Published in final edited form as:
IEEE Trans Image Process. 2009 September ; 18(9): 1988–2003. doi:10.1109/TIP.2009.2024582.

Accurate Image Rotation using Hermite Expansions

Wooram Park[1], **Gregory Leibon**[2], **Daniel N. Rockmore**[2], and **Gregory S. Chirikjian**[1]

[1] Department of Mechanical Engineering, Johns Hopkins University

[2] Departments of Mathematics and Computer Science, Dartmouth College

Abstract

In this paper we propose an approach for the accurate rotation of a digital image using Hermite expansions. This exploits the fact that if a 2D continuous band-limited Hermite expansion is rotated, the resulting function can be expressed as a Hermite expansion with the same band limit. Furthermore, the Hermite coefficients of the initial 2D expansion and the rotated expansion are mapped through an invertible linear relationship. Two efficient methods to compute the mapping between Hermite coefficients during rotation are proposed. We also propose a method for connecting the Hermite expansion and a discrete image. Using this method, we can obtain the Hermite expansion from a discrete image and vice versa. Combining these techniques, we propose new methods for the rotation of discrete images. We assess the accuracy of our methods and compare with existing FFT-based methods implementing three shears. We find that the method proposed here consistently has better accuracy than FFT-based methods.

Keywords

Image rotation; Hermite functions; Hermite expansions; Band-limited expansions

1 Introduction

Techniques for effecting a two-dimensional rotation of a discrete image are very important in many different disciplines. Examples include medical imaging, digital photography, and computer graphics. Unfortunately, the action of rotation is poorly matched with both the (necessarily) discretized representation of a digital image as well as the discrete Fourier transform, which is one of the most commonly used tools in image processing. Inaccuracy in image rotation can cause subtle problems. At the most superficial level, a composition of rotations that results in an overall rotation that is a multiple of 2π may not bring an image back to itself. This can have unfortunate effects in digital imaging. For example, in medical imaging, the loss of information during image rotation may cause the loss of small, but important, anatomical features.

The interactions among Fourier analysis, digitization, and rotation are more interesting and more subtle. While the classical continuous-domain Fourier transform is invariant under rotation, the discrete Fourier expansions used in image analysis do not share this invariance. This may seem surprising at first. Let us consider the Fourier transform, $\tilde{f}(\omega)$, for a nonperiodic function $f(\mathbf{x})$, where $\mathbf{x}, \omega \in \mathbb{R}^2$.

If we apply the Fourier transform to the rotated version of $f(\mathbf{x})$,

$$g(\mathbf{x})=f(R^{-1}\mathbf{x})$$

for some rotation R of the plane, then we have

$$\tilde{g}(\omega) = \tilde{f}(R^{-1}\omega).$$

In other words, the Fourier transform of a rotated function is the rotated Fourier transform. However, the continuous Fourier transform is not the one relevant for image analysis, rather in this setting we need to make use of the discrete Fourier transform (DFT). In this context, the DFT should be viewed as a sampled version of the Fourier series of a periodic function, which has a discrete (rather than continuous) spectrum. Under such assumptions the image is identified with a sampled function on the 2-torus (i.e., periodic in two independent directions) and rotation of the finite 2D lattice (naturally identified with the torus) is not compatible with the Fourier transform, much less the finite lattice, which is not generally mapped to itself under an arbitrary rotation. While this geometric incompatibility could be addressed via linear or spline interpolations, these are at best approximate methods. Since there is no mathematically exact way to rotate and resample the underlying band-limited Fourier series from which the discrete samples were assumed to be drawn, there is no way to "exactly" interpolate discrete values from the original grid to the rotated one.

In this paper we show how Hermite functions provide a representation better suited for rotation and discuss how they might be used for image representation. This derives largely from the fact that the Hermite functions of a given total degree span invariant spaces for the rotation operator. Their "near-eigenfunction" behavior implies that (1) finite Hermite expansions of a given highest degree (so-called band-limited Hermite expansions) are mapped back to expansions of the same degree under rotation and (2) there is an explicit analytic relation between the coefficients in the original and rotated expansions that is derived here using various special function relations.

The remainder of this paper is organized as follows: In Section 2, a brief literature review is given in the field of image rotations and Hermite-function-based image processing. In Section 3, using the fact that the rotation of a 2D band-limited Hermite expansion preserves the band limit, we derive the relationship between the Hermite coefficients before and after rotation. We propose two methods for implementing the conversion of the Hermite coefficients. In Section 4 we propose a method to connect the band-limited Hermite expansion and an arbitrary two-dimensional array of discrete data. Then in Section 5, we present computational results and compare the accuracy of our new methods with an existing FFT-based image rotation approach. Finally in Section 6 we summarize our conclusions.

2 Related works

Non-Hermite based methods for effecting rotation are the standard and there exist various approximate methods which are simple and efficient. Paeth showed that a 2D rotation can be effected by applying three shear transformations to a raster image [20]. This method has been widely investigated [19,24] and extended to three-dimensional rotations of image volumes [2,4,23]. However, as Toffoli et al. [23] indicated, shear transformations of discrete images introduce certain errors when local interpolation is applied. The main reason for this loss of information is that a discrete image is defined on a grid, and when performing shears, interpolations to new grid points must be performed. Local interpolation is an inexact step that destroys global information about the image. In spite of these disadvantages, an FFT-based rotation algorithm using three shears shows fast and accurate results [8,19]. The basic concept of that method is that one can apply the 3 shear processes to the original image and each shear process is implemented by the FFT. We will compare our new methods with this method.

For the purpose of achieving global rotations, steerable filters [5] can be a reasonable tool, since the class of the steerable filters has the special property of remaining band-limited under rotation. A steerable filter at any orientation can be constructed as a linear combination of the basis filters. This special kind of filters has been widely investigated. Steerable filters are widely used (see e.g., [6,25,26,29]) and have been extended using Lie group theory [14].

A Hermite transform can be viewed as a steerable filter [12,13,26]. Interestingly, to our knowledge, global image rotation using Hermite functions has not been investigated previously. The use of the Hermite transform (i.e., expansion of a function in $\mathbb{R}^2$ using Hermite functions) is a relatively new approach to image processing. The related literature includes approaches to image compression [25,26], local image analysis [12] and deblocking of a compressed image [16]. In the case of local image orientation analysis using the Hermite transform, many copies of small Gaussian windows are translated so that the whole image can be covered by a combination of Hermite polynomials and the windows to capture local orientational features of the image [12,13,25,26]. Therefore, it is difficult to adapt this method to perform a global rotation, even though the method shows successful performance in local feature analysis. In our approach, we consider only one Gaussian window placed at the center of the image. In practice, Hermite functions act as a filter, because they are the product of the Hermite polynomials and the Gaussian window.

3 Rotation of 2D Hermite Expansions

3.1 2D Hermite Expansions and Steerable Filters

Hermite polynomials are generated by the Rodrigues formula

$$H_n(x)=(-1)^n e^{x2}\frac{d^n}{dx^n}(e^{-x^2}).$$

Hermite functions are defined as [3,27]

$$h_n(x)=\frac{1}{2^{n/2}\sqrt{n!\,\sqrt{\pi}}}H_n(x)e^{-\frac{x^2}{2}}. \tag{1}$$

This definition satisfies the orthonormality condition $\int_{-\infty}^{\infty}h_m(x)h_n(x)dx=\delta_{mn}$. The 2D band-limited Hermite expansion can be defined as [22]

$$f(\mathbf{x})=f(x,y)=\sum_{m=0}^{N}\sum_{n=0}^{N-m}\widehat{f}_{m,n}h_m(x)h_n(y), \tag{2}$$

or equivalently

$$f(x,y)=\sum_{m=0}^{N}\sum_{n=0}^{m}\widehat{f}_{n,m-n}h_n(x)h_{m-n}(y), \tag{3}$$

where $\hat{f}_{n,m-n}$ is the Hermite coefficients.

A steerable filter of order m can be exactly constructed by taking linear combinations of the filters of order m [5]. Since the product of two Hermite functions is a steerable filter on the plane [12,26], its steerability can be written as [25]

$$\begin{aligned}\mathcal{R}^{\theta}(h_n(x)h_{m-n}(y)) &= h_n(x\cos\theta+y\sin\theta)h_{m-n}(\\ &\quad -x\sin\theta+y\cos\theta)=\sum_{k=0}^{m}S_{k,n}^{m}(\theta)h_k(x)h_{m-k}(y)\end{aligned} \tag{4}$$

where $\mathcal{R}(\cdot)$ is the rotation operator and $S_{k,n}^{m}(\theta)$ is the steering coefficients. If we use the following property of the Hermite polynomials [27]

$$H_n(z\sin t+w\cos t)=\sum_{k=0}^{n}C_n^k\sin^k t\cos^{n-k}tH_k(z)H_{n-k}(w), \tag{5}$$

where $C_n^k=n!/(k!(n-k)!)$, we can also derive (4) without prior knowledge of steerable filters, as was done in [21]

Using (4), the rotation of a 2D Hermite expansion of (3) can be written as

$$f^{\theta}(x,y)=\sum_{m=0}^{N}\sum_{k=0}^{m}\left(\sum_{n=0}^{m}\widehat{f}_{n,m-n}S_{k,n}^{m}(\theta)\right)h_k(x)h_{m-k}(y), \tag{6}$$

Note that the band limit of the resulting Hermite expansion is preserved. The new Hermite coefficients can be written as

$$\widehat{f}_{k,m-k}^{\theta}=\sum_{n=0}^{m}S_{k,n}^{m}(\theta)\widehat{f}_{n,m-n} \tag{7}$$

for $m = 0, 1, \ldots, N$. It is clear that the coefficients $\hat{f}^{\theta}$ and $\hat{f}$ are linearly and locally related. Namely, the relation can be written as

$$\mathbf{f}_m^{\theta}=S^m(\theta)\mathbf{f}_m, \tag{8}$$

where $S_{k,n}^{m}(\theta)$ is the $(k+1, n+1)$ element of the matrix, $S^m(\theta)$,

$$\mathbf{f}_m^{\theta}=\left[\widehat{f}_{0,m}^{\theta}\ \ \widehat{f}_{1,m-1}^{\theta}\ \ \cdots\ \ \widehat{f}_{m,0}^{\theta}\right]^T$$

and

$$\mathbf{f}_m=\left[\widehat{f}_{0,m}\ \ \widehat{f}_{1,m-1}\ \ \cdots\ \ \widehat{f}_{m,0}\right]^T.$$

3.2 Recurrence formulas for $S^m_{q,n}(\theta)$

For low orders, the transformation matrix, $S^m(\theta)$ can be derived using (4) and (5). The first few $S^m(\theta)$ matrices are

$$S^0(\theta)=1,$$

$$S^1(\theta)=\begin{pmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{pmatrix},$$

$$S^2(\theta)=\begin{pmatrix}\cos^2\theta & \sqrt{2}\cos\theta\sin\theta & \sin^2\theta\\ -\sqrt{2}\cos\theta\sin\theta & \cos^2\theta-\sin^2\theta & \sqrt{2}\cos\theta\sin\theta\\ \sin^2\theta & -\sqrt{2}\cos\theta\sin\theta & \cos^2\theta\end{pmatrix}.$$

$S^1(\theta)$ and $S^2(\theta)$ were derived also in [13] and were sufficient for local orientation analysis. However, we need the higher order results since we are looking at the case of large band limits for global image rotations. Instead of obtaining the closed form formula for $S^m(\theta)$ directly, we try to derive the recurrence formula for the elements of this transformation matrix.

Multiplying $h_q(x)h_{m-q}(y)$ on both sides of (4) and integrating over $\mathbb{R}^2$ gives

$$S^m_{q,n}(\theta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h_q(x)h_{m-q}(y)h_n(x\cos\theta+y\sin\theta)h_{m-n}(-x\sin\theta+y\cos\theta)dxdy. \tag{9}$$

Let us consider

$$S^{m+1}_{q+1,n}(\theta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h_{q+1}(x)h_{m-q}(y)h_n(x\cos\theta+y\sin\theta)h_{m+1-n}(-x\sin\theta+y\cos\theta)dxdy.$$

If we apply the recurrence formula for Hermite functions

$$\sqrt{n+1}h_{n+1}(x)=\sqrt{2}xh_n(x)-\sqrt{n}h_{n-1}(x), \tag{10}$$

then we have

$$\begin{aligned}&\sqrt{q+1}S^{m+1}_{q+1,n}(\theta)\\ &=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\sqrt{2}xh_q(x)-\sqrt{q}h_{q-1}(x)\right)h_{m-q}(y)h_n(x\cos\theta+y\sin\theta)h_{m+1-n}(-x\sin\theta+y\cos\theta)dxdy.\\ &=\sqrt{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}xh_q(x)h_{m-q}(y)h_n(x\cos\theta+y\sin\theta)h_{m+1-n}(-x\sin\theta+y\cos\theta)dxdy,\end{aligned}$$

since (4) implies that when $m' \neq m$,

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h_q(x)h_{m'-q}(y)h_n(x\cos\theta+y\sin\theta)h_{m-n}(-x\sin\theta+y\cos\theta)dxdy=0. \tag{11}$$

Similarly, it can be shown that

$$\begin{aligned}&\sqrt{m-q+1}S^{m+1}_{q,n}(\theta)\\ &\quad=\sqrt{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h_q(x)\cdot y\cdot h_{m-q}(y)h_n(x\cos\theta+y\sin\theta)h_{m+1-n}(-x\sin\theta+y\cos\theta)dxdy.\end{aligned}$$

Now, we compute

$$\begin{aligned}&\sqrt{q+1}S^{m+1}_{q+1,n}(\theta)\cos\theta+\sqrt{m-q+1}S^{m+1}_{q,n}(\theta)\sin\theta\\ &=\sqrt{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h_q(x)h_{m-q}(y)(x\cos\theta+y\sin\theta)h_n(x\cos\theta+y\sin\theta)h_{m+1-n}(-x\sin\theta+y\cos\theta)dxdy.\end{aligned}$$

Using the recurrence formula (10), we have

$$\begin{aligned}&\sqrt{q+1}S^{m+1}_{q+1,n}(\theta)\cos\theta+\sqrt{m-q+1}S^{m+1}_{q,n}(\theta)\sin\theta\\ &=\sqrt{n+1}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h_q(x)h_{m-q}(y)h_{n+1}(x\cos\theta+y\sin\theta)h_{m+1-n}(-x\sin\theta+y\cos\theta)dxdy\\ &+\sqrt{n}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h_q(x)h_{m-q}(y)h_{n-1}(x\cos\theta+y\sin\theta)h_{m+1-n}(-x\sin\theta+y\cos\theta)dxdy.\end{aligned}$$

Since the first term of the righthand side is zero due to (11), we have

$$\sqrt{q+1}S^{m+1}_{q+1,n}(\theta)\cos\theta+\sqrt{m-q+1}S^{m+1}_{q,n}(\theta)\sin\theta=\sqrt{n}S^{m}_{q,n-1}(\theta).$$

Similarly, it can be shown that

$$-\sqrt{q+1}S^{m+1}_{q+1,n}(\theta)\sin\theta+\sqrt{m-q+1}S^{m+1}_{q,n}(\theta)\cos\theta=\sqrt{m-n+1}S^{m}_{q,n}(\theta).$$

Consequently,

$$\begin{pmatrix} S^{m+1}_{q+1,n}(\theta) \\ S^{m+1}_{q,n}(\theta) \end{pmatrix} = \begin{pmatrix} 1/\sqrt{q+1} & 0 \\ 0 & 1/\sqrt{m-q+1} \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \sqrt{n}S^{m}_{q,n-1}(\theta) \\ \sqrt{m-n+1}S^{m}_{q,n}(\theta) \end{pmatrix}. \tag{12}$$

Likewise, we can apply the same process to $S^{m+1}_{q,n+1}(\theta)$ and $S^{m+1}_{q,n}(\theta)$ and have

$$\begin{pmatrix} S^{m+1}_{q,n+1}(\theta) \\ S^{m+1}_{q,n}(\theta) \end{pmatrix} = \begin{pmatrix} 1/\sqrt{n+1} & 0 \\ 0 & 1/\sqrt{m-n+1} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \sqrt{q}S^{m}_{q-1,n}(\theta) \\ \sqrt{m-q+1}S^{m}_{q,n}(\theta) \end{pmatrix}. \tag{13}$$

3.3 Properties of $S^m(\theta)$

It is easy to show that

$$\{S^m(\theta)\}^T S^m(\theta) = I_m, \tag{14}$$

where I_m is the $(m+1)\times(m+1)$ identity matrix. The detailed derivation is given in Appendix A. Since $S^m(\theta)$ is a real-valued square matrix and (14) holds, $S^m(\theta)$ is an orthogonal matrix.

On the other hand, using (9) we can compute $S^m_{p,q}(-\theta)$ as

$$\begin{aligned} S^m_{p,q}(-\theta) &= \int_{\mathbb{R}^2} h_p(x)h_{m-p}(y)h_q(x\cos\theta - y\sin\theta)h_{m-q}(x\sin\theta + y\cos\theta)dxdy \\ &= \int_{\mathbb{R}^2} h_q(x')h_{m-q}(y')h_p(x'\cos\theta + y'\sin\theta)h_{m-p}(-x'\sin\theta + y'\cos\theta)dx'dy' = S^m_{q,p}(\theta). \end{aligned}$$

Note that we used the coordinate conversion $x' = x\cos\theta - y\sin\theta$ and $y' = x\sin\theta + y\cos\theta$. Therefore, $S^m(-\theta) = S^m(\theta)^T = S^m(\theta)^{-1}$. This guarantees the invertibility of the rotation process, since $S^m(\theta)^{-1}$ always exists. In addition, the orthogonality of $S^m(\theta)$ guarantees the stable inversion. Consequently, the rotation process by (7) is lossless in exact arithmetic.

Since the rotation on the plane is decomposable and commutative, we have $S^m(\theta_1 + \theta_2) = S^m(\theta_1)S^m(\theta_2) = S^m(\theta_2)S^m(\theta_1)$. Thus, $S^m(\theta)$ is the exponential of a skew-symmetric matrix multiplied by θ as

$$S^m(\theta) = \exp(\theta\Omega^m), \tag{15}$$

where Ω^m is the skew-symmetric matrix. Ω^m can be obtained by

$$\left.\frac{dS^m(\theta)}{d\theta}\right|_{\theta=0} = \Omega^m.$$

Explicitly, we can compute

$$\Omega^m_{q,n} = \frac{d}{d\theta}(S^m_{q,n}(\theta))|_{\theta=0} = \sqrt{n}\,\sqrt{m-n+1}\,\delta_{q,n-1} - \sqrt{n+1}\,\sqrt{m-n}\,\delta_{q,n+1}.$$

The detailed derivation is given in Appendix B.

3.4 Relation between $S^m(\theta)$ and Generalized Associated Legendre Function

In this subsection, we connect $S^m(\theta)$ and the generalized associated Legendre Function, $P^l_{mn}(x)$ [27]. This is expected to give us a way to compute $S^m(\theta)$ by simple matrix multiplications.

The irreducible unitary representation (IUR) matrix elements of SO(3) are given by [3]

$$U^l_{mn}(R(\alpha,\beta,\gamma)) = (-1)^{n-m} e^{-i(m\alpha+n\gamma)} P^l_{mn}(\cos\beta),$$

where α, β and γ are the Euler angles, and $P^l_{mn}(x)$ is the generalized associated Legendre functions. The integral form of $P^l_{mn}(x)$ is

$$P^l_{mn}(\cos\beta) = \frac{i^{n-m}}{2\pi}\left[\frac{(l-m)!(l+m)!}{(l-n)!(l+n)!}\right]^{1/2} \int_0^{2\pi} \left(\cos\frac{\beta}{2} e^{i\phi/2} + i\sin\frac{\beta}{2} e^{-i\phi/2}\right)^{l-n} \times \left(\cos\frac{\beta}{2} e^{-i\phi/2} + i\sin\frac{\beta}{2} e^{i\phi/2}\right)^{l+n} e^{im\phi} d\phi. \quad (16)$$

When α and γ are zero, we can have $U(\beta) \equiv U(R(0, \beta, 0))$.

Since U^l_{mn} is the IUR matrix of SO(3), it follows that $U^l(\theta_1 + \theta_2) = U^l(\theta_1)U^l(\theta_2)$. Furthermore, since U^l is unitary and $U^l(R(0, \beta, 0))$ is real, $U^l(\beta)$ is orthogonal. Thus there exist a skew-symmetric matrix, Γ^l such that $U^l(2\theta) = \exp(\Gamma^l\theta)$. In order to find Γ, we will compute $dU^l_{mn}(2\theta)/d\theta$ at $\theta = 0$. We consider $U^l_{mn}(2\theta)$ instead of $U^l_{mn}(\theta)$, because the former is easier to connect to $S^m(\theta)$.

The derivative at $\theta = 0$ is

$$\Gamma^l_{mn} = \frac{dU^l_{mn}(2\theta)}{d\theta}\Big|_{\theta=0} = (-1)^{n-m}\frac{dP^l_{mn}(\cos 2\theta)}{d\theta}\Big|_{\theta=0},$$

and

$$\frac{dP^l_{mn}(\cos 2\theta)}{d\theta}\Big|_{\theta=0} = \frac{i^{n-m}}{2\pi}\left[\frac{(l-m)!(l+m)!}{(l-n)!(l+n)!}\right]^{1/2} \times \frac{d}{d\theta}\left(\int_0^{2\pi}\left(\cos\theta e^{i\phi/2} + i\sin\theta e^{-i\phi/2}\right)^{l-n}\left(\cos\theta e^{-i\phi/2} + i\sin\theta e^{i\phi/2}\right)^{l+n} e^{im\phi}d\phi\right)\Big|_{\theta=0} = \sqrt{(l-n)(l+n+1)}\delta_{m,n+1} - \sqrt{(l+n)(l-n+1)}\delta_{m+1,n}$$

Therefore

$$\Gamma^l_{mn} = (-1)^{n-m}\frac{dP^l_{mn}(\cos 2\theta)}{d\theta}|_{\theta=0} = \sqrt{(l+n)(l-n+1)}\delta_{m+1,n} - \sqrt{(l-n)(l+n+1)}\delta_{m,n+1}.$$

If we let $m' = 2l$, $n' = m + l$ and $q' = n + l$, then

$$\Gamma^l_{mn} = \sqrt{(q')(m'-q'+1)}\delta_{n'+1,q'} - \sqrt{(m'-q')(q'+1)}\delta_{n',q'+1} = \Omega^{m'}_{n',q'},$$

where Ω is the skew-symmetric matrix defined in Section 3.3. Therefore, we have shown an analytic proof of the following:

$$U^l(2\theta) = \exp(\Gamma^l\theta) = \exp(\Omega^{m'}\theta) = S^{m'}(\theta)$$

and

$$S^{m'}_{n'q'}(\theta) = (-1)^{n-m}P^l_{mn}(\cos 2\theta) \tag{17}$$

where $m' = 2l$, $n' = m + l$ and $q' = n + l$.

It is worthwhile to consider the following identity for P^l_{mn} (from [18])

$$P^l_{MM'}(\cos\beta) = i^{M-M'}\sum_{M''=-l}^{l} P^l_{M''M}(\cos(\pi/2))e^{-iM''\beta}P^l_{M''M'}(\cos(\pi/2)). \tag{18}$$

Note that $P^l_{mn}(\cos\beta)$ is also denoted by $d^l_{mn}(\beta)$ in many literatures including [18]. Since $d^l_{mn}(\beta)$ is defined by the same form as (16), $P^l_{mn}(\cos\beta)$ is equivalent to $d^l_{mn}(\beta)$.

Combining (17) and (18), we have the matrix expression as,

$$S^m(\theta)=E\cdot(S^m(\pi/4))^T\cdot G(\theta)\cdot S^m(\pi/4)\cdot E^*, \tag{19}$$

where E = diagonal($[i^0\ i^1\ \cdots\ i^m]$), $G(\theta)$ = diagonal($[e^{im\theta}\ e^{i(m-2)\theta}\ \cdots\ e^{-im\theta}]$) and E^* is the conjugate transpose of E. This matrix multiplication gives an alternative way to compute $S^m(\theta)$.

4 Connection between 2D Band-limited Hermite Expansion with discrete images

Digital images are generally defined on a discrete grid of points, but the Hermite expansion that we are considering is a 2D continuous function. Therefore, we need to connect the discrete images and the Hermite expansion. Here we examine both directions: computing the Hermite expansion from a given discrete image and computing the discrete image from a given Hermite expansion.

A method to find the Hermite expansion from a discrete image has been reported previously [22]. In that method, a cost function that computes the difference between the discrete image and the sampled values from the Hermite expansion was defined. By minimizing the cost function with respect to the Hermite coefficients, we can fit the Hermite expansion to the given image. The 'Hermite-filtered' image was obtained by resampling the Hermite expansion on the original grid. However, the method is computationally sensitive and expensive, because it explicitly inverts a large matrix. We propose here a new method that shows better performance.

The new method uses a Fourier series to connect the discrete image and the 2D band-limited Hermite expansion. Since one can easily extend the 1D case to the 2D case, let us focus on the 1D case. Given sample values $f_n = f(2\pi n/N)$ for $n \in [0, N-1]$ we can construct exactly the continuous periodic function

$$f(\theta)=\sum_{k=-B}^{B}\tilde{f}_k e^{ik\theta}$$

on the continuous domain $[-\pi, \pi]$ that hits these sample points exactly. The set of coefficients $\{\tilde{f}_k\}$ can be obtained from $\{f_n\}$ by FFT. The Fourier series contains $2B + 1$ terms, and in the case of the FFT, the number of sampled data, N, is usually taken to be a power of 2. This discrepancy is rectified if the constraint $\tilde{f}_B = \tilde{f}_{-B}$ is imposed. Then there are $N = 2B$ free parameters in both [3]. The detailed derivation of $\{\tilde{f}_k\}$ from a discrete data set is given in the Appendix C. When implemented using the FFT this is performed in $O(N \log N)$ arithmetic operations. Now we need to connect the Fourier series and the Hermite expansion.

4.1 Hermite Expansion fit to Fourier Series

Once the discrete data has already been captured in a continuous Fourier series, we can find a Hermite series to best capture the same data without sampling. In other words, we can seek the Hermite coefficients $\{\hat{f}_k\}$ such that

$$C(\{\hat{f}_k\})=\int_{-\pi}^{\pi}\left|\sum_{k=-B}^{B}\tilde{f}_k e^{ik\theta}-\sum_{l=0}^{M}\hat{f}_l h_l(\theta/s)\right|^2 d\theta$$

is minimized. Here s is a scaling factor that we choose so as to capture the best resolution for a given Hermite band limit, M. In other words, there will be an optimal $s = s(M)$ for a given image.

Expanding the above cost function out, we find that

$$C=\sum_{l_1=0}^{M}\sum_{l_2=0}^{M}\widehat{f}_{l_1}\widehat{f}_{l_2}a_{l_1,l_2} - \sum_{l=0}^{M}\sum_{k=-B}^{B}\widehat{f}_l(\tilde{f}_k b_{k,l}+\overline{\tilde{f}_k}b_{-k,l})+\|f\|^2$$

where

$$a_{l_1,l_2}=\int_{-\pi}^{\pi}h_{l_1}(\theta/s)h_{l_2}(\theta/s)d\theta,$$
$$b_{k,l}=\int_{-\pi}^{\pi}h_l(\theta/s)e^{ik\theta}d\theta,$$
$$\|f\|^2=\int_{-\pi}^{\pi}|f(\theta)|^2d\theta,$$

and $\bar{c}$ is the complex conjugate of c. Now, a simple approximation can compute them accurately, assuming a sufficient zero padding of the image and the optimal scaling factor, s.

Namely, if we let $x = \theta/s$, then for functions that decay to zero sufficiently rapidly,

$$\int_{-\pi}^{\pi}F(\theta/s)d\theta=s\int_{-\pi/s}^{\pi/s}F(x)dx \approx s\int_{-\infty}^{\infty}F(x)dx. \tag{20}$$

This means that, due to the orthonormality of Hermite functions over the real line,

$$a_{l_1,l_2} \approx s\delta_{l_1,l_2}$$

and

$$b_{k,l} \approx s\int_{-\infty}^{\infty}h_l(x)e^{iksx}dx = \sqrt{2\pi}i^l s h_l(ks). \tag{21}$$

The equality in (21) is due to the fact that Hermite functions are eigenfunctions of the Fourier transform:

$$\int_{-\infty}^{\infty}e^{i\omega x}h_n(x)dx=\sqrt{2\pi}i^n h_n(\omega).$$

Similarly,

$$b_{-k,l} \approx \sqrt{2\pi} i^l s h_l(- ks) = \sqrt{2\pi} (i)^l s h_l(ks)(-1)^l = (-1)^l b_{k,l},$$

since $h_n(-x) = (-1)^n h_n(x)$.

We can rewrite the cost function as

$$C = s \sum_{l=0}^{M} \left(\widehat{f}_l^2 - \sqrt{2\pi} i^l \sum_{k=-B}^{B} (\tilde{f}_k + \overline{\tilde{f}_k}(-1)^l) h_l(ks) \widehat{f}_l \right) + \|f\|^2.$$

Therefore, the Hermite coefficients for the Hermite series that approximates this with minimal mean-squared error will be

$$\widehat{f}_l \approx \sqrt{\frac{\pi}{2}} i^l \sum_{k=-B}^{B} (\tilde{f}_k + \overline{\tilde{f}_k}(-1)^l) h_l(ks). \tag{22}$$

For an appropriate choice of s, and M we should expect that this Hermite series will drive the RMS error to zero. And the values at the sample points should also converge to the original specified values. The most critical factor is that the scale and zero padding is chosen so that (20) is an accurate statement for the integrals that were approximated for $a_{l1,l2}$ and b_{kl}.

4.2 Fourier Series Fit to Hermite Expansion

In the previous subsections we obtained the Fourier series on a continuous domain from a discrete data set and computed the Hermite expansion fit to that Fourier series. One might expect that sampling the Hermite expansion back onto the original data grid would give filtered data close to the original data. However, since the Hermite expansion may have a high band limit relative to the Fourier series, sampling it directly can cause aliasing. It is therefore more consistent to use the Fourier series again as an intermediate between the Hermite expansion and the resulting sampled values. In this way, the Fourier series connects the Hermite expansion and the discrete data in both directions.

Now we will try to find the Fourier series that best fits to a band-limited Hermite expansion. The cost function is the same form as in the previous subsection except that the argument is now the Fourier series coefficients rather than the Hermite coefficients, which are the inputs in this context. The cost function is

$$C\prime(\{\tilde{f}_l\}) = \int_{-\pi}^{\pi} \left| \sum_{k=-B}^{B} \tilde{f}_k e^{ik\theta} - \sum_{l=0}^{M} \widehat{f}_l h_l(\theta/s) \right|^2 d\theta = 2\pi \sum_{k=-B}^{B} \left(\tilde{f}_k \overline{\tilde{f}_k} - \frac{s}{\sqrt{2\pi}} \sum_{l=0}^{M} (\tilde{f}_k i^l + \overline{\tilde{f}_k}(-i)^l) h_l(ks) \widehat{f}_l \right) + s\|f\|^2,$$

which can be rewritten as

$$C_I = 2\pi \sum_{k=-B}^{B} \left| \tilde{f}_k - \frac{s}{\sqrt{2\pi}} \sum_{l=0}^{M} (-i)^l h_l(ks) \widehat{f_l} \right|^2 + \text{constant}.$$

Therefore, the Fourier series coefficients that minimize the cost function will be

$$\tilde{f}_k = \frac{s}{\sqrt{2\pi}} \sum_{l=0}^{M} (-i)^l h_l(ks) \widehat{f_l}.$$

4.3 Determination of Scaling Factor

From (20) and the definition of $a_{l1,l2}$ and $b_{k,l}$, it is clear that the range of the integral, $(-\pi/s, \pi/s)$ should be chosen to capture enough of the Hermite function. Let us consider the following:

$$K_n(s) = \int_{-\pi/s}^{\pi/s} (h_n(x))^2 dx.$$

$K_n(s)$ will monotonically increase to 1 as s decreases due to the orthonormality of the Hermite functions. If $K_n(s)$ is close enough to 1, one can conclude that the range of the integral, $(-\pi/s, \pi/s)$ captures enough of the Hermite function.

The contour plot of $K_n(s)$ as a function of n and s is shown in Fig. 1. For example, if the band limit of the Hermite expansion is 450, a value of 0.1 or less is good for s, since the corresponding range guarantees $K_n(0.1)$ is close enough to 1 for $n = 0, 1, \ldots, 450$. While a smaller value of s looks better in this context, too small a value of s may cause a problem when we find the Hermite expansion fit to a Fourier series. If s is very small, the Hermite functions can be meaningful only in the very small part (near the origin) of the range, $(-\pi/s, \pi/s)$. This means that the Hermite expansion can describe the reference Fourier series near the origin only In order to avoid this, we need to increase the value of s. Thus, we need to find a balanced value for s based on these two criteria.

When the image shown in Fig. 2(a) is tested, Fig. 3(a) shows the normalized least-squared error (NLSE) between the original image and Hermite-filtered image for various s values with several band limits of the Hermite expansion, The NLSE of the two images, $R(m, n)$ and $I(m, n)$ is defined as [11]

$$\text{NLSE} = \sqrt{\frac{\sum_{m=1}^{N} \sum_{n=1}^{N} [R(m,n) - I(m,n)]^2}{\sum_{m=1}^{N} \sum_{n=1}^{N} [R(m,n)]^2}}.$$

We prepared a 512×512 image shown in Fig.2, where a 256×256 initial image is surrounded by zero-padding. The best value for the scaling factor in this Lena image can be determined as $s = 0.085$ based on Fig. 3(a). Fig. 4(a) shows the inverted amplified difference image between the original and the Hermite-filtered Lena images when the band limit is $N = 620$ and the scaling factor is $s = 0.085$. If we apply the same process to the baboon image as shown in Fig. 2(b),

we have the different error curve shown in Fig. 3(b) and $s = 0.08$ is the best value for the scaling factor in this case and the corresponding inverted difference image is shown in Fig. 4(b). For better visualization, we presented the inverted difference images instead of the regular difference images that contain black pixels in most area.

Although the choice of the optimal scaling factor is dependent on the original image of interest, the image error is not sensitive to the scaling factor around the optimal one. Thus, one can determine the 'quasi-optimal' value for the scaling factor independent of images for a given image size. Fig. 6 shows the errors between the original and the Hermite-filtered images as a function of the scaling factor, when various test images (512 × 512) shown in Fig. 5 are used. As seen in Fig. 6, the scaling factor $s = 0.085$ is a reasonable choice for all the six test images. Generalizing this result, we choose the scaling factor $s = 0.085$ as a quasi-optimal value for the 512 × 512 images. We can simply apply this to another sizes of images.

5 Rotation of Images and Accuracy Test

5.1 Computation of $S^m(\theta)$

Once we have the Hermite coefficients from a discrete image, the rotation can be applied to the coefficients using (8). In order to compute $S^m(\theta)$, we may use the recurrence formulas, (12) and (13), or the exponential mapping, (15), or the matrix multiplication, (19). Since the matrix exponential of large matrices increases the computational time, we pursue two methods; (1) the recurrence formulas; and (2) the matrix multiplication. For convenience, we will denote the former method by 'Method A' and the latter method by 'Method B'.

We will consider only rotations between 0 and $\pi/4$ radians. Since $S^m(-\theta) = (S^m(\theta))^T$ and any rotation angle can be accomplished by a combination of rotation by $k \times \pi/2$ and rotation by θ, where k is an integer and $-\pi/4 \le \theta \le \pi/4$. Note that rotation of discrete images by $k \times \pi/2$ can be perfectly obtained by reassigning the pixel values to a square grid.

Since, by the discussion in Section 3.3, in exact arithmetic $S^m(\theta)$ is an orthogonal matrix, we use the following as a measure of a degree of accuracy in its computation:

$$K(m,\theta) = \|S^m(\theta)S^m(\theta)^T - I_m\| / \|I_m\|,$$

where I_m is the $(m + 1) \times (m + 1)$ identity matrix. Fig. 7(a) shows the contour plot of $K(m, \theta)$, when we apply the first recurrence formula in (12). If we average all 4 recurrence formulas in (12) and (13), $K(m, \theta)$ is improved as shown in Fig. 7(b). However, it still shows the instability with higher orders, when the rotation angle is above some threshold. This can be overcome by using the multiplicative property of $S^m(\theta)$, which is $S^m(\theta_1 + \theta_2) = S^m(\theta_1)S^m(\theta_2)$. For example, $S^m(\pi/4)$ can be stably obtained by the product $S^m(\pi/8) \times S^m(\pi/8)$. Consequently, we compute $S^m(\theta)$ by $(S^m(\theta/2))^2$, when $\theta > \pi/8$. We should implement this by $\mathrm{f}_m^{\theta} = S^m(\theta/2) \times (S^m(\theta/2) \times \mathrm{f}_m)$, instead of direct multiplication of the two 2D matrices, in order to avoid increasing the computational complexity. Using this method, we can obtain the rotated images as shown in Fig. 8 for the original images shown in Fig. 2.

Alternatively, we can compute $S^m(\theta)$ by (19), which is denoted by Method B. Since the matrices, E and $S^m(\pi/4)$ in (19) are constant and $G(\theta)$ requires only scalar exponential mapping, this method can be implemented without unstable calculation. One thing that we should be careful of is handling $S^m(\pi/4)$. We can compute this, store it into a computer memory (or storage device) and then use it, when we rotate images. Since computation of $S^m(\pi/4)$ is performed

once before rotation, we can sacrifice the computation time to have good accuracy. The matrix exponential mapping by Padé algorithm [15] can be used for this end. Explicitly, we use

$$S^m(\pi/4)=\exp(\Omega^m \times \frac{\pi}{4}),$$

where Ω^m was defined in Section 3.3. Practically, all $S^m(\pi/4)$ for $m = 0, 1, \cdots, N$ cannot be stored in the memory of current PCs, when the bandlimit, N exceeds 400. We stored them to disk and used it to compute $S^m(\theta)$. Using this method, we can obtain the rotated images as shown in Fig. 9 for the original images shown in Fig. 2.

5.2 Accuracy Test

Now let us consider the accuracy of our methods. If the original image were defined as a continuous 2D function, its rotated version would also be a continuous function. This rotated continuous image would be a 'perfect' answer with which to assess the accuracy of our image rotation method. However, this is not the case. Since the original image is defined only on a discrete grid and the result is on a different discrete grid, there is no 'perfect' baseline to compare against for image rotations except for special rotation angles ($\theta = 0, \pi/2, \pi\ldots$). Therefore, as an alternative, we perform the following test:

Test 1 We fit the rotated image to an appropriate continuous 2D function and resample it onto the original grid to have an image close to the original image. We compare the resulting image and the original image.

Test 1 assesses how much information of the original image remains after the rotation. In order to compare the original images and the resulting images in accuracy tests, we will use the NLSE, Sobolev norm and relative entropy.

The Sobolev norm which is defined as [17,28]

$$Sob(f,g)=\left(\frac{1}{(n(U))^2}\sum_{u\in U}(1+|\eta_u|^2)^{\delta}|F(u)-G(u)|^2\right)^{1/2},$$

where f and g are image functions, and F and G are the discrete Fourier transforms of f and g, respectively. η_u is the two dimensional frequency vector associated with u. U is the frequency domain which is a lattice of the same dimensions as the domain of the image functions. $n(U)$ is the number of pixels of the lattice. We will use $\delta = 1/2$ in this paper. The Sobolev norm includes the difference between two images in terms of the derivatives.

Relative entropy (also known as Kullback-Leibler distance) is a non-commutative measure of the distance between two probability distributions [7]. It can also be used for measuring the difference between images. It is defined as [1]

$$D(p\|\widehat{p})=\sum_{\mathbf{x}}p(\mathbf{x})\log\frac{p(\mathbf{x})}{\widehat{p}(\mathbf{x})},$$

where $p(\mathbf{x})$ and $\hat{p}(\mathbf{x})$ are two image functions. In this paper, we will let $\hat{p}(\mathbf{x})$ be the original image and let $p(\mathbf{x})$ be the resulting image in the accuracy tests.

The NLSEs, the Sobolev norms, and the relative entropies between the original image and the resampled image by Test 1 are shown in Table 1 and 2. The 2D Fourier series and bi-cubic interpolation were used in Table 1 and 2, respectively. We report the results by the two methods (Method A and Method B) for computing $S^m(\theta)$. We also report the test results of the FFT-based image rotation method developed by Larkin et al. [8] and Owen et al. [19] by applying their method to our example images. As mentioned earlier, in this FFT-based method, three consecutive shears are implemented using FFT. Since Test 1 is dependent on the choice of the reference continuous 2D function, this test is not a sufficient test for accuracy, even though our methods are better in this test.

As briefly mentioned in the previous subsection, rotations by a multiple of $\pi/2$(rad) can be used as a perfect baseline for comparison, since those rotations of a discrete image can be exactly obtained by reassigning the pixel values to a rectangular grid. The first natural test using this perfect baseline would be that we rotate an image by $\pi/2$(rad) using an image rotation algorithm and compare the result to the perfect answer. However, this test does not reflect the accuracy of the FFT-based methods, because $\pi/2$(rad) rotation by three shear processes is equivalent to reassignment of the pixel values to the $\pi/2$(rad) rotated grid. Specifically, the amount of translation of pixels in shear processes is a multiple of the pixel size, when the rotation angle is $\pi/2$. This means that we can not assess the effect of interpolation of the FFT-based method in this particular case. As an indirect way to assess the accuracy, we run the following tests:

Test 2 We rotate an image by $\pi/4$(rad) 2 times consecutively and compare the resulting image and the image rotated perfectly by $\pi/2$(rad).

Test 3 We rotate an image by $\pi/6$(rad) 12 times consecutively and compare the resulting image and the original image.

Test 4 We rotate an image by 9 random angles consecutively and then rotate by the negative of the sum of the angles. The first 9 angles are sampled from a uniform distribution on the interval $[-\pi/4, \pi/4]$ (rad). We compare the resulting image and the original image.

These three tests are based on the fact that if an image rotation method is accurate, it will preserve the information of the original image over the consecutive rotations and will give the resulting image close to the perfect answer.

Since these three tests requires consecutive rotations, we need to define a concept of the multiple rotations. Strategy 1 shown in Fig. 10(a) fully exploits the steerability of the Hermite functions. First we obtain the Hermite expansion corresponding to a discrete image via the Fourier series. Then we perform the rotation process on the Hermite coefficients. In order to display the resulting image, one should find the Fourier series fit to the Hermite expansion and sample it. In this strategy, we retain the Hermite coefficients for the consecutive rotations instead of the resulting image. Therefore, the loss of information occurs only once when we fit the Hermite expansion to a discrete data. However, it is more natural to keep the rotated 'image' for subsequent rotations. Therefore we also suggest Strategy 2 as shown in Fig. 10(b). After we apply the rotation process to the Hermite coefficients computed from a given image, we compute the rotated discrete image by computing the Fourier series fit to the new (rotated) Hermite expansion and sampling the Fourier series. In order to rotate it again, we apply the same process to the rotated image. In the numerical test, we will use only Strategy 2, since it is closer to the real situation of image process.

The test results for Test 2, 3 and 4 are shown in Table 3, 4 and 5. We report the accuracy of our method when using Method A and Method B. We also report the test results of the FFT-based method. The band limit of the Hermite expansion is fixed to $N = 620$. The test results with the three measurement methods show our new methods have lower error than the FFT-

based method. Even though we do not pursue Strategy 1, the NLSEs in Test 2, 3 and 4 with Lena image are 0.0017, when we use Strategy 1. This small error is possible, since the error in our method occurs only once when the Hermite filtering is performed.

Fig. 11 shows the visual quality of the FFT-based method and our method. In this figure, the difference images between the original Lena image and the resulting image in Test 3 are shown. With the Hermite-based method, the difference occurs mainly at the sharp edges in the image. However, with the FFT-based method, the difference can be seen in the wide area around the sharp edges.

5.3 Computation Time

Table 6 shows the elapsed time in seconds for the image rotation process, when various sizes of images are rotated by $\pi/4$. The first and second rows show the computation time for Hermite transform proposed in Section 4.1 and 4.2, respectively. The total elapsed time for Method A will be the summation of the first three rows. Similarly the total elapsed time for Method B is the summation of the first, second, fourth and fifth rows. Ideally the time for loading $S^m(\pi/4)$ can be excluded from the total elapsed time, since the time for loading the data can be avoided if the data is built in a memory. In this aspect, Method B shows better performance than Method A.

The algorithm for computing the Hermite coefficients from a discrete image is implemented with $\mathcal{O}(N^3)$ computation, when the size of image is $\mathcal{O}(N^2)$ and the band limit of the Hermite expansion is $\mathcal{O}(N)$. It is achieved by expressing (22) as a matrix multiplication. In principle, we could reduce the computation cost to $\mathcal{O}(N^2 \log^2 N)$ if we used the algorithm in [9,10]. However, the rotation algorithm on the Hermite expansion in Section 3 is the dominant contribution to the overall computation time. In Method A, the computation of $S^m(\theta)$ needs $\mathcal{O}(m^2)$ computations, since each matrix has the size of $(m + 1) \times (m + 1)$ and all elements are computed by the recurrence formulas. Because $S^m(\theta)$ is multiplied by a $(m + 1)$-dimensional vector for $m = 1, 2, \cdots, N$, the complexity of the whole rotation process is $\mathcal{O}(N^3)$. Method B using (19) also has $\mathcal{O}(N^3)$ complexity, since computation of $S^m(\theta)$ using (19) needs $\mathcal{O}(m^2)$ calculation.

6 Conclusions

In this work, in order to rotate images, we used the fact that the rotation of a 2D band-limited Hermite expansion (with a band limit of a special form) is of the same form and has the same band limit as the original expansion. We observed that the rotation of a two-dimensional band-limited Hermite expansion results in a linear operation on the Hermite coefficients. We proposed two ways to compute the matrix representing the linear operation; (1) recurrence formulas; and (2) matrix multiplication formula. In addition, we proposed a method of connecting the band-limited Hermite expansion and the discrete images. We used the Fourier series on a continuous domain for the connection. Combining these techniques, we suggested the image rotation method and the example results were presented. We also designed the tests to assess the accuracy of rotation methods. We showed the accuracy of our image rotation methods is better than that of the existing image rotation technique using FFT and 3 shears.

Reducing computation time of our methods remains ongoing work. The FFT-based method works in less than one second with the images used in Table 6. Nevertheless, it is important to note that our methods are based on a direct physical rotation on the plane, while the FFT-based method uses the three shears. This discrepancy is thought to bring the benefit in accuracy to our methods.

Acknowledgments

This work was performed under the support of grants: NIH R01GM075310 “Group-Theoretic Methods in Protein Structure Determination,” and NIH Grant R01EB006435 “Steering Flexible Needles in Soft Tissue.”

Appendix A

Orthogonality of $S^m(\theta)$

Let use consider the following:

$$E_m = \int_{\mathbb{R}^2} f_m(x,y)\overline{g_m(x,y)}dxdy,$$

where $f_m(x,y) = \sum_{n=0}^{m} \widehat{f}_{n,m-n} h_n(x) h_{m-n}(y)$, $g_m(x,y) = \sum_{n=0}^{m} \widehat{g}_{n,m-n} h_n(x) h_{m-n}(y)$, and $\bar{c}$ is the complex conjugate of c. Using the orthonormality of the Hermite functions, we can rewrite this as

$$E_m = \sum_{n=0}^{m} \widehat{f}_{n,m-n} \widehat{g}_{n,m-n}, \tag{23}$$

since we are considering real-valued functions, and $\hat{f}_{n,m-n}$ and $\hat{g}_{n,m-n}$ are real-valued scalars.

Since E_m is invariant under rotation of $f_m(x, y)g_m(x, y)$, we have

$$E_m = \int_{\mathbb{R}^2} \left(\sum_{n=0}^{m} \widehat{f}^{\theta}_{n,m-n} h_n(x) h_{m-n}(y) \right) \left(\sum_{k=0}^{m} \widehat{g}^{\theta}_{k,m-k} h_k(x) h_{m-k}(y) \right) dxdy = \sum_{n=0}^{m} \widehat{f}^{\theta}_{n,m-n} \widehat{g}^{\theta}_{n,m-n}.$$

Using (7), we can rewrite this as

$$E_m = \sum_{n=0}^{m} \left(\sum_{p=0}^{m} S^m_{n,p}(\theta) \widehat{f}_{p,m-p} \sum_{q=0}^{m} S^m_{n,q}(\theta) \widehat{g}_{q,m-q} \right).$$

Therefore,

$$E_m = \sum_{p=0}^{m} \sum_{q=0}^{m} \left(\sum_{n=0}^{m} S^m_{n,p}(\theta) S^m_{n,q}(\theta) \right) \widehat{f}_{p,m-p} \widehat{g}_{q,m-q}. \tag{24}$$

Equating (23) and (24), we have

$$\sum_{n=0}^{m} \widehat{f}_{n,m-n}\, \widehat{g}_{n,m-n} = \sum_{p=0}^{m}\sum_{q=0}^{m} \widehat{f}_{p,m-p} \left(\sum_{n=0}^{m} S_{n,p}^{m}(\theta)\, S_{n,q}^{m}(\theta) \right) \widehat{g}_{q,m-q}$$

for all possible $\hat{f}_{n,m-n}$ and $\hat{g}_{n,m-n}$. We can conclude that

$$\sum_{n=0}^{m} S_{n,p}^{m}(\theta)\, S_{n,q}^{m}(\theta) = \delta_{p,q},$$

or

$$\{S^{m}(\theta)\}^{T} S^{m}(\theta) = I_{m}, \tag{25}$$

where I_m is the $(m + 1) \times (m + 1)$ identity matrix. Since $S^m(\theta)$ is a real-valued square matrix and (25) holds, $S^m(\theta)$ is an orthogonal matrix

Appendix B

The Skew-symmetric Matrix log ($S^m(\theta)$)

From the definition of $S_{q,n}^{m}(\theta)$,

$$\begin{aligned}\frac{d}{d\theta}\left(S_{q,n}^{m}(\theta)\right) = \int_{\mathbb{R}^2} h_q(x)\, h_{m-q}(y) \left(\frac{dh_n(t)}{dt}\Big|_{t=x\cos\theta+y\sin\theta} \right) \frac{d(x\cos\theta+y\sin\theta)}{d\theta} \\ \times h_{m-n}(-x\sin\theta+y\cos\theta)\, dxdy + \int\int h_q(x)\, h_{m-q}(y)\, h_n(x\cos\theta+y\sin\theta) \\ \times \left(\frac{dh_{m-n}(t)}{dt}\Big|_{t=-x\sin\theta+y\cos\theta} \right) \frac{d(-x\sin\theta+y\cos\theta)}{d\theta}\, dxdy.\end{aligned}$$

If we define $A(\theta)$ and $B(\theta)$ as

$$\begin{aligned}A(\theta) &= \int_{\mathbb{R}^2} h_q(x)\, h_{m-q}(y) \left(\frac{dh_n(t)}{dt}\Big|_{t=x\cos\theta+y\sin\theta} \right) \frac{d(x\cos\theta+y\sin\theta)}{d\theta}\, h_{m-n}(-x\sin\theta+y\cos\theta)\, dxdy, \\ B(\theta) &= \int_{\mathbb{R}^2} h_q(x)\, h_{m-q}(y)\, h_n(x\cos\theta+y\sin\theta) \left(\frac{dh_{m-n}(t)}{dt}\Big|_{t=-x\sin\theta+y\cos\theta} \right) \frac{d(-x\sin\theta+y\cos\theta)}{d\theta}\, dxdy\end{aligned}$$

then $\Omega^m = A(0) + B(0)$.

The derivative of Hermite function can be computed as [3]

$$\frac{dh_n(x)}{dx} = \frac{1}{\sqrt{2}} \left(\sqrt{n} h_{n-1}(x) - \sqrt{n+1} h_{n+1}(x) \right).$$

Therefore,

$$\frac{dh_n(t)}{dt}\Big|_{t=x\cos\theta+y\sin\theta} = \frac{1}{\sqrt{2}}\left(\sqrt{n}h_{n-1}(x\cos\theta+y\sin\theta) - \sqrt{n+1}h_{n+1}(x\cos\theta+y\sin\theta)\right).$$

On the other hand,

$$\frac{d(x\cos\theta+y\sin\theta)}{d\theta} h_{m-n}(-x\sin\theta+y\cos\theta) = (-x\sin\theta+y\cos\theta)\, h_{m-n}(-x\sin\theta+y\cos\theta)$$
$$= \frac{1}{\sqrt{2}}\left(\sqrt{m-n+1}h_{m-n+1}(-x\sin\theta+y\cos\theta) + \sqrt{m-n}h_{m-n-1}(-x\sin\theta+y\cos\theta)\right)$$

Using these identities and (11), we have

$$A_{q,n}(0) = B_{q,n}(0) = \frac{1}{2}\sqrt{n}\sqrt{m-n+1}\delta_{q,n-1} - \frac{1}{2}\sqrt{n+1}\sqrt{m-n}\delta_{q,n+1}$$

Therefore,

$$\Omega^m_{q,n} = \frac{d}{d\theta}\left(S^m_{q,n}(\theta)\right)\Big|_{\theta=0} = A_{q,n}(0) + B_{q,n}(0) = \sqrt{n}\sqrt{m-n+1}\delta_{q,n-1} - \sqrt{n+1}\sqrt{m-n}\delta_{q,n+1}.$$

The matrix, Ω^m is a sparse skew-symmetric matrix.

Appendix C

Connection between Fourier Series and DFT

Here we find the Fourier series exactly hitting the given discrete points. For convenience, we assume that the number of data, N, is even. For an odd number of data points, we can add zero-padding to obtain an even number of data points.

For given N equally-spaced sampled points on the unit circle, we have to find the corresponding Fourier series such that

$$f(\theta_j) = \sum_{k=-B}^{B} \tilde{f}_k e^{ik\theta_j}$$

where $f(\theta_j)$ is the given sampled points and $\theta_j = 2\pi j/N$, $j = 0, 1, \ldots (N-1)$. We can expand $f(\theta_j)$ as

$$f(\theta_j)=\tilde{f}_0 e^{i\cdot 0\cdot 2\pi j/N}+\tilde{f}_1 e^{i\cdot 1\cdot 2\pi j/N}+\ldots+\tilde{f}_B e^{i\cdot B\cdot 2\pi j/N} + \tilde{f}_{-1} e^{i\cdot -1\cdot 2\pi j/N}+\ldots+\tilde{f}_{-B} e^{i\cdot (-B)\cdot 2\pi j/N} \quad (26)$$

On the other hand, we can apply the DFT(discrete Fourier transform) to the N samples and have

$$f(\theta_j)=f_j=\frac{1}{N}\sum_{n=0}^{N-1}\hat{f}_n e^{\frac{2\pi}{N}ijn},$$

where $\hat{f}_n=\sum_{j=0}^{N-1} f_j e^{-\frac{2\pi}{N}ijn}$. Since we assume N is even, we can have

$$Nf(\theta_j)=\hat{f}_0 e^{i\cdot 0\cdot 2\pi j/N}+\hat{f}_1 e^{i\cdot 1\cdot 2\pi j/N}+\ldots+\hat{f}_{(\frac{N}{2}-1)} e^{i\cdot (N/2-1)\cdot 2\pi j/N}+\hat{f}_{N/2} e^{i\cdot (N/2)\cdot 2\pi j/N} +\hat{f}_{(\frac{N}{2}+1)} e^{i\cdot (N/2+1)\cdot 2\pi j/N}+\ldots+\hat{f}_{(N-1)} e^{i\cdot (N-1)\cdot 2\pi j/N}. \quad (27)$$

Since $e^{i\cdot(N/2)\cdot 2\pi j/N} = e^{i\pi j} = e^{-i\cdot(N/2)\cdot 2\pi j/N}$ and

$$\hat{f}_{(\frac{N}{2}+1)} e^{i\cdot (N/2+1)\cdot 2\pi j/N}+\ldots+\hat{f}_{(N-1)} e^{i\cdot (N-1)\cdot 2\pi j/N} =\hat{f}_{(\frac{N}{2}+1)} e^{i\cdot -(N/2-1)\cdot 2\pi j/N}+\ldots+\hat{f}_{(N-1)} e^{i\cdot (-1)\cdot 2\pi j/N},$$

we can rewrite (27) as

$$f(\theta_j)=\frac{1}{N}\left(\hat{f}_0 e^{i\cdot 0\cdot 2\pi j/N}+\hat{f}_1 e^{i\cdot 1\cdot 2\pi j/N}+\ldots+\hat{f}_{(\frac{N}{2}-1)} e^{i\cdot (N/2-1)\cdot 2\pi j/N}+\frac{1}{2}\hat{f}_{N/2} e^{i\cdot (N/2)\cdot 2\pi j/N} +\hat{f}_{(N-1)} e^{i\cdot (-1)\cdot 2\pi j/N}+\ldots+\hat{f}_{(\frac{N}{2}+1)} e^{i\cdot -(N/2-1)\cdot 2\pi j/N}+\frac{1}{2}\hat{f}_{N/2} e^{i\cdot -(N/2)\cdot 2\pi j/N}\right). \quad (28)$$

Equating (26) and (28), we can have the relationship between the DFT and the Fourier series as follows:

$$\tilde{f}_j=\begin{cases} & B=N/2, \\ \frac{1}{N}\hat{f}_j & j=0,1,\ldots,N/2-1 \\ \frac{1}{N}\hat{f}_{N+j} & j=-1,-2\ldots,-(N/2-1) \\ \frac{f_{N/2}}{2N} & j=N/2 \\ \frac{f_{N/2}}{2N} & j=-N/2 \end{cases},$$

for an even number, N.

References

1. Avcıbas İ, Sankur B, Sayood K. Statistical evaluation of image quality measures. Journal of Electronic Imaging 2002;11(2):206–223.
2. Chen B, Kaufman A. 3D volume rotation using shear transformations. Graphical Models 2000;62(4): 308–322.
3. Chirikjian, GS.; Kyatkin, AB. Engineering Applications of Noncommutative Harmonic Analysis. CRC Press; Boca Raton, FL: 2001.
4. Cox RW, Tong R. Two- and three-dimensional image rotation using the FFT. IEEE Transactions on Image Processing 1999;8(9):1297–1299. [PubMed: 18267547]
5. Freeman WT, Adelson EH. The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 1991;13(9):891–906.
6. Jacob M, Unser M. Design of steerable filters for feature detection using canny-like criteria. IEEE Transactions on Pattern Analysis and Machine Intelligence 2004;26(8):1007–1019. [PubMed: 15641731]
7. Kullback S, Leibler RA. On Information and Sufficiency. Annals of Mathematical Statistics 1951;22 (1):79–86.
8. Larkin KG, Oldfield MA, Klemm H. Fast Fourier method for the accurate rotation of sampled images. Optics Commuications 1997;139:99–106.
9. Leibon, G.; Rockmore, D.; Chirikjian, GS. A fast Hermite transform with applications to protein structure determination. SNC '07: Proceedings of the 2007 international workshop on Symbolic-numeric computation; 2007. p. 117-124.
10. Leibon G, Rockmore D, Park W, Taintor R, Chirikjian GS. A fast Hermite transform. Theoretical Computer Science 2008;409:211–228. [PubMed: 20027202]
11. Liu Z, Laganiere R. Phase congruence measurement for image similarity assessment. Pattern Recognition Letter 2007;28:166–172.
12. Martens JB. Local orientation analysis in images by means of the Hermite transform. IEEE Transactions on Image Processing 1997;6(8):1103–1116. [PubMed: 18283000]
13. Martens JB. The Hermite transform: A survey. EURASIP journal on applied signal processing 2006;2006:1–20. [PubMed: 16758000]
14. Michaelis M, Sommer G. A Lie group approach to steerable filters. Pattern Recognition Letters 1995;16(11):1165–1174.
15. Moler C, Van Loan C. Nineteen dubious ways to compute the exponential of a matrix, twenty-five Years late. SIAM Review 2003;45:3–49.
16. Najafi, M.; Krylov, A.; Kortchagine, D. Image deblocking with 2D Hermite transform. Proceedings,International Conference Graphicon; 2003. p. 180-183.
17. Natterer, F. The Mathematics of Computerized Tomography. Wiley; New York: 1986.
18. Nikiforov, AF.; Suslov, SK.; Uvarov, VB. Springer Series in Computational Physics. Springer-Verlag; Berlin: 1991. Classical orthogonal polynomials of a discrete Variable.
19. Owen, CB.; Makedon, F. High quality alias free image rotation. Proceedings of 30th Asilomar Conference on Signals, Systems, and Computers Pacific Grove; 1996. p. 115-119.
20. Paeth A. A fast algorithm for general raster rotation. Proceedings,Graphics Interface 1986:77–81.
21. Park, W. Ph D dissertation. Johns Hopkins Univeristy; 2008. Inverse problems in structural biology and flexible needle steering.
22. Park W, Chirikjian GS. Interconversion between truncated Cartesian and polar expansions of images. IEEE, Transactions on Image Processing 2007;16(8):1946–1955. [PubMed: 17688200]
23. Toffoli T, Quick J. Three-dimensional rotations by three shears. Graphical Models and Image Processing 1997;59(2):89–95.
24. Unser M, Thevenaz P, Yaroslavsky LP. Convolution-based interpolation for fast, high-quality rotation of images. IEEE Transactions on Image Processing 1995;4(10):1371–1381. [PubMed: 18291969]
25. van Dijk, AM.; Martens, JB. Feature-based image compression with steered Hermite transforms. Proceedings, IEEE International Conference on Image Processing; 1996. p. 205-208.

26. van Dijk AM, Martens JB. Image representation and compression with steered Hermite transforms. Signal Processing 1996;56:1–16.

27. Vilenkin, NJ.; Klimyk, AU. Representation of Lie Groups and Special Functions. Vol. 1-3. Kluwer Academic Publ.; Dordrecht, Holland: 1991.

28. Wilson DL, Baddeley AJ, Owens RA. A New Metric for Grey-Scale Image Comparison. International Journal of Computer Vision 1997;24(1):5–17.

29. Yu W, Daniilidis K, Sommer G. Approximate Orientation Steerability Based on Angular Gaussians. IEEE, Transactions on Image Processing 2001;10(2):193–205. [PubMed: 18249611]

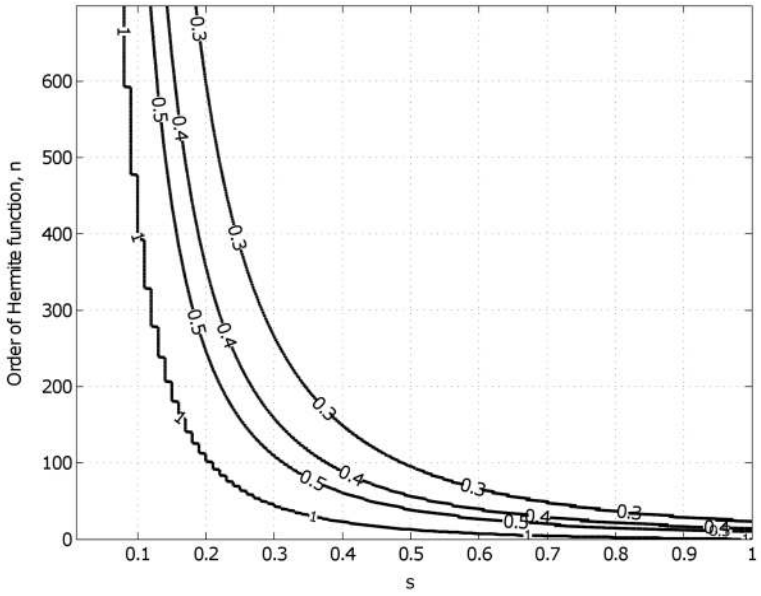
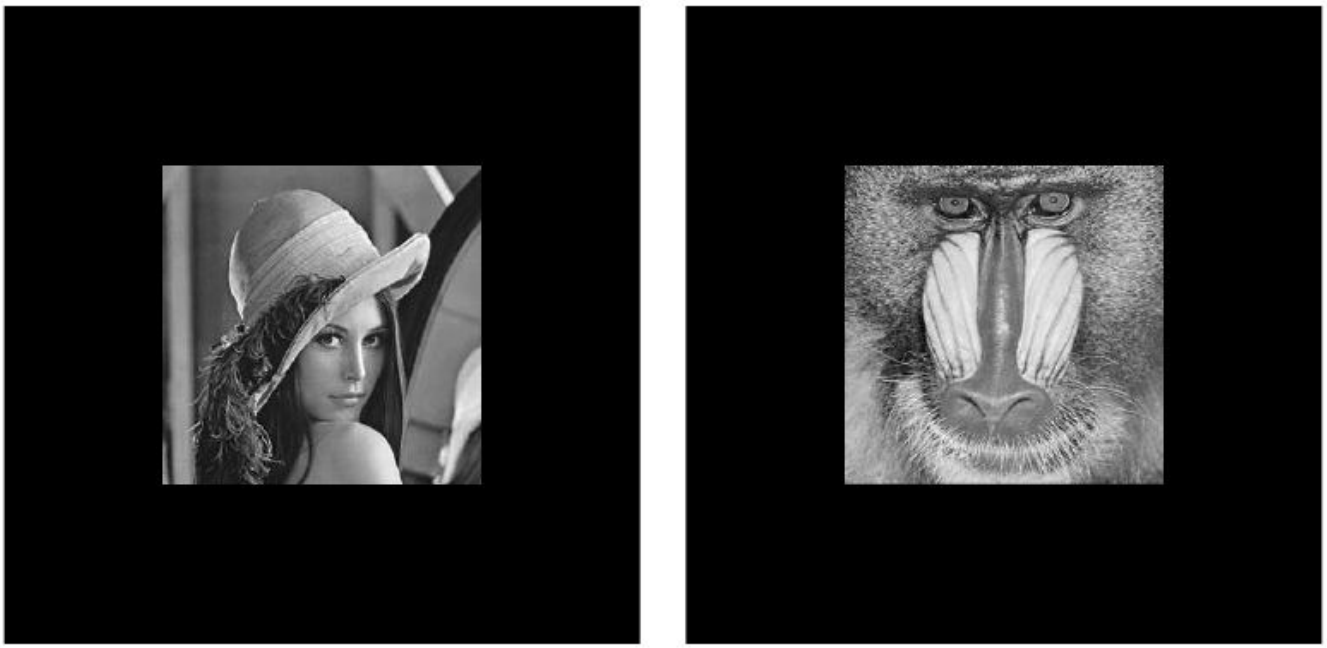
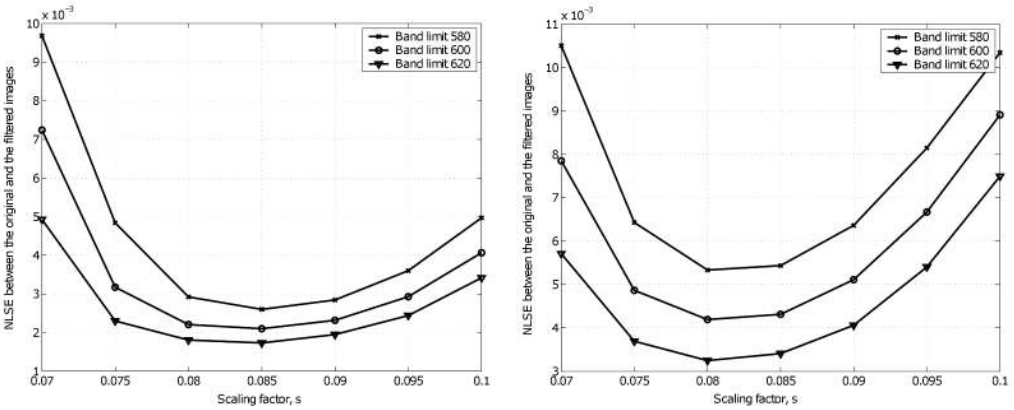


Figure 1.
Contour plot of $K_n(s)$



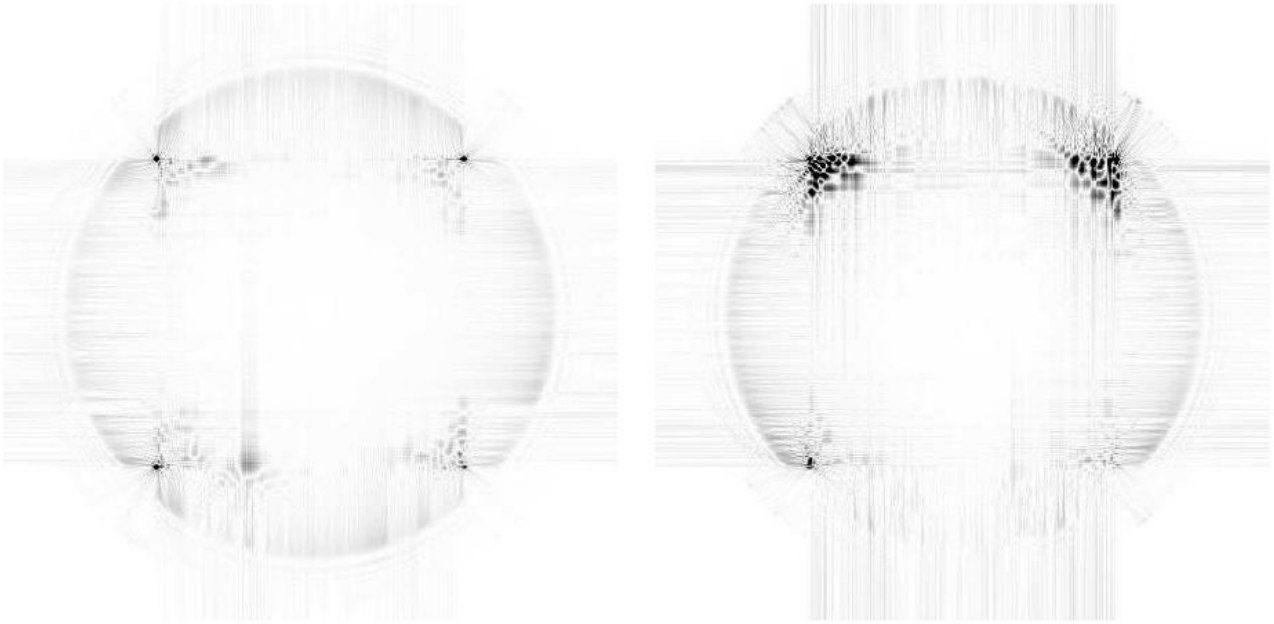
(a) Lena image (b) Baboon image

Figure 2.
Test images(512 × 512) from http://sipi.usc.edu/database/index.html



(a) Error in Lena image (b) Error in baboon image

Figure 3.
Normalized least-square error between the original image and the Hermite-filtered images



(a) Inverted difference image in Hermite filtering with Lena image

(b) Inverted difference image in Hermite filtering with baboon image

Figure 4.
Inverted difference images between the original and Hermite- filtered images (×100 amplified)

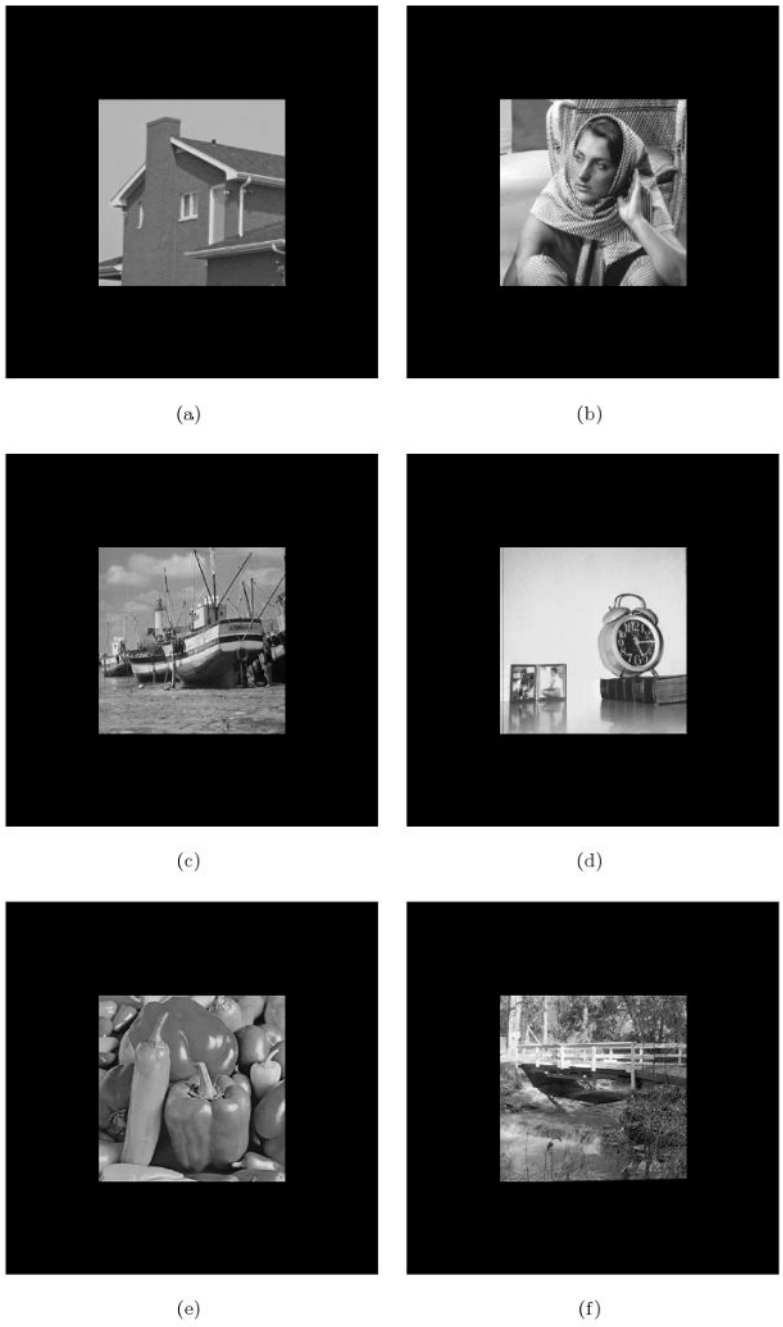


Figure 5.
Additional test images (512 × 512) from http://sipi.usc.edu/database/index.html (a) a house (b) Barbara (c) boats (d) a clock (e) peppers (f) a bridge

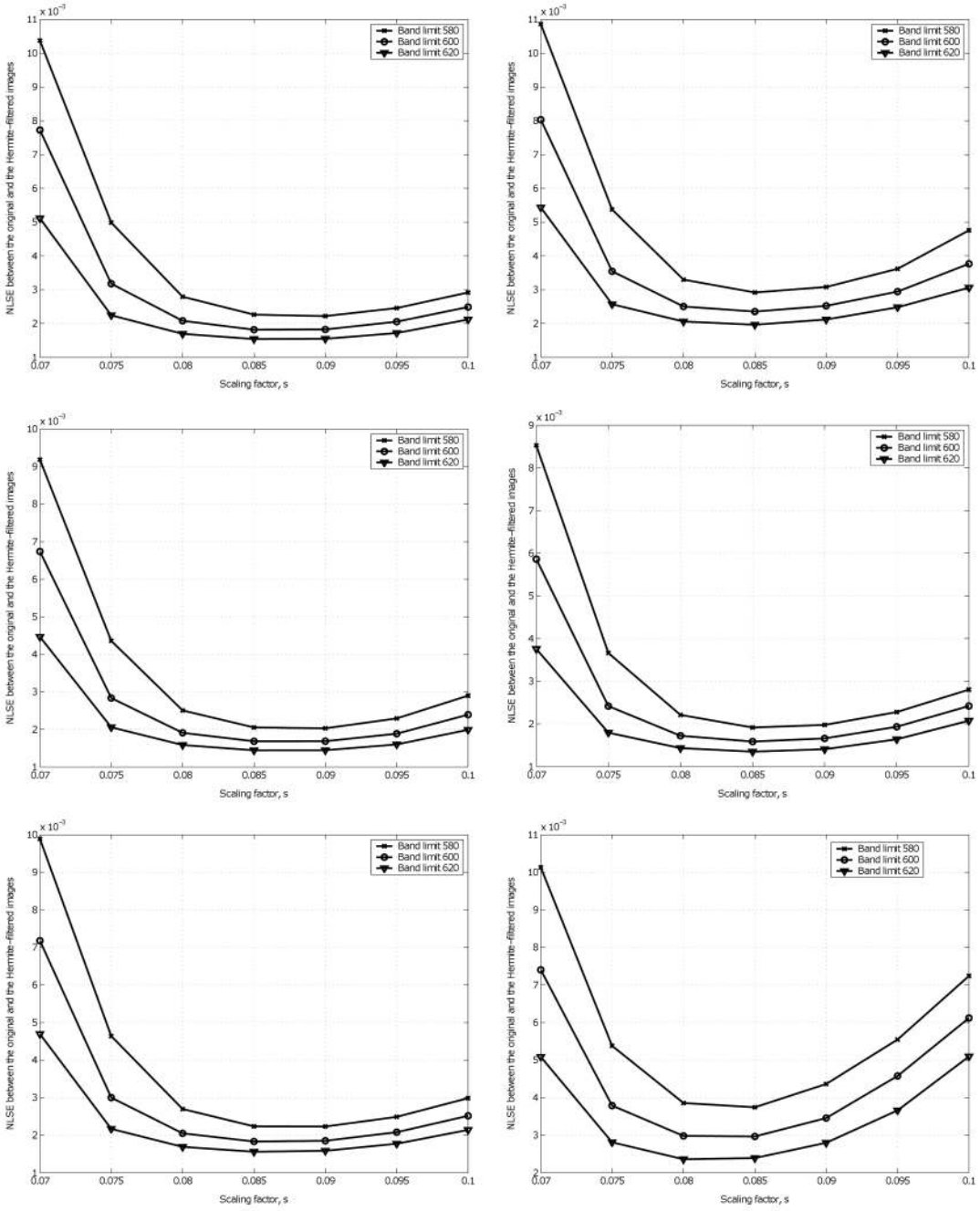
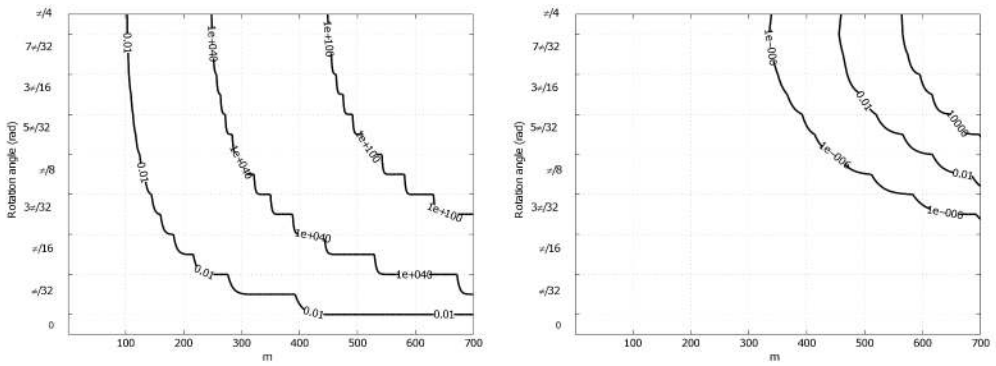


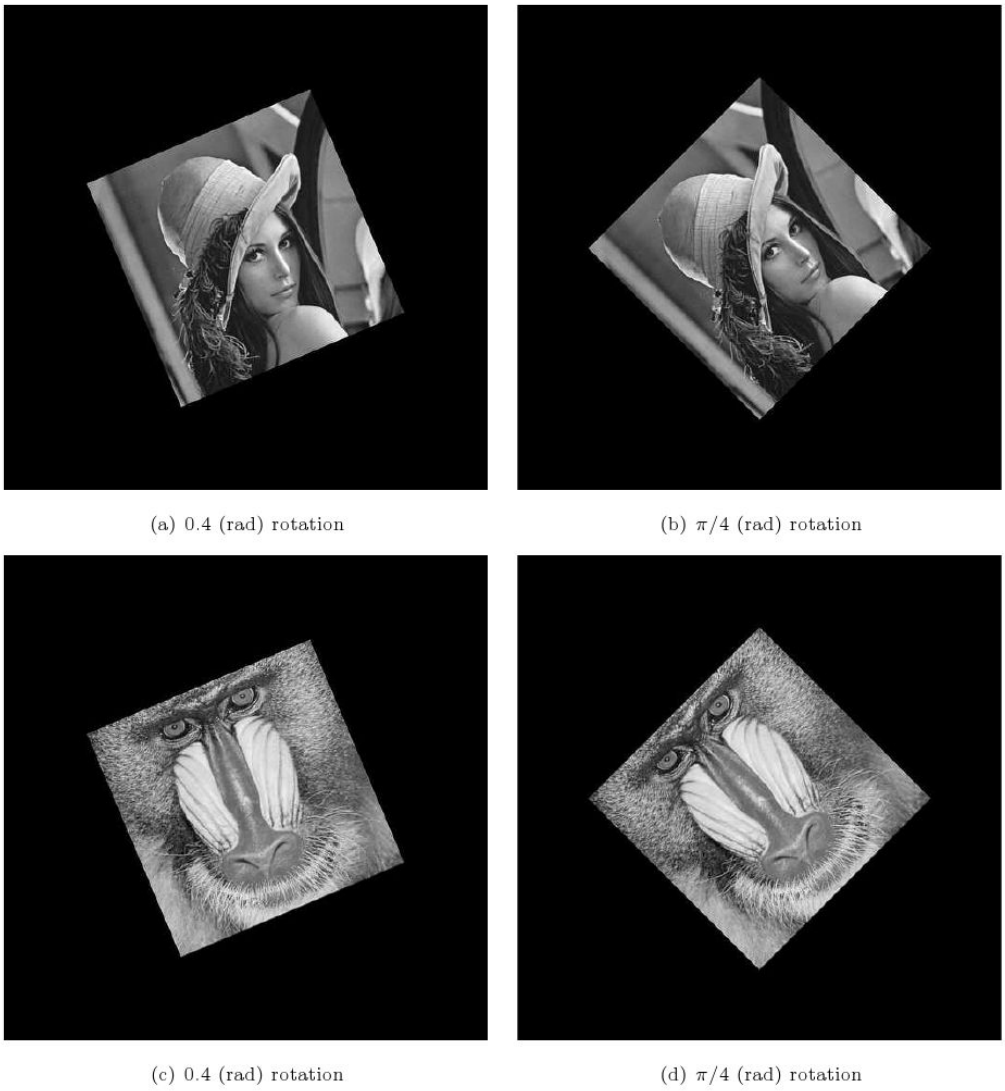
Figure 6.
Normalized least-square error between the original and the Hermite-filtered images. The images in Fig. 5 were used for this test.



(a) $\|S^m(\theta)S^m(\theta)^T - I\|/\|I\|$ when one recurrence formula is used.

(b) $\|S^m(\theta)S^m(\theta)^T - I\|/\|I\|$ when the average of the recurrence formulas is used.

Figure 7.
Contour plots of $\|S^m(\theta)S^m(\theta)^T - I\|/\|I\|$, where I is the identity matrix



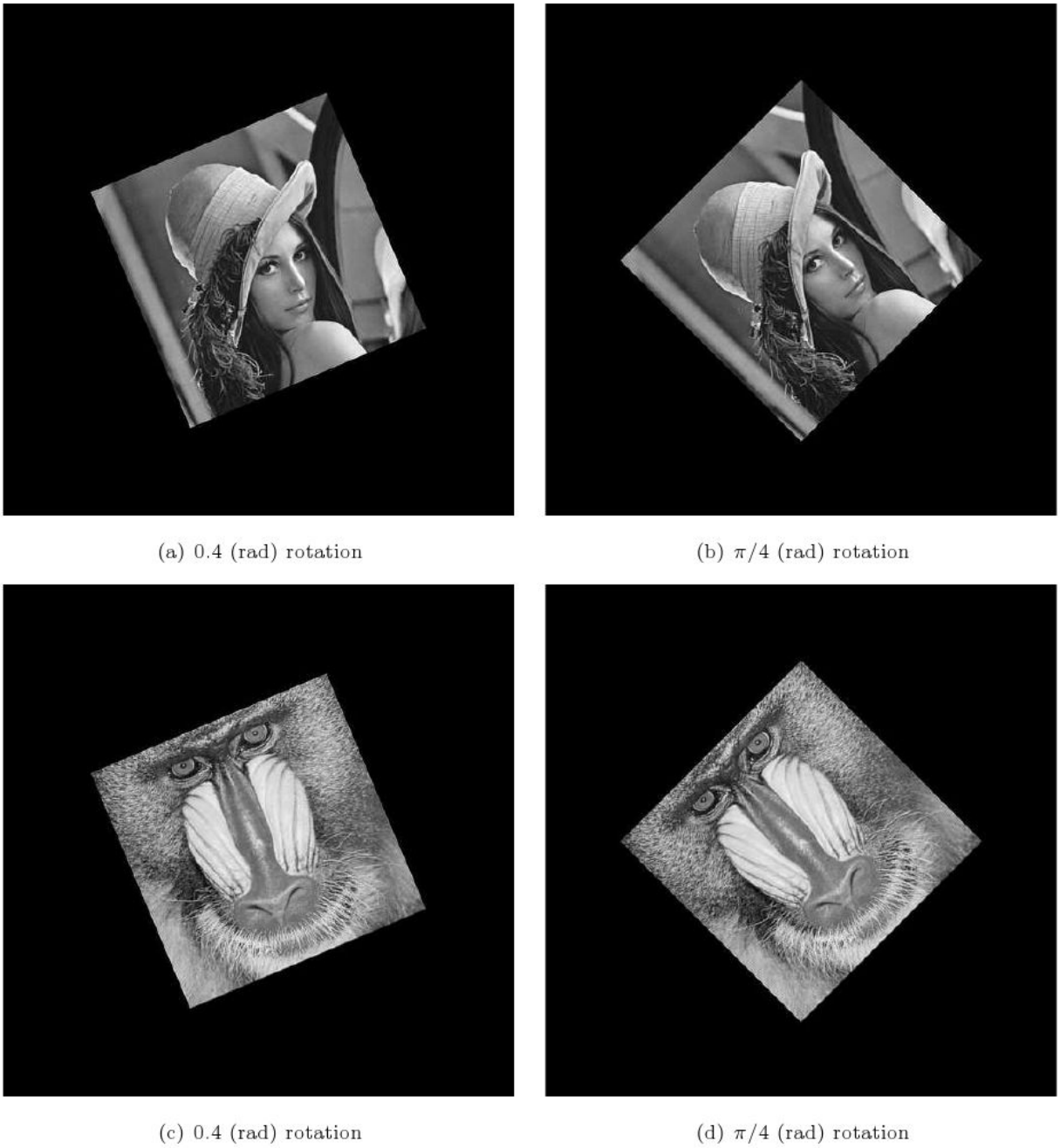
(a) 0.4 (rad) rotation

(b) $\pi/4$ (rad) rotation

(c) 0.4 (rad) rotation

(d) $\pi/4$ (rad) rotation

Figure 8.
Rotated images by Method A. The recurrence formulas were used to compute $S^m(\theta)$



(a) 0.4 (rad) rotation

(b) $\pi/4$ (rad) rotation

(c) 0.4 (rad) rotation

(d) $\pi/4$ (rad) rotation

Figure 9.
Rotated images by Method B. The matrix multiplication in (19) was used to compute $S^m(\theta)$.

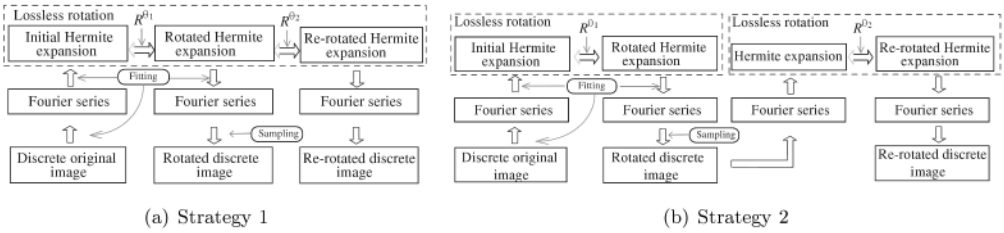
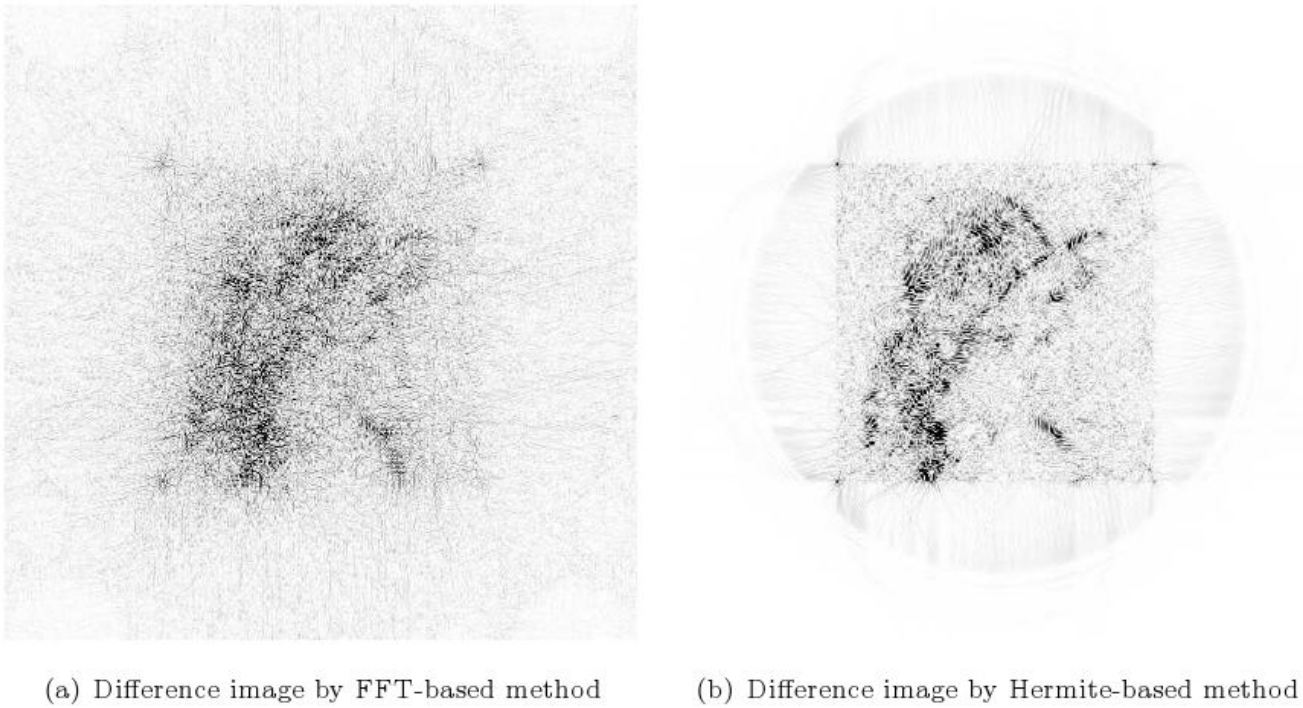


Figure 10.
Strategies for consecutive rotations



(a) Difference image by FFT-based method (b) Difference image by Hermite-based method

Figure 11.
Inverted difference images between the original Lena image and the resulting image in Test 3 (× 30 amplified)

Table 1

Image similarity measurements in Test 1 using the Fourier series for fitting and resampling. The images are rotated using each method. The rotated images are fitted to the Fourier series and then resampled in the original grid. The difference between the original images and the resampled images are obtained using the NLSE, the Sobolev norm and the relative entropy.

	0.4(rad)	**π/4(rad)**		**0.4(rad)**	**π/4(rad)**
FFT method	0.0170	0.0250	FFT method	0.0281	0.0379
Method A	0.0132	0.0163	Method A	0.0213	0.0248
Method B	0.0132	0.0164	Method B	0.0213	0.0249
(a) NLSE with Lena image			(b) NLSE with baboon image		
	0.4(rad)	π/4(rad)		0.4(rad)	π/4(rad)
FFT method	4.9021	8.8305	FFT method	7.6456	12.6874
Method A	1.5208	1.8713	Method A	3.0052	3.5018
Method B	1.5208	1.8713	Method B	3.0052	3.5018
(c) Sobolev norm with Lena image			(d) Sobolev norm with baboon image		
	0.4(rad)	π/4(rad)		0.4(rad)	π/4(rad)
FFT method	0.9885×10^4	3.6614×10^4	FFT method	1.9264×10^4	5.5844×10^4
Method A	0.1153×10^4	0.1729×10^4	Method A	0.2692×10^4	0.3695×10^4
Method B	0.1153×10^4	0.1729×10^4	Method B	0.2692×10^4	0.3695×10^4
(e) Relative entropy with Lena image			(f) Relative entropy with baboon image		

Table 2

Image similarity measurements in Test 1 using the Bi-cubic interpolation for resampling. The images are rotated using each method. The rotated images are resampled in the original grid using the Bi-cubic interpolation. The difference between the original images and the resampled images are obtained using the NLSE, the Sobolev norm and the relative entropy.

	0.4(rad)	**π/4(rad)**		**0.4(rad)**	**π/4(rad)**
FFT method	0.0280	0.0303	FFT method	0.0397	0.0427
Method A	0.0270	0.0280	Method A	0.0390	0.0394
Method B	0.0278	0.0280	Method B	0.0390	0.0394
(a) NLSE with Lena image			(b) NLSE with baboon image		
	0.4(rad)	π/4(rad)		0.4(rad)	π/4(rad)
FFT method	5.0721	8.3898	FFT method	7.8606	11.9270
Method A	2.9962	3.0589	Method A	5.2014	5.3275
Method B	2.9962	3.0589	Method B	5.2014	5.3275
(c) Sobolev norm with Lena image			(d) Sobolev norm with baboon image		
	0.4(rad)	π/4(rad)		0.4(rad)	π/4(rad)
FFT method	0.7223×10^4	3.1048×10^4	FFT method	1.7012×10^4	4.7262×10^4
Method A	0.0415×10^4	0.1948×10^4	Method A	0.5350×10^4	0.6762×10^4
Method B	0.0415×10^4	0.1948×10^4	Method B	0.5350×10^4	0.6762×10^4
(e) Relative entropy with Lena image			(f) Relative entropy with baboon image		

Table 3

Normalized least-square errors in accuracy tests

	Test 2	**Test 3**	**Test 4**		**Test 2**	**Test 3**	**Test 4**
FFT method	0.0286	0.0240	0.0192	FFT method	0.0461	0.0395	0.0315
Method A	0.0164	0.0178	0.0142	Method A	0.0248	0.0277	0.0226
Method B	0.0164	0.0178	0.0142	Method B	0.0248	0.0277	0.0226
(a) NLSE with Fig. 2(a)				(b) NLSE with Fig. 2(b)			
	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	0.0111	0.0101	0.0076	FFT method	0.0402	0.0312	0.0232
Method A	0.0058	0.0066	0.0058	Method A	0.0150	0.0201	0.0163
Method B	0.0058	0.0066	0.0058	Method B	0.0150	0.0201	0.0163
(c) NLSE with Fig. 5(a)				(d) NLSE with Fig. 5(b)			
	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	0.0188	0.0170	0.0115	FFT method	0.0128	0.0111	0.0091
Method A	0.0085	0.0104	0.0084	Method A	0.0067	0.0077	0.0066
Method B	0.0085	0.0104	0.0084	Method B	0.0067	0.0077	0.0066
(e) NLSE with Fig. 5(c)				(f) NLSE with Fig. 5(d)			
	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	0.0133	0.0118	0.0091	FFT method	0.0343	0.0291	0.0230
Method A	0.0069	0.0081	0.0068	Method A	0.0182	0.0204	0.0167
Method B	0.0069	0.0081	0.0068	Method B	0.0182	0.0204	0.0167

	Test 2	Test 3	Test 4
(g) NLSE with Fig. 5(e)			

	Test 2	Test 3	Test 4
(h) NLSE with Fig. 5(f)			

Table 4

Sobolev norms in accuracy tests

	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	3.2326	2.6526	2.202	FFT method	6.3865	5.3231	4.4255
Method A	1.8687	2.0172	1.6251	Method A	3.4914	3.8744	3.1799
Method B	1.8687	2.0172	1.6251	Method B	3.4914	3.8744	3.1799
(a) Sobolev norms with Fig. 2(a)				(b) Sobolev norms with Fig. 2(b)			
	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	1.4386	1.3029	1.0224	FFT method	5.5646	4.1711	3.1612
Method A	0.7720	0.8622	0.7535	Method A	2.089	2.7784	2.2495
Method B	0.7720	0.8622	0.7535	Method B	2.089	2.7784	2.2495
(c) Sobolev norms with Fig. 5(a)				(d) Sobolev norms with Fig. 5(b)			
	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	2.4147	2.1361	1.506	FFT method	2.3477	1.9894	1.6905
Method A	1.1041	1.3342	1.0769	Method A	1.2500	1.4248	1.2134
Method B	1.1041	1.3342	1.0769	Method B	1.2500	1.4248	1.2134
(e) Sobolev norms with Fig. 5(c)				(f) Sobolev norms with Fig. 5(d)			
	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	1.5474	1.363	1.0796	FFT method	4.0430	3.3884	2.762
Method A	0.8160	0.9449	0.793	Method A	2.1701	2.4189	1.9961
Method B	0.8160	0.9449	0.793	Method B	2.1701	2.4189	1.9961

	Test 2	Test 3	Test 4
(g) Sobolev norms with Fig. 5(e)			

	Test 2	Test 3	Test 4
(h) Sobolev norms with Fig. 5(f)			

Table 5

Relative entropies in accuracy tests

	Test 2	**Test 3**	**Test 4**		**Test 2**	**Test 3**	**Test 4**
FFT method	5.5712	3.4537	2.1942	FFT method	1.2774	0.7828	0.5224
Method A	1.7029	1.9912	1.2616	Method A	0.3670	0.4562	0.3015
Method B	1.7029	1.9912	1.2616	Method B	0.3670	0.4562	0.3015
(a) Relative entropies ($\times 10^3$) with Fig. 2(a)				(b) Relative entropies ($\times 10^4$) with Fig. 2(b)			
	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	6.4998	4.1312	2.6968	FFT method	8.7381	5.2935	2.8239
Method A	1.5382	1.3286	1.0871	Method A	1.1608	1.9864	1.4085
Method B	1.5382	1.3286	1.0871	Method B	1.1608	1.9864	1.4085
(c) Relative entropies ($\times 10^2$) with Fig. 5(a)				(d) Relative entropies ($\times 10^3$) with Fig. 5(b)			
	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	2.4017	1.586	0.7964	FFT method	1.922	1.1588	0.8024
Method A	0.4629	0.6305	0.4045	Method A	0.4963	0.5854	0.4185
Method B	0.4629	0.6305	0.4045	Method B	0.4963	0.5854	0.4185
(e) Relative entropies ($\times 10^3$) with Fig. 5(c)				(f) Relative entropies ($\times 10^3$) with Fig. 5(d)			
	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	9.7337	5.8649	3.7901	FFT method	6.8864	4.3443	2.8083
Method A	2.3185	2.7185	1.8420	Method A	1.9387	2.3715	1.6125
Method B	2.3185	2.7185	1.8420	Method B	1.9387	2.3715	1.6125

IEEE Trans Image Process. Author manuscript; available in PMC 2010 March 12.

	Test 2	Test 3	Test 4
(g) Relative entropies ($\times 10^2$)with Fig. 5(e)			

	Test 2	Test 3	Test 4
(h) Relative entropies($\times 10^3$) with Fig. 5(f)			

Table 6

Elapsed time in seconds for the image rotation process measured in seconds. Forward and Inverse Hermite transforms refer to calculation in Section 4.1 and 4.2, respectively.

Image size (band limit)	128×128 (155)	256×256 (310)	512×512 (620)
Forward Hermite transform	0.094	0.469	2.875
Inverse Hermite transform	0.063	0.453	2.675
Recurrence calculation in Method A	0.688	6.281	56.422
Loading $S^m(\pi/4)$ for Method B	0.719	0.703	23.250
Matrix multiplication for Method B	0.109	1.078	11.110