per resolution level).

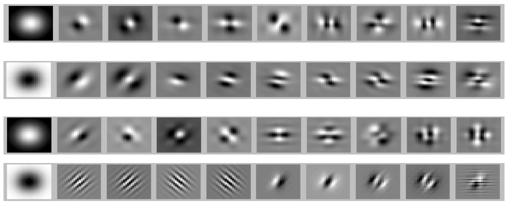

Fig. 7. Ten first eigenimages for the four pairs of images from the Brodatz album used in this article. $N = 10000$, $\delta = 24$, $\varepsilon = 6$, $\sigma = 5$

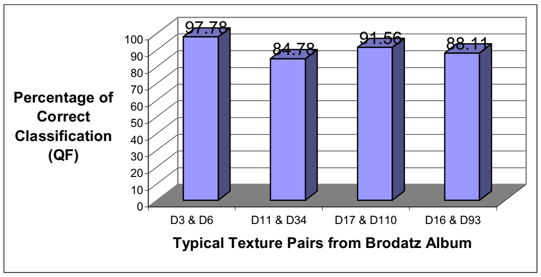

Fig. 8. Quality of eigenimage texture classification with Gaussian derivatives as eigenimages. Parameters: $\delta_1 = 12$, $\delta_2 = 24$, $\sigma_1 = 5.5$, $\sigma_2 = 11.5$, R = 20 (N and ε were only used for training and hence do not play a role anymore)

Although the present article has shown the feasibility of the uncommitted multiresolution eigenimage approach to texture analysis, further work will involve: 1) the optimization of the algorithms, 2) the benchmarking of it in comparison with other methods for texture analysis, and 3) the application of this method in real-world situations.

REFERENCES

[1] Bart M. ter Haar Romeny, *Front-end vision and multiscale image analysis*, (Kluwer Academic Publishers, 2002).

[2] A. Materka and M. Strzelecki, “Texture analysis methods-A review”, Technical University of Lodz, Institute of Electronics, COSTB 11 report, Brussels, 1998.

[3] A. Baraldi and F. Parmiggiani, “An investigation of texture characteristics associated with gray level co-occurrence matrix structural parameters”, IEEE Trans. on Geoscience and Remote Sensing, Vol. 33, No. 2, pp. 293-304, 1995.

[4] R.M. Haralick, “Statistical and structural approaches to texture”, Proceedings of the IEEE, Vol. 7, No. 5, pp. 786-804, May 1979.

[5] L. Van Gool, P. Dewaele and A. Oosterlinck, “A review of texture analysis anno 1983”, CVGIP, Vol. 29, pp. 336-357, 1985.

[6] A.K. Jain and F. Farrokhnia, “Unsupervised texture segmentation using Gabor filters”, Pattern Recognition, Vol. 24, No. 12, pp. 1167-1186, 1991.

[7] T.R. Reed and J.M. Hans du Buf, “A review of recent texture segmentation and feature extraction techniques”, VGIP: Image Understanding, Vol. 57, No. 3, pp. 359-372, 1993.

[8] S. Krishnamachari and R. Chellappa, “Multi-resolution Gauss-Markov random field models for texture segmentation”, IEEE Trans. on Image Processing, Vol. 6, No. 2, pp. 251-267, 1997.

[9] B. Chandhuri, N. Sarkar, “Texture segmentation using fractal dimensions”, IEEE Trans. on PAMI, Vol. 17, No. 1, pp. 72-77, 1995.