Submitted to Image and Vision Computing, Elsevier, 2004

Texture Analysis and Indexing Using Gabor-like Hermite Filters

Carlos Joel Rivero-Moreno Stéphane Bres
LIRIS, FRE 2672 CNRS, Lab. d'InfoRmatique en Images et Systèmes d'information
INSA de Lyon, Bât. Jules Verne, 17 av. Jean Capelle
Villeurbanne Cedex, 69621 FRANCE

+33.4.72.43.63.72 +33.4.72.43.80.93

rivero@rfv.insa-lyon.fr stephane.bres@insa-lyon.fr

ABSTRACT
In this paper, we study texture discrimination based on two filter families, Gabor and Hermite, which agree with the Gaussian derivative model of the human visual system. In the first part, discrimination of different textures, based on the output energy of these filters, is compared using the Fisher criterion and classification result. Results show that the presented filter bank is suitable for purposes of texture feature extraction and discrimination. The second part of this paper is dedicated to global and local texture feature extraction for image indexing and retrieval based on Hermite filters.

Keywords
Texture, Hermite transform, Gaussian derivatives, Gabor filters, feature extraction, indexation, image retrieval, discrimination, Fisher criterion.

1. INTRODUCTION
An important task in image processing is texture analysis [1]. It allows to achieve one of demanded applications in computer vision: image indexing and retrieval based on textural information [2] [3] [4] [5]. Without losing generality, two stages are required. The first one consists of assigning a texture feature vector to each pixel in an image. The feature extractor, which is usually based on filtering the textured image [6] [7], must be then able to discriminate among several textures. The goal of the second stage is indexing based on the extracted features vectors. In order to allow dimensionality reduction, the feature vectors are also combined to provide a new index of texture features as a quantitative description which is useful for image searching and retrieval by means of a similarity measure.

Texture feature extraction lies essentially in image processing techniques. It is usually performed by linearly transforming or filtering the textured image [8] [6] [9] [7] followed by some energy measure or non-linear operator (e.g. rectification). In this work, we would like to use a texture feature extractor that approximates the human visual system (HVS), in such a way that impulse responses of the filters are equivalent to those of the receptive field profiles (RFPs) of simple cells in the visual cortex.

In this paper, we focus on the multi-channel filtering (MCF) approach. It is inspired by the MCF theory for processing visual information in the early stages of the HVS [7] [9], where RFPs of the visual cortex can be modeled as a set of independent channels, each one with a particular orientation and frequency tuning. It then involves the decomposition of an input image into multiple features images by filtering. Each such an image captures textural features occurring in a narrow band of spatial frequency and orientation. This approach has the property of exploiting spatial interactions between the pixels of a neighborhood at different scales.

Among the MCF models having the above properties, Gabor filters [10] have been widely used in texture feature extraction [11] [6] [7] and image indexing and retrieval [2] [3] [5]. Another model corresponds to Hermite filters of the Hermite transform [12] that agree with the Gaussian derivative model of the HVS [13]. It has also been shown analytically that Hermite and Gabor filters are equivalent models of RFPs of the HVS [14] [15] and they well match cortical data. However, Hermite filters have some advantages over Gabor ones, like being an orthogonal basis leading to perfect image reconstruction after decomposition and it has a discrete representation and filter separability which allows efficient implementation. Furthermore, they can be implemented as both scale-space and pyramid representations because of their agreement with Gaussian derivatives. Thus, they are also suitable for detecting shape features such as edges, bars, and corners; something that differs from the Gabor multi-scale representation. Despite these advantages, Hermite filters are not used as much as Gabor filters for texture feature extraction and, as we will show, cartesian Hermite filters do not perform an equivalent texture discrimination like Gabor filters. This fact is due to their definition in the frequency domain. Therefore, in order to increase texture discrimination and maintain the equivalence of the two models in this aspect, we introduce the Gabor-like Hermite filters, which correspond to a modified version of steered Hermite filters [16]. They are designed to have a frequency coverage similar to Gabor filters.

In this work, we use the output energy of filters in the bank as texture features. These responses are closely related to the local power spectrum. Texture discrimination is measured following the methodology proposed by Grigorescu *et al.* [11]. It uses Fisher linear discriminant analysis [17]. The interest of using such an approach is that discrimination is decoupled from classification (which arises often in segmentation) and it is then purely measured on statistics of two projected clusters of texture feature

vectors. Indeed, the Fisher criterion measures the maximum separability of two concerned clusters in the reduced space. We also apply a supervised classification to segment some textured images in order to compare the similarity discrimination between Gabor and Gabor-like Hermite filters from a visual viewpoint.

Almost all proposed texture image retrieval systems in the literature, where indexing is based on extracted texture features, deal with uniform texture images, i.e., each of the images in the database is supposed to represent only one texture. A query texture pattern can be any texture pattern, belonging or not to the database. Such indexes represent globally the texture in the whole image. It thus corresponds to a *global* texture feature extraction and indexing. In cases where the analyzed image is composed of several textures forming uniform regions, then this approach for image browsing and retrieval will not work. Since in such cases texture depends on distinct spatial locations, then a *local* texture feature extraction and indexing is required. A similarity matching between extracted features at different spatial locations and features of the query pattern can therefore be applied to search for similar looking regions. We present in this paper both global and local approaches of texture feature extraction and indexing.

Our contributions can be summarized as follows: we introduce a frequency design into the definition of cartesian Hermite filters to achieve a steerable version that yields Gabor-like Hermite filters and we show how this filter bank improves texture discrimination. We present a normalized recurrence relation for the Krawtchouk filters that allows fast computations. We present a dimensionality reduction approach based on relevant statistic parameters, which represents a compact texture model, by exploiting the joint statistics of Hermite coefficients. Finally, we introduce a local approach, based on the same texture feature extraction method, for images with several textured regions. This approach preserves the dimensionality reduction requirement and it is useful for texture image retrieval and location without correspondence.

The paper is organized as follows. In section 2, we show definitions of Gabor and cartesian Hermite filters, and we present the Gabor-like Hermite filter bank. In section 3, we describe the Fisher criterion and give results on texture discrimination. In section 4, we apply filters for supervised texture segmentation and discuss results. In section 5, we explain the parametric texture model used for indexing. In section 6, we present results and discussion of the global approach. Section 7, concerns the local approach. Section 8 contains final conclusions.

2. GABOR-LIKE HERMITE FILTER BANK

2.1 Cartesian Hermite Filters

Hermite filters $d_{n-m,m}(x,y)$ decompose a localized signal $l_v(x\text{-}p,y\text{-}q) = v^2(x\text{-}p,y\text{-}q)\ l(x,y)$ by a Gaussian window $v(x,y)$ with spread σ and unit energy, into a set of Hermite orthogonal polynomials $H_{n-m,m}(x/\sigma\ ,\ y/\sigma)$. Coefficients $l_{n-m,m}(p,q)$ at lattice positions $(p,q)\in P$ are then derived from the signal $l(x,y)$ by convolving with the Hermite filters. These filters are equal to Gaussian derivatives where $n–m$ and m are respectively the derivative orders in x- and y-directions, for n=0,…,D and m=0,…,n. Thus, the two parameters of Hermite filters are the maximum derivative order D (or polynomial degree) and the scale σ.

Hermite filters are separable both in spatial and polar coordinates, so they can be implemented very efficiently. Thus, $d_{n-m,m}(x,y) = d_{n-m}(x)\ d_m(y)$, where each 1-D filter is:

$$d_n(x) = \left((-1)^n \Big/ (\sqrt{2^n \cdot n!}\sqrt{\pi}\sigma)\right) H_n(x/\sigma) e^{-x^2/\sigma^2} \qquad (1)$$

where Hermite polynomials $H_n(x)$, which are orthogonal with respect to the weighting function $exp(-x^2)$, are defined by Rodrigues' formula [18] as:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2} \qquad (2)$$

In the frequency domain, these filters are Gaussian-like band-pass filters with extreme value for $(\omega\sigma)^2 = 2n$ [14] [15], and hence filters of increasing order analyze successively higher frequencies in the signal.

2.2 Krawtchouk Filters

Krawtchouk filters are the discrete equivalent of Hermite filters. They are equal to Krawtchouk polynomials multiplied by a binomial window $v^2(x) = C_N^x / 2^N$, which is the discrete counterpart of a Gaussian window. These polynomials are orthonormal with respect to this window and they are defined as [18] :

$$K_n(x) = \frac{1}{\sqrt{C_N^n}} \sum_{\tau=0}^{n} (-1)^{n-\tau} C_{N-x}^{n-\tau} C_x^{\tau} \qquad (3)$$

for x=0,…,N and n=0,…,D with $D \le N$.

It can be shown that the Krawtchouk filters of length N approximates the Hermite filters of spread $\sigma = \sqrt{N/2}$. In order to achieve fast computations, we present a normalized recurrence relation to compute these filters:

$$K_{n+1}(x) = \frac{1}{\sqrt{(N-n)(n+1)}} \Big[(2x-N) K_n(x) - \sqrt{n(N-n+1)} K_{n-1}(x) \Big] \quad , \quad n \ge 1 \qquad (4)$$

with initial conditions $K_0(x) = 1$, $K_1(x) = \frac{2}{\sqrt{N}} \left(x - \frac{N}{2} \right)$.

2.3 Steered Hermite filters

In order to have a MCF approach based on Hermite filters, they must be adapted to orientation selectivity and multi-scale selection. For doing this, we apply their property of steerability [16]. The resulting filters then may be interpreted as directional derivatives of a Gaussian (i.e. the low-pass kernel).

Since all Hermite filters are polynomials times a radially symmetric window function (i.e. a Gaussian), it is easy to prove that the n+1 Hermite filters of order n form a steerable basis [19] for every individual filter of order n. More specifically, rotated versions of a filter of order n can be constructed by taking linear combinations of the filter of order n. The Fourier transform of Hermite filters $d_{n-m,m}(x,y)$ can be expressed in polar coordinates

$\omega_x = \omega\cos\theta$ and $\omega_y = \omega\sin\theta$ as $\hat{d}_{n-m,m}(\omega_x,\omega_y) = \hat{d}_n(\omega)\alpha_{n-m,m}(\theta)$

where $\hat{d}_n(\omega)$, which expresses radial frequency selectivity, is the 1-D Fourier transform of the nth Gaussian derivative in (1) but with radial coordinate r instead of x. The cartesian angular functions of order n for m=0,…,n, are given as

$$\alpha_{n-m,m}(\theta) = \sqrt{C_n^m}\cos^{n-m}\theta\cdot\sin^m\theta \qquad (5)$$

which express the directional selectivity of the filter.

Steered coefficients $l_n(\theta)$ resulting of filtering the signal $l(x,y)$ with these steered filters can be directly obtained by steering the cartesian Hermite coefficients $l_{n-m,m}$ as:

$$l_n(\theta) = \sum_{m=0}^{n} l_{n-m,m}\cdot\alpha_{n-m,m}(\theta) \qquad (6)$$

2.4 Gabor-like Hermite Filters

In order to turn the steered Hermite filters into a MCF bank, we construct a multi-scale representation that fulfils the desired constraints in the frequency domain, which are mainly the number of scales S (radial frequencies ω_0) and the number of orientations R in the filter bank. Since previous works have been done essentially with Gabor filters, we have then adopted a similar multi-channel design. Moreover, both Hermite and Gabor filters are similar models of the RFPs of the HVS [14] [15]. For these reasons, we have named the resulting filters as Gabor-like Hermite filters.

The strategy design in the frequency domain is the same as that presented in [2] [3]. It is to ensure that the half peak magnitude supports of the filter responses in the frequency spectrum touch each other. Let $g(x,y)$ be a Gabor-like Hermite filter. Then, its scaled and oriented versions $g_{s,r}(x,y)$ are given by:

$$g_{s,r}(x,y) = a^{-s}g(x',y') \quad , \quad a>1 \quad , \quad s,r\in\mathbb{N} \qquad (7)$$

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

where $\theta = r\pi/R$, r=0,…,R–1, s=0,…,S–1. The scale factor a^{-s} in (7) is meant to ensure that the energy is independent of scale s. Let σ_x and σ_u be, respectively, the spatial and frequency spreads of a 1-D Hermite filter as defined in (1) which has radial frequency selectivity ω_0. Let f_l and f_h denote the lower and upper normalized center frequencies (between 0 and ½) of interest for the MCF bank. Then, for each scale, parameters a , σ_x , σ_u , and ω_0 of each channel are computed as:

$$\omega_0 = 2\pi f_0 \ , \ \sigma_x = 1/(2\pi\sigma_u) \ , \ f_0 = a^{-s}f_h$$
$$a = (f_h/f_l)^{\frac{1}{S-1}} \ , \ \sigma_u = \frac{(a-1)f_0}{(a+1)\sqrt{2\ln 2}} \qquad (8)$$

Gabor-like Hermite filters already have zero mean (null DC). The (discrete) Krawtchouk filters are linked to Hermite filters by these parameters as:

$$N = \left\lceil 2\sigma_x^2 \right\rceil \quad \text{and} \quad D = \left\lceil (\sigma_x\omega_0)^2/2 \right\rceil \qquad (9)$$

where $\lceil\cdot\rceil$ rounds to the nearest integer whereas $\lceil\cdot\rceil$ does it too but towards infinity. Notice that for each scale there is a set of parameters (N,D).

In summary, construction of a Gabor-like Hermite filter bank requires the following procedure. First of all, set the number of desired scales S and orientations R and for each of the scales s=0,…,S–1 compute :

- the radial central frequency ω_0 and the spatial spread σ_x of respective filters through (8).
- Krawtchouk's parameters such as window length N and filter order D through (9).
- Krawtchouk filters: get the corresponding Krawtchouk polynomials through (4) and multiply them by a binomial window of length N.
- Convolutions of input image with Krawtchouk filters to obtain cartesian coefficients.
- Steering coefficients to desired orientations through (6) and (5) to obtain the equivalent multi-channel outputs.

2.5 Real Gabor Filters

We use real Gabor filters to maintain the equivalence presented in [14] [15]. Nevertheless, complex Gabor filters can be used since the results in this paper remain valid. The design of the Gabor filter bank is the same as that presented in the previous subsection for the Gabor-like Hermite filter bank, and expressions (7) and (8) are applied. The only difference is the definition of $g(x,y)$ in (7). It corresponds to a real Gabor filter which is defined as:

$$g(x,y) = \left(1/(2\pi\sigma_x\sigma_y)\right)e^{-(x^2/\sigma_x^2+y^2/\sigma_y^2)/2}\cdot\cos(\omega_0 x) \qquad (10)$$

and the y-coordinate is involved into (8) with:

$$\sigma_y = 1/(2\pi\sigma_v) \ , \ z = -(2\ln 2)\sigma_u^2/f_0$$
$$\sigma_v = \tan\left(\frac{\pi}{2R}\right)\left[f_0+z\right]\left[2\ln 2 - z^2/\sigma_u^2\right]^{-1/2} \qquad (11)$$

For simplicity, the three filter banks, Gabor, Gabor-like Hermite and cartesian Hermite, are denoted respectively by G1, G2 and H1.

3. FISHER CRITERION-BASED DISCRIMINATION

The Fisher criterion measures the maximum separability of two projected clusters into a 1-D space, i.e. into the line which links the center of the concerned clusters in the multidimensional feature space. This one is achieved by a linear transform, W, applied to each of the feature vectors, x, belonging to the multidimensional space, and is called the Fisher linear discriminant (FLD) function. It is defined as:

$$y = W^T x \quad , \quad W = S^{-1}\left[\mu_1 - \mu_2\right] \qquad (12)$$

where y are the projected vectors into the 1-D space (resulting in scalars), μ_1 and μ_2 are the means of the two clusters, S^{-1} is the inverse of the pooled covariance matrix, S, of the two clusters.

This approach supposes that there are two clusters associated to two textures (i.e. two classes of textures to be discriminated). S, the within-class scatter matrix, is computed as the sum of two covariance matrices, $S=S_1+S_2$, one for each texture class. They are computed as:

$$S_k = \sum_{i=1}^{L} (x_i - \mu_k)(x_i - \mu_k)^T \qquad (13)$$

where k=1,2; x_i is the associated vector to pixel i within cluster k. L is the total number of pixels and $K=S \cdot R$ is the number of filters in the bank corresponding to the dimension of the feature space. Fisher criterion, J, is given as:

$$J = \frac{|\eta_1 - \eta_2|}{\sqrt{\sigma_1^2 + \sigma_2^2}} \qquad (14)$$

where η_1 and η_2 are the projections of the means μ_1 and μ_2, and σ_1^2 and σ_2^2 are the variances of the distributions of the projected feature vectors. J is the distance between two clusters relative to their size. The larger the value of J, the better the discrimination between two textures performed by the filters.

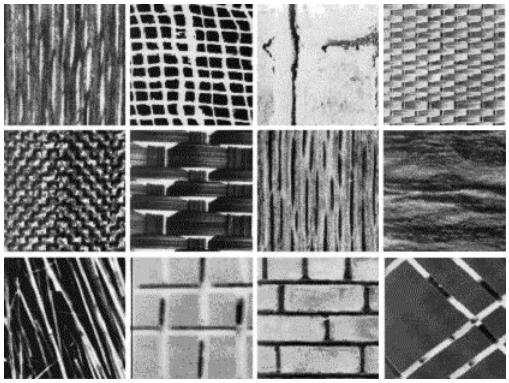

Figure 1. Set of texture images for computing Fisher criterion values.

Similar to [11], we have applied each of the three filter banks to each of the textures shown in fig 1. We have made our experiments on a well-known set of images by Brodatz [20]. We chose, for Gabor (G1) and Gabor-like Hermite (G2) filter banks, S=5 scales and R=4 orientations, which results in a bank of K=20 filters. By setting D=5, one obtains the same number of channels for the cartesian Hermite (H1) filter bank. This one is, in such a way, a normalization of the space dimension, so we can obtain vectors of the same dimension K. For simplicity, we have only taken the first L=1000 pixels from each image. We have computed, for M=12 textures (fig. 1), all pair-wise combinations of Fisher criterion. Thus, there are $M(M-1)/2$=66 Fisher criterion values. The statistics and distributions of these values, for each of the filter banks, are shown in the boxplot representation of fig. 2.

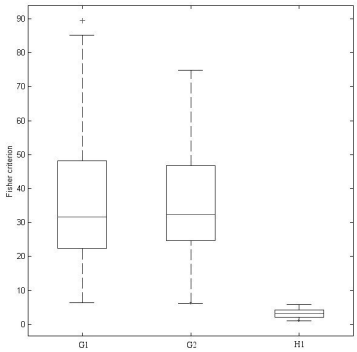

Figure 2. Boxplot representation of Fisher criterion values obtained with the three filter banks (from left to right: G1, G2, and H1).

The Fisher criterion results show that better cluster separation, i.e. better texture discrimination, is obtained from Gabor (G1) and Gabor-like Hermite (G2) filter banks. Indeed, from fig. 2, one can see that their discrimination is similar.

4. TEXTURE SEGMENTATION RESULTS

We have applied a supervised classification method, the k-means algorithm, to the texture feature vectors (which form vectors in the multidimensional feature space) obtained as energy outputs of each of the three filter banks G1, G2 and H1. In this case, k is the number of classes or textures in the images (k=5). The resulting classification, based on the texture discrimination performed by the extracted texture feature vectors, yields a segmented image. Fig. 3 shows two textured images, with five textures each one, and their segmented images. A classification similarity between G1 and G2 is expressed by the percentage of correctly classified pixels as shown in table 1.

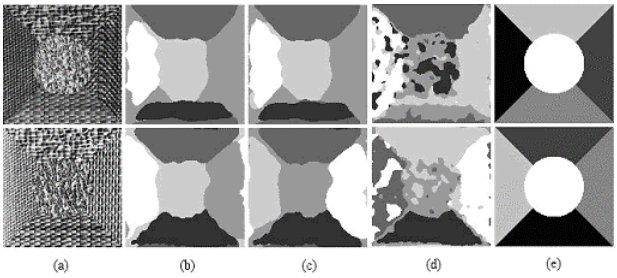

Figure 3. Two texture images and segmentation results with the k-means classification. (a) Original image with k=5 textures, (b) segmentation with G1, (c) segmentation with G2, (d) segmentation with H1, and (e) perfect segmentation.

Table 1. Percentage of correctly classified pixels using the k-means classification for the images in fig. 3.

	G1	G2	H1
Image 1	80.8655	80.9998	69.2383
Image 2	89.0503	86.9751	79.8401

Texture segmentation results confirm that G1 and G2 are equivalent models from a visual viewpoint and they perform similar classifications.

5. PARAMETRIC TEXTURE MODEL

The Gabor-like Hermite MCF-bank presented in the section 2 is applied to decompose a given textured image into a set of filtered images that represent the image information at different frequencies and at different orientations. Therefore, each of the channel outputs of the filter bank can be considered as one component of a texture feature vector of dimension $S \cdot R$. Thus, there are as much feature vectors as pixels in the image. For our application, we chose $S = 4$ scales and $R = 6$ orientations, which results in a bank of 24 filters. There is then an important dimensionality increasing which is $S \cdot R = 24$ times the original image size. Dimensionality reduction is thus an important goal in image indexing techniques, since one needs to store such indexes. Parametric texture models use combinations of parameters to characterize textures [21]. We only keep parameters which describe well the essential structure of texture. For this purpose, we have tested different combinations of parameters and we have found that the best results, for texture indexing, are obtained for only considering the spatial auto-correlation of coefficients of each subband.

Since auto-correlations imply a spatial lag from the central pixel in both x- and y-directions, we have then fixed it to $M=7$ as in [21]. It then represents, for each of the subbands, a dimensionality reduction which goes from the image size to $(M^2+1)/2$ parameters, since the spatial correlation is symmetric. It yields $S \cdot R \cdot (M^2+1)/2 = 600$ parameters, which is a significant dimensionality reduction. These parameters represent structures in images (e.g., edges, bars, corners).

Because of the multi-scale approach, there is a different analysis window size for each of the scales S. Lower scales are related to higher frequencies (fine levels) and thus window sizes are shorter. On the other hand, higher scales are related to lower frequencies (coarse levels) and thus window sizes are larger. The more the scale increases, the more the window size does too. In order to compare spatial interactions in coarser levels where smoothing is very high, we have subsampled the auto-correlations by a factor of two, in both directions, from a fine scale level to the next coarser level.

6. GLOBAL TEXTURE FEATURE EXTRACTION AND INDEXING

As we have already mentioned in the introduction of this work, our approach addresses both global and local texture feature extraction and indexing. In this section, indexes characterize globally the texture in the whole image and each of the images in the database represent only one texture.

6.1 Texture Database and Similarity Measure

In this study, texture under consideration are either gray-scale or luminance-based. We have made our experiments on a well-known set of images by Brodatz [20]. The texture database consists of 112 different texture images where each one has an image with size of 640x640 pixels. Each of the images is divided into 9 256x256 overlapping subimages, thus creating a database of 1008 images.

We have then applied the texture feature extraction approach described in the previous section to every subimage in the database. The resulting feature vectors were saved as texture image indexes. Thus, one has a texture-based retrieval system in which distances between the query pattern and patterns in the database are computed. We have used as distance $1-|\rho|$, where ρ is a normalized correlation coefficient in [-1;1]. It is computed as a normalized dot product. Let x_1 and x_2 be two column vectors of autocorrelation parameters (with length of 600) of two textures. The distance between them is then computed as:

$$d = 1 - \left| x_1^T x_2 \Big/ \left(\sqrt{x_1^T x_1} \sqrt{x_2^T x_2} \right) \right| \qquad (15)$$

In the ideal case all the top 9 retrievals, which have the lowest distance to the query, are from the same large image. We have measured the performance of the indexing system in terms of the average retrieval rate, which is defined as the average percentage number of patterns belonging to the same image as the query pattern in the top 9 matches (*average recall*). This one is shown in fig. 5, where the horizontal axis represents the number of retrieved images and the vertical axis represents the percentage retrieval performance. However, this way of assessing our indexing system is not quite adapted to the ground-truth. Subimages coming from the same large image can have a texture visually different.

6.2 Experimental Results and Discussion

The average retrieval rate (retrieval efficiency) according to the number of top matches considered is 81.29% for the top 9 retrievals. It means that, in average, about less than 2 textures among the 9 most relevant ones are apparently not well classified. However, in general, they have a visual similarity from the texture viewpoint (fig. 4: D88-6 and D89-9). The visual similarity (ground-truth) is considered by our texture indexing *method* whereas the retrieval efficiency, which is used as a system *assessment*, does not. Indeed, the latter only takes into account the correct matches which come from the same large image. Hence, the misclassifications due to the assessment are compensated by the visual similarity according to the following reasons:

(1) Some texture subimages which result from two different large images have a similar visual aspect (fig 4.: D88-6 and D89-9). The method is right in considering them as similar textures (9 top matches) whereas the system assessment is wrong in considering them as two different textures (because of two large images).

(2) Some texture subimages which result from the same large image have different visual aspects (fig 4.: D59-3 and D59-7). This is the case of images in the database which have large structures and their extracted subimages have no longer the same visual structures. The method is right in not matching them (different textures). However, the system assessment supposes that they have to be associated. On the other hand, the method performs well since it matches D59-3 with D88-6 in the 9 top (see fig 4).

(3) Some texture images are included twice in the database: the image itself and its negative (images above D100). The method considers them as identical textures, because of the correlation-based distance. This one is right since texture is more related to

spatial structures than their luminance variations. On the other hand, the system assessment considers them as different textures.

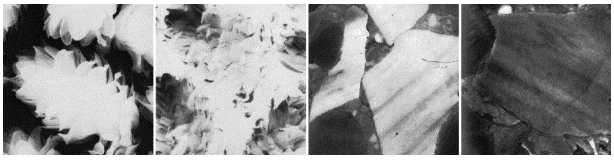

Figure 4. Some examples of retrieved textures. From left to right: D88-6, D89-9, D59-3, and D59-7.

Therefore, one should take as much the visual similarity as the retrieval efficiency into consideration, since both of them are involved in which one considers as a relevant retrieval.

Fig. 8 shows some example retrievals where the first top image is considered as the query for the system assessment and the following images in the column are in the top most similar ones.

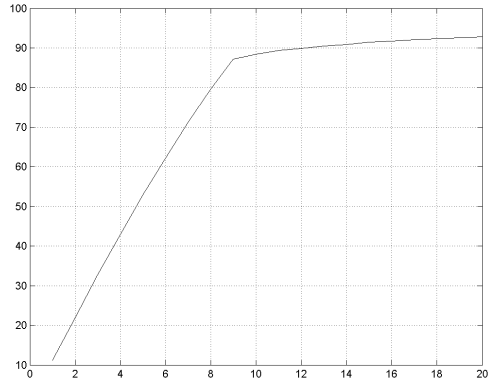

Figure 5. Percentage retrieval performance (average recall).

7. LOCAL TEXTURE FEATURE EXTRACTION AND INDEXING

In this section, we address the case of indexing and retrieving an image composed of several texture regions. Since texture information varies at distinct spatial locations, we apply the texture feature extraction method of the previous section to spatial neighborhoods localized at positions of a sample grid of the image. The sample grid is computed adaptively so that dimensionality reduction is achieved. As we have already mentioned, this is an important constraint in image indexing techniques.

Another important issue in texture image browsing and retrieval is not only retrieving images that match similar texture regions with respect to a texture query but locating in such a way these regions that help a user visualize their respective positions within the retrieved images. In image searching applications one is interested in having a general insight of such texture regions matching a query pattern. This characteristic simplifies the task of texture localization since only neighbored points belonging to the matched texture are useful. This way of texture localization differs from texture segmentation and texture edge detection. The former obtain a segmented image as a set of regions which are homogeneous in texture whereas the latter locate contours delimiting each texture region in the image. However, in both cases, the boundary between texture regions is finely detected.

7.1 Implementation Details

7.1.1 Sample grid

Local texture feature extraction consists of applying the parametric texture model of section 5 to spatial neighborhoods localized at positions of a sample grid of the image. The maximum size of the sample grid is the image size, where each position correspond to each pixel in the image. In this case, there is a vector of 600 parameters associated to each pixel in the image. Hence, the set of texture features increases dramatically and thereby the constraint of dimensionality reduction is not achieved for indexing purposes. Indeed, one should store 600 times the image size, which is not suitable as an index at all. On the other hand, the minimum size of the sample grid is one, the central pixel in the image, which correspond to the global feature extraction scheme presented in section 6. Nevertheless, this one supposes a unique uniform texture in the image, which is not the case. Hence, it is not suitable either. Therefore, there is a tradeoff between accuracy localization and index dimension, which is related to the size of the sample grid. Thus, a representative index should be of a dimensionality lower than or equal to the image size. In order to achieve accurate localization while avoiding dimensionality increasing with respect to the image size, the sample grid is computed adaptively in such a way the resulting index has the same size as the image size. Two subsampling steps, T_x and T_y, in x- and y-directions, respectively, are thus computed for indicating the positions in the image of the sample grid. The square size of the neighborhoods is equal to the maximum of subsampling steps, so that there is no overlapping between them at two adjacent positions, at least in one direction, whereas their tile cover all the image.

Let us define the following parameters: $N_c = S{\cdot}R{\cdot}(M^2+1)/2$, the total number of parameters of local autocorrelation at each position of the sample grid;. $X_s = X_r{\cdot}X_c$, the image size of X_r rows and X_c columns; K, the estimated number of samples in the grid; N, the neighborhood size; and N_p, the resulting number of samples in the grid when subsampling steps, T_x and T_y, are used.

$$K = \lceil X_s / N_c \rceil \ , \ T_x = \left\lceil X_c / \sqrt{K} \right\rceil \ , \ T_y = \left\lceil X_r / \sqrt{K} \right\rceil$$

$$N = \max(T_x, T_y) \ , \ N_p = \lceil X_c / T_x \rceil \cdot \lceil X_r / T_y \rceil \qquad (16)$$

The total number of values, i.e. the index size, which is close to the image size, is obtained as $N_c{\cdot}N_p$. Fig. 6 shows an input image composed of four different square-shaped texture regions and its sample grid.

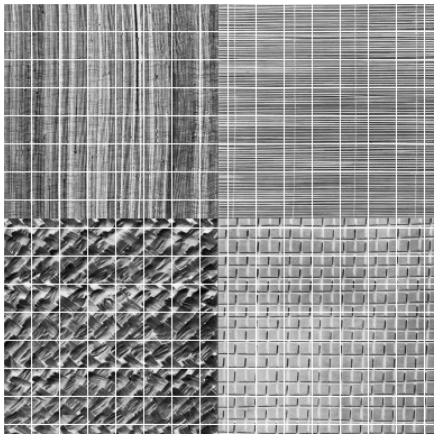

Figure 6. Sample grid of an image with four texture regions. Top row: D106 and D49. Bottom row: D18 and D01.

7.1.2 Local Autocorrelation

The local autocorrelation is computed for each neighborhood at positions of the sample grid. This process requires to compute 600 parameters at each position, which involves the 24 filters of the bank and the 25 lags for the autocorrelation values. However, in order to optimize computations, we have performed a filtering-based local autocorrelation. For each of the bands, 25 matrices with size of the image have been obtained. Their values are the resulting products between the band and its shifted versions. Then, for each of the matrices, a convolution with a 2D-rectangular window (size of NxN) is computed. This is equivalent to take summations at each pixel position. Finally, with the subsampling steps, T_x and T_y, in x- and y-directions, respectively, the local autocorrelation values are taken at the sample grid positions. At the end, one has 600 subsampled matrices, whose elements correspond to a vector of 600 parameters at the located position in the sample grid.

7.2 Experimental Results and Discussion

Retrieval is based on a similarity measure. We apply the same distance criterion presented in the previous section between each of the texture feature vectors at each position of the sample grid and the texture feature vector of a given query pattern. Notice that the query pattern is indexed by the methodology presented in section 6, since it is supposed to be a homogeneous texture in the whole image.

The neighborhoods with a small distance value indicate zones that match the searching query. Thus, we can use a *distance map*, i.e. a grayscale image associated to distance values at each position of the sample grid, where each tile represents an indexed neighborhood, to visualize the location of the retrieved texture. This way of localizing texture regions that match the query pattern can be seen as a texture region location without correspondence, since texture region identification is based on a matching criterion and there is not an accurate boundary detection. Besides, an analyzed texture region by a sample grid implies that the local texture can match a global one since there might be local similarities from the structure viewpoint.

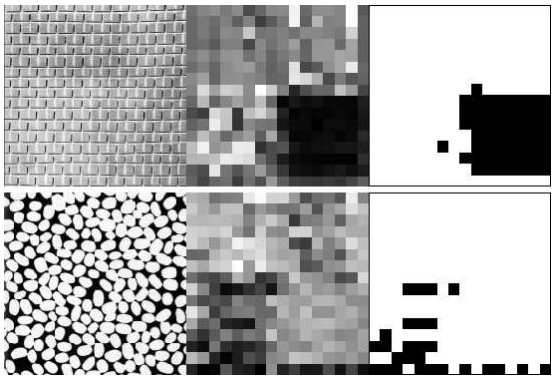

Figure 7. Distance map of two query texture responses. Left column: texture query. Middle column: gray distance map. Right column: binary map. Top row: D01. Bottom row: D75.

Fig. 7 shows the distance map of two query patterns. One corresponds to a region of the analyzed image presented in fig. 6 whereas the other is not present at all. The distance values are rescaled into 256 gray levels (from 0 to 255) for display purposes. Each tile of the distance map corresponds to a tile at a specified position of the sample grid of the original image. From fig. 7, it can be shown that a present texture in the analyzed image is detected (the image is thus retrieved) and the associated region is localized visually by the values of lower distances. A binarization of distance map enhances localization (where black zones are the detected texture region with respect to the query). In our experiments, we have chosen for binarization a percentage threshold of 20% (p = 0.2). The threshold distance, d_{thr}, is easily computed as $d_{thr} = p \cdot d_{max} + (1-p) \cdot d_{min}$, where d_{max} and d_{min} are the maximum and minimum distance values, respectively. However, dark information representing the smaller distances are not the same from one map to another. Further post-processing might improve region localization. For instance, one could validate a subregion texture if it has a minimum number of adjacent tiles. Of course, this does not allow to detect regions smaller than a minimal size. From fig. 7, one can see that a query texture which is not present in the analyzed image leads in general to a (binary) distance map that does not localize any texture. However, retrieval could occur since some tiles are anyway matched. It could be interpreted as an error, but it depends on the observer since perhaps there is locally a similarity, although image bounds could cause inexact detection due to discontinuity effects.

8. CONCLUSION

We have implemented an efficient texture feature extractor for texture discrimination based on a Gabor-like Hermite filter bank, which is a multi-scale decomposition of steered discrete Hermite filters. The Fisher criterion results show that better cluster separation, i.e. better texture discrimination, is obtained from Gabor (G1) and Gabor-like Hermite (G2) filter banks. Indeed, from fig. 2, one can see that their discrimination is similar. On the other hand, the cartesian Hermite (H1) filter bank perform less texture discrimination. Moreover, texture segmentation results confirm that G1 and G2 are equivalent models from a visual viewpoint and they perform similar classifications. Thus, texture discrimination is improved by the Gabor-like Hermite filter bank over the original cartesian Hermite filter bank.

Efficient global indexing and retrieval has been achieved by a powerful dimensionality reduction, which is based on spatial auto-correlations of all multi-channel outputs.

Spatial local autocorrelations at positions of a sample grid allow to index and localize without correspondence texture regions within an image composed of several textures. Indexing uses the same texture feature extraction scheme and texture region localization is based on both gray-scale and binary distance maps.

The system assessment for global indexing is not suitable enough to define a real retrieval efficiency as a function of the true similarity since it is only based on considering similar subimages when they result from the same large image. Despite this drawback, we have about 82% of good matches which increases up to 88% (see fig. 5) if some images with large structures and those with negatives are removed from the database. This proves that our method is suitable for texture indexing purposes, since the only available information to perform texture discrimination and indexing is that resulting from the steered filters of the Gabor-like Hermite filter bank. Thus, these filters characterize well enough texture.

9. ACKNOWLEDGMENTS

This work was supported by the National Council of Science and Technology (CONACyT) of Mexico, grant 111539, and by the SEP of Mexico.

10. REFERENCES

[1] Zhang, J. and Tan, T., Brief review of invariant texture analysis methods. *Pattern Recognition*, vol. 35, pp. 735-747, 2002.

[2] Manjunath, B.S. and Ma, W.Y., Texture features for browsing and retrieval of image data, *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 18, pp. 837-842, 1996.

[3] Wu, P., Manjunath, B.S., Newsam, S., and Shin, H.D., A texture descriptor for browsing and similarity retrieval. *Signal Processing*: *Image Communication*, vol. 16, no. 1,2, pp. 33-43, 2000.

[4] Huang, P.W. and Dai, S.K., Image retrieval by texture similarity, *Pattern Recognition*, vol. 36, pp. 665-679, 2003.

[5] Puzicha, J., Hofmann, T., and Buhmann, J.M., Non-parametric similarity measures for unsupervised texture segmentation and image retrieval, In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 267-272, 1997.

[6] Randen, T. and Husøy, J.H., Filtering for texture classification: A comparative study, *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 21, pp. 291-310, 1999.

[7] Bovik, A.C., Clark, M., and Geisler, W.S., Multichannel texture analysis using localized spatial filters, *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 12, pp. 55-73, 1990.

[8] Chen, C.C. and Chen, C.C., Filtering methods for texture discrimination, *Pattern Recognition Letters*, vol. 20, pp. 783-790, 1999.

[9] Randen, T. and Husøy, J.H., Multichannel filtering for image texture segmentation, *Optical Engineering*, vol. 33, pp. 2617-2625, 1994.

[10] Porat, M. and Zeevi, Y.Y., The generalized Gabor scheme of image representation in biological and machine vision, *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 10, pp. 452-468, 1988.

[11] Grigorescu, S.E., Petkov, N., and Kruizinga, P., Comparison of texture features based on Gabor filters, *IEEE Trans. Image Processing*, vol. 11, pp. 1160-1167, 2002.

[12] Martens, J.-B., The Hermite transform – Theory, *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 9, pp. 1595-1606, 1990.

[13] Young, R.A., Lesperance, R.M., and Meyer, W.W., The Gaussian derivative model for spatial-temporal vision: I. Cortical model. *Spatial Vision*, vol. 14, no. 3,4, pp. 261-319, 2001.

[14] Rivero-Moreno, C.J. and Bres, S., Les filtres de Hermite et de Gabor donnent-ils des modèles équivalents du système visuel humain?, In *Proc. ORASIS'03*, pp. 423-432, 2003.

[15] Rivero-Moreno C.J. and Bres , S., Conditions of similarity between Hermite and Gabor filters as models of the human visual system, In Petkov, N. and Westenberg, M.A. (eds.): Computer Analysis of Images and Patterns, *Lectures Notes in Computer Science*, vol. 2756, Springer-Verlag, Berlin Heidelberg, pp. 762-769, 2003.

[16] van Dijk, A.M. and Martens, J.-B., Image representation and compression with steered Hermite transforms, *Signal Processing*, vol. 56, pp. 1-16, 1997.

[17] Fukunaga K., *Introduction to Statistical Pattern Recognition, 2nd ed*. New York: Academic Press, 1991.

[18] Koekoek, R. and Swarttouw, R.F., *The Askey-Scheme of Hypergeometric Orthogonal Polynomials and its q-Analogue*. Technical Report 98-17, Delft University of Technology, Faculty of Information Technology and Systems, Department of Technical Mathematics and Informatics, The Netherlands, 1998.

[19] Freeman, W.T. and Adelson, E.H., The design and use of steerable filters, *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 13, pp. 891-906, 1991.

[20] Brodatz, P., *Textures: A Photographic Album for Artists and Designers*. New York: Dover, 1966.

[21] Portilla, J. and Simoncelli, E.P., A parametric texture model based on joint statistics of complex wavelet coefficients, *Int. Journal Computer Vision*, vol. 40, pp. 49-71, 2000.

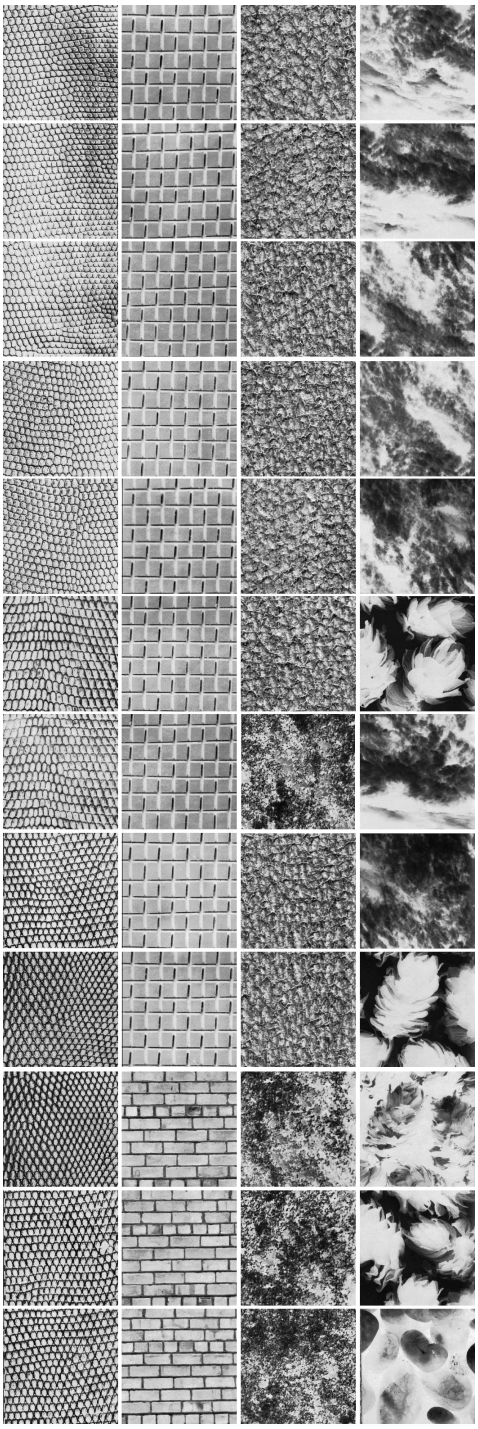

Figure 8. Example of retrieval images where the first top image is considered as the query.