International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 27, No. 4 (2013) 1354006 (25 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218001413540062

World Scientific
www.worldscientific.com

STEERABILITY OF HERMITE KERNEL

BO YANG*, JAN FLUSSER† and TOMÁŠ SUK‡

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
18208, Praha 8, Czech Republic
**bo.yang@hotmail.fr*
†*flusser@utia.cas.cz*
‡*suk@utia.cas.cz*

Received 5 November 2012
Accepted 7 May 2013
Published 12 July 2013

Steerability is a useful and important property of "kernel" functions. It enables certain complicated operations involving orientation manipulation on images to be executed with high efficiency. Thus, we focus our attention on the steerability of Hermite polynomials and their versions modulated by the Gaussian function with different powers, defined as the Hermite kernel. Certain special cases of such kernel, Hermite polynomials, Hermite functions and Gaussian derivatives are discussed in detail. Correspondingly, these cases demonstrate that the Hermite kernel is a powerful and effective tool for image processing. Furthermore, the steerability of the Hermite kernel is proved with the help of a property of Hermite polynomials revealing the rule concerning the product of two Hermite polynomials after coordination rotation. Consequently, any order of the Hermite kernel inherits steerability. Moreover, a couple sets of an explicit interpolation function and basis function can be directly obtained. We provide some examples to verify steerability of the Hermite kernel. Experimental results show the effectiveness of steerability and its potential applications in the fields of image processing and computer vision.

Keywords: Hermite polynomials; Hermite kernel; steerability; adaptive filtering.

1. Introduction

Since the steerable filters were first proposed by Freeman and Adelson in 1991, they became widely utilized for image processing.[9,14,22,23,28,30] Steerable filters are characterized as a class of filters in which a filter of arbitrary orientation is synthesized as a linear combination of a set of "basis filters". A mathematical explanation of steerable filter is presented in Eq. (1)

$$f^{\theta}(r,\phi) = \sum_{j=1}^{M} k_j(\theta) g_j(r,\phi), \tag{1}$$

*Corresponding author.

where $k_j(\theta)$ and $g_j(r, \phi)$ serve as interpolation functions and basis functions, respectively. Steerable filters allow effective and simple practical implementations of deformable filters.

Historically, the idea of steerable filters can be found in the work of Danielsson and Knutsson without explicitly mentioning its conception.[4,10] It was Freeman and Adelson who coined the concept of "steerability". They introduced steerable filters based on Gaussian derivatives and concluded that a product of a polynomial and a rotationally symmetric window function results in a steerable function.[9] Together with steerability in orientation, Simoncelli *et al.* also investigated steerability in translation and scale. The terminology "shiftability" is proposed for steerability in different parameters resulting in various deformations. Simoncelli and his colleagues also addressed joint shiftability, covering steerability in both orientation and position, as well as in both scale and position simultaneously.[22] A class of steerable wedge filters were designed by Simoncelli and Farid. Ambiguity on the orientation map was efficiently eliminated by using steerable wedge filters.[24] The Lie-group theory is used to develop steerable filters as well. Typically, steerable filters were designed by Fourier decompositions, effectively generalized by Michaelis and Sommer with the Lie-group theory.[13] Teo and Hel-Or based handling of problems related to computational aspects of steerable functions on the Lie-group theory.[29] Comparatively, Beil proposed a method to derive a minimum number of basis functions based on Cartesian tensor calculus. Rotation invariants were expressed by tensor calculus. These invariant components thus enable construction of a steerable function.[3] More recently, the design of multi-steerable filters was formulated by Muhlich *et al.* They demonstrated that the multi-steerable filters can be constructed by a correct combination of single steerable filters. The polynomial techniques for generating multi-steerable filters and some applications such as junction analysis were also given.[14] Approximation of a given deformed function by basis functions is investigated in relationship with steerable filters. Perona defined the quality of approximation and proposed a technique useful for finding the best basis functions to approximate the given function within a predefined error or to exactly represent the function.[16,17] Yu *et al.* proposed approximate steerability which decomposes the signal locally in the spatial domain with Gaussian masks as basis filters. Their method yields both higher orientation resolution and lower complexity.[33]

On the other hand, Hermite polynomials and their weighted versions are powerful tools and thus are widely used in image processing and computer vision. Here, we call Hermite polynomials and their weighted versions the "Hermite kernel" for convenience. Theoretically, the kernel function is rigorously defined in relation to the inner product and Hilbert space. Some kernel functions such as the linear kernel, polynomial kernel and Gaussian kernel are frequently used in practice. For an instance, support vector machines are one of the most well-known algorithms based on the kernel functions.[27] The Hermite kernel is a kind of "pseudo" kernel function according to the definition. However, we can still find wide applications of the Hermite kernel.

Certain terminologies such as "Hermite transform",[12] "Shapelets transform",[20,11] "Hermite moments",[26] and "Gaussian−Hermite moments"[25,31,32] are all based on the Hermite kernel. In this paper we offer a solution of the steerability of the Hermite kernel. We are capable of decomposing a rotated kernel function of any order by the specific interpolation functions and basis functions, no matter what the kernel represents: a Hermite polynomial, a Hermite function, a Gaussian derivative or any other Hermite polynomial based function. Freeman and Adelson formulated a method to compute interpolation functions of the pth partial derivative of the Gaussian function $G_{p,0}$. Comparatively, our method offers computation of interpolation functions for any Hermite kernel. It is not necessary for one index (either p or q) to equal 0, nor does the kernel have to be confined to Gaussian derivatives. Actually, the kernel can be either Hermite polynomials, Hermite functions, or even the Hermite polynomial-related versions in the 2D space.

The paper is organized as follows: we present the construction of a steerable function and the definition of the Hermite kernel in Sec. 2. Further, we discuss special cases of the Hermite kernel, which correspond to Hermite polynomials, Hermite functions and Gaussian derivatives, respectively. The derivations of the interpolation and basis functions are formulated in Sec. 3. Subsequently in Sec. 4, we provide applications of such steerability in image processing. Section 5 concludes the paper.

2. Hermite Kernel and Its Steerability

Generally, the topic of steerability consists of three aspects. The first one is constructing the functions which are steerable. The second focuses on the required number of basis functions sufficient to synthesize the rotated function. The last and most important aspect focuses on the definition of interpolation function and basis function used to decompose the rotated function.

2.1. *Hermite kernel*

In this section we construct the so-called Hermite kernel and discuss its steerability. Firstly, a brief introduction of Hermite polynomials is necessary. The Hermite polynomial $H_p(x)$ is an orthogonal polynomials defined over the domain $(-\infty, \infty)$

$$H_p(x) = (-1)^p \exp(x^2) \frac{d^p}{dx^p} \exp(-x^2). \tag{2}$$

It can be written in the form of series often used in practice

$$H_p(x) = \sum_{k=0}^{[p/2]} \frac{(-1)^k p!}{k!(p-2k)!} (2x)^{p-2k}. \tag{3}$$

Figure 1 shows the Hermite polynomials up to the fifth-order. Since Hermite polynomials increase significantly with the increasing order, we normalized their values by the squares of corresponding orders for graphing purposes. Hermite polynomials

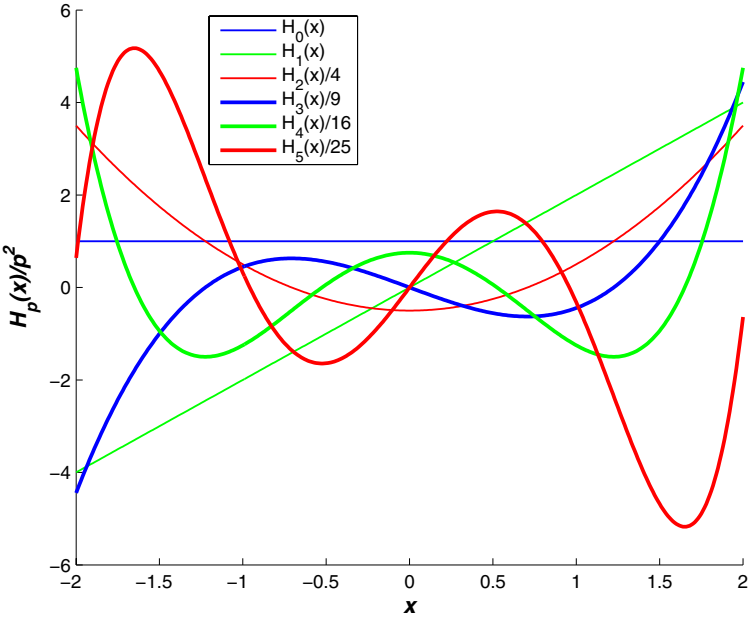


Fig. 1. Hermite polynomials of order 0 up to 5.

are orthogonal with the Gaussian function being their weight function. Orthogonality is formulated as follows:

$$\int_{-\infty}^{\infty} H_p(x)H_q(x)\exp(-x^2)dx = \begin{cases} 0 & p \neq q \\ 2^p p!\sqrt{\pi} & p = q. \end{cases} \tag{4}$$

Freeman and Adelson proposed a method to construct a steerable filter from any arbitrary polynomial.[9] They reduced the method to a theorem cited below:

Theorem 1. *Let $f(x,y) = W(r)P_N(x,y)$ where $W(r)$ is an arbitrary modulating function, and $P_N(x,y)$ is an Nth order polynomial in x and y, whose coefficients may depend on r. Linear combinations of $2N+1$ basis functions are sufficient to synthesize $f(x,y) = W(r)P_N(x,y)$ rotated to any angle. If $P_N(x,y)$ contains only even [odd] order terms (terms $x^p y^q$ for $p+q$ even [odd]), then $N+1$ basis functions are sufficient.*

In Theorem 1, $W(r)$ is a radial function with $r = \sqrt{x^2+y^2}$. Theorem 1 ensures the existence of steerability of the polynomial-based kernel. Accordingly, it remains quite trivial to construct steerable filters from different kinds of polynomials. The implementation is simple, only requiring an arbitrary symmetric modulating function $W(r)$. The selection of $W(r)$ is free and will not affect the steerability of the constructed kernel. We prefer to choose a Gaussian function as the window function, because it is smooth and separable thus enabling efficient implementation in practice. Moreover, we choose the Gaussian function since it is a weight function of Hermite polynomials. The steerable and orthogonal properties might be obtained from the

constructed kernel simultaneously with Gaussian modulation. Consequently, we define the 1D Hermite kernel as follows

$$\tilde{H}_p^n(x;\sigma) = H_p\left(\frac{x}{\sigma}\right)\exp\left(-\frac{nx^2}{2\sigma^2}\right) \quad n \in \mathbf{R}. \tag{5}$$

It is obvious that the 1D Hermite kernel is actually a Hermite polynomial modulated by a Gaussian envelope of different power. In Eq. (5) n is a real number. Steerability is meaningless for the 1D Hermite kernel. In order to discuss steerability, we should extend the kernel to 2D space. The Hermite kernel in 2D space is defined with the separability of both Hermite polynomials and Gaussian functions

$$\begin{aligned} B_{p,q}^n(x,y;\sigma) &= \tilde{H}_p^n(x;\sigma)\tilde{H}_q^n(y;\sigma) \\ &= H_p\left(\frac{x}{\sigma}\right)H_q\left(\frac{y}{\sigma}\right)\exp\left(-\frac{nx^2+ny^2}{2\sigma^2}\right). \end{aligned} \tag{6}$$

Equation (6) yields an explicit expression of the Hermite kernel which is steerable. For convenience, we use the term "Hermite kernel" when speaking about the "2D Hermite kernel" in the remainder of this paper. The term $H_p(\frac{x}{\sigma})H_q(\frac{y}{\sigma})$ in Eq. (6) corresponds to $P_N(x,y)$ and $\exp(-\frac{nx^2+ny^2}{2\sigma^2})$ is the counterpart of $W(r)$ as mentioned in Theorem 1. The number of basis functions to steer Eq. (6) is obvious according to Theorem 1. The number M should equal $p+q+1$, since the product of Hermite polynomials contains only either even or odd orders of the term $x^p y^q$.

2.2. *Special cases of the Hermite kernel*

There are four parameters p, q, σ and n which control the final form of the Hermite kernel. The effect of the first three parameters p, q and σ on the kernel can be easily understood. They determine the different order of the Hermite polynomial in the kernel as well as the different scale with which the kernel expands. However, the effect of the parameter n on the kernel cannot be easily understood directly. In this subsection, we show some special cases of the kernel whose n is corresponding to 0, 1 and 2, respectively.

2.2.1. *Scaled Hermite polynomials*

When $n = 0$, Eq. (6) defines a scaled Hermite polynomial

$$B_{p,q}^0(x,y;\sigma) = H_p\left(\frac{x}{\sigma}\right)H_q\left(\frac{y}{\sigma}\right). \tag{7}$$

So far, the Hermite kernel in this case has been used by Shen to define Hermite moments.[26] Because the amplitude of $H_p(x)$ explodes for larger x, especially when the order is higher, we show the Hermite kernel only for orders from $(0,0)$ up to $(5,5)$ in Fig. 2. It should be noted that we draw the kernels with $\frac{x}{\sigma}$ being limited to the

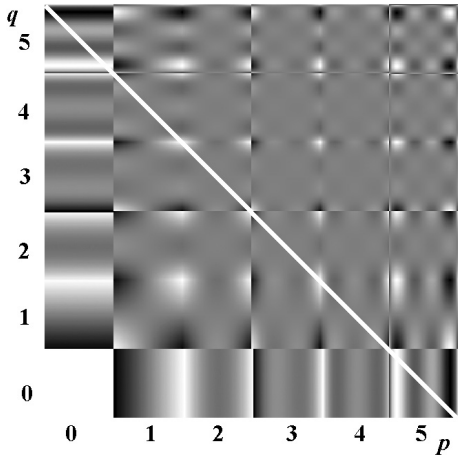


Fig. 2. The Hermite kernel ($n = 0, \sigma = 1.0$) is shown in square windows whose sizes are 91×91 pixels. These kernels are generated by 2D scaled Hermite polynomials. Each square is drawn by mapping the values of the corresponding kernel to the interval of gray level $[0, 255]$. From bottom to top and from left to right, it shows the Hermite kernel of order $(0, 0)$ to $(5, 5)$, respectively. The kernels located in different diagonal directions are basis functions for the Hermite kernel of different orders. For example, the kernels crossed by the white line serve as basis functions for the kernel of order 5.

interval $[-2.1, 2.1]$, since the Hermite polynomial $H_5(x)$ has its greatest root around 2.0202. Therefore, selecting this interval ensures that all roots of selected polynomials (here they are Hermite polynomials of orders from 0 to 5) will be included. At the same time it will produce minor distortion when drawing these kernels.

2.2.2. *Hermite functions*

The Hermite kernel corresponds to a non-coefficient 2D Hermite function with $n = 1$. Its expression is detailed by

$$B^1_{p,q}(x, y; \sigma) = H_p\left(\frac{x}{\sigma}\right) H_q\left(\frac{y}{\sigma}\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \tag{8}$$

One property of the Hermite functions is that they are eigenfunctions of the Fourier transform. This means that the Fourier transform of a Hermite function is still a Hermite function multiplied by a scaling factor. For example

$$\mathcal{F}\left[H_p(x) \exp\left(-\frac{x^2}{2}\right)\right] = (-i)^p H_p(\omega) \exp\left(-\frac{\omega^2}{2}\right), \tag{9}$$

where $\mathcal{F}$ represents the Fourier transform and $i = \sqrt{-1}$. The Hermite kernel in this case also inherits this property due to the separability of the Fourier transform in 2D space. Actually, some orthogonal moments and transforms are based on such kernel. The terminologies "Gaussian−Hermite moments"[25,31,32] and "Shapelets transform"[11,20] use bases similar to this kernel. The only difference is those bases

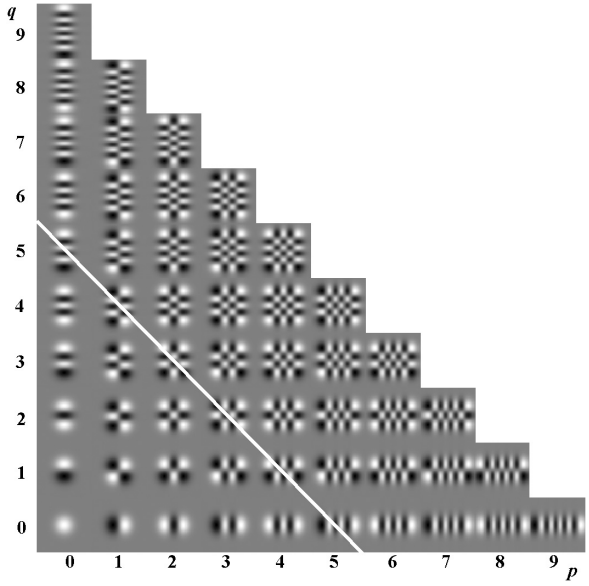


Fig. 3. The Hermite kernel ($n = 1, \sigma = 10.0$) is shown in square windows whose sizes are 91×91 pixels. From bottom to top and from left to right, each square shows the Hermite kernel of order 0 to 9. The image shown in each square is generated by mapping the values of the corresponding kernel to the interval of gray level $[0, 255]$. The kernels located in different diagonal directions are basis functions for the Hermite kernel of different orders. The white line marks the basis functions for the kernel of order 5.

have normalized coefficients, the Hermite kernel, however, has no coefficient in its definition. Figure 3 shows the Hermite kernel of orders from 0 to 9.

2.2.3. *Gaussian derivatives*

When we set $n = 2$, the Hermite kernel is defined as:

$$B^2_{p,q}(x, y; \sigma) = H_p\left(\frac{x}{\sigma}\right) H_q\left(\frac{y}{\sigma}\right) \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right). \tag{10}$$

In this case the Hermite kernel is substantially a non-coefficient filter function of the Hermite transform. Martens pointed out that the filter function of the Hermite transform is substantially identical to the corresponding Gaussian derivative.[12] The relation between the Gaussian derivative and the Hermite kernel is obvious from the following equation:

$$\frac{\partial^p G(x; \sigma)}{\partial x^p} = \left(-\frac{1}{\sqrt{2}\sigma}\right)^p \sqrt{2\pi}\sigma H_p\left(\frac{x}{\sqrt{2}\sigma}\right)\left(G\left(\frac{x}{\sqrt{2}}; \sigma\right)\right)^2, \tag{11}$$

where $G(x; \sigma)$ is a Gaussian function and σ is its standard deviation. The rightmost term of Eq. (11) is actually a 1D Hermite kernel if the coefficient is ignored. Hence,

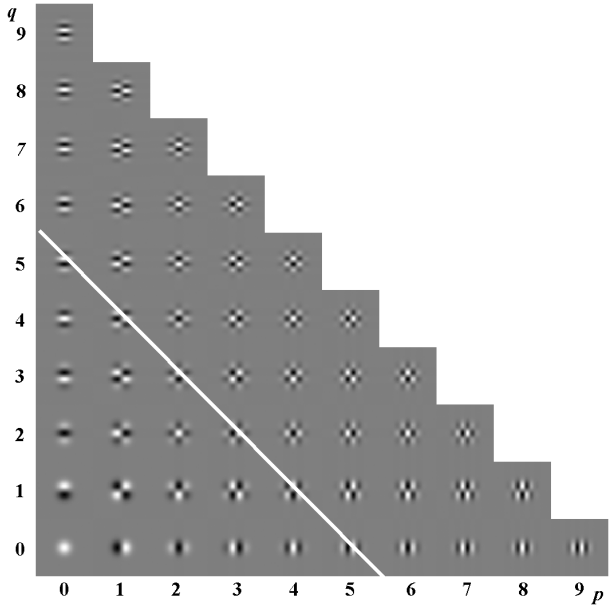


Fig. 4. Hermite kernel ($n = 2, \sigma = 10.0$) is shown in square windows whose sizes are 91×91 pixels. From bottom to top and from left to right, each square shows Hermite kernel of order 0 to 9. The image shown in each square is generated by mapping the values of the corresponding kernel to the interval of gray level $[0, 255]$. The kernels located in different diagonal directions are basis functions for the Hermite kernel of different orders. The white line marks the basis functions for the kernel of order 5.

the Hermite kernel in this case also corresponds to non-coefficient Gaussian derivatives in 2D space. Figure 4 displays the kernel of different orders. The generic shape of each kernel resembles greatly to that of the kernel when $n = 1$. However, with the order increasing, the effect of the additional Gaussian function on the kernel becomes apparent. The contour of the high-order kernel is weakened greatly by the second Gaussian function.

3. Interpolation Functions and Basis Functions

3.1. *Prior knowledge*

Before we search for the interpolation functions and basis functions of the Hermite kernel, we must review a theorem derived by Yang *et al.* related to Hermite polynomials in the case of rotated coordinates.[32] This theorem can be taken advantage of demonstrating steerability of the Hermite kernel.

Theorem 2. *Let p, q be two non-negative integers. If the coordinates are rotated*

$$\begin{cases} \widehat{x} = x\cos\theta - y\sin\theta \\ \widehat{y} = x\sin\theta + y\cos\theta, \end{cases} \tag{12}$$

then the product of the Hermite polynomial can be expressed as:

$$H_p(\hat{x})H_q(\hat{y}) = \sum_{r=0}^{p+q} k(r,p,q,\theta)H_{p+q-r}(x)H_r(y), \tag{13}$$

where $k(r,p,q,\theta)$ *is a coefficient determined by* p, q, θ. *It can be calculated from*

$$k(r,p,q,\theta) = \begin{cases} \displaystyle\sum_{m=0}^{r} (-1)^m \binom{p}{m}\binom{q}{r-m} \cos^{p+r-2m}\theta \sin^{q-r+2m}\theta \\ \qquad\qquad \text{if } 0 \le r \le \min(p,q) \\ \displaystyle\sum_{m=0}^{p} (-1)^m \binom{p}{m}\binom{q}{r-m} \cos^{p+r-2m}\theta \sin^{q-r+2m}\theta \\ \qquad\qquad \text{if } p < r \le q \\ \displaystyle\sum_{m=0}^{q} (-1)^{r-m} \binom{p}{r-m}\binom{q}{m} \cos^{p-r+2m}\theta \sin^{q+r-2m}\theta \\ \qquad\qquad \text{if } q < r \le p \\ \displaystyle\sum_{m=0}^{p+q-r} (-1)^{p-m} \binom{p}{p-m}\binom{q}{r-p+m} \cos^{r-p+2m}\theta \sin^{2p+q-r-2m}\theta \\ \qquad\qquad \text{if } \max(p,q) < r \le p+q. \end{cases} \tag{14}$$

Theorem 2 describes an interesting rule of Hermite polynomials when coordinate rotation occurs. Inspired by this theorem, we have the possibility to derive the interpolation and basis function set for the Hermite kernel.

3.2. *Derivation of interpolation functions and basis functions*

We first define the coordinates which are used throughout the entire paper. The coordinates are defined in such fashion, that the x direction is pointing to the right and the y direction is pointing up. Rotation of an image by the angle θ is counter-clockwise. Essentially, image rotation only changes the position of the image, it never changes its content. Therefore, given any angle θ, the rotated version of Eq. (6) follows the relationship

$${}^{\theta}B^n_{p,q}(\hat{x},\hat{y};\sigma) = B^n_{p,q}(x,y;\sigma). \tag{15}$$

A more explicit expression can be obtained with the help of Eq. (12)

$${}^{\theta}B^n_{p,q}(x,y;\sigma) = B^n_{p,q}(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta;\sigma). \tag{16}$$

With the substitution of Eq. (6) into Eq. (16), it can be further expanded as follows

$${}^{\theta}B^n_{p,q}(x,y;\sigma) = H_p\left(\frac{x\cos\theta + y\sin\theta}{\sigma}\right) H_q\left(\frac{-x\sin\theta + y\cos\theta}{\sigma}\right) \exp\left(-\frac{nx^2+ny^2}{2\sigma^2}\right). \tag{17}$$

Using the conclusion of Theorem 2, Eq. (17) can be rewritten as:

$$\begin{aligned} {}^{\theta}B^{n}_{p,q}(x,y;\sigma) &= \left(\sum_{r=0}^{p+q} k(r,p,q,-\theta) H_{p+q-r}\Big(\frac{x}{\sigma}\Big) H_r\Big(\frac{y}{\sigma}\Big)\right) \exp\left(-\frac{nx^2+ny^2}{2\sigma^2}\right) \\ &= \sum_{r=0}^{p+q} l(r,p,q,\theta)\left(H_{p+q-r}\Big(\frac{x}{\sigma}\Big) H_r\Big(\frac{y}{\sigma}\Big) \exp\left(-\frac{nx^2+ny^2}{2\sigma^2}\right)\right) \\ &= \sum_{r=0}^{p+q} l(r,p,q,\theta) B^{n}_{p+q-r,r}(x,y;\sigma). \end{aligned} \tag{18}$$

Equation (18) yields us one set of interpolation functions and basis functions to steer Eq. (6). Specifically, $l(r,p,q,\theta)$ serves as an interpolation function, with the corresponding explicit form

$$l(r,p,q,\theta) = k(r,p,q,-\theta). \tag{19}$$

The basis functions, on the other hand, are a complete set of the Hermite kernel

$$\begin{aligned} \Phi(p,q) &= B^{n}_{i,j}(x,y;\sigma) \\ &= H_i\Big(\frac{x}{\sigma}\Big) H_j\Big(\frac{y}{\sigma}\Big) \exp\left(-\frac{nx^2+ny^2}{2\sigma^2}\right) \quad i,j \in \mathbf{N}^* \quad \text{and} \quad i+j=p+q. \end{aligned} \tag{20}$$

As long as p and q are known, the complete set of basis functions can be found without difficulties. Figures 2–4 illustrate the general shapes of basis functions. When $p+q$ is fixed, the corresponding basis functions are located in the diagonals of image squares as shown in figures above. For example, the white lines crossing the image squares cover the basis functions for the kernel of order 5. Also, the interpolation functions can be computed from Eq. (19).

Steerability can be obviously generalized to 3D case. 3D rotation is defined by rotating α along the z axis, then $-\beta$ along y axis and finally γ along x axis[a]

$$\mathbf{R} = \mathbf{R_x}(\gamma)\mathbf{R_y}(-\beta)\mathbf{R_z}(\alpha), \tag{21}$$

where

$$\mathbf{R_z}(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{22}$$

$$\mathbf{R_y}(-\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \tag{23}$$

[a] Rotation 3D, "http://en.wikipedia.org/wiki/Rotation_matrix"

and

$$\mathbf{R}_{\mathbf{x}}(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix}. \tag{24}$$

Equation (21) represents a 3D rotation achieved by three successive 2D rotation defined by Eq. (12). Using the property mentioned in Theorem 2, then for each 2D rotation defined by Eqs. (22)−(24), we can obtain the expression of the 3D Hermite kernel of order (p, q, r) in the following form:

$$\begin{aligned} {}^{\mathbf{R}}B^n_{p,q,r}(x, y, z; \sigma) &= \sum_{s=0}^{q+r} k(s, q, r, -\gamma) \sum_{t=0}^{p+s} k(t, p, s, \beta) \\ &\quad \times \sum_{u=0}^{p+q+r-t} k(u, p+s-t, q+r-s, -\alpha) B^n_{p+q+r-t-u,u,z}(x, y, z; \sigma) \\ &= \sum_{i=1}^{L(p,q,r)} \mathrm{con}_i(p, q, r, -\alpha, \beta, -\gamma) B^n_{p+q+r-t-u,u,z}(x, y, z; \sigma), \end{aligned} \tag{25}$$

where $L(p, q, r)$ is a certain number determined by p, q, r and con_i represents a constant sequence specially related to $p, q, r, -\alpha, \beta$ and $-\gamma$. Likewise, the interpolation functions are given as con_i and the basis functions are denoted by $B^n_{p+q+r-t-u,u,z}(x, y, z; \sigma)$. A total of $L(p, q, r)$ basis functions to construct the rotated kernel of order (p, q, r).

Following is an example for a 2D case $p + q = 3$. This example will show us how the transformed function is synthesized by the basis function set. From Figs. 5(a)−5(d), the images show the kernels $B^1_{3,0}$, $B^1_{2,1}$, $B^1_{1,2}$ and $B^1_{0,3}$. According to the previous analysis, they serve as the basis functions of the Hermite kernel of order 3 ($p + q = 3$), which are non-coefficient Hermite functions in 2D space. It should be noted that the basis functions $B^1_{3,0}$, $B^1_{2,1}$, $B^1_{1,2}$ and $B^1_{0,3}$ in this experiment are generated with $\sigma = 4.0$ and shown in the windows whose sizes are 41×41 pixels. These four basis functions are used to produce the rotated function and are designed for convolution with the tested image. For presentation purposes, the images in Figs. 5(a)−5(e) are magnified. Figure 5(e) shows the transformed version of $B^1_{2,1}$ by rotating $\frac{\pi}{3}$ and is actually synthesized from the linear combination of $B^1_{3,0}$, $B^1_{2,1}$, $B^1_{1,2}$ and $B^1_{0,3}$. We can compute the coefficient $l(r, 2, 1, \frac{\pi}{3})$ of the linear combination from Eq. (19). Its values are saved in the vector $\mathbf{K} = [-0.2165, -0.6250, -0.2165, 0.3750]$. As convolution is a linear operation, we can synthesize the response of an image to the rotated version of $B^1_{2,1}$ from the response of the image to the basis functions. Here, a gray image shows a white circle in black background in (f). The result of convoluting (f) with (e) can be indirectly obtained by the linear combination of (g) to (j), which are the responses of (f) to (a), (b), (c) and (d), respectively. The coefficient of the linear combination is the element of vector $\mathbf{K}$.

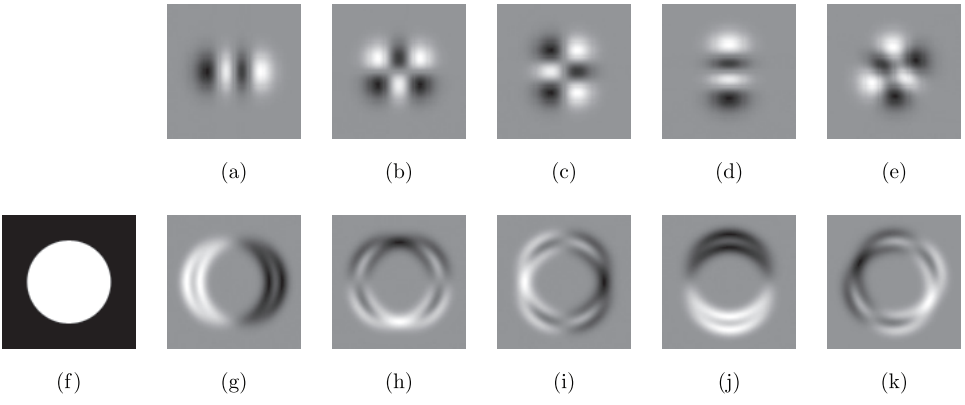


Fig. 5. Hermite kernels with $(p+q=3)$ are shown in (a)−(d), which correspond to $B^1_{3,0}$, $B^1_{2,1}$, $B^1_{1,2}$ and $B^1_{0,3}$, respectively. The rotation of (b) by $\frac{\pi}{3}$ is displayed in (e). It is actually implemented by $(-0.2165B^1_{3,0} - 0.6250B^1_{2,1} - 0.2165B^1_{1,2} + 0.3750B^1_{0,3})$. (f) A gray image containing a circle. The images from (g) to (j) give the results of convoluting (f), respectively, with (a)−(d). (k) The linear combination of (g) to (j) with the corresponding coefficient denoted by vector $\mathbf{K}$. (k) is actually identical to the direct convolution between (f) and (e).

4. Applications

In this section, we introduce several applications of the Hermite kernel, especially its steerability in image processing. Freeman and Adelson proposed several fundamental applications of the steerable filter, such as local orientation estimation, adaptive filtering, contour detection and shape-from-shading analysis.[9] The Hermite kernel can certainly also serve as such a filter and accomplish most of these tasks. Considering the reported work, we provide three examples which effectively demonstrate the steerability of the Hermite kernel and present its application in image processing. Strictly speaking, we focus our attention on indirect moment computation, operator approximation and orientation adaptive filtering based on the Hermite kernel.

4.1. *Indirect moment computation for rotated image*

When steerability is mentioned, the first application that comes into mind are steerable filters and their impressive efficiency related to convolution with an arbitrary oriented filter. Indeed such application of Hermite kernel will be presented too, however, the utilization of steerability goes far beyond simplifying convolution. It can be used for example to compute moments or transform coefficients related to rotation (for a survey how moments can be used in image analysis see Ref. 7). As the first application, we present moment computation with respect to the Hermite kernel, where the computation of Gaussian−Hermite moments (GHMs) serves as an example. GHM is defined as[32]:

$$\eta_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \psi_{p,q}(x,y;\sigma) f(x,y) dxdy, \tag{26}$$

with

$$\psi_{p,q}(x,y;\sigma) = c(p,q,\sigma)B^1_{p,q}(x,y;\sigma), \tag{27}$$

where

$$c(p,q,\sigma) = \left(2^{p+q}p!q!\pi\sigma^2\right)^{-\frac{1}{2}}. \tag{28}$$

For an image rotated by angle θ, its GHM is computed correspondingly by:

$$\eta^{\theta}_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \psi_{p,q}(x,y;\sigma)f^{\theta}(x,y)dxdy, \tag{29}$$

where f^{θ} represents the image after rotation. With Eqs. (26)–(28), Eqs. (16) and (18), we can simplify Eq. (29) to

$$\begin{aligned}
\eta^{\theta}_{p,q} &= c(p,q,\sigma)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)^{-\theta}B^1_{p,q}(x,y;\sigma)dxdy \\
&= c(p,q,\sigma)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\sum_{r=0}^{p+q} l(r,p,q,-\theta)B^1_{p+q-r,r}(x,y;\sigma)dxdy \\
&= \sum_{r=0}^{p+q}\frac{c(p,q,\sigma)l(r,p,q,-\theta)}{c(p+q-r,r,\sigma)}\eta_{p+q-r,r} \\
&= \sum_{r=0}^{p+q}\sqrt{\frac{(p+q-r)!r!}{p!q!}}l(r,p,q,-\theta)\eta_{p+q-r,r}.
\end{aligned} \tag{30}$$

When p or q is higher, the factorial operation for such number will result in numerical overflow. Under this condition, the following equation is advised to be used instead.

$$\eta^{\theta}_{p,q} = \sum_{r=0}^{p+q}\sqrt{\prod_{i=1}^{|\max(p+q-r,r)-\min(p,q)|}\frac{\min(p,q)+i}{\min(p+q-r,r)+i}}l(r,p,q,-\theta)\eta_{p+q-r,r}. \tag{31}$$

When it comes to the digital image, moment computation from Eq. (26) and Eqs. (30) and (31) is implemented by zeroth-order approximation (ZOA). Thus, for a digital image whose size is $N_x \times N_y$ pixels, its GHM is actually computed from

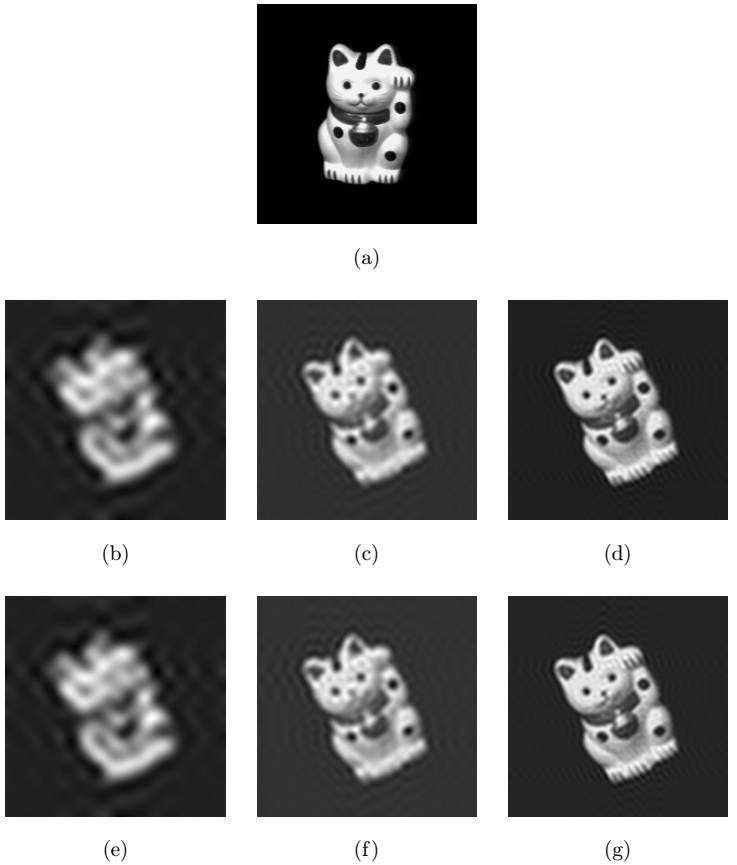
$$\eta_{p,q} = \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\psi_{p,q}(x_i,y_j;\sigma)f(x_i,y_j). \tag{32}$$

Likewise, the same method can be used to calculate the moments of the rotated image. According to the computation for the images before and after rotation, $\psi_{p,q}(x_i,y_j;\sigma)$ are implemented by sampling a square region. The basis functions are the same for the original image and the rotated one. However, there are changes in the image rotation because when rotating a digital image resampling and interpolation take place, leading to changes in the image intensity function and coordinates. Inevitably, some errors are produced in the moments of the rotated image, however Eq. (30) offers a

way to avoid such errors. It indicates that when both the moments of the original image and the rotation angles are known, the moments of the rotated image can be computed successfully without generating the rotated image. Thus, the errors generated by resampling and interpolation are completely avoided. In the rest of this subsection, moment computation by Eq. (30) is denoted as indirect computation and that by Eq. (32) as direct computation. Clearly, both indirect and direct moment computations have significance only for computation of the rotated image. Moreover, indirect moment computation provides theoretical results in moment computation.

An experiment was designed to verify indirect moment computation and show the influences of different interpolation methods on moment computation. A gray image showing a Chinese cat was selected as the original image. The image is illustrated by Fig. 6(a) and its size is 200×200 pixels. Its rotated versions were generated by a program coded in Matlab. We successively selected "Nearest", "Bilinear" and "Bicubic" interpolation methods to accomplish the discrete rotation operation. $\sigma = 0.25$ for moment computation in this experiment. A complete set of GHMs were computed for the rotated images, several moments are listed in Table 1. In Table 1, "Nearest", "Bilinear" and "Bicubic" represent direct moment computation with the rotated images generated by these interpolation methods. Obviously, when the angle is 90° or the multiples of 90°, there are no interpolation operations involved in the rotated images. Correspondingly, direct moment computation yields the same values as indirect moment computation. In this case, direct moment computation is capable of giving true values; however, for general angles such as 10°, 70°, 140°, 230° and 315°, direct moment computation produces slightly different moment values depending on the different interpolation methods. In other words, the errors are created during direct moment computation and they will definitively influence our work. On the contrary, indirect moment computation avoids such errors and offers theoretical values of moments. This experiment shows that indirect moment computation offers more accurate moment computation than the traditional method.

Indirect moment computation can be used in the cases when accurate moment computation for rotated images is required. Accurate rotation of a digital image is necessary in practice and is accomplished by Hermite expansions.[15] Accurate rotation will be achieved if there is no practical rotation operation involved in generation of the rotated images. We use an example of image reconstruction to show how indirect moment computation realizes accurate rotation. Figure 6(a) is taken as the original image. We conduct the reconstruction with two implementations. In the first implementation, we rotated the image by 30° first and then reconstructed the rotated image using the method proposed by Yang and Dai.[31] The reconstructed images are shown in Figs. 6(b)−6(d). Comparatively, in the second implementation we computed the moments of a rotated image using Eqs. (30) and (31) without generating the rotated image. Subsequently, we reconstructed the images from the computed moments. The results are illustrated in Figs. 6(e)−6(g). Figure 6 suggests that the reconstructed images from these two implementations are almost the



(a)

(b) (c) (d)

(e) (f) (g)

Fig. 6. Chinese cat and its reconstructed versions. (a) The original image. (b)−(d) Show the reconstructed rotated image from GHMs of orders $(0, 0)$ up to $(19, 19)$, $(39, 39)$ and $(59, 59)$, respectively. (e)−(g) Display the reconstructed images using the implementation of indirect moment computation. The moments are used for the same orders as in (b)−(d).

same; the differences are barely distinguishable by a human eye. This experiment demonstrates that indirect moment computation is effective when computing moments of rotated images. Moreover, it can be used to reconstruct images rotated by any angle. It frees us from generating rotated images and avoids unnecessary errors. Therefore, indirect moment computation is much more efficient than the traditional method used for computing moments of rotated images.

Additionally, it should be noted that indirect moment computation is still valid when computing Hermite moments, coefficients of the Hermite transform, and even the Gaussian derivatives. However, some adaptive modifications must be implemented, since these moments, the transform and filters are substantially based on the Hermite kernel, from which steerability is inherited.

Table 1. GHMs of the rotated images calculated by indirect computation and direct computation.

	GHM	$\eta_{3,8}$	$\eta_{4,9}$	$\eta_{5,4}$	$\eta_{6,6}$	$\eta_{7,6}$	$\eta_{9,2}$	$\eta_{10,8}$	$\eta_{11,0}$
	Indir (truth)	−4.6931	0.7827	−1.2782	−2.0752	−2.2959	−2.4376	0.5978	4.1184
10°	Nearest	−4.7312	0.7563	−1.2124	−2.0624	−2.2746	−2.4045	0.6369	4.1124
	Bilinear	−4.6825	0.7820	−1.2758	−2.0698	−2.2909	−2.4300	0.5956	4.1105
	Bicubic	−4.6999	0.7853	−1.2793	−2.0765	−2.3005	−2.4393	0.5982	4.1221
	Indir (truth)	4.0096	−0.7263	5.2284	−1.1111	2.7230	1.5008	1.4193	−4.1303
70°	Nearest	4.0680	−0.7286	5.1940	−1.1039	2.6616	1.5365	1.4135	−4.1095
	Bilinear	3.9986	−0.7216	5.2161	−1.1100	2.7136	1.4966	1.4139	−4.1169
	Bicubic	4.0164	−0.7236	5.2313	−1.1106	2.7230	1.4981	1.4212	−4.1314
	Indir (truth)	−3.3539	−1.5266	−2.7294	0.5858	3.9403	−1.5822	−2.3991	−0.7505
140°	Nearest	−3.3379	−1.5150	−2.8227	0.6058	3.8640	−1.5698	−2.3869	−0.7457
	Bilinear	−3.3420	−1.5260	−2.7177	0.5844	3.9322	−1.5780	−2.3886	−0.7446
	Bicubic	−3.3581	−1.5243	−2.7298	0.5953	3.9410	−1.5819	−2.3942	−0.7477
	Indir (truth)	4.7100	4.4772	0.6517	0.5858	3.5347	2.8467	−1.7949	0.8491
230°	Nearest	4.6986	4.4188	0.6428	0.6058	3.5472	2.8084	−1.7722	0.8895
	Bilinear	4.6970	4.4667	0.6520	0.5844	3.5238	2.8397	−1.7892	0.8459
	Bicubic	4.7168	4.4770	0.6540	0.5953	3.5306	2.8434	−1.7919	0.8482
	Indir (truth)	5.5294	1.7131	4.8924	−0.2674	−2.4260	3.3335	−1.8308	1.0150
315°	Nearest	5.5891	1.6997	4.8792	−0.3280	−2.4550	3.3365	−1.7940	0.9926
	Bilinear	5.5249	1.7067	4.8848	−0.2669	−2.4181	3.3250	−1.8256	1.0060
	Bicubic	5.5355	1.7100	4.8977	−0.2617	−2.4274	3.3416	−1.8236	1.0124

4.2. *Operator approximation*

In this subsection, we construct operators designed for detecting line features based on the Hermite kernel. Lines are important features in low level machine vision. So far, much research has been done on detecting lines in images and satisfactory results have been achieved. Here, we review Davies' work related to line detection from orthogonal masks.[5,6] Davies proposed two sets of operators whose sizes are 3×3 and 7×7 pixels, respectively. The operator L_0 whose size is 7×7 pixels has the following explicit form:

$$L_0 = \begin{pmatrix} 0 & 0 & -1 & -2 & -1 & 0 & 0 \\ 0 & 0 & -1 & -2 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 2 & 2 & 0 & 0 & 0 & 2 & 2 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & -2 & -1 & 0 & 0 \\ 0 & 0 & -1 & -2 & -1 & 0 & 0 \end{pmatrix}. \tag{33}$$

Davies' operators show good performance in detecting lines and estimating their orientation. The core of the method is based on generating two operators which have generic shapes as shown in Fig. 7. Given any image $f(x, y)$ we can obtain the effective line contrast by convoluting the image with the operators mentioned above. The

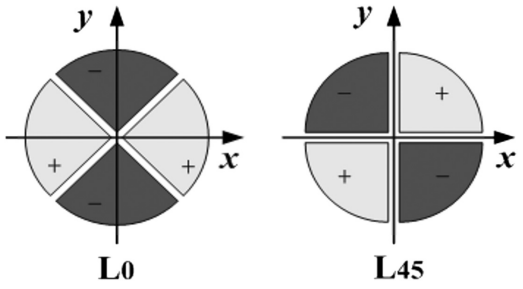


Fig. 7. The generic shapes of Davies' operators for line feature detection.

following equation offers the measurement:

$$M_L = \sqrt{(L_0 * f)^2 + (L_{45} * f)^2}, \tag{34}$$

where the symbol "*" represents convolution. It is possible to implement these two orthogonal operators from the Hermite kernel using its steerability. It is obvious that the Hermite kernel $B^1_{1,1}$ has a similar shape to that of L_{45}. However, there are no existing kernels whose shapes correspond to L_0. Both Davies' operators are similar, the only difference is the phase in the spatial domain. Specifically, L_0 is the rotated version of L_{45} with the clockwise rotation by $\frac{\pi}{4}$. Therefore, we can use the rotated version of $B^1_{1,1}$ to implement L_0. The interpolation functions are given in vector $\mathbf{C}$ whose expression is $\mathbf{C} = [-\cos\theta\sin\theta, \cos^2\theta - \sin^2\theta, \cos\theta\sin\theta]_{\theta=-\frac{\pi}{4}}$. From left to right, each element of $\mathbf{C}$ represents the corresponding interpolation function for $B^1_{2,0}$, $B^1_{1,1}$ and $B^1_{0,2}$, respectively. Figure 8 describes the way how the rotated version of $B^1_{1,1}$ is implemented using $B^1_{2,0}$, $B^1_{1,1}$ and $B^1_{0,2}$.

Since different σ results in different approximating operators, we choose σ which provide the maximum correlation coefficient between Davies' operator (L_0) and the approximating operator. For σ with values ranging from 0.50 to 2.0 with the

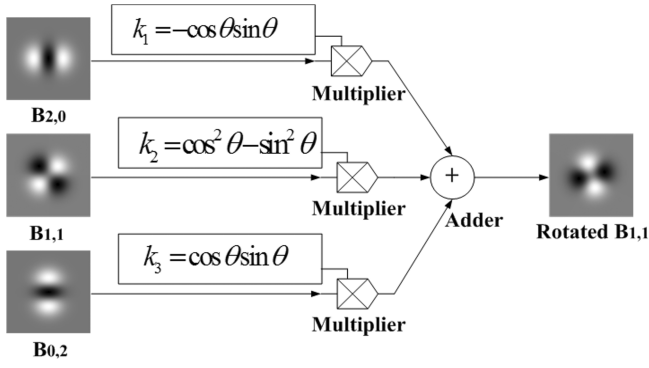


Fig. 8. Generation of the rotated $B^1_{1,1}$ from the Hermite kernel ($p + q = 2$).

increment of 0.1, we find that $\sigma = 0.8$ produces the maximum correlation coefficient. The approximating operators were tested on an image of a Prague street (Czech Republic). As to make a comparison, we also use Davies' operators to conduct the experiment. Figure 9 gives the results from these two sets of operators. When $\sigma = 0.8$

(a) (b) (c) (d) (e) (f) (g)

Fig. 9. Line feature detection from orthogonal operators. (a) Shows the original image whose size is 400×400 pixels. (b) The detection result by using Davies' operators whose sizes are 7×7 pixels. The binary image for enhancing line features is shown in (c). (d) and (e) show the corresponding detection result and the binary image from the approximating operators whose sizes are 7×7 pixels with $\sigma = 0.8$. The result using approximating operators whose sizes are 9×9 pixels with $\sigma = 1.2$ are displayed in (f) and (g). It should be noted that the threshold for generating binary images is set to 0.3 for all cases and the range of gray level is mapped to the interval $[0, 1]$.

and the size of the approximating operator is 7×7 pixels, the results show that both methods are competent to detect lines, because the road lamp, the roof edges and the main separation lines on the wall are selected successfully. Moreover, the binary images show that Davies' operators perform slightly better than the approximating operators, since the former contains the windows' edges and separate lines for the floor, which are poorly exhibited in the latter. But, when we implement a set of operators whose sizes are 9×9 pixels, more lines are highlighted by the approximating operators. The obvious improvement is seen in the appearance of separation lines of two floors and the vertical windows edges. Undoubtedly, it is possible to implement such operators with different σ and different operator size. Apparently, this experiment reflects the validity of steerability and it also shows that the Hermite kernel can be used as a substitute for certain standard operators. Furthermore, the approximating operators are determined by parameters. Various approximating operators will be generated depending on the selection of different parameters. In this respect, approximating operators are more flexible than certain standard operators in practice. We also noted that the coefficients of operator influence the detection results. Pouliquen *et al.* discussed optimal methods how to derive coefficients of these two operators.[19] However, the operators implemented by the Hermite kernel are not among the optimal coefficients selected. The coefficients are actually the sampling values of the continuous Hermite kernel in 2D space. In order to better approximate the continuous kernel, we sample the kernel every half pixel. The approximating operators whose size is 7×7 pixels are given in Appendix.

4.3. *Orientation adaptive filtering*

The effective application of steerable filters is orientation adaptive filtering. In order to execute orientation adaptive filtering, it is first necessary to acquire orientation information. So far, orientation estimation is still an important task, especially in processing medical, seismic and microscopic images containing directional textures or structures. Research on orientation estimation provided many important findings; one of the most significant and widely utilized in medical image processing being orientation estimation based on the Hessian matrix.[8,21] For an image $f(x, y)$, its Hessian matrix is defined as:

$$\mathbf{H} = \begin{pmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial y \partial x} & \dfrac{\partial^2 f}{\partial y^2} \end{pmatrix}. \tag{35}$$

The eigenvalues and eigenvectors of the Hessian matrix reveal information on local structures. More specifically, the eigenvector which corresponds to the eignevalue having the least absolute value determines local orientation.[18] In this subsection, we will first conduct orientation estimation and then perform an orientation adaptive filtering to enhance features in an image. Equation (35) indicates that the second

derivatives of an image have to be calculated in order to construct the Hessian matrix. Generally, such derivative computation is implemented by convoluting the second Gaussian derivatives with the image, since the convolution with the Gaussian derivative can tune the filter response to the specific widths of lines as well as to reduce the effect of noise.[21]

We can implement the Hessian matrix from the Hermite kernel, as we demonstrate that the Hermite kernel is substantially identical to Gaussian derivatives when $n = 2$. Note that we do not consider the coefficient of Gaussian derivatives; in this case the Gaussian derivatives of order $(2, 0)$, $(1, 1)$ and $(0, 2)$ have the same coefficients and the coefficients do not influence the filtering results. The Hessian matrix constructed using the Hermite kernel has the following form:

$$\mathbf{H}_B(x, y; \sigma_H, f) = \begin{pmatrix} B^2_{2,0}(x, y; \sigma_H) * f & B^2_{1,1}(x, y; \sigma_H) * f \\ B^2_{1,1}(x, y; \sigma_H) * f & B^2_{0,2}(x, y; \sigma_H) * f \end{pmatrix}. \tag{36}$$

Figure 10(a) displays a medical image showing clusters of mutually crossing and overlapping blood vessels.[2] The blood vessels vary in size: some are massive and clearly visible; however, other are thin and appear dim. Hence, it is demanding to enhance these blood vessels for diagnosis purposes. In this experiment, we first compute the orientation map based on Eq. (36). $B^2_{2,0}$, $B^2_{1,1}$ and $B^2_{0,2}$ are implemented by sampling continuous kernels in windows whose sizes are 21×21 pixels and $\sigma_H = 2.0$. It should also be noted that we sample the continuous kernels every half a pixel. Secondly, we conduct orientation adaptive filtering on the image with $B^2_{2,0}$ being the line enhancement filter. The filtering process is directed by orientation of each pixel in the image. The filtering result is shown in (b). The main vessels, located in the bottom half of the image, are well enhanced with satisfactory quality. The thin ones are still maintained and stand out at the same time. Performance under noisy condition is also evaluated. This image is contaminated by Gaussian white noise and a noisy image whose SNR $= -1.9288$ is obtained and demonstrated in (c). As shown in the image, some thin blood vessels and textures disappeared in the noise and some other vessels become more obscure. The filtering result of the noisy image is given in (d). Most vessels are emphasized with clear contours. Some small vessel crossings also stand out. We can visualize them in the bottom of (d). Two vessel crossings which appear to be tiny wedges are underlined in both the left and right part of the lower half of the image. However, in some parts, the noise degrades the contours of vessels; the orientation information is greatly affected by noise. Those parts are poorly enhanced, as shown in the upper left half of (d): the massive vessel is still obscure and its contour is shown vaguely.

In this step, we use different parameters to build different filters and compare the filtering results correspondingly. First, we evaluate the influence of σ on the filtering result. We use a larger σ to construct the filter and then filter (c) directionally. The result is illustrated in (e), where we clearly see that the massive vessels are further enhanced. Nevertheless, certain thin vessels become blurred. This experiment

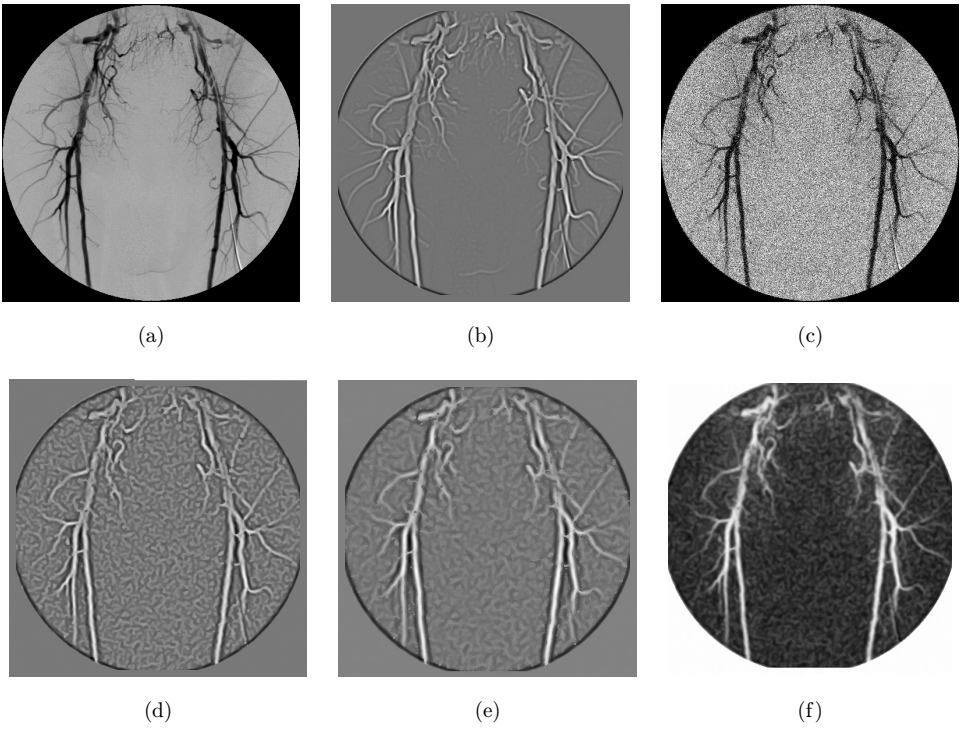


Fig. 10. Orientation adaptive filtering based on the Hermite kernel. (a) A medical image whose size is 512×512 pixels. $B^2_{2,0}$, $B^2_{1,1}$ and $B^2_{0,2}$ are used to build the Hessian matrix and are sampled using 21×21 pixel windows with $\sigma_H = 2.0$ ($W = 21, \sigma = 2.0$); (b) Orientation adaptive filtering of (a) by $B^2_{2,0}$ ($W = 25, \sigma = 1.5$). (c) Noisy image (SNR $= -1.9288$) created by adding Gaussian white noise to (a). (d) Orientation adaptive filtering of (c) by $B^2_{2,0}$ ($W = 25, \sigma = 2.0$). (e) Orientation adaptive filtering of (c) by $B^2_{2,0}$ ($W = 25, \sigma = 3.0$). (f) Orientation adaptive filtering of (c) by $B^5_{2,0}$ ($W = 25, \sigma = 3.0$).

indicates that σ determines the scale of the pattern. Different σ responds differently to the patterns with different scales. It also indicates that the Hermite kernel can be potentially used in multi-scale fashion, originally introduced to the Gaussian function and the corresponding kernel.[1] The effect of n on the filtering result can be evaluated as well, since the Hermite kernel behaves like a "low-pass" filter with n increasing. Correspondingly, (f) depicts the result of filtering by B^5_{20}. In summary, we observe that steerability is verified once again by the experiments. The experiments also demonstrate that Gaussian derivatives are subsets of the Hermite kernel, which implies that the Hermite kernel has a significant potential for practical applications.

5. Conclusion

In this paper, we define the Hermite kernel as Hermite polynomials modulated by the Gaussian envelope of different powers. Both Hermite functions and Gaussian

derivatives correspond to the Hermite kernel with different parameters. We investigate the steerability of the Hermite kernel based on a property of Hermite polynomials. We conclude that any order of the Hermite kernel is steerable, and the corresponding interpolation functions and basis functions can be easily obtained. Moreover, we formulate methods how to compute the explicit interpolation functions as well as the complete set of basis functions. Three examples, computation for GHMs, operator approximation and orientation adaptive filtering, are introduced to testify the steerability and expand on the applications. The results show validity of the inherent steerability together with its potential applications in image processing.

Finally, we emphasize that steerability offers a specific way to steer the high-order Hermite kernel, for instance, high-order Gaussian derivatives. So far, most researchers report applications of low-order Gaussian derivatives because they are characterized in low frequency to which the "general features" such as edges, contours, and corners, belong. With progress in related research fields, we are convinced that the high-order Gaussian derivatives and Hermite functions will be utilized in significant applications. Steerability will serve as a useful property to explore applications of the Hermite kernel.

Acknowledgments

This research was performed under the project No. P103/11/1552 financed by the Czech Science Foundation. Bo Yang was also supported by an ERCIM "Alain Bensoussan" Fellowship Programme. This Programme is funded by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

Appendix A

The Hermite kernel of order 2 is listed as follows. We sample Eq. (8) every half pixel with $\sigma = 0.8$ and the window size is 7×7 pixels ($W = 7, \sigma = 0.8$).

$$B^1_{2,0} = \begin{pmatrix} 0.3586 & 0.3355 & -0.0621 & -0.3448 & -0.0621 & 0.3355 & 0.3586 \\ 0.9522 & 0.8908 & -0.1648 & -0.9157 & -0.1648 & 0.8908 & 0.9522 \\ 1.7108 & 1.6006 & -0.2960 & -1.6452 & -0.2960 & 1.6006 & 1.7108 \\ 2.0798 & 1.9458 & -0.3599 & -2.0000 & -0.3599 & 1.9458 & 2.0798 \\ 1.7108 & 1.6006 & -0.2960 & -1.6452 & -0.2960 & 1.6006 & 1.7108 \\ 0.9522 & 0.8908 & -0.1648 & -0.9157 & -0.1648 & 0.8908 & 0.9522 \\ 0.3586 & 0.3355 & -0.0621 & -0.3448 & -0.0621 & 0.3355 & 0.3586 \end{pmatrix}, \tag{A.1}$$

$$B^1_{1,1} = \begin{pmatrix} -0.4181 & -0.7401 & -0.6648 & 0.0000 & 0.6648 & 0.7401 & 0.4181 \\ -0.7401 & -1.3101 & -1.1769 & 0.0000 & 1.1769 & 1.3101 & 0.7401 \\ -0.6648 & -1.1769 & -1.0572 & 0.0000 & 1.0572 & 1.1769 & 0.6648 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.6648 & 1.1769 & 1.0572 & 0.0000 & -1.0572 & -1.1769 & -0.6648 \\ 0.7401 & 1.3101 & 1.1769 & 0.0000 & -1.1769 & -1.3101 & -0.7401 \\ 0.4181 & 0.7401 & 0.6648 & 0.0000 & -0.6648 & -0.7401 & -0.4181 \end{pmatrix}, \tag{A.2}$$

$$B^1_{0,2} = (B^1_{2,0})^T, \tag{A.3}$$

where the symbol "T" represents the transposition operation of a matrix.

References

1. P. Andrew, Scale-space filtering, in *Proc. 8th Int. Joint Conf. Artificial Intelligence*, San Francisco, USA (1983), pp. 1019–1022.
2. S. Barré, available at: http://barre.nom.fr/medical/samples/#CT-MONO2-16-chest (accessed 22 June 2012).
3. W. Beil, Steerable filters and invariance theory, *Pattern Recogn. Lett.* **15** (1994) 453–460.
4. P. E. Danielsson, Rotation-invariant linear operators with directional response, *Int. Conf. Pattern Recognition*, Miami, USA, December 1980.
5. E. R. Davies, Designing efficient line segment detectors with high orientation accuracy, *Conf. 6th, Image Processing and Its Applications*, Dublin, Ireland, July 1997, pp. 636–640.
6. E. R. Davies, D. R. Mason, M. Bateman, J. Chambers and C. Ridgway, Linear feature detectors and their application to cereal inspection, in *Proc. of EUSIPCO, Signal Processing*, Rhodes, Greece, September 1998, pp. 2561–2564.
7. J. Flusser, T. Suk and B. Zitova, *Moments and Moment Invariants in Pattern Recognition* (Wiley, Chicester, 2009), p. 317.
8. A. F. Frangi, W. J. Niessen, K. L. Vincken and M. A. Viergever, Multiscale vessel enhancement filtering, *Lect. Notes Comput. Sci.* **1496** (1998) 130–137.
9. W. T. Freeman and E. H. Adelson, The design and use of steerable filters, *IEEE Trans. Pattern Anal. Mach. Intell.* **13** (1991) 891–906.
10. H. Knutsson and G. H. Granlund, Texture analysis using two-dimensional quadrature filters, *Workshop Computer Architecture for Pattern Analysis and Image Database Manangement*, Pasadena, USA, October 1983.
11. R. Massey and A. Refregier, Polar shapelets, *Mon. Not. R. Astron. Soc.* **363** (2005) 197–210.
12. J. B. Martens, The Hermite transform — theory, *IEEE Trans. Acoust. Speech Signal Process.* **38** (1990) 1595–1606.
13. M. Michaelis and G. Sommer, A lie group approach to steerable filters, *Pattern Recogn. Lett.* **16** (1995) 1165–1174.
14. M. Muhlich, D. Friedrich and T. Aach, Design and implementation of multisteerable matched filters, *IEEE Trans. Pattern Anal. Mach. Intell.* **34** (2012) 279–291.

15. W. Park, G. Leibon, D. N. Rockmore and G. S. Chirikjian, Accurate image rotation using Hermite expansions, *IEEE Trans. Image Process.* **18** (2009) 1988−2003.
16. P. Perona, Steerable-scalable kernels for edge detection and junction analysis, *Image Vision Comput.* **10** (1992) 663−672.
17. P. Perona, Deformable kernels for early vision, *IEEE Trans. Pattern Anal. Mach. Intell.* **17** (1995) 488−499.
18. Z. Peter, B. Perrenot, A. Larrue and F. Peyrin, Hessian based orientation analysis of the canal network in corticalbone micro CT images, *Symp. Nuclear Science* (2008), pp. 4361−4363.
19. F. L. Pouliquen, Operateurs discrets pour l'estimation adaptative et optimale de l'orientation application àl'imagerie sismique, Ph.D. thesis, L'université Bordeaux 1 (2003) (in French).
20. A. Refregier, Shapelets — A method for image analysis, *Mon. Not. R. Astron. Soc.* **338** (2003) 35−47.
21. Y. Sato, S. Nakajima, N. Shiraga and H. Atsumi, Three-dimensional multi-scale line filter for segmentation andvisualization of curvilinear structures in medical images, *Med. Image Anal.* **2** (1998) 143−168.
22. E. P. Simoncelli, W. T. Freeman, E. H. Adelson and D. J. Heeger, Shiftable multiscale transforms, *IEEE Trans. Inf. Theory* **38** (1992) 587−606.
23. E. P. Simoncelli and W. T. Freeman, The steerable pyramid: A flexible architecture for multi-scale derivative computation, *Int. Conf. Image Processing*, Washington DC, USA, October 1995, pp. 444−447.
24. E. P. Simoncelli and H. Farid, Steerable wedge filters for local orientation analysis, *IEEE Trans. Image Process.* **5** (1996) 1377−1382.
25. J. Shen, Orthogonal Gaussian-Hermite moments for image characterization, in *Proc. SPIE Intelligent Robots and Computer Vision*, Pittsburgh, USA, October 1997, pp. 224−233.
26. J. Shen, W. Shen and D. F. Shen, On geometric and orthogonal moments, *Int. J. Pattern Recogn. Artif. Intell.* **14** (2000) 875−894.
27. J. S. Taylor and S. L. Sun, A review of optimization methodologies in support vector machines, *Neurocomputing* **74** (2011) 3609−3618.
28. P. C. Teo, Theory and applications of steerable functions, Ph.D. thesis, Stanford University (1998).
29. P. C. Teo and Y. Hel-Or, Lie generators for computing steerable functions, *Pattern Recogn. Lett.* **19** (1998) 7−17.
30. M. Van Ginkel, Image analysis using orientation space based on steerable filters, Ph.D. thesis, Technische Universiteit Delft (2002).
31. B. Yang and M. Dai, Image reconstruction from continuous Gaussian-Hermite moments implemented by discrete algorithm, *Pattern Recogn.* **45** (2012) 1602−1616.
32. B. Yang, G. X. Li, H. L. Zhang and M. Dai, Rotation and translation invariants of Gaussian-Hermite moments, *Pattern Recogn. Lett.* **32** (2011) 1283−1298.
33. W. C. Yu, K. Daniilidis and G. Sommer, Approximate orientation steerability based on angular Gaussians, *IEEE Trans. Image Process.* **10** (2001) 193−205.

Bo Yang received his M.Sc. degree in Control Science and Engineering from the Northwestern Polytechnical University, Xi'an, China in 2007, his Ph.D. in Image Science from the University of Bordeaux 3 in 2011. Currently, he is a ERCIM Fellow and on a postdoctoral position at the Institute of Information Theory and Automation, Prague, in Czech Republic. His research interests include image processing and pattern recognition.

Jan Flusser received his M.Sc. degree in Mathematical Engineering from the Czech Technical University, Prague, Czech Republic in 1985, his Ph.D. in Computer Science from the Czechoslovak Academy of Sciences in 1990, and his D.Sc. degree in Technical Cybernetics in 2001. Since 1985 he has been with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague. Between 1995−2007, he was holding the position of head of department of image processing. Currently (since 2007) he is a director of the institute. He is a full professor of Computer Science at the Czech Technical University and at the Charles University, Prague, Czech Republic, where he gives undergraduate and graduate courses on digital image processing, pattern recognition, and moment invariants and wavelets. Jan Flusser's research interests cover moments and moment invariants, image registration, image fusion, multichannel blind deconvolution, and super-resolution imaging. He has authored and co-authored more than 150 research publications in these areas, including the monograph "Moments and Moment Invariants in Pattern Recognition" (Wiley, 2009), tutorials and invited/keynote talks at major international conferences. In 2007 Jan Flusser received the award of the chairman of the Czech Science Foundation for the best research project and won the prize of the Academy of Sciences of the Czech Republic for his contribution to image fusion theory. In 2010, he was awarded the prestigious SCOPUS 1000 Award presented by Elsevier.

Tomáš Suk received his M.Sc. degree in Electrical Engineering from the Czech Technical University, Faculty of Electrical Engineering, Prague, in 1987. He also received his CSc. degree (corresponds to Ph.D.) in Computer Science from the Czechoslovak Academy of Sciences, Prague, in 1992. From 1991 he has been a researcher with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague, member of the Department of Image Processing. He has authored 14 journal papers and 31 conference papers. He is a co-author of the book "Moments and Moment Invariants in Pattern Recognition" (Wiley, 2009). Tomáš Suk's research interests includes digital image processing, pattern recognition, image filtering, invariant features, moment and point invariants, geometric transformations of images, and applications in remote sensing, astronomy, medicine, botany and computer vision. In 2002, Tomáš Suk received the Otto Wichterle premium of the Academy of Sciences of the Czech Republic for young scientists.