The Angular Difference Function and its application to Image Registration

Y. Keller[2], Y. Shkolnisky[1] ,A. Averbuch[1]

[1]School of Computer Science,

Tel Aviv University, Tel Aviv 69978, Israel

[2]Math department

Yale University, Connecticut, USA

Phone: +1-203-432-4345

Email: yosi.keller@yale.edu

Submitted to the IEEE Transactions on PAMI

May 2004

Abstract

An important problem in image registration is to estimate large motions without prior knowledge. 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 pseudo-polar Fourier transform. Unlike other Fourier based registration schemes, the suggested approach does not require any interpolation. Thus, it is more accurate and significantly faster.

Keywords: Global motion estimation, Sub-pixel registration, Gradient methods, image alignment

EDICS Category={2-ANAL, 2-MOTD}

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], 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 on images that have small misalignment. 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 pseudo-polar 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 by using the ADF.

A notion similar to the ADF was first introduced in [16] and [17] for rotation estimation and symmetry detection. The main idea of the algorithms in [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 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 pseudo-polar 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

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

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 Eq. (2.2) we get

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

or equivalently,

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

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

$$\mathrm{Corr}(x, y) \triangleq \mathcal{F}^{-1}\left(\frac{\widehat{I}_2(\omega_x, \omega_y)}{\widehat{I}_1(\omega_x, \omega_y)}\right) = \delta(x + \Delta x, y + \Delta y), \tag{2.5}$$

which means that $\mathrm{Corr}(x, y)$ is different from zero 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, 18]. Formally,

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

where $*$ denotes the complex conjugate. This scheme is proven to robustly estimate large translations, even when the corresponding overlap between the registered images is as small as 30% of the smallest image size [14]. There are no smoothness assumptions on the registered functions, and therefore non-smooth and noisy functions (such as the 2D DFT coefficients) can be accurately registered. Shekarforoush et. al. [18] extended the phase-correlation based algorithm to subpixel accuracy by analyzing the shape of $\widetilde{\text{Corr}}\,(\omega_x, \omega_y)$, given in Eq. (2.7), around its maximum.

A different approach for phase correlation based translation estimation is given in [13, 14, 19]. From Eq. (2.7) we get that Δx and Δy satisfy the equation

$$\imath \log\left(\widetilde{\text{Corr}}\,(\omega_x, \omega_y)\right) = \omega_x \Delta x + \omega_y \Delta y,$$

which can be solved using linear regression. This approach may be inaccurate ([18]) 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 modelling 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, 20]. Let $I_1(x, y)$ and $I_2(x, y)$ be two images such that

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

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 Eq. (2.8) in polar coordinates is

$$\widehat{I_2}\left(r,\theta\right) = e^{\imath(\omega_x \Delta x + \omega_y \Delta y)} \widehat{I_1}(r, \theta + \theta_0). \tag{2.9}$$

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

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

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 Eq. (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 [21], and interpolation of the 2D DFT of the image in the Fourier domain [9, 20]. 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.

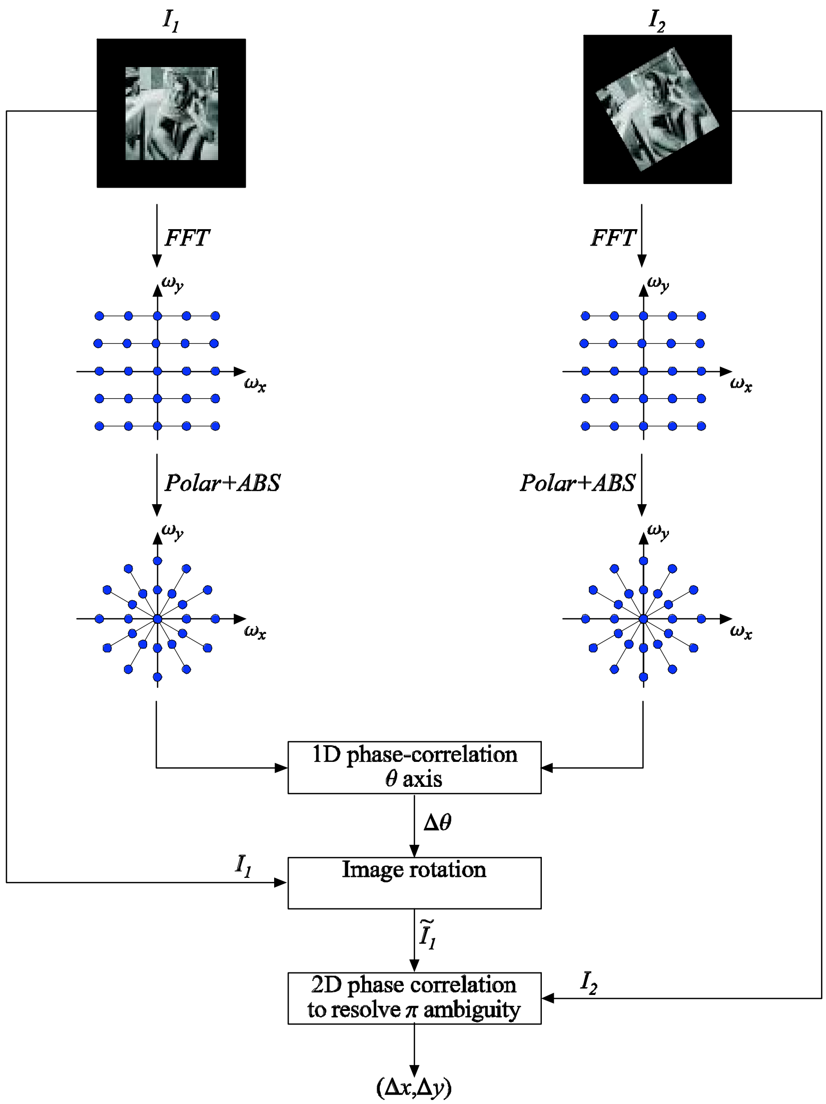


Figure 1: The flow of a FFT based image registration: 1. The magnitudes of the polar DFTs are approximated by interpolating the magnitudes of the 2D FFTs. 2. The rotation angle $\Delta\theta$ is recovered using 1D phase-correlation on the θ axis. 3. One of the input images is rotated by the angle $\Delta\theta$.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

3 The pseudo-polar 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

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

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 Eq. (3.1) has the form

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

$k, l = -\frac{M}{2}, \ldots, \frac{M}{2} - 1$, 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 Eq. (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 Eq. (3.1) on the grid

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

for which the Fourier transform in Eq. (3.1) has the form

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

The grid that is given by Eq. (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.

In section 3.1 we give the definition of the pseudo-polar 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.

3.1 Definition of the pseudo-polar Fourier transform

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

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

where

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

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

See Figs. 2a and 2b for an illustration of the sets P_1 and P_2. The pseudo-polar grid P is illustrated in Fig. 3. As can be seen from Figs. 2a and 2b, k serves as a "pseudo-radius" and l serves as a "pseudo-angle". The resolution of the pseudo-polar grid is $N+1$ in the angular direction and $M = 2N+1$ in the radial direction. Using (r, θ) representation, the pseudo-polar grid is given by

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

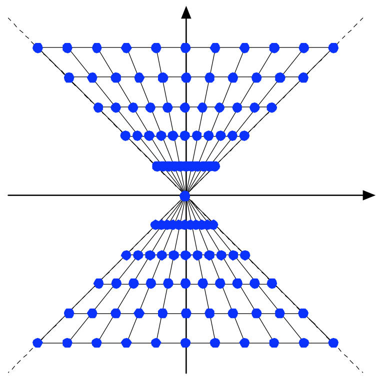
$$r_k^1 = k\sqrt{4\left(\frac{l}{N}\right)^2 + 1}, \quad r_k^2 = k\sqrt{4\left(\frac{l}{N}\right)^2 + 1}, \tag{3.11}$$

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

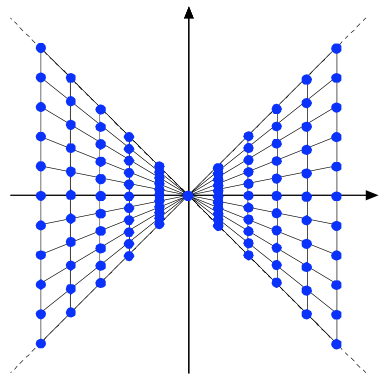
where $k = -N, \ldots, N$ and $l = -N/2, \ldots, N/2$. We define the pseudo-polar Fourier transform as the samples of the Fourier transform $\widehat{I}$, given in Eq. (3.1), on the pseudo-polar grid P, given in Eq. (3.7). Formally, the pseudo-polar Fourier transform $\widehat{I}_{PP}^j$ $(j = 1, 2)$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

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

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



The pseudo-polar sector P_1



The pseudo-polar sector P_2

Figure 2: The pseudo-polar sectors P_1 and P_2

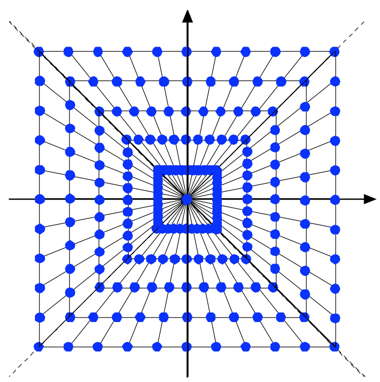


Figure 3: The pseudo-polar grid $P = P_1 \cup P_2$

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

As we can see in Fig. 3, for each fixed angle l, the samples of the pseudo-polar 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 Eq. (3.12).

Two important properties of the pseudo-polar Fourier transform are that it is invertible and that both the forward and inverse pseudo-polar 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 re-gridding or interpolation.

3.2 The fractional Fourier transform

The algorithm for computing the pseudo-polar Fourier transform is based on the fractional Fourier transform (FRFT). The fractional Fourier transform [22], with its generalization given by the Chirp Z-transform [23], 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 $\alpha \in \mathbb{R}$.

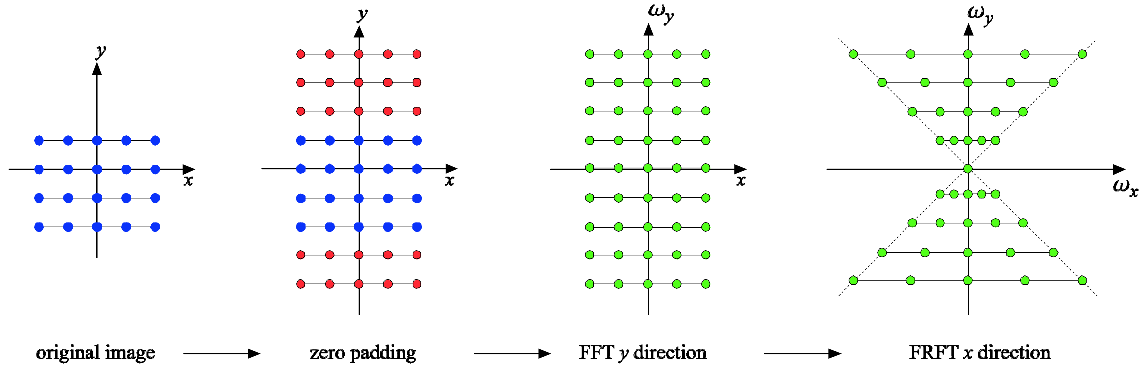


Figure 4: Computing $\widehat{I}^1_{PP}$

3.3 Computing the pseudo-polar Fourier transform

By using the fractional Fourier transform we compute the pseudo-polar Fourier transform $\widehat{I}^1_{PP}$, given in Eq. (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. 4. 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 pseudo-polar 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. 5a). We denote by $g_2(x)$ the flipped and shifted version of $f_2(x)$ (see Fig.5b)

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

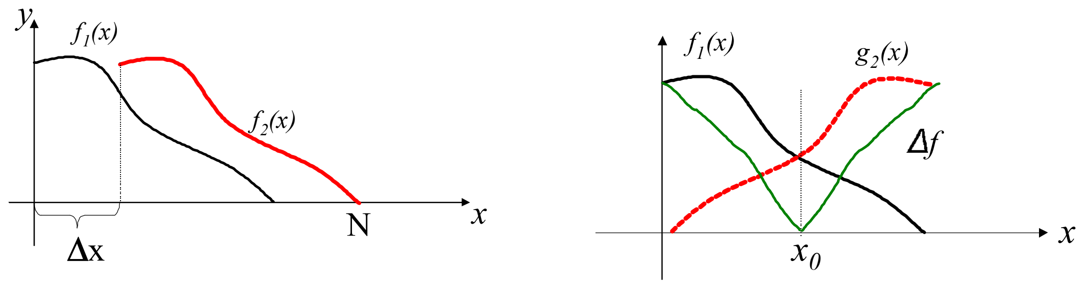
We define the difference function (DF) Δf by

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

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)

Figure 5: 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 $\Delta M\left(\theta\right)$, 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 Eq. (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

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

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

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

Note that since the 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 Eq. (5.3), is accurately implemented by flipping the input image along the x or y axes. Mathematically,

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

The PPFT, presented in section 3, is used to derive a fast and accurate algorithm for computing Ω. The PPFT evaluates the DFT of a given image on the pseudo-polar grid. For each angle θ in the pseudo-polar 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 Eq. (5.3) using numerical integration

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

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

5. Compute Ω by

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

where K is the size of the pseudo-polar grid. Equation (5.10) is the discrete equivalent of Eq. (5.6). Two samples of ΔM^d,given in Eq. (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 Eq. (5.9) with the normalized correlation [24]

$$\Delta M_N^d(\theta_i) = \frac{\sum\limits_{0 \le r_j \le 1} \left(\overline{M_1^d}(r_j, \theta_i) - \overline{M_2^d}(r_j, -\theta_i) \right)^2}{\sigma_r\left(M_1^d\right) \sigma_r\left(M_2^d\right)}, \qquad \theta_i \in [0, \pi] \tag{5.11}$$

where

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

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

$$\sigma_r\left(M_k^d\right) = \sqrt{\frac{1}{j_{\max}} \sum_{0 \le r_j \le 1} \left(M_k^d(r_j, \theta_i) - \overline{M_k^d}(r_j, \theta_i) \right)^2}.$$

Similarly to Eq. (5.10), Ω_N, the normalized angular difference function, is given by

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

Figure 6 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.

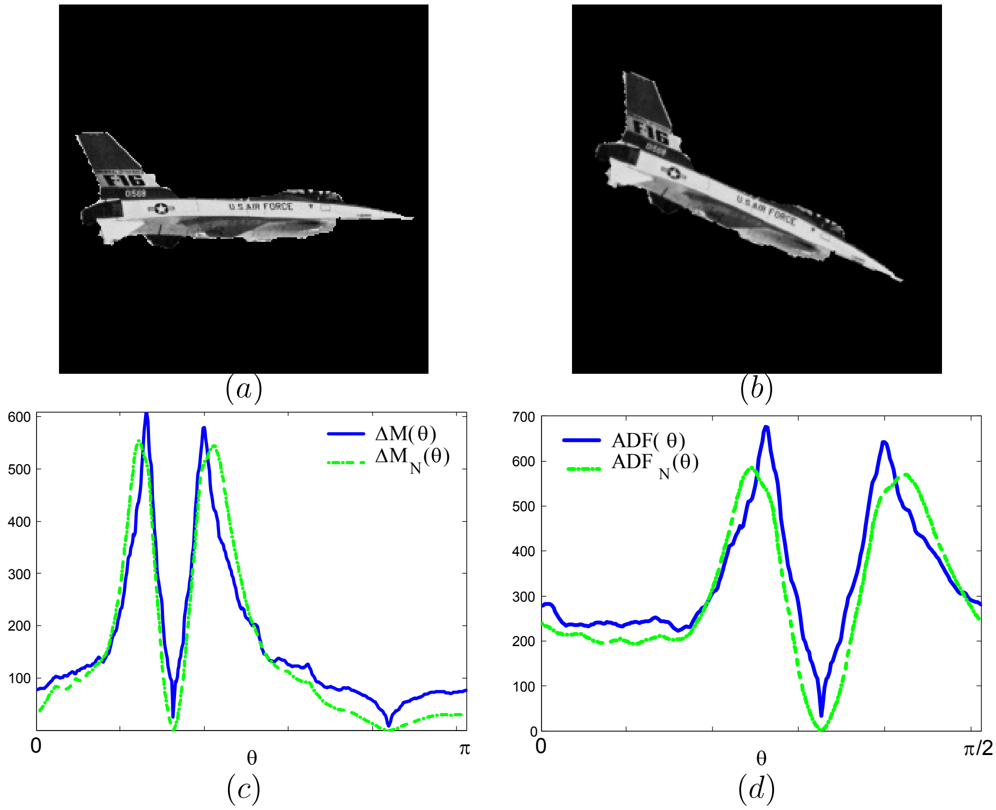


Figure 6: The difference function and the angular difference function of rotated sample images. (a), (b) The input images . (c) The difference function ΔM^d and the normalized difference function ΔM_N^d of (a) and (b). Two minima with an offset of $\frac{\pi}{2}$ can be observed in ΔM^d and ΔM_N^d. (d) 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

$$\Delta\theta = 2 \cdot \theta\left(i_0\right) \tag{5.13}$$

where

$$i_0 = \arg\min_i\{\Omega\left(i\right)\}, \tag{5.14}$$

and $\theta\left(i\right)$ is given by Eq. (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. This complexity if dominated by 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 computational complexity of most other registration schemes [9, 20] is related to the complexity of interpolating 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. 7) 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.

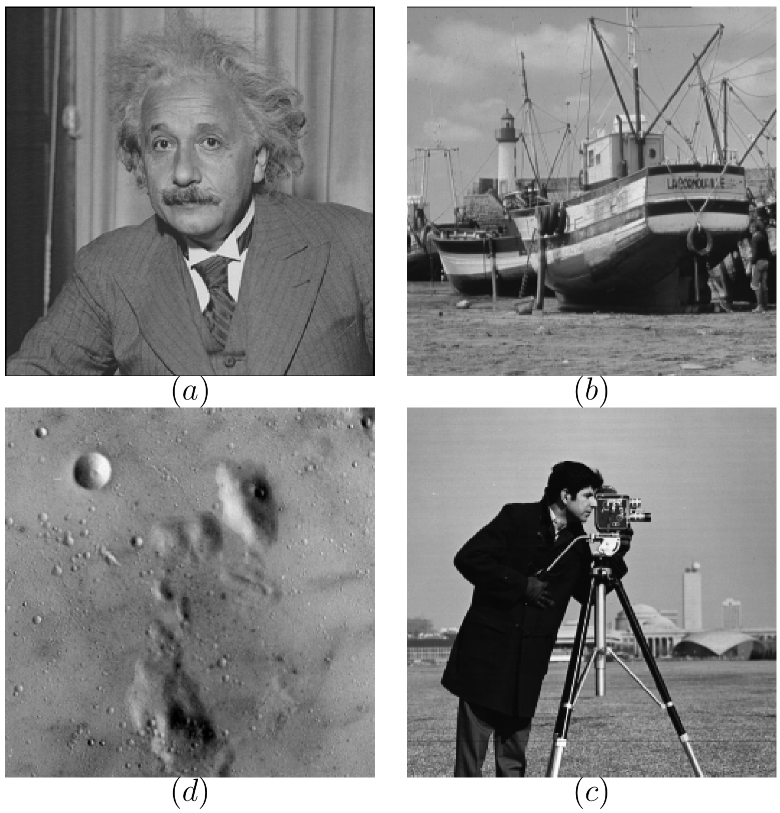


Figure 7: Images used for registration accuracy estimation.

The robustness to noise was tested using the F16 and Lena images, given in Figs. 6a and 9c. 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

	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	$0.83°$	$2.13°$	$0.79°$	$1.67°$

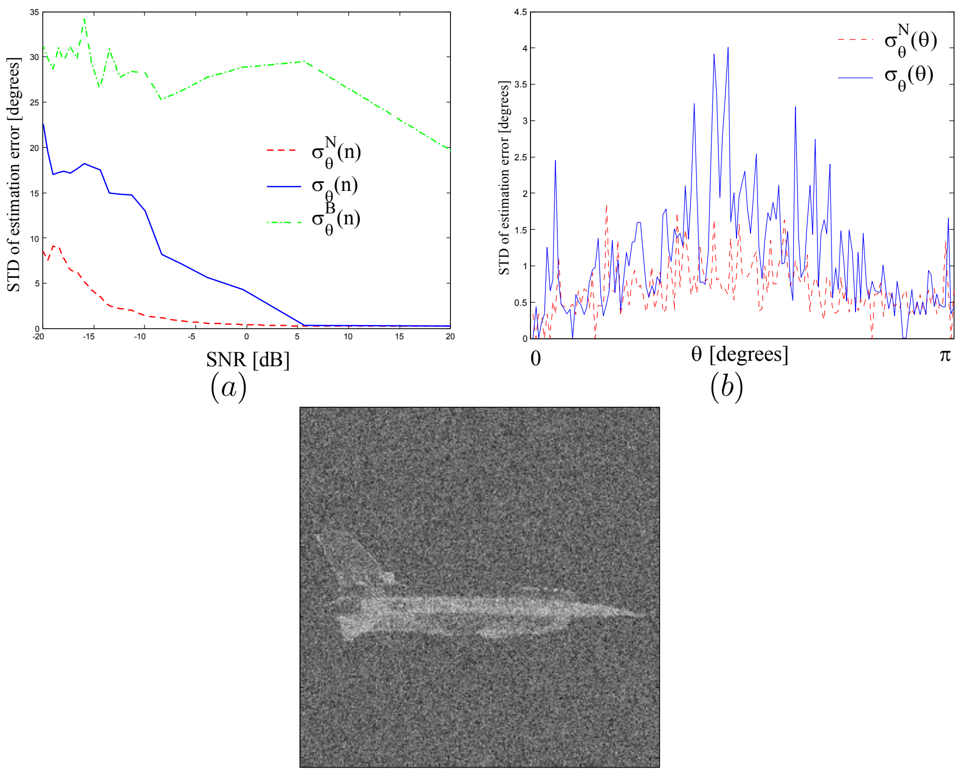
Table 1: Registration results for the images in Fig. 7. The errors are estimated over a range of rotations and translations. The mean error is similar for Ω and Ω_N, while there is an improvement in the maximal error.

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. The implementation of the proposed algorithm uses a standard implementation of the phase correlation algorithm [8, 9].

Figures 8a and 9a 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 non-noisy input images the registration error of the proposed algorithm is less than $1°$, which corresponds to twice the average angular resolution of the PPFT, which is approximately $0.7°$. The accuracy of the registration slightly degrades as the noise power increases. In particular, for SNRs up to about $9dB$, the estimation errors remain within $0.1°$; below that threshold and down to about $0dB$, the errors become of the order of $0.5°$. At about $-10dB$, the error is around $2°$. Below this SNR and down to about $-15dB$, the errors become larger. The noisy images at $SNR = -8dB$ are depicted in Figs. 8c and 9d, where the F16 and Lena are hardly visible. Ω_N 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°. To conclude, the algorithm we propose successfully registers extremely noisy images, where feature-based techniques usually fail.



(c)

Figure 8: 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°.

The dependence between the registration accuracy and the rotation angle is studied in Figs. 8b and 9b. 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. 9b) are unrelated to θ, while for the F16 image (Fig. 8b), the largest error is observed around $\theta = 90$. This is attributed to the difference in the spectral content of the images (see Fig. 10). 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

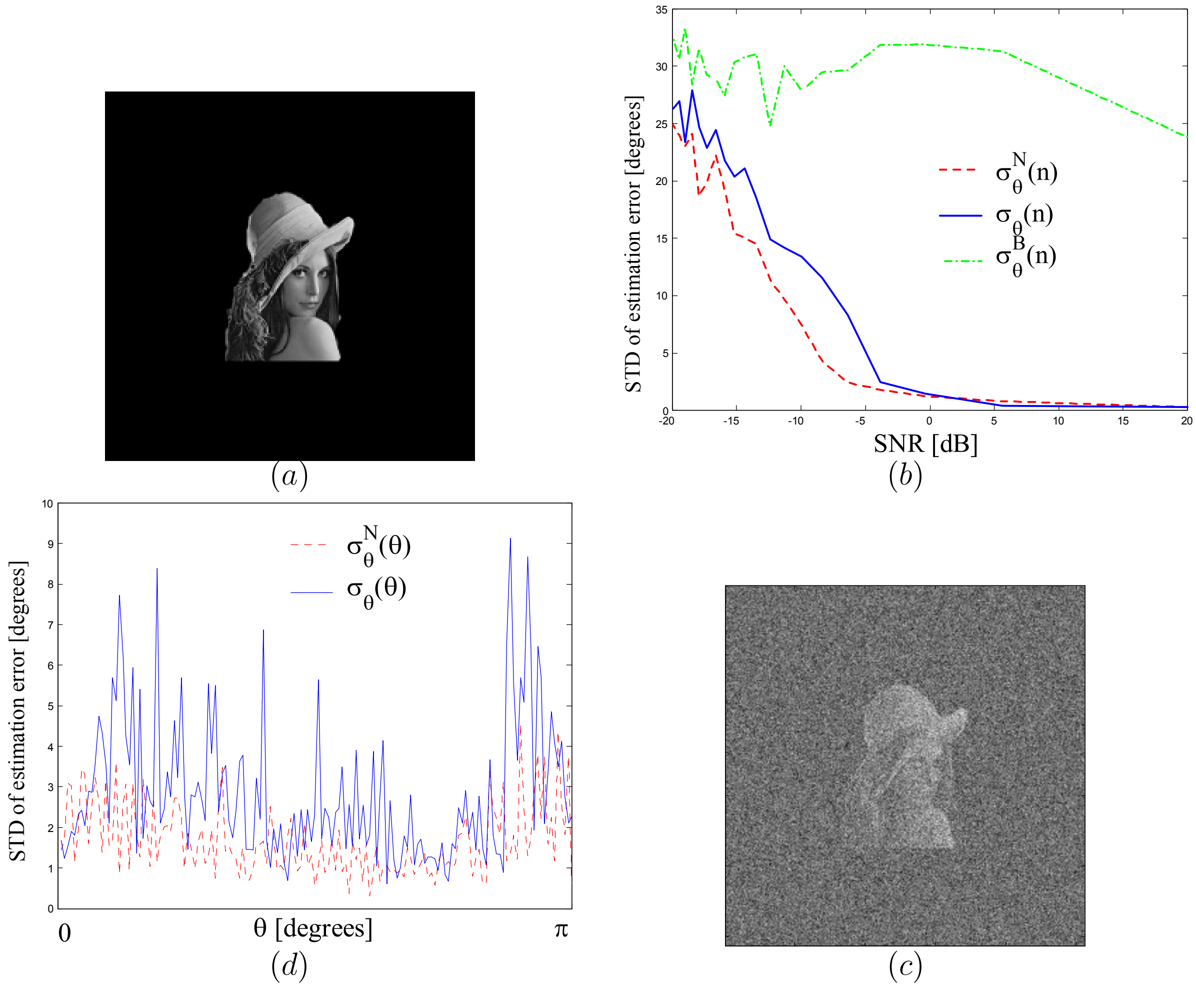


Figure 9: Registration accuracy for the Lena image. (a) The STD of the registration error (in degrees) as a function of the noise. σ_θ and σ_θ^N are the STD of Ω and Ω_N, respectively. σ_θ^B is the STD of the bilinear interpolation based scheme given in [9]. For high SNR, the performance of Ω and Ω_N 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 $3°$.

spread over $[0, \pi]$.

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

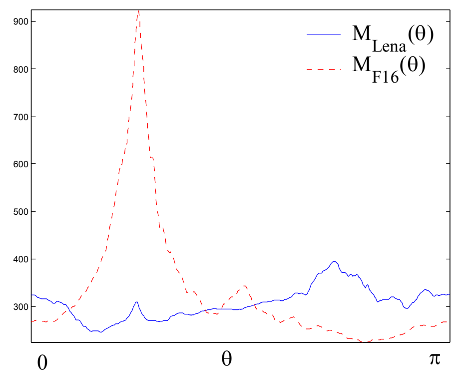


Figure 10: The spectral content of the F16 and Lena images. The spectral energy of F16 is concentrated in $\left[0, \frac{\pi}{2}\right]$, while the spectral energy of Lena is evenly spread over $[0, \pi]$.

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 pseudo-polar 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 application like symmetry detection and 3D image registration, which are currently being investigated.

References

[1] F. Dufaux and J. Konrad, “Efficient, robust, and fast global motion estimation for video coding,” *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 497–501, March 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,” *Journal of Visual Communication and Image Representation*, vol. 4, no. 4, pp. 324–335, December 1993.

[4] S. Mann and R. Picard, “Virtual bellows: constructing high quality stills from video.” Austin, TX: IEEE Int. Conf. Image Processing, November 13-16 1994, pp. 363–367.

[5] R. Szeliski, “Image mosaicking for tele-reality applications,” in *Proc. of IEEE Workshop on Applications of Computer Vision*, May 1994., pp. 44–53.

[6] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, “Mosaic based representations of video sequences and their applications,” *Signal Processing: Image Communication*, vol. 8, no. 4, pp. 327–351, May 1996.

[7] G. Wolberg and S. Zokai, “Robust image registration using log-polar transform,” *IEEE Intl. Conference on Image Processing*, vol. 1, pp. 493 – 496, September 2000.

[8] C. D. Kuglin and D. C. Hines, “The phase correlation image alignment method,” *IEEE Conference on Cybernetics and Societ*, pp. 163–165, September 1975.

[9] S. Reddy and B. N. Chatterji, “An fft-based technique for translation, rotation, and scale-invariant image registration,” *IEEE Trans. on Image Processing*, vol. 3, no. 8, pp. 1266–1270, August 1996.

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

[11] P. Milanfar, “Projection-based, frequency-domain estimation of superimposed translational motions,” *Journal of the Optical Society of America: A, Optics and Image Science*, vol. 13, no. 11, pp. 2151–2162, November 1996.

[12] ——, “Two-dimensional matched filtering for motion estimation,” *IEEE Trans. Image Processing*, vol. 8, no. 3, pp. 438–443, March 1999.

[13] Y. Chou and H. Hang, “A new motion estimation method using frequency components,” *Journal of Visual Communication 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. on Geoscience and Remote Sensing*, vol. 39, no. 10, pp. 2235–2243, October 2001.

[15] Y. Shkolnisky, A. Averbuch, D. Donoho, 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,” *to appear in SIAM Scientific Computing*.

[16] L. Lucchese and G. Cortelazzo, “A noise-robust frequency domain technique for estimating planar roto-translations,” *IEEE Transactions on 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," in *of Int'l Conference on Image Processing (ICIP)*, vol. II, Rochester, NY, September 2002, pp. 793–796.

[18] H. Foroosh, J. Zerubia, and M. Berthod, "Extension of phase correlation to subpixel registration," *IEEE Trans. on Image Processing*, vol. 11, no. 3, pp. 188 –200, March 2002.

[19] D. J. Fleet, "Disparity from local weighted phase-correlation," *The IEEE International Conference on Systems, Man, and Cybernetics*, pp. 48–56, 1994.

[20] Q. Chen, M. Defrise, and F. Deconinck, "Symmetric phase-only matched filtering of fourier-mellin transforms for image registration and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 12, pp. 1156–1168, December 1994.

[21] 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: CVIU*, vol. 83, no. 1, pp. 57–78, July 2001.

[22] D. H. Bailey and P. N. Swarztrauber, "The fractional fourier transform and applications," *SIAM Review*, vol. 33, no. 3, pp. 389–404, September 1991.

[23] L.R. Rabiner, R.W. Schafer, and C.M. Rader, "The chirp z-transform algorithm," *IEEE Transcations on Audio ElectroScoustics*, vol. AU, no. 17, pp. 86–92, June 1969.

[24] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Reading, Massachusetts: Addison-Wesley, 1992.