Pattern Retrieval in Image Databases Based on Adaptive Signal Decomposition

W. Y. Ma

Dept. of Electrical
and Computer Engineering
University of California
Santa Barbara, CA 93106
e-mail: wei@chandra.ece.ucsb.edu

B. S. Manjunath

Dept. of Electrical
and Computer Engineering
University of California
Santa Barbara, CA 93106
e-mail: manj@ece.ucsb.edu

Abstract

Efficient retrieval of information from image and other multimedia databases is an important and challenging problem. Among techniques developed for this purpose, query by example is the most effective way for searching large collections of images. In this paper we propose a new approach to extracting image features based on the local intensity pattern. The user can select any region of interest within an image and request that the system retrieve all similar looking patterns. No domain specific or context information is used in analysis, and the feature representation has desirable properties such as reduced dimensionality, preserving perceptual similarity, and offering a natural hierarchical indexing structure. Experimental results are presented.

1 Introduction

Fast retrieval of image data from large databases is an interesting and challenging problem. Of particular interest are techniques which allow users to query the database by picture content, i.e., query by picture example. The user identifies a region of interest in an image and requests that the system retrieve all similar looking patterns. The region of interest could be a vegetation patch in a satellite image, or a tumor in a brain MR scan. The input information is thus a rectangular region of pixel intensities, and the desired result is a set of images (or subimages) from the database which are perceptually similar to the given pattern.

There are two issues in image database search: one is that of pattern recognition, and the other related to search and indexing. Image analysis for recognition is an extremely difficult and an open research issue. Fortunately, in most image database applications one is often not interested in an exact match but in retrieving a small number of images which the user can browse through, and select or modify the search criteria. On the other hand, issues related to search time can not be ignored in a database context. Traditional alphanumeric databases could be ordered, indexed, and searched for matching patterns in an efficient manner. However, such techniques are in principle inadequate for image data.

Computation cost associated with image pattern retrieval has two main components: one associated with the preprocessing of the image to extract feature information; and the other with the actual search in the feature space to identify similar patterns. Feature vectors derived from the intensity patterns are typically of large dimensions (of the order of few hundreds), and conventional indexing mechanisms do not generalize well to such large dimensions. Although sequential search has been found to be adequate for small databases (consisting of few thousand images), there is a need to develop new strategies for larger databases.

1.1 Related Work on Pictorial Queries

Pictorial queries can be broadly classified into three categories based on the level of image analysis. At the low level, which requires no domain specific or context information, features could include color, histogram [1], and texture information [2][3]. At the intermediate level shape primitives are often used [2][4][5]. Preprocessing here requires robust segmentation to detect region boundaries. Domain specific information could be used in applications such as face recognition [6] and satellite image data [7][8]. An intelligent visual database system will have all three components of visual processing in addition to textual annotations and non-image specific information integrated into the query system.

In this paper an approach to extracting feature information from an image pattern that transforms it into a low dimensional feature vector is presented. The concept of image feature dictionaries is introduced in Section 2, wherein a small subset of filters from a large dictionary are used to represent the pattern information. This effectively

1058-6393/95 $4.00 © 1995 IEEE

partitions the data into different classes and makes the search process more efficient. The feature representation scheme is explained in Section 3, and experimental results using textures from the Brodatz album are presented in Section 4. We conclude with discussions in Section 5.

2 Filter Dictionary for Pattern Representation

2.1 Review of Matching Pursuit Algorithm

In many signal and image processing applications one is concerned with decomposing the signal into a linear expansion using a set of basis functions. The coefficients of this expansion characterize the signal. Examples include the Fourier representation and wavelet transformations. In recent years, wavelets have found many useful applications in image compression and segmentation. However, no single basis set can characterize all image patterns adequately, and this observation is the basis for developing the concept of waveform dictionaries for adaptive signal decomposition.

The basic idea is to start from a large and redundant set of functions in the dictionary. A given pattern is then represented as a linear combination of a subset of these functions (or waveforms). Selection of this subset is adaptive and best matches the given signal in some sense. In applications such as data compression, the commonly used criterion is minimizing the reconstruction error. The matching pursuit algorithm [9] is one such algorithm which selects the subset of functions to minimize this error.

The matching pursuit algorithm successively approximates the given signal with orthogonal projections onto the waveforms in a given dictionary. For example, if the given dictionary consists of a set of functions $\{g_i\}$ appropriately quantized in time and frequency. Let f be the signal of interest. Then f can be decomposed into

$$f = \langle f, g_i \rangle g_i + Rf \qquad (1)$$

where $\langle f, g_i \rangle$ is the inner product and Rf is the residual after approximating f using g_i, and is orthogonal to g_i. To minimize $|Rf|^2$ one must choose g_i such that $|\langle f, g_i \rangle|^2$ is maximum. The matching pursuit algorithm iteratively decomposes Rf by projecting it into the family of $\{g_i\}$ and selecting the functions which best matches it. Thus it is similar to the projection pursuit algorithms that were developed in the statistical literature. One can also relate this to some of the vector quantization techniques.

While the concept of filter dictionaries is very appealing, for images of even modest dimensions the associated computational complexity can be extremely large. Unlike data compression algorithms, in classification and recognition tasks one is not as much interested in reconstruction. We suggest a simple adaptation of this strategy for encoding image pattern using a set of Gabor functions in the filter dictionary.

2.2 Image Feature Dictionary

In this paper the Gabor functions are chosen to construct the image feature dictionary for computing the localized feature information. These functions are often used in modeling orientation selective and spatial frequency selective receptive field properties of cortical cell. They have also been proved to be optimal in the sense of minimizing the joint two-dimensional uncertainty in space and frequency [10]. This localization property has found to be very useful in many image analysis problems such as texture segmentation [11] and human face recognition [12]. Gabor functions are Gaussians modulated by complex sinusoids. In two-dimensions they take the form:

$$G(x, y) = g(x', y') \cdot \exp[2\pi j(Ux + Vy)] \qquad (2)$$

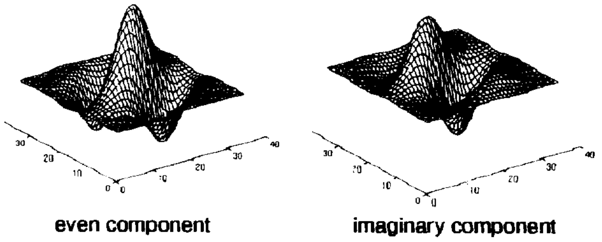
where $(x', y') = (x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)$ and

$$g(x, y) = \left(\frac{\lambda}{2\pi\sigma^2}\right) \cdot \exp\left(\frac{\lambda^2 x^2 + y^2}{2\sigma^2}\right) \qquad (3)$$

where λ is the spatial aspect ratio and θ is the rotation angle of the major axis of Gaussian. In the following discussion we assume that $\lambda = 2$ and consider the orientation of Gaussian envelope to be the same as the complex sinusoids with $\theta = \mathrm{atan}(V/U)$. Thus we have:

$$G(x, y) = g(x', y') \cdot \exp(2\pi jWx') \qquad (4)$$

where $W = \sqrt{U^2 + V^2}$. Figure 1 shows the 3-D profiles of the real (even) and imaginary (odd) components of a Gabor function.



even component

imaginary component

FIGURE 1. 3-D profile of the real (even) and imaginary (odd) components of a Gabor function

Gabor functions form a complete but non-orthogonal basis set and any given function $f(x, y)$ can be expanded in terms of these basis functions. This expansion provides a localized frequency description. The region of support of the functions in (4) adjusts with the frequency of the modulating sinusoids: The family of the functions in the dictionary are thus obtained by appropriate dilations and rotations of $G(x, y)$ through the generating function

$$G_{m\theta}(x, y) = a^{-m} G(x', y') \quad (5)$$

$$x' = a^{-m}(x\cos\theta + y\sin\theta)$$

$$y' = a^{-m}(-x\sin\theta + y\cos\theta)$$

where $a > 1$ and m is an integer. Note that the frequency spectrum of interest must be adequately covered by the entire dictionary. The scale factor a^{-m} in (5) is meant to ensure that all the functions in the dictionary have equal energy. The image feature dictionary used in the experiments consists of 120 different functions (10 distinct scales and 12 orientations at each scale).

3 A Compact Feature Representation Scheme

Here we introduce a simple algorithm for converting the image pattern into a very low dimensional feature representation. This algorithm consists of two stages. The first one is computing the projection of local intensity pattern onto all the functions in the dictionary at each image pixel location, and the second stage is based on selecting a subset of the functions which best match the local intensity pattern.

Consider $I(x, y)$ as the given image pattern. In the first stage the image pattern is processed through all the functions in the image feature dictionary, and the energy associated with function $G_{m\theta}$ at each pixel location (x, y) is computed:

$$\eta_{i,j}(x, y) = \left|I(x, y) * G_{m_i\theta_j}(x, y)\right| \quad (6)$$

where $i \in \{1, 2, \ldots, M\}$ and $j \in \{1, 2, \ldots, N\}$. M and N are the total number of different scales and orientations in the dictionary respectively. One can visualize this step as projecting the local intensity pattern of $I(x, y)$ onto the functions defined in the dictionary. This projection energy basically tell us how well the associated function matches the local intensity pattern. Then we select those functions with largest energy as the component of the feature vector at the corresponding pixel location. This selection strategy is conducted only within the functions at the same scale. So it is equivalent to finding the orientation which characterizes best the information at a given scale at that location. Thus it differs from the matching pursuit algorithm in that each scale is individually analyzed, and residuals are not expanded iteratively. Let us denote the selected function at location (x, y) and scale m_i as $J_i(x, y) \in \{1, 2, \ldots, N\}$ Then,

$$J_i(x, y) = \arg\max_j \eta_{i,j}(x, y) \quad (7)$$

In the second stage, all the local features are combined together to obtain one simple representation for the entire image pattern. Let $k(i, j)$ be the number of times the function $G_{m_i\theta_j}$ is selected:

$$k(i, j) = \sum_{x,y} \delta(J_i(x, y) - j) \quad (8)$$

Then, for each scale i we order $k(i, j)$ such that

$$k\left(i, j_i^{(1)}\right) \geq k\left(i, j_i^{(2)}\right) \geq \ldots \geq k\left(i, j_i^{(N)}\right)$$

In the experiments we keep only the top two most frequently chosen functions to construct feature vectors $\vec{\Theta}^{(1)}$ and $\vec{\Theta}^{(2)}$, which are defined as

$$\vec{\Theta}^{(1)} = \left[j_1^{(1)}, j_2^{(1)}, \ldots, j_M^{(1)}\right]$$

$$\vec{\Theta}^{(2)} = \left[j_1^{(2)}, j_2^{(2)}, \ldots, j_M^{(2)}\right]$$

The average of the energy associated with them is also used to form two other feature vectors $\vec{\zeta}^{(1)}$ and $\vec{\zeta}^{(2)}$ which are organized as

$$\vec{\zeta}^{(1)} = \left[E\left(\eta_{1,j_1^{(1)}}\right), E\left(\eta_{2,j_2^{(1)}}\right), \ldots, E\left(\eta_{M,j_M^{(1)}}\right)\right]$$

$$\vec{\zeta}^{(2)} = \left[E\left(\eta_{1,j_1^{(2)}}\right), E\left(\eta_{2,j_2^{(2)}}\right), \ldots, E\left(\eta_{M,j_M^{(2)}}\right)\right]$$

where $E(\eta_{i,j})$ is defined by

$$E(\eta_{i,j}) = \frac{1}{k(i,j)} \sum_{x,y} \eta_{i,j}(x, y) \cdot \delta(J_i(x, y) - j). \quad (9)$$

Thus for each image pattern, the feature representation will consist of four set of vectors $\vec{\Theta}^{(1)}$, $\vec{\Theta}^{(2)}$, $\vec{\zeta}^{(1)}$ and $\vec{\zeta}^{(2)}$.

Let $[\vec{\Theta}_a^{(1)}, \vec{\Theta}_a^{(2)}, \vec{\zeta}_a^{(1)}, \vec{\zeta}_a^{(2)}]$ and $[\vec{\Theta}_b^{(1)}, \vec{\Theta}_b^{(2)}, \vec{\zeta}_b^{(1)}, \vec{\zeta}_b^{(2)}]$ be the two feature representations extracted from different images. We first define

$$D(i, 1) = d\left[\vec{\Theta}_a^{(1)}(i), \vec{\Theta}_b^{(1)}(i)\right] + d\left[\vec{\Theta}_a^{(2)}(i), \vec{\Theta}_b^{(2)}(i)\right],$$

$$D(i, 2) = d\left[\vec{\Theta}_a^{(1)}(i), \vec{\Theta}_b^{(2)}(i)\right] + d\left[\vec{\Theta}_a^{(2)}(i), \vec{\Theta}_b^{(1)}(i)\right]$$

where index i means the ith component of the vector and the operator d is modulo N (N is the number of orientations). The distance measure between two feature representations using just the features $[\vec{\Theta}^{(1)}, \vec{\Theta}^{(2)}]$ is defined by

$$s(a, b) = \sum_{i=1}^{M} \min[D(i, 1), D(i, 2)]. \quad (10)$$

The smaller the value $s(a, b)$ is, the more similar the two feature representations are. This measure is first used to group the image patterns in the database into different classes and the distance between features $[\vec{\zeta}^{(1)}, \vec{\zeta}^{(2)}]$ is further used to distinguish the image patterns in the same class. This measure is similar to the previous for features $[\vec{\Theta}^{(1)}, \vec{\Theta}^{(2)}]$ except that the operator d is the euclidean distance. Figure 2 shows the schematic diagram of the retrieval process.

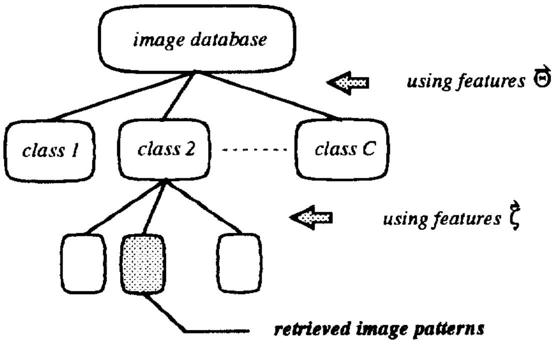


FIGURE 2. The schematic diagram of the retrieval process using the proposed feature representation scheme.

4 Experimental Results

The objective of the experiments is to illustrate that the feature representation preserves perceptual similarity between patterns in the feature space while facilitating the search process. The image data used in the experiments are from the Brodatz album. There are fifty-six different 512 x 512 Brodatz texture images in our system. Each of these 512 x 512 images is further divided into sixteen 128 x 128 subimages, thus constructing a database of 896, 128 x 128 images. Each of these images is encoded using the feature representation described in the previous section. We arbitrarily select one image as the test pattern and compute the similarity measure with other image patterns in the database. The top 20 most similar ones are retrieved and displayed. Figure 3 shows some of the results. The upper left is the test pattern, and the retrieved 20 patterns are displayed in row scan order with decreasing similarity. Notice that most of the retrieved patterns are from the same texture as the test pattern.

In our experiments, we chose to retain only the top two orientations at each scale for each pattern. If a pattern has more than two distinct orientations at each scale, it affects the retrieval accuracy. If all the 120 filters are used one can obtain close to 100% classification in the sense that the majority of the top 20 retrieved patterns all belong to the same texture.

5 Discussions

This paper describes an image feature extraction strategy for image data retrieval which allows queries to be composed of local intensity patterns. No domain specific or context information is used in this image feature analysis. In developing this algorithm the focus was not only on pattern recognition but also on issues related to indexing and search. Thus, the proposed feature representation in addition to preserving the perceptual similarity, also has some nice properties such as reduced dimensionality and a hierarchical structure for designing efficient search algorithms. In the current implementation the feature representation consists of 4 sets of feature vectors, each with dimensionality 10, and the search process sequential. However, indexing structures such as R-tree can be used to create a more efficient retrieval system.

The feature representation strategy in the paper is based on selecting a subset of functions which best characterize the local intensity pattern of the given image, thus it does not depend on the database contents as much. Currently we are exploring a new filter selection strategy which not only considers the query image pattern but also uses the information about pattern distribution in the database. The associated feature representation is computed in parallel with the search process, and each feature component will be used to eliminate non-candidate patterns in each search cycle. Preliminary experimental results using this new strategy are very encouraging.

Acknowledgments: This work was partially supported by grants from NASA and NSF/UCSB Alexandria Project on Digital Libraries.

6 References

[1] F. Arman, A. Hsu, and M. Y. Chiu, "Feature management for large video databases," in Proc. SPIE, Storage and Retrieval for Image and Video Databases, vol. 1908, (San Jose, CA), pp. 2-12, February 1993.

[2] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: tools for content based manipulation of image databases," in proc. SPIE conference on storage and retrieval of image and video databases -II, Vol. 2185, pp. 34-47, San Jose, February 1994.

[3] B. S. Manjunath, W. Y. Ma, "Information Retrieval in Image Databases," in Int. Conf. on Distributed Multimedia Systems and Applications, IASTED/ISMM, pp. 6-9, Honolulu, August, 1994.

[4] H. V. Jagadish, "A retrieval technique for similar shapes," in Int. Conf. on Management of Data, SIGMOID '91, pp. 208-217, Denver, CO, May 1991.

[5] R. Mehrotra, J. E. Gary, "Feature-based retrieval of similar shapes," in Proc. 9th Data Engineering Conference, (Vienna, Austria), pp. 108-115, April 1993.

[6] J. R. Bach, S. Paul, and R. Jain, "A visual information management system for the interactive retrieval of

faces," IEEE Trans. Knowledge and Data Engineering, vol. 5, No. 4, pp. 619-628, August 1993.

[7] R. Samadani, C. Han, L. K. Katragadda, "Content-based event selection from satellite images of the aurora," in Proc. of SPIE, Storage and Retrieval for Image and Video Databases, (San Jose, CA), vol. 1908, pp. 50-59, February 1993.

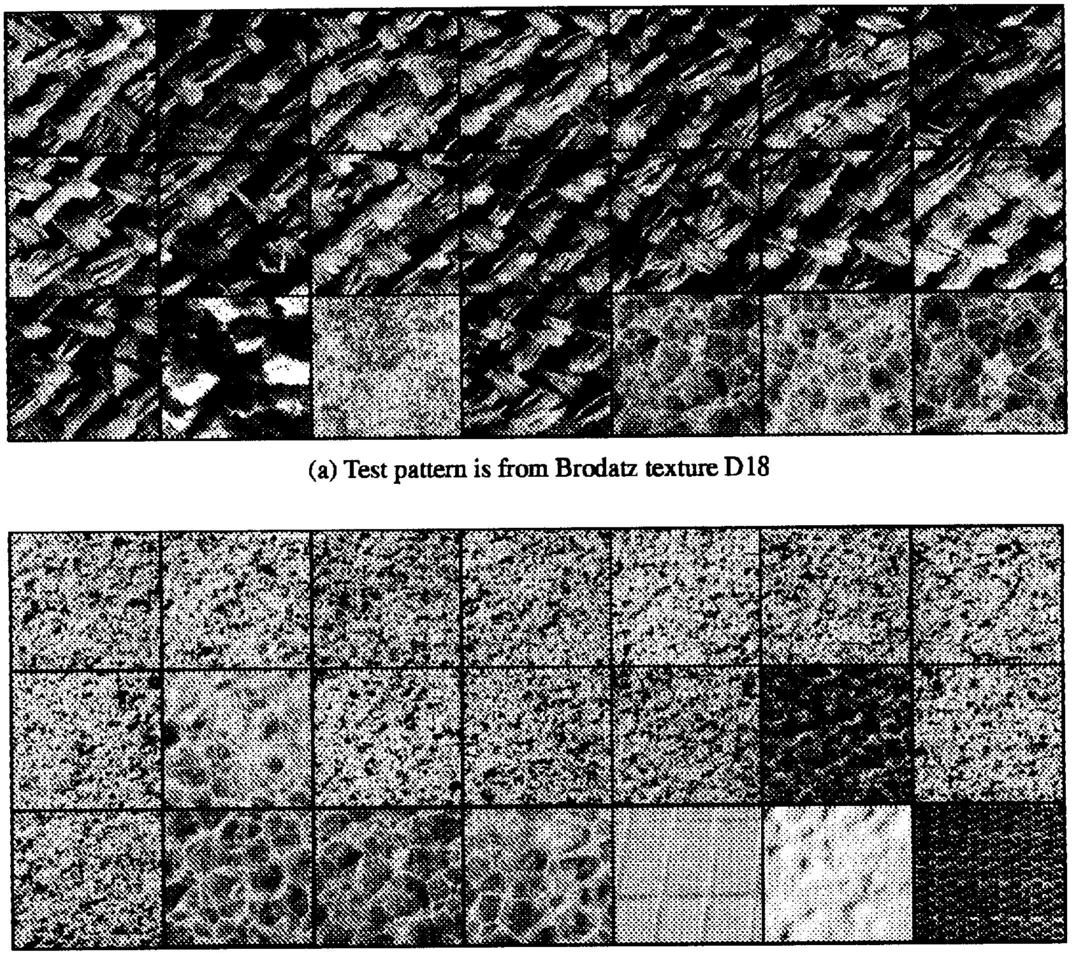
[8] A. Kitamoto, C. Zhou, M. Takagi, "Similarity retrieval of NOAA satellite imagery by graph matching," in Proc. of SPIE, Storage and Retrieval for Image and Video Databases, vol. 1908, (San Jose, CA), pp. 60-73, February 1993.

[9] S. Mallat, Z. Zhang, "Matching pursuits with time-frequency dictionary," in IEEE Trans. on Signal Processing, vol. 41, no. 12, pp. 3397-3415, December 1993.

[10] J. G. Daugman, "Relaxation neural network for non-orthogonal image transforms," in Proc. Int. Conf. on Neural Networks, vol. 1, (San Diego, CA), pp. 547-560, June 1988.

[11] B. S. Manjunath, R. Chellappa, "A unified approach to boundary detection," in IEEE Trans. Neural Network, vol. 4, No. 1, pp. 96-108, January 1993.

[12] B. S. Manjunath, R.Chellappa, "A feature based approach to face recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition '92, (Champaign, IL), pp. 373-378, June 1992.



(a) Test pattern is from Brodatz texture D18

(b) Test pattern is from Brodatz texture D29

FIGURE 3. The upper left image is the test pattern and the top 20 most similar patterns in the database are retrieved and displayed in row scan order with decreasing similarity.