

# Pseudopolar-Based Estimation of Large Translations, Rotations, and Scalings in Images

Yosi Keller, Amir Averbuch, and Moshe Israeli

**Abstract**—One of the major challenges related to image registration is the estimation of large motions without prior knowledge. This paper presents a Fourier-based approach that estimates large translations, scalings, and rotations. The algorithm uses the pseudopolar (PP) Fourier transform to achieve substantial improved approximations of the polar and log-polar Fourier transforms of an image. Thus, rotations and scalings are reduced to translations which are estimated using phase correlation. By utilizing the PP grid, we increase the performance (accuracy, speed, and robustness) of the registration algorithms. Scales up to 4 and arbitrary rotation angles can be robustly recovered, compared to a maximum scaling of 2 recovered by state-of-the-art algorithms. The algorithm only utilizes one-dimensional fast Fourier transform computations whose overall complexity is significantly lower than prior works. Experimental results demonstrate the applicability of the proposed algorithms.

**Index Terms**—Global motion estimation, gradient methods, image alignment, subpixel registration.

## I. INTRODUCTION

IMAGE registration plays a vital role in many image processing applications such as video compression [1], [2], video enhancement [3], and scene representation [4]–[6] to name a few. This problem was analyzed by various computational techniques, such as pixel domain gradient methods [1], [4], [5], correlation techniques [20], and discrete Fourier (DFT) domain algorithms [10], [16]. Gradient methods are considered to be the state-of-the-art but may fail unless the two images are misaligned by a moderate motion. Fourier-based schemes, which are able to estimate large rotations, scalings, and translations, are often used to bootstrap the more accurate gradient methods. The basic notion related to Fourier-based schemes is the *shift property* [23] of the Fourier transform which allows robust estimation of translations using *normalized phase correlation* [10], [12]–[15]. Hence, in order to account for rotations and scaling, the image is transformed into a uniform polar or log-polar Fourier representation, where rotations and

scalings are reduced to translations, which are estimated by phase correlation.

In this paper, we propose to estimate the polar and log-polar DFT using the PP fast Fourier transform (PPFFT) [26], [27]. The PPFFT estimates the DFT on a non-Cartesian grid, which is geometrically similar to the polar grid. Hence, by interpolating the polar and log-polar DFT using the PPFFT, a significantly smaller interpolation error is achieved. Furthermore, the smaller the relative rotation/scaling between the images, the smaller the difference between their PPFFT magnitudes and the interpolated polar/log-polar DFT magnitudes. In prior works [16], [21], the relative interpolation error is not decreased by lowering the relative motion.

The resulting algorithm is able to robustly register images rotated by arbitrary angles and scaled up to a factor of 4. It should be noted that the maximum scale factor recovered in [16] and [21] was 2.0 and 1.8, respectively. In particular, the proposed algorithm does not result to interpolation in either spatial or Fourier domain. Only one-dimensional (1-D) FFT operations are used, making it faster and especially suited for real-time applications.

The rest of paper is organized as follows. Prior techniques related to FFT-based image registration are given in Section II, while the PPFFT, which is used as a basis for the proposed algorithm is presented in Section III. The proposed algorithm is presented in Section IV, and its accuracy is analyzed in Section V. Experimental results are discussed in Section VI, and final conclusions are given in Section VII.

## II. PREVIOUS WORK

### A. Translation Estimation

The basis of the Fourier-based motion estimation is the *shift property* [23] of the Fourier transform. Denote by

$$\mathfrak{F}\{f(x, y)\} \triangleq \hat{f}(\omega_x, \omega_y) \quad (2.1)$$

the Fourier transform of  $f(x, y)$ . Then

$$\begin{aligned} \mathfrak{F}\{f(x + \Delta x, y + \Delta y)\} \\ = \hat{f}(\omega_x, \omega_y) e^{j(\omega_x \Delta x + \omega_y \Delta y)}. \end{aligned} \quad (2.2)$$

Equation (2.2) can be used for the estimation of image translation [10], [15]. Assume the images  $I_1$  and  $I_2$  satisfy

$$I_1(x + \Delta x, y + \Delta y) = I_2(x, y). \quad (2.3)$$

Equation (2.3) is Fourier transformed, yielding

$$\hat{I}_1(\omega_x, \omega_y) e^{j(\omega_x \Delta x + \omega_y \Delta y)} = \hat{I}_2(\omega_x, \omega_y) \quad (2.4)$$

Manuscript received September 12, 2002; revised February 13, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ivan W. Selesnick.

Y. Keller is with the Department of Mathematics, Yale University, New Haven, CT 06520 USA (e-mail: yosi.keller@yale.edu).

A. Averbuch is with the Department of Computer Science, School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel (e-mail: amir@math.tau.ac.il).

M. Israeli is with the Faculty of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: israeli@cs.technion.ac.il).

Digital Object Identifier 10.1109/TIP.2004.838692

and

$$\frac{\hat{I}_2(\omega_x, \omega_y)}{\hat{I}_1(\omega_x, \omega_y)} = e^{j(\omega_x \Delta x + \omega_y \Delta y)}. \quad (2.5)$$

Thus, the translation parameters  $(\Delta x, \Delta y)$  can be estimated in the spatial domain [10], [11] by computing the inverse FFT of (2.5)

$$\begin{aligned} \text{Corr}(x, y) &\triangleq \mathfrak{F}^{-1} \left\{ e^{j(\omega_x \Delta x + \omega_y \Delta y)} \right\} \\ &= \delta(x + \Delta x, y + \Delta y) \end{aligned} \quad (2.6)$$

and finding the position of the maximum value of the correlation function  $\text{Corr}(x, y)$

$$(x, y) = \arg \max_{(\tilde{x}, \tilde{y})} \{ \text{Corr}(\tilde{x}, \tilde{y}) \}. \quad (2.7)$$

Further robustness to intensity differences between the images is achieved using the normalized phase correlation [10], [11]

$$\begin{aligned} \widetilde{\text{Corr}}(\omega_x, \omega_y) &\triangleq \frac{\hat{I}_2(\omega_x, \omega_y) \left| \hat{I}_1(\omega_x, \omega_y) \right|}{\left| \hat{I}_2(\omega_x, \omega_y) \right| \left| \hat{I}_1(\omega_x, \omega_y) \right|} \\ &= \frac{\hat{I}_2(\omega_x, \omega_y) \hat{I}_1^*(\omega_x, \omega_y)}{\left| \hat{I}_2(\omega_x, \omega_y) \right| \left| \hat{I}_1(\omega_x, \omega_y) \right|} \\ &= e^{j(\omega_x \Delta x + \omega_y \Delta y)} \end{aligned} \quad (2.8)$$

where  $*$  denotes the complex conjugate.

This scheme was proven to robustly estimate large translations where the corresponding overlap between the images to be registered is down to 30% of the smallest image size [15]. No smoothness assumption is used, thus, nonsmooth and noisy functions [such as two-dimensional (2-D) DFT coefficients] can be accurately registered. Shekarforoush *et al.* [11] extended the phase correlation-based algorithm to subpixel accuracy by analyzing the shape of  $\widetilde{\text{Corr}}(\omega_x, \omega_y)$  in (2.8) around its maximum.

A different approach to phase correlation-based translation estimation [14], [15], [9] utilizes 2-D linear regression to fit the phase values calculated in (2.8) to a two-dimensional (2-D) linear function

$$-j \cdot \log \left( \widetilde{\text{Corr}}(\omega_x, \omega_y) \right) = \omega_x \Delta x + \omega_y \Delta y \quad \forall (\omega_x, \omega_y). \quad (2.9)$$

Solving (2.9) using linear regression might prove inaccurate [11] due to aliasing and phase wrapping (around  $2\pi$ ) of the spectra at low frequencies. An iterative solution to phase unwrapping was suggested in [14], while Stone *et al.* [15] presented two approaches for modeling aliasing effects and improving the registration accuracy.

### B. Polar Fourier Representations

The polar Fourier representation (Fourier-Mellin transform) is used to register images that are misaligned due to translation, rotation and scale [16], [21]. Let  $I_2$  be a translated, rotated, and scaled replica of  $I_1$

$$\begin{aligned} I_2(x, y) &= I_1(s \cdot x \cdot \cos \theta_0 + s \cdot y \cdot \sin \theta_0 + \Delta x, \\ &\quad -s \cdot x \cdot \sin \theta_0 + s \cdot y \cdot \cos \theta_0 + \Delta y) \end{aligned} \quad (2.10)$$

where  $\theta_0$ ,  $s$  and  $(\Delta x, \Delta y)$  are the rotation angle, scale factor, and translation parameters, respectively. The Fourier transform of (2.10) is

$$\hat{I}_2(r, \theta) = e^{j(\omega_x \Delta x + \omega_y \Delta y)} \frac{1}{s^2} \hat{I}_1\left(\frac{r}{|s|}, \theta + \theta_0\right). \quad (2.11)$$

Hence,  $M_1$  and  $M_2$ , the magnitudes of  $\hat{I}_1$  and  $\hat{I}_2$ , respectively, are related by rotation and scaling around the dc component

$$M_1(r, \theta) = \frac{1}{s^2} M_2\left(\frac{r}{|s|}, \theta + \theta_0\right) \quad (2.12)$$

where  $(r, \theta)$  are the polar coordinates

$$\begin{aligned} \omega_x &= r \cos \theta \\ \omega_y &= r \sin \theta. \end{aligned}$$

Thus, rotations and scalings are recovered first, regardless of the translation parameters. Using a polar Fourier Transform, rotations are reduced to translations, which can be recovered by phase-correlation Using (2.12) to estimate the rotation angle  $\theta$  results in an ambiguity of  $\pi$  [16] in the estimation of the rotation angle. This ambiguity is resolved by rotating  $I_2$  according to both hypotheses ( $\theta$  and  $\theta + \pi$ ) and recovering the translational motion ( $\Delta x$  and  $\Delta y$ ) and the correlation peak for each hypothesis. The parameters corresponding to the highest correlation peak are chosen as the result.

The following algorithm is used to account for rotation and scaling.

- 1) The input images are transformed into a polar or log-polar Fourier grid using bi-linear interpolation.
- 2) Rotations and scalings in these representations are reduced to translations and are estimated by phase-correlation **(2.8)**.
- 3) The translation is estimated twice – for rotations of  $\theta$  and  $\theta + \pi$ . The result with the highest correlation peak is chosen.

The estimation of the Fourier transform on a polar or log-polar grid can be conducted using either Image domain warping [20] and then 2-D FFT calculation [19] or interpolation of the 2-D DFT transform in the Fourier domain [16], [21].

Fig. 1 illustrates the above algorithm, where the interpolation used in step 1) induces significant errors to the approximation of the polar and log-polar FFT, thus limiting the accuracy of the motion estimation [20].

### III. PP FOURIER TRANSFORM

The proposed registration algorithm is based on the PP Fourier transform (PPFT) [30]. The PPFT evaluates the 2-D DFT of a function on an oversampled set of angularly nonequispaced frequencies, which we call the PP grid. Both the forward and inverse PPFTs are implemented using fast algorithms. Moreover, their implementation requires only 1-D equispaced FFTs. In particular, the algorithm does not require regridding or interpolation. For a detailed description of the PPFT, see [30].

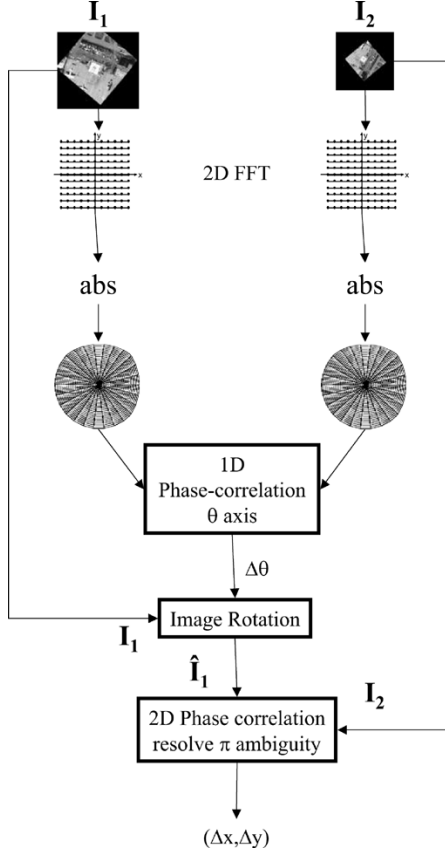


Fig. 1. Flow of the state-of-the-art FFT-based image registration. 1) The magnitudes of the polar DFT are approximated by interpolating the magnitudes of the 2-D FFT. 2) The rotation  $\Delta\theta$  is recovered using 1-D phase-correlation on the  $\theta$  axis. 3) One of the input images is rotated by angle  $\Delta\theta$ . 4) The translation is recovered and the ambiguity in  $\theta$  is resolved by applying a 2-D phase-correlation twice: using  $I_1$  rotated by both  $\theta$  and  $\theta + \pi$ .

In Section III-A, we present the fractional FFT. In Section III-B, we describe PPFT and how to compute it using the Fractional FFT. In Section III-C, we conclude the presentation of the PPFT by presenting its geometric interpretation.

#### A. Fractional FFT

The fractional FFT (FRFT) [28], with its generalization given by the chirp Fourier transform [29], is a fast  $O(N \log N)$  algorithm that evaluates the DFT of a function on any equally spaced set of  $N$  points on the unit circle. Specifically, the FRFT evaluates the DFT on the points

$$\omega_k = k\Delta\omega, \quad k = 0, 1, \dots, N-1 \quad (3.1)$$

where  $\Delta\omega$  is an arbitrary frequency spacing and  $N$  is the length of the input signal. For  $\Delta\omega = 2\pi/N$ , the FRFT evaluates the standard DFT. The frequencies, at which the FRFT evaluates the DFT, are illustrated in Fig. 2. We denote by  $F_\alpha$  the centralized fractional FFT (CFRFT), which computes the FRFT around the dc component. Thus, the CFRFT  $F_\alpha$  “compresses” the FFT of the input signal around the dc component, where  $\alpha$  is the compression ratio. Fig. 2(b) is an example of CFRFT with  $\alpha = 0.333$ .

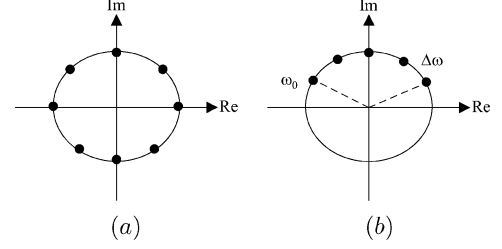


Fig. 2. Frequency samples of the FFT and Fractional FFT on the unit circle using  $N$  samples and a spacing of  $\Delta\omega$ . (a) The FFT samples the Fourier transform equally over  $[0, 2\pi]$ , where  $\Delta\omega = 2\pi/N$ . (b) The FRFFT samples the Fourier transform equally with any spacing  $\Delta\omega$  and initial phase  $\omega_0$ .

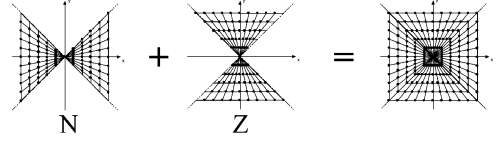


Fig. 3. Construction of the PP grid from the  $Z$  and  $N$  subgrids.

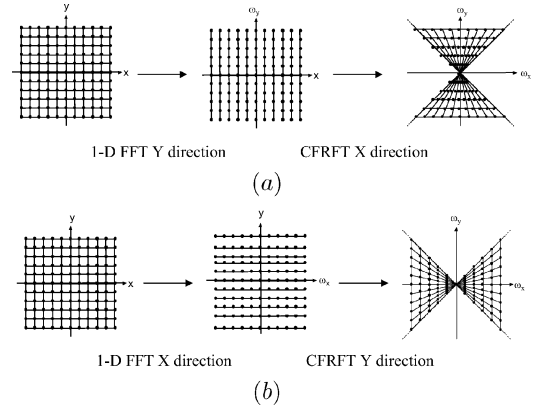


Fig. 4. Computation of the PP subgrids. (a) The computation of the  $Z$  subgrid. The input image is 1-D FFT transformed in the  $Y$  direction. Then, the fractional FFT is applied in the  $X$  direction. (b) The computation of the  $N$  subgrid. The input image is 1-D FFT transformed in the  $X$  direction. Then, the fractional FFT is applied in the  $Y$  direction.

#### B. Computing the PPFFT

We decompose the PP grid into two subgrids, denoted  $Z$  and  $N$ , as shown in Figs. 3 and 4. We denote by  $PP_Z$  and  $PP_N$  the values of the DFT evaluated on the subgrids  $Z$  and  $N$ , respectively. Next, we describe the algorithm that computes  $PP_Z$ . The algorithm for  $PP_N$  is easily obtained by switching the roles of the  $X$  and  $Y$  axes and a thorough description of the algorithm is given in [30].

1) *Computing  $PP_Z$* : The computation of  $PP_Z$  is based on the separability of the 2-D DFT.  $PP_Z$  is computed by applying the 1-D FFT in the  $Y$  direction and then applying the CFRFT in the  $X$  direction. A varying  $\alpha$  factor for the CFRFT is chosen such that the  $Z$  subgrid geometry is obtained.

##### a) Algorithm Flow:

- 1) Let  $I$  be the input image of size  $(n_0, m_0)$ .  $I$  is zero-padded to size

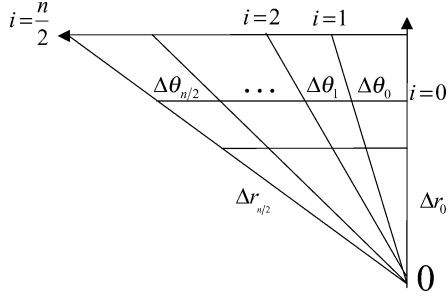


Fig. 5. Geometrical properties of the PP grid. The angular and radial spacings  $\Delta\theta_{PP}$  and  $\Delta r_{PP}$ , respectively, vary smoothly as a function of  $i$ .

$(n, n) = (2^k, 2^k)$   $k \in \mathbb{Z}$  where  $k$  satisfies  $2^k \geq 2 \cdot \text{Max}(n_0, m_0)$ .

2) Apply the 1-D FFT in the  $Y$  direction  $I_x^\wedge = \text{FFT}_X(I)$ . Cyclically shift the result such that the dc component is in the center of  $I_x^\wedge$ . The shifting operation is similar to the "fftshift" command in Matlab.

3) Apply the CFRFT operator  $F_\alpha$  to the rows of  $I_x^\wedge$

$$\text{PP}_Z^i = F_{\alpha_i} \left( I_x^\wedge \right) \quad (3.2)$$

where  $\text{PP}_Z^i$  is the  $i$ th row of  $\text{PP}_Z$ ,  $F_{\alpha_i}$  is the CFRFT operator with a compression ratio of  $\alpha_i$ ,  $I_x^\wedge$  is the  $i$ th row of  $I_x^\wedge$ , and

$$\alpha_i = \begin{cases} \frac{n-2i}{n} & i \leq \frac{n}{2} \\ 0 & i = \frac{n}{2} \\ -\frac{n-2i}{n} & i > \frac{n}{2} \end{cases} \quad (3.3)$$

To evaluate  $\text{PP}_N$ , we transpose the input image  $I$  and reapply the above algorithm.

### C. Geometric Interpretation and Properties

$\text{PP}_Z$  and  $\text{PP}_N$  are matrices of size  $n \times n$ , whose elements are the values of the DFT on the PP grid. By examining Fig. 3, we see that each column in the matrices  $\text{PP}_Z$  and  $\text{PP}_N$  corresponds to a ray on the PP grid with a fixed angle  $\theta$ . Similarly, each row corresponds to some radius  $r$ . The polar and PP grids differ due the nonuniformity of the angular and radial spacings of the PP grid shown in Fig. 5. For the polar grid, we have

$$\Delta\theta_{\text{Polar}}(i) = \frac{2\pi}{n}, \quad \Delta r_{\text{Polar}}(i) = \Delta r_0$$

while, for the PP grid,  $\Delta\theta_{PP}(i)$  and  $\Delta r_{PP}(j)$  vary smoothly as a function of  $i$ . Specifically, for the PP grid, we have

$$\theta_{PP}(i) = \arctan\left(\frac{2i}{n}\right), \quad i = 0, \dots, \frac{n}{2} \quad (3.4)$$

where  $n$  is the side of a  $n \times n$  image and  $i$  is the index of the ray. Therefore

$$\Delta\theta_{PP}(i) \triangleq \theta_{PP}(i+1) - \theta_{PP}(i). \quad (3.5)$$

$$\begin{aligned} \Delta r_{PP}(i) &= \frac{\sqrt{\left(\frac{n}{2}\right)^2 + i^2}}{\left(\frac{n}{2}\right)} \\ &= \sqrt{1 + 4\left(\frac{i}{n}\right)^2}, \quad i = 0, \dots, \frac{n}{2}. \end{aligned} \quad (3.6)$$

## IV. PROPOSED REGISTRATION ALGORITHM

This section presents the proposed image registration algorithms, which use the PPFFT to reduce rotations and scalings to translations in the polar and log-polar domains, respectively. These translations are recovered using the phase-correlation scheme presented in Section II-A. Thus, the phase-correlation algorithm is applied on three domains: spatial, polar DFT's magnitude, and log-polar DFT's magnitude. Table I summarizes the translation estimation properties in each domain. Detailed implementation issues are given in Section IV-A.

Section IV-B introduces the rotation and translation estimation algorithm illustrated in Fig. 6, while the simultaneous estimation of scaling, rotation and translation is presented in Section IV-C and Fig. 7.

### A. Translation Estimation Implementation

The translation is estimated by (2.8), where implementation issues are discussed according to the domain of the motion. The evaluation of (2.8) requires the input signals to be zero-padded in order to have the same size. Furthermore, implementing (2.8) using the DFT estimates the *cyclic correlation* [24], which differs from the regular correlation needed for translation estimation and can also be solved by zero-padding [24]. Section IV-A.1 presents the padding used for spatial domain translation estimation, while in Sections IV-A.2 and IV-A.3, the  $\vec{\theta}$  axis is cyclic and no padding is needed.

1) *Spatial Domain Translation Estimation*: Denote by  $\text{length}(I)$  the length of the 1-D signal  $I$ . Then, in order to avoid the cyclic wrap-around problem in the DFT computation, both signals are zero-padded such that

$$\begin{aligned} \text{length}(\tilde{I}_1) &= \max(\text{length}(I_1), \text{length}(I_2)) + \Delta X_{\max} \\ \text{length}(\tilde{I}_2) &= \text{length}(\tilde{I}_1) \end{aligned} \quad (4.1)$$

where

- 1)  $\tilde{I}_1$  and  $\tilde{I}_2$  are the zero-padded versions of  $I_1$  and  $I_2$ , respectively.
- 2)  $\Delta X_{\max}$  is the maximum translation value for which the correlation function is evaluated.

In the 2-D case, the input signals  $I_1$  and  $I_2$  are zero padded in both axes. This is done according to (4.1), using the maximal translations ( $\Delta X_{\max}$  and  $\Delta Y_{\max}$ ) in both axes.

2) *Translation Estimation in the Polar Domain*: In the polar domain, the angular axis  $\vec{\theta}$  is cyclic over  $[0, 2\pi]$ . Hence, no zero padding is needed and the images are padded to be of the same size.

3) *Translation Estimation in the Log-Polar Domain*: In the log-polar domain, a distinction is made between the angular axis  $\vec{\theta}$  and the radial axis  $\vec{r}$ . The motion along the angular axis  $\vec{\theta}$  is

TABLE I  
PROPERTIES OF THE TRANSLATION ESTIMATION ALGORITHMS IN VARIOUS DOMAINS: SPATIAL, POLAR DFT'S MAGNITUDE, AND LOG-POLAR DFT'S MAGNITUDE

Estimated motion	Image domain used for registration	Domain properties	Translation estimation	Described in section
translation	spatial	non-cyclic	$\vec{r}$ and $\vec{\theta}$ axes	IV-A.1
rotation	polar DFT magnitude	cyclic $\vec{\theta}$ axis	$\vec{\theta}$ axis	IV-A.2
rotation+scaling	log-polar DFT magnitude	cyclic $\vec{\theta}$ axis	$\vec{r}$ and $\vec{\theta}$ axes	IV-A.3

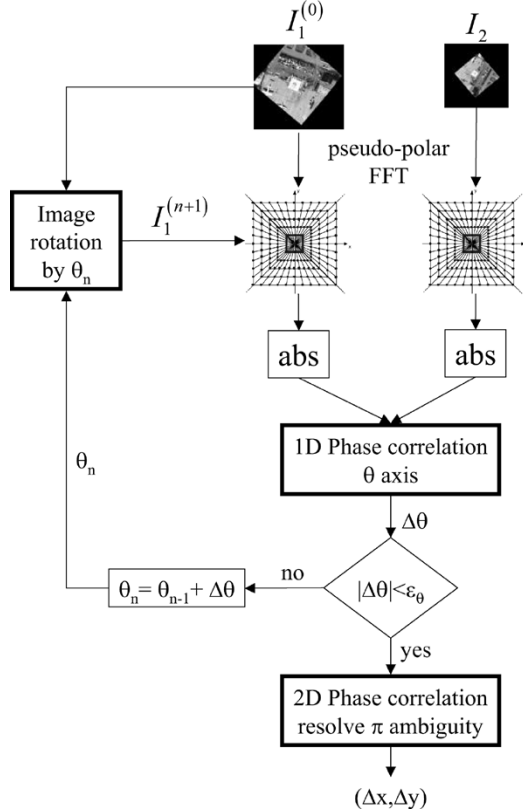


Fig. 6. Rotation and translation registration using the PPFFT. 1) The magnitude of the PPFFT is computed after the input images are zero-padded to the same size. 2) The rotation  $\Delta\theta$  is recovered by applying 1-D phase correlation on the  $\theta$  axis in the PP domain. 3) One of the input images is rotated by the accumulated rotation angle  $\theta_n$ . 4) Steps 1)–3) are reiterated until a stoppage condition is met. 5) The translation is recovered and the  $\theta$  ambiguity is resolved by applying the 2-D phase correlation twice: using  $I_1$  rotated by  $\theta$  and  $\theta + \pi$ .

estimated according to Section IV-A.2, while the motion along the radial axis  $\vec{r}$  is estimated according to Section IV-A.1.

### B. Estimation of Rotation and Translation

Let  $I_1$  and  $I_2$  be the input images to be registered.  $I_1^{(n)}$  denotes the fact that  $I_1$  evolves throughout the registration process where initially  $I_1^{(0)} = I_1$  ( $n = 0$ ) and  $I_2$  remains unchanged. The algorithm operates as follows.

1) Let  $(m_1, l_1)$  and  $(m_2, l_2)$  be the sizes of  $I_1$  and  $I_2$ , respectively. Then, at iteration  $n = 0$ ,  $I_1$  and  $I_2$  are zero-padded such that

$$m_1 = l_1 = m_2 = l_2 = 2^k, k \in \mathbb{Z}. \quad (4.2)$$

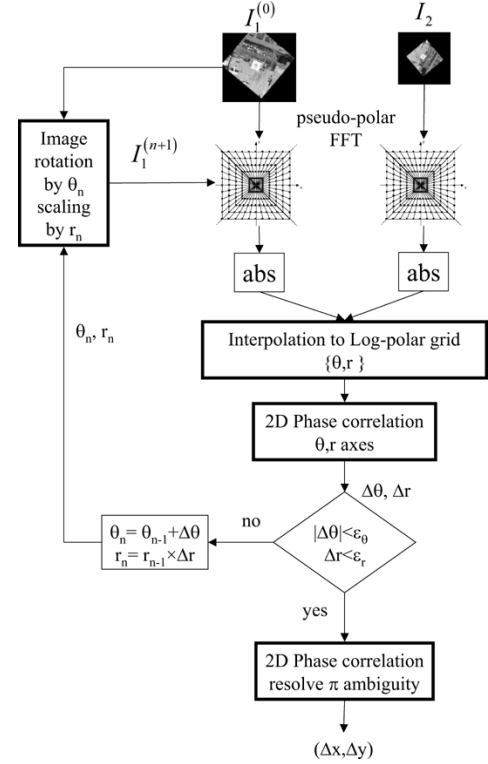


Fig. 7. Scaling, rotation, and translation estimation using the PPFFT. 1) The magnitude of the PPFFT is computed after the input images were zero-padded. 2) The rotation  $\Delta\theta$  and scaling  $\Delta r$  are recovered by applying 2-D phase correlation on the  $\theta$  and  $r$  axes in the PP domain. 3) One of the input images is scaled and rotated by an accumulated rotation  $\theta_n$  and scale  $\Delta r$ . 4) Steps 1)–3) are reiterated until a stoppage criteria is met. 5) The translation is recovered and the ambiguity in the value of  $\theta$  is resolved by applying the 2-D phase correlation twice.

- 2) The magnitudes  $M_1^{\text{PP}}(\theta_i, r_j)$  and  $M_2^{\text{PP}}(\theta_i, r_j)$  of the PPFFTs of  $I_1^{(n)}$  and  $I_2$  are computed, respectively.
- 3) The polar DFT's magnitudes  $\widehat{M}_1^{\text{Polar}}(\theta_i, r_j)$  and  $\widehat{M}_2^{\text{Polar}}(\theta_i, r_j)$  of  $I_1^{(n)}$  and  $I_2$  are substituted by  $M_1^{\text{PP}}(\theta_i, r_j)$  and  $M_2^{\text{PP}}(\theta_i, r_j)$ , respectively.
- 4) The translation along the  $\vec{\theta}$  axis of  $M_1^{\text{PP}}(\theta_i, r_j)$  and  $M_2^{\text{PP}}(\theta_i, r_j)$  is estimated using the procedure described in **Section IV-A.2** and the result is denoted  $\Delta\theta_n$ .
- 5) Let  $\theta_n$  be the accumulated rotation angle estimated at iteration  $n$

$$\theta_n \triangleq \sum_{i=0}^n \Delta\theta_i = \theta_{n-1} + \Delta\theta_n.$$

Then, the input image  $I_1$  is rotated by  $\theta_n$  (around the center of the image) using a fast FFT-based image rotation [7]

$$I_1^{(n+1)}(\theta, r) = I_1^{(0)}(\theta + \theta_n, r), n = 1, \dots$$

Rotation around the central pixel of  $I_1$  is recommend, making the bounding rectangular of the rotated image as small as possible.

6) Steps 2)-5) are reiterated until the angular refinement term  $\Delta\theta_n$  is smaller than a predefined threshold  $\varepsilon_\theta$ , i.e.,  $|\Delta\theta_n| < \varepsilon_\theta$ , or a predefined number of iterations  $n_{\max}$  is reached.

7) The rotated images  $I_1^{(n)}$  and  $I_2$  are used as inputs to the spatial domain translation estimation algorithm given in **Section IV-A.1** and the rotation ambiguity is resolved by **Section II-B**.

### C. Estimation of Rotation, Translation, and Scaling

Let  $I_1$  and  $I_2$  be the input images to be registered.  $I_1^{(n)}$  denotes the fact that  $I_1$  evolves throughout the registration process, while  $I_2$  remains unchanged. We assume that, initially,  $I_1^{(0)} = I_1$  ( $n = 0$ ).

The algorithm operates as follows.

1) Let  $(m_1, l_1)$  and  $(m_2, l_2)$  be the sizes of  $I_1^{(n)}$  and  $I_2$ , respectively. Then, at iteration  $n = 0$ ,  $I_1^{(0)}(i, j)$  and  $I_2$  are zero-padded such that

$$m_1 = l_1 = m_2 = l_2 = 2^k, k \in \mathbb{Z}. \quad (4.3)$$

The padding is needed to abide by the limitations of the PPFFT described in **Section III**.

2)  $M_1^{\text{PP}}(\theta, r)$  and  $M_2^{\text{PP}}(\theta, r)$ , the magnitudes of the PPFFTs of  $I_1^{(n)}$  and  $I_2$  are calculated, respectively.

3) Let

$$\text{base}^{\text{length}(M_2^{\text{PP}}(\theta_i, r_j))} = \text{length}(M_2^{\text{PP}}(\theta_i, r_j)) \quad (4.4)$$

be the logarithmic base used to define the log domain in the  $\vec{r}$  axis. The log-polar DFTs of  $I_1^{(n)}$  and  $I_2$  are approximated by  $\widehat{M}_1^{\text{PP}}(\theta, r)$  and  $\widehat{M}_2^{\text{PP}}(\theta, r)$ , respectively, by

$$\begin{aligned} \widehat{M}_1^{\text{Log-Polar}}(i, j) &= M_1^{\text{PP}}(i, \lfloor \text{base}^j \rfloor) \\ \widehat{M}_2^{\text{Log-Polar}}(i, j) &= M_2^{\text{PP}}(i, \lfloor \text{base}^j \rfloor) \end{aligned} \quad (4.5)$$

where  $\lfloor x \rfloor$  denotes the integral part of  $x$ . The polar axis is approximated using the same procedure as in **Section IV-B**, while the radial axis is approximated by nearest-neighbor interpolation.

4) The relative translation between  $\widehat{M}_1^{\text{Log-Polar}}$  and  $\widehat{M}_2^{\text{Log-Polar}}$  is recovered by a

2-D phase correlation on the  $\vec{\theta}$  and  $\vec{r}$  axes, given in **Section IV-A.3**.

5) Let  $\Delta\theta_n$  and  $\Delta r_n$  be the rotation angle and the scaling value estimated at iteration  $n$ , respectively. Then, the input image  $I_1$  is rotated (around the center of the image) [7] and then scaled using DFT domain zero-padding

$$I_1^{(n+1)}(\theta, r) = I_1^{(0)}(\theta + \theta_n, r \cdot r_n) \quad (4.6)$$

where

$$\begin{aligned} \theta_n &= \sum_{i=0}^n \Delta\theta_i = \theta_{n-1} + \Delta\theta_n \\ r_n &= \prod_{i=0}^n \Delta r_i = r_{n-1} \cdot \Delta r_n. \end{aligned} \quad (4.7)$$

6) Steps 1)-3) are reiterated until either the updated estimates are smaller than predefined thresholds  $|\Delta\theta_n| < \varepsilon_\theta$  and  $\Delta r_n < \varepsilon_r$ , or a predefined number of iterations  $n_{\max}$  is reached.

7) The rotation and scale parameters  $\theta_n$  and  $r_n$ , respectively, are used as inputs to the spatial domain registration algorithm presented in **Section IV-B** step 7).

### D. Complexity Estimation

The overall complexity of the proposed algorithm is  $O(n^2 \log n)$ , since it is dominated by the complexity of the 2-D FFT.

- 1) The PPFFT is computed using  $O(n^2 \log n)$  operations (Section III).
- 2) The fast image rotation algorithm [7] uses  $O(n^2 \log n)$  operations.
- 3)  $O(n^2 \log n)$  operations are needed for image zooming using FFT domain zero-padding.

### V. INTERPOLATION ACCURACY ANALYSIS

This section provides performance analysis of the proposed algorithm by comparing it to prior state-of-the-art techniques. The registration accuracy and robustness of FFT-based algorithms in the polar and log-polar grids are related to the interpolation accuracy of the DFT on these grids. In prior works [13], [16], the DFT's magnitudes on polar and log-polar grids were interpolated using the Cartesian 2-D DFT, while the proposed algorithm uses the PPFFT.

The interpolation accuracy depends on the distance between the interpolated grid points (polar and log-polar) and the source grid used for the interpolation (PP and Cartesian). We show that the accumulated distance between the PP and polar grids is smaller than the accumulated distance between the Cartesian and polar grids.

The angular differences between the PPFFT and uniform polar grids are analyzed in Section V-A, while radial accuracy analysis (with respect to the polar domain) is given in Section V-B. Accumulated distances for several image sizes are

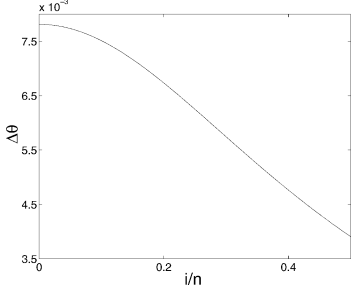


Fig. 8. Illustration of the nonuniformity of  $\Delta\theta_{PP}$ , the angular spacing of the PP grid.

given in Section V-C, where it is compared to the accumulated distances of interpolations based on the Cartesian grid.

#### A. Angular Accuracy Analysis

The polar grid shown in Fig. 5 has a constant angular spacing

$$\Delta\theta_{\text{Polar}} = \frac{2\pi}{n} \quad (5.1)$$

where  $n$  is the size of the polar  $\vec{\theta}$  axis.  $\Delta\theta_{PP}$ , the PPFFT's angular spacing is given in (3.5) and illustrated in Fig. 8.

In order to analyze the nonuniformity of  $\Delta\theta_{PP}$ , we expand  $\Delta\theta_{PP}$  using a second-order Taylor series expansion

$$\arctan(x+h) - \arctan(x) = \frac{1}{1+x^2}h + \frac{1}{2} \frac{2x}{(1+x^2)^2}h^2 + O(h)^3. \quad (5.2)$$

By substituting (3.5) into (5.2) ( $x = 2i/n$ ,  $h = 2/n$ ), we get

$$\begin{aligned} \widetilde{\Delta\theta_{PP}}(h, i) &\triangleq \frac{1}{1 + \left(\frac{2i}{n}\right)^2} \left(\frac{2}{n}\right) + \frac{\frac{2i}{n}}{\left(1 + \left(\frac{2i}{n}\right)^2\right)^2} \left(\frac{2}{n}\right)^2 \\ &= \frac{2n}{n^2 + 4i^2} + \frac{8i}{(n^2 + 4i^2)^2 n} \\ &= \frac{2n}{n^2 + 4i^2} + O\left(\frac{1}{n^5}\right). \end{aligned} \quad (5.3)$$

For common image sizes ( $n \approx 300 \sim 500$ ),  $O(1/n^5) = O(10^{-13})$ , the error term in (5.3) is negligible

$$\widetilde{\Delta\theta_{PP}} \approx \frac{2n}{n^2 + 4i^2} \quad (5.4)$$

and a constant angular spacing is achieved for low angles  $i \ll n/2$ .

#### B. Radial Accuracy Analysis

$\Delta r_{PP}$ , the radial spacing of the PP grid, is shown in Fig. 5.  $\Delta r_{PP}$  is constant along each ray and it varies smoothly from ray to ray according to (3.6). Fig. 9 shows  $\Delta r_{PP}$  as a function of the ray index and the finest spacing ( $\Delta r_{PP} = 1$ ) is achieved for low angles  $\theta \rightarrow (\pi/2)k$ ,  $k \in \mathbb{Z}$ .

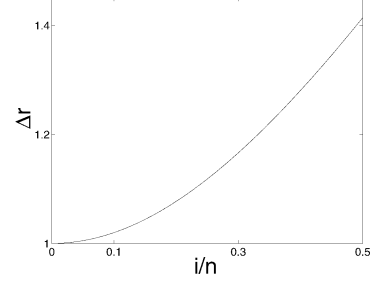


Fig. 9. PP grid spacing  $\Delta r$ .

#### C. Accuracy Analysis of Polar DFT Approximation Using the PPFFT

We compare the approximation accuracy of the polar DFT using the PPFFT and FFT. As the proposed algorithm uses a fast nearest-neighbor interpolation, first-order approximation errors are considered.

Denote by  $\varepsilon_{\text{Polar}}$  the first-order approximation errors of the polar DFT's magnitude

$$\begin{aligned} \varepsilon_P(\theta_i, r_j) &= \left| \widehat{M}_1^{\text{Polar}}(\theta_i, r_j) - M_1^{\text{Polar}}(\theta_i, r_j) \right| \\ &\approx \sqrt{\left( \frac{\partial}{\partial \theta} M_1^{\text{Polar}} \right)^2 + \left( \frac{\partial}{\partial r} M_1^{\text{Polar}} \right)^2} |\varepsilon_{\text{Grid}}(\theta_i, r_j)| \end{aligned} \quad (5.5)$$

where  $M_1^P(\theta_i, r_j)$  and  $\widehat{M}_1^P(\theta_i, r_j)$  are the magnitude and approximated magnitude of the polar DFT at a spatial frequency  $(\theta_i, r_j)$ . The interpolation error  $\varepsilon_{\text{Polar}}$  defined in (5.5) depends on the divergence of the PP and polar grids.  $\varepsilon_{\text{Grid}}(\theta_i, r_j)$  is the  $L_2$  distance between a point on the polar grid and the closest point on the PP grid

$$\varepsilon_{\text{Grid}}^2(\theta_i, r_j) = (X_{\text{Pseudopolar}}(i, j) - X_{\text{Polar}}(i, j))^2 + (Y_{\text{Pseudopolar}}(i, j) - Y_{\text{Polar}}(i, j))^2 \quad (5.6)$$

where

$$\begin{aligned} \{X_{\text{Pseudopolar}}(i, j), X_{\text{Pseudopolar}}(i, j)\} \\ = \{\cos(\theta_i) \Delta r(j), \cos(\theta_i) \Delta r(j)\} \end{aligned} \quad (5.7)$$

$$\begin{aligned} \{X_{\text{Polar}}(i, j), X_{\text{Polar}}(i, j)\} \\ = \left\{ \cos\left(\frac{2\pi i}{n}\right) \cdot j, \cos\left(\frac{2\pi i}{n}\right) \cdot j \right\}. \end{aligned} \quad (5.8)$$

The PP coordinates  $\theta_i$  and  $\Delta r(i)$  are defined in (3.4) and (3.6), respectively.

1) *Numerical Simulation of Grid Errors:* In order to compute the interpolation accuracy, the PP and polar grid are defined on the domain  $[-\pi, \pi]$ . Then, the nearest-neighbor distance between the two grids is calculated. The distance is computed by assigning per each polar destination point, the closest PP point, and accumulating the square distances in coordinates. The experiment is repeated with the PP grid replaced by the Cartesian grid. A graph of the average distance as a function of the density

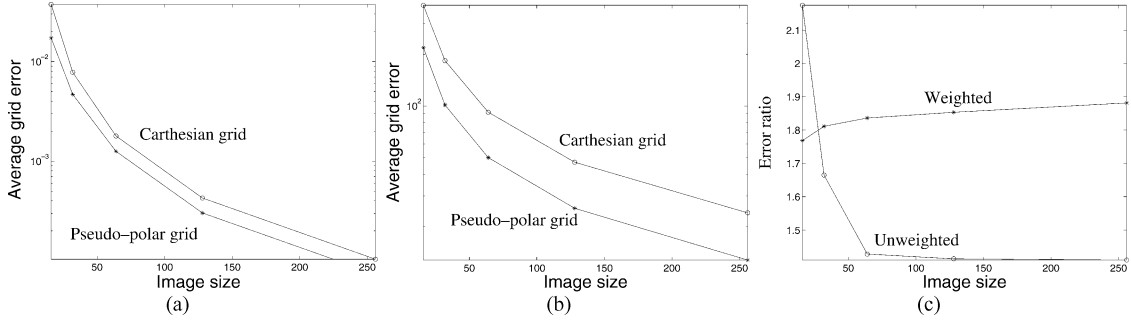


Fig. 10. Average distances between grids. The distance is computed by assigning per each polar destination point, the closest polar point, and accumulating the square distances in coordinates. (a) The distances between the cartesian and polar grids are larger than the distance between the PP and polar grid. (b) In order to simulate nearest-neighbor interpolation error, the distances are weighted by *Lena*'s FFT magnitude derivative, yielding a larger improvement for the PP grid. (c) The ratio between cartesian and PP-based interpolations for the weighted and unweighted errors.

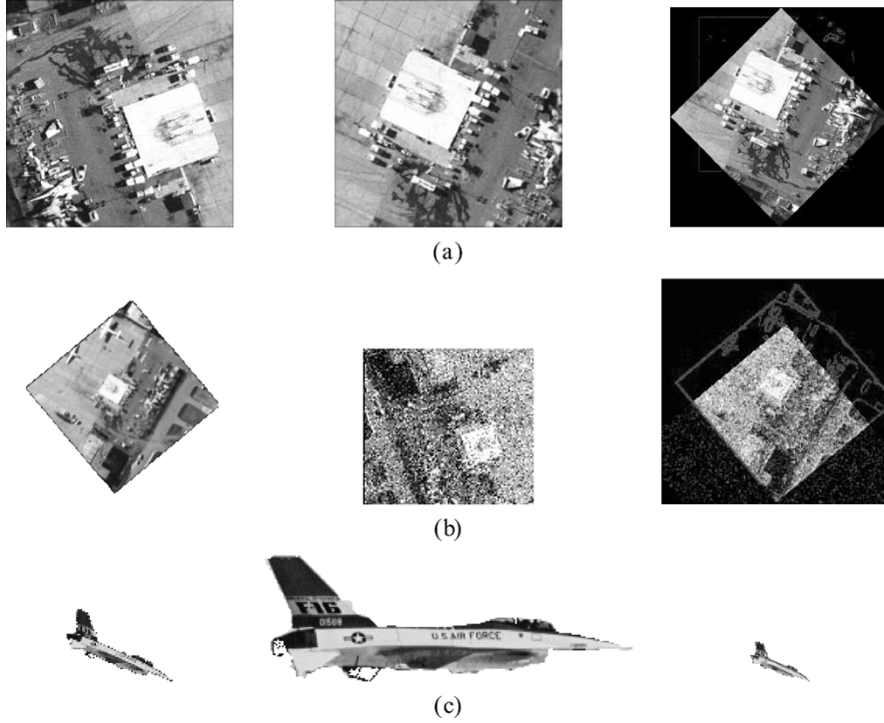


Fig. 11. Registration results of the *Airfield* and *F16* images. In each row, the left and center images were registered. The edges of the images in the center column are overlaid on the left images to illustrate the registration accuracy. The results are shown in the right column. (a) Scale = 1 and Angle = 100°. (b) Robustness was tested by adding noise ( $\mu = 0$  and  $\sigma = 40$ ) to the images related by Scale = 1 and Angle = 130°. (c) Registration of a segmented *F16* image. Scale = 3 and Angle = 23°.

of the starting grid is given in Fig. 10(a). The PP grid provides a 30% approximation error reduction as illustrated in Fig. 10(c).

The previous experiment analyzed the distance between grids. However, as we go far from the origin in the frequency domain, the magnitude of the DFT drops very sharply and, thus, errors taking place far from the origin should be down weighted according to (5.5). In order to take this into account, weighting was added to the previous experiment using the gradient of the magnitude of *Lena*'s FFT.

Fig. 10(b) and (c) show that an interpolation error reduction of 47% is achieved by the PP grid in the weighted case.

## VI. EXPERIMENTAL RESULTS

The proposed registration algorithm was tested using a large set of test images which were rotated, scaled and translated (similar to [16], [20]) to create the test pairs, similar to the

ones shown in Figs. 11 and 12. The images were registered by the rotation and rotation/scale algorithms presented in Sections IV-B and IV-C, respectively. We were able to consistently estimate scaling factors up to 4 and arbitrary rotation angles. We present the graphic results for the *Lena*, *Airfield*, and *F16* images. The results are illustrated by overlaying the edges of one image over the other according to the estimated motion parameters, while the corresponding numerical results are presented in Tables II and III, respectively. The translations were estimated by the phase-correlation algorithm discussed in Sections II-A and IV-A.1. We note that the choice of integral scaling factors for the experiments is not significant, the proposed algorithm was able to recover any scaling factor within its dynamic range that is discussed later.

Table II and Fig. 11 present the registration results of the *Airfield* image, which contains man-made objects characterized by



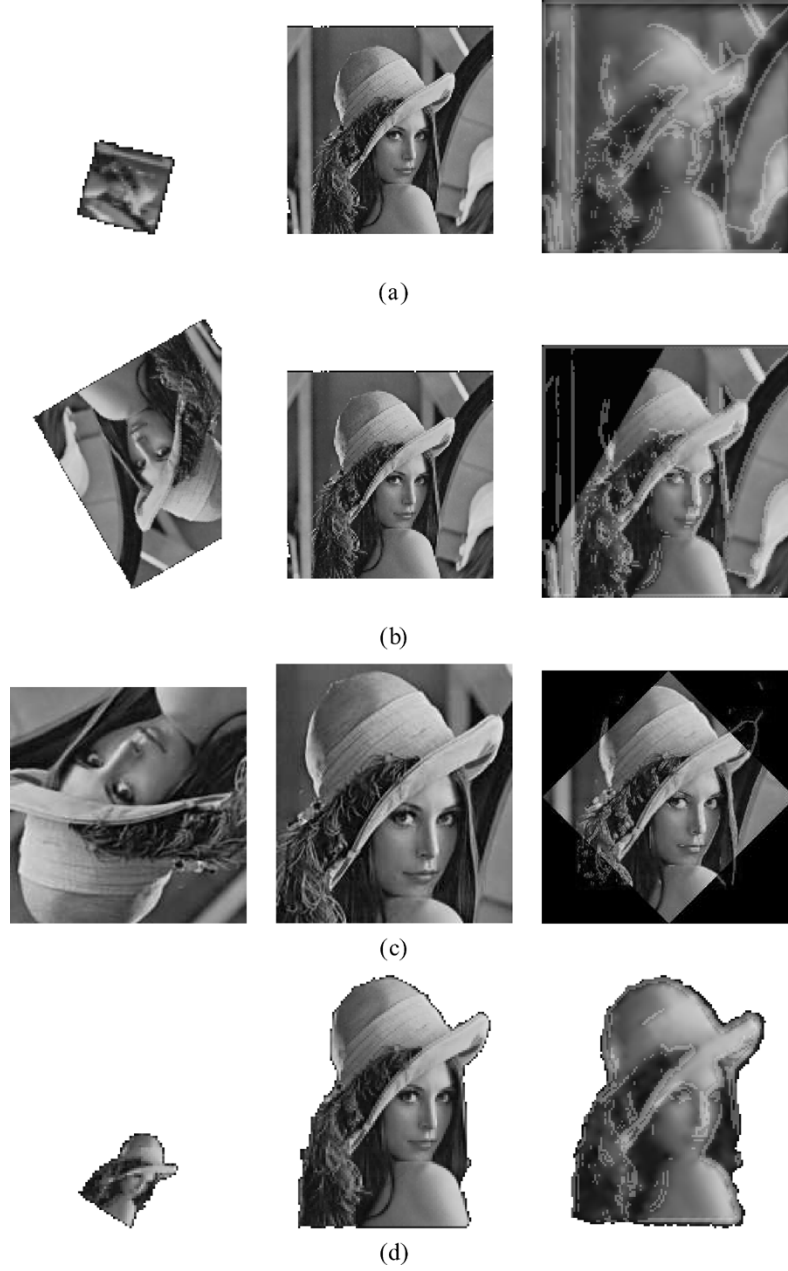


Fig. 12. Registration results of the *Lena* image when large scalings factors are present. In each row, the left and center images were registered. The edges of the center image are overlayed on the left image to illustrate the registration accuracy. The output is given in the right column. (a) Scale = 4 and Angle =  $105^\circ$ . (b) Images with partial non overlapping areas: Scale = 1 and Angle =  $150^\circ$ . (c) Scale = 1 and Angle =  $135^\circ$ . (d) Scale = 3 and Angle =  $33^\circ$ .

TABLE II  
REGISTRATION RESULTS OF THE IMAGE SETS PRESENTED IN FIG. 11 USING THE PP-BASED ALGORITHM. THE RESULTS DEMONSTRATE THE ALGORITHM'S ROBUSTNESS TO NOISE AND PARTIAL ALIGNMENT

Images	Correct ( $Scale, \theta$ )	Registration mode	Computed ( $Scale, \theta, \Delta x, \Delta y$ )
a	(1, $100^\circ$ )	scale,rotation,translation	(1, $100.65^\circ$ )
b	(1, $130^\circ$ )	rotation,translation	(1, $129.72^\circ$ )
c	(3, $23^\circ$ )	rotation,translation	(3, $23.22^\circ$ )

TABLE III  
REGISTRATION RESULTS OF THE *LENA* IMAGES PRESENTED IN FIG. 12 USING THE PP-BASED ALGORITHM

Image sequence	Registration mode	Correct ( $Scale, \theta$ )	Computed ( $Scale, \theta$ )
a	scale,rotation,translation	(4.0, $105^\circ$ )	(4.0, $103.78^\circ$ )
b	rotation,translation	(1, $150^\circ$ )	(1, $150.11^\circ$ )
c	scale,rotation,translation	(1, $135^\circ$ )	(1.03, $135.2^\circ$ )
d	scale,rotation,translation	(3, $33^\circ$ )	(3.01, $32.9^\circ$ )

sharp edges surrounded by smooth regions. The algorithm was able to register the images in Fig. 11(a) having a scaling factor of 2 and a large rotation of  $145^\circ$ . This test was repeated for var-

ious rotation angles, resulting in similar accurate registrations. The registration of these images was not influenced by the image boundaries, whose alignment corresponds to zero motion.

Fig. 11(b) shows image sets with partial alignment, which were registered by the proposed algorithm. The results are attributed to the statistical interpretation of the DFT's magnitude as the periodogram [23] of the images, which can be considered an approximation of the image's power spectrum [23]. These statistical properties are invariant to partial alignment where the corresponding parts contain textures with similar statistical properties.

Robustness to noise was examined using the images in Fig. 11(b), where significant white noise ( $\mu = 0$  and  $\sigma = 40$ ) was added. The images were successfully registered using the rotation mode and the numerical results in Table II establish that when the algorithm converges, it achieves high accuracy and no consistent misalignments (local minima) were observed. The scale/rotation mode was found to be less stable numerically, failing to align the images in Fig. 11(b). When the algorithm converged, convergence was achieved within 3–4 iterations for all image sets taking 5–7 s on a 1.5-MHz P4 computer for a C++ implementation.

The images in Fig. 12(a) were used to test the estimation of large scaling factors. Prior techniques [13], [16] were reported to recover scalings up to a factor of 2, while the proposed algorithm successfully recovered scalings factors up to 4. The numerical results are presented in Table III. We were not able to consistently recover scalings larger than 4. In Fig. 12(b), images with large nonoverlapping supports were registered. The scale/rotation mode was found to be less stable than the rotation-only mode. We conclude that the proposed algorithm is able to register images with scalings up to 4 with good accuracy. Within this range, any rotation angle can be estimated.

## VII. CONCLUSION

In this paper, we proposed an FFT-based image registration algorithm, which was shown to be able to recover large rotations and scaling factors. This algorithm which uses the PPFFT [26] enhance the current state-of-the-art image registration algorithms. The overall complexity is dominated by FFT operations and is  $O(n^2 \log n)$ . The algorithm can achieve close to real-time performance by using machine-specific optimized FFT implementations which are widely available.

## ACKNOWLEDGMENT

The authors would like to thank the associate editor and the anonymous reviewers for providing helpful comments and suggestions that were used in preparing the final manuscript.

## REFERENCES

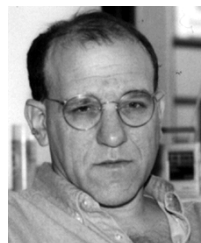
- [1] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 497–501, Mar. 2000.
- [2] A. Tekalp, *Digital Video Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [3] M. Irani and S. Peleg, "Motion analysis for image enhancement: Resolution, occlusion and transparency," *J. Vis. Commun. Image Rep.*, vol. 4, no. 4, pp. 324–335, Dec. 1993.
- [4] S. Man and R. W. Picard, "Virtual bellows: Constructing high quality stills from video," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 13–16, 1994.
- [5] R. Szeliski, "Image mosaicking for tele-reality applications," CRL 94/2, 1994.
- [6] M. Irani, P. Anandan, and S. Hsu, "Mosaic based representations of video sequences and their applications," in *Proc. Int. Conf. Computer Vision*, 1995, pp. 605–611.
- [7] M. Unser, P. Thvenaz, and L. Yaroslavsky, "Convolution-based interpolation for fast, high-quality rotation of images," *IEEE Trans. Image Process.*, vol. 4, no. 10, pp. 1371–1381, Oct. 1995.
- [8] S. Kruger and A. Calway, "Image registration using multiresolution frequency domain correlation," in *Proc. British Machine Vision Conf.*, Sep. 1998, pp. 316–325.
- [9] D. J. Fleet, "Disparity from local weighted phase-correlation," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, San Antonio, TX, 1994, pp. 48–56.
- [10] C. D. Kuglin and D. C. Hines, "The phase correlation image alignment method," in *Proc. IEEE Conf. Cybernetics and Society*, Sep. 1975, pp. 163–165.
- [11] H. Foroosh, J. B. Zerubia, and M. Berthod, "Extension of phase correlation to subpixel registration," *IEEE Trans. Image Process.*, vol. 11, no. 3, pp. 188–200, Mar. 2002.
- [12] P. Milanfar, "Projection-based, frequency-domain estimation of superimposed translational motions," *J. Opt. Soc. Amer.*, vol. 13, no. 11, pp. 2151–2162, Nov. 1996.
- [13] —, "Two dimensional matched filtering for motion estimation," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 438–443, Mar. 1999.
- [14] Y. Chou and H. Hang, "A new motion estimation method using frequency components," *J. Vis. Commun. Image Rep.*, vol. 8, no. 1, pp. 83–96, 1997.
- [15] H. Stone, M. Orchard, and E. Chang, "Subpixel registration of images," in *Proc. 33rd Asilomar Conf. Signals, Systems, and Computers*, Oct. 1999.
- [16] S. Reddy and B. N. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *IEEE Trans. Image Process.*, vol. 3, no. 8, pp. 1266–1270, Aug. 1996.
- [17] A. Taberner, J. Portilla, and R. Navarro, "Duality of log-polar image representations in the space and spatial-frequency domains," *IEEE Trans. Signal Process.*, vol. 47, no. 9, pp. 2469–2479, Sep. 1999.
- [18] L. Lucchese, "A frequency domain technique based on energy radial projections for robust estimation of global 2-D affine transformations," *Comput. Vis. Image Understanding*, vol. 82, no. 1, pp. 72–116, Apr. 2001.
- [19] S. Derrode and F. Ghorbel, "Robust and efficient Fourier-Mellin transform approximations for gray-level image reconstruction and complete invariant description," *Comput. Vis. Image Understanding*, vol. 83, no. 1, pp. 57–78, Jul. 2001.
- [20] G. Wolberg and S. Zokai, "Robust image registration using log-polar transform," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, BC, Canada, Sep. 2000.
- [21] Q. Chen, M. Defrise, and F. Deconinck, "Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 12, pp. 1156–1168, Dec. 1994.
- [22] S. A. Kruger and A. D. Calway, "Image registration using multiresolution frequency domain correlation," in *Proc. British Machine Vision Conf.*, 1998.
- [23] B. Porat, *A Course in Digital Signal Processing*. New York: Wiley, 1997.
- [24] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1998.
- [25] J. Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Santa Barbara, CA, Oct. 1997, pp. 536–539.
- [26] A. Averbuch, D. L. Donoho, R. R. Coifman, and M. Israeli, "Fast slant stack: A notion of radon transform for data in cartesian grid which is rapidly computable, algebraically exact, geometrically faithful and invertible," *SIAM Sci. Comput.*, to be published.
- [27] A. Averbuch, D. Donoho, and Y. Shkolnisky, "The 2-D discrete radon transform," *Fourier Anal. Applicat.*, to be published.
- [28] D. H. Bailey and P. N. Swartztrauber, "The fractional Fourier transform applications," *SIAM Rev.*, vol. 33, no. 3, pp. 389–404, Sep. 1991.
- [29] L. R. Rabiner, R. W. Schaffer, and C. M. Rader, "The chirp z-transform algorithm," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 86–92, Jun. 1969.
- [30] R. R. Coifman, A. Averbuch, D. L. Donoho, M. Israeli, and Y. Shkolnisky, "Fast slant stack: A notion of radon transform for data in cartesian grid which is rapidly computable, algebraically exact, geometrically faithful and invertible," *SIAM Sci. Comput.*, to be published.



**Yosi Keller** received the B.Sc. degree in electrical engineering from the Technion—Israel Institute of Technology, Haifa, in 1994 and the M.Sc. and Ph.D. degrees in electrical engineering from Tel Aviv University, Tel Aviv, Israel, in 1998 and 2003, respectively.

From 1994 to 1998, he was an R&D Officer with the Israeli Intelligence Force. He is currently a Visiting Assistant Professor with the Department of Mathematics, Yale University, New Haven, CT. His research interests include motion estimation, video

analysis, image restoration, and statistical pattern analysis.



**Amir Averbuch** was born in Tel Aviv, Israel. He received the B.Sc. and M.Sc. degrees in mathematics from the Hebrew University, Jerusalem, Israel, in 1971 and 1975, respectively, and the Ph.D. degree in computer science from Columbia University, New York, in 1983.

From 1966 to 1970 and from 1973 to 1976, he served in the Israeli Defense Forces. From 1976 to 1986, he was a Research Staff Member, Department of Computer Science, IBM T. J. Watson Research Center, Yorktown Heights, NY. 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).

**Moshe Israeli** was born in Bulgaria. He received the B.Sc. degree (summa cum laude) and the M.Sc. degree (with distinction) in aeronautical engineering from the Technion—Israel Institute of Technology, Haifa, Israel, and he received the Ph.D. degree in applied mathematics from the Massachusetts Institute of Technology (MIT), Cambridge, in 1971.

From 1971 to 1973 and from 1977 to 1978, he was with MIT. From 1985 to 1987, he was with Princeton University, Princeton, NJ. He has been a Consultant to Rafael, Haifa; Flow Research, Inc., Seattle, WA; and Cambridge Hydrodynamics, Inc., Princeton. Since October 1987, he has held the Callner-Miller Chair of Computer Science at the Technion—Israel Institute of Technology. His current research activities include multiresolution analysis, signal and image processing, acoustic and electromagnetic scattering, and the numerical solution of differential and integral equations.