

Accurate Image Rotation Using Hermite Expansions

Wooram Park, *Student Member, IEEE*, Gregory Leibon, Daniel N. Rockmore, and Gregory S. Chirikjian, *Senior Member, IEEE*

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 2-D continuous bandlimited Hermite expansion is rotated, the resulting function can be expressed as a Hermite expansion with the same bandlimit. Furthermore, the Hermite coefficients of the initial 2-D 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 them with an existing FFT-based method implementing three shears. We find that the method proposed here consistently has better accuracy than the FFT-based method.

Index Terms—Bandlimited expansions, Hermite expansions, image rotation.

I. INTRODUCTION

TECHNIQUES for effecting a 2-D 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 behaves well under rotation, the discrete Fourier expansions used in image analysis do

Manuscript received August 06, 2008; revised May 05, 2009. First published June 05, 2009; current version published August 14, 2009. This work was supported in part by NIH Grant R01GM075310 “Group-Theoretic Methods in Protein Structure Determination” and in part by NIH Grant R01EB006435 “Steering Flexible Needles in Soft Tissue.” The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thomas S. Denney.

W. Park and G. S. Chirikjian are with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: wpark7@jhu.edu; gregc@jhu.edu).

G. Leibon and D. N. Rockmore are with the Department of Mathematics, Dartmouth College, Hanover, NH 03755 USA (e-mail: gleibon@gmail.com; rockmore@math.dartmouth.edu).

Digital Object Identifier 10.1109/TIP.2009.2024582

not share this property. This may seem surprising at first. Let us consider the Fourier transform, $\tilde{f}(\omega)$, for a nonperiodic function $f(x)$, where $x, \omega \in \mathbb{R}^2$. If we apply the Fourier transform to the rotated version of $f(x)$

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

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 2-D 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 bandlimited 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 bandlimited 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 II, a brief literature review is given in the field of image rotations and Hermite-function-based image processing. In Section III, using the fact that the rotation of a 2-D bandlimited Hermite expansion preserves the bandlimit, 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 IV we propose a method to connect the bandlimited Hermite expansion and an arbitrary

2-D array of discrete data. Then in Section V, we present computational results and compare the accuracy of our new methods with an existing FFT-based image rotation approach. Finally, in Section VI, we summarize our conclusions.

II. RELATED WORK

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 2-D 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 3-D 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 three shear processes to the original image and each shear process is implemented by the FFT. In Section V, we will compare our new methods with this method.

For the purpose of achieving image rotations, steerable filters [5] can be a reasonable tool, since the class of the steerable filters has the special property of remaining bandlimited under rotation. A steerable filter at any orientation can be constructed as a linear combination of the basis filters. Steerable filters have been studied extensively and are widely used (see, e.g., [6], [25], [26], and [29]). They have also been extended using the theory of Lie groups [14].

A Hermite transform (i.e., expansion of a function in \mathbb{R}^2 using Hermite functions) 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 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. This enables the capture of 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.

III. ROTATION OF 2-D HERMITE EXPANSIONS

A. 2-D Hermite Expansions and Steerable Filters

Hermite polynomials are generated by the Rodrigues formula

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

Hermite functions are defined as [3], [27]

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

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

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

or equivalently

$$f(x, y) = \sum_{m=0}^N \sum_{n=0}^m \hat{f}_{n,m-n} h_n(x) h_{m-n}(y) \quad (2)$$

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

Rotation of 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, its steerability can be written as [25]

$$\begin{aligned} & \mathcal{R}^\theta(h_n(x) h_{m-n}(y)) \\ &= h_n(x \cos \theta + y \sin \theta) h_{m-n}(-x \sin \theta + y \cos \theta) \\ &= \sum_{k=0}^m S_{k,n}^m(\theta) h_k(x) h_{m-k}(y) \end{aligned} \quad (3)$$

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

$$H_n(z \sin t + w \cos t) = \sum_{k=0}^n C_n^k \sin^k t \cos^{n-k} t H_k(z) H_{n-k}(w) \quad (4)$$

where $C_n^k = n!/(k!(n-k)!)$, we can also derive (3) without prior knowledge of steerable filters, as was done in [21].

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

$$f^\theta(x, y) = \sum_{m=0}^N \sum_{k=0}^m \left(\sum_{n=0}^m \hat{f}_{n,m-n} S_{k,n}^m(\theta) \right) h_k(x) h_{m-k}(y). \quad (5)$$

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

$$\hat{f}_{k,m-k}^\theta = \sum_{n=0}^m S_{k,n}^m(\theta) \hat{f}_{n,m-n} \quad (6)$$

for $m = 0, 1, \dots, N$. It is clear that the coefficients \hat{f}^θ and \hat{f} are linearly and locally related. Namely, the relation (6) can be written as

$$\hat{f}_m^\theta = S^m(\theta) \hat{f}_m \quad (7)$$

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

$$\hat{f}_m^\theta = [\hat{f}_{0,m}^\theta \quad \hat{f}_{1,m-1}^\theta \quad \cdots \quad \hat{f}_{m,0}^\theta]^T$$

and

$$\hat{f}_m = [\hat{f}_{0,m} \quad \hat{f}_{1,m-1} \quad \cdots \quad \hat{f}_{m,0}]^T.$$

B. Recurrence Formulas for $S^m(\theta)$

For low orders, the transformation matrix $S^m(\theta)$ can be derived using (3) and (4). The first few $S^m(\theta)$ matrices are

$$\begin{aligned} S^0(\theta) &= 1 \\ S^1(\theta) &= \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \\ S^2(\theta) &= \begin{pmatrix} \cos^2 \theta & \sqrt{2} \cos \theta \sin \theta & \sin^2 \theta \\ -\sqrt{2} \cos \theta \sin \theta & \cos^2 \theta - \sin^2 \theta & \sqrt{2} \cos \theta \sin \theta \\ \sin^2 \theta & -\sqrt{2} \cos \theta \sin \theta & \cos^2 \theta \end{pmatrix}. \end{aligned}$$

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

Multiplying $h_q(x)h_{m-q}(y)$ on both sides of (3) and integrating over \mathbb{R}^2 gives

$$\begin{aligned} S_{q,n}^m(\theta) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_q(x)h_{m-q}(y)h_n(x \cos \theta + y \sin \theta) \\ &\quad \times h_{m-n}(-x \sin \theta + y \cos \theta) dx dy. \end{aligned} \quad (8)$$

Let us consider

$$\begin{aligned} S_{q+1,n}^{m+1}(\theta) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_{q+1}(x)h_{m-q}(y) \\ &\quad \times h_n(x \cos \theta + y \sin \theta) \\ &\quad \times h_{m+1-n}(-x \sin \theta + y \cos \theta) dx dy. \end{aligned}$$

If we apply the recurrence formula for Hermite functions

$$\sqrt{n+1}h_{n+1}(x) = \sqrt{2}xh_n(x) - \sqrt{n}h_{n-1}(x) \quad (9)$$

then we have

$$\begin{aligned} &\sqrt{q+1}S_{q+1,n}^{m+1}(\theta) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(\sqrt{2}xh_q(x) - \sqrt{q}h_{q-1}(x) \right) h_{m-q}(y) \\ &\quad \times h_n(x \cos \theta + y \sin \theta) \\ &\quad \times h_{m+1-n}(-x \sin \theta + y \cos \theta) dx dy \\ &= \sqrt{2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xh_q(x)h_{m-q}(y)h_n(x \cos \theta + y \sin \theta) \\ &\quad \times h_{m+1-n}(-x \sin \theta + y \cos \theta) dx dy \end{aligned}$$

since (3) implies that when $m' \neq m$

$$\begin{aligned} &\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_q(x)h_{m'-q}(y)h_n(x \cos \theta + y \sin \theta) \\ &\quad \times h_{m-n}(-x \sin \theta + y \cos \theta) dx dy = 0. \end{aligned} \quad (10)$$

Similarly, it can be shown that

$$\begin{aligned} &\sqrt{m-q+1}S_{q,n}^{m+1}(\theta) \\ &= \sqrt{2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} yh_q(x)h_{m-q}(y) \\ &\quad \times h_n(x \cos \theta + y \sin \theta) \\ &\quad \times h_{m+1-n}(-x \sin \theta + y \cos \theta) dx dy. \end{aligned}$$

Now, we compute

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

Using the recurrence formula (9), we have

$$\begin{aligned} &\sqrt{q+1}S_{q+1,n}^{m+1}(\theta) \cos \theta + \sqrt{m-q+1}S_{q,n}^{m+1}(\theta) \sin \theta \\ &= \sqrt{n+1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_q(x)h_{m-q}(y) \\ &\quad \times h_{n+1}(x \cos \theta + y \sin \theta) \\ &\quad \times h_{m+1-n}(-x \sin \theta + y \cos \theta) dx dy \\ &+ \sqrt{n} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_q(x)h_{m-q}(y) \\ &\quad \times h_{n-1}(x \cos \theta + y \sin \theta) \\ &\quad \times h_{m+1-n}(-x \sin \theta + y \cos \theta) dx dy. \end{aligned}$$

Since the first term of the righthand side is zero due to (10), we have

$$\begin{aligned} &\sqrt{q+1}S_{q+1,n}^{m+1}(\theta) \cos \theta + \sqrt{m-q+1}S_{q,n}^{m+1}(\theta) \sin \theta \\ &= \sqrt{n}S_{q,n-1}^m(\theta). \end{aligned}$$

Similarly, it can be shown that

$$\begin{aligned} &-\sqrt{q+1}S_{q+1,n}^{m+1}(\theta) \sin \theta + \sqrt{m-q+1}S_{q,n}^{m+1}(\theta) \cos \theta \\ &= \sqrt{m-n+1}S_{q,n}^m(\theta). \end{aligned}$$

Consequently, we have

$$\begin{aligned} \begin{pmatrix} S_{q+1,n}^{m+1}(\theta) \\ S_{q,n}^{m+1}(\theta) \end{pmatrix} &= \begin{pmatrix} \frac{1}{\sqrt{q+1}} & 0 \\ 0 & \frac{1}{\sqrt{m-n+1}} \end{pmatrix} \\ &\times \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \sqrt{n}S_{q,n-1}^m(\theta) \\ \sqrt{m-n+1}S_{q,n}^m(\theta) \end{pmatrix}. \end{aligned} \quad (11)$$

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

$$\begin{aligned} \begin{pmatrix} S_{q,n+1}^{m+1}(\theta) \\ S_{q,n}^{m+1}(\theta) \end{pmatrix} &= \begin{pmatrix} \frac{1}{\sqrt{n+1}} & 0 \\ 0 & \frac{1}{\sqrt{m-n+1}} \end{pmatrix} \\ &\times \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \sqrt{q}S_{q-1,n}^m(\theta) \\ \sqrt{m-q+1}S_{q,n}^m(\theta) \end{pmatrix}. \end{aligned} \quad (12)$$

C. Properties of $S^m(\theta)$

It is easy to show that

$$\{S^m(\theta)\}^T S^m(\theta) = I_m \quad (13)$$

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 (13) holds, $S^m(\theta)$ is an orthogonal matrix.

On the other hand, using (8) we can compute $S_{p,q}^m(-\theta)$ as

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

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 (6) 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

$$S^m(\theta) = \exp(\theta \Omega^m) \quad (14)$$

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

$$\begin{aligned} \Omega_{q,n}^m &= \frac{d}{d\theta} (S_{q,n}^m(\theta))|_{\theta=0} \\ &= \sqrt{n} \sqrt{m-n+1} \delta_{q,n-1} \\ &\quad - \sqrt{n+1} \sqrt{m-n} \delta_{q,n+1}. \end{aligned}$$

The detailed derivation is given in Appendix B.

D. Relation Between $S^m(\theta)$ and Generalized Associated Legendre Function

In this subsection, we connect $S^m(\theta)$ and the generalized associated Legendre functions, $P_{mn}^l(x)$. This provides a simple factorization of the matrix $S^m(\theta)$ that will facilitate its calculation.

The irreducible unitary representation (IUR) matrix elements of $\text{SO}(3)$ are given by [3]

$$U_{mn}^l(R(\alpha, \beta, \gamma)) = (-1)^{n-m} e^{-i(m\alpha+n\gamma)} P_{mn}^l(\cos \beta)$$

where α, β and γ are the ZYZ Euler angles, and $P_{mn}^l(x)$ is the generalized associated Legendre function. The integral form of $P_{mn}^l(x)$ is

$$\begin{aligned} P_{mn}^l(\cos \beta) &= \frac{i^{n-m}}{2\pi} \left[\frac{(l-m)!(l+m)!}{(l-n)!(l+n)!} \right]^{1/2} \\ &\quad \times \int_0^{2\pi} \left(\cos \frac{\beta}{2} e^{i\phi/2} + i \sin \frac{\beta}{2} e^{-i\phi/2} \right)^{l-n} \\ &\quad \times \left(\cos \frac{\beta}{2} e^{-i\phi/2} + i \sin \frac{\beta}{2} e^{i\phi/2} \right)^{l+n} e^{im\phi} d\phi. \quad (15) \end{aligned}$$

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

Since U_{mn}^l are the IUR matrix of $\text{SO}(3)$, 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 exists a skew-symmetric matrix, Γ^l such that $U^l(2\theta) = \exp(\Gamma^l \theta)$. In order to find Γ , we will compute $dU_{mn}^l(2\theta)/d\theta$ at $\theta = 0$. We consider $U_{mn}^l(2\theta)$ instead of $U_{mn}^l(\theta)$, because the former is easier to connect to $S^m(\theta)$.

The derivative at $\theta = 0$ is

$$\begin{aligned} \Gamma_{mn}^l &= \left. \frac{dU_{mn}^l(2\theta)}{d\theta} \right|_{\theta=0} \\ &= (-1)^{n-m} \left. \frac{dP_{mn}^l(\cos 2\theta)}{d\theta} \right|_{\theta=0} \end{aligned}$$

and

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

Therefore

$$\begin{aligned} \Gamma_{mn}^l &= (-1)^{n-m} \left. \frac{dP_{mn}^l(\cos 2\theta)}{d\theta} \right|_{\theta=0} \\ &= \sqrt{(l+n)(l-n+1)} \delta_{m+1,n} - \sqrt{(l-n)(l+n+1)} \delta_{m,n+1}. \end{aligned}$$

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

$$\begin{aligned} \Gamma_{mn}^l &= \sqrt{(q')(m'-q'+1)} \delta_{n'+1,q'} \\ &\quad - \sqrt{(m'-q')(q'+1)} \delta_{n',q'+1} \\ &= \Omega_{n',q'}^{m'}. \end{aligned}$$

where Ω is the skew-symmetric matrix defined in Section III-C. 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

$$S_{n'q'}^{m'}(\theta) = (-1)^{n-m} P_{mn}^l(\cos 2\theta) \quad (16)$$

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

It is worthwhile to consider the following identity for P_{mn}^l (from [18]):

$$\begin{aligned} P_{MM'}^l(\cos \beta) &= i^{M-M'} \sum_{M''=-l}^l P_{M''M}^l \left(\cos \left(\frac{\pi}{2} \right) \right) \\ &\times e^{-iM''\beta} P_{M''M'}^l \left(\cos \left(\frac{\pi}{2} \right) \right). \end{aligned} \quad (17)$$

Note that $P_{mn}^l(\cos \beta)$ is also denoted by $d_{mn}^l(\beta)$ in many references including [18]. Since $d_{mn}^l(\beta)$ is defined by the same form as (15), $P_{mn}^l(\cos \beta)$ is equivalent to $d_{mn}^l(\beta)$.

Combining (16) and (17), we have the matrix factorization as

$$S^m(\theta) = E \cdot \left(S^m \left(\frac{\pi}{4} \right) \right)^T \cdot G(\theta) \cdot S^m \left(\frac{\pi}{4} \right) \cdot E^* \quad (18)$$

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

IV. CONNECTION BETWEEN 2-D BANDLIMITED HERMITE EXPANSION WITH DISCRETE IMAGES

In order to use Hermite expansions for image processing we need to adapt the continuous 2-D Hermite transform to the discrete setting. In this section we examine both the problem of discrete analysis (computing the Hermite expansion of a discretized image) as well as synthesis (computing a discretized image from a 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 2-D bandlimited Hermite expansion. Since one can easily extend the 1-D case to the 2-D case, let us focus on the 1-D 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 the 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.

A. Fitting a Hermite Expansion to a Fourier Series

Once the discrete data has 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 seek the Hermite coefficients $\{\hat{f}_k\}$ such that

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

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

Expanding the above cost function, we find that

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

where

$$\begin{aligned} a_{l_1, l_2} &= \int_{-\pi}^{\pi} h_{l_1} \left(\frac{\theta}{s} \right) h_{l_2} \left(\frac{\theta}{s} \right) d\theta \\ b_{k,l} &= \int_{-\pi}^{\pi} h_l \left(\frac{\theta}{s} \right) e^{ik\theta} d\theta \\ \|f\|^2 &= \int_{-\pi}^{\pi} |f(\theta)|^2 d\theta \end{aligned}$$

and \overline{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

$$\int_{-\pi}^{\pi} F \left(\frac{\theta}{s} \right) d\theta = s \int_{-\pi/s}^{\pi/s} F(x) dx \approx s \int_{-\infty}^{\infty} F(x) dx. \quad (19)$$

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

$$b_{k,l} \approx s \int_{-\infty}^{\infty} h_l(x) e^{iksx} dx = \sqrt{2\pi} i^l s h_l(ks). \quad (20)$$

The equality in (20) 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

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

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

We can rewrite the cost function as

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

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

$$\hat{f}_l \approx \sqrt{\frac{\pi}{2}} i^l \sum_{k=-B}^B (\tilde{f}_k + \tilde{f}_k^*(-1)^l) h_l(ks). \quad (21)$$

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. Thus, the scaling factor should be determined so that (19) is an accurate statement for the integrals that were approximated for a_{l_1, l_2} , and b_{kl} .

B. Fitting a Fourier Series to a 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 bandlimit 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 a bandlimited 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

$$\begin{aligned} C'(\{\tilde{f}_l\}) &= \int_{-\pi}^{\pi} \left| \sum_{k=-B}^B \tilde{f}_k e^{ik\theta} - \sum_{l=0}^M \hat{f}_l h_l \left(\frac{\theta}{s} \right) \right|^2 d\theta \\ &= 2\pi \sum_{k=-B}^B \left| \tilde{f}_k - \frac{s}{\sqrt{2\pi}} \sum_{l=0}^M (-i)^l h_l(ks) \hat{f}_l \right|^2 \\ &\quad + \text{constant}. \end{aligned}$$

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

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

C. Determination of the Scaling Factor

From (19) and the definition of a_{l_1, l_2} and $b_{k,l}$, it is clear that the range of the integral, $(-\pi/s, \pi/s)$ should be chosen to cap-

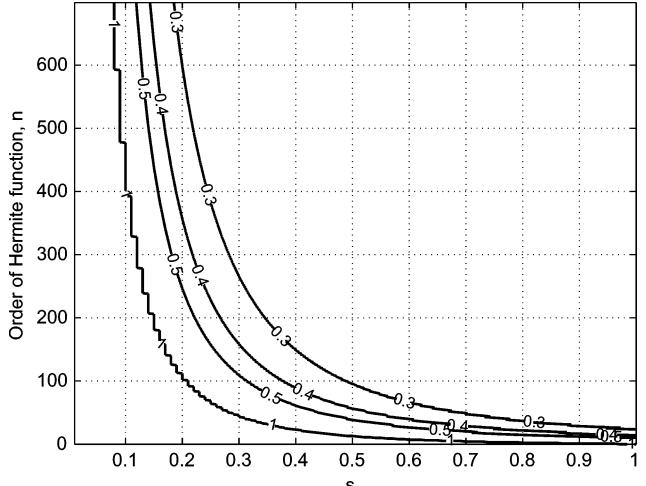


Fig. 1. Contour plot of $K_n(s)$.

ture 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, we 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 bandlimit of the Hermite expansion is 450, then $s \leq 0.1$ is good, since the corresponding range guarantees $K_n(0.1)$ is close enough to 1 for $n = 0, 1, \dots, 450$. 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 bandlimits 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 prepare 512×512 images from 256×256 images as shown in Fig. 2 by zero-padding the original image in such a way that they are centered in the larger grid. 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

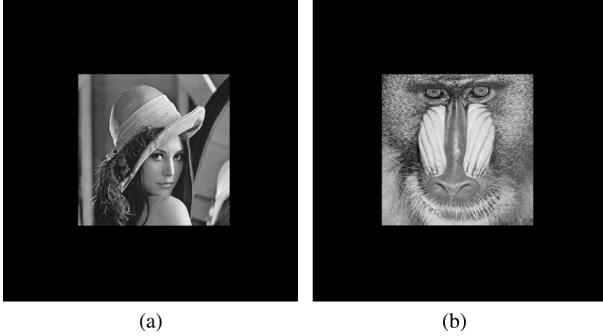


Fig. 2. Test images (512×512) from <http://sipi.usc.edu/database/index.html>. (a) Lena image. (b) Baboon image.

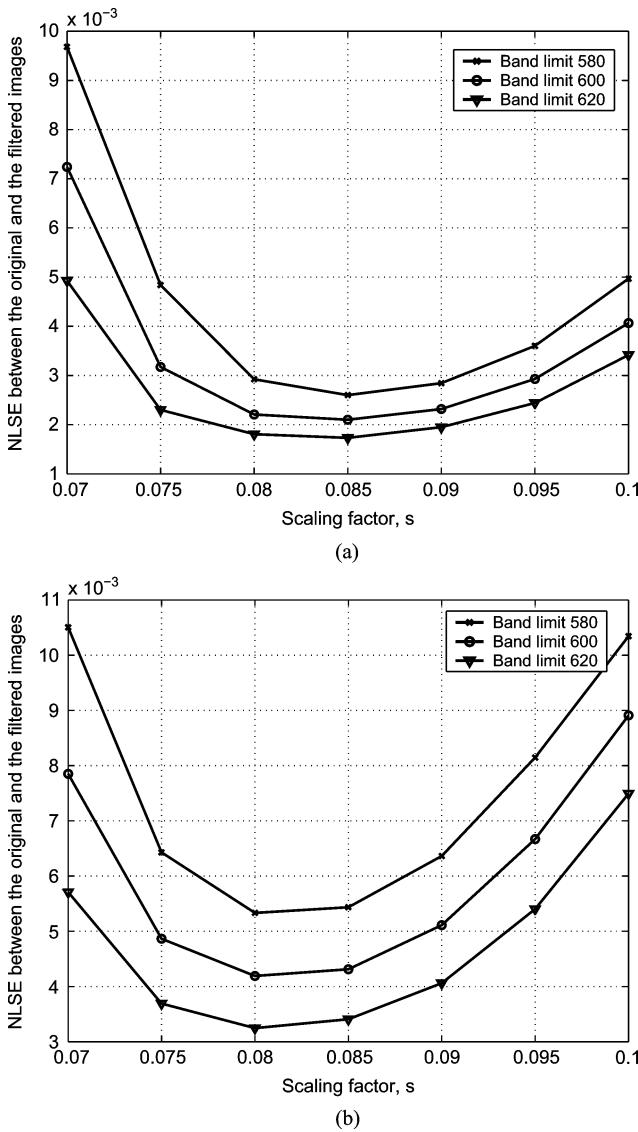


Fig. 3. Normalized least squared error between the original image and the Hermite-filtered images. (a) Error in Lena image. (b) Error in baboon image.

image between the original and the Hermite-filtered Lena images when the bandlimit is $N = 620$ and the scaling factor is $s = 0.085$. If we apply the same process to the baboon image shown in Fig. 2(b), we have the different error curve shown in

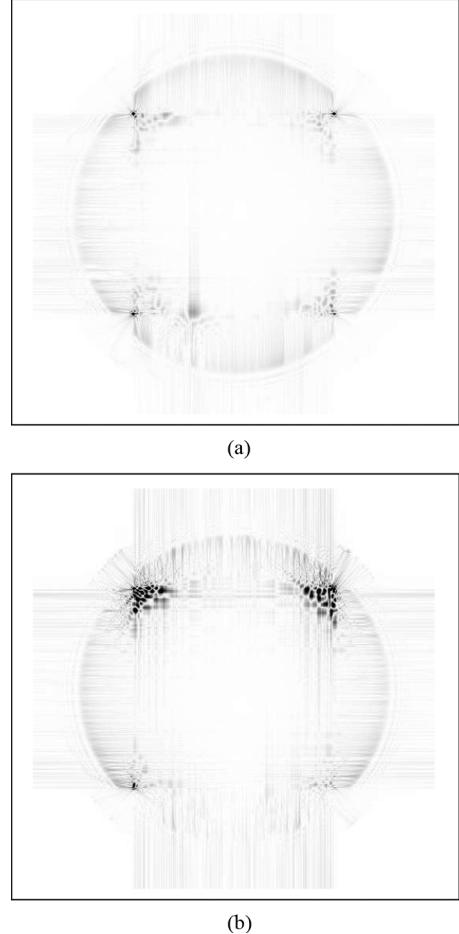


Fig. 4. Inverted difference images between the original and Hermite-filtered images ($\times 100$ amplified). (a) Inverted difference image in Hermite filtering with Lena image. (b) Inverted difference image in Hermite filtering with baboon image.

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 present the inverted difference images instead of the regular difference images that contain black pixels in most areas.

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.

V. ROTATION OF IMAGES AND ACCURACY TEST

A. Computation of $S^m(\theta)$

Once we have the Hermite coefficients from a discrete image, the rotation can be applied to the coefficients using (7). In order to compute $S^m(\theta)$, we may use the recurrence formulas, (11)

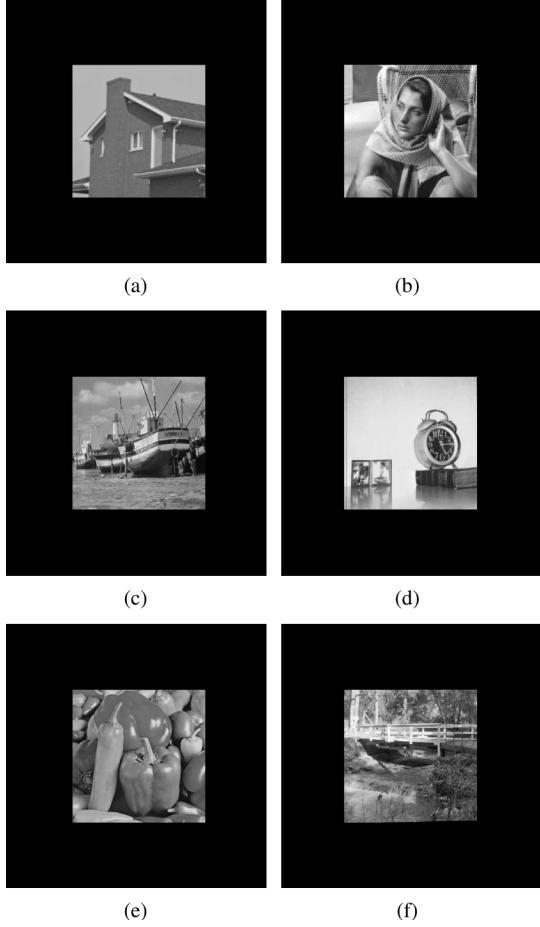


Fig. 5. Additional test images (512×512) from <http://sipi.usc.edu/database/index.html>. (a) A house. (b) Barbara. (c) Boats. (d) A clock. (e) Peppers. (f) A bridge.

and (12), or the exponential mapping, (14), or the matrix multiplication, (18). Since the matrix exponential of large matrices increases the computational time, we pursue two methods: the recurrence formulas and 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 $-\pi/4 \leq \theta \leq \pi/4$. 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 III-C, 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:

$$L(m, \theta) = \frac{\|S^m(\theta)S^m(\theta)^T - I_m\|}{\|I_m\|}$$

where I_m is the $(m+1) \times (m+1)$ identity matrix. Fig. 7(a) shows the contour plot of $L(m, \theta)$, when we apply the first recurrence formula in (11). If we average all 4 recurrence formulas in (11) and (12), $L(m, \theta)$ 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 $\mathbf{f}_m^\theta = S^m(\theta/2) \times (S^m(\theta/2) \times \mathbf{f}_m)$, instead of direct multiplication of the two 2-D matrices, in order to avoid increasing the computational complexity.

Alternatively, we can compute $S^m(\theta)$ by (18), which is denoted by Method B. Since the matrices, E and $S^m(\pi/4)$ in (18) 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 $S^m(\pi/4)$. Since it is constant, we will compute it, store it into a computer memory (or storage device) and then use it for rotations. Since computation of $S^m(\pi/4)$ is performed only once and is stored, we can sacrifice the computation time to have accurate $S^m(\pi/4)$. The matrix exponential mapping by Padé algorithm [15] can be used for this end. Explicitly, we use

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

where Ω^m is a skew-symmetric matrix defined in Section III-C. Practically, all $S^m(\pi/4)$ for $m = 0, 1, \dots, N$ 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 $S^m(\theta)$.

To compare Method A and B, Figs. 8 and 9 show examples of rotated images, when the original test images in Fig. 2 are used. In these examples, the NLSEs between Method A and B are less than 10^{-6} .

B. Accuracy Test

Now let us consider the accuracy of our methods. If the original image were defined as a continuous 2-D 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 ($\theta = 0, \pi/2, \pi, \dots$). Therefore, as an alternative, we perform the following test:

Test 1: Fit the rotated image to an appropriate continuous 2-D function and resample it onto the original grid to have an image close to the original image. 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]

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

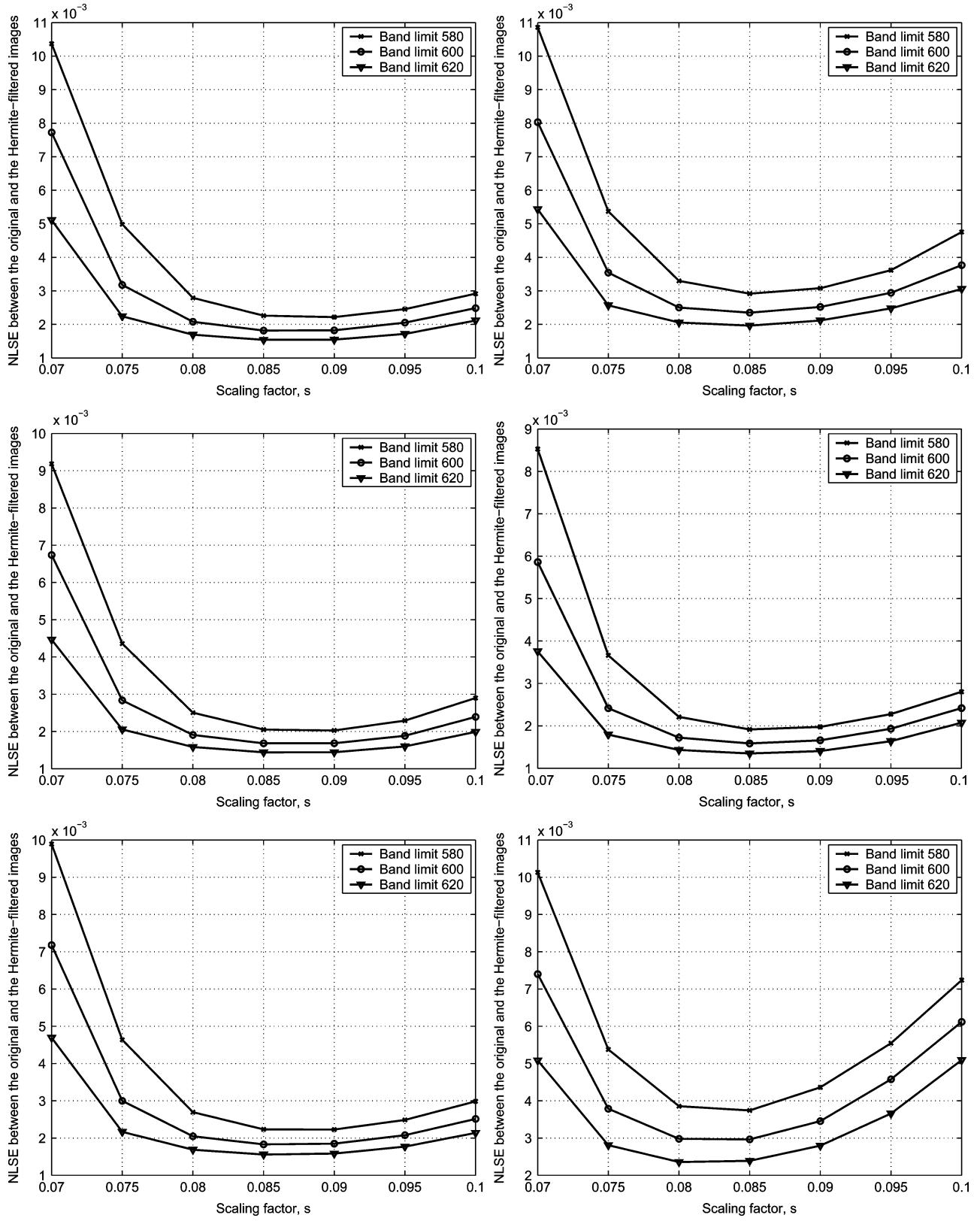


Fig. 6. Normalized least squared error between the original and the Hermite-filtered images. The images in Fig. 5 were used for this test.

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 2-D 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.

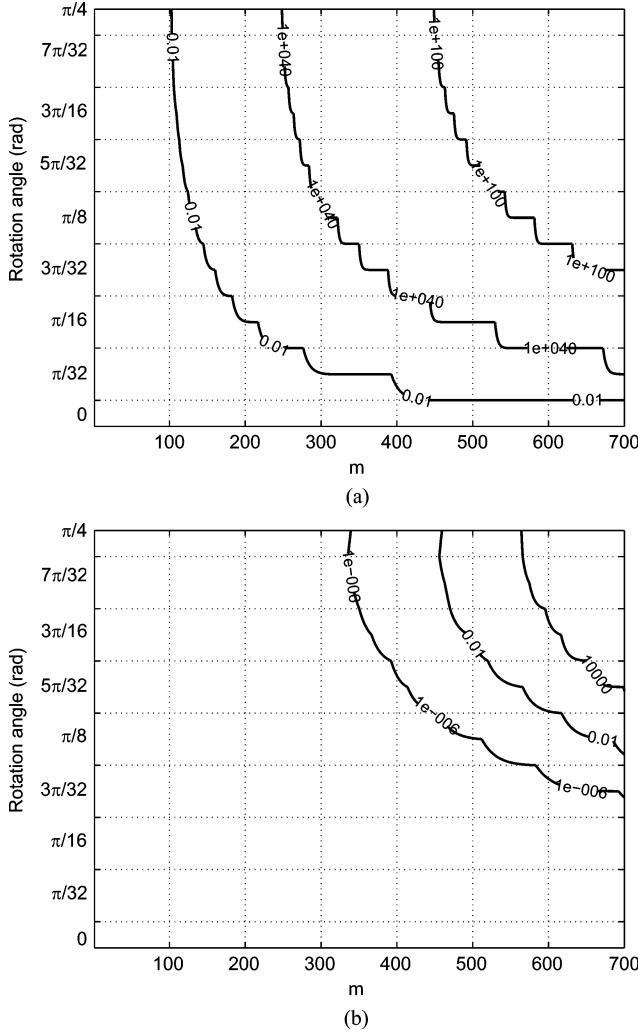


Fig. 7. Contour plots of $\|S^m(\theta)S^m(\theta)^T - I\|/\|I\|$, where I is the identity matrix. (a) $\|S^m(\theta)S^m(\theta)^T - I\|/\|I\|$ when one recurrence formula in (11) is used. (b) $\|S^m(\theta)S^m(\theta)^T - I\|/\|I\|$ when the average of the recurrence formulas in (11) and (12) is used.

Relative entropy (also known as Kullback–Leibler divergence) is an asymmetric measure of the discrepancy between two probability distributions [7]. It can also be used for measuring the difference between images. It is defined as [1]

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

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 Tables I and II. The 2-D Fourier series and bi-cubic interpolation were used in Tables I and II, 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,

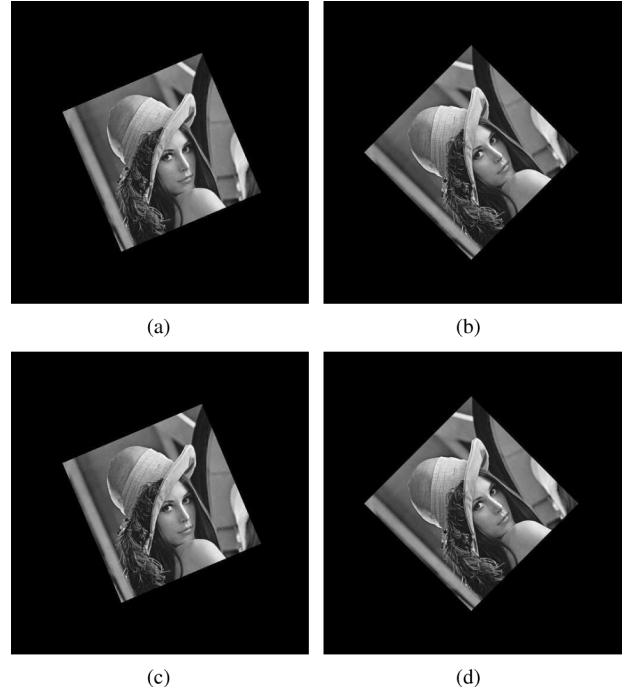


Fig. 8. Rotated Lena images. (a) 0.4 (rad) rotation by Method A, (b) $\pi/4$ (rad) rotation by Method A, (c) 0.4 (rad) rotation by Method B, (d) $\pi/4$ (rad) rotation by Method B. Method A uses the recurrence formulas (11) and (12), and Method B uses the matrix multiplication in (18).

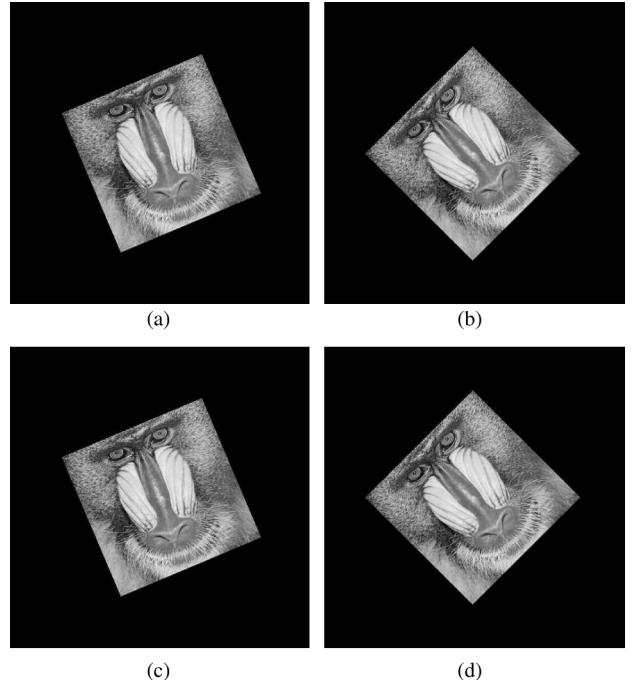


Fig. 9. Rotated baboon images. (a) 0.4 (rad) rotation by Method A, (b) $\pi/4$ (rad) rotation by Method A, (c) 0.4 (rad) rotation by Method B, (d) $\pi/4$ (rad) rotation by Method B. Method A uses the recurrence formulas (11) and (12), and Method B uses the matrix multiplication in (18).

three consecutive shears are implemented using FFT. Since Test 1 depends on the choice of the reference continuous 2-D function, this test is not a sufficient test for accuracy, even though our methods perform better in this test.

TABLE I
IMAGE DIFFERENCE MEASUREMENTS IN TEST 1 USING THE FOURIER SERIES FOR FITTING AND RESAMPLING.

	0.4(rad)	$\pi/4$ (rad)
FFT method	0.0170	0.0250
Method A	0.0132	0.0163
Method B	0.0132	0.0164

(a) NLSE with Lena image	(b) NLSE with baboon image																								
<table border="1"> <thead> <tr> <th></th> <th>0.4(rad)</th> <th>$\pi/4$(rad)</th> </tr> </thead> <tbody> <tr> <td>FFT method</td><td>4.9021</td><td>8.8305</td></tr> <tr> <td>Method A</td><td>1.5208</td><td>1.8713</td></tr> <tr> <td>Method B</td><td>1.5208</td><td>1.8713</td></tr> </tbody> </table>		0.4(rad)	$\pi/4$ (rad)	FFT method	4.9021	8.8305	Method A	1.5208	1.8713	Method B	1.5208	1.8713	<table border="1"> <thead> <tr> <th></th> <th>0.4(rad)</th> <th>$\pi/4$(rad)</th> </tr> </thead> <tbody> <tr> <td>FFT method</td><td>7.6456</td><td>12.6874</td></tr> <tr> <td>Method A</td><td>3.0052</td><td>3.5018</td></tr> <tr> <td>Method B</td><td>3.0052</td><td>3.5018</td></tr> </tbody> </table>		0.4(rad)	$\pi/4$ (rad)	FFT method	7.6456	12.6874	Method A	3.0052	3.5018	Method B	3.0052	3.5018
	0.4(rad)	$\pi/4$ (rad)																							
FFT method	4.9021	8.8305																							
Method A	1.5208	1.8713																							
Method B	1.5208	1.8713																							
	0.4(rad)	$\pi/4$ (rad)																							
FFT method	7.6456	12.6874																							
Method A	3.0052	3.5018																							
Method B	3.0052	3.5018																							
(c) Sobolev norm with Lena image	(d) Sobolev norm with baboon image																								
<table border="1"> <thead> <tr> <th></th> <th>0.4(rad)</th> <th>$\pi/4$(rad)</th> </tr> </thead> <tbody> <tr> <td>FFT method</td><td>0.9885×10^4</td><td>3.6614×10^4</td></tr> <tr> <td>Method A</td><td>0.1153×10^4</td><td>0.1729×10^4</td></tr> <tr> <td>Method B</td><td>0.1153×10^4</td><td>0.1729×10^4</td></tr> </tbody> </table>		0.4(rad)	$\pi/4$ (rad)	FFT method	0.9885×10^4	3.6614×10^4	Method A	0.1153×10^4	0.1729×10^4	Method B	0.1153×10^4	0.1729×10^4	<table border="1"> <thead> <tr> <th></th> <th>0.4(rad)</th> <th>$\pi/4$(rad)</th> </tr> </thead> <tbody> <tr> <td>FFT method</td><td>1.9264×10^4</td><td>5.5844×10^4</td></tr> <tr> <td>Method A</td><td>0.2692×10^4</td><td>0.3695×10^4</td></tr> <tr> <td>Method B</td><td>0.2692×10^4</td><td>0.3695×10^4</td></tr> </tbody> </table>		0.4(rad)	$\pi/4$ (rad)	FFT method	1.9264×10^4	5.5844×10^4	Method A	0.2692×10^4	0.3695×10^4	Method B	0.2692×10^4	0.3695×10^4
	0.4(rad)	$\pi/4$ (rad)																							
FFT method	0.9885×10^4	3.6614×10^4																							
Method A	0.1153×10^4	0.1729×10^4																							
Method B	0.1153×10^4	0.1729×10^4																							
	0.4(rad)	$\pi/4$ (rad)																							
FFT method	1.9264×10^4	5.5844×10^4																							
Method A	0.2692×10^4	0.3695×10^4																							
Method B	0.2692×10^4	0.3695×10^4																							
(e) Relative entropy with Lena image	(f) Relative entropy with baboon image																								

TABLE II
IMAGE DIFFERENCE MEASUREMENTS IN TEST 1 USING THE BI-CUBIC INTERPOLATION FOR FITTING AND RESAMPLING

	0.4(rad)	$\pi/4$ (rad)
FFT method	0.0280	0.0303
Method A	0.0270	0.0280
Method B	0.0278	0.0280

(a) NLSE with Lena image	(b) NLSE with baboon image																								
<table border="1"> <thead> <tr> <th></th> <th>0.4(rad)</th> <th>$\pi/4$(rad)</th> </tr> </thead> <tbody> <tr> <td>FFT method</td><td>5.0721</td><td>8.3898</td></tr> <tr> <td>Method A</td><td>2.9962</td><td>3.0589</td></tr> <tr> <td>Method B</td><td>2.9962</td><td>3.0589</td></tr> </tbody> </table>		0.4(rad)	$\pi/4$ (rad)	FFT method	5.0721	8.3898	Method A	2.9962	3.0589	Method B	2.9962	3.0589	<table border="1"> <thead> <tr> <th></th> <th>0.4(rad)</th> <th>$\pi/4$(rad)</th> </tr> </thead> <tbody> <tr> <td>FFT method</td><td>7.8606</td><td>11.9270</td></tr> <tr> <td>Method A</td><td>5.2014</td><td>5.3275</td></tr> <tr> <td>Method B</td><td>5.2014</td><td>5.3275</td></tr> </tbody> </table>		0.4(rad)	$\pi/4$ (rad)	FFT method	7.8606	11.9270	Method A	5.2014	5.3275	Method B	5.2014	5.3275
	0.4(rad)	$\pi/4$ (rad)																							
FFT method	5.0721	8.3898																							
Method A	2.9962	3.0589																							
Method B	2.9962	3.0589																							
	0.4(rad)	$\pi/4$ (rad)																							
FFT method	7.8606	11.9270																							
Method A	5.2014	5.3275																							
Method B	5.2014	5.3275																							
(c) Sobolev norm with Lena image	(d) Sobolev norm with baboon image																								
<table border="1"> <thead> <tr> <th></th> <th>0.4(rad)</th> <th>$\pi/4$(rad)</th> </tr> </thead> <tbody> <tr> <td>FFT method</td><td>0.7223×10^4</td><td>3.1048×10^4</td></tr> <tr> <td>Method A</td><td>0.0415×10^4</td><td>0.1948×10^4</td></tr> <tr> <td>Method B</td><td>0.0415×10^4</td><td>0.1948×10^4</td></tr> </tbody> </table>		0.4(rad)	$\pi/4$ (rad)	FFT method	0.7223×10^4	3.1048×10^4	Method A	0.0415×10^4	0.1948×10^4	Method B	0.0415×10^4	0.1948×10^4	<table border="1"> <thead> <tr> <th></th> <th>0.4(rad)</th> <th>$\pi/4$(rad)</th> </tr> </thead> <tbody> <tr> <td>FFT method</td><td>1.7012×10^4</td><td>4.7262×10^4</td></tr> <tr> <td>Method A</td><td>0.5350×10^4</td><td>0.6762×10^4</td></tr> <tr> <td>Method B</td><td>0.5350×10^4</td><td>0.6762×10^4</td></tr> </tbody> </table>		0.4(rad)	$\pi/4$ (rad)	FFT method	1.7012×10^4	4.7262×10^4	Method A	0.5350×10^4	0.6762×10^4	Method B	0.5350×10^4	0.6762×10^4
	0.4(rad)	$\pi/4$ (rad)																							
FFT method	0.7223×10^4	3.1048×10^4																							
Method A	0.0415×10^4	0.1948×10^4																							
Method B	0.0415×10^4	0.1948×10^4																							
	0.4(rad)	$\pi/4$ (rad)																							
FFT method	1.7012×10^4	4.7262×10^4																							
Method A	0.5350×10^4	0.6762×10^4																							
Method B	0.5350×10^4	0.6762×10^4																							
(e) Relative entropy with Lena image	(f) Relative entropy with baboon image																								

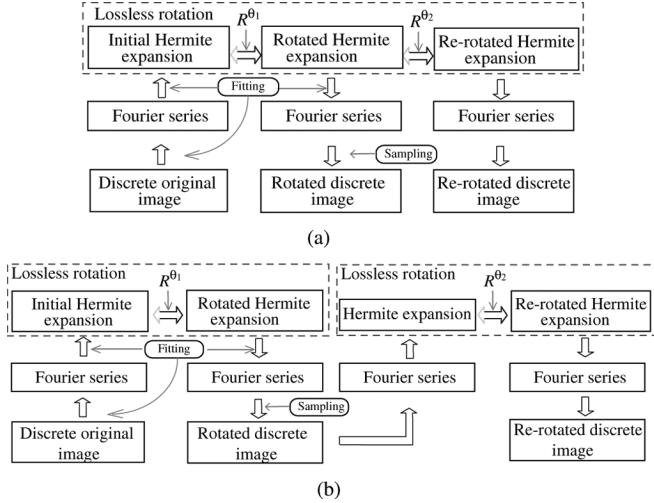


Fig. 10. Strategies for consecutive rotations. (a) Strategy 1. (b) Strategy 2.

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 square grid. The first natural test using this perfect baseline would be to 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 cannot 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: 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: Rotate an image by $\pi/6$ (rad) 12 times consecutively and compare the resulting image and the original image.

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

TABLE III
NORMALIZED LEAST SQUARED 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
(g) NLSE with Fig. 5(e)				(h) NLSE with Fig. 5(f)			

TABLE IV
SOBOLEV NORMS IN ACCURACY TESTS

	Test 2	Test 3	Test 4		Test 2	Test 3	Test 4
FFT method	3.2326	2.6526	2.2020	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	0.7930	Method A	2.1701	2.4189	1.9961
Method B	0.8160	0.9449	0.7930	Method B	2.1701	2.4189	1.9961
(g) Sobolev norms with Fig. 5(e)				(h) Sobolev norms with Fig. 5(f)			

Since these three tests require 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, we 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 se-

ries. 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 processing.

The test results for Test 2, 3, and 4 are shown in Tables III–V. 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. When we measure the image difference, we compute it over the region including zero padding. The bandlimit 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 the Lena image are 0.0017, when we use Strategy 1. This small error is possible since the error in that strategy 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

TABLE V
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)			
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)			
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)			
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
(g) Relative entropies ($\times 10^2$) with Fig. 5(e)				(h) Relative entropies ($\times 10^3$) with Fig. 5(f)			

the original Lena image and the resulting images in Test 3 are shown. With the Hermite-based method, the difference occurs mainly at the sharp edges in the image as shown Fig. 11(b). However, with the FFT-based method, the difference can be seen in the wide area around the sharp edges as shown Fig. 11(a). In addition, the Hermite-based method gives fewer artifacts in the zero-padding region.

C. Computation Time

Table VI 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 Sections IV-A and IV-B, 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 depends on the hardware and is not part of the intrinsic efficiency of the algorithm. It is in this sense that Method B shows a faster performance than Method A.

The algorithm for computing the Hermite coefficients from a discrete image is implemented with an $\mathcal{O}(N^3)$ computation, when the size of image is $N \times N$ and the bandlimit of the Hermite expansion is $\mathcal{O}(N)$. It is achieved by expressing (21) as a matrix multiplication. In principle, we could reduce the computational cost to $\mathcal{O}(N^2 \log^2 N)$ if we used the algorithm in [9], [10]. However, the rotation algorithm for the Hermite expansion in Section III 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 a 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 $m = 0, 1, 2, \dots, N$, the complexity of the whole rotation process is $\mathcal{O}(N^3)$. Method B using (18) also has $\mathcal{O}(N^3)$ complexity, since computation of $S^m(\theta)$ using (18) needs $\mathcal{O}(m^2)$ calculation.

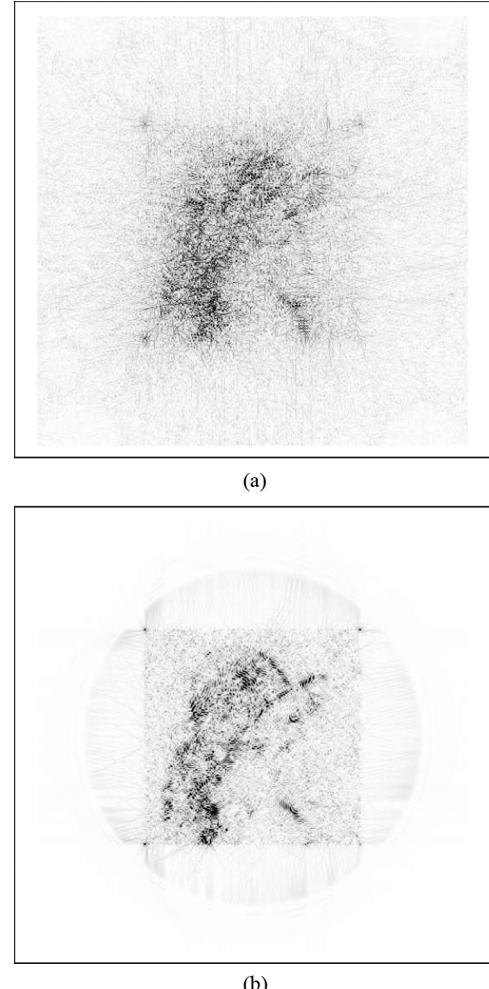


Fig. 11. Inverted difference images between the original Lena image and the resulting images in Test 3 ($\times 30$ amplified). (a) Difference image obtained using FFT-based method. (b) Difference image obtained using Hermite-based method.

TABLE VI
ELAPSED TIME IN SECONDS FOR THE IMAGE ROTATION PROCESS

Image size (bandlimit)	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

VI. CONCLUSION

In this work, in order to rotate images, we used the fact that the rotation of a 2-D bandlimited Hermite expansion (with a bandlimit of a special form) is of the same form and has the same bandlimit as the original expansion. We observed that the rotation of a 2-D bandlimited 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 bandlimited 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 presented the example results. We also designed the tests to assess the accuracy of rotation methods. We showed that the accuracy of our image rotation methods is better than that of the existing image rotation technique using FFT and three 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 VI. 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. We believe that this is one reason for the better accuracy of our methods.

APPENDIX A ORTHOGONALITY OF $S^m(\theta)$

Let us consider the following:

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

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

$$E_m = \sum_{n=0}^m \hat{f}_{n,m-n} \hat{g}_{n,m-n} \quad (22)$$

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

$$\begin{aligned} E_m &= \int_{\mathbb{R}^2} \left(\sum_{n=0}^m \hat{f}_{n,m-n}^{\theta} h_n(x) h_{m-n}(y) \right) \\ &\quad \times \left(\sum_{k=0}^m \hat{g}_{k,m-k}^{\theta} h_k(x) h_{m-k}(y) \right) dx dy \\ &= \sum_{n=0}^m \hat{f}_{n,m-n}^{\theta} \hat{g}_{n,m-n}^{\theta}. \end{aligned}$$

Using (6), we can rewrite this as

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

Therefore

$$E_m = \sum_{p=0}^m \sum_{q=0}^m \left(\sum_{n=0}^m S_{n,p}^m(\theta) S_{n,q}^m(\theta) \right) \hat{f}_{p,m-p} \hat{g}_{q,m-q}. \quad (23)$$

Equating (22) and (23), we have

$$\begin{aligned} \sum_{n=0}^m \hat{f}_{n,m-n} \hat{g}_{n,m-n} &= \sum_{p=0}^m \sum_{q=0}^m \hat{f}_{p,m-p} \left(\sum_{n=0}^m S_{n,p}^m(\theta) S_{n,q}^m(\theta) \right) \hat{g}_{q,m-q} \end{aligned}$$

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

$$\{S^m(\theta)\}^T S^m(\theta) = I_m \quad (24)$$

where I_m is the $(m+1) \times (m+1)$ identity matrix. Since $S^m(\theta)$ is a real-valued square matrix and (24) holds, $S^m(\theta)$ is an orthogonal matrix.

APPENDIX B SKEW-SYMMETRIC MATRIX $\log(S^m(\theta))$

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

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

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

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

then $\Omega^m = A(0) + B(0)$. Using the derivative of Hermite function as [3]

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

and (10), we have

$$\begin{aligned} \Omega_{q,n}^m &= \frac{d}{d\theta} (S_{q,n}^m(\theta)) |_{\theta=0} \\ &= A_{q,n}(0) + B_{q,n}(0) \\ &= \sqrt{n} \sqrt{m-n+1} \delta_{q,n-1} \\ &\quad - \sqrt{n+1} \sqrt{m-n} \delta_{q,n+1}. \end{aligned}$$

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

APPENDIX C CONNECTION BETWEEN FOURIER SERIES AND THE DFT

Here, we find the Fourier series interpolating a given sampling. For convenience, we assume that the number of data points, 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 $\theta_j = 2\pi j/N$, $j = 0, 1, \dots, (N-1)$. We can expand $f(\theta_j)$ as

$$\begin{aligned} f(\theta_j) &= \tilde{f}_0 e^{i \cdot 0 \cdot 2\pi j/N} + \tilde{f}_1 e^{i \cdot 1 \cdot 2\pi j/N} \\ &\quad + \dots + \tilde{f}_B e^{i \cdot B \cdot 2\pi j/N} + \tilde{f}_{-1} e^{i \cdot -1 \cdot 2\pi j/N} \\ &\quad + \dots + \tilde{f}_{-B} e^{i \cdot (-B) \cdot 2\pi j/N}. \end{aligned} \quad (25)$$

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

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

where $\hat{f}_n = \sum_{j=0}^{N-1} f_j e^{-2\pi/Nijn}$. Since we assume N is even, we have the following relationship between the DFT and Fourier series:

$$\begin{aligned} Nf(\theta_j) &= \hat{f}_0 e^{i \cdot 0 \cdot 2\pi j/N} + \hat{f}_1 e^{i \cdot 1 \cdot 2\pi j/N} \\ &\quad + \dots + \hat{f}_{(N/2-1)} e^{i \cdot (N/2-1) \cdot 2\pi j/N} \\ &\quad + \hat{f}_{N/2} e^{i \cdot (N/2) \cdot 2\pi j/N} \\ &\quad + \hat{f}_{(N/2+1)} e^{i \cdot (N/2+1) \cdot 2\pi j/N} \\ &\quad + \dots + \hat{f}_{(N-1)} e^{i \cdot (N-1) \cdot 2\pi j/N}. \end{aligned} \quad (26)$$

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

$$\begin{aligned} &\hat{f}_{(N/2+1)} e^{i \cdot (N/2+1) \cdot 2\pi j/N} \\ &\quad + \dots + \hat{f}_{(N-1)} e^{i \cdot (N-1) \cdot 2\pi j/N} \\ &= \hat{f}_{(N/2+1)} e^{i \cdot -(N/2-1) \cdot 2\pi j/N} \\ &\quad + \dots + \hat{f}_{(N-1)} e^{i \cdot (-1) \cdot 2\pi j/N} \end{aligned}$$

we can rewrite (26) as

$$\begin{aligned} f(\theta_j) &= \frac{1}{N} \left(\hat{f}_0 e^{i \cdot 0 \cdot 2\pi j/N} + \hat{f}_1 e^{i \cdot 1 \cdot 2\pi j/N} \right. \\ &\quad + \dots + \hat{f}_{((N)/2-1)} e^{i \cdot ((N/2)-1) \cdot 2\pi j/N} \\ &\quad + \frac{1}{2} \hat{f}_{N/2} e^{i \cdot (N/2) \cdot 2\pi j/N} \\ &\quad + \hat{f}_{(N-1)} e^{i \cdot (-1) \cdot 2\pi j/N} \\ &\quad + \dots + \hat{f}_{(N/2+1)} e^{i \cdot -(N/2-1) \cdot 2\pi j/N} \\ &\quad \left. + \frac{1}{2} \hat{f}_{N/2} e^{i \cdot -(N/2) \cdot 2\pi j/N} \right). \end{aligned} \quad (27)$$

Equating (25) and (27), we can write the relationship between the DFT and the Fourier series as follows:

$$B = \frac{N}{2}$$

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

for an even number, N .

REFERENCES

- [1] İ. Avcıbas, B. Sankur, and K. Sayood, "Statistical evaluation of image quality measures," *J. Electron. Imag.*, vol. 11, no. 2, pp. 206–223, 2002.
- [2] B. Chen and A. Kaufman, "3D volume rotation using shear transformations," *Graph. Models.*, vol. 62, no. 4, pp. 308–322, 2000.
- [3] G. S. Chirikjian and A. B. Kyatkin, *Engineering Applications of Non-commutative Harmonic Analysis*. Boca Raton, FL: CRC, 2001.
- [4] R. W. Cox and R. Tong, "Two- and three-dimensional image rotation using the FFT," *IEEE Trans. Image Process.*, vol. 8, pp. 1297–1299, 1999.
- [5] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 891–906, Aug. 1991.
- [6] M. Jacob and M. Unser, "Design of steerable filters for feature detection using canny-like criteria," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1007–1019, Aug. 2004.
- [7] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.
- [8] K. G. Larkin, M. A. Oldfield, and H. Klemm, "Fast Fourier method for the accurate rotation of sampled images," *Opt. Commun.*, vol. 139, pp. 99–106, 1997.
- [9] G. Leibon, D. Rockmore, and G. S. Chirikjian, "A fast Hermite transform with applications to protein structure determination," in *Proc. Int. Workshop Symbolic-Numeric Computation*, 2007, pp. 117–124.
- [10] G. Leibon, D. Rockmore, W. Park, R. Taintor, and G. S. Chirikjian, "A fast Hermite transform," *Theoretical Comput. Sci.*, vol. 409, pp. 211–228, 2008.
- [11] Z. Liu and R. Laganiere, "Phase congruence measurement for image similarity assessment," *Pattern Recognit. Lett.*, vol. 28, pp. 166–172, 2007.
- [12] J. B. Martens, "Local orientation analysis in images by means of the Hermite transform," *IEEE Trans. Image Process.*, vol. 6, pp. 1103–1116, 1997.
- [13] J. B. Martens, "The Hermite transform: A survey," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 1–20, 2006.
- [14] M. Michaelis and G. Sommer, "A LIE group approach to steerable filters," *Pattern Recognit. Lett.*, vol. 16, no. 11, pp. 1165–1174, 1995.
- [15] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years late," *SIAM Rev.*, vol. 45, pp. 3–49, 2003.
- [16] M. Najafi, A. Krylov, and D. Kortchagine, "Image deblocking with 2D Hermite transform," in *Proc. Int. Conf. Graphicon*, 2003, pp. 180–183.
- [17] F. Natterer, *The Mathematics of Computerized Tomography*. New York: Wiley, 1986.
- [18] A. F. Nikiforov, S. K. Suslov, and V. B. Uvarov, *Classical Orthogonal Polynomials of a Discrete Variable*. Springer Series in Computational Physics. Berlin, Germany: Springer-Verlag, 1991.
- [19] C. B. Owen and F. Makedon, "High quality alias free image rotation," in *Proc. 30th Asilomar Conf. Signals, Systems, and Computers Pacific Grove*, 1996, pp. 115–119.
- [20] A. Paeth, "A fast algorithm for general raster rotation," in *Proc. Graphics Interface*, 1986, pp. 77–81.
- [21] W. Park, "Inverse Problems in Structural Biology and Flexible Needle Steering," Ph.D. dissertation, Johns Hopkins Univ., Baltimore, MD, 2008.
- [22] W. Park and G. S. Chirikjian, "Interconversion between truncated cartesian and polar expansions of images," *IEEE Trans. Image Process.*, vol. 16, pp. 1946–1955, 2007.
- [23] T. Toffoli and J. Quick, "Three-dimensional rotations by three shears," *Graph. Models Image Process.*, vol. 59, no. 2, pp. 89–95, 1997.

- [24] M. Unser, P. Thevenaz, and L. P. Yaroslavsky, "Convolution-based interpolation for fast, high-quality rotation of images," *IEEE Trans. Image Process.*, vol. 4, pp. 1371–1381, 1995.
- [25] A. M. van Dijk and J. B. Martens, "Feature-based image compression with steered Hermite transforms," in *Proc. IEEE Int. Conf. Image Processing*, 1996, vol. 1, pp. 205–208.
- [26] A. M. van Dijk and J. B. Martens, "Image representation and compression with steered Hermite transforms," *Signal Process.*, vol. 56, pp. 1–16, 1996.
- [27] N. J. Vilenkin and A. U. Klimyk, *Representation of Lie Groups and Special Functions*. Dordrecht, The Netherlands: Kluwer, 1991, vol. 1–3.
- [28] D. L. Wilson, A. J. Baddeley, and R. A. Owens, "A new metric for grey-scale image comparison," *Int. J. Comput. Vis.*, vol. 24, no. 1, pp. 5–17, 1997.
- [29] W. Yu, K. Daniilidis, and G. Sommer, "Approximate orientation steerability based on angular Gaussians," *IEEE Trans. Image Process.*, vol. 10, pp. 193–205, 2001.



Wooram Park (S'06) received the B.S.E. and M.S.E. degrees in mechanical engineering from Seoul National University, Korea, in 1999 and 2003, respectively, and the Ph.D. degree in mechanical engineering from Johns Hopkins University, Baltimore, MD, in 2008.

He is currently a postdoctoral Fellow at the Department of Mechanical Engineering, Johns Hopkins University. His research interests include image modeling, image processing, and medical robotics.



Gregory Leibon received the Ph.D. from the Mathematics Department, University of California at San Diego, La Jolla, in 1999.

He currently splits his time between being the Chief Mathematician at Memento and a Research Associate in the Department of Mathematics, Dartmouth College, Hanover, NH. As a Chief Mathematician, he has brought the tools of statistical learning to the problem of detecting fraudulent and collusive behavior in various banking systems. As a Research Associate, he works on various applied mathematics problems, including problems from the mathematics of gene discovery, image processing, statistical learning, and complex systems.

Daniel N. Rockmore received the B.A. degree in mathematics from Princeton University, Princeton, NJ, in 1984, and the M.S. and Ph.D. degrees in mathematics from Harvard University, Cambridge, MA, in 1986 and 1989, respectively.

He is currently a Professor of computer science and John G. Kemeny Parents Professor of Mathematics at Dartmouth College, Hanover, NH, where is also Chairman of the Department of Mathematics. He is also on the External Faculty of the Santa Fe Institute. His research interests are in computational harmonic analysis and complex systems.

Dr. Rockmore was a 1995 recipient of an NSF Presidential Faculty Fellowship, 2005–2007 Sigma Distinguished Lecturer, and the 2008 SIAM I. E. Block Community Lecturer.



Gregory S. Chirikjian (M'93) received the B.S.E. degree in engineering mechanics, the M.S.E. degree in mechanical engineering, and the B.A. degree in mathematics, all from The Johns Hopkins University, Baltimore, MD, in 1988, and the Ph.D. degree from the California Institute of Technology, Pasadena, CA, in 1992.

Since the summer of 1992, he has been with the Department of Mechanical Engineering, Johns Hopkins University, where he is now Professor and Chair. His research interests include mathematical aspects of robotics and imaging. In recent years, he has also been applying methods from robotics to model conformational transitions in biological macromolecules.

Dr. Chirikjian is a 1993 NSF Young investigator, a 1994 Presidential Faculty Fellow, and a 1996 recipient of the ASME Pi Tau Sigma Gold Medal.