Algebraically Accurate Volume Registration using Euler's Theorem and the 3-D Pseudo-polar FFT

Y. Keller[1], Y. Shkolnisky[2],A. Averbuch[2]
[1]Math department
Yale University, Connecticut, USA
Phone: +1-203-432-4345
Email: yosi.keller@yale.edu
[2]School of Computer Science
Tel Aviv University, Tel Aviv 69978, Israel
Submitted to the IEEE Transactions on Image Processing
August 2004

Abstract

This paper introduces an algorithm for the registration of rotated and translated volumes, which operates in the frequency domain. The Fourier domain allows to compute the rotation and translation parameters separately, thus reducing a problem with six degrees of freedom to two problems of three degrees of freedom each. We propose a three-step procedure. The first step estimates the rotation axis. The second computes the planar rotation relative to the rotation axis, and the third recovers the translational displacement by using the phase correlation technique. The rotation estimation is based on Euler's theorem, which allows to represent a rotation using only three parameters. Two parameters represent the rotation axis and one parameter represents the planar rotation perpendicular to the axis. By using the 3-D pseudo-polar FFT, the estimation of the rotation axis is shown to be algebraically accurate. A variant of the angular difference function registration algorithm is derived for the estimation of the planar rotation around the axis. The experimental results show that the algorithm is accurate and robust to noise.

1 Introduction

Rigid volume registration is a major component of 3-D object modelling in a diverse range of applications such as assembly of 3-D models from complementary patches [1, 2, 3], range imaging [4] and bio-informatics [5, 6].

*This work was supported by a grant from the Ministry of Science, Israel.

Let $V_1(\overrightarrow{x})$ and $V_2(\overrightarrow{x})$, $\overrightarrow{x} = (x, y, z)^T \in \mathbb{R}^3$ be two partially overlapping volumes, related by a 3-D rigid transformation, such that

$$V_1(\overrightarrow{x}) = V_2\left(R\overrightarrow{x} + \overrightarrow{\Delta x}\right), \tag{1.1}$$

where R is a 3-D rotation matrix and $\overrightarrow{\Delta x} = (\Delta x, \Delta y, \Delta z)^T \in \mathbb{R}^3$. The purpose of the registration process is to estimate R and $\overrightarrow{\Delta x}$.

There are several approaches to 3-D registration, which can be categorized as either feature or intensity based. Feature based approaches [2, 7, 8, 9] detect a set of features in the registered volumes, and align the volumes by using the coordinates of these features. Intensity based schemes [5, 10] align the volumes by using the intensity value of the voxels.

A large family of feature based algorithms is based on the ICP algorithm and its modifications. The ICP algorithm, first presented in [7], is based on finding the transformation that minimizes the Euclidean distance between pairs of corresponding points on the two surfaces. This is an iterative approach that at each iteration computes points correspondence based on the transformation found at the previous iteration, and then, finds the transformation that best maps points on one surface to their corresponding points on the other surface. The method assumes that the two surfaces are roughly aligned and therefore, each point on one surface corresponds to its closest point on the other surface. [7] proves that this method always converges to a local minimum. Thus, to obtain the correct registration, which corresponds to the global minimum, the ICP algorithm requires an initial guess that is close to the correct solution.

[8] gives a modification of the ICP algorithm, called the "iterative closest reciprocal point" (ICRP), which requires the closest-point relation to be symmetric. [11] then uses this algorithm to derive another modification of the ICP algorithm, which tries to avoid the local minimum problem and reduces the computational complexity of the ICP and ICRP algorithms. The algorithm computes local intrinsic differential structures using the curvature of the surface. These structures are invariant to Euclidean transformations. By using the local intrinsic differential structures, the algorithm generates a large numbers of potential transformations. Only transformations that pass a verification step are refined by using the ICRP algorithm, which uses only points on the local intrinsic differential structures. The refined transformations are verified again using the whole surfaces. Transformations that pass this second verification step are refined again by the ICRP algorithm using points from the whole surfaces. In the final step, the algorithm selects the transformation with the minimal average Euclidean distance of reciprocal points. Using several initial hypotheses reduces the probability that the ICRP algorithm will converge to a local minimum. The algorithm is not fully automatic as it requires the user to supply the level of mean curvature, which depends on the shape of the registered surfaces.

[2] gives a different acceleration scheme for the ICP algorithm. It accelerates the establishment of point correspondences, which is a very expensive procedure in the ICP algorithm, by using a z-buffer. This approach can be extended to simultaneous registration of N surfaces. Simultaneous registration, as opposed to a sequence of pair-wise registrations, distributes the registration error homogeneously between the different registered surfaces, thus reconstructing a more consistent object.

[9] introduces a generalization of the ICP algorithm, where instead of using the Euclidean distance as in [7], it defines a distance that is based on invariant features. Specifically, it defines a vector that contains information about curvature, moments, and spherical harmonics, and uses the

Euclidean distance on a vector that is a combination of these invariant features and the positional coordinates. The paper shows that like [7], the suggested algorithm converges monotonically to a local minimum.

The robustness of various ICP algorithms to the initial registration is studied in [12]. This paper shows that the performance of the algorithms greatly depend on the initial registration of the two objects, the parameter setting, and the noise level.

Most registration algorithms assume that the input images are error-free. That is, they ignore any acquisition errors. [13] suggests a method similar to the ICP that considers also the measurement errors in the input images. Similarly to the ICP, the algorithm assumes that the input images are coarsely aligned. One image is represented as a set of triangular patches and the other as a set of points. At each iteration, the algorithm establishes patch-to-point correspondence by using the current transformation estimate. Then, it computes a new transformation estimate by minimizing a cost function that takes into consideration the measurement error.

[14] suggests an optimization based algorithm for multiple view registration. The algorithm consists of two phases. The first phase, the local registration phase, registers pairs of views, while the second phase, the global registration phase, encodes the pair-wise registration results in a model graph, which encodes the connectivity between the overlapping views. The algorithm uses an optimization procedure to search this graph for a connected subgraph that contains a consistent model. This subgraph corresponds to the absolute pose of the input views.

The optimization approach in [15] defines the cost function as the sum of Euclidean distances between corresponding points in the two range images. Point correspondence is established by inverting the calibration equations of the acquisition device. The algorithm then uses the very fast simulated re-annealing method [16] to find the global minimum.

A problem that is closely related to registration is matching complementary objects. Complementary objects are objects that can be attached together to form a bigger object. Such problems arise, for example, in computer-aided manufacturing and archaeological applications. [3] presents an optimization based algorithm for this problem. The algorithm defines a matching error for a given pose based on the point-to-point distance between mutually visible faces of the parts. The matching error is efficiently computed by using the z-buffer algorithm and simulated annealing is used to minimize it.

An example for an iterative approach is presented by [17], which suggests a method for simultaneous registration and integration of multiple range images into a single object. The algorithm uses the signed distance field (SDF) for shape representation and performs integration, registration and outlier rejection by matching SDFs. The algorithm performs registration and integration alternately, until the input images are properly registered to the integrated object.

[18] suggests a surface representation scheme that captures the surface curvature and generates almost unique surface signatures. The registration procedure first establishes point correspondence by matching surface signatures, and then, uses this correspondence to estimate the transformation parameters.

Intensity based algorithms include optimization and Fourier based schemes. Optimization based schemes formulate the registration problem as the optimization of some cost function, such as the L_2 norm [19], and then use a general purpose optimization algorithm to optimize the cost function. [19] extends the widely used gradient methods to 3-D image registration. It uses Newton's method to minimize the L_2 norm of the intensity differences as a function of the motion parameters. Due to the properties of non-linear optimization, these algorithms are unable to es-

timate large motions. To estimate large motions, these algorithms are used in conjunction with a bootstrapping method, which computes a pre-alignment that is close to the optimum.

Frequency domain methods are of particular interest to the current paper. Such methods use the properties of the Fourier transform to estimate rotation and translation separately. This reduces a problem with six degrees of freedom to two problems with three degrees of freedom. [10] and [20] are examples of such methods. The algorithm [10] consists of three steps. The first recovers the rotation axis, the second recovers the rotation angle, and the third recovers the translation parameters. To recover the rotation parameters, the algorithm normalizes the Fourier transform of the input objects, and integrates it in the radial direction. The direction in which this integral is minimal gives the direction of the rotation axis. Once it recovers the rotation parameters, the algorithm recovers the translation parameters by using phase correlation. The integration in the radial direction suffers from inaccuracies caused by discretization. Therefore, the algorithm cannot achieve high accuracy. Nevertheless, it can be used as a good pre-alignment method for ICP based algorithms. Another frequency domain approach is given in [20]. This algorithm recovers the rotation parameters by formulating the problem as a linear system, whose entries are computed by the frequency domain relations of the two objects. As before, the translation parameters are recovered by using phase correlation. As in [10], the algorithm requires integration in the radial direction, which incurs inaccuracies. [5, 6] align density volumes for Protein-Protein docking by computing the polar Fourier transform of the density volumes. Thus, rotations are reduced to translations in the spherical coordinate system, which are recovered by using phase correlation [21].

In this paper we present a Fourier based approach, which does not require any interpolation to recover the rotation. It is based on the 2-D pseudo-polar FFT (PPFT2D) [22] and 3-D pseudo-polar FFT (PPFT3D) [23], which compute the Discrete Fourier Transform (DFT) on non-Cartesian grids. This allows a fast and algebraically accurate registration, which draws on Euler's theorem for the estimation of the 3-D registration parameters in three steps. First, the PPFT3D is used to recover the rotation axis. Then, the rotation around the axis is estimated using a pseudo-cylindrical representation computed with the PPFT2D. Finally, the translation is computed by using phase correlation [21]. The algorithm accurately estimates arbitrary large rotations without requiring an optimization scheme. It is robust to noise and the accuracy can be increase arbitrarily. The implementation requires only 1-D operations and is appropriate for real-time implementations. Moreover, the execution time is independent of the geometry of the registered volumes.

The paper is organized as follows. Section 2 gives the mathematical background relevant to Fourier based volume registration. Section 3 applies Euler's theorem to 3-D rotation and derives the rotation estimation scheme. Section 4 presents the PPFT3D, which is used in section 5 to estimate the rotation axis. Section 6 describes the computation of planar rotations and section 7 describes the estimation of the relative translation of the input volumes. Sections 8 and 9 present the experimental results and give some concluding remarks.

2 Rigid motion of volumes in the Fourier domain

Let $V_1(\overrightarrow{x})$ and $V_2(\overrightarrow{x})$ be two input volumes such that

$$V_1(\overrightarrow{x}) = V_2\left(R\overrightarrow{x} + \overrightarrow{\Delta x}\right), \tag{2.1}$$

where R is a 3-D rotation matrix and $\overrightarrow{\Delta x} = (\Delta x, \Delta y, \Delta z)^T \in \mathbb{R}^3$. Denote by $\widehat{V}(\overrightarrow{\omega}) \triangleq \mathcal{F}\{V(\overrightarrow{x})\}$ the Fourier transform of V, where $\overrightarrow{\omega} = (\omega_x, \omega_y, \omega_z)$. Then, [24]

$$\mathcal{F}\left\{V\left(R\overrightarrow{x} + \overrightarrow{\Delta x}\right)\right\} = \widehat{V}(R\overrightarrow{\omega})\, e^{\imath\left(\overrightarrow{\omega}\cdot\overrightarrow{\Delta x}\right)}. \tag{2.2}$$

By applying Eq. (2.2) to Eq. (2.1) we get

$$\widehat{V}_1(\overrightarrow{\omega}) = \widehat{V}_2(R\overrightarrow{\omega})\, e^{\imath\left(\overrightarrow{\omega}\cdot\overrightarrow{\Delta x}\right)} \tag{2.3}$$

and

$$M_1(\overrightarrow{\omega}) = M_2(R\overrightarrow{\omega}), \tag{2.4}$$

where

$$M_1 = |\widehat{V}_1|, \quad M_2 = |\widehat{V}_2|. \tag{2.5}$$

Hence, M_1 and M_2 are related by a rotation around the origin, which can be estimated first, regardless of the translation $\overrightarrow{\Delta x}$. Given the rotation R, the translation is recovered by computing the phase correlation $C(\overrightarrow{\omega})$ [21, 25, 26]. Specifically, by using Eq. (2.3) we define

$$\widehat{C}(\overrightarrow{\omega}) = e^{\imath\left(\overrightarrow{\omega}\cdot\overrightarrow{\Delta x}\right)} = \frac{\widehat{V}_2(R\overrightarrow{\omega})}{\widehat{V}_1(\omega)}. \tag{2.6}$$

The translation $\overrightarrow{\Delta x}$ is given by

$$\overrightarrow{\Delta x} = \arg\max C(\overrightarrow{x}), \tag{2.7}$$

where

$$C(\overrightarrow{x}) \triangleq \mathcal{F}^{-1}\{C(\overrightarrow{\omega})\} = \delta\left(\overrightarrow{x} - \overrightarrow{\Delta x}\right). \tag{2.8}$$

3 Euler's theorem and 3-D rotation estimation

Rotations in a 3-D Cartesian coordinate system may be represented using various formulations; in this paper we adopt the Euler angles representation [27], shown in Fig. 1a. 3-D rotations are expressed as a θ angle rotation about an axis whose direction is given by the angles α and β depicted in Fig. 1a. The angles α and β define the rotation axis $\overrightarrow{n} \triangleq (n_x, n_y, n_z)$, which is invariant under the 3-D rotation R.

Euler's Rotation Theorem An arbitrary 3-D rotation can be described as a rotation by an angle θ around an axis given by a unit vector $\overrightarrow{n} = (n_x, n_y, n_z)^T$.

The rotation matrix R is given by [27]

$$R = I\cos\theta + (1-\cos\theta)\begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_y n_x & n_y^2 & n_y n_z \\ n_z n_x & n_z n_y & n_z^2 \end{bmatrix} + \sin\theta \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ n_y & n_x & 0 \end{bmatrix}, \tag{3.1}$$

where I is the identity matrix and $\overrightarrow{n} = (n_x, n_y, n_z)$ is the rotation axis (see Fig. 1b). The representation of R is not unique. The same rotation is obtained by a rotation of $(-\theta)$ around the axis $(-\overrightarrow{n})$ [27].

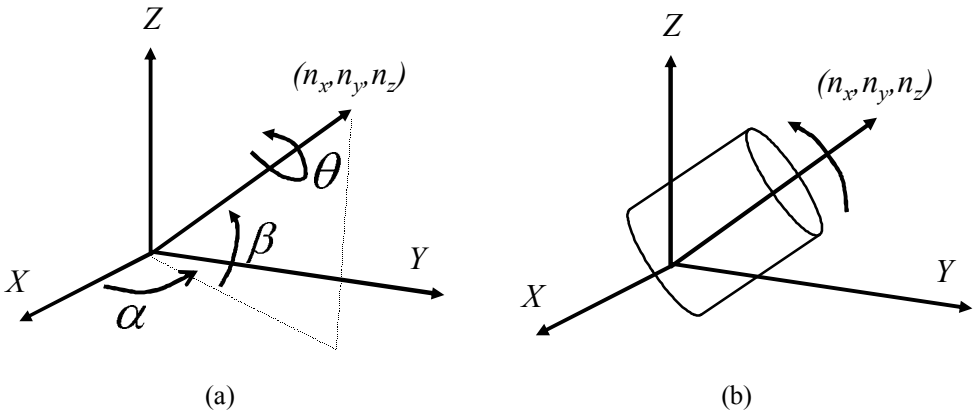


Figure 1: Euler angles and Euler's Rotation Theorem. (a) Definition of Euler angles: an arbitrary rotation is given by a rotation θ around the axis (n_x, n_y, n_z). The axis can be described by the angles α and β. (b) Euler's Rotation Theorem: any 3-D rotation can be expressed as a rotation about an axis.

Both θ and $\overrightarrow{n}$ can be easily recovered from the rotation matrix R. The three eigenvalues of R are $\lambda_1 = 1$ and $\lambda_{2,3} = e^{\pm\imath\theta}$. The rotation axis $\overrightarrow{n}$ can be computed as $\overrightarrow{n} = \overrightarrow{v_1}/\left|\overrightarrow{v_1}\right|$, where $\overrightarrow{v_1}$ is the eigenvector corresponding to λ_1, and θ can be recovered from $\lambda_{2,3}$. Thus, following Fig. 1b, any point $\overrightarrow{v}$ on the rotation axis $\overrightarrow{n}$ is invariant under R, as $\overrightarrow{n}$ is an eigenvector of R

$$R\overrightarrow{v} = R\overrightarrow{n}\left|\overrightarrow{v}\right| = \lambda_1\overrightarrow{n}\left|\overrightarrow{v}\right| = \overrightarrow{v}.$$

The rotation axis $\overrightarrow{n}$ can be recovered by finding the vector along which the difference between the volume and its rotated replica is minimal. Thus, given volumes V_1 and V_2, where $V_1(x) = V_2(Rx)$, the rotation axis can be recovered from ΔV, the 3-D angular difference function (ADF)

$$\Delta V\left(\alpha,\beta\right) = \int_0^\infty \left|V_1\left(\alpha,\beta,r\right) - V_2\left(\alpha,\beta,r\right)\right| dr, \tag{3.2}$$

where (α, β, r) are spherical coordinates, as

$$\left(\alpha_0,\beta_0\right) = \arg\min_{\widetilde{\alpha},\widetilde{\beta}} \Delta V\left(\widetilde{\alpha},\widetilde{\beta}\right). \tag{3.3}$$

Ideally $\Delta V\left(\alpha_0,\beta_0\right) = 0$, but due to noise and partial overlapping, we use Eq. (3.3) instead. For non-centered rotations, where the volumes V_1 and V_2 are translated and rotated, Eq. (3.2) can be applied to M_1 and M_2, the magnitudes of the Fourier transforms of V_1 and V_2.

Given the rotation axis $\overrightarrow{n}$, the volumes V_1 and V_2 are related by

$$V_1(R_n\overrightarrow{x}) = V_2(R_{z,\theta}R_n\overrightarrow{x}), \tag{3.4}$$

where R_n is a 3-D rotation which aligns $\overrightarrow{n}$ with the Z axis, and $R_{z,\theta}$ is a rotation of angle θ about the Z axis.

Thus, the framework of the proposed volume registration algorithm is as follows. Given volumes V_1 and V_2, where V_1 is a rotated and translated replica of V_2:

1. M_1 and M_2, the magnitudes of the Fourier transforms of V_1 and V_2, respectively, are computed. According to Eq. (2.4) they are related by a rotation around the origin.

2. The rotation axis $\overrightarrow{n}$ is recovered by computing $\Delta V\left(\alpha,\beta\right)$ and locating its minimum.

3. Given the rotation axis $\overrightarrow{n}$, V_1 and V_2 are rotated such that $\overrightarrow{n}$ is parallel to the Z axis, and the Fourier magnitudes of the rotated volumes are denoted $\widetilde{M_1}$ and $\widetilde{M_2}$, respectively.

4. $\widetilde{M_1}$ and $\widetilde{M_2}$ are related by a planar rotation of angle θ around the Z axis, which can be recovered by using 2-D registration.

5. Given (α,β,θ), the 3-D rotation matrix R is computed by using Eq. (3.1) and applied to V_2. $V_1\left(\overrightarrow{x}\right)$ and $V_2(R\overrightarrow{x})$ are related by a 3-D translation, which can be recovered using phase correlation.

We propose a fast and algebraically accurate scheme for the computation of the difference function ΔV in step #2, which is based on the PPFT3D [23]. A modification of the 2-D rotation estimation algorithm, given in [28], is then used to estimate the planner rotations in step #4.

4 The 3-D pseudo-polar FFT

Given an 3-D image I of size $N\times N\times N$, its 3-D Fourier transform, denoted $\hat{I}(\omega_x,\omega_y,\omega_z)$, is given by

$$\hat{I}(\omega_x,\omega_y,\omega_z)=\sum_{u,v,w=-N/2}^{N/2-1}I(u,v,w)e^{-\frac{2\pi\imath}{M}(u\omega_x+v\omega_y+w\omega_z)},\quad \omega_x,\omega_y,\omega_z\in\mathbb{R}. \tag{4.1}$$

We assume for simplicity that the image I has equal dimensions in the x, y, and z directions and that N is even. For ω_x, ω_y, and ω_z sampled on the Cartesian grid $(\omega_x,\omega_y,\omega_z)=(k,l,j)$, $k,l,j=-\frac{M}{2},\ldots,\frac{M}{2}-1$, the Fourier transform in Eq. (4.1) has the form

$$\hat{I}_{Cart}(k,l,j)\triangleq\hat{I}(k,l,j)=\sum_{u,v,w=-N/2}^{N/2-1}I(u,v,w)e^{-\frac{2\pi\imath}{M}(uk+vl+wj)}, \tag{4.2}$$

with $k,l,j=-\frac{M}{2},\ldots,\frac{M}{2}-1$, which is usually referred to as the 3-D 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. (4.2), can be computed in $O(M^3\log M)$ operations.

For some application it is desirable to compute the Fourier transform of I in spherical coordinates. Formally, we want to sample the Fourier transform in Eq. (4.1) on the grid $(\omega_x,\omega_y,\omega_z)$ where

$$\begin{gathered}\omega_x=r_k\cos\theta_l\sin\phi_j,\quad \omega_y=r_k\sin\theta_l\sin\phi_j,\quad \omega_z=r_k\cos\phi_j,\\ r_k=k,\quad \theta_l=2\pi l/L,\quad \phi=\pi j/J,\\ k=0,\ldots,M-1,\quad l=0,\ldots,L-1,\quad j=0,\ldots,J-1.\end{gathered} \tag{4.3}$$

The Fourier transform of I in spherical coordinates has the form

$$\hat{I}_{sph}(k,l) \triangleq \sum_{u,v,w=-N/2}^{N/2-1} I(u,v,w) e^{-\frac{2\pi\imath k}{M}(u\cos\theta_l \sin\phi_j + v\sin\theta_l \sin\phi_j + w\cos\phi_j)}. \tag{4.4}$$

The spherical grid in Eq. (4.3) is equally spaced both in the radial and angular directions

$$\Delta r = r_{k+1} - r_k = 1, \tag{4.5}$$

$$\Delta\theta = \theta_{l+1} - \theta_l = \frac{2\pi}{L}, \tag{4.6}$$

$$\Delta\phi = \phi_{j+1} - \phi_j = \frac{\pi}{J}. \tag{4.7}$$

Unfortunately, there is no fast algorithm for computing the Fourier transform of the image I in spherical coordinates.

The 3-D pseudo-polar Fourier transform (PPFT3D) evaluates the 3-D Fourier transform of an image on the 3-D pseudo-polar grid, which approximates the 3-D spherical grid, given in Eq. (4.3). Formally, the 3-D pseudo-polar grid is given by the set of samples

$$P \triangleq P_1 \cup P_2 \cup P_3, \tag{4.8}$$

where

$$P_1 \triangleq \{(k, -\frac{2l}{N}k, -\frac{2j}{N}k)\}, \tag{4.9}$$

$$P_2 \triangleq \{(-\frac{2l}{N}k, k, -\frac{2j}{N}k)\}, \tag{4.10}$$

$$P_3 \triangleq \{(-\frac{2l}{n}k, -\frac{2j}{n}k, k)\}, \tag{4.11}$$

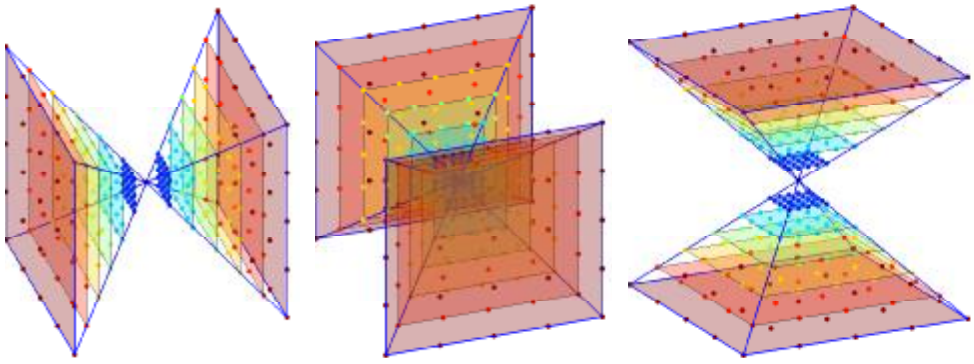
and $l, j = -\frac{N}{2}, \ldots, \frac{N}{2}$, $k = -\frac{3N}{2}, \ldots, \frac{3N}{2}$. See Figs. 2(a), 2(b), and 2(c) for an illustration of the sets P_1, P_2, and P_3, respectively. We define the 3-D pseudo-polar Fourier transform of I as the samples of the Fourier transform $\hat{I}$, given in Eq. (4.1), on the 3-D pseudo-polar grid P, given by Eqs. (4.8) – (4.11). Formally, the 3-D pseudo-polar Fourier transform $\hat{I}^s_{PP}$ $(s = 1, 2, 3)$ is a linear transformation, which is defined for $k = -\frac{3N}{2}, \ldots, \frac{3N}{2}$ and $l, j = -\frac{N}{2}, \ldots, \frac{N}{2}$, as

$$\hat{I}^1_{PP}(k,l,j) \triangleq \widehat{I}(k, -\frac{2l}{N}k, -\frac{2j}{N}k) = \sum_{u,v,w=-N/2}^{N/2-1} I(u,v,w) e^{-\frac{2\pi\imath}{M}(ku - \frac{2l}{N}kv - \frac{2j}{N}kw)}, \tag{4.12}$$

$$\hat{I}^2_{PP}(k,l,j) \triangleq \widehat{I}(-\frac{2l}{N}k, k, -\frac{2j}{N}k) = \sum_{u,v,w=-N/2}^{N/2-1} I(u,v,w) e^{-\frac{2\pi\imath}{M}(-\frac{2l}{N}ku + kv - \frac{2j}{N}kw)}, \tag{4.13}$$

$$\hat{I}^3_{PP}(k,l,j) \triangleq \widehat{I}(-\frac{2l}{n}k, -\frac{2j}{n}k, k) = \sum_{u,v,w=-N/2}^{N/2-1} I(u,v,w) e^{-\frac{2\pi\imath}{M}(-\frac{2l}{N}ku - \frac{2j}{n}kv + kw)}, \tag{4.14}$$

where $\hat{I}$ is given by Eq. (4.1).



(a) 3D pseudo-polar sector P_1 (b) 3D pseudo-polar sector P_2 (c) 3D pseudo-polar sector P_3

Figure 2: The 3-D pseudo-polar grid

As we can see from Figs. 2(a)–2(c), for fixed angles l and j, the samples of the 3-D pseudo-polar grid are equally spaced in the radial direction. However, this spacing is different for each angle. Also, the grid is not equally spaced in the angular direction, but has equally spaced slopes.

Two important properties of the 3-D 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, the implementations require only 1D equispaced FFT's. In particular, the algorithms do not require re-gridding or interpolation.

The algorithm for computing the 3-D pseudo-polar Fourier transform is based on the fractional Fourier transform. The fractional Fourier transform [29], with its generalization given by the chirp z-transform [30], is an 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

$$(F^{\alpha}X)(l) = \sum_{u=-N/2}^{N/2-1} X(u) e^{-2\pi\imath\alpha l u/N}, \quad l = -N/2, \ldots, N/2. \tag{4.15}$$

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

$$\omega_k = \alpha l/N, \qquad l = -N/2, \ldots, N/2, \tag{4.16}$$

and its complexity for a given vector X of length N and an arbitrary $\alpha \in \mathbb{R}$ is $O(N \log N)$ operations.

The algorithm for computing the 3-D pseudo-polar Fourier transform samples the Fourier transform of an image I on the pseudo-polar grid, with arbitrary frequency resolution in the radial and angular directions. The algorithm we present uses frequency resolution of $3N+1$ in the radial direction and $N+1$ in the angular directions. Denote,

1. E– Zero padding operator that accepts a 3-D image I of size $N \times N \times N$ and zero pads it to size $(3N+1) \times N \times N$ (along the x direction).

2. F – 1D DFT.

3. F_3 – 3-D DFT.

4. F_m^α – Fractional Fourier transform with factor α. The operator F_m^α accepts a sequence of length N, pads it symmetrically to length $3N+1$, applies to it the fractional Fourier transform with factor α, and returns the $N+1$ central elements.

5. $G_{k,n} \triangleq F_m^{2k/n} \circ F_n^{-1}$

The algorithm for computing $\hat{I}_{PP}^1$, given by Eq. (4.12), is then given by

1. Let
$$\widehat{I}_d \leftarrow F_3(E(I)).$$

2. For each k and l
$$U = \widehat{I}_d(k, l, \cdot)$$
and compute
$$T_1(k, l, \cdot) = G_{k,n} U.$$

3. For each k and j
$$V = T_1(k, \cdot, j)$$
and compute
$$T_1'(k, \cdot, j) = G_{k,n} V.$$

4. For each k,l,j
$$\hat{I}_{PP}^1(k, l, j) = T_1'(k, -l, -j).$$

The algorithm for computing $\hat{I}_{PP}^2$ and $\hat{I}_{PP}^3$, given by Eqs. (4.13) and (4.14), is similar. The complexity of the algorithm for computing $\hat{I}_{PP}^1$ is $O(N^3 \log N)$. Since the complexity of computing $\hat{I}_{PP}^2$ and $\hat{I}_{PP}^3$ is also $O(N^3 \log N)$, the total complexity of computing the 3-D pseudo-polar Fourier transform is $O(N^3 \log N)$.

5 Rotation axis estimation

An important property of $\Delta V\,(\alpha, \beta)$, given by Eq. (3.2), is that it can be discretized using very general sampling grids. Specifically, the discretization of ΔV does not require a uniform spherical representation of the Fourier transforms of V_1 and V_2. We present a discretization of ΔV that is based on the PPFT3D, presented in section 4. We denote by ΔV^d the discretization of ΔV. The algorithm for computing ΔV^d is

1. Compute M_1^d and M_2^d, where M_j^d is the magnitude of the PPFT3D of V_j, $j = 1, 2$.

2. Evaluate Eq. (3.2) as

$$\Delta V^d\left(\alpha_i,\beta_j\right)=\sum_{0\le r_k\le\pi}\left|M_1^d\left(r_k,\alpha_i,\beta_j\right)-M_2^d\left(r_k,\alpha_i,\beta_j\right)\right|\Delta r_{i,j}, \tag{5.1}$$

where $\Delta r_{i,j}$ is the radial sampling interval. Note that the integration is computed over rays of the same length ($0\le r_k\le\pi$) within the sphere bounded in a $N\times N\times N$ cube in the Fourier domain.

The PPFT3D was shown to be algebraically accurate [23], and hence, the approximation error in Eq. (5.1) results from the integration being computed over $0\le r_k\le r_{i,j}^{\max}$ instead of $0\le r_k\le\pi$, where for each ray $r_{i,j}^{\max}$ is the sample nearest to $r=\pi$ such that $r_{i,j}^{\max}\le\pi$. The integration interval error for each ray is

$$\left|\pi-r_{i,j}^{\max}\right|<\frac{\pi\sqrt{2}}{N}.$$

Furthermore, the interval $\left[r_{i,j}^{\max},\pi\right]$ is located at the high frequency range, where the magnitude of the PPFT3D is negligible.

5.1 The Normalized 3-D ADF

In order to improve the robustness of the algorithm with respect to noise and intensity changes, the L_1 norm used in Eq. (5.1) is replaced with the normalized correlation [31]. Thus. we define the normalized correlation ΔV_N^d as

$$\Delta V_N^d\left(\alpha_i,\beta_j\right)\triangleq\frac{\sum_{0\le r_k\le\pi}\left(\overline{M_1^d}\left(r_k,\alpha_i,\beta_j\right)-\overline{M_2^d}\left(r_k,\alpha_i,\beta_j\right)\right)^2}{\sigma_r\left(M_1^d\right)\sigma_r\left(M_2^d\right)}, \tag{5.2}$$

where

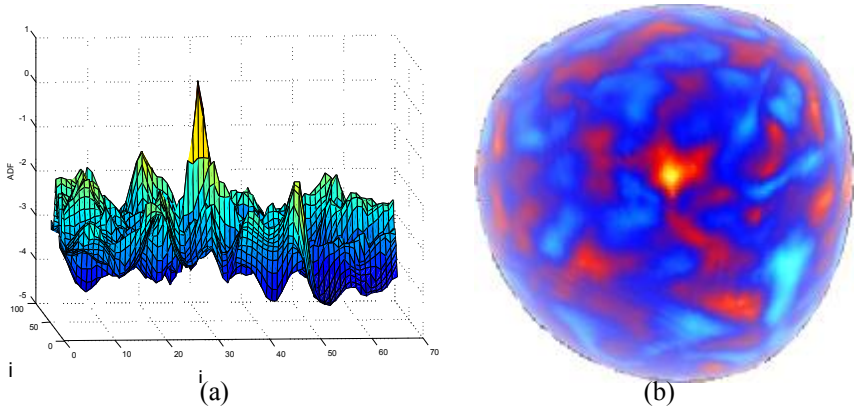
$$\overline{M_i^d}\left(r_j,\theta_i\right)\triangleq M_i^d\left(r_k,\alpha_i,\beta_j\right)-\frac{1}{j_{\max}}\sum_{0\le r_k\le\pi}M_i^d\left(r_k,\alpha_i,\beta_j\right),$$

$$r\left(k_{\max}\right)=\pi,$$

and

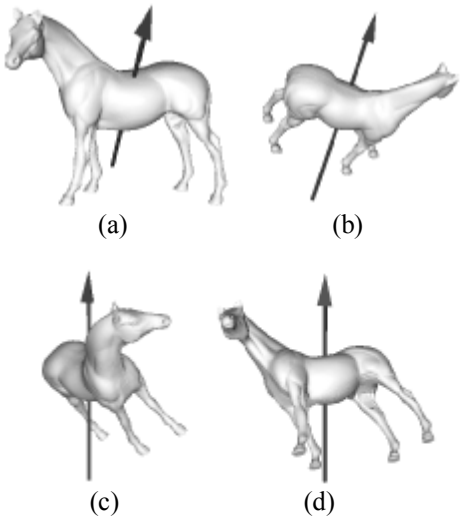
$$\sigma_r\left(M_i^d\right)\triangleq\sqrt{\frac{1}{j_{\max}}\sum_{0\le r_k\le\pi}\left(M_i^d\left(r_k,\alpha_i,\beta_j\right)-\overline{M_i^d}\left(r_k,\alpha_i,\beta_j\right)\right)^2}. \tag{5.3}$$

An example of ΔV_N^d for the Skull volume (Fig. 6e) is shown in Fig. 3. Fig. 3a is the surface $-\log(\Delta V_N^d)$, where the maximum is clearly visible and detectable. Figure 3b shows the small support of the minimum of ΔV_N^d and demonstrates the importance of using the PPFT3D. Achieving the same accuracy when computing ΔV_N^d using interpolation [10] would require significant computational complexity.



(a) (b)

Figure 3: The angular difference function ΔV_N^d for the rotated Skull volume shown in Fig. 6e. (a) A surface visualization of $-\log\left(\Delta V_N^d\right)$. The surface's peak corresponds to the minimum of the ΔV_N^d. (b) ΔV_N^d overlaid on a sphere. The "hot" values correspond to the minimum. Notice the small angular support of the minimum.



(a) (b)

(c) (d)

Figure 4: Alignment of the rotation axis. The volumes in (a) and (b) are the input volumes, which are related by a rotation about the rotation axis. After recovering the rotation axis, the volumes are rotated such that the rotation axis is parallel to the Z axis. Thus, the volumes in (c) and (d) are related by a translation and a planar rotation about the Z axis.

6 Planar rotation

Given the rotation axis $\overrightarrow{n}$, we use Eq. (3.4) to rotate V_1 and V_2 (Figs. 4a and 4b) such that the rotation axis $\overrightarrow{n}$ is parallel to the Z axis (Figs. 4c and 4d). This results in translated and rotated volumes, whose relative rotation is around the Z axis.

The 3-D rotation that aligns $\overrightarrow{n}$ with the Z axis can be computed using Quaternions [32]. Specifically, given two vectors $\overrightarrow{n}_1, \overrightarrow{n}_2 \in \mathbb{R}^3$, the 3-D rotation that transforms $\overrightarrow{n}_1$ to $\overrightarrow{n}_2$ is given by $\widetilde{R} = (\overrightarrow{n}, \psi)$, where

$$\psi = \arccos\left(\frac{\overrightarrow{n}_1 \cdot \overrightarrow{n}_2}{|\overrightarrow{n}_1|\,|\overrightarrow{n}_2|}\right) \tag{6.1}$$

is the rotation angle and

$$n = \overrightarrow{n}_1 \times \overrightarrow{n}_2 \tag{6.2}$$

is the rotation axis. In order to align the rotation axis computed by Eq. (3.3) with the Z axis, we set $\overrightarrow{n}_1 = (0, 0, 1)$, compute $\overrightarrow{n}_2$ by using Eq. (3.3), and use Eqs. (6.1) and (6.2) to compute the alignment parameters.

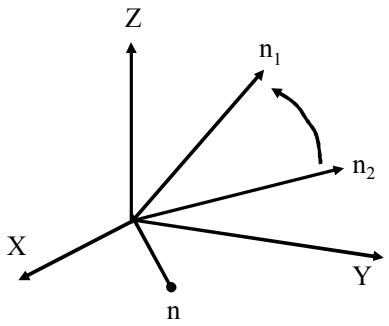


Figure 5: Computing the axes alignment parameters. Given two vectors $\overrightarrow{n}_1$ and $\overrightarrow{n}_2$, the rotation axis $\overrightarrow{n}$ is given by $\overrightarrow{n} = \overrightarrow{n}_1 \times \overrightarrow{n}_2$ and the rotation angle is given by $\psi = \arccos \frac{\overrightarrow{n}_1 \cdot \overrightarrow{n}_2}{|\overrightarrow{n}_1||\overrightarrow{n}_2|}$.

We apply $\widetilde{R}$, given by Eqs. (6.1) and (6.2), to V_1 and V_2 and denote the resulting volumes by $\widetilde{V}_1$ and $\widetilde{V}_2$, respectively. $\widetilde{V}_1$ and $\widetilde{V}_2$ are related by a translation and a planar rotation about the Z axis (see Fig. 4b). While it is possible to estimate the planar rotation by using any corresponding pair of XY planes in $\widetilde{V}_1$ and $\widetilde{V}_2$ [33, 34], we increase the robustness of the estimate by using a variant of the angular differencee function (ADF) [28] registration algorithm. The general framework in introduced in section 6.1 while the implementation issues are discussed in section 6.2.

6.1 ADF based estimation of planar rotations

In order to recover the relative planar rotation of $\widetilde{V}_1$ and $\widetilde{V}_2$, we define $\mathcal{F}^C$, the Cylindrical FFT. Let $V(x, y, x)$ be a 3-D volume. $\mathcal{F}^C$ is computed as follows:

1. Compute the 1-D Fourier transform in the Z direction

$$V_Z(x, y, \omega_z) = \mathcal{F}^Z_{1D}\{V(x, y, z)\}.$$

2. compute the 2-D Fourier transform of each XY plane in $V_Z(x, y, \omega_z)$

$$V(\theta, r, \omega_z) = \mathcal{F}_{2D}\{V_Z(x, y, \omega_z)\},$$

where $\mathcal{F}_{2D}$ is the 2-D Fourier transform.

Denote

$$\widetilde{M_1} = \left|\mathcal{F}^C(\widetilde{V_1})\right|, \qquad \widetilde{M_2} = \left|\mathcal{F}^C(\widetilde{V_2})\right|. \tag{6.3}$$

$\widetilde{M_1}$ and $\widetilde{M_2}$ are related by a planar rotation around the Z axis, with no relative translation. In other words, each XY plane in $\widetilde{M_1}$ is a rotated replica of the corresponding plane in $\widetilde{M_2}$

$$\widetilde{M_1}(\theta, r, \omega_z) = \widetilde{M_2}(\theta + \Delta\theta, r, \omega_z). \tag{6.4}$$

Next, we recover the rotation $\Delta\theta$ by computing the difference function $\Delta\widetilde{M}(\theta)$

$$\Delta\widetilde{M}(\theta) = \int_0^{\pi}\int_0^{\pi} \left|\widetilde{M_1}(\theta, r, \omega_z) - \widetilde{M_2}(-\theta, r, \omega_z)\right| dr\, d\omega_z. \tag{6.5}$$

We substitute Eq. (6.4) into Eq. (6.5) and get

$$\Delta\widetilde{M}(\theta) = \int_0^{\pi}\int_0^{\pi} \left|\widetilde{M_1}(\theta, r, \omega_z) - \widetilde{M_1}(-\theta - \Delta\theta, r, \omega_z)\right| dr\, d\omega_z,$$

where the value of $\Delta\widetilde{M}(\theta_0)$ is zero if

$$\theta_0 + \Delta\theta = -\theta_0 \qquad \text{or} \qquad \theta_0 + \Delta\theta = -\theta_0 + \pi, \tag{6.6}$$

and the second zero in Eq. (6.4) is due to the conjugate symmetry of $\widetilde{M_1}$ and $\widetilde{M_2}$. Thus, the two zeros of $\Delta\widetilde{M}(\theta)$, obtained at θ_0^1 and θ_0^2, are related to the rotation $\Delta\theta$ by

$$\theta_0^{(1)} = -\frac{\Delta\theta}{2}, \qquad \theta_0^{(2)} = -\frac{\Delta\theta}{2} + \frac{\pi}{2}. \tag{6.7}$$

We see from Eq. (6.7) that the zeros θ_0^1 and θ_0^2 are $\pi/2$ radians apart and this property is true in general. For each zero θ_0 of $\Delta\widetilde{M}(\theta)$, $\theta_0 + \frac{\pi}{2}$ is also a zero. Therefore, we define the 2-D angular difference function by

$$\Omega(\theta) = \Delta\widetilde{M}(\theta) + \Delta\widetilde{M}\left(\theta + \frac{\pi}{2}\right), \qquad \theta \in \left[0, \frac{\pi}{2}\right]. \tag{6.8}$$

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

$$\theta_0 = -\frac{\Delta\theta}{2}. \tag{6.9}$$

Due to the conjugate symmetry, the rotation angle can be either $\Delta\theta$ or $\Delta\theta + \pi$. The ambiguity is resolved by rotating V_1 using the rotations defined by $(\alpha_0, \beta_0, \Delta\theta)$ and $(\alpha_0, \beta_0, \Delta\theta + \pi)$, recovering the translation, and choosing the parameters the correspond to the best alignment.

6.2 Numerical implementation

The scheme is discretized similarly to the 2-D rotation estimation algorithm presented in [28]. Given the discrete input volumes V_1^d and $\widetilde{V}_2^d$, the discrete Cylindrical FFT $\mathcal{F}_d^C$ is computed by applying the 1-D FFT in the Z direction

$$V_Z^d(i,j,\omega_k) = FFT_{1D}^Z\left\{V(i,j,k)\right\}.$$

and computing the 2-D pseudo-polar FFT of each XY plane in $V_Z^d(i,j,\omega_k)$

$$V^d(\theta_i, r_i, \omega_k) = PPFT2D\left\{V_Z^d(\theta_i, r_i, \omega_k)\right\}.$$

The reversal of the angular axis in Eq. (6.5) is given by

$$\widetilde{M_2}(-\theta_i, r_i, \omega_k) = \left|\mathcal{F}_d^C\left\{FlipLR_{XY}\left\{V_2^d(i,j,k)\right\}\right\}\right|,$$

where $FlipLR_{XY}\left\{V(i,j,k)\right\}$ is a discrete volume where each XY plane is a flipped left-right replica of the corresponding plane in V. In the 2-D pseudo-polar grid, for each angle θ, the grid also contains the angle $\theta + \pi/2$, and therefore, Eq. (6.8) can be accurately computed.

7 3-D Translation estimation

Given the rotation parameters $(\alpha, \beta, \Delta\theta)$, the 3-D rotation matrix R is given by Eq. (3.1). Denote by $\widetilde{V}_2$ the replica of V_2 rotated by R. V_1 and $\widetilde{V}_2$ are related by a 3-D translation, which is recovered using the phase-correlation algorithm [21, 33, 34], given in Eqs. (2.7) and (2.8). The peak of the phase correlation function $C(\overrightarrow{x})$ measures the quality of the alignment. Thus, given the hypotheses $(\alpha, \beta, \Delta\theta)$ and $(\alpha, \beta, \Delta\theta + \pi)$, computed in section 6.1, the hypothesis that corresponds to the highest correlation peak is chosen as the correct registration. The accuracy of the phase correlation scheme is limited to integral values. Subpixel accuracy and improved robustness to noise can be achieved by applying either [26] or [35].

8 Experimental Results

The proposed algorithm was applied to the volumes shown in Fig. 6. For each input volume, a set of rotated and translated replicas was created using bilinear interpolation, without applying any other processing, such as smoothing and denoising. All the volumes were of size 64^3 voxels, and the average spacing of the PPFT3D grid was $\Delta\alpha = \frac{90^\circ}{2N} = 0.7^\circ$ and $\Delta\beta = \frac{180^\circ}{N} = 1.4^\circ$. The translations were randomly chosen in the range of $[-10, 10]$ pixels in each axis. We do not present the accuracy of neither the translation estimation nor the resolving between θ and $\theta + \pi$, since these are computed by a straightforward implementation of the 2-D phase-correlation algorithm and are not the focal point of this paper.

The results, given in Table 1, show the accuracy of the proposed algorithm. In all cases the registration accuracy was in the order of the angular spacing of the 3-D pseudo-polar grid. The accuracy of the registration was the same for all volumes.

Figure 7 presents σ_ε, the standard deviation (STD) of the angular registration error, as a function of σ_n, the STD of the additive White Gaussian Noise (WGN). The results were computed by

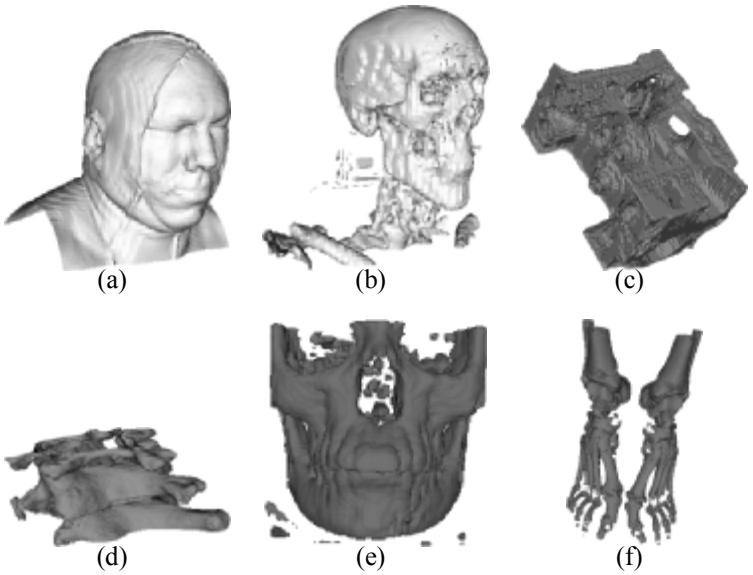


Figure 6: Volumes used to evaluate the algorithm. (a) and (b) are iso-surface of the same MRI scan of a human head. (c) A CAD generated model of an engine. (d) A MRI scan of a human spine. (e) A CT scan of a human skull. (f) A MRI scan of human feet.

	Actual parameters			Estimated parameters			Estimated errors		
	α	β	θ	α	β	θ	$\lvert\varepsilon_\alpha\rvert$	$\lvert\varepsilon_\beta\rvert$	$\lvert\varepsilon_\theta\rvert$
Human Head	82.89	45.00	28.21	83.39	47.81	28.11	0.50	2.81	0.10
	80.98	25.84	44.50	80.36	25.11	44.80	0.62	0.73	0.30
	12.02	15.11	78.47	11.89	15.70	79.42	0.13	0.59	0.95
Engine	82.89	45.00	28.21	81.37	43.19	27.78	1.51	1.80	0.42
	80.98	25.84	44.50	80.78	24.15	44.86	0.19	1.68	0.36
	12.02	15.11	78.47	12.88	15.67	76.94	0.86	0.56	1.52
Spine	82.89	45.00	28.21	84.46	43.76	26.36	1.57	1.23	1.84
	80.98	25.84	44.50	80.07	27.21	44.33	0.90	1.37	0.16
	12.02	15.11	78.47	11.03	13.80	79.94	0.98	1.30	1.47
Skull	82.89	45.00	28.21	84.35	43.68	29.94	1.46	1.31	1.73
	80.98	25.84	44.50	79.90	27.81	43.55	1.0	1.97	0.94
	12.02	15.11	78.47	13.23	14.86	77.11	1.21	0.24	1.35
feet	82.89	45.00	28.21	84.52	44.36	29.70	1.63	0.63	1.49
	80.98	25.84	44.50	79.90	25.09	43.45	1.0	0.74	1.0
	12.02	15.11	78.47	10.97	14.57	79.05	1.0	0.53	0.58

Table 1: Registration results using the non-normalized ADF ΔV^d for the volumes shown in Fig. 6. Actual rotation parameters (columns 1 through 3) and their estimates (columns 4 through 6), with relative errors, (columns 7 through 9). All the volumes were of size $64 \times 64 \times 64$.

adding a range of noise signals to the Skull and Head volumes, shown in Fig. 6, and averaging the results that correspond to the same noise level σ_n. It follows that for non-noisy input volumes the accuracy of the proposed algorithm is in the range of $0^\circ - 1^\circ$, which corresponds to the average angular spacing of the 3-D pseudo-polar grid. The error of the normalized ADF ΔV_N^d is less sensitive to additive noise. Up to $\sigma_n = 200$ the scheme is not affected by the noise, and from that point on, its accuracy degrades as a function of the noise. The non-normalized ΔV^d achieves an accuracy of $\sigma_\varepsilon < 1$ for non-noisy volumes and then degrades. To conclude, the proposed algorithm is capable of giving reasonable estimates in extremely noisy situations.

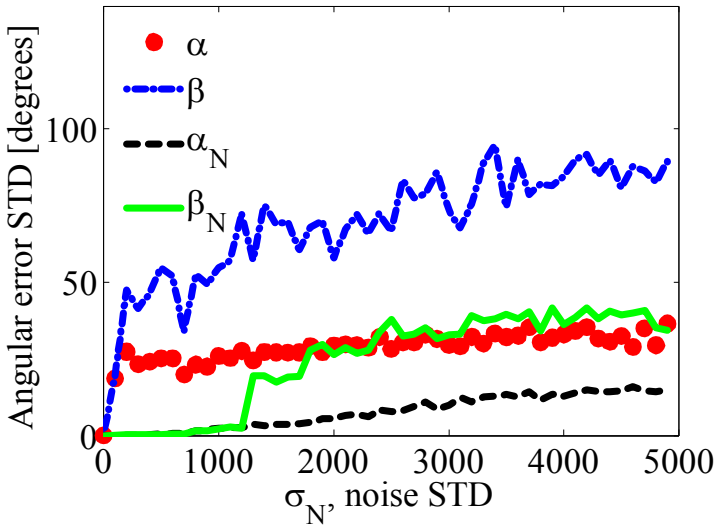


Figure 7: Angular registration error as a function of the standard deviation of the noise. α and β are the errors in the estimation of the rotation axis when using the non-normalized angular difference function ΔV^d. α_n and β_n are the estimation errors when using the normalized difference function ΔV_N^d. Using ΔV_N^d results in improved accuracy especially for the noisy volumes.

The execution time on a P2800 computer is approximately 3 minutes for volumes of size 64^3 using a non-optimized C++ implementation. Most of the time (2 minutes) is spent on computing the PPFT3D. The difference in the execution time between ΔV_N^d and ΔV^d is negligible.

9 Summary and conclusions

The paper presents a general purpose volume registration algorithm which operates in the frequency domain and solves the alignment problem by computing the ADF in two and three dimensions. By using Euler's theorem, the original problem, which involves six parameters, is decoupled into three sub-problems: estimating the rotation axis, estimating the planar rotation, and computing the translation. The computation of the ADF is based on the PPFT2D and PPFT3D and is algebraically accurate. This results in improved registration accuracy and robustness to noise. Future work includes the application of the proposed algorithm to medical imaging problems utilizing its robustness to noise.

References

[1] J. Wyngaerd and L. Van Gool, "Automatic crude patch registration: Toward automatic 3D model building," *Computer Vision and Image Understanding*, vol. 87, no. 1-3, pp. 8–26, July 2002.

[2] Raouf Benjemaa and Francis Schmitt, "Fast global registration of 3D sampled surfaces using a multi-Z-buffer technique," in *International Conference on Recent Advances in 3D Digital Imaging and Modeling*, 1997, pp. 113–120.

[3] Georgios Papaioannou, Evaggelia-Aggeliki Karabassi, and Theoharis Theoharis, "Reconstruction of three-dimensional objects through matching of their parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 114–124, January 2002.

[4] Marcos Rodrigues, Robert Fisher, and Yonghuai Liu, "Introduction: Special issue on registration and fusion of range images," *Computer Vision and Image Understanding*, vol. 87, pp. 1–7, 2002.

[5] J. A. Kovacs, P. Chacón, Y. Cong, E. Metwally, and W. Wriggers, "Fast rotational matching of rigid bodies by fast Fourier transform acceleration of five degrees of freedom," *Acta Cryst.*, vol. D59, no. 2, pp. 1371–1376, 2003.

[6] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A. A. Friesem, C. Aflalo, and I. A. Vakser, "Molecular surface recognition: Determination of geometric fit between proteins and their ligands by correlation techniques," in *Proc. Natl. Acad. Sci. USA*, vol. 89, March 1992, pp. 2195–2199.

[7] Paul J. Besl and Neil D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–255, February 1992.

[8] T. Pajdla and L. Van Gool, "Matching of 3-D curves using semi-differential invariants," in *5th International Conference on Computer Vision*. IEEE Computer Society Press, Cambridge, MA, 1995, pp. 390–395.

[9] Gregory C. Sharp, Sang W. Lee, and David K. Wehe, "ICP registration using invariant features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 90–102, January 2002.

[10] G. C. L. Lucchese, G. Doretto, "A frequency domain technique for 3-D view registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1468–1484, 2002.

[11] Pavel Krsek, Tomáš Pajdla, and Václav Hlaváč, "Differential invariant as the base of triangulated surface registration," *Computer Vision and Image Understanding*, vol. 87, pp. 27–38, 2002.

[12] G. Dalley, "Pair-wise range image registration: A study in outlier classification," *Computer Vision and Image Understanding*, vol. 87, pp. 104–115, 2002.

[13] I. S. Okatani, "A method for fine registration of multiple view range images considering the measurement error properties," *Computer Vision and Image Understanding*, vol. 87, pp. 66–77, 2002.

[14] Daniel F. Huber and Martial Hebert, "Full automatic registration of multiple 3D data sets," *Image and Vision Computing*, vol. 21, pp. 637–650, 2003.

[15] Gérard Blais and Martin D. Levine, "Registering multiview range data to create 3D computer objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 820–824, August 1995.

[16] L. Ingber, "Very fast simulated reannealing (VFSR)," *Mathematical and Computer Modeling*, vol. 12, no. 8, 1989.

[17] T. Masuda, "Registration and integration of multiple range images by matching signed distance fields for object shape modeling," *Computer Vision and Image Understanding*, vol. 87, pp. 51–65, 2002.

[18] Sameh M. Yamany and Aly A. Farag, "Surface signatures: An orientation independent free-form surface representation scheme for the purpose of objects registration and matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1105–1120, August 2002.

[19] P. Thévenaz, U. Ruttimann, and M. Unser, "A pyramid approach to subpixel registration based on intensity," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 27–41, January 1998.

[20] Guido Maria Cortelazzo, Gianfranco Doretto, and Luca Lucchese, "Free-form textured surfaces registration by a frequency domain technique," *International Conference on Image Processing (ICIP)*, vol. 1, pp. 813–817, October 1998.

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

[22] A. Averbuch, D. Donoho, R. Coifman, M. Israeli, and Y. Shkolnisky, "Fast slant stack: A notion of Radon transform for data in cartesian grid which is rapidly computable, algebrically exact, geometrically faithful and invertible," *SIAM Scientific Computing*, To appear.

[23] A. Averbuch and Y. Shkolnisky, "3D Fourier based discrete Radon transform," *Applied and Computational Harmonic Analysis*, vol. 15, pp. 33–69, 2003.

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

[25] 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.

[26] W. Hoge and D. Mitsouras, "Registration of multi-dimensional image data via sub-pixel resolution phase correlation," *International Conference on Image Processing (ICIP03)*, vol. 2, pp. 707–710, September 2003.

[27] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. New Jersey: Prentice-Hall, 1998, vol. pp. 333 - 334.

[28] Y. Keller, Y. Shkolnisky, and A. Averbuch, "The angular difference function and its application to image registration," *Submitted to the IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

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

[30] 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.

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

[32] D. Hearn and M. P. Baker, *Computer Graphics, C Version*, 2nd ed. Prentice-Hall, 1994.

[33] P. Milanfar, "Two-dimensional matched filtering for motion estimation," *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 438–443, March 1999.

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

[35] Y. Keller and A. Averbuch, "A projection-based extension of the phase correlation method," *Submitted to the IEEE Transactions on Signal Processing*, 2004.