

An Adaptable-Multilayer Fractional Fourier Transform Approach for Image Registration

Wei Pan, Kaihuai Qin, and Yao Chen

Abstract—A novel adaptable accurate way for calculating the polar FFT and the log-polar FFT is developed in this paper, namely, *Multilayer Fractional Fourier Transform* (MLFFT). MLFFT is a necessary addition to the pseudopolar FFT for the following reasons: It has lower interpolation errors in both polar and log-polar Fourier transforms, it reaches better accuracy with the nearly same computing complexity as the pseudopolar FFT, and provides a mechanism to increase the accuracy by increasing the user-defined computing level. This paper demonstrates both MLFFT itself and its advantages theoretically and experimentally. By emphasizing applications of MLFFT in image registration with rotation and scaling, our experiments suggest two major advantages of MLFFT: 1) Scaling up to 5 and arbitrary rotation angles or scaling up to 10 without rotation can be recovered by MLFFT, while, currently, the result recovered by the state-of-the-art algorithms is the maximum scaling of 4. 2) No iteration is needed to recover large rotation and scaling values of images by MLFFT; hence, it is more efficient than the pseudopolar-based FFT methods for image registration.

Index Terms—Image processing and computer vision, signal processing, computation of transforms, pattern matching.

1 INTRODUCTION

IMAGE registration is an old yet still hot topic [1] and it is involved in many different research areas [2] from image mosaic [22] to image compression [20] and video enhancement [21], to name a few. Current methods are mainly pixel gradient methods [5], [10], correlation methods [25], Fourier transform methods [2], [3], [7], and feature-based methods [27], [28].

Compared with other modern spatial image registration techniques such as matching key points with the SIFT method [27], [28], the log-polar mapping followed by a Levenberg-Marquardt (LM) nonlinear least-squares solution [6], the boot-strap method [26], etc. Fourier-based spectral methods generally have poor performance and limited applications. For instance, Fourier-based methods cannot support large scales with arbitrary rotations (i.e., over 4x), while scholars in [6], [26] accomplished such registrations with ease. Besides, Fourier-based methods cannot recover perspective images. However, it must be mentioned here that there are a few cases in which Fourier-based methods demonstrate advantages over the traditional spatial registration algorithms:

1. Fourier-based methods are robust to noise and time varying illumination disturbances.
2. They have a low computational complexity and take a fixed period of time in registering any images. On the contrary, most spatial methods have high computational costs and the time used in registering

two images with spatial methods is unpredictable, depending on the images to be registered.

3. They are easy to implement and parallelize.
4. They could be used for generating an initial value for nonlinear optimization based spatial registration methods (e.g., in [10]).

We will explain the above three advantages in detail later.

The theory background of Fourier image registration is described in [3] and [9]. Two major techniques in Fourier-based image registration are the phase correlation and the log-polar transform. The translation between two images can be determined by the phase correlation method, which has excellent accuracy and robustness [9] and recently was extended to the subpixel translation level [7]. We give a brief introduction of the phase correlation first.

Consider two infinite signals $f_1(x, y)$ and $f_2(x, y)$ and $f_2(x, y)$ is a translated replica of $f_1(x, y)$:

$$f_2(x, y) = f_1(x - x_0, y - y_0). \quad (1)$$

Their Fourier transforms are related by

$$F_2(u, v) = e^{-j2\pi(ux_0 + vy_0)} \times F_1(u, v). \quad (2)$$

The cross-power spectrum holds that

$$\frac{F_1(u, v)F_2^*(u, v)}{|F_1(u, v)F_2^*(u, v)|} = e^{-j2\pi(ux_0 + vy_0)}. \quad (3)$$

Since the inverse transform of $e^{-j2\pi(ux_0 + vy_0)}$ is a Dirichlet function at (x_0, y_0) , we can simply determine the (x_0, y_0) by finding the pulse peak on the inverse Fourier transform of the cross-power spectrum.

To recover rotation and scaling parameters, a common way is the log-polar Fourier transform; let I_1 and I_2 be two images, and I_2 be a translated, rotated, and zoomed replica of I_1 , that is,

• W. Pan is with the Computer Science Department, Dartmouth College, 6211 Sudikoff, Hanover, NH 03755. E-mail: pway@cs.dartmouth.edu.

• K. Qin and Y. Chen are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P.R. China. E-mail: qkh-dcs@tsinghua.edu.cn, alenda.chen@gmail.com.

Manuscript received 8 Dec. 2006; revised 4 Oct. 2007; accepted 11 Jan. 2008; published online 1 Apr. 2008.

Recommended for acceptance by F. Dellaert.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0856-1206. Digital Object Identifier no. 10.1109/TPAMI.2008.83.

$$I_2(x, y) = I_1(s(x \cos \theta_0 + y \sin \theta_0) + \Delta x, \\ s(-x \sin \theta_0 + y \cos \theta_0) + \Delta y), \quad (4)$$

where s , θ_0 , and $(\Delta x, \Delta y)$ are the scale factor, rotation angle, and translation parameters, respectively.

After applying the infinite continuous Fourier transform to I_1 and I_2 [3] and replacing ω_x and ω_y with the following equations:

$$\omega_x = r \cos \theta, \\ \omega_y = r \sin \theta, \quad (5)$$

(4) turns to

$$\hat{I}_2 = e^{j(\omega_x \Delta x + \omega_y \Delta y)} s^{-2} \hat{I}_1(s^{-1}r, \theta + \theta_0). \quad (6)$$

Supposing that the magnitudes of I_1 and I_2 are M_1 and M_2 , we have

$$M_2(r, \theta) = s^{-2} M_1(s^{-1}r, \theta + \theta_0). \quad (7)$$

Moreover, by applying the log operation to the radius r , we have

$$M_2(\log r, \theta) = s^{-2} M_1(\log r - \log s, \theta + \theta_0). \quad (8)$$

Then, the phase correlation is again applied on M_1 and M_2 to determine s and θ_0 .

In real applications, the discrete Fourier transform (DFT) is used instead of the infinite continuous Fourier transform in (4)-(8). Thus, the calculation can be processed by the Fast Fourier Transform (FFT).

Such an FFT registration algorithm is introduced in [3] and researchers succeeded in recovering any rotation at scales up to 2. With a more careful study on the error due to using DFT as an approximate method of the infinite continuous Fourier transform, Stone et al. suggested in [11] that a better filter and a window function should be applied to the original images, which could reduce the alias caused by rotation and recover scales up to 2.5 with any rotation.

The FFT algorithm's largest challenge is to evaluate the log-polar Fourier transform efficiently and accurately, and different ideas have been proposed in recent years to solve such a problem. Two methods were developed in the past to calculate polar or log-polar Fourier transforms [2]: the image warping [25] followed by a 2D FFT [16] and the interpolation directly in the 2D Cartesian FFT domain [3]. However, due to the interpolation error, FFT-based image registration methods cannot recover the values of rotation angles and scale factors accurately; it cannot even recover large scale factors and it often suffers from the false peak during the phase correlation. Although by using the Unequally-Spaced FFT (USFFT) [4] and related interpolation methods, the interpolation error can be reduced, USFFT is very slow for high accuracy and it is hard to choose the parameter of the oversampling rate S as well. The pseudopolar FFT was then proposed and tested in image registration [2]. It significantly decreases the interpolation error in calculating the polar Fourier transform with the same complexity as the 2D FFT and, under certain conditions [14], [2], even no interpolation is needed for the pseudopolar method. The pseudopolar FFT-based algorithm (PPFFT) suggested in [2] can recover scales

up to 4 by an iteration process. Though the pseudopolar is a tremendous improvement in the fast polar Fourier transform, the pseudopolar grid is still unsatisfying in image registration because it is not suitable for the log-polar Fourier transform (still needs interpolation [2]) and is unstable (fails in large scale factor cases [2]) and inaccurate (fails in an accurate rotation angle recovering) performance. To increase the accuracy of pseudopolar-based image processing, researchers in [14] and [19] proposed the angular difference algorithm to estimate the rotation angle of an image. The method in [14] is quite similar to the signature method described in [11] and avoids the interpolation step. However, this method is not reliable for certain images because it reduces 2D correlations to 1D correlations, which certainly causes a loss of the correlation information; it puts uncertainty in the rotation recovering because pseudopolar grids are not distributed angular-uniformly, and the most important thing is that it still cannot handle images scaled by a large scale factor.

The pseudo-log-polar FFT was proposed in [8]. For the pseudo-log-polar is still based on interpolation in calculating a pseudo-log-polar grid, it fails in reducing the interpolation error, which is also covered in this paper.

Our research started with implementing the above pseudopolar FFT-based registration methods. While we successfully apply the pseudopolar FFT to artificial images in Figs. 7, 8, and 10 as [2] suggests, it fails in real applications with actually hand-taken pictures (described in detail in Section 4). Why does the pseudopolar FFT work well in [2] and in our tests in Fig. 8? It is because these image pairs registered are derived from a single image, and as a result, the tests cannot fully testify the performance of the registration algorithm fairly as there are no noise, distortions and effects of other transforms (e.g., the perspective transform) in the corresponding images at all besides scaling, rotating, and translation. In addition, in real pictures, scenes originally outside the photo boundaries come into the photos when we zoom out the camera focus length, and these additional scenes turn to noises in registration. For real photos, it is unreasonable to suggest that pictures should be as perfect as these artificial test benchmarks used in [2] and our experiments, and FFT-based algorithms must be more robust. Under such a circumstance, we decided to search for a better way for image registration. Recall the phase correlation method in (1)-(3). Most unsuccessful registrations are due to the false peaks during the phase correlation step for recovering the scale factor and the rotating angle. However, in real applications, there will always be some other peaks in the results rather than one peak. It is difficult to tell false peaks from the true peak besides taking the peak that has the largest value as the true one; thus, FFT-based algorithm may be unsuccessful once unexpected large false peak values occur.

In Section 3, we will prove that the true peak value is related to SNR (defined in Section 3.2) of the log-polar magnitude spectrum in detail; thus, we will show limitations for Fourier-based methods in detail. However, it is important to mention here that there are four main factors that affect Fourier-based methods in image registration with only rotation and scaling transforms:

1. The rotation aliasing [11]. We use DFT to approach the infinite Fourier transform, but, in images with rotations, such approximating approach generates aliasing in the results.
2. The scaling overlapping. (We will discuss below.)
3. The interpolation error.
4. Other errors (due to perspective projections and distortions introduced by cameras).

Rotation aliasing is difficult to overcome; the scaling overlapping will be inevitable and critical for registering images with large-scale factors, as stated in Section 3.2. We also cannot expect real images to be free of distortions and perspective projections. Nevertheless, we realized that we can still improve the pseudo-log-polar technique and push it to its limit by reducing the interpolation error.

In this paper, a new technique for calculating both the polar and the log-polar Fourier transforms in two dimensions is presented, and it is easy to port the algorithm to a higher dimension. Our belief is that interpolation does not necessarily lead to inaccuracy and our novel algorithm has an interpolation process from a multilayer method to the real polar or log-polar grid, so we named it Multilayer Fractional Fourier Transform (MLFFT). This paper will demonstrate that MLFFT has advantages over the pseudo-polar-based image registration described in [2], [14] in the following aspects:

1. high accuracy in recovering large scale factors and large rotation angles,
2. adaptability, for different precision requirements,
3. working well with both the log-polar and the polar transforms,
4. easy implementation, and
5. fast and parallelizable computing with just serial fractional FFT algorithms.

This paper contains five sections: Section 1 introduces the previous work. MLFFT is described in Section 2. Some discussions on improving the registration precision of MLFFT are given in Section 3. All of the experimental results are in Section 4. Section 5 concludes this paper.

2 DESCRIPTION OF MLFFT

2.1 2D Discrete Fractional Fourier Transform

It is well known how DFT works. After an infinite continuous signal is sampled and windowed, its sampled Fourier transform in frequency domain can be calculated as follows [13]:

$$F(e^{j\omega}) = \sum_{n=0}^{N-1} f(n)e^{-j\omega n}. \quad (9)$$

It should hold that $0 \leq \omega \leq 2\pi$. Suppose $0 \leq k < N$ and we simply rewrite

$$F\left(e^{j\frac{2\pi k}{N}}\right)$$

as $F(k)$, then we have

$$F(k) = \sum_{n=0}^{N-1} f(n)e^{-j\frac{2\pi k}{N}n}. \quad (10)$$

Formula (9) is known as DFT. The fastest way to calculate DFT is FFT. We recommend FFTW [15] C library when FFT needs to be embedded in real programs. The FFT returns the value of $F(k)$. Note that if k is not an integer, (10) still holds.

Based on (10), the fractional Fourier transform is defined as follows, where α is an additional parameter to the typical Fourier transform:

$$F^\alpha(k) = \sum_{n=0}^{N-1} f(n)e^{-j\frac{2\pi k}{N}\alpha n}. \quad (11)$$

We introduce the centered fractional Fourier transform here, which is used in MLFFT instead of (11). Given a vector $f = f(x)$, $-N/2 \leq x \leq N/2$, the fractional Fourier transform is defined as

$$F^\alpha(k) = \sum_{n=-N/2}^{N/2} f(n)e^{-j\frac{2\pi k}{N+1}\alpha n}, (-N/2 \leq k \leq N/2). \quad (12)$$

1. When $\alpha = 1$, the fractional Fourier transform is equal to DFT, thus we get the values of the $N + 1$ frequencies that are distributed uniformly in $-\pi \leq \omega \leq \pi$ in the frequency domain.
2. When $0 < \alpha < 1$, the fractional Fourier transform returns the values of the $N + 1$ frequencies that are scattered uniformly in $-\alpha\pi \leq \omega \leq \alpha\pi$ in the frequency domain.

Based on the chirp-Z transform [12], [18], the computational complexity of the fractional Fourier transform is three times that of the 1D FFT, within the same magnitude.

Now, we define the 2D fractional Fourier transform.

Given an $(N + 1) \times (N + 1)$ discrete signal

$$f(n_1, n_2), -N/2 \leq n_1, n_2 \leq N/2,$$

where N is an even number, the 2D fractional Fourier transform is defined as follows:

$$F^\alpha(k_1, k_2) = \sum_{n_1=-N/2}^{N/2} \sum_{n_2=-N/2}^{N/2} f(n_1, n_2)e^{-j\frac{2\pi n_1}{N+1}\alpha k_1 - j\frac{2\pi n_2}{N+1}\alpha k_2}. \quad (13)$$

Calculating the 2D fractional Fourier transform is similar to calculating a 2D FFT, that is, first, applying a 1D fractional Fourier transform to each row, then applying a 1D fractional Fourier transform to each column.

To help understand the 2D fractional Fourier transform, we have the following conclusion by combining (9) and (13). Suppose F is the Fourier transform of a 2D discrete signal, we have the following:

Theorem 1. For $0 < \alpha \leq 1$, we have

$$F(\alpha u, \alpha v) = F^\alpha(u, v). \quad (14)$$

Proof. Expanding both sides of (14) using (11) and (13) in two dimensions [13], we have

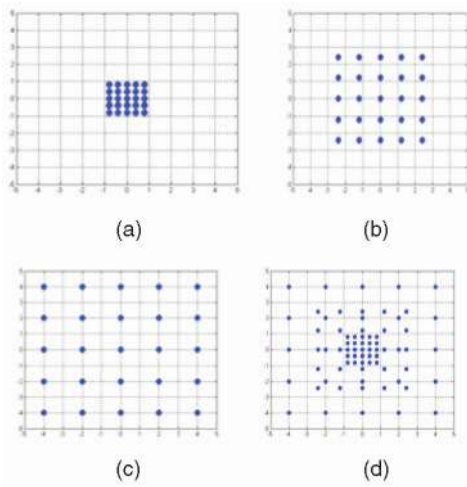


Fig. 1. The construction of an MLFFT grid with $\{Cut\} = \{0.1, 0.6, 1\}$. (a) P_1 grid. (b) P_2 grid. (c) P_3 grid. (d) P grid.

$$F^\alpha(u, v) = \sum_{n_1=-N/2}^{N/2} \sum_{n_2=-N/2}^{N/2} f(n_1, n_2) e^{-j\frac{2\pi n_1}{N}\alpha u - j\frac{2\pi n_2}{N}\alpha v},$$

$$F(\alpha u, \alpha v) = \sum_{n_1=-N/2}^{N/2} \sum_{n_2=-N/2}^{N/2} f(n_1, n_2) e^{-j\frac{2\pi n_1}{N}\alpha u - j\frac{2\pi n_2}{N}\alpha v}.$$

It is clear that the left of (14) is equal to the right. \square

2.2 The MLFFT Grid

The accuracy of the MLFFT grid depends on two user-specified parameters: the number of approaching levels and the approximating set $\{Cut\}$. In the following, we will describe their meanings and finally reach our description of MLFFT grid.

The number of approaching levels is defined as

$$L_{LAYER} = L, \quad L \geq 1. \quad (15)$$

The approximating set $\{Cut\}$ is defined as

$$\{Cut\} = \{\alpha_i; 0 < \alpha_1 < \alpha_2 < \dots < \alpha_{L-1} < \alpha_L = 1\}. \quad (16)$$

Note that the cardinality of $\{Cut\}$ is equal to L_{LAYER} . These two parameters are manually given in our definition of the MLFFT grid, but we will discuss the methods for deciding and optimizing these values in Section 3.

Now, we define the MLFFT grid on the frequency domain. Given L_{LAYER} and $\{Cut\}$, we define the grid as follows:

$$P_i = \{(\alpha_i u, \alpha_i v), -N/2 \leq u, v \leq N/2\},$$

$$\alpha_i \in \{Cut\}, 1 \leq i \leq L_{LAYER}. \quad (17)$$

Each point in Set P_i corresponds to a certain frequency on the 2D frequency domain plane. Suppose we have an image of size $(N+1) \times (N+1)$ centered at $(0, 0)$, (N is an even number), then, by applying the 2D fractional Fourier transform, we can calculate the value at any frequency in P_i by (14).

It should be emphasized that the computation of different layers is independent, and it is easy to compute each layer in a parallel way on multi-CPU systems.

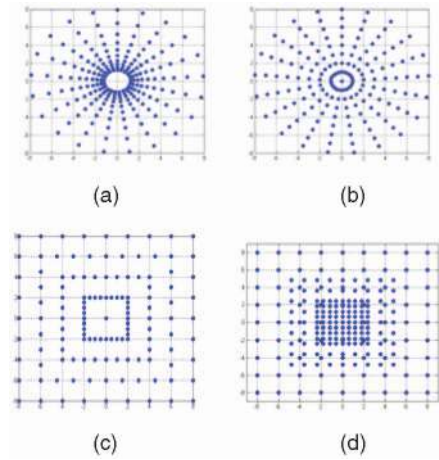


Fig. 2. (a) The log-polar grid. (b) The polar grid. (c) The pseudopolar grid. (d) The MLFFT grid with three layers. The MLFFT grid is much closer to the log-polar grid.

Finally, we get the MLFFT grid P (Fig. 1):

$$P = \bigcup_{i=1}^{L_{LAYER}} P_i. \quad (18)$$

Fig. 1 explains how a three-layer MLFFT grid is constructed through the three individual grids.

Fig. 2 demonstrates why MLFFT is much close to a log-polar grid than other methods qualitatively.

Based on the grid in Fig. 2d, we are able to generate the polar/log-polar Fourier transform at different resampling rates by interpolating on the MLFFT grid. We will see in Section 2.3 that such an interpolation leads to a significant improvement, compared with other methods in both the polar and the log-polar grids.

2.3 The Interpolation Error

This section discusses the interpolation error of the 2D Cartesian FFT, the pseudopolar FFT, and our method, MLFFT. We will explain advantages of our method over other methods in three respects: 1) MLFFT performs better than other methods for the polar Fourier transform, 2) MLFFT performs better than other methods for the log-polar Fourier transform, and 3) MLFFT performs even much better for the log-polar Fourier transform in image registration.

For a certain image's spectrum magnitude M , we define the first-order interpolation error [2] as

$$\varepsilon_{P/LP}^1 \approx \sum_{i,j} |\varepsilon_{Grid}(r_i, \theta_j)| |\nabla M(r_i, \theta_j)|, \quad (19)$$

where $\varepsilon_{Grid}(r_i, \theta_j)$ is the distance between the actual point in frequency plane and the closest point in the grids:

$$\varepsilon_{Grid}^2(r_i, \theta_j) = (\hat{X}_{closest} - X_{real-polar})^2 + (\hat{Y}_{closest} - Y_{real-polar})^2. \quad (20)$$

Because $|\nabla M(r_i, \theta_j)|$ are different for different signals, we substitute $|\nabla M(r_i, \theta_j)|$ with $\max\{|\nabla M(r_i, \theta_j)|\}$, the upper bound of all the $|\nabla M(r_i, \theta_j)|$ of different signals that is a

No	Definition of the Grid
A	N=200 2D FFT
B	N=200 2D Pseudo-Polar FFT
C	N=200 L=2 {Cut}={0.4,1} (Unoptimized)
D	N=200 L=3 {Cut}={0.1,0.6,1} (Unoptimized)
E	N=200 L=4 {Cut}={0.1,0.3,0.7,1} (Unoptimized)
F	N=200 L=3 {Cut}={0.2, 0.6, 1} (Optimized)

Fig. 3. Cases tested for their first interpolation error.

constant for a group of signals to calculate the maximum interpolation error, and we have

$$\begin{aligned} \varepsilon_{P/LP}^1 &\leq \max\{\varepsilon_{P/LP}^1\} \\ &\approx \sum_{i,j} |\varepsilon_{Grid}(r_i, \theta_j)| \times \max\{|\nabla M(r_i, \theta_j)|\}. \end{aligned} \quad (21)$$

By calculating (21) with such a constant, we are able to compare the max interpolation errors of different polar transform methods without considering the signal itself. However, for signals in an image, $|\nabla M(r_i, \theta_j)|$ is much higher in low frequency than in high frequency. We will prove that MLFFT works even better with images that have high low-frequency energy in Fig. 4c.

We will test all of the grids in Fig. 3 on the polar and log-polar transforms. The sampling rates of the polar and the log polar are given as follows: In the polar case, we test with two angular sampling rates: $\Delta\theta = 0.1$ and $\Delta\theta = 0.01$. In the log-polar case, we test the same two angular sampling rates, and we use 1.047 as the log-base, which is calculated according to that in [3]. It should be pointed out that, for MLFFT, both the layer number and $\{Cut\}$ are critical to a low interpolation error and our tests will demonstrate that. In our tests, the $\{Cut\}$ is casually chosen for cases C, D, and E to show the performance of MLFFT with an almost arbitrary $\{Cut\}$, while case F takes an optimized parameter using the method described in Section 3.2. Figs. 4a and 4b are the maximum interpolation errors calculated by (21). Fig. 4c is the first-order interpolation error for the Lena image (256×256) calculated by (19).

The results in Fig. 4 can be understood in four respects:

1. MLFFT achieves a lower interpolation error than the pseudopolar when the layer number is greater than 2 for general polar and log-polar Fourier transforms.
2. MLFFT can approach better interpolation accuracy by increasing the layer number.
3. A lower interpolation error can be achieved by optimizing $\{Cut\}$.
4. MLFFT even outperforms in evaluating the image's log-polar Fourier transform, which results in successful recovery in the extreme cases in the experiments of Section 4.

2.4 Complexity Analysis

For the 1D fractional Fourier transform, the computing cost is three times as the normal 1D FFT. Therefore, for each layer of MLFFT, the complexity is still $O(N \times N \log N)$. Considering the precise computing cost, a three-layer Fourier transform costs about three times as much as the pseudopolar FFT. However, Keller et al. [2] suggests that 3 or 4 iterations are

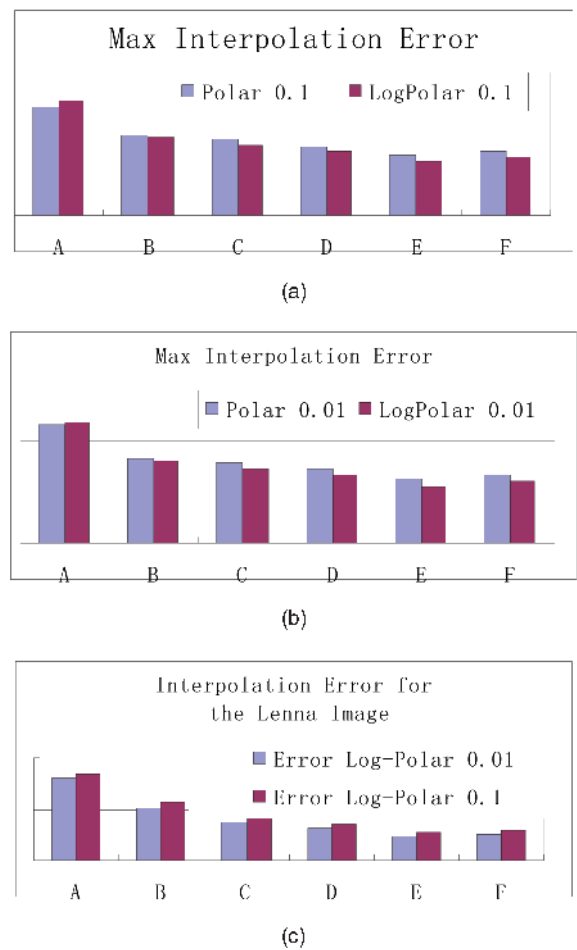


Fig. 4. The first-order max interpolation errors with angular sampling rates (a) 0.1 and (b) 0.01 for polar and log-polar Fourier transforms. (c) The interpolation error for the real-image Lena by formula (19). All of the results are normalized for better presentation.

needed for better accuracy, but, by MLFFT, the same accuracy can be achieved without any iteration process.

2.5 A Framework for Image Registration with MLFFT

Here, we introduce our MLFFT-based algorithm for image registration with large scaling, rotation, and translation. It is the same as the algorithm in [3] except for Steps 3 and 4.

1. Input images I_1 and I_2 apply a high-pass filter and a Blackman window function to them.
2. Choose the proper N_{LAYER} and $\{Cut\}$.
3. Calculate the MLFFT magnitudes of I_1 and I_2 and we get $M_1^{L,\{Cut\}}$ and $M_2^{L,\{Cut\}}$.
4. Calculate the log-polar Fourier transform magnitude spectrums of I_1 and I_2 by interpolation on the MLFFT grid.
5. Calculate the scale factor and the rotation angle by the phase correlation technique on the log-polar Fourier transform spectrums of both images.
6. Apply the scale factor and the rotation angle to the original image and use the phase correlation again to detect the translation.

3 SOME DISCUSSIONS ON ACCURACY

3.1 Choose the Correct Layer Number and $\{Cut\}$

MLFFT is designed to be adaptable through the adjustable parameters. We can adjust the parameters to maximize both the speed and the accuracy performances of MLFFT under different precisions and polar/log-polar resampling rates.

Choosing the proper layer number. The layer number depends on the kind of its application.

A two-layer MLFFT grid has the same number of points as the pseudopolar grid and the difference between these grids are how these points are located. In our experiments in Section 2.4, we have already shown that the two-layer MLFFT performs better than the pseudopolar. However, generally, for more stable and satisfying image registration, we recommend 3-4 layers for most cases and it should exceed the performance of the pseudopolar a lot. To sum up, the right way is to increase the level number if the accuracy level is unsatisfying or the registration fails. On the contrary, if the pseudopolar fails, one can do nothing about it. Thus, we say that MLFFT is adaptive.

Choosing the proper $\{Cut\}$ set. Once the layer number is decided, how to determine the $\{Cut\}$ is critical. Differently from the pseudopolar FFT, one can choose $\{Cut\}$ properly for different polar/log-polar resampling rates on the 2D Cartesian MLFFT grid, while the pseudopolar FFT only works well with certain nonextreme cases [2]. The idea of $\{Cut\}$ comes from the partitioned function imitation technique in numerical analysis.

Choosing $\{Cut\}$ is actually a nonlinear optimizing problem. $\{Cut\}$ can be optimized for the least interpolation error described in (21) in Section 2. Numerical methods like the L-M method [23] can be applied to calculate $\{Cut\}$ as a vector to minimize the error.

However, the numerical methods for nonlinear optimization are relatively slow, so we recommend two approximate models that calculate the interpolation error only for a few resampling points rather than all the points in the log-polar grid. In most cases, we use a set of partial points $\{(\rho, \theta); 1 < \rho < N/2, 0 < \theta < \pi/4\}$ and (21) for calculating the interpolation error with a certain $\{Cut\}$. (Suppose we have an image of size $N \times N$). In addition, it is reasonable to use points on a line $\{(\rho, \theta); 1 < \rho < N/2, \theta = 0\}$ for undemanding cases.

We present here an example of MLFFT on a 200×200 image, where $layer = 2$ and the resampling rate is $log\ base = 1.047$ (calculated according to [3]) and $\Delta\theta = 0.1$. $\{Cut\}$ is evaluated with both partial-interpolation-error L-M method and whole-interpolation-error L-M method. The approximate best $\{Cut\}$ is $\{0.24, 1\}$, while the real best $\{Cut\}$ is $\{0.17, 1\}$. Fig. 5 shows the interpolation errors of points of the whole grid and the interpolation errors of points on part of the whole frequency domain corresponding to different α_1 in a two-layer $\{Cut\} = \{\alpha_1, 1\}$. The L-M will converge at the nadir. In addition, using a line model $\{(\rho, \theta); 1 < \rho < N/2, \theta = 0\}$ for the optimization, the result will still be acceptable. In the previous case, line-interpolation-based L-M optimizing results in $\{Cut\} = \{0.22, 1\}$. Fig. 5 is also helpful in understanding the affect of different $\{Cut\}$ on interpolation accuracy: If the deviation from the

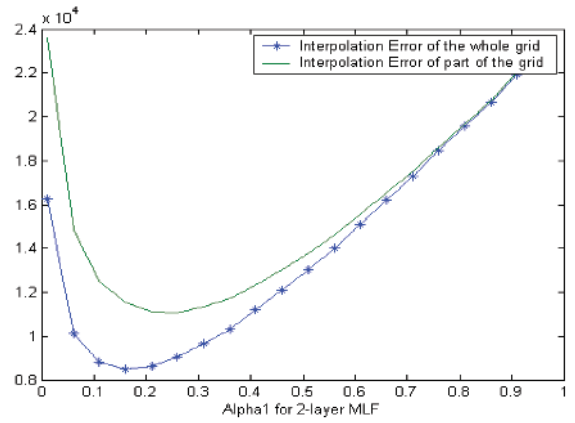


Fig. 5. The first-order interpolation error with different $\{Cut\}$ and different grids. The interpolation error of a partial grid is multiplied by 8 as it only includes 1/8 of the points of the whole grid.

optimized $\{Cut\}$ is not large, the interpolation error does not raise much.

To sum up, we suggest three interpolation models for nonlinear optimization of $\{Cut\}$: the whole grid, the partial grid, and the line grid. If, after optimizing $\{Cut\}$, the interpolation error is still unsatisfying, we can increase the layer number and choose an optimized $\{Cut\}$ again. Once the two parameters are set according to the requirements in an application, we do not need to adjust them during the image registration process.

3.2 SNR and True Peak Value

First, we will introduce the definition of *PeakER* and *SNR* here and show their relations in the phase correlation. Suppose f_2 and f_1 are two images that actually are the log-polar transform magnitude results after Step 6 in Section 2.5 and n is the white Gaussian noise (WGN) function. F_1, F_2 , and N are their spectrums after the Fourier transform:

$$\begin{aligned} f_2(x, y) &= f_1(x - x_0, y - y_0), \\ f_2^{NOISE}(x, y) &= f_2(x, y) + n(x, y). \end{aligned} \quad (22)$$

We define *SNR* as

$$SNR \stackrel{DEF}{=} \frac{\text{Signal Energy}}{\text{Noise Energy}} = \frac{\iint |f_2(x, y)|^2 dx dy}{\iint |n(x, y)|^2 dx dy}. \quad (23)$$

Notice (23) does not involve a log operation which the normal *SNR* definition always does.

SNR is a key factor to the peak energy value while using the phase correlation to determine the rotation angle and the scale factor, as we can see in the following theorem. We also give a simple mathematical proof based on the Parseval Theorem. It should be noted that our concept of Peak Energy Ratio (*PeakER*) is different from the peak value, because *PeakER* indicates the possibility of identifying the true peak from noise besides the peak value itself.

Theorem 2. *In the inverse Fourier transform result of the phase correlation method, the PeakER defined as follows holds:*

$$PeakER \stackrel{DEF}{=} \frac{\text{Peak energy}}{\text{Energy not at the peak}} \approx k \times SNR \quad (24)$$

(k is a constant).

Proof. After applying the Fourier transform to (22), we can calculate the cross-power spectrum as follows:

$$P(u, v) = \frac{F_1(F_2^{NOISE})^*}{\|F_1\| \|F_2^{NOISE}\|} = \frac{F_1 * N^* + F_2 * F_1^*}{\|F_1\| \|F_2^{NOISE}\|} \quad (25)$$

$$= \frac{F_2 * F_1^*}{\|F_1\| \|F_2^{NOISE}\|} + \frac{F_1 * N^*}{\|F_1\| \|F_2^{NOISE}\|}.$$

We know that, after applying the inverse Fourier transform to $P(u, v)$, the first term on the right-hand side of (25),

$$\frac{F_2 * F_1^*}{\|F_1\| \|F_2^{NOISE}\|},$$

should be transformed into a single pulse on the plane and, by the Parseval theorem, the energy of this part will concentrate at the true peak in the inverse Fourier transform result; the last term of (25),

$$\frac{F_1 * N^*}{\|F_1\| \|F_2^{NOISE}\|},$$

will be scattered all over the plane uniformly by the Parseval theorem as it is merely WGN. Suppose $p(x, y)$ is calculated by applying the inverse Fourier transform to $P(u, v)$, then we have

$$PeakER \stackrel{DEF}{=} \frac{\text{Peak energy}}{\text{Energy not at the peak}}$$

$$= \frac{\iint_{\text{peakpoint}} |p(x, y)|^2 dx dy}{\iint_{\text{outsidepeak}} |p(x, y)|^2 dx dy} \approx \iint \left\| \frac{F_2 * F_1^*}{F_1 * N^*} \right\|^2 dudv, \quad (26)$$

$$= \iint \left\| \frac{F_2}{N} \right\|^2 dudv = k \iint \left| \frac{f_2(x, y)}{n(x, y)} \right|^2 dx dy \quad (27)$$

$$= k \frac{\text{Signal Energy}}{\text{Noise Energy}} = kSNR.$$

Thus, Theorem 2 is proven. \square

Theorem 2 is very important in unveiling the relation between SNR and the peak value. A low SNR results in a low $PeakER$, which means more false peaks and a low-value true peak.

Next, we will discuss the source of the noise $n(x, y)$. Besides the well-known rotating aliasing [11], which is an important reason for noise, here we propose another reason that causes a significant noise: the scaling overlapping factor.

For a signal sampled with the highest frequency, $f_{\text{highest}} < f_s/2$, the signal could be retrieved from its inverse DFT with no loss, which is known as the Nyquist sampling principle. The sampling principle is based on the fact that DFT of a discrete signal is periodical and the energy in high frequency may overlap the high-frequency energy in the next period. If one image is zoomed in by a certain scale factor, its

spectrum is also scaled by the same scale factor. Since DFT is periodical, the high-frequency part of the image spectrum in one period will certainly overlap the frequency in the adjacent periods; thus noise is produced due to the overlapping in the frequency domain. The formal expression of this idea is described as follows:

Suppose $F(u, v)$ is the log-polar magnitude spectrum of an image, in Step 5 of the algorithm described in Section 2.5. We apply a scaling operation with a scale factor s to the image and determine what part of $F(u, v)$ is overlapped and what part of $F(u, v)$ is not overlapped in the periodical DFT spectrum after the scaling. Then, we have the following conclusion:

$$SNR_s \stackrel{DEF}{=} \frac{\iint_{\text{NotOverlapped}} |F(u, v)|^2 dudv}{\iint_{\text{Overlapped}} |F(u, v)|^2 dudv} \approx SNR. \quad (28)$$

For an image that is scaled with the scale factor s , during FFT-based registration of the scaled image and the original one, the $PeakER$ holds:

$$PeakER \approx k \times SNR_s. \quad (29)$$

It is hard to present a mathematical analysis for the above conclusion, but here we present a general explanation with experimental results: In the cases without any rotation and translation, it is obvious that the main trouble in the phase correlation method is the overlapping noise due to scaling [24]; thus the overlapping noise is dominant and the first part of the conclusion, i.e., (28), is right. By combining Theorem 2 and (28), we reach the second part of the conclusion, i.e., (29).

Our experiment in Fig. 6a also suggests that this conclusion is correct. In our tests, we first apply different scale factors to the famous Lena image and register the scaled image with the original one. Then, both $PeakER$ and SNR_s of the registrations with different scale factors are calculated, normalized, and showed in Fig. 6a.

The experimental results in Fig. 6a indicate that the relation between the true $PeakER$ and the SNR_s matches the above conclusion and prove again that SNR_s is the dominant factor for the peak energy in the cases without rotation. We also give vivid examples of the peak values of images with different scale factors in Figs. 6b and 6c.

Such overlapping could be avoided by applying low-pass filters, but, in very large scales, the band limit for such filters is hard to decide and such filters will eliminate useful information.

However, for the case of rotated and scaled images, the overlapping noises and the rotating aliasing noises are the dominant sources of noises and the true phase correlation peak is merged in the noise peaks by Theorem 2 in the extreme cases of large scaling and rotating factors.

Though there are some approaches in reducing noises, they are not perfect. In [24], the scholars tried to reduce noises by applying a projecting operator in the phase correlation process under the assumption that the noise energy is a magnitude lower than the total energy. Such a statement is not true in most cases, where there exists larger noise energy. In [11], [3], the researchers suggested a filter to reduce noises that also eliminate the signal energy.

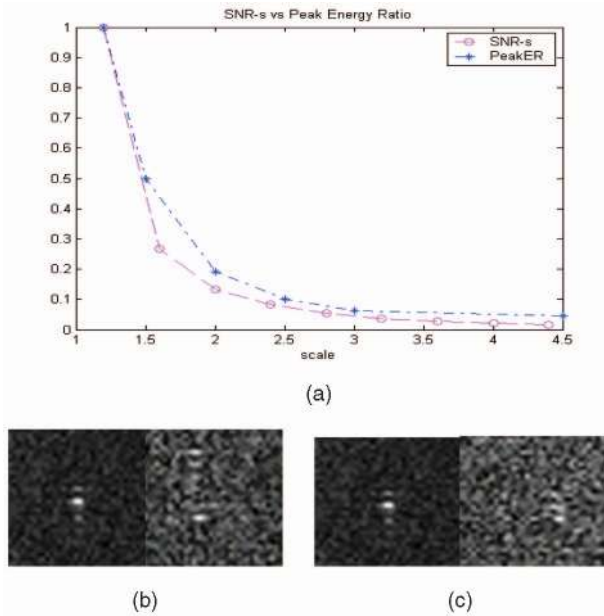


Fig. 6. (a) *SNR*, *PeakERs* at different scale factors of Lena image. (b) Phase correlation results of Step 5 of the algorithm described in Section 2.5, the left one is an image with scale factor 1.20, the right one is an image with scale factor 4.50. We can see that a larger scale factor results in more noise dots around the peak. (c) The phase-correlation results of a 1.2x Lena image using the five-layer MLFFT (left) and the pseudopolar (right), the interpolation error is largely reduced by MLFFT, as we can see that there are fewer noise dots around the peak in the left image than in the right image.

As a conclusion, MLFFT largely removes noises caused by the interpolation errors, but it is still hard to reduce the scaling overlapping, rotation aliasing, and other minor factors that make FFT-based image registration methods in [2], [11], [14], [3] unsuccessful.

3.3 Why Pseudo-Log-Polar [8] Does Not Work

Other scholars presented the pseudo-log-polar method [8] for image registration, which, from our point of view, cannot achieve a better result compared with the pseudopolar method. Here, we will not include their techniques completely but only to point out its internal problems. The idea is to separate the plane to BH and BV (BH and BV are the two subsets of points defined in [8]). Take BV as an example, the process of the pseudo-log-polar applies the 1D FFT to each column of the image and then the 1D FFT to each row of the result obtained by the 1D FFT. However, the pseudo-log-polar chooses nonintegral elevation direction frequencies like $F(0.2)$ and $F(1.7)$ rather than integral elevation direction frequencies like $F(0)$ and $F(1)$ after the 1D FFT step. Notice that it is impossible to get the exact value of arbitrary nonintegral frequencies by the ordinary 1D FFT, so such an algorithm in [8] requires another inaccurate interpolation operation on the ordinary FFT grid which results in an underlying error. However, for the pseudopolar and MLFFT, each point's value on both the pseudopolar grid and the MLFFT grid is from the exact Fourier-transform results of the corresponding frequencies rather than the interpolated results.

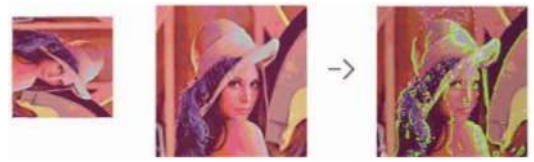


Fig. 7. The image registration. Determining the transform parameter of the left two images. The third image is produced by combining the left two images after applying the inverse transform with the parameter we just recovered and a Sobel edge detecting process to one of them.

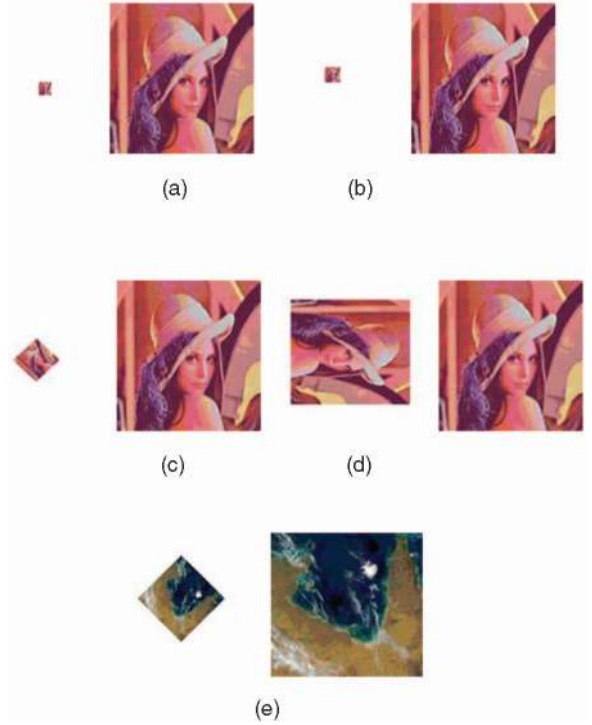


Fig. 8. The five benchmark groups for different image registration methods. The results are listed in Figs. 9 and 11.

4 EXPERIMENTS AND RESULTS

In this section, MLFFT is tested along with the pseudopolar and the traditional 2D Cartesian FFT in image registration, and we will use these algorithms to recover the scale factors, rotation angles, and translation parameters of two images (demonstrated in Fig. 7). All of the test pictures are preprocessed by a high-pass filter and a Blackman window according to [3] and [11]. The tested picture groups are shown in Figs. 8 and 10. We will recover the scale factor, rotating angle, and translation for the left and right images in each group (Fig. 8). We use a bilinear interpolation for all the experiments, but we can use the nearest neighbor interpolation instead. All our experiments are accomplished on a 32-bit AMD Windows XP platform with the Intel Image Process Library 2.5. All images in Figs. 7, 8, 10, 14, 15, and 16 are at the resolution of 256×256 .

We first present in Fig. 9 the results of the experiments of Fig. 8. The real scaling and rotation factors are listed in the second column in Fig. 9. The registration results for different algorithms are listed in the other following columns.

Take a close look at the numbers in Fig. 9. For the images of groups A, B, and C, MLFFT succeeds in retrieving the

GROUP	(SCALE, ROTATION)	FFT*	PPFFT*	MLFFT L=2	MLFFT L=3	MLFFT L=4	MLFFT L=5
CORRECT		COMPUTED					
A	(10.2,0)	FAIL	FAIL	(10.15,-0.06)	(10.15,-0.06)	(10.15,-0.06)	(10.15,-0.06)
B	(6.0,0.0)	FAIL	FAIL	(5.79,0.03)	(5.93,0.03)	(5.93,0.03)	(6.01,-0.03)
C	(5.0,-0.79)	FAIL	FAIL	(5.00,-0.81)	(5.00,-0.81)	(5.00,-0.81)	(5.00,-0.81)
D	(1.5,1.57)	(1.38,1.67)	(1.51,1.52)	(1.49,1.61)	(1.49,1.59)	(1.49,1.56)	(1.50,1.57)
E	(1.75,0.79)	FAIL	(1.97,0.8)	FAIL	(1.75,0.78)	(1.75,0.78)	(1.75,0.79)

Fig. 9. The registration results of benchmark pictures in Fig. 8.

parameters of rotations and scaling, while other methods failed. Our experiments show that, actually, MLFFT can recover scale factors up to 10 without rotation or up to 5 with arbitrary rotation, while the current best result [2] is only up to 4. Recall that it has been covered in Section 3.2 why it is hard to recover a larger scale factor.

In groups D and E, MLFFT reaches more accurate results without any iteration, unlike [2], and its rotation error is below 1° (as accurate as [14]). In addition, the experiment indicates that the normal three-layer or four-layer MLFFT performs well enough in image registration.

We also have done experiments of MLFFT using other images in Fig. 10 to test MLFFT's accuracy and we randomly rotate them to different angles and enlarge them to different scales in acceptable range that these algorithms do not fail. The overall accuracy statistics for all of the benchmark images in Figs. 8 and 10 are available in Fig. 11. Note that only the images registered successfully are counted in Fig. 11. By analyzing the data in Fig. 11, we realize that the angular accuracy of MLFFT is as good as [14] and the scaling accuracy is enough satisfying. However, unlike MLFFT, [14] works only for rotation estimation, it cannot be applied to the estimation of both translations and scaling in images. The experiments fully demonstrate the advantages of MLFFT over all other current Fourier-based methods, including pseudopolar Fourier transforms.

We also used the research image data set from the University of Southern California (USC) [26] for the

benchmark and converted all USC images to a size of 256×256 . This data set has around 300 pictures from textures to aerial pictures and its wide coverage made us believe that it is a good test suitable for our purpose. We use all of these images by synthesizing many rotated and scaled images from the original images and registering them. Thus, we could build up thousands of test cases. Rather than running through the whole data set with different algorithms for image registration, we choose to separate our benchmark into two experiments.

In the first experiment, we test only two extreme cases (5x zoom with rotation and 10x zoom without rotation) to see the performance and the success rates are listed in Fig. 12. We also tested the SIFT method by using the matched key points to calculate the transformation matrices. Notice in our tests that we did not use any iteration, while, in [2], the researchers applied several times of the pseudopolar-based registrations iteratively on two images to reach the correct results.

From Fig. 12, we can tell that the MLFFT is much better than the pseudopolar method in the extreme cases, which is enough to prove that the MLFFT does eliminate much more interpolation noises in registration tasks than the pseudopolar method. In addition, in cases with large scale factors, MLFFT stands out from all three algorithms.

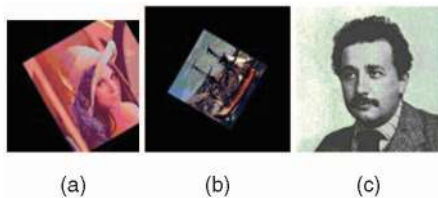


Fig. 10. Other benchmark images.

	MLFFT	PSEUDO POLAR
Scaling Factor	0.80%	0.75%
Rotation Angle	0.85°	2.01°

Fig. 11. Mean error statistics for all benchmark images in Figs. 8 and 10.

	Zoom 5x, rotation: $\theta = 0.1$	Zoom 10x, no rotation
MLFFT(6 layers)	25%	27%
Pseudo-Polar	0%	0%
SIFT	31%	0%

Fig. 12. Success rate of the USC image registration benchmark tests under two circumstances.



(a) (b)



(c)

Fig. 13. We perform massive tests on the three categories of the USC image library. (a) Repeated textures. (b) and (c) Scenes.

Test Cases	
1	$s = 1.5, \theta = 0.10$, small translations
2	$s = 1.5, \theta = 0.10$, large translations
3	$s = 3, \theta = 0.10$, small translations
4	$s = 2, \theta = 0.75$, small translations

(a)

	Repeated textures	Scenes	Overall Average
MLFFT(6 layers)	63%	82%	72%
SIFT	49%	77%	63%

(b)

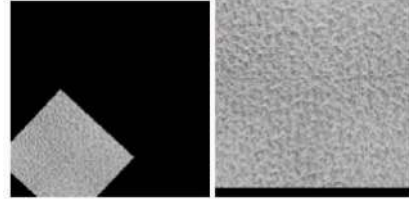
Fig. 14. Success rate on categorized images with MLFFT and SIFT method over the USC image database. (a) All test cases. (b) Success rates over the USC image database.

Another test we conducted is that we manually categorize the whole USC image library to two categories, repeated textures and scenes, since the database is mainly consisted of these two kinds of images, as shown in Fig. 13.

We run SIFT and MLFFT on randomly selected rotation angles and scaling values, as described in Fig. 14a, with moderate translations and calculate the overall success rate



(a)



(b)

Fig. 15. (a) Gaussian noises are added to the right image and SIFT fails in such registration tasks. (b) The repeated texture is also a bad case for the SIFT method.



(a)



(b)



(c)



(d)

Fig. 16. Two images (a) and (b) taken by a handheld camera are combined into one picture, shown in (c) and (d), using MLFFT and the pseudopolar FFT, respectively.

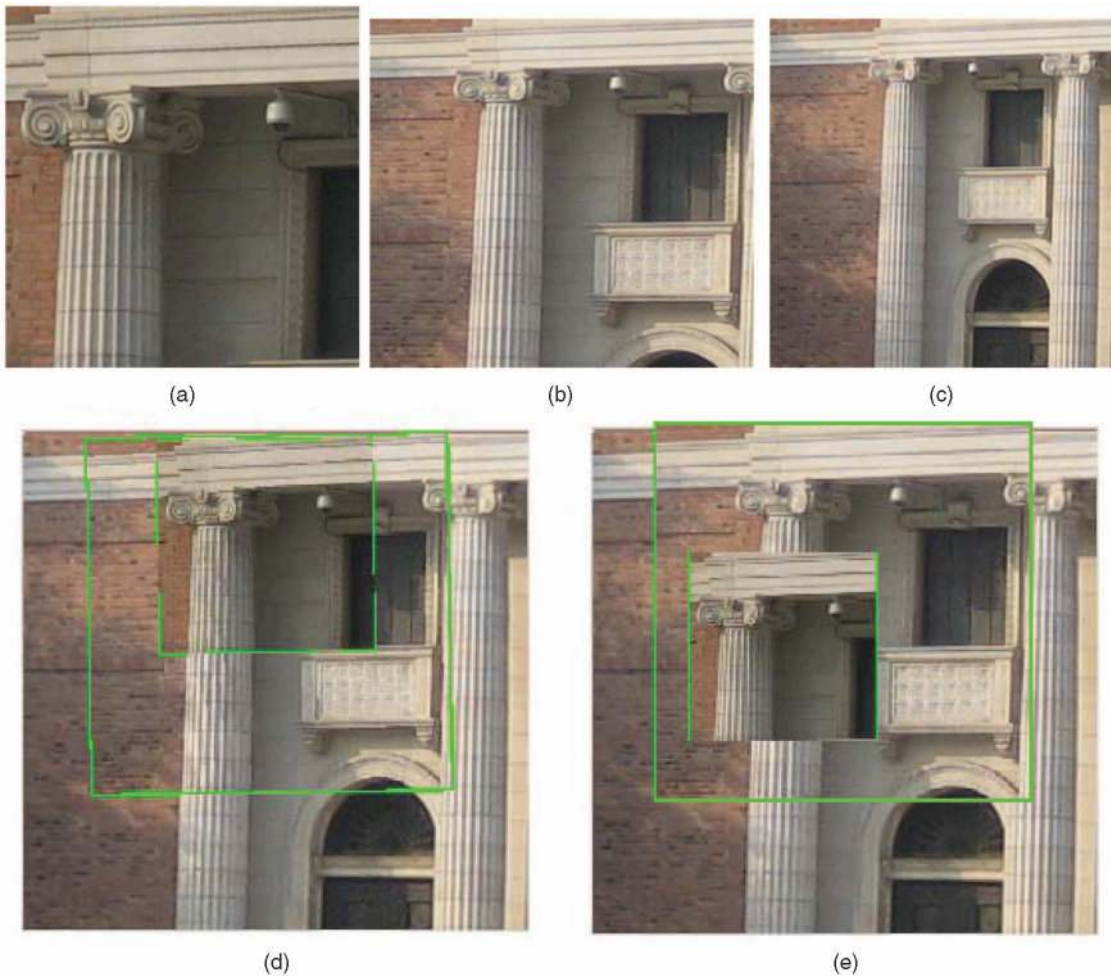


Fig. 17. Using MLFFT to recover transforms among the above pictures (a), (b), and (c), which were hand-taken with **various focus lengths**. (d) Registration Result of MLFFT. (e) Registration Result of the pseudo-polar FFT.

for MLFFT and SIFT, as described in Fig. 14b. Small translations mean over 1/5 of the images are translated out of the picture, while large translations mean more than half of the images are translated out of the picture.

This experiment suggests the nearly equal performance of MLFFT and SIFT in the typical cases. For the cases with simple textures and cases with large translations, MLFFT and SIFT's overall success rates are not perfect. However, in most typical cases with moderate translations and usual images, both MLFFT and SIFT could reach a success rate over 90 percent.

To conclude, comparing Fourier-based methods with the state-of-the-art SIFT-based spatial image registration techniques, we realized that SIFT, as a representative of current spatial registration techniques, is generally the best option for image registration: It supports any transformation models besides perspective models; it has a high success rate in our tests. However, we see three advantages for the MLFFT method over SIFT: 1) For large scaled image, SIFT does not work at all. 2) For noised image and texture-like image, e.g., in Figs. 15a and 15b, the SIFT method fails, while the MLFFT succeeds in retrieving the transforming parameters. 3) The computational complexity for spatial methods is high, while FFT-based methods have a fixed and lower computational complexity, especially for high-resolution pictures.

As a conclusion as well as a guideline for readers, we learn from our attempts and experiments that MLFFT is accurate on images with rich high-frequency component, like the aerial images of the USC image database, in which values vary significantly from pixel to pixel. MLFFT works less satisfyingly on smooth images like human face in Fig. 13b, which fails in our tests. Both MLFFT and SIFT methods suffer from huge translations. However, MLFFT is more suitable for the cases with significant scaling, noise interference, or texturelike images.

Here we present a couple of image mosaicking examples in Figs. 16, 17, 18, and 20, produced by the framework presented in this paper. In Fig. 20, we present a panorama, which is a combination of 13 pictures by using MLFFT. Three sets of pictures taken casually with a hand-held camera in Figs. 16, 17, and 18 are combined to three larger images by using the results recovered by the PFFT [2] and the MLFFT algorithm in Section 2.5, respectively. The parameter values of the translating, rotating, and scaling transforms recovered in Figs. 16, 17, and 18 are shown in Fig. 19. If not using MLFFT, we cannot combine them with such accuracy. The current state-of-the-art pseudopolar FFT does not work so accurately for such image mosaics and even fails in some cases. In Fig. 16, we demonstrate the rotation accuracy of MLFFT. In contrast, the pseudopolar FFT failed to recover the correct transform parameters.

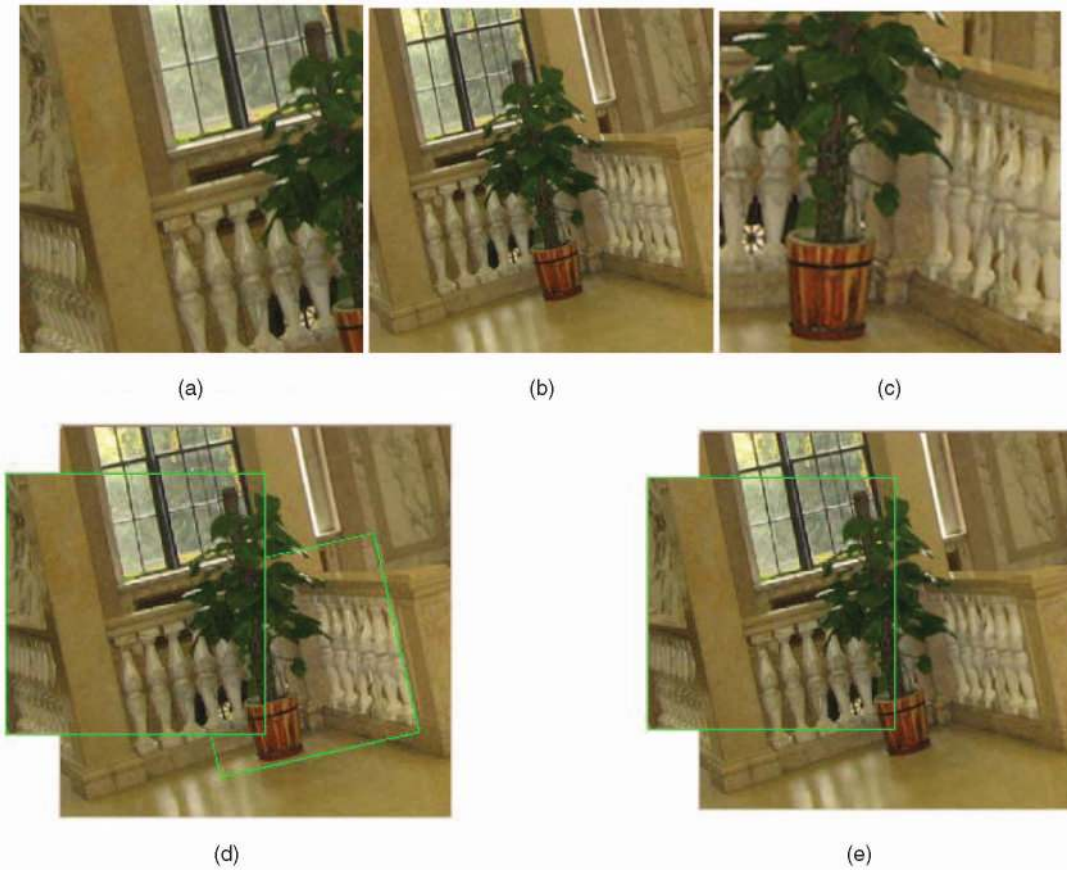


Fig. 18. Using MLFFT to register the above three pictures (a), (b), and (c), which are **taken casually**. (d) Result registered by MLFFT. (e) Result registered by the pseudo-polar FFT: The recovered parameters of (c) result in a huge deviation (with an unreasonable scale factor, see Fig. 15), so (c) is not visible in the result.

IMAGE PAIRS	4-LAYER MLFFT ($\theta, s, \Delta x, \Delta y$)	PPFFT ($\theta, s, \Delta x, \Delta y$)
Figure 12 (a), (b) *	(0.360, 1.02, 97, 17)	(0.000, 2.00, -23, 20)
Figure 13 (b), (a)*	(0.000, 1.67, 0, -49)	(0.000, 1.50, -32, -2)
Figure 13 (c), (b)	(0.0150, 1.39, -3, -34)	(0.000, 1.33, -1, -35)
Figure 14 (b), (a)	(0.000, 1.5, -78, -11)	(0.000, 1.5, -78, -11)
Figure 14 (b), (c) *	(0.225, 1.76, 27, 22)	(0.005, 28.12, N/A, N/A)

Fig. 19. The detailed registration results of the images in Fig. 12 with PPFFT and our algorithm, MLFFT. PPFFT failed in the row with a star *. For the case of that in Figs. 14b and 14c, the deviation of the scale factor by PPFFT is so large that there is no need to do phase correlation to detect translations. The central point is used as the axis point in rerotating and rescaling the image before the translation estimation.

Notice that there are two-to-three-pixel errors in the registration results in Fig. 17 and this is mainly because the camera-taken pictures have a more complex transform model, which the MLFFT cannot handle, rather than simple rotating, scaling, and translating. This example presents the ability of MLFFT in recovering scale factors. However, the pseudopolar FFT performs less accurately: Mismatches are remarkable in the top-left corner in Fig. 17e. Although there are errors at the top-right corner in Fig. 18d, Fig. 18 illustrates that the MLFFT can handle situations in which rotation, scaling, and translation are all involved and the pseudopolar FFT failed to get a satisfying result (see Figs. 18e and 19).

In the experiments in Figs. 8, 9, 10, and 11, we have successfully repeated the experiments in [2], [3] and recovered scale factors over 4 with the pseudopolar FFT, again proving the results in [2]. We further demonstrate the power of MLFFT by recovering even larger scale factors. As we stated in the introduction, the pseudopolar FFT could

not handle cases involving real images well, but, in our experiments, we fully demonstrate that our MLFFT method is able to register images taken casually by a handheld camera. Our target of minimizing the interpolation error is achieved and we believe that MLFFT is a strong candidate among Fourier-based registration methods and could be used as a stand-alone tool and a bootstrap for image-mosaicking applications.

For a normal 256×256 image, it takes 1.5 seconds to go through the entire algorithm with a three-layer MLFFT on a 2 GHz PC. The short interval persuades us that our algorithm may be suitable for interactive applications with a dual-core CPU platform for the MLFFT algorithm can run concurrently for each independent layer.

5 CONCLUSIONS

In this paper, a novel method MLFFT is presented for the fast and accurate polar/log-polar Fourier transforms. Its



Fig. 20. The image registration results by combining 13 images with MLFFT. You can see that the 13 images are well registered. It cannot be done if we only take the pseudopolar in the image registration.

advantages over other fast polar and log-polar Fourier transform methods and its efficiency are proven theoretically and experimentally. In addition, the presented method is applied to image registration and succeeds in recovering scale factors up to 10 and rotation angles with high accuracy than results in [2], [14], which demonstrates MLFFT's accuracy again. Unlike other pseudopolar-based registration schemes [2], our method does not require any interpolation and iteration. Thus, it is more accurate and significantly faster. We believe that the MLFFT method can also be applied to other research that involves fast polar/log-polar 2D Fourier transforms not only in image registration. In addition, as the multicore CPU architecture is becoming more popular today, parallel computing each layer in a multicore CPU system may make the MLFFT more applicable and useful.

In the future, we will extend MLFFT to other signal processing applications. We will also continue to improve FFT-based image registration techniques and extend them to more complicated situations (e.g., efficiently registering videos into panoramic mosaics, 3D object recognition, etc.) besides image mosaics.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their perceptive comments. This work is supported by the National High Technology Research and Development Program ("863" Program) of China (2006AA01Z326), the National Natural Science Foundation of China (60773144 and 60610360), and the Beijing Natural Sciences Foundation (4072015). Dr. Kaihuai Qin is the corresponding author for this paper.

REFERENCES

- [1] B. Zitova and J. Flusser, "Image Registration Methods: A Survey," *Image Vision Computing*, vol. 21, no. 11, pp. 977-1000, Nov. 2003.
- [2] Y. Keller, A. Averbuch, and M. Israeli, "Pseudopolar-Based Estimation of Large Translations, Rotations and Scalings in Images," *IEEE Trans. Image Processing*, vol. 14, no. 1, pp. 12-22, Jan. 2005.
- [3] B.S. Reddy and B.N. Chatterji, "An FFT-Based Technique for Translation, Rotation and Scale-Invariant Image Registration," *IEEE Trans. Image Processing*, vol. 5, no. 8, pp. 1266-1271, Aug. 1996.
- [4] A. Averbuch, R.R. Coifman, D.L. Donoho, M. Elad, and M. Israeli, "Fast and Accurate Polar Fourier Transform," *Applied and Computational Harmonic Analysis*, vol. 21, pp. 145-167, 2006.
- [5] S. Man and R.W. Picard, "Virtual Bellows, Constructing High Quality Still from Video," *Proc. IEEE Int'l Conf. Image Processing*, Nov. 1994.
- [6] S. Zokai and G. Wolberg, "Image Registration Using Log-Polar Mappings for Recovery of Large-Scale Similarity and Projective Transformations," *IEEE Trans. Image Processing*, vol. 14, no. 10, Oct. 2005.
- [7] H. Foroosh, J.B. Zerubia, and M. Berthod, "Extension of Phase Correlation to Subpixel Registration," *IEEE Trans. Image Processing*, vol. 11, no. 3, Mar. 2002.
- [8] H. Liu, B. Guo, and Z. Feng, "Pseudo-Log-Polar Fourier Transform for Image Registration," *IEEE Signal Processing Letters*, vol. 13, no. 1, Jan. 2006.
- [9] C.D. Kuglin and D.C. Hines, "The Phase Correlation Image Alignment Method," *Proc. IEEE. Conf. Cybernetics Soc.*, pp. 163-165, Sept. 1975.
- [10] H.Y. Shurn and R. Szeliski, "Construction and Refinement of Panoramic Mosaics with Global and Local Alignment," *Proc. IEEE. Conf. Computer Vision*, 1998.
- [11] H.S. Stone, B. Tao, and M. McGuire, "Analysis of Image Registration Noise Due to Rotationally Dependent Aliasing," <http://tux.cs.brown.edu/people/morgan>, 2008.
- [12] P.N. Swartztrauber and D.H. Bailey, "The Fractional Fourier Transform and Applications," *SIAM Rev.*, vol. 33, no. 3, pp. 389-404, Sept. 1991.
- [13] A.V. Oppenheim and R.W. Schaffer, *Discrete Time Signal Processing*, second ed. Prentice Hall, 1999.
- [14] Y. Keller, Y. Shkolnisky, and A. Averbuch, "The Angular Difference Function and Its Application to Image Registration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, June 2005.
- [15] M. Frigo and S.G. Johnson, "FFTW: The Fast Fourier Transform in the West," MIT, www.fftw.org, 2008.
- [16] 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.
- [17] A. Averbuch, R.R. Coifman, D.L. Donoho, M. Elad, and M. Israeli, *Fast and Accurate Polar Fourier Transform*, www.cs.technion.ac.il/~elad/publications/journals/2004/30_PolarFFT_ACHA.pdf.
- [18] L.R. Rabiner, R.W. Schaffer, and C.M. Rader, "The Chirp-Z Transform Algorithm," *IEEE Trans. Audio Electroacoustics*, vol. 17, pp. 86-92, June 1969.
- [19] Y. Keller, A. Averbuch, and Y. Shkolnisky, "Algebraically Accurate Volume Registration Using Euler's Theorem and the 3D Pseudo-Polar FFT," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 795-800, 2005.
- [20] 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.
- [21] 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.
- [22] D. Steedly, C. Pal, and R. Szeliski, "Efficiently Registering Video into Panoramic Mosaics," *Proc. 10th IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 1300-1307, Oct. 2005.
- [23] J. More, "Levenberg-Marquardt Algorithm: Implementation and Theory," *Proc. Conf. Numerical Analysis*, 1967.

- [24] Y. Keller, A. Avenbuch, and O. Miller, "Robust Phase Correlation," *Proc. IEEE 14th Int'l Conf. Pattern Recognition*, vol. 2, pp. 740-743, Aug. 2004.
- [25] G. Wolberg and S. Zokai, "Robust Image Registration Using Log-Polar Transform," *Proc. IEEE Int'l Conf. Image Processing*, pp. 493-496, Sept. 2000.
- [26] "The USC-SIPI Image Database," Univ. of Southern California, <http://sipi.usc.edu/database/>, 2008.
- [27] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.
- [28] M. Brown and D.G. Lowe, "Recognising Panoramas," *Proc. Ninth IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 1218-1225, Oct. 2003.



Wei Pan received the BEng degree in computer science from Tsinghua University, Beijing, in 2007. He is currently a first year PhD student in the Computer Science Department at Dartmouth College, New Hampshire, where he is also a research assistant. He was a research assistant in the Media Lab and the Graph Lab in the Department of Computer Science and Technology at Tsinghua University. He also worked for Google China for more than one year. His

research interests include system engineering, machine learning, vision, and, recently, computer music. He received several college merit and science scholarships, as well as different nationwide and citywide honor titles before he came to the US.



Kaihui Qin received the BEng degree from South China University of Technology in 1982 and the MEng and the PhD degrees from Huazhong University of Science and Technology in 1984 and 1990. He is a professor of computer science and technology at Tsinghua University, Beijing, and also an adjunct professor of computer science at Shenyang Institute of Aeronautical Engineering, Shenyang, Liaoning Province, China. He was a teaching assistant

and a lecturer at the Huazhong University of Science and Technology in 1984-1986 and 1987, respectively. He was a postdoctoral research fellow from 1990 to 1992, and then joined the Department of Computer Science and Technology at Tsinghua University as an associate professor in 1992. He was a visiting scholar at the Surgical Planning Lab (SPL), BWH, Harvard Medical School, Harvard University, Boston, from 1999 to 2000. Since 1998, he has visited the Department of Computer Science and Engineering at the Chinese University of Hong Kong many times, doing joint research work for a few months every time. He visited the Department of MEEM at City University of Hong Kong as a senior research fellow in July-September 2008. His research interests include computer graphics and animation, tiled projection display and real-time rendering, computer-aided geometric design, physics-based geometric modeling, wavelets, image processing and visualization, virtual reality, and CAD/CAM.



Yao Chen received the BEng degree in computer science from Tsinghua University, Beijing, in 2007. She is currently a graduate student in computer science at Tsinghua University. Her main research interests include image processing and computer vision.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.