

Generalized Pseudo-Polar Fourier Grids and Applications in Registering Ophthalmic Optical Coherence Tomography Images

Nigel Chou*, Joseph A. Izatt[†], Sina Farsiu[‡]

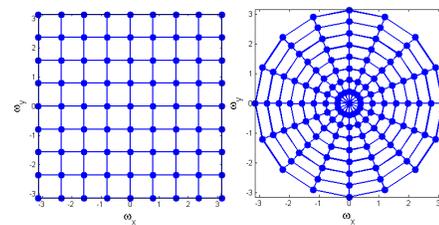
* Dept. of Biomedical Engineering, Duke University, Durham, NC, Email: nsc3@duke.edu

[†] Depts. of Biomedical Engineering and Ophthalmology, Duke University, Durham, NC, Email: joseph.izatt@duke.edu

[‡] Depts. of Ophthalmology and Biomedical Engineering, Duke University, Durham, NC, Email: sina.farsiu@duke.edu

Abstract—Fast and accurate estimation of the Fourier transform in polar coordinates has long been a major challenge for many image processing applications including phase-correlation based motion estimation. To address this problem, it has been proposed to calculate the Fourier transform coefficients on the pseudo-polar coordinates. To acquire better image registration accuracy, we introduce the generalized pseudo-polar motion estimation framework that encompasses the classic pseudo-polar and related techniques. Experimental results on controlled data as well as real world spectral domain ophthalmic optical coherence tomography images of adults and neonatal patients attest to the effectiveness of the presented method.

for many real world applications such an interpolation process is not accurate enough and results in significant errors.



(a) Cartesian

(b) Polar

Fig. 1: Comparison of the Cartesian and Polar grids.

I. INTRODUCTION

Motion estimation is of great importance to many commercial and scientific image processing and computer vision applications. Numerous image registration techniques have been developed throughout the years [1]. Of these, phase correlation based methods [2], [3], which solve the image registration problem in the Fourier domain, are among the most popular. For the case of translational motion, these techniques can directly take advantage of the fast Fourier transform (FFT), resulting in fast and accurate estimation of large displacements. As we will briefly review in Sect. II, estimating the more general motion models that include rotation and scale requires computation of Fourier coefficients on the polar grid (Fig. 1). Since FFT is commonly defined on the Cartesian grid, one might choose to use the discrete Fourier transform (DFT) to calculate the polar coordinates coefficients. This approach has a computational complexity in the order of $O(n^4)$ for a $n \times n$ image. Alternatively, to reduce the computational complexity to $O(n^2 \log(n))$, one might calculate the the FFT coefficients on the Cartesian grid and then interpolate them to the polar grid (what we call the *Cartesian* technique). Unfortunately, as we will see in Sect. IV,

In fact, aside from registration, a wide range of practical imaging problems rely on fast and accurate estimation of the Fourier transform in polar coordinates (see [4] for applications in tomography). Based on the early works on digital tomography reconstruction [5], a technique called pseudo-polar (PP) FFT has recently been developed [6], in which evaluation of coefficients are done in an oversampled set of nonangularly equispaced points (PP-grid). The PP-FFT framework was adapted in [7] for the motion estimation application, which demonstrated more accurate estimation of registration parameters compared to the Cartesian method. Many subsequent publications have adapted the PP-grid for a variety of applications ([7], [8], [9], [10]). Very recently, the mathematical framework for estimating FFT coefficients on an Octa-Polar (OP) grid which resembles the polar coordinates more closely than the PP-grid was established in the Masters thesis of Ofir Harari [4]. While possible applications of this technique in X-ray tomography reconstruction have been discussed, in [4] no experimental results on validity and accuracy of this technique has been offered. Also, surprisingly, to the best of our knowledge, no peer reviewed publication has discussed the OP-FFT work of [4].

Dr. Farsiu's work was supported in part by the Hartwell Foundation and the Knights Templar Eye Foundation, Inc. Pediatric Ophthalmology Research Grant. Dr. Izatt and N. Chou's work was supported in part by the National Institutes of Health grant R21EY017393. Our MATLAB software implementation of the proposed algorithms is in part based on the Polar-FFT software, developed by Prof. Michael Elad of the Technion - Israel Institute of Technology. We would like to thank Prof. Yosi Keller of Bar Ilan University for invaluable comments and Prof. Cynthia A. Toth of Duke University for providing the neonatal ophthalmic SDOCT data.

The contribution of our paper is development of a general framework for estimating the Fourier transform in polar coordinates using FFT, which encompasses the PP-FFT and OP-FFT techniques. This paper is organized as follows: In Sect. II, we briefly review the Fractional Fourier Transform (FRFT). Sect. III describes the generalized PP-FFT (GPP-

FFT) framework. Sect. IV presents the experimental results on registering spectral domain optical coherence tomography (SDOCT) retinal images. Concluding remarks on implementation and computation issues are given in Sect. V.

II. REVIEW: THE FRFT AND APPLICATION TO 2D TRANSFORMS

In this section, we review a cardinal technique that will be used in the next sections to develop the GPP-FFT based motion estimation technique. The FRFT, a generalization of the Fourier Transform [11], is defined as

$$\hat{f}_\alpha(\omega) = \sum_{i=0}^{n-1} f(x) e^{-i \cdot 2\pi x \omega \cdot \alpha}. \quad (1)$$

This definition is identical to the DFT except for the factor α ; it reduces to the DFT, when $\alpha = \frac{1}{n}$. Computational complexity of FRFT is approximately four times higher than that of the FFT, of $O(n \log n)$ [12].

While the DFT is limited to computing points in the frequency domain over the full unit circle, the FRFT evaluates a set of equispaced points on the frequency domain over a fraction of this range, determined by α , as shown in Fig. 2a-b. The FRFT can be centered about $\omega = 0$ by pre-multiplying the input vector by a phase ramp, which we denote as C :

$$C = e^{i2\pi\alpha\left(\frac{n}{2}-1\right) \cdot x}, \quad (2)$$

where n is the length of the vector (Fig. 2c).

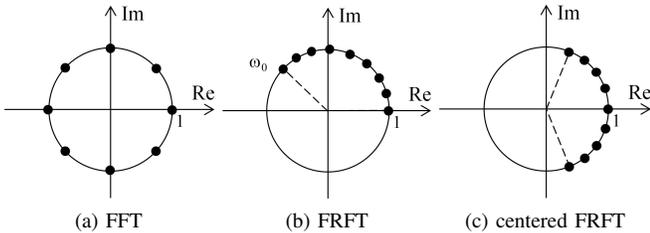


Fig. 2: (a) Samples of an 8-point FFT on the unit circle. (b) Samples of the FRFT computed with (1), with $\alpha = \frac{\omega_0}{2\pi n}$ (c) The same FRFT centered around $\omega = 0$.

The FRFT has a particularly interesting property when applied to 2D images. By linearly varying α along one dimension (ω_x or ω_y) while keeping it constant in the other, we may ‘carve out’ shapes, such as triangles, in the frequency space. We will exploit this property in Sect. III-A.

III. GENERALIZED PSEUDO-POLAR FRAMEWORK

A. Concept behind the Algorithm

As noted in the previous publications [13], [7], rotation estimation involves computing the Fourier transform of an image in polar coordinates (henceforth known as the polar Fourier transform). Traditionally, polar Fourier is either computed by exploiting the accurate but computationally heavy DFT, or by the faster but less accurate method of interpolating Cartesian

2D FFT coefficients (Fig. 1a) to the polar grid (Fig. 1b). A quick consultation with Fig. 1 shows that the points at the center of the polar grid are more densely packed than those at the edges, while in the Cartesian grid the points are evenly spaced throughout. Therefore, achieving comparable accuracy requires oversampling of the Cartesian grid to densities similar to that at the center of the polar transform.

In order to reduce interpolation error while keeping the computational cost low, we search for grids that can be computed using fast algorithms (such as FRFTs or FFTs), yet better approximate the polar grid than a Cartesian grid. Based on the early works on digital tomography reconstruction [5], such grids were first proposed in the form of PP-grid [6]. Recently, the mathematical foundation for an alternative grid, called the OP-grid, has been developed that more closely approximates the polar-grid [4]. We propose that OP and PP are part of a larger conceptual framework of increasingly polar grids (see Fig. 3), called GPP-grid, of which PP-grid and OP-grid are the first and second order approximations, respectively. Indeed, as higher-order grids become increasingly accurate and better resolved, the computational complexity will also be increased.

The main idea behind the GPP transform is to approximate

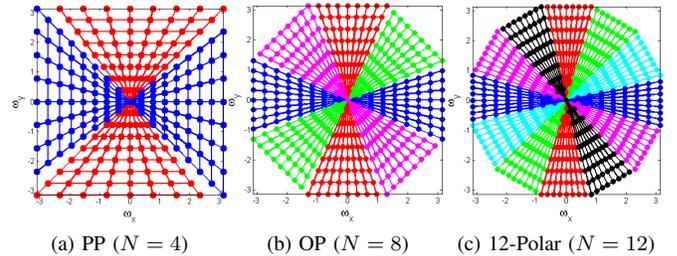


Fig. 3: A series of increasingly polar grids (left to right). The PP-grid (a) is the combination of two pairs of wedges (colored as red and blue), OP-grid (b) consists of four pairs of wedges, and 12-polar grid (c) has 6 pairs of wedges.

the polar grid with a N -sided regular polygon (consisting of $\frac{N}{2}$ wedge-pairs). As the number of sides (N) approaches ∞ , the polygon approaches a circle. The polygon is created by combining pairs of triangles (wedges) at various angles (Fig. 3). The more wedges used, the more sides the polygon will have, hence the closer to polar the transform will be. Notice that a wedge-pair is nothing but two connected triangles, with the same number of points in each row, which are spaced increasingly closer toward the center of the transform. Fortunately, estimating the triangular shaped regions of the frequency domain can be efficiently performed by exploiting FRFTs (Sect. II).

The fractional constant α in (1) determines the closeness of the points on each row or column of the FRFT. To achieve a wedge-shape, we keep α constant along the vertical axis, resulting in equispaced points along each ray, while linearly decreasing $|\alpha|$ on the horizontal axis to create a tapering effect. Indeed, such an approach will only compute vertical or

horizontal wedges. To calculate the angled wedges, we exploit the Fourier rotation property stating that the magnitude of the transform rotates by the same angle as the rotated image. So, we prerotate the image with by desired angle to acquire angled wedges corresponding to a rotated transform.

The general wedge-pair calculation procedure, illustrated in Fig. ??, is as follows.

- 1) Rotate the $n \times n$ input image with respect to the angle of the wedge, while zero-padding the image to $2n \times 2n$.
- 2) Perform FFTs in the vertical direction, shifting transform so that DC is at the center.
- 3) Perform FRFTs in the horizontal direction with variable α ranging from $-|\alpha_{\max}|$ to $+|\alpha_{\max}|$, after pre-multiplying with appropriate C to centralize the transform.

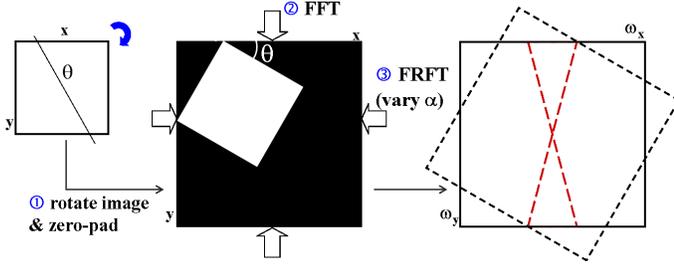


Fig. 4: Diagram showing the general method of computing points on a wedge-pair. θ is the angle of the wedge. The red dotted lines represent the angled wedge in frequency space that has been computed. The black dotted line shows the part of the FT of the rotated and zero-padded image that is equivalent to a transform of the original unrotated image.

The orthogonal counterpart to each wedge-pair (a second wedge-pair at a 90° angle to the first) can be calculated concurrently, simply by switching the horizontal and vertical directions in steps (2) and (3). This reduces the number of times the image matrix needs to be rotated. An alternative method to compute these points is to take the transpose of the image matrix (I_i^T) and repeat steps (2) and (3).

Special Cases: Transforms of certain orders can benefit from more accurate computational routines, namely the OP-transform (8-polar), where the image is rotated by 45° . The rotation and zero-padding step can be replaced by a change of variables resulting in a fast and exact rotation, as suggested by [4]. However, this change of variables also upsamples the image, requiring the use of FRFTs in step (2) of the general procedure, as well as appropriately scaled α .

B. Interpolation from GPP to Polar Grid

We recover the polar grid's Fourier coefficients by interpolating the coefficients calculated on the GPP-grid. Interpolation implementation requires knowledge of the relative distance between the GPP and polar grid points. Here, without going

into the lengthy details, we generalize the PP to polar grid interpolation algorithm of [6] for the case of GPP-grid. This algorithm is composed of two independent 1-D interpolation steps. For each wedge-pair, we formulate the GPP and corresponding polar grid coordinates that are used in each step¹. The first interpolation step, known as 'rotating the rays', involves transforming rays equispaced in gradient to rays equispaced in angle (Fig. 5a). By referring to the horizontal (angular) coordinate as k , and the vertical (radial) coordinate as l , the two sets of points are given by

$$\text{PP}_l = \frac{l}{n} \frac{k}{n} 2\pi \tan\left(\frac{\pi}{N}\right) \quad k \in \left(-\frac{n}{2} + 1, \frac{n}{2}\right) \quad (3)$$

$$\text{polar}_l = \frac{l}{n} \pi \tan\left(\frac{2\pi k}{nN}\right) \quad k \in \left(-\frac{n}{2} + 1, \frac{n}{2}\right). \quad (4)$$

The second interpolation step, known as 'circling the polygon', involves converting straight edges of a polygon into arcs of a circle (Fig. 5b). Note that for this step all points along a ray are scaled by the same factor $\cos\left(\frac{2\pi k}{nN}\right)$. The two sets of points are listed below:

$$\text{PP}_k = \frac{l}{n} \frac{\pi}{\cos\left(\frac{2\pi k}{nN}\right)} \quad l \in (-n + 1, n) \quad (5)$$

$$\text{polar}_k = \frac{l}{n} \pi \quad l \in (-n + 1, n) \quad (6)$$

The first interpolation step can be done on each row of I_i concurrently with Step 3 of Sect. III-A, after each row is computed. The second interpolation step is done as the final step of the algorithm. Due to angular symmetry, the same sets of GPP and polar grid-points can be used for interpolation in wedge-pairs at any angle.

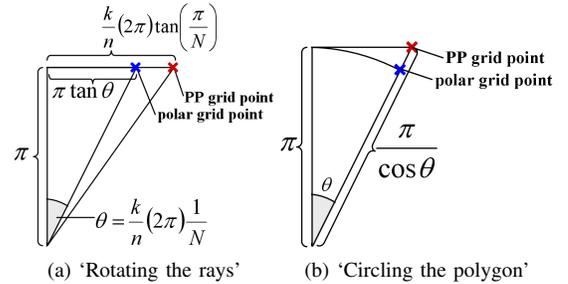


Fig. 5: (a) Rays of constant slope are converted to rays of constant angle. (b) Edges of a polygon are converted into arcs. While θ varies for each ray, each point on the ray is scaled by the same factor.

C. Complete Algorithm

In this section, we present a general algorithm for computing GPP transforms of any order. Note that we have split the algorithm into 2 sections to make use of some

¹Once the input and output (GPP and polar, respectively) coordinates are known, any 1-D interpolation function such as MATLAB's *interp1* implementation of the *cubic* (used for all experiments of this paper) interpolation maybe used to estimate the polar grid coefficients.

simplifications for the vertical and horizontal wedge-pairs.

A Compute vertical / horizontal wedge-pairs

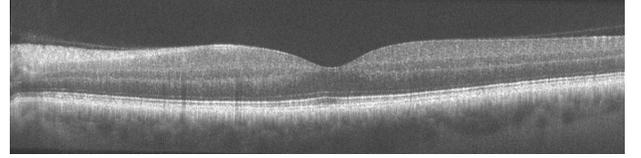
- A1 For each column
 i) zero-pad to length $2n$
 ii) perform FFT
 iii) shift transform so DC is at center.
- A2 For each row $l = -n+1:n$
 i) pre-multiply by factor
 $C = e^{i2\pi\alpha(\frac{n}{2}-1)\cdot x}$
 ii) perform FRFT with variable α
 where $\alpha = \frac{\gamma l}{n^2}$ and $\gamma = \tan \frac{\pi}{N}$
 iii) perform interpolation: 'rotating the rays'.
- A3 Label resulting set of points W_{V0} .
- A4 Perform interpolation: 'circling the polygon'.
- A5 Repeat steps A1-A4 on I^T .
- A6 Label resulting set of points W_{H0} .
- B If $N > 4$, compute angled wedge-pairs: For $q = 1: \frac{N}{4} - 1$

- B1 Rotate I by $q\frac{2\pi}{N}$ and resize to $2n \times 2n$.
 Label as \tilde{I}_q .
- B2 For each column
 i) perform FFT
 ii) shift transform so DC is at center.
- B3 For each row $l = -n+1:n$
 i) pre-multiply by factor
 $C = e^{i2\pi\alpha_2(n-1)\cdot x}$
 ii) perform FRFT with variable
 $\alpha = \frac{\gamma l}{n^2}$
 iii) perform interpolation: 'rotating the rays', reducing number of points to n .
- B4 Label resulting set of points $W_{V,q}$.
- B5 Perform interpolation: 'circling the polygon'.
- B6 Repeat steps B2-B5 on $(\tilde{I}_q)^T$.
- B7 Label resulting set of points $W_{H,q}$.
- By horizontally concatenating wedges in the following order, we obtain the complete transform: $W_{V,\frac{N}{4}-1}, \dots, W_{V1}, W_{V0}, W_{H,\frac{N}{4}-1}, \dots, W_{H1}, W_{H0}$.

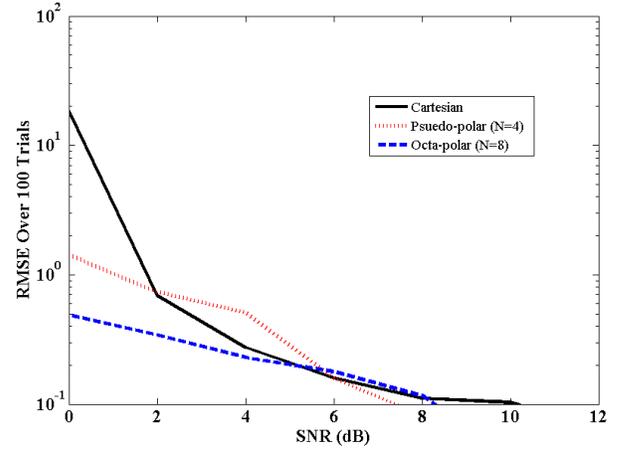
IV. EXPERIMENTS

Two sets of experiments on simulated and real ophthalmic SDOCT datasets were performed. The simulated experiment was set up by first averaging a set of 17 retinal images to significantly reduce noise. The quality of this (clean) image (Fig. 6a) was then degraded by adding Gaussian noise. The

same level of noise was added to a warped version of the clean image that was rotated by 1.8° . By registering these two frames using the Cartesian, PP, and OP methods, we estimated this rotation angle at different Signal to Noise Ratio² (SNR) levels. At each noise level, we repeated the experiment 100 times, and reported the root mean squared error (RMSE) of the estimated rotated angle (Fig. 6b). Note that the higher order grids are more efficient at low-SNR cases (SNR < 10dB), which correspond to the SNR of *in vivo* retinal SDOCT images. For higher SNR cases, all algorithms performed equally.



(a) Clean frame



(b) Root mean squared error plot

Fig. 6: Simulated Monte-Carlo registration experiment, comparing the accuracy of Cartesian, PP, and OP methods at different levels of additive Gaussian noise. For these experiments (a) was registered to its rotated by 1.8° version. (b) RMSE plot of rotation angle estimation error at different SNR levels.

The goal of the second (non-simulated) experiment is registering two images from the retina of a neonate captured directly by a handheld SDOCT system (Fig. 7a & Fig. 7b). A zoomed in section of these images is shown in Fig. 7c to better illustrate the retinal layers. We estimated the relative translation and rotation between these two frames using the Cartesian, PP, and OP methods. Using the estimated parameters, we warped the second image and averaged these images to reduce the speckle noise. The registered images are zoomed in Fig. 7d, Fig. 7e, and Fig. 7f, which correspond to the Cartesian, PP, and OP methods, respectively. In this example, the only accurate

²SNR is defined as $10 \log_{10} \frac{\sigma^2}{\sigma_n^2}$, where σ^2 , σ_n^2 are variance of a clean frame and noise, respectively.

registration which did not result in blurring of the retinal layers, was the result of the highest order grid (OP, $N = 8$).

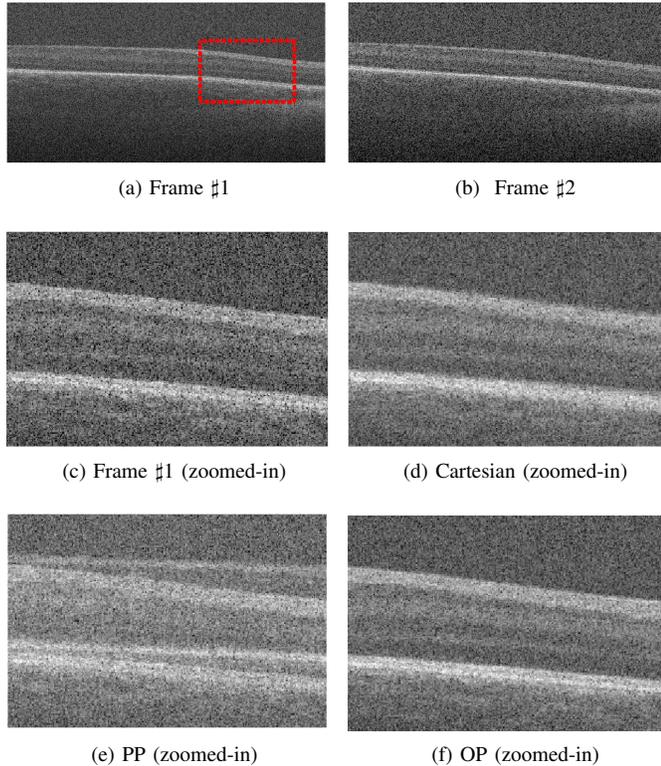


Fig. 7: Registration of two non-simulated SDOCT images from the retina of a neonate using different registration methods. The two images directly captured by the SDOCT system are shown in (a) and (b). Selected region of (a) as marked by the dashed lines is zoomed in (c) showing different retinal layers. Zoomed in section of the registered and averaged images using Cartesian, PP ($N = 4$), OP ($N = 8$) grids are shown in (d), (e), and (f), respectively.

V. CONCLUSION

In this paper, we proposed a novel framework for calculating the polar Fourier transform, called the GPP-transform. This framework is a generalization of the techniques known as PP [6] and OP [4] transforms, which are first and second order approximations of the GPP-transform. Aside from theoretical justifications, we experimentally showed that using higher order GPP-transforms, in some cases, increases motion estimation accuracy especially in lower SNR cases. We demonstrated the real world application of the high order GPP-transforms in registration of retinal SDOCT images of neonates.

Using higher order GPP-grids linearly increases the computational complexity. Ignoring the interpolation time required to rotate the wedges (Fig.4), the complexity of a N -polar transform is approximately $\frac{N}{4}$ times the complexity of the first-order transform. In other words, the 8-polar is more than double that of the 4-polar, the 12-polar is about 1.5 times that

of the 8-polar and so on. Implementation of the rotation in step B1 of Sect. III-C, for the GPP-grids of order $N > 8$ requires interpolation that might introduce some inaccuracies. Such errors do not occur in 4-polar (PP) transforms, since the only rotation involved is $\frac{\pi}{2}$.

If scale estimation is required, one needs to compute a log-polar transform. Since the points on each ray are equally spaced, interpolation to the log grid might become significantly inaccurate. A possible solution is incorporating the recent ‘multilayer’ method in [10], which exploits FRFTs to compute a second set of more densely spaced grid-points at the center of the transform, and provides a better grid for computing the log-polar coordinates. We note that the ‘multilayer’ method may also take advantage of the GPP-grids and these two frameworks are not mutually exclusive: it is possible to have a multi-layered N -polar transform. Investigation of a computationally more efficient algorithm is part of our ongoing work.

REFERENCES

- [1] L. G. Brown, “A survey of image registration techniques,” *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, December 1992.
- [2] M. Alkhanhal, D. Turaga, and T. Chen, “Correlation based search algorithms for motion estimation,” in *Picture Coding Symposium*, Apr. 1999.
- [3] H. S. Stone, M. Orchard, E.-C. Chang, and S. Martucci, “A fast direct Fourier-based algorithm for subpixel registration of images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 10, pp. 2235–2243, October 2001.
- [4] O. Harari, “A New Nearly Polar FFT and Analysis of Fourier-Radon Relations in Discrete Spaces,” Master’s thesis, Ben-Gurion University of the Negev, Beersheba, June 2007.
- [5] R. Mersereau and A. Oppenheim, “Digital reconstruction of multidimensional signals from their projections,” *Proceedings of the IEEE*, vol. 62, no. 10, pp. 1319–1338, Oct. 1974.
- [6] A. Averbuch, R. Coifman, D. Donoho, M. Elad, and M. Israeli, “Fast and accurate polar Fourier transform,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 2, pp. 145–167, 2006.
- [7] Y. Keller, A. Averbuch, and M. Israeli, “Pseudopolar-based estimation of large translations, rotations, and scalings in images,” *Image Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 12–22, Jan. 2005.
- [8] Y. Keller, Y. Shkolnisky, and A. Averbuch, “Volume registration using the 3-d pseudopolar Fourier transform,” *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4323–4331, Nov. 2006.
- [9] H. Liu, B. Guo, and Z. Feng, “Pseudo-log-polar fourier transform for image registration,” *Signal Processing Letters, IEEE*, vol. 13, no. 1, pp. 17–20, Jan. 2006.
- [10] W. Pan, K. Qin, and Y. Chen, “An adaptable-multilayer fractional fourier transform approach for image registration,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 3, pp. 400–414, March 2009.
- [11] L. Almeida, “The fractional fourier transform and time-frequency representations,” *Signal Processing, IEEE Transactions on*, vol. 42, no. 11, pp. 3084–3091, Nov 1994.
- [12] D. H. Bailey and P. N. Swartztrauber, “The Fractional Fourier Transform and Applications,” Mail Stop T27-A, Moffett Field, CA 94035, Tech. Rep. RNR-90-004, April 1990. [Online]. Available: citeseer.ist.psu.edu/bailey95fractional.html
- [13] S. B. Reddy and B. N. Chatterji, “An FFT-based technique for translation, rotation, and scale-invariant image registration,” *IEEE Transactions on Image Processing*, vol. 5, no. 8, pp. 1266–1271, August 1996.