The Angular Difference Function and Its Application to Image Registration

Yosi Keller, *Member*, *IEEE*, Yoel Shkolnisky, and Amir Averbuch

Abstract—The estimation of large motions without prior knowledge is an important problem in image registration. In this paper, we present the angular difference function (ADF) and demonstrate its applicability to rotation estimation. The ADF of two functions is defined as the integral of their spectral difference along the radial direction. It is efficiently computed using the pseudopolar Fourier transform, which computes the discrete Fourier transform of an image on a near spherical grid. Unlike other Fourier-based registration schemes, the suggested approach does not require any interpolation. Thus, it is more accurate and significantly faster.

Index Terms—Global motion estimation, Fourier domain, pseudopolar FFT, image alignment.

1 Introduction

Image registration plays a major role in many image processing applications such as video compression [1], [2], video enhancement [3], and scene representation [4], [5], [6]. The main approaches that are being used to study this problem are pixel domain Gradient methods [1], [4], [5], spatial correlation techniques [7], and discrete Fourier transform (DFT)-based algorithms [8], [9]. Gradient methods are considered as the state-of-the-art. However, they work well only when applied to images that have small misalignment. In [7], the image is interpolated on a log-polar grid and the rotations and scalings are reduced to translations recovered by spatial correlation. As this representation is not translation invariant, the interpolation is recomputed for each possible translation, resulting in high computational complexity.

Fourier-based schemes, which are able to estimate large rotations, scalings, and translations, are rather crude and often used to bootstrap the more accurate gradient methods. Most of the DFT-based approaches utilize the *shift property* [10] of the Fourier transform, which enables robust estimation of translations using *normalized phase-correlation* [8], [11], [12], [13], [14]. To handle rotations and scalings, the image is transformed into a uniform polar or log-polar Fourier representation, where rotations and scalings are reduced to translations. The rotations and scalings are then estimated by using phase-correlation.

In this paper, we present the angular difference function (ADF) and demonstrate its applicability to estimation of relative rotation of images. Generally, given two images I_1 and I_2, we compute the magnitude of the pseudopolar Fourier transform (PPFT) [15] of each of the images, denoted M_1 and M_2. We then compute the absolute value of the difference of M_1 and M_2 and use this difference to numerically compute the ADF of I_1 and I_2. The relative rotation of I_1 and I_2 induces a cross pattern on the Fourier transform of $I_1 - I_2$. This cross pattern is robustly and accurately detected using the ADF.

A notion similar to the ADF was first introduced in [16], [17], [18], [19] for rotation estimation and symmetry detection. The main idea of [16] and [17] is to use the multiscale Hough transform to detect the aforementioned cross patterns. This method incurs inaccuracies as it necessarily requires some sort of interpolation.

The Fourier domain-based schemes mentioned above [9], [16], [17], use a uniform polar FFT representation. There is no fast and accurate algorithm for the computation of the polar FFT. Thus, such schemes use interpolation in the Fourier magnitude domain. Due to the oscillatory nature of the Fourier magnitude, its interpolation results in significant errors. The proposed method does not require a uniform polar representation and uses the Fourier computed on the pseudopolar grid. This computation is algebraically accurate and fast ($O(n^2 \log n)$).

The paper is organized as follows: In Section 2, we give the mathematical background relevant to Fourier-based image registration. In Section 3, we describe the pseudopolar Fourier transform. In Section 4, we derive a 1D shift estimation algorithm, which is based on difference functions. This algorithm is a simplified 1D implementation of the ideas presented in Section 5, where we define the ADF for the 2D case and derive a fast and accurate algorithm for its computation. We then use this algorithm in Section 6 for rotation estimation. We conclude the paper with experimental results and some concluding remarks in Sections 6 and 7, respectively.

2 Mathematical Preliminaries

2.1 Translation Estimation

We denote the Fourier transform of the function $f(x,y)$ by $\mathcal{F}\{f(x,y)\}$ or $\widehat{f}(\omega_x,\omega_y)$. The *shift property* of the Fourier transform [10] is then given by

$$\mathcal{F}\{f(x+\Delta x, y+\Delta y)\} = \widehat{f}(\omega_x,\omega_y)e^{\imath(\omega_x\Delta x+\omega_y\Delta y)}. \qquad (2.1)$$

- *Y. Keller is with the Department of Mathematics, Yale University, PO Box 208283, New Haven, CT 06520. E-mail: yosi.keller@yale.edu.*
- *Y. Shkolnisky and A. Averbuch are with the Department of Computer Science, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: {yoel, amir}@math.tau.ac.il.*

Manuscript received 25 May 2004; revised 22 Dec. 2004; accepted 29 Dec. 2004; published online 14 Apr. 2005.
Recommended for acceptance by T. Tan.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0260-0504.

Equation (2.1) is the basis of the Fourier-based translation estimation algorithms [8], [14]. Assume that $I_1(x, y)$ and $I_2(x, y)$ are two images that satisfy

$$I_1(x + \Delta x, y + \Delta y) = I_2(x, y). \tag{2.2}$$

By applying the Fourier transform on both sides of (2.2), we get

$$\widehat{I}_1\big(\omega_x, \omega_y\big)e^{\imath\left(\omega_x \Delta x + \omega_y \Delta y\right)} = \widehat{I}_2\big(\omega_x, \omega_y\big) \tag{2.3}$$

or, equivalently,

$$Corr\big(\omega_x, \omega_y\big) \triangleq \frac{\widehat{I}_2\big(\omega_x, \omega_y\big)}{\widehat{I}_1\big(\omega_x, \omega_y\big)} = e^{\imath\left(\omega_x \Delta x + \omega_y \Delta y\right)}. \tag{2.4}$$

Taking the inverse Fourier transform of both sides of (2.4), we get

$$\begin{aligned} \mathrm{Corr}(x, y) &\triangleq \mathcal{F}^{-1}\big(Corr\big(\omega_x, \omega_y\big)\big) \\ &= \delta(x + \Delta x, y + \Delta y), \end{aligned} \tag{2.5}$$

which means that $\mathrm{Corr}(x, y)$ is nonzero only at $(\Delta x, \Delta y)$. For discrete images, we replace the Fourier transform in the computations above with the FFT, and $\delta(x + \Delta x, y + \Delta y)$ is replaced by a function that has a dominant maximum at $(\Delta x, \Delta y)$. We then find $(\Delta x, \Delta y)$ as

$$(\Delta x, \Delta y) = \arg \max_{(x,y)} \{\mathrm{Corr(x, y)}\}. \tag{2.6}$$

We make the procedure above more robust by using normalized phase correlation $\widetilde{\mathrm{Corr}}(\omega_x, \omega_y)$ [8], [20]. Formally,

$$\begin{aligned} \widetilde{\mathrm{Corr}}\big(\omega_x, \omega_y\big) &\triangleq \frac{\widehat{I}_2\big(\omega_x, \omega_y\big)}{\widehat{I}_1\big(\omega_x, \omega_y\big)} \frac{\left|\widehat{I}_1\big(\omega_x, \omega_y\big)\right|}{\left|\widehat{I}_2\big(\omega_x, \omega_y\big)\right|} \\ &= \frac{\widehat{I}_2\big(\omega_x, \omega_y\big)\widehat{I}_1^*\big(\omega_x, \omega_y\big)}{\left|\widehat{I}_2\big(\omega_x, \omega_y\big)\right|\left|\widehat{I}_1^*\big(\omega_x, \omega_y\big)\right|} = e^{\imath\left(\omega_x \Delta x + \omega_y \Delta y\right)}, \end{aligned} \tag{2.7}$$

where $*$ denotes the complex conjugate and $\widetilde{\mathrm{Corr}}\big(\omega_x, \omega_y\big)$ replaces $Corr\big(\omega_x, \omega_y\big)$ to (2.5). This scheme was proven to robustly estimate large translations [14]. There are no smoothness assumptions on the registered functions and, therefore, nonsmooth and noisy functions (such as the 2D DFT coefficients) can be accurately registered. Foroosh et al. [20] extended the phase-correlation-based algorithm to subpixel accuracy by analyzing the shape of $\widetilde{\mathrm{Corr}}\big(\omega_x, \omega_y\big)$, given in (2.7), around its maximum.

A different approach for phase correlation-based translation estimation is given in [13], [14], [21]. From (2.7), we get that Δx and Δy satisfy the equation

$$-\imath \log\Big(\widetilde{\mathrm{Corr}}\big(\omega_x, \omega_y\big)\Big) = \omega_x \Delta x + \omega_y \Delta y,$$

which can be solved using linear regression. This approach may be inaccurate ([20]) due to aliasing and phase wrapping of the spectra around 2π. To solve these problems, [13] suggests an iterative solution to phase unwrapping and [14] presents two approaches for modeling aliasing effects and improving registration accuracy.

2.2 Polar Fourier Representations

The polar Fourier representation (Fourier-Mellin transform) is used to register images that have both translational and rotational misalignments [9], [22]. Let I_1 and I_2 be two images such that

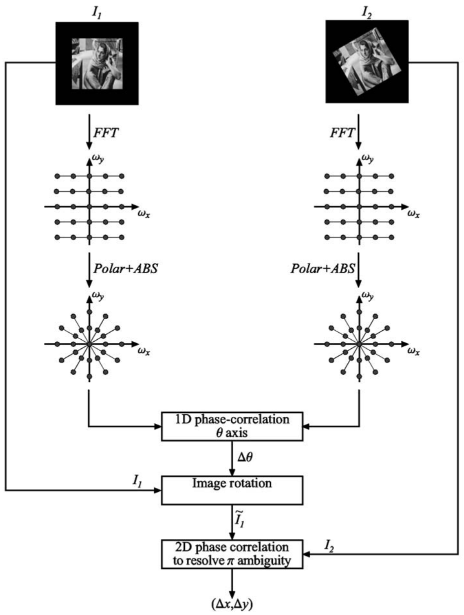


Fig. 1. The flow of a FFT-based image registration: 1) The magnitudes of the polar DFTs are approximated by interpolating the magnitudes of the 2D FFTs. 2) The rotation angle $\Delta\theta$ is recovered using 1D phase-correlation on the θ axis. 3) One of the input images is rotated by the angle $\Delta\theta$. 4) The translation is recovered and the θ ambiguity is resolved by applying a 2D phase-correlation twice: once on θ and once on $\theta + \pi$.

$$\begin{aligned} &I_2(x, y) = \\ &I_1(x \cos\theta_0 + y \sin\theta_0 + \Delta x, -x \sin\theta_0 + y \cos\theta_0 + \Delta y), \end{aligned} \tag{2.8}$$

where θ_0 and $(\Delta x, \Delta y)$ are the relative rotation and translation of I_1 and I_2, respectively. The Fourier Transform of (2.8) in polar coordinates is

$$\widehat{I}_2(r, \theta) = e^{\imath\left(\omega_x \Delta x + \omega_y \Delta y\right)}\widehat{I}_1(r, \theta + \theta_0). \tag{2.9}$$

If we denote by M_1 and M_2 the magnitudes of $\widehat{I}_1$ and $\widehat{I}_2$, i.e.,

$$M_1 = |\widehat{I}_1|, \qquad M_2 = |\widehat{I}_2|, \tag{2.10}$$

then we get that M_1 and M_2 are related by

$$M_1(r, \theta) = M_2(r, \theta + \theta_0). \tag{2.11}$$

Equation (2.11) states that we can recover the relative rotation of I_1 and I_2 regardless of their relative translation. Using a polar Fourier transform, rotations are reduced to translations, which can be recovered by phase-correlation techniques. Using (2.11) to estimate the rotation angle θ_0 results in an ambiguity of π [9]. We resolve this ambiguity by rotating I_2 by the two possible angles θ and $\theta + \pi$, and then recovering the relative translation $(\Delta x, \Delta y)$ for each angle. We then compute the correlation peak for each of the angles and choose the rotation and translation parameters that correspond to the highest correlation peak. The flow of the algorithm is given in Fig. 1.

Two common methods for evaluating the polar Fourier transform are image domain warping [7] followed by a 2D FFT [19], and interpolation of the 2D DFT of the image in the Fourier domain [9], [22]. Since the resampling of Cartesian frequency values on a polar grid is very sensitive to interpolation, the accuracy of motion estimation algorithms is severely degraded by the approximation errors inherited in the computation of the polar and log-polar DFT. In this paper, we present an approach that does not suffer from such approximation errors.

3 The Pseudopolar Fourier Transform

Given an image I of size $N \times N$, its 2D Fourier transform, denoted $\widehat{I}(\omega_x, \omega_y)$, is given by

$$\widehat{I}(\omega_x, \omega_y) = \sum_{u,v=-N/2}^{N/2-1} I(u,v) e^{-\frac{2\pi\imath}{M}(u\omega_x + v\omega_y)}, \quad \omega_x, \omega_y \in \mathbb{R}. \tag{3.1}$$

We assume for simplicity that the image I has equal dimensions in the x and y directions and that N is even. For ω_x and ω_y sampled on the Cartesian grid $(\omega_x, \omega_y) = (k, l)$, $k, l = -M/2, \ldots, M/2 - 1$, $M = 2N + 1$, the Fourier transform in (3.1) has the form

$$\widehat{I}_{Cart}(k, l) \triangleq \widehat{I}(k, l) = \sum_{u,v=-N/2}^{N/2-1} I(u,v) e^{-\frac{2\pi\imath}{M}(uk + vl)}, \tag{3.2}$$

$k, l = -\frac{M}{2}, \ldots, \frac{M}{2} - 1$, which is usually referred to as the 2D DFT of the image I. The parameter M $(M > N)$ sets the frequency resolution of the DFT. It is well-known that the DFT of I, given by (3.2), can be computed in $O(M^2 \log M)$ operations.

For some applications, it is desirable to compute the Fourier transform of I on a polar grid. Formally, we want to sample the Fourier transform in (3.1) on the grid

$$\begin{gathered} \omega_x = r_k \cos\theta_l, \quad \omega_y = r_k \sin\theta_l, \\ r_k = k, \quad \theta_l = 2\pi l / L, \\ k = 0, \ldots, M-1, \quad l = 0, \ldots, L-1, \end{gathered} \tag{3.3}$$

where M and L are the numbers of samples on the redial and angular axes, respectively. For such a grid, the Fourier transform in (3.1) has the form

$$\begin{aligned} \widehat{I}_{polar}(k, l) &\triangleq \widehat{I}(r_k \cos\theta_l, r_k \sin\theta_l) \\ &= \sum_{u,v=-N/2}^{N/2-1} I(u,v) e^{-\frac{2\pi\imath k}{M}(u \cos\theta_l + v \sin\theta_l)}. \end{aligned} \tag{3.4}$$

The grid that is given by (3.3) is equally spaced both in the radial and angular directions

$$\Delta r_p(k) \triangleq r_{k+1} - r_k = 1 \tag{3.5}$$

$$\Delta\theta_p(l) \triangleq \theta_{l+1} - \theta_l = \frac{2\pi}{L}. \tag{3.6}$$

Unfortunately, there is no fast algorithm for computing the Fourier transform of the image I in polar coordinates, as images are given on Cartesian grids.

In Section 3.1, we give the definition of the pseudopolar Fourier transform (PPFT). In Section 3.2, we present the fractional Fourier transform, which is the primary numerical tool used in the PPFT algorithm. We conclude the presentation of the PPFT in Section 3.3 by describing an efficient algorithm for computing the PPFT.

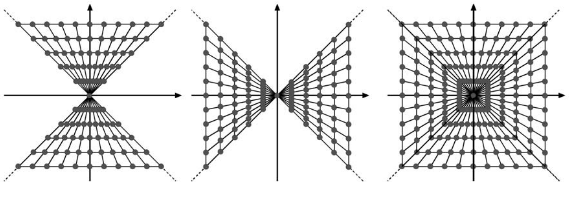


Fig. 2. The pseudopolar grid. (a) and (b) are the pseudopolar sectors P_1 and P_2, respectively. (c) The pseudopolar grid $P = P_1 \cup P_2$.

3.1 Definition of the Pseudopolar Fourier Transform

The pseudopolar Fourier transform (PPFT) evaluates the 2D Fourier transform of an image on the pseudopolar grid, which is an approximation to the polar grid. Formally, the pseudopolar grid is given by the set of samples

$$P \triangleq P_1 \cup P_2, \tag{3.7}$$

where

$$P_1 \triangleq \left\{ \left(-\frac{2l}{N}k, k \right) \mid -\frac{N}{2} \le l \le \frac{N}{2}, \ -N \le k \le N \right\} \tag{3.8}$$

$$P_2 \triangleq \left\{ \left(k, -\frac{2l}{N}k \right) \mid -\frac{N}{2} \le l \le \frac{N}{2}, \ -N \le k \le N \right\}. \tag{3.9}$$

See Figs. 2a and 2b for an illustration of the sets P_1 and P_2. The pseudopolar grid P is illustrated in Fig. 2c. As can be seen from Figs. 2a and 2b, k serves as a "pseudoradius" and l serves as a "pseudoangle." The resolution of the pseudopolar grid is $N + 1$ in the angular direction and $M = 2N + 1$ in the radial direction. Using (r, θ) representation, the pseudopolar grid is given by

$$P_1(k, l) = (r_k^1, \theta_l^1), \quad P_2(k, l) = (r_k^2, \theta_l^2), \tag{3.10}$$

$$r_k^1 = k\sqrt{4\left(\frac{l}{N}\right)^2 + 1}, \quad r_k^2 = k\sqrt{4\left(\frac{l}{N}\right)^2 + 1}, \tag{3.11}$$

$$\theta_l^1 = \pi/2 - \arctan\left(\frac{2l}{N}\right), \quad \theta_l^2 = \arctan\left(\frac{2l}{N}\right), \tag{3.12}$$

where $k = -N, \ldots, N$ and $l = -N/2, \ldots, N/2$. We define the pseudopolar Fourier transform as the samples of the Fourier transform $\widehat{I}$, given in (3.1), on the pseudopolar grid P, given in (3.7). Formally, the pseudopolar Fourier transform $\widehat{I}_{PP}^j$ $(j = 1, 2)$ is a linear transformation, which is defined for $k = -N, \ldots, N$ and $l = -N/2, \ldots, N/2$, as

$$\begin{aligned} \widehat{I}_{PP}^1(k, l) &\triangleq \widehat{I}\left(-\frac{2l}{N}k, k \right) \\ &= \sum_{u,v=-N/2}^{N/2-1} I(u,v) e^{-\frac{2\pi\imath}{M}(-\frac{2l}{N}ku + kv)}, \end{aligned} \tag{3.13}$$

$$\begin{aligned} \widehat{I}_{PP}^2(k, l) &\triangleq \widehat{I}\left(k, -\frac{2l}{N}k \right) \\ &= \sum_{u,v=-N/2}^{N/2-1} I(u,v) e^{-\frac{2\pi\imath}{M}(ku - \frac{2l}{N}kv)}, \end{aligned} \tag{3.14}$$

where $\widehat{I}$ is given by (3.1).

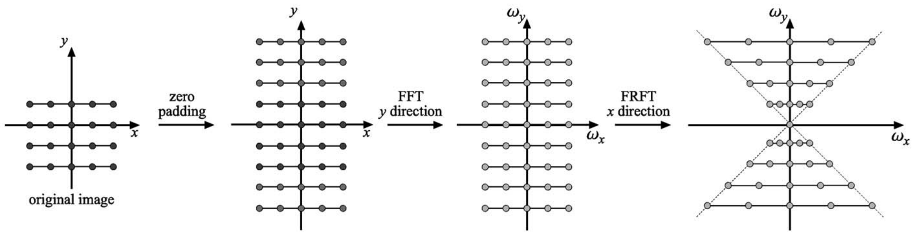


Fig. 3. Computing I^1_{PP}.

As we can see in Fig. 2c, for each fixed angle l, the samples of the pseudopolar grid are equally spaced in the radial direction. However, this spacing is different for different angles. Also, the grid is not equally spaced in the angular direction, but has equally spaced slopes. Formally,

$$\Delta \tan \theta^1_{pp}(l) \triangleq \cot \theta^1_{l+1} - \cot \theta^1_l = \frac{2}{N}, \tag{3.15}$$

$$\Delta \tan \theta^2_{pp}(l) \triangleq \tan \theta^2_{l+1} - \tan \theta^2_l = \frac{2}{N}, \tag{3.16}$$

where θ^1_l and θ^2_l are given in (3.12).

Two important properties of the pseudopolar Fourier transform are that it is invertible and that both the forward and inverse pseudopolar Fourier transforms can be implemented using fast algorithms. Moreover, their implementations require only the application of 1D equispaced FFTs. In particular, the algorithms do not require regridding or interpolation.

3.2 The Fractional Fourier Transform

The algorithm for computing the pseudopolar Fourier transform is based on the fractional Fourier transform (FRFT). The fractional Fourier transform [23], with its generalization given by the Chirp Z-transform [24], is a fast $O(N \log N)$ algorithm that evaluates the Fourier transform of a sequence X on any equally spaced set of N points on the unit circle. Specifically, given a vector X of length N, $X = (X(j),\ j = -N/2, \ldots, N/2 - 1)$, and an arbitrary $\alpha \in \mathbb{R}$, the fractional Fourier transform is defined as

$$(F^\alpha X)(l) = \sum_{u=-N/2}^{N/2-1} X(u) e^{-2\pi\imath\alpha l u/N}, \quad l = -N/2, \ldots, N/2. \tag{3.17}$$

The fractional Fourier transform samples the spectrum of X at the frequencies

$$\omega_k = \alpha l/N, \qquad l = -N/2, \ldots, N/2. \tag{3.18}$$

The fractional Fourier transform of a given vector X of length N can be computed in $O(N \log N)$ operations for any $\alpha \in \mathbb{R}$.

3.3 Computing the Pseudopolar Fourier Transform

By using the fractional Fourier transform we compute the pseudopolar Fourier transform $\widehat{I}^1_{PP}$, given in (3.13) as follows:

1. Zero pad the image I to size $N \times (2N+1)$ (along the y direction).
2. Apply the 1D Fourier transform to each column of I (along the y direction).
3. Apply the fractional Fourier transform to each row (in the x direction) with $\alpha = 2k/N$, where k is the index of the row.

This procedure is illustrated in Fig. 3. The algorithm for computing $\widehat{I}^2_{PP}$ is similar. The complexity of computing $\widehat{I}^1_{PP}$ for an $N \times N$ image is $O(N^2 \log N)$. Since the complexity of computing $\widehat{I}^2_{PP}$ is also $O(N^2 \log N)$, the total complexity of computing the pseudopolar Fourier transform is $O(N^2 \log N)$.

4 Difference Functions

We begin the derivation of 2D difference functions (DF) with a 1D example. Difference functions enable us to derive a naive algorithm for 1D shift estimation. Let $f_1(x)$ and $f_2(x)$, $x \in [0, N]$, be two shifted versions of the same function. Specifically, $f_1(x) = f_2(x + \Delta x)$ (see Fig. 4a) and denote by $g_2(x)$ the flipped and shifted version of $f_2(x)$ (see Fig. 4b)

$$g_2(x) = f_2(-x + N). \tag{4.1}$$

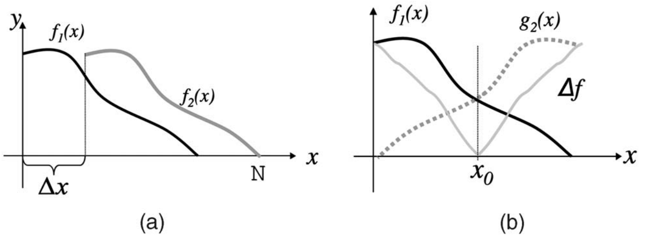
We define the difference function (DF) Δf by

$$\begin{aligned} \Delta f(x) &= f_1(x) - g_2(x) \\ &= f_1(x) - f_2(-x + N) \\ &= f_2(x + \Delta x) - f_2(-x + N) \end{aligned} \tag{4.2}$$

and consider its zeros $\Delta f(x) = 0$. One of its zeros necessarily satisfies

$$\begin{aligned} x_0 + \Delta x &= -x_0 + N, \\ \Delta x &= N - 2x_0, \end{aligned} \tag{4.3}$$

which means that we can estimate the relative translation from the location of the zero of Δf. Equation (4.3) holds for arbitrarily sampled functions $f_1(x)$ and $f_2(x)$. Since $f_1(x)$ and $f_2(x)$ are discrete functions, we search for the minimum of $|\Delta f|$, instead of searching for the zero of Δf. In general, the equation $\Delta f(x) = 0$ does not have a unique solution.



(a)

(b)

Fig. 4. Translation estimation using the difference function. Given relatively translated functions $f_1(x)$ and $f_2(x)$ (a), the translation is estimated (b) by flipping $f_2(x)$, and computing the difference function Δf, whose zero corresponds to twice the shift.

5 Rotation Estimation Using the Angular Difference Function

In this section, we derive the difference function for 2D images. Given two images I_1 and I_2, we denote by $M_1(r,\theta)$ and $M_2(r,\theta)$ the magnitudes of their Fourier transforms in polar coordinates. If I_2 is a rotated and translated version of I_1, i.e.,

$$I_2(x,y)= \\ I_1(x\cos\Delta\theta+y\sin\Delta\theta+\Delta x,-x\sin\Delta\theta+y\cos\Delta\theta+\Delta y), \tag{5.1}$$

then

$$M_1(r,\theta)=M_2(r,\theta+\Delta\theta). \tag{5.2}$$

We define the difference function of $M_1(r,\theta)$ and $M_2(r,\theta)$ in the angular direction as

$$\Delta M(\theta)=\int_0^\infty |M_1(r,\theta)-M_2(r,-\theta)|dr, \qquad \theta\in[0,\pi]. \tag{5.3}$$

The value of $\Delta M(\theta_0)$ is zero if

$$\theta_0+\Delta\theta=-\theta_0 \qquad \text{or} \qquad \theta_0+\Delta\theta=-\theta_0+\pi, \tag{5.4}$$

where the second zero is due to the conjugate symmetry of M_1 and M_2. Thus, we get that the two zeros of $\Delta M(\theta)$, obtained at θ_0^1 and θ_0^2, are related to the relative rotation $\Delta\theta$ by

$$\theta_0^1=-\frac{\Delta\theta}{2}, \qquad \theta_0^2=-\frac{\Delta\theta}{2}+\frac{\pi}{2}. \tag{5.5}$$

We see from (5.5) that the zeros θ_0^1 and θ_0^2 are $\pi/2$ radians apart. This property is true for all zeros of ΔM: If θ_0 is a zero of ΔM, then $\theta_0+\frac{\pi}{2}$ is also a zero. Therefore, we define the angular difference function $\Omega(\theta)$ by

$$\Omega(\theta)\triangleq\Delta M(\theta)+\Delta M\Big(\theta+\frac{\pi}{2}\Big), \qquad \theta\in\Big[0,\frac{\pi}{2}\Big]. \tag{5.6}$$

The zero θ_0 of $\Omega(\theta)$ is related to the rotation angle $\Delta\theta$ by

$$\theta_0^1=-\frac{\Delta\theta}{2}. \tag{5.7}$$

Note that since we compute $\Omega(\theta)$ using the magnitude of the Fourier transform, it is invariant to translations of the input images.

5.1 Computing the Angular Difference Function for Discrete Images

An important property of Ω is that it can be discretized using very general sampling grids. The only requirement from the sampling grid is that, if θ is a sampling point, then, $\theta+\frac{\pi}{2}$ is also a sampling point. Therefore, to compute Ω accurately we do not need a true polar representation of the Fourier transforms of I_1 and I_2.

The reversal of the angular axis, indicated by (5.3), is accurately implemented by flipping the input image along the x or y axes. Mathematically,

$$\widetilde{I}(r_j,\theta_i)=I(r_j,-\theta_i) \quad \Leftrightarrow \quad \widetilde{I}(i,j)=I(-i,j). \tag{5.8}$$

The PPFT, presented in Section 3, is used to derive an algorithm for computing Ω. The PPFT evaluates the DFT of a given image on the pseudopolar grid. For each angle θ in the pseudopolar grid, the grid contains also the angle $\theta+\frac{\pi}{2}$. Thus, we use the PPFT algorithm to compute Ω as follows: Given input images I_1 and I_2, defined on a Cartesian grid,

1. Flip I_1 in the left → right direction.
2. Compute the PPFT of I_1 and I_2.
3. Compute M_1^d and M_2^d, where M_1^d and M_2^d are the magnitudes of the PPFT of I_1 and I_2, respectively.
4. Evaluate (5.3) using numerical integration

$$\Delta M^d(\theta_i)= \\ \sum_{0\leq r_j\leq 1}\big|M_1^d(r_j,\theta_i)-M_2^d(r_j,\theta_i)\big|\Delta r_i \,/\theta_i\in[0,\pi]. \tag{5.9}$$

   Note that the integration in (5.9) is computed over rays of the same length, where Δr_i is the radial sampling interval.
5. Compute Ω by

$$\Omega(\theta_i)=\Delta M^d(\theta_i)+\Delta M^d(\theta_{i+K}), \tag{5.10}$$

   where $K=2N+1$ is the size of the pseudopolar grid (the input image is of size $N\times N$). Equation (5.10) is the discrete equivalent of (5.6). Two samples of ΔM^d, given in (5.9), that are K samples apart correspond to frequencies that are $\frac{\pi}{2}$ apart.

5.1.1 The Normalized Angular Difference Function

In order to improve the robustness of the algorithm to image noise and intensity changes, we replace the L_1 norm in (5.9) with the normalized correlation [25]

$$\Delta M_N^d(\theta_i)= \\ \frac{\sum_{0\leq r_j\leq 1}\Big(\overline{M_1^d}(r_j,\theta_i)-\overline{M_2^d}(r_j,\theta_i)\Big)^2}{\sigma_r\big(M_1^d\big)\sigma_r\big(M_2^d\big)}, \qquad \theta_i\in[0,\pi], \tag{5.11}$$

where

$$\overline{M_k^d}(r_j,\theta_i)= \\ M_k^d(r_j,\theta_i)-\frac{1}{j_{\max}}\sum_{0\leq r_l\leq 1}M_k^d(r_l,\theta_i),\ k=1,2$$

and $j_{\max}$ is the index of the maximal radial sample of each ray i, such that $r(j_{\max})=1$. The standard deviation is then given by

$$\sigma_r\big(M_k^d\big)= \\ \sqrt{\frac{1}{j_{\max}}\sum_{0\leq r_l\leq 1}\Big(M_k^d(r_l,\theta_i)-\overline{M_k^d}(r_l,\theta_i)\Big)^2},\ k=1,2.$$

Similar to (5.10), Ω_N, the normalized angular difference function, is given by

$$\Omega_N(\theta_i)=\Delta M_N^d(\theta_i)+\Delta M_N^d(\theta_{i+K}). \tag{5.12}$$

Fig. 5 shows an example of Ω and ΔM for the F16 image pair, where the minima of Ω is clearly visible. Ω_N is significantly smoother than Ω and it is experimentally shown (see Section 6) to improve registration accuracy and robustness to noise.

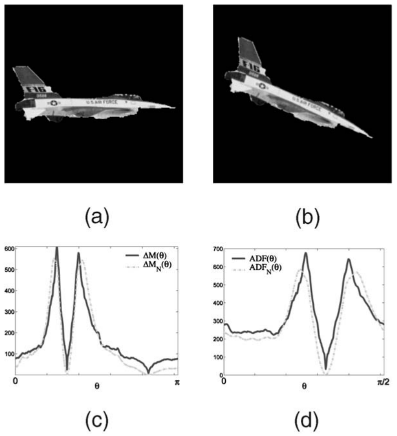


Fig. 5. The difference function and the angular difference function of rotated sample images. (a) and (b) The input images. (c) The difference function ΔM^d and the normalized difference function ΔM_N^d of (a) and (b). Two minima with an offset of $\frac{\pi}{2}$ can be observed in ΔM^d and ΔM_N^d. (d) The angular difference functions Ω and Ω_N are computed using ΔM^d and ΔM_N^d. Ω_N is smoother than Ω and has a lower minimum.

5.2 Rotation Estimation Algorithm

For two input images I_1 and I_2, where I_2 is a rotated and translated version of I_1, the rotation estimation algorithm is as follows:

1. Compute Ω (or Ω_N) of I_1 and I_2 as described in Section 5.1.
2. Compute the rotation angle $\Delta\theta$ as
$$\Delta\theta = 2\theta(i_0), \tag{5.13}$$
where
$$i_0 = \arg_i \min\{\Omega(\theta(i))\}, \tag{5.14}$$
and $\theta(i)$ is given by (3.12).
3. For $\Delta\theta$ found in Step 2, the true rotation angle is either $\Delta\theta$ or $\Delta\theta + \pi$. Check each of these angles and recover the translation parameters by using phase-correlation.

The main computational advantage of the algorithm is that the computation of Ω and Ω_N as well as the estimation of $\Delta\theta$ do not require any interpolations or approximations. The complexity of the algorithm is $O(N^2 \log N)$, which is the same as applying a single 2D FFT on the image. Most of the complexity is due to the computation of the PPFT, which requires $O(N^2 \log N)$ operations [15]. Once the PPFT is computed, the complexity of computing Ω and Ω_N is negligible. The complexity of other registration schemes [9], [22] is related to the interpolation of the Cartesian frequency grid into the polar grid. In any case, this complexity is at least $O(N^2 \log N)$ since the algorithm must compute the FFT of the input images at least once.

6 Experimental Results

We tested the registration algorithm on four images (see Fig. 6) with different rotation angles and noise levels. These images are rotated at angles $0^\circ \leq \theta \leq 175^\circ$ with increment $\Delta\theta = 5^\circ$, and translated randomly in the range $[-20, 20] \times [-20, 20]$ pixels. The results are given in Table 1, where we present both the average and maximal angular error. Note that the accuracy of the proposed algorithm is of the order of the angular sampling resolution and the mean and average errors are of the same order. The mean error is similar for Ω and Ω_N, while there is an improvement in the maximal error.

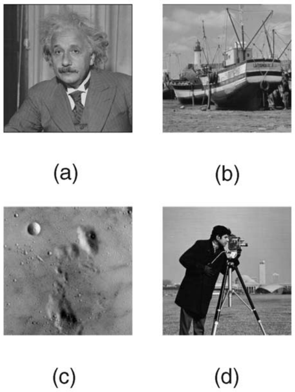


Fig. 6. Images used for registration accuracy estimation.

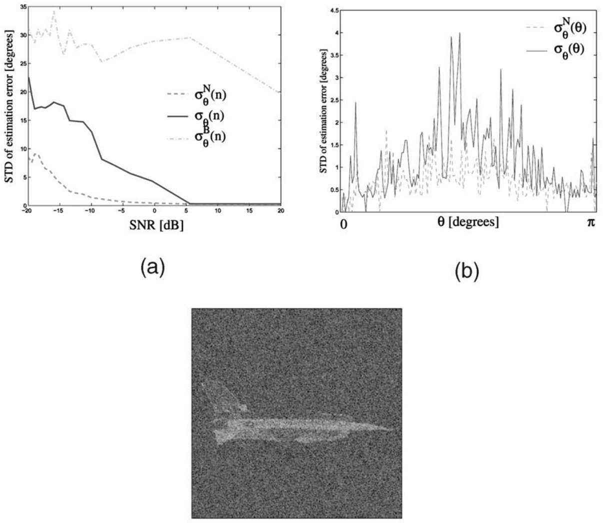
The robustness to noise was tested using the F16 and Lena images, given in Figs. 5a and 8c. The images were manually segmented to avoid boundary problems due to the synthetic rotation. Each image was rotated at all angles $0^\circ \leq \theta \leq 175^\circ$ with increment $\Delta\theta = 5^\circ$. The images were randomly translated with $(\Delta x, \Delta y)$ in the range $[-20, 20] \times [-20, 20]$ pixels. The rotated and translated images were then registered with the original image. This set of tests was repeated when additive white noise in the range $0 \leq \sigma_n \leq 400$ with increments $\Delta\sigma_n = 20$ was added. The noise was added to each image after the completion of its rotation. To assure the statistical validity of the results, each test set was repeated 10 times. Translations were estimated by a standard implementation of the phase correlation algorithm [8], [9].

Figs. 7a and 8a present σ_θ and σ_θ^N, which are the standard deviation (STD) of the registration error, computed using Ω, and Ω_N, respectively, as a function of the noise σ_n. It follows that for nonnoisy input images the registration error of the proposed algorithm is less than 1 degree, which corresponds to twice the average angular resolution of the PPFT, which is approximately 0.7 degree. The accuracy of the registration slightly degrades as the noise power increases. In particular, for SNRs above $9dB$, the estimation errors remain within 0.1 degree; below that threshold and down to about $0dB$, the errors become of the order of 0.5 degree. At about $-10dB$, the error is around 2 degree. Below this SNR and down to about $-15dB$, the errors become larger. The noisy images at $SNR = -8dB$ are depicted in Figs. 7c and 8d, where the F16 and Lena are hardly visible. Ω_N

TABLE 1
Registration Results for the Images in Fig. 6

	Ω		Ω_N	
	Mean error	Max error	Mean error	Max error
a	0.68°	1.27°	0.72°	1.13°
b	0.73°	1.77°	0.71°	1.28°
c	0.89°	1.43°	0.86°	1.21°
d	0.83°	2.13°	0.79°	1.67°

The errors are estimated over a range of rotations and translations. The mean error is similar for Ω and Ω_N, while there is an improvement in the maximal error.



(a)

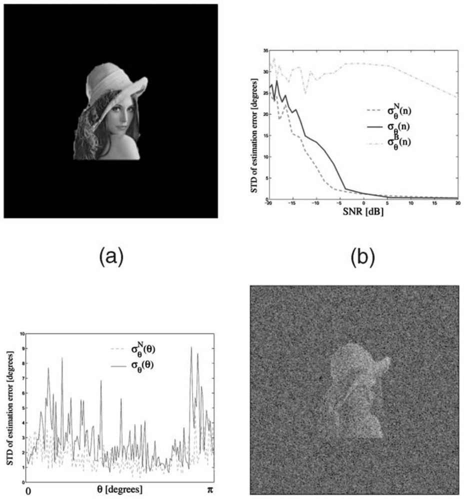
(b)

(c)

Fig. 7. Registration accuracy for the F16 image. (a) The standard deviation (STD) of the registration error (in degrees) as a function of the noise. σ_θ and σ_θ^N are the STD of Ω and Ω_N, respectively. σ_θ^B is the STD of the bilinear interpolation-based scheme given in [9]. For high SNR, the performance of Ω and Ω_N is comparable, while for low SNR Ω_N is significantly superior. (b) The STD of the registration error as a function of the rotation angle. The error of Ω_N is lower for any rotation angle. (c) A noisy F16 image (SNR=$-2dB$). The algorithm achieves an accuracy of 3 degrees.

outperforms Ω for low SNR values, while for high SNR they achieve comparable results.

The images were also registered using the interpolation-based scheme given in [9]. As expected, due to interpolation errors, this scheme is highly sensitive to noise. Moreover, even for noise-free images, significant estimation errors were obtained for rotation angles larger than 45 degrees. To conclude, the algorithm we propose successfully registers extremely noisy images, where feature-based techniques usually fail.



(a)

(b)

(c)

(d)

Fig. 8. Registration accuracy for the Lena image. (a) The STD of the registration error (in degrees) as a function of the noise. σ_θ and σ_θ^N are the STD of Ω and Ω_N, respectively. σ_θ^B is the STD of the bilinear interpolation based scheme given in [9]. For high SNR, the performance of Ω and Ω_N is comparable, while for low SNR Ω_N is significantly superior. (b) The STD of the registration error as a function of the rotation angle. The error of Ω_N is lower for any rotation angle. (c) A noisy F16 image (SNR=$-2dB$). The algorithm achieves an accuracy of 3 degrees.

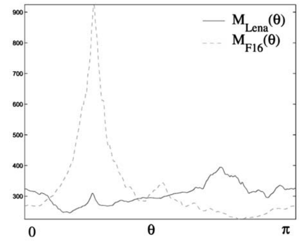
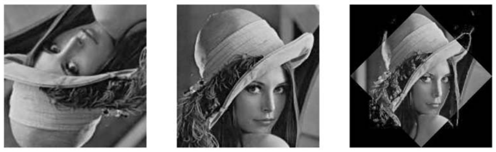


Fig. 9. The spectral content of the F16 and Lena images. The spectral energy of F16 is concentrated in $\left[0, \frac{\pi}{2}\right]$, while the spectral energy of Lena is evenly spread over $[0, \pi]$.

The dependence between the registration accuracy and the rotation angle is studied in Figs. 7b and 8b. The STD of the registration error with respect to the rotation angle θ, denoted σ_θ and σ_θ^N, were computed over the range of all noise levels. σ_θ and σ_θ^N for the Lena image (Fig. 8b) are unrelated to θ, while for the F16 image (Fig. 7b), the largest error is observed around $\theta = 90$. This is attributed to the difference in the spectral content of the images (see Fig. 9). The spectral energy of the F16 image is limited to $\left[0, \frac{\pi}{2}\right]$, while the spectral energy of the Lena image is evenly spread over $[0, \pi]$.

Fig. 10 shows an example of registering images with noncorresponding parts. This often happens when large motions are estimated. As the proposed scheme is robust, it can better handle such cases.

A typical application of the proposed scheme is given in Fig. 11, where we register fMRI images. Due to their acquisition process, such images (and medical imagery, in general) are often noisy and different images of the same object might have different intensities due to the calibrations of the imaging devices and postprocessing. Note the high accuracy of alignment although the images have some not common parts.

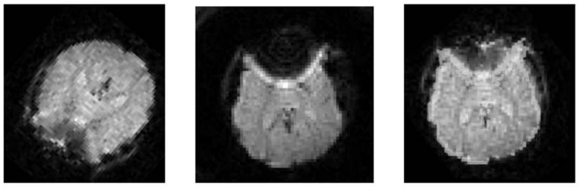


(a)

(b)

(c)

Fig. 10. Registration of images with noncorresponding parts. Image (a) is aligned to image (b). (c) is created by aligning (a) to (b). The edges in (b) are detected and overlayed on the aligned image (a).



(a)

(b)

(c)

Fig. 11. Registration of fMRI images. The image (a) is aligned to image (b). (c) is created by aligning (a) to (b). The edges in (b) are detected and overlayed on the aligned image (a).

The algorithm was implemented in C++ and the execution time for registering 256×256 images was approximately 1 second for each pair of images, using a 2.8GHz PC. Profiling shows that 95 percent of the execution time is spent on the computation of the PPFT, and there were no timing differences between using either Ω or Ω_N.

7 Conclusions

We introduced the angular difference function and its application to image registration. We showed that the angular difference function can be used to derive a robust registration algorithm, which has the same complexity as the FFT. The key point in the implementation of the algorithm is that it is possible to accurately compute the angular difference function by using the pseudopolar Fourier transform. Due to the low complexity of the proposed algorithm, it can be extended to real-time applications. It can also be extended to applications such as symmetry detection and 3D image registration (using a 3D extensions of the PPFT and the ADF), which are currently being investigated.

References

[1] F. Dufaux and J. Konrad, "Efficient, Robust, and Fast Global Motion Estimation for Video Coding," *IEEE Trans. Image Processing,* vol. 9, no. 3, pp. 497-501, Mar. 2000.

[2] A. Tekalp, *Digital Video Processing.* Prentice Hall, 1995.

[3] M. Irani and S. Peleg, "Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency," *J. Visual Comm. and Image Representation,* vol. 4, no. 4, pp. 324-335, Dec. 1993.

[4] S. Mann and R. Picard, "Virtual Bellows: Constructing High Quality Stills from Video," *Proc. IEEE Int'l Conf. Image Processing,* pp. 363-367, Nov. 1994.

[5] R. Szeliski, "Image Mosaicking for Tele-Reality Applications," *Proc. IEEE Workshop Applications of Computer Vision,* pp. 44-53, May 1994.

[6] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, "Mosaic Based Representations of Video Sequences and Their Applications," *Signal Processing: Image Comm.,* vol. 8, no. 4, pp. 327-351, May 1996.

[7] G. Wolberg and S. Zokai, "Robust Image Registration Using Log-Polar Transform," *Proc. IEEE Int'l Conf. Image Processing,* vol. 1, pp. 493-496, Sept. 2000.

[8] C.D. Kuglin and D.C. Hines, "The Phase Correlation Image Alignment Method," *Proc. IEEE Conf. Cybernetics and Soc.,* pp. 163-165, Sept. 1975.

[9] S. Reddy and B.N. Chatterji, "An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration," *IEEE Trans. Image Processing,* vol. 3, no. 8, pp. 1266-1270, Aug. 1996.

[10] B. Porat, *A Course in Digital Signal Processing.* John Wiley, 1997.

[11] P. Milanfar, "Projection-Based, Frequency-Domain Estimation of Superimposed Translational Motions," *J. Optical Soc. Am.: A, Optics and Image Science,* vol. 13, no. 11, pp. 2151-2162, Nov. 1996.

[12] P. Milanfar, "Two-Dimensional Matched Filtering for Motion Estimation," *IEEE Trans. Image Processing,* vol. 8, no. 3, pp. 438-443, Mar. 1999.

[13] Y. Chou and H. Hang, "A New Motion Estimation Method Using Frequency Components," *J. Visual Comm. and Image Representation,* vol. 8, no. 1, pp. 83-96, 1997.

[14] H. Stone, M. Orchard, E.-C. Chang, and S. Martucci, "A Fast Direct Fourier-Based Algorithm for Subpixel Registration of Images," *IEEE Trans. Geoscience and Remote Sensing,* vol. 39, no. 10, pp. 2235-2243, Oct. 2001.

[15] A. Averbuch, D. Donoho, R. Coifman, M. Israeli, and Y. Shkolnisky, "Fast Slant Stack: A Notion of Radon Transform for Data in Cartesian Grid which Is Rapidly Computable, Algebrically Exact, Geometrically Faithful and Invertible," *SIAM Scientific Computing,* to appear.

[16] L. Lucchese and G. Cortelazzo, "A Noise-Robust Frequency Domain Technique for Estimating Planar Roto-Translations," *IEEE Trans. Signal Processing,* vol. 48, no. 3, pp. 1769-1786, June 2000.

[17] L. Lucchese, "A Frequency Domain Algorithm for Dectection and Classification of Cyclic and Dihedral Symmetries in Two-Dimensional Patterns," *Proc. IEEE Int'l Conf. Image Processing (ICIP),* vol. II, pp. 793-796, Sept. 2002.

[18] S. Derrode and F. Ghorbel, "Shape Analysis and Symmetry Detection in Gray-Level Objects Using the Analytical Fourier-Mellin Representation," *Signal Processing,* vol. 84, no. 1, pp. 25-39, Jan. 2004.

[19] S. Derrode and F. Ghorbel, "Robust and Efficient Fourier-Mellin Transform Approximations for Gray-Level Image Reconstruction and Complete Invariant Description," *Computer Vision and Image Understanding,* vol. 83, no. 1, pp. 57-78, July 2001.

[20] H. Foroosh, J. Zerubia, and M. Berthod, "Extension of Phase Correlation to Subpixel Registration," *IEEE Trans. Image Processing,* vol. 11, no. 3, pp. 188-200, Mar. 2002.

[21] D.J. Fleet, "Disparity from Local Weighted Phase-Correlation," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics,* pp. 48-56, 1994.

[22] Q. Chen, M. Defrise, and F. Deconinck, "Symmetric Phase-Only Matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 12, pp. 1156-1168, Dec. 1994.

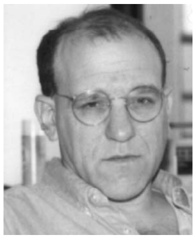
[23] D.H. Bailey and P.N. Swarztrauber, "The Fractional Fourier Transform and Applications," *SIAM Rev.,* vol. 33, no. 3, pp. 389-404, Sept. 1991.

[24] L.R. Rabiner, R.W. Schafer, and C.M. Rader, "The Chirp Z-Transform Algorithm," *IEEE Trans. Audio ElectroScoustics,* vol. AU, no. 17, pp. 86-92, June 1969.

[25] R.C. Gonzalez and R.E. Woods, *Digital Image Processing,* chapter 9, third ed. Reading, Mass.: Addison-Wesley, 1992.

Yosi Keller received the BSc degree in electrical engineering in 1994 from The Technion-Israel Institute of Technology, Haifa. He received the MSc and PhD degrees in electrical engineering from Tel-Aviv University, Tel-Aviv, in 1998 and 2003, respectively. From 1994 to 1998, he was an R&D Officer in the Israeli Intelligence Force. He is a visiting assistant professor with the Department of Mathematics, Yale University. His research interests include motion estimation, video analysis, image restoration, and statistical pattern analysis. He is a member of the IEEE.

Yoel Shkolnisky received the BSc degree in mathematics and computer science in 1996 from Tel-Aviv University, Tel-Aviv, Israel. He received the MSc degree in computer science in 2001 from Tel-Aviv University. He is currently a PhD student at Tel-Aviv University under the supervision of Professor Amir Averbuch. His research interests include computational harmonic analysis and scientific computing in problems related to irregular sampling and polar processing. His work was supported by a grant from the Ministry of Science, Israel.



Amir Averbuch received the BSc and MSc degrees in mathematics from the Hebrew University in Jerusalem, Israel, in 1971 and 1975, respectively. He received the PhD degree in computer science from Columbia University, New York, in 1983. From 1966-1970 and 1973-1976, he served in the Israeli Defense Forces. From 1976-1986, he was a research staff member at IBM T.J. Watson Research Center, Yorktown Heights, New York, Department of Computer Science. In 1987, he joined the Department of Computer Science, School of Mathematical Sciences, Tel Aviv University, where he is now professor of computer science. His research interests include wavelets, signal/image processing, multiresolution analysis, numerical computation for the solutions of PDEs, scientific computing (fast algorithms), and parallel and supercomputing (software and algorithms).