A New Method for Texture-Based Image Analysis

Andrey V. Kutovoi, Andrey S. Krylov
Moscow State University, Moscow, Russia
kutovoi@gmail.com
kryl@cs.msu.ru

Abstract

This paper presents a new method in processing textured images. It is based on the 2-D Hermite transform for generic lattice position. It is equivalent to Hermite projection method. The optimization is based on applying the Fast 2-D Hermite projection method combining with Polar Hermite transform. Fast Hermite projection method transform allows to efficiently calculate the coefficients of the 2-D Cartesian Hermite transform for generic lattice using the Gaussian quadrature. Polar Hermite coefficients have simple properties under rotations and thus allow to avoid similar calculations while obtaining Hermite transform coefficients for the source image and it's rotations that is a common task in texture processing. We discuss application of the method to the tasks of obtaining the texture feature vectors (texture parameterization) and texture-base image segmentation.

Keywords: *Texture, Hermite transform for generic lattice position, Fast Hermite projection method, Polar Hermite transform, Image segmentation.*

1. INTRODUCTION

Texture has always been an important part of the visual world. A considerable part of human eyesight perception is based not only on the recognition of brightness, contrast, color of the image but also on detecting texture patterns. In [1] the human and monkey texture perception is described and the assumption is made that it is possible to create the image processing method that will model the human texture perception process. In the article [2] the comparison between modern texture processing methods is performed.

In most of the studies the relation of the texture to the local in image spectrum is established through features which are obtained by filtering with a set of two-dimensional filters. Such filters are local and are characterized by a preferred orientation and a preferred spatial frequency. Roughly speaking, it acts as a local band-pass filter with certain optimal localization properties in both the spatial domain and the spatial frequency domain. Typically, a multi-channel filtering scheme is used: an image is filtered with a set of filters with different preferred orientations and spatial frequencies, which cover appropriately the spatial frequency domain, and the features which are obtained form a feature vector field which is used further.

In Section 2 we will describe the 2-D Cartesian Hermite transform [3] for generic lattice position as the basic method of obtaining the feature vectors for textures. In Section 3 we will present the Fast Hermite projection method that allows to appreciably increase the speed of image processing.

Polar Hermite Transform was developed by Martens in [4] and it is useful for texture analysis. Current multiscale Hermite transform investigations [5] also look promising to be used in texture discrimination. We will describe Polar Hermite Transform in Section 4. We will also show in Section 5, how the Fast Hermite projection method and the Polar Hermite Transform can be combined to perform a new optimization of the Hermite transform for generic lattice position.

In Section 6 we will perform a comparison between different methods and show how the optimization allows to solve such tasks as texture parameterization and texture-based image segmentation in acceptable time.

2. 2-D CARTESIAN HERMITE TRANSFORM

The 2-D Hermite transform with detailed description was originally introduced in [3] and [6]. Let us consider an image *l(x,y)*. For the Cartesian Hermite transform for generic lattice position [4] we obtain the coefficients using the formula

$$l^c_{n-m,m} = \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} l(x,y)\cdot \psi^c_{n-m,m}(-x,-y)dy \tag{1}$$

where $\psi^c_{n,m}(x,y)$ are the 2-D Hermite functions in Cartesian coordinate system. The 2-D Hermite functions are defined as:

$$\psi_{nm}(x,y) = \frac{(-1)^{n+m} e^{(x^2+y^2)/2}}{\sqrt{2^{n+m} n! m! \pi}} \cdot \frac{d^n(e^{-x^2})}{dx^n} \cdot \frac{d^m(e^{-y^2})}{dy^m} \tag{2}$$

The functions with the lower indexes have traditionally been used as preprocessing filters in computer vision, for they are the derivatives of Gaussian. One important advantage of the 2-D Hermite functions is that they are separable. This means that the calculation of the coefficients can be implemented very efficiently. The functions also can be determined by superposition of the 1-D recurrent formulae:

$$\begin{gathered}
\psi'_0(z) = \frac{1}{\sqrt[4]{\pi}} \cdot e^{-z^2/2} \\
\psi'_1(z) = \frac{\sqrt{2}z}{\sqrt[4]{\pi}} \cdot e^{-z^2/2} \\
\begin{cases} \psi'_n(z) = z\sqrt{\dfrac{2}{n}} \cdot \psi'_{n-1}(z) - \sqrt{\dfrac{n-1}{n}} \cdot \psi'_{n-2}(z) \\ \forall n \geq 2 \end{cases} \\
\psi_{nm}(x,y) = \psi_n(x) \cdot \psi_m(y) \\
\psi_n(z) = \psi'_n(z),\ z = x \vee y\,.
\end{gathered} \tag{3}$$

The 2-D Hermite functions satisfy an important feature for image processing, and can be also used as a "substitution" of trigonometric functions in image spectrum analysis. This approach is used in Hermite projection method [8], [9]. The Hermite projection method is equivalent to Cartesian Hermite transform for generic lattice position.

The graphs of the 2-D Hermite functions are shown in Fig.1:

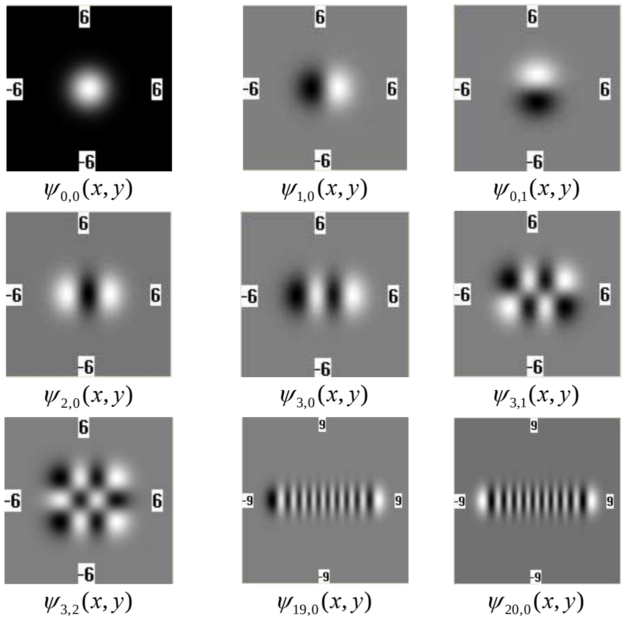

Figure 1: The 2-D Hermite functions.

3. POLAR HERMITE TRANSFORM

Another important approach to Hermite transform is to perform it in the polar coordinates. Although the Cartesian Hermite transform for generic lattice position is easy to calculate, owing to the fact that the filters are separable, proceeding with filtering in polar domain has an important advantage. The advantage is that the coefficients of the transform change in a simple way upon a rotation of the coordinate axes.

Adopting local coordinate axes (for instance, with the *x*-axis oriented along the gradient) is one of the basic operations in differential geometry because many expressions for important surface properties can be simplified greatly in an adaptive coordinate system [4]. Rotating the coordinate axes has a complicated effect on the Cartesian Hermite coordinates. Therefore, it is not without interest to try to apply the theory of local signal decompositions in polar coordinates in order to obtain a representation which behaves much better upon rotation of the coordinate axes. The detailed description of the polar transform is described in [4].

Meanwhile having obtained the set of Cartesian Hermite coefficients for generic lattice position we can transform them to Polar Hermite Coefficients using the following formulae: [2], [4]

$$l^{p}_{n-k,k} = \sum_{m=0}^{n} G_n^{c\tilde{p}}(m,k)\cdot l^{c}_{n-m,m} \tag{12}$$

$$G_n^{c\tilde{p}}(m,k) = <\alpha^{c}_{n-m,m}(\omega), \tilde{\alpha}^{p}_{n-k,k}(\omega)>,$$

$$<f(\omega), g(\omega)> = \frac{1}{2\pi}\int_0^{2\pi} f^{*}(\omega)g(\omega)d\omega;$$

$$\alpha^{c}_{n-m,m}(\omega) = \cos^{n-m}\omega\cdot\sin^{m}\omega\sqrt{\frac{n!}{(n-m)!m!}},\ m=0..n$$

$$\tilde{\alpha}^{p}_{n-m,m}(\omega) = \sqrt{\frac{2^n(n-m)!m!}{n!}}\cdot\begin{cases}\sqrt{2}\cos(n-2m)\omega, & 0\le m<\frac{n}{2}\\ 1, & m=\frac{n}{2}\\ \sqrt{2}\sin(n-2m)\omega, & \frac{n}{2}\le m<n\end{cases}.$$

The polar Hermite Coefficients have a remarkable feature. The coefficients for the source image and for it's rotated at an angle ϕ_0 version are related as follows:

$$\begin{bmatrix} l^{p}_{n-m,m}(\varphi_0)\\ l^{p}_{m,n-m}(\varphi_0)\end{bmatrix} = R_{n-m,m}(\varphi_0)\cdot\begin{bmatrix} l^{p}_{n-m,m}\\ l^{p}_{m,n-m}\end{bmatrix} \tag{13}$$

with

$$R_{n-m,m}(\varphi_0) = \begin{bmatrix}\cos(n-2m)\varphi_0 & \sin(n-2m)\varphi_0\\ -\sin(n-2m)\varphi_0 & \cos(n-2m)\varphi_0\end{bmatrix}$$

For *m<n-m,* and $l^{p}_{n-m,m}(\varphi_0) = l^{p}_{n-m,m}$ for *m=n-m.*

Hence, an important advantage of the polar Hermite transform is that the coefficients change in a simple way upon a rotation of the coordinate axes.

4. DESCRIPTION OF THE METHOD

In texture processing there are two commonly used important texture characteristics: frequency and orientation.

Below is described the algorithm of obtaining the feature vector for a texture sample. Texture sample is an image for which the following is true: 1) the sample image contains the characteristic pixels set for the current texture and 2) if we crop the sample image then some part of important texture pixels distribution will be lost and it will be impossible for a human to recognize the texture.

At first using the 2-D Fast Hermite projection method we calculate the $l^{c}_{n,m}$, $n=0..31$, $m=0..31$ coefficients of Hermite decomposition. In Fig.2 there are plotted source texture samples and the sets of coefficients.

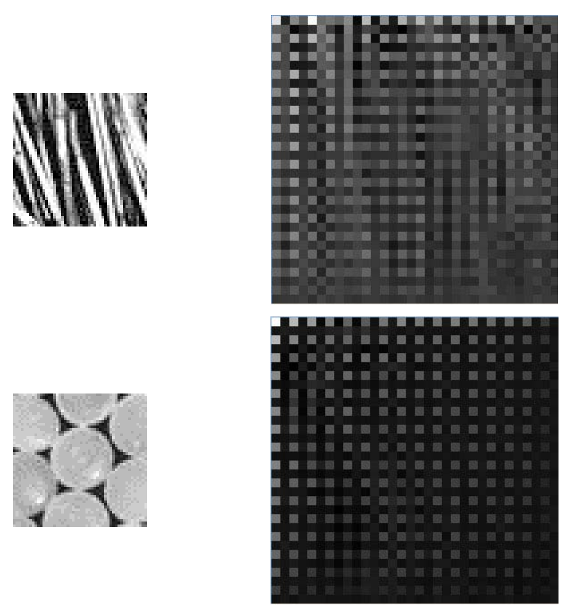

Figure 2:Texture samples and the graphs of coefficients.

Then using the Cartesian to Polar coefficients transform formula (12) we obtain the set of polar Hermite coefficients $l^p_{n-m,m},\ n = 0..31,\ m = 0..n$. As described above the Polar Hermite transform and the coefficients have a remarkable characteristic property. They change in a simple way if we consider a rotated source image together with the source image. The coefficients of the Polar Transform for the source and the rotated images are related by a simple multiplication by a rotation matrix (13).
We use 8 orientations for the texture samples to obtain characteristic texture features. The orientations are 0˚, 22.5˚, 45˚, 67.5˚, 90˚, 112.5˚, 135˚, 157.5˚. This is illustrated in Figure 3 where a texture sample and it's rotated versions are displayed.

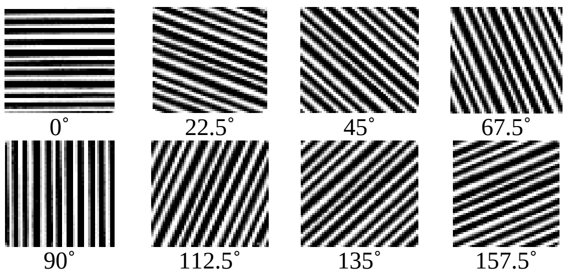

Figure 3:The source texture sample and it's rotations at concerned angles.

Then we transform the rotated polar coefficients back to Cartesian. The Polar to Cartesian transform of Hermite coefficients is actually the reverse transform to the Cartesian to Polar transform due to the correlation[4]:

$$l^c_{n-m,m} = \sum_{k=0}^{n} G_n^{c\tilde{p}}(m,k)\cdot l^p_{n-k,k},\ m = 0..n \qquad (14)$$

On this stage we only obtain the coefficients $l^{c,\varphi}_{n,0}$, n=0..31, φ=0˚, 22.5˚, 45˚, 67.5˚, 90˚, 112.5˚, 135˚, 157.5˚.
So at this moment we have 32 * 8 = 8192 coefficients.
The last step is to combine the coefficients into the feature vectors. For the textures (because even different samples of the same texture very rarely have similar coefficient values in account of the fortuitous shifts and rotations and noise in texture samples) it is a commonly used approach to obtain and later on compare the energy values. For the set of coefficients we obtain

$$e_{1,\varphi} = (l^{c,\varphi}_{1,0})^2 + (l^{c,\varphi}_{2,0})^2$$
$$e_{2,\varphi} = (l^{c,\varphi}_{3,0})^2 + (l^{c,\varphi}_{4,0})^2$$
$$e_{3,\varphi} = (l^{c,\varphi}_{5,0})^2 + (l^{c,\varphi}_{6,0})^2 + (l^{c,\varphi}_{7,0})^2 + (l^{c,\varphi}_{8,0})^2$$
$$e_{4,\varphi} = \begin{array}{l}(l^{c,\varphi}_{9,0})^2 + (l^{c,\varphi}_{10,0})^2 + (l^{c,\varphi}_{11,0})^2 + (l^{c,\varphi}_{12,0})^2 + (l^{c,\varphi}_{13,0})^2 + (l^{c,\varphi}_{14,0})^2 + \\ (l^{c,\varphi}_{15,0})^2 + (l^{c,\varphi}_{16,0})^2\end{array}$$
$$e_{5,\varphi} = \begin{array}{l}(l^{c,\varphi}_{17,0})^2 + (l^{c,\varphi}_{18,0})^2 + (l^{c,\varphi}_{19,0})^2 + (l^{c,\varphi}_{20,0})^2 + (l^{c,\varphi}_{21,0})^2 + (l^{c,\varphi}_{22,0})^2 + \\ (l^{c,\varphi}_{23,0})^2 + (l^{c,\varphi}_{24,0})^2 + (l^{c,\varphi}_{25,0})^2 + (l^{c,\varphi}_{26,0})^2 + (l^{c,\varphi}_{27,0})^2 + (l^{c,\varphi}_{28,0})^2 + \\ (l^{c,\varphi}_{29,0})^2 + (l^{c,\varphi}_{30,0})^2 + (l^{c,\varphi}_{31,0})^2\end{array}$$

for each φ =0˚, 22.5˚, 45˚, 67.5˚, 90˚, 112.5˚, 135˚, 157.5˚. So we have the 8 * 5 coefficients feature-vector.

5. RESULTS

Using the described algorithm we obtained the following feature vectors for several texture examples:

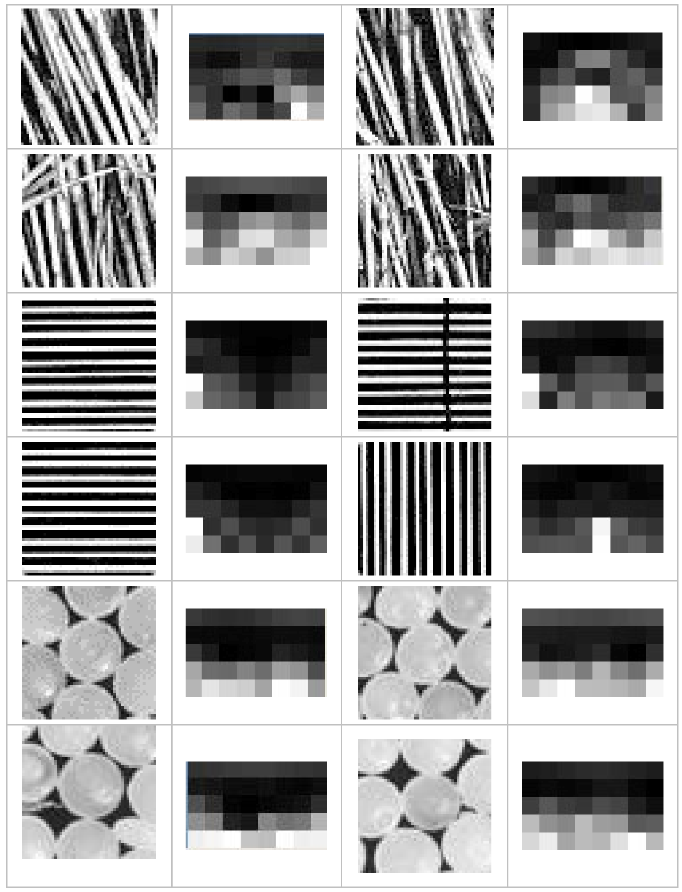

Figure 4: The diagrams of the feature vectors for some texture samples.

We also applied the method to the task of texture-based image segmentation. The result is presented n Figure 5.

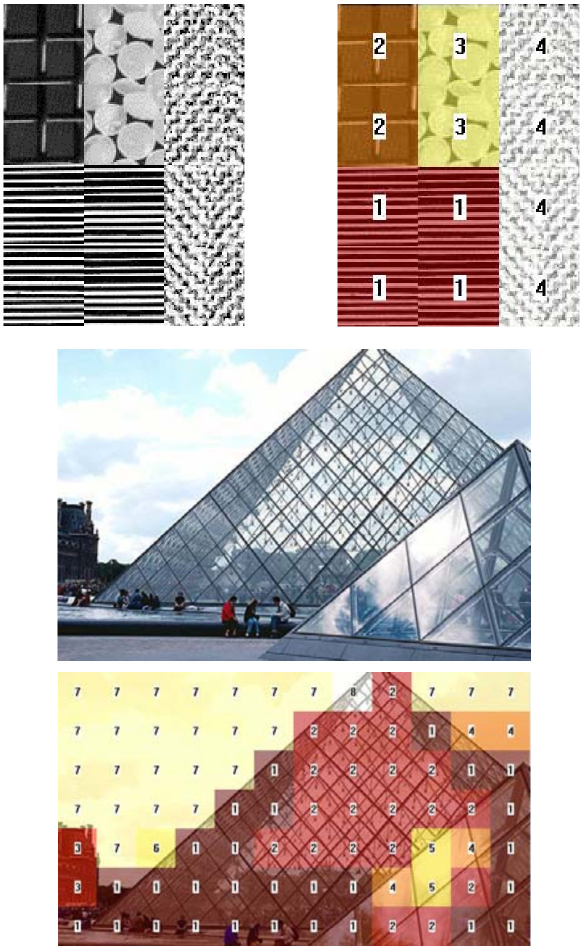

Figure 5: Texture-based segmentation of an artificial textured image and an ordinary image with textures.

The basic advantage of the method over the method presented in [9] is that the processing can be performed much faster because of the ease of obtaining the feature vectors for the rotated samples of textures. In Table 1 is presented the speed of segmentation of two images. The processing was performed on an Intel Pentium M 1.86 GHz processor with 512 Mbytes of RAM. As can be seen from the following table the method described in this article allows increase the speed of texture-based image segmentation more than ten times.

Method \ Textured image		
Segmentation time: Hermite transform not optimized, sec	62	487
Segmentation time: Optimized Hermite transform, sec	5	34

Table 1: Comparison of times of processing the images with texture-based segmentation.

6. CONCLUSION

Combining Fast Hermite projection method and Polar Hermite transform results in a new method for texture parameterization. The new method shows remarkable results in the processing rate concerning the texture parameterization and image segmentation tasks. The method can be applied in different tasks of texture-based image processing to increase their productivity.

7. REFERENCES

[1] N. Petkov, P. Kruizinga, *"Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells" Biol. Cybern. 76, 83–96 (1997).*

[2]Trygve Randen, John Hakon Husoy *"Filtering For Texture Classification: A Comparative Study" IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 4, april 1999.*

[3] Jean-Bernard Martens, *"The Hermite Transform - Theory" IEEE Transactions On Acoustics, Speech and Signal Processing, vol. 38, no. 9, september 1990*

[4] Jean-Bernard Martens, *"Local Orientation Analysis in Images by Means of the Hermite Transform" IEEE Transactions On Iimage Processing, vol. 6, no. 8, pp. 1103-116, 1997.*

[5] J. L. Silván-Cárdenas and B. Escalante-Ramírez, *"The Multiscale Hermite Transform for Local Orientation Analysis", IEEE Transactions On Image Processing, vol. 15, no. 5, MAY 2006*

[6] Jean-Bernard Martens, *"The Hermite Transform - Applications" IEEE Transactions On Acoustics, Speech and Signal Processing, vol. 38, no. 9, september 1990.*

[7] V.I. Krylov, *"Numerical Integral Evaluation", Moscow: Science, pp. 116-147, 1967 (in Russian).*

[8] A. Krylov, D. Kortchagine, "*Image Database Retrieval by Fast Hermite Projection Method", Proceedings Int. Conference Graphicon 2005, Moscow.*

[9] A. Krylov, A. Kutovoy, "*Texture Parameterization With Hermite Functions", Proceedings Int. Conference Graphicon 2002, Nizhniy Novgorod.*

About the author

Andrey V. Kutovoi is a Ph.D. student at Moscow State University, Department of Computational Mathematics and Cybernetics. His contact email is kutovoi@mail.ru.

Andrey S. Krylov is associated professor at Moscow State University, Department of Computational Mathematics and Cybernetics. His contact email is kryl@cs.msu.su.