Efficient Registration of Nonrigid 3-D Bodies

Huizhong Chen, *Student Member, IEEE*, and Nick Kingsbury, *Member, IEEE*

***Abstract*—We present a novel method to perform an accurate registration of 3-D nonrigid bodies by using phase-shift properties of the dual-tree complex wavelet transform (*DT-ℂWT*). Since the phases of *DT-ℂWT* coefficients change approximately linearly with the amount of feature displacement in the spatial domain, motion can be estimated using the phase information from these coefficients. The motion estimation is performed iteratively: first by using coarser level complex coefficients to determine large motion components and then by employing finer level coefficients to refine the motion field. We use a parametric affine model to describe the motion, where the affine parameters are found locally by substituting into an optical flow model and by solving the resulting overdetermined set of equations. From the estimated affine parameters, the motion field between the sensed and the reference data sets can be generated, and the sensed data set then can be shifted and interpolated spatially to align with the reference data set.**

***Index Terms*—Dual-tree complex wavelet transform (DT-ℂWT), image registration, optical flow.**

I. Introduction

With the increasing availability of 3-D imaging systems such as computed tomography (CT) and magnetic resonance imaging (MRI), multidimensional image analysis has become a key topic for research. In most image processing tasks that involve combining data from multiple sources, accurate estimation of motion between data sets is of great importance. The objective of image registration is to align geometrically multiple images of a similar scene, which are acquired at different times and positions and with different imaging devices. Registration is widely used in many applications such as medical imaging (e.g., aligning data sets for disease diagnosis and treatment planning), computer vision (e.g., object tracking and structure-from-motion), and remote sensing (e.g., change detection, mosaicing, image fusion, and super resolution). In this paper, we mainly consider applying the registration algorithm to medical images. Medical image registration differs from other 3-D object registrations in three main aspects: 1) images of a medical object can change significantly with time due to elastic tissue structures or in the presence of abnormalities; 2) medical image registration tends to focus on internal object structures; 3) accurate registration must be achieved for optimal diagnosis. However, it should be noted that our algorithm does not make assumptions about specific types of image data; therefore, the proposed method can be generalized to other image categories.

3-D motion estimation has been studied for some time, but now, improved computation and memory resources make higher performance methods increasingly feasible. In 3-D registration, motion can occur along all three dimensions; therefore, it is suboptimal to register data sets slice by slice. In addition, for nonrigid objects such as a human tissue, the movement is typically nonuniform throughout the data set; hence, the motion should be described locally rather than just globally. Further, the object of interest may have experienced changes in the sensed and the reference data sets, e.g., the removal of tumor in medical images before and after a clinical intervention. Hence, a good multidimensional registration algorithm should be accurate, robust, and computationally efficient.

Previous work on image registration can be broadly classified into feature-based and area-based methods. Feature-based approaches extract salient features from the sensed and reference data sets and aim to find the transformation that minimizes the distance between corresponding features. The extracted features can be regions, edges, or interest points. Matas *et al.* [2] detect maximally stable extremal regions and set up the correspondence between pairs of images based on these stable regions. Li *et al.* [3] proposed a contour-based method, which uses region boundaries and other strong edges as matching primitives. In [4], Takacs *et al.* use Features from Accelerated Segment Test keypoints [5] and perform descriptor matching to estimate the image transformation. Ta *et al.* [6] proposed SURFTrac, which matches Hessian interest points inside a 3-D image pyramid. In contrast to feature-based methods, area-based methods attempt to perform registration without extracting salient features. Common area-based approaches include cross correlation (CC) methods and mutual information (MI) methods. For CC methods such as the algorithms described in [7] and [8], a displacement vector is computed over pairs of tiles in the sensed and the reference images by maximizing the normalized CC. Another type of a registration technique that is closely related to the CC methods is the sequential similarity detection algorithm, which maximizes the sum of absolute differences of the image intensity [9]. Image registration with MI was first proposed in [10] and [11], where the registration is done by optimizing the MI between pairs of data sets. Thevenaz *et al.* [12] employed Parzen windows to compute the joint probability histogram and developed an MI optimizer with the Marquardt–Levenberg method for multimodal image registration. In [13], the joint probability was approximated by discrete histograms, and the maximization of MI was achieved by a multiresolution hill-climbing algorithm. For a more complete review on image registration methods, readers are referred to [14]–[19].

Manuscript received October 12, 2010; revised January 27, 2011 and May 13, 2011; accepted June 16, 2011. Date of publication June 30, 2011; date of current version December 16, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. T. Gevers.

H. Chen is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305-9505 USA (e-mail: hchen2@stanford.edu).

N. Kingsbury is with the Department of Engineering, University of Cambridge, CB2 1PZ Cambridge, U.K. (e-mail: ngk@eng.cam.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2160958

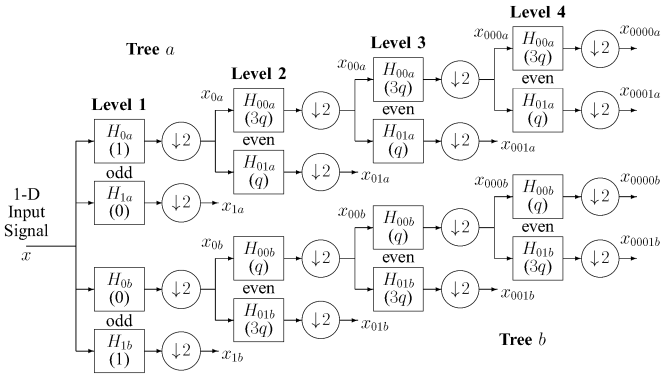


Fig. 1. Dual tree of real filters for the Q-shift CWT, giving real and imaginary parts of complex coefficients from trees a and b, respectively. Figures in brackets indicate the approximate delay for each filter, where $q = 1/4$ sample period (adapted from [1]).

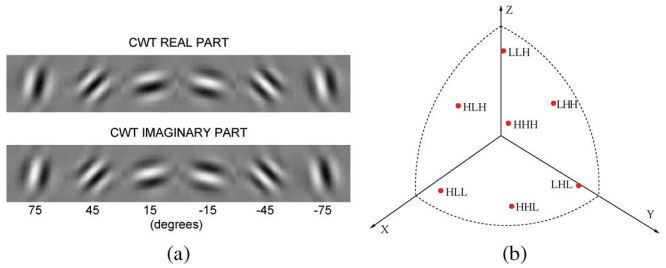


Fig. 2. (a) Impulse responses of the 2-D DT-$\mathbb{C}$WT subbands at level 4, in the order HL, HH, LH, $\mathrm{L}^*\mathrm{H}$, $\mathrm{H}^*\mathrm{H}$, and $\mathrm{H}^*\mathrm{L}$. The filtering is performed on rows first and then on columns, e.g., LH means low-pass filtering on rows followed by high-pass filtering on columns. (b) Orientation of 3-D DT-$\mathbb{C}$WT subbands on one quadrant of a hemisphere. The filtering is performed in the sequence: rows, columns, and slices.

Our nonrigid-body registration algorithm is based on the ideas of Hemmendorff [20], [21] with significant changes designed to improve computational efficiency. In particular, the dual-tree complex wavelet transform (DT-$\mathbb{C}$WT) [1], [22] is used as the front-end filter bank to take advantage of its near shift invariance and directional selectivity, which are combined with relatively low redundancy and computation load. The algorithm is fully automated and may be applied to estimate motion for a wide range of nonrigid objects. In addition, since the motion of a rigid body is just a special case of nonrigid motion, our proposed motion registration method can also be used to register rigid objects, although difficulties may arise at sharp motion boundaries due to the spatial support regions of the filters used. Finally, our image registration algorithm is based on aligning the phases of the DT-$\mathbb{C}$WT coefficients, which are robust to local mean and contrast changes of the two data sets being registered. The proposed algorithm is well suited to medical data where any large motion is mainly global, and smaller local relative motions can be obtained from the finer resolution motion estimation in later stages of the algorithm. To handle very large local motion, an extension of our method is to perform motion estimation by searching over adjacent regions where the local motion has occurred.

II. DT-$\mathbb{C}$WT

This section contains a brief introduction to the DT-$\mathbb{C}$WT and its properties for image registration. The DT-$\mathbb{C}$WT in 3-D will be discussed in some detail since we focus on registering 3-D data sets.

A. DT-$\mathbb{C}$WT and Its Properties

The DT-$\mathbb{C}$WT is an enhancement of the conventional discrete wavelet transform (DWT), with two distinctive properties: near shift invariance and directional selectivity in 2-D or higher dimensions [23]. The basic idea of the DT-$\mathbb{C}$WT is to employ a Hilbert pair of real DWTs in parallel [22] to produce the real and imaginary parts of complex wavelet coefficients. Its framework for 1-D signals is shown in Fig. 1.

The near-shift-invariance property of the DT-$\mathbb{C}$WT means that the impulse response of a given subband from the transform input to the inverse-transform output is approximately independent of shift and, hence, free of aliasing [1]. Simoncelli *et al.* [24] showed that shift invariance is equivalent to interpolability of the subband coefficients. Hence, for image registration purposes, the DT-$\mathbb{C}$WT coefficients from each subband can be interpolated to represent the transformation of shifted signals in the spatial domain.

The 1-D DT-$\mathbb{C}$WT can be extended to 2-D to produce directionally selective wavelet subbands [22], [23], as shown in Fig. 2(a). The impulse responses of the oriented wavelet filters are important for motion estimation because the motion in the direction normal to the stripes of the filter response will cause an approximately linear phase change in the corresponding subband coefficients. In other words, the DT-$\mathbb{C}$WT has an analogy to the Fourier transform, in the way that a shift in the spatial domain corresponds to a linear phase change in the frequency domain. For the DT-$\mathbb{C}$WT in 3-D, each transform level has 28 subbands, which are selective to near-planar surfaces. The orientations of these surfaces for the 28 subbands correspond to approximately equally spaced patches on the surface of a hemisphere; one quadrant of which is shown in Fig. 2(b).

The bandwidths of the wavelet filters are approximately one octave wide, which are well suited to detection of edges (2-D) or surfaces (3-D), since they are wide enough in frequency to be well localized in space and yet narrow enough for their responses to approximate modulated waves with linear phase-versus-displacement characteristics. At any one scale, they approximately uniformly tile the 2-D frequency plane or 3-D frequency volume. Hence, we feel that they are close to optimal in their characteristics.

B. DT-$\mathbb{C}$WT in 3-D

The DT-$\mathbb{C}$WT is implemented in 3-D by performing separable filtering on rows, columns, and slices of the 3-D data set. This filtering process produces the structure shown in Fig. 3, which contains eight bands, i.e., LLL, HLL, LHL, HHL, LLH, HLH, LHH, and HHH, as found in a conventional 3-D DWT. However, in the dual-tree version, since there are two trees of filtering on each dimension, the DT-$\mathbb{C}$WT produces an octal-tree system, which introduces a redundancy of 2:1 on each dimension and gives a total redundancy of 8:1 in 3-D.

Each subband in the octal-tree system produces eight real coefficients (one from each tree) at each spatial location, and these can yield four directional subbands of complex coefficients by

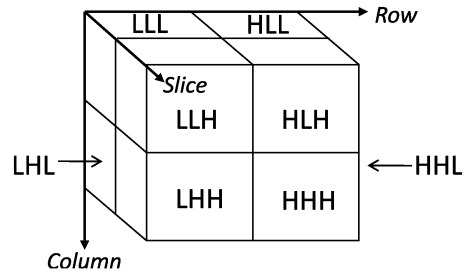


Fig. 3. Eight-band structure of 3-D DT-ℂWT, where each cube represents a subband of one of the eight DWTs in a DT-ℂWT. The filtering is performed in the sequence of rows, columns, and slices. For example, HLL means high-pass filtering on rows and low-pass filtering on columns and slices.

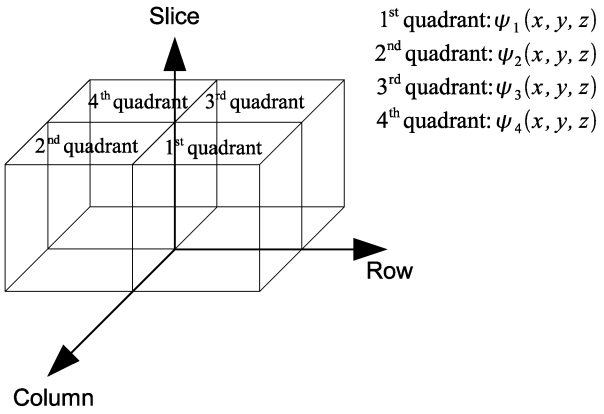


Fig. 4. Upper half-space of the frequency spectrum of 3-D data sets.

simple arithmetic sum and difference operations [22]. Because of the Hilbert pair relationships, these four subbands correspond to different quadrants of the 3-D spectral half-space, as depicted in Fig. 4. Since the LLL band is used for the next level of CWTs, there are altogether $(8-1)\times 4 = 28$ directional subbands for each DT-ℂWT level. These subbands are produced by all the bands from the eight-band stucture except for the LLL band. To see how a quad of directional subbands can be generated by the coefficients of one band of the eight-band structure, we specifically consider the HHH band at a certain level of the DT-ℂWT (other bands follow the same derivation). The 3-D wavelet for the first-quadrant HHH subband can be written as

$$\begin{aligned}\psi_1(x,y,z) &= [\psi_a(x)+j\psi_b(x)]\,[\psi_a(y)+j\psi_b(y)] \\ &\quad\times[\psi_a(z)+j\psi_b(z)] \qquad (1)\\ &= [\psi_a(x)\psi_a(y)\psi_a(z)-\psi_b(x)\psi_b(y)\psi_a(z) \\ &\quad -\psi_a(x)\psi_b(y)\psi_b(z)-\psi_b(x)\psi_a(y)\psi_b(z)] \\ &\quad + j\,[\psi_a(x)\psi_a(y)\psi_b(z)-\psi_b(x)\psi_b(y)\psi_b(z) \\ &\quad +\psi_a(x)\psi_b(y)\psi_a(z)+\psi_b(x)\psi_a(y)\psi_a(z)]. \qquad (2)\end{aligned}$$

Subscripts a and b in the above equations denote trees a (generating the real part) and b (generating the imaginary part) in Fig. 1. In (2), $\psi_a(i)$ and $\psi_b(i)$ are the high-pass filters of trees a and b being applied to dimension $i \in \{x, y, z\}$, with x, y, and z denoting the axes of rows, columns, and slices, respectively. Note that the 1-D DT-ℂWT filters suppress the negative half of

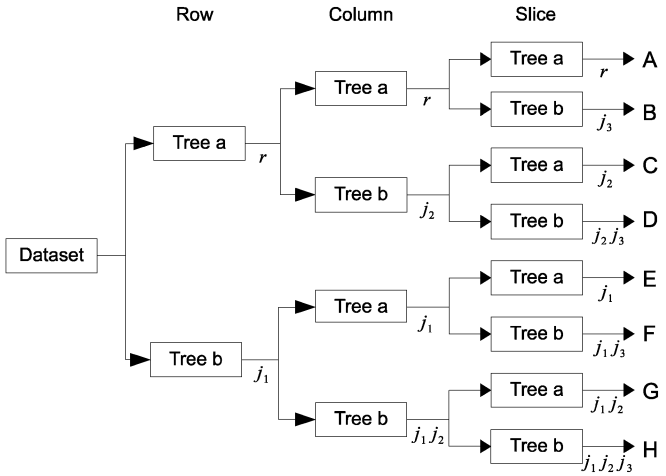


Fig. 5. For each of the row, column, and slice dimensions, high-pass or low-pass filters are applied to generate the corresponding octal-tree components. r means a real component, whereas j_1, j_2, and j_3 denote the imaginary operator j for the row, column, and slice directions, respectively. For components in the first quadrant, $j_1 = +j$, $j_2 = +j$, and $j_3 = +j$; for the second quadrant, $j_1 = -j$, $j_2 = +j$, and $j_3 = +j$; for the third quadrant, $j_1 = +j$, $j_2 = -j$, and $j_3 = +j$; for the fourth quadrant, $j_1 = -j$, $j_2 = -j$, and $j_3 = +j$.

the frequency spectrum. Therefore, the wavelet in (2) only represents the spectrum in the first quadrant of the upper half of the frequency domain. However, real 3-D data sets contain independent frequency components in the first, second, third, and fourth quadrants of the upper-half frequency space, as depicted in Fig. 4. The wavelets for the second, third, and fourth quadrants can be obtained from the following equations:

$$\begin{aligned}\psi_2(x,y,z) &= [\psi_a(x)-j\psi_b(x)]\,[\psi_a(y)+j\psi_b(y)] \\ &\quad\times[\psi_a(z)+j\psi_b(z)] \qquad (3)\\ \psi_3(x,y,z) &= [\psi_a(x)+j\psi_b(x)]\,[\psi_a(y)-j\psi_b(y)] \\ &\quad\times[\psi_a(z)+j\psi_b(z)] \qquad (4)\\ \psi_4(x,y,z) &= [\psi_a(x)-j\psi_b(x)]\,[\psi_a(y)-j\psi_b(y)] \\ &\quad\times[\psi_a(z)+j\psi_b(z)]. \qquad (5)\end{aligned}$$

It can be seen that $\psi_2(x,y,z)$, $\psi_3(x,y,z)$, and $\psi_4(x,y,z)$ can be easily obtained by arithmetic sum and difference operations on the terms of $\psi_1(x,y,z)$ in (2). If we rewrite (2) as

$$\psi_1(x,y,z) = [A-G-D-F]+j[B-H+C+E] \quad (6)$$

where symbols $A-H$ represent the components in (2) in the same order. Then $\psi_2(x,y,z)$, $\psi_3(x,y,z)$, and $\psi_4(x,y,z)$ in (3)–(5) can be rewritten as

$$\begin{aligned}\psi_2(x,y,z) &= [A-G+D+F]+j[-B+H+C+E] \quad (7)\\ \psi_3(x,y,z) &= [A+G+D-F]+j[B+H-C+E] \quad (8)\\ \psi_4(x,y,z) &= [A+G-D+F]+j[-B-H-C+E]. \quad (9)\end{aligned}$$

Fig. 5 illustrates how the components of the octal-tree system in (6) and (7)–(9) are produced. Outputs $A-H$ in Fig. 5 have the same meaning as in the equations.

C. Memory Considerations for the 3-D DT-ℂWT

The redundancy of the 3-D DT-ℂWT is 8:1 which, although modest for a shift-invariant 3-D transform, still tends to cause

heavy computation and large memory usage compared with a nonredundant transform. To solve this problem, our algorithm only keeps the LLL band of the DT-$\mathbb{C}$WT at level 1, and all other level-1 bands are discarded. In this way, the redundancy is eliminated, giving a 1:1 transform. Note that, although all high-pass bands at level 1 are ignored, this does not cause excessive loss of information because typical real-world data sets do not have very sharp edges, and the highest frequency components are largely dominated by noise. In addition, the level-1 wavelets have poorer linear phase-shift properties than at coarser levels and therefore are less useful for registration.

III. Overview of Motion Estimation

A. Description of Motion: The Affine Model

Each motion vector is in the direction of the displacement, with amplitude equal to the amount of shift. For nonrigid registration purposes, the motion vectors should be described locally. The full set of motion vectors, which contains the motion of the data set at every location, is called the motion field.

In this paper, we describe the 3-D motion field by the affine transform [25], which can model typical motions such as translation, rotation, scaling, and shear. A major advantage of using the affine model lies in the fact that, if the motions at two locations are from the same affine model (e.g., shift, rotation, scaling, or shear, typically belonging to the same object), their affine parameters should also be the same. This property is important when overcoming the problems of ill-conditioning due to limited aperture (known as the aperture problem [26]), for estimating motions of rigid bodies (see Section V-A) and smoothing the motion estimates across a region of the data set (see Section IV-E).

B. Theoretical Background of Motion Estimation

The model described here is based on the parametric model introduced by Hemmendorff [21], with significant changes in order to take full advantage of the efficient DT-$\mathbb{C}$WT front end.

1) Motion Constraint: We define the four-element homogeneous displacement vector at location $\mathbf{x}$ to be

$$\tilde{\mathbf{v}}(\mathbf{x}) = \begin{bmatrix} \mathbf{v}(\mathbf{x}) \\ 1 \end{bmatrix} \tag{10}$$

where $\mathbf{v}(\mathbf{x})$ is the motion vector at location $\mathbf{x} = [x, y, z]^T$. A motion constraint vector is a four-element vector $\mathbf{c}(\mathbf{x})$ that defines a plane in 3-D space and satisfies

$$\mathbf{c}^T(\mathbf{x})\tilde{\mathbf{v}}(\mathbf{x}) = 0. \tag{11}$$

Horn and Schunk [27] showed that the motion constraints can be estimated as the spatiotemporal gradient of the image intensity. This is known as the optical flow model. In the context of 3-D DT-$\mathbb{C}$WT, since the phase of each complex coefficient has an approximately linear relationship with the local shift vector $\mathbf{v}(\mathbf{x})$, we have the following equation:

$$\frac{\partial \theta_d}{\partial t} = \nabla_{\mathbf{x}}\theta_d \cdot \mathbf{v}(\mathbf{x}) \quad \text{which gives} \quad \begin{bmatrix} \nabla_{\mathbf{x}}\theta_d \\ -\frac{\partial \theta_d}{\partial t} \end{bmatrix}^T \tilde{\mathbf{v}}(\mathbf{x}) = 0 \tag{12}$$

where $\nabla_{\mathbf{x}}\theta_d = [(\partial\theta_d/\partial x) \quad (\partial\theta_d/\partial y) \quad (\partial\theta_d/\partial z)]^T$, representing the phase gradient at $\mathbf{x}$ for subband d in the directions of x, y, and z. Note that d predominantly indexes the different directions of subbands at a particular scale, but it may also index different subband scales too. The term $\partial\theta_d/\partial t$ is the phase gradient between the two data sets being registered, i.e., the phase change at $\mathbf{x}$ of the DT-$\mathbb{C}$WT coefficients of subband d between the two data sets. Comparing (11) and (12), it is clear that the motion constraint vector satisfies the following expression:

$$\mathbf{c}_d(\mathbf{x}) = C_d(\mathbf{x}) \begin{bmatrix} \nabla_{\mathbf{x}}\theta_d \\ -\frac{\partial \theta_d}{\partial t} \end{bmatrix} \tag{13}$$

where $C_d(\mathbf{x})$ is a scalar weighting factor, which can be designed to reflect the confidence of the motion constraint at $\mathbf{x}$ in the direction of subband d. In applications where the two data sets being registered are not identical, it is desirable to give a larger weight to locations containing similar features and a smaller weight to locations with inconsistent features between the two data sets. These inconsistent features may be caused by the need to improve the image visibility (i.e., by injection of radiocontrast agents, as shown in Fig. 9), the actual development of disease (i.e., removal of tumor, as shown in Fig. 10), or image noise. In order to limit the motion estimation outliers caused by the inconsistent features, we design the confidence measure $C_d(\mathbf{x})$ defined by the following expression, which gives the highest weight to consistent features and a lower weight to less consistent features:

$$C_d(\mathbf{x}) = \frac{\left|\sum_{k=1}^{8} u_k^* v_k\right|^2}{\sum_{k=1}^{8} \left(|u_k|^3 + |v_k|^3\right) + \varepsilon} \tag{14}$$

where u_k and v_k are the wavelet coefficients in the reference and sensed data sets, respectively, and subscripts $k = 1 \ldots 8$ denote the eight neighboring wavelet coefficients, which are on a cube centered at location $\mathbf{x}$ in subband d. The small positive constant ε prevents the denominator from going to zero if the wavelet coefficients become very small. It should be comparable with the cube of the expected amplitude of the measurement noise. Note that the numerator in (14) is proportional to the fourth power of the coefficient amplitudes, whereas the denominator is proportional to their cubes (ignoring ε); therefore, overall C_d varies linearly with amplitude, which naturally gives a greater weight to stronger features in the data set.

2) Cost Function: The affine model equation is written as

$$\mathbf{v}(\mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 & x & 0 & 0 & y & 0 & 0 & z & 0 & 0 \\ 0 & 1 & 0 & 0 & x & 0 & 0 & y & 0 & 0 & z & 0 \\ 0 & 0 & 1 & 0 & 0 & x & 0 & 0 & y & 0 & 0 & z \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_{12} \end{bmatrix} = \mathbf{K}(\mathbf{x})\mathbf{a}. \tag{15}$$

Defining $\tilde{\mathbf{K}}(\mathbf{x}) = \begin{bmatrix} \mathbf{K}(\mathbf{x}) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$ and $\tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix}$, and then using (10), the homogeneous motion vector is given by

$$\tilde{\mathbf{v}}(\mathbf{x}) = \tilde{\mathbf{K}}(\mathbf{x})\tilde{\mathbf{a}}. \tag{16}$$

Combining (11) and (16) for all 28 subband directions, we have

$$\mathbf{c}_d(\mathbf{x})^T \tilde{\mathbf{K}}(\mathbf{x})\tilde{\mathbf{a}} = 0 \quad \text{for} \quad d = 1 \ldots 28. \tag{17}$$

Thus, for each location $\mathbf{x}$, there are 28 constraint equations, which is an overdetermined set for the 12 unknown affine parameters in $\tilde{\mathbf{a}}$. Hence, we find the value of $\tilde{\mathbf{a}}$, which minimizes the squared error $\epsilon(\mathbf{x})$. $\epsilon(\mathbf{x})$ is the cost function of our algorithm, which is given by

$$\begin{aligned}\epsilon(\mathbf{x}) &= \sum_{d=1}^{28} \left\| \mathbf{c}_d^T(\mathbf{x}) \tilde{\mathbf{K}}(\mathbf{x}) \tilde{\mathbf{a}} \right\|^2 \\ &= \sum_{d=1}^{28} \tilde{\mathbf{a}}^T \tilde{\mathbf{K}}^T(\mathbf{x}) \mathbf{c}_d(\mathbf{x}) \mathbf{c}_d^T(\mathbf{x}) \tilde{\mathbf{K}}(\mathbf{x}) \tilde{\mathbf{a}} \\ &= \tilde{\mathbf{a}}^T \tilde{\mathbf{Q}}(\mathbf{x}) \tilde{\mathbf{a}} \end{aligned} \tag{18}$$

where

$$\tilde{\mathbf{Q}}(\mathbf{x}) = \sum_{d=1}^{28} \tilde{\mathbf{K}}^T(\mathbf{x}) \mathbf{c}_d(\mathbf{x}) \mathbf{c}_d^T(\mathbf{x}) \tilde{\mathbf{K}}(\mathbf{x}). \tag{19}$$

In practice, in order to handle the registration of dissimilar image features as well as dealing with the aperture problem, it is often helpful to combine the $\tilde{\mathbf{Q}}(\mathbf{x})$ matrices across more than one level of the DT-$\mathbb{C}$WT and over a slightly wider area within each level to produce the most accurate estimate of the affine parameters. We therefore define locality $\boldsymbol{\chi}$ to represent this wider spatial and interscale region, such that

$$\tilde{\mathbf{Q}}_{\boldsymbol{\chi}} = \sum_{\mathbf{x} \in \boldsymbol{\chi}} \tilde{\mathbf{Q}}(\mathbf{x}). \tag{20}$$

The $\tilde{\mathbf{Q}}$ matrices are symmetric; therefore, $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$ can be written in the following form:

$$\tilde{\mathbf{Q}}_{\boldsymbol{\chi}} = \begin{bmatrix} \mathbf{Q}_{\boldsymbol{\chi}} & \mathbf{q}_{\boldsymbol{\chi}} \\ \mathbf{q}_{\boldsymbol{\chi}}^T & q_{0,\boldsymbol{\chi}} \end{bmatrix} \tag{21}$$

where $\mathbf{q}_{\boldsymbol{\chi}}$ is a 12-element vector and $q_{0,\boldsymbol{\chi}}$ is a scalar.

Substituting (21) into (18) and (20), the cost function is thus expressed as

$$\begin{aligned}\epsilon_{\boldsymbol{\chi}} &= \sum_{\mathbf{x} \in \boldsymbol{\chi}} \epsilon(\mathbf{x}) = \tilde{\mathbf{a}}^T \tilde{\mathbf{Q}}_{\boldsymbol{\chi}} \tilde{\mathbf{a}} \\ &= \begin{bmatrix} \mathbf{a}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{\boldsymbol{\chi}} & \mathbf{q}_{\boldsymbol{\chi}} \\ \mathbf{q}_{\boldsymbol{\chi}}^T & q_{0,\boldsymbol{\chi}} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix} \\ &= \mathbf{a}^T \mathbf{Q}_{\boldsymbol{\chi}} \mathbf{a} + 2\mathbf{a}^T \mathbf{q}_{\boldsymbol{\chi}} + q_{0,\boldsymbol{\chi}} \end{aligned} \tag{22}$$

To minimize $\epsilon_{\boldsymbol{\chi}}$, we differentiate the expression of $\epsilon_{\boldsymbol{\chi}}$ with respect to $\mathbf{a}$ and set the derivative to zero. Hence

$$\nabla_{\mathbf{a}} \epsilon_{\boldsymbol{\chi}} = 2\mathbf{Q}_{\boldsymbol{\chi}} \mathbf{a} + 2\mathbf{q}_{\boldsymbol{\chi}} = 0 \tag{23}$$

and the local affine parameter vector, which gives the least squared error, is

$$\mathbf{a}_{\boldsymbol{\chi}} = -\mathbf{Q}_{\boldsymbol{\chi}}^{-1} \mathbf{q}_{\boldsymbol{\chi}}. \tag{24}$$

Once the affine parameters have been obtained, the corresponding motion is obtained by substituting the elements of $\mathbf{a}_{\boldsymbol{\chi}}$ into (15).

IV. Image Registration Algorithm

There are three major steps in our registration algorithm: 1) transform the data to the DT-$\mathbb{C}$WT domain; 2) perform motion estimation; and 3) register the sensed data set to the reference data set using the estimated motion. A flowchart illustrating the algorithm is shown in Fig. 6. The pair of data sets to be registered are first transformed by the DT-$\mathbb{C}$WT, and the levels that will be used for motion estimation are selected. The subband coefficients of the sensed data set are shifted according to the motion field produced by previous iterations. The shifted coefficients of the sensed data set, together with the coefficients of reference data set, are used to generate motion constraints. The $\tilde{\mathbf{Q}}(\mathbf{x})$ matrix can then be calculated at each location $\mathbf{x}$ and smoothed across each locality $\boldsymbol{\chi}$ to give $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$. Next, the affine parameters, which minimize the least squared error, are obtained and added to the affine parameter estimates from previous iterations to generate the motion field. After all the iterations, the sensed data set is finally interpolated in the 3-D spatial domain to register to the reference data set. The main blocks within the algorithm will now be described.

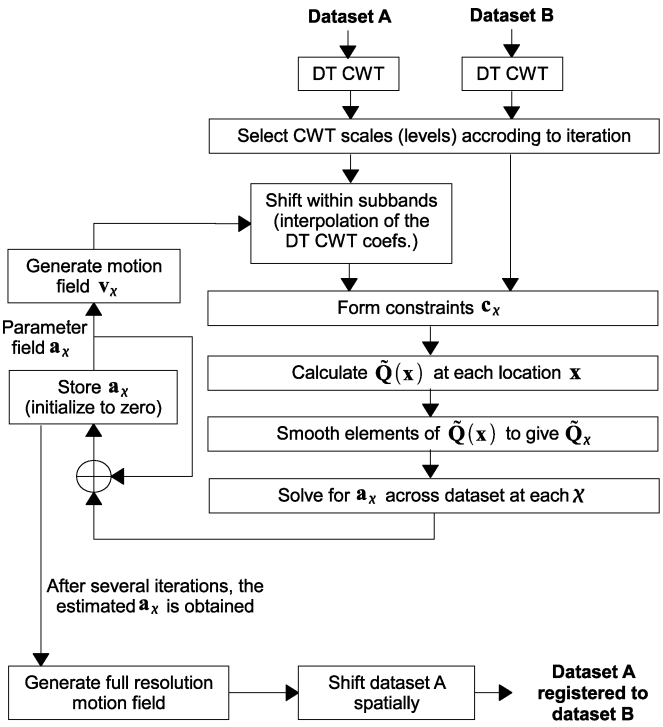


Fig. 6. Flowchart of the image registration algorithm.

A. DT-$\mathbb{C}$WT on Input Data Sets

The algorithm takes the 3-D data sets A and B as inputs, where B is the reference data set and A is the sensed data set. The two data sets are forward-transformed with the 3-D DT-$\mathbb{C}$WT, and their complex wavelet coefficients at each level (except level 1, see Section II-C) are obtained.

B. Select CWT Scales (Levels) According to Iteration

The multiresolution nature of the DT-$\mathbb{C}$WT means that we can estimate the motion from different scales. For example, if we choose the complex wavelet coefficients at level 3 to perform motion estimation, then each coefficient corresponds to a block of 8^3 voxels in the original data set. This means that one affine vector will be estimated for each of these 8^3 blocks.

It is important to realize that a wrong motion vector is likely to be produced if the phase change of a given complex wavelet coefficient exceeds the range $-\pi$ to $+\pi$ as this causes ambiguity for deciding the motion. For a given amount of motion, the phase change of coarser level complex coefficients will be smaller than that of finer level coefficients by a factor of 2 per scale level [28]. This shows that, by using coarser level coefficients, the estimation algorithm can cope with larger motion, whereas with finer level coefficients, it can only tackle small motion. However, performing motion estimation from coarser levels only may lead to inaccurate local motion estimates since the selected scale may be too coarse to capture all the important data features and the real local motion. This reveals a tradeoff for selecting the levels at which the motion estimation should be performed: coarser level coefficients can estimate large motion but with limited accuracy, whereas finer level coefficients can give good local motion accuracy but cannot reliably estimate large motion vectors.

In order to achieve an accurate motion estimation over a wide range, we adopt a coarse-to-fine approach to estimate the motion iteratively. In the first few iterations, the motion field should be estimated from coarse-level coefficients only (to handle large motion), and this motion field is then used to shift the sensed data set toward the reference. Once the large motion between the two data sets has been compensated, later iterations can focus on estimating the small local residual motion errors by using finer level coefficients. In this way, the motion estimate is gradually refined to approach the true motion field.

C. Shift Within Subbands: Interpolation of the DT-$\mathbb{C}$WT Coefficients

The feasibility of interpolating DT-$\mathbb{C}$WT coefficients within each subband separately relies on the transform's shift-invariant property, as introduced in Section II-A. The DT-$\mathbb{C}$WT coefficients of data set A need to be shifted and interpolated using estimated motion from previous iterations. After interpolation, the complex coefficients of data set A should look more similar to those of data set B because the amount of motion between A and B has been reduced. Finer level coefficients may then be used to perform motion estimation in the subsequent iterations.

One may argue that data set A could be shifted with the estimated motion in the spatial domain and then transformed with the DT-$\mathbb{C}$WT to get the wavelet coefficients for the shifted data set. This method is feasible but tends to be slow since shifting a 3-D data set is computationally demanding. The advantage of shifting in the complex wavelet domain is that it provides a fast and smooth way of aligning the data sets as the number of DT-$\mathbb{C}$WT coefficients at any level above 1 is much smaller than the sample size of the original data set, and the coefficients are well bandlimited. Moreover, the computations of performing the DT-$\mathbb{C}$WT are avoided within the iterative loop.

We must be aware that complex coefficients in each subband are bandpass signals and should not be interpolated in the normal way for low-pass signals. The DT-$\mathbb{C}$WT filters introduce a different phase offset rate to each of the subbands; therefore, direct interpolation will not produce the correct result. Our method for interpolating complex coefficients is as follows:

1) derotate the phase of the bandpass complex coefficients to compensate for the phase offset rate, and center the subband on zero frequency;
2) interpolate the real and imaginary parts of the derotated coefficients, using conventional (tri-)linear or cubic methods;
3) rerotate the phase of the complex coefficients to restore their phase offset rate and to correct their bandpass center frequency.

The details of subband interpolation are discussed in Appendix A.

D. Form Constraints and Calculate $\tilde{\mathbf{Q}}(\mathbf{x})$ at Each Location $\mathbf{x}$

The phase of the DT-$\mathbb{C}$WT coefficients can be used to form the constraints, as described by (13). With the constraint vectors, we can then obtain the $\tilde{\mathbf{Q}}(\mathbf{x})$ matrix at each location $\mathbf{x}$ and at each chosen scale by using (19).

At each wavelet scale, locations $\mathbf{x}$ are chosen to be at the centers of cubes bounded by eight adjacent coefficients [i.e., the u_k and v_k of (14)]. The derivatives of the column vector in (13) are then calculated by forming the two cubes of complex coefficients $\mathbf{u}$ and $\mathbf{v}$ into corners of a 4-D hypercube and by taking conjugate products across each pair of hyperfaces of the hypercube in turn. The eight conjugate products from each hyperface pair are then summed (similar to the summation in the numerator of (14), which is used for the $\partial\theta/\partial t$ term), and the phase of the complex resultant is the required phase derivative. Thus, the derivatives are the average phase shifts at the center of each hypercube with respect to x, y, z, and t, where the averages are effectively weighted by the magnitudes of the conjugate product terms so that large coefficient pairs contribute more to the phase "average" than smaller pairs.

E. Smooth Elements of $\tilde{\mathbf{Q}}(\mathbf{x})$ to Give $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$

Recall that each motion constraint vector from (13) has the weighting factor $C_d(\mathbf{x})$, which reflects our confidence in the constraint. Equation (19) shows that a large weight motion constraint $\mathbf{c}_d(\mathbf{x})$ will contribute more to the elements of $\tilde{\mathbf{Q}}(\mathbf{x})$ than a smaller one. Similar in (20), larger $\tilde{\mathbf{Q}}(\mathbf{x})$ matrices will contribute more to $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$ than smaller ones for each locality $\boldsymbol{\chi}$.

However, it can be seen from (21) and (24) that, when $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$ is used to solve for the affine vector $\mathbf{a}_{\boldsymbol{\chi}}$, the solution will be independent of the total weight of $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$. In this way, the weighting factors assigned to the motion constraints affect their relative contributions to the affine motion solution, but the overall weight does not affect the final result. In smooth data regions or where there are no consistent features between data sets A and B, some or all of the eigenvalues of $\mathbf{Q}_{\boldsymbol{\chi}}$ will be small, and the resulting affine parameters will not be reliable.

A solution to this problem is to smooth the elements of the $\tilde{\mathbf{Q}}(\mathbf{x})$ matrices spatially across the data set. The basic idea is to bring the motion information from locations with larger weight constraint vectors to those with smaller weight. This can be achieved by expanding each locality $\boldsymbol{\chi}$ to include regions that overlap with other adjacent localities and by using a smoothly

decaying weighting $w(\mathbf{x}-\boldsymbol{\chi}_0)$ as one moves away from $\boldsymbol{\chi}_0$, i.e., the center of $\boldsymbol{\chi}$. Hence, (20) becomes

$$\tilde{\mathbf{Q}}_{\boldsymbol{\chi}} = \sum_{\mathbf{x}\in\boldsymbol{\chi}} w(\mathbf{x}-\boldsymbol{\chi}_0)\tilde{\mathbf{Q}}(\mathbf{x}) \tag{25}$$

and the localities $\boldsymbol{\chi}$ are now larger and overlapping. This effectively applies a spatial low-pass smoothing filter, which is defined by the $w(\mathbf{x}-\boldsymbol{\chi}_0)$, to the $\tilde{\mathbf{Q}}(\mathbf{x})$ matrices. Typically, there are many localities $\boldsymbol{\chi}$ covering the whole data-set volume.

We find that a simple triangular low-pass filter gives good performance. The choice of the smoothing filter size (i.e., number of taps) is application dependent. A large filter should be used if the motion of the object is believed to be relatively smooth, i.e., the 12-element affine vectors are likely to be similar at nearby localities. However, if the motion varies in a significantly non-affine way across the data set, then a small smoothing filter will be more desirable. It would be relatively easy to make the filter adaptive to the data, but we have not investigated this.

The filtering in (25) also allows an easy combination of the $\tilde{\mathbf{Q}}(\mathbf{x})$ matrices across the scale, where the basic sampling intervals of $\mathbf{x}$ are different. We use simple triangular filters to upsample or downsample the matrices from different scales; therefore, they are all sampled at the chosen grid for localities $\boldsymbol{\chi}$. Typically, the grid for $\boldsymbol{\chi}$ corresponds to that for the level-3 or level-4 wavelet coefficients.

F. Solve for $\mathbf{a}_\chi$ and Generate the Motion Field

The algorithm estimates the motion iteratively. Every iteration produces a set of affine vectors from (24), and they are added to $\mathbf{a}_\chi$ estimated from previous iterations. As described in Section IV-C, this motion field is used to shift the DT-$\mathbb{C}$WT coefficients of the sensed data set, and the shifted coefficients are used for the next iteration of motion estimation.

The 3-D motion field is computed at the resolution and grid points of each wavelet scale that is being used by linearly interpolating affine vectors from the nearest locality centers $\boldsymbol{\chi}_0$ and then by using the affine expression of (15) at each grid point. The multiscale motion fields at the selected scales are finally passed to the block "shift within subbands" to complete the iterative loop.

G. Registration by Spatial Transform

After several iterations, a set of affine parameters are obtained, which accurately describe the motion between the sensed and reference data sets. The affine parameters are then upsampled to produce the motion field at the full resolution, i.e., each voxel of the data set has a corresponding motion vector. With this motion field, the sensed data set can then be registered by interpolation in the 3-D spatial domain to align it with the reference data set. An alternative method of registration would be to motion-compensate every subband in the wavelet domain and then to inverse-transform the result, but this tends to introduce slight artifacts due to the transform being only approximately shift invariant. Furthermore, it does not save any computation. Hence, we recommend the use of spatial-domain interpolation to perform the final registration step.

V. Details of the Registration Algorithm

This section includes in-depth discussions on some of the details of our algorithm.

A. Rigid-Body Registration

The difference between rigid-body registration and nonrigid-body registration is that the motion of a rigid body always conforms to a single affine model (i.e., $\mathbf{a}_{\boldsymbol{\chi}}$ must be the same at every locality of the data set), whereas the motion of a nonrigid body is locally characterized. Thus, for rigid body registration, the $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$ matrices at each locality should be the same. This implies that, when estimating rigid motion, the $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$ matrices at each locality can simply be averaged to produce global $\tilde{\mathbf{Q}}$, which is denoted by $\tilde{\mathbf{Q}}_{\text{mean}}$. Then, $\tilde{\mathbf{Q}}_{\text{mean}}$ replaces the original local $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$ matrix at all localities before solving for the single affine vector $\mathbf{a}_{\text{mean}}$.

Even for nonrigid-body registration, it can often be best to perform the first few iterations using the rigid-body registration mode before carrying out the nonrigid-body motion estimation. This means that we start by registering the two data sets as a whole using coarse wavelet scales only to compensate for possibly large motion and then to bring in local motion information on later iterations by switching to the nonrigid-body mode and finer wavelet scales. Using the rigid-body mode additionally helps to reduce the chance of $\mathbf{Q}$ from becoming ill-conditioned since it adopts the largest possible aperture (the whole image) for the purely coarse-level iterations, at which ill conditioning is most likely because of low spatial detail.

B. Selecting DT-$\mathbb{C}$WT Levels for Iterations

For every iteration, one or several levels of DT-$\mathbb{C}$WT can be selected to carry out motion estimation. The levels should be chosen in the sequence of coarse to fine, in order to estimate the large components of motion first and then gradually refine the estimates. A typical choice of DT-$\mathbb{C}$WT levels ranges from level 5 or levels 5 and 4 in early iterations to levels 4, 3, and 2 in the final few iterations, which take much longer to compute. There is a big computational gain from minimizing the number of iterations, on which level 2 is used as the number of constraint equations increases by a factor of 8, as each finer level is introduced.

It should be noted that the DT-$\mathbb{C}$WT in its simplest form requires the size of the data set to be a multiple of 2^K in each dimension, where K is the coarsest transform level used. However, the DT-$\mathbb{C}$WT can be modified slightly to break the restriction on size by appending symmetrically extended (mirrored) coefficients as necessary at opposite edges of the LLL band after each level of transform so as to ensure that the size of the LLL band is always divisible by 8 in each dimension before it is passed to the next coarser level. The modified DT-$\mathbb{C}$WT only requires that the initial data sets be a multiple of 4 along each dimension. Some care must be taken, however, in the programming of the motion estimation parts of the algorithm to allow for the changes in subband sizes that this modification produces.

C. Motion Estimation at Data-Set Boundaries

In the DT-$\mathbb{C}$WT, to avoid boundary discontinuities, the data set is symmetrically extended along each of the dimensions before being filtered by the high-pass or low-pass filters. Although

symmetric extension is a good technique for reducing edge effects, it leads to inaccuracy in estimating motion normal to the boundary of the data set. This is because motion near the data set boundary will cause reflected motion in the symmetrically extended region. The components of these two motions, which are normal to the boundary, move in opposite directions to each other and thus tend to result in poor motion estimation near boundaries.

Unfortunately, it is not straightforward to estimate accurately the motion at data-set boundaries. To reduce the estimation error, we recommend using motion information at inner localities to infer motion at the boundary localities. This scheme can simply be achieved by zeroing out some or all of the motion constraint vectors at the boundary localities when the constraints are generated. This is equivalent to saying that there is no information about certain components of motion at the boundary localities. If all motion constraints near the data-set boundaries are set to zero, then $\tilde{\mathbf{Q}}(\mathbf{x})$ will also be zero at these boundary localities. When the $\tilde{\mathbf{Q}}(\mathbf{x})$ matrices are smoothed across the localities, as described in Section IV-E, the information of the $\tilde{\mathbf{Q}}(\mathbf{x})$ matrices at neighboring inner localities will leak into the $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$ near the boundaries. In this way, the affine motion models at the boundaries will be extrapolated from the affine models of the inner regions.

D. Modification to the $\mathbf{Q}_{\boldsymbol{\chi}}$ Matrix

We have shown in (24) that the optimal affine parameter vector solution is $\mathbf{a}_\chi = -\mathbf{Q}_{\boldsymbol{\chi}}^{-1}\mathbf{q}_{\boldsymbol{\chi}}$. However, in practice, $\mathbf{Q}_{\boldsymbol{\chi}}$ at some localities may be found to be near singular (very ill-conditioned). This is undesirable because small numerical errors in $\mathbf{Q}_{\boldsymbol{\chi}}$ will lead to large inaccuracies in the solution of $\mathbf{a}_\chi$. The ill-conditioning of $\mathbf{Q}_{\boldsymbol{\chi}}$ implies that the 12-parameter affine model is overfitting the local motion. The source of the ill-conditioning is often due to the local data content being relatively simple; therefore, the local motion can be described by fewer than 12 parameters. To remedy this problem (often known as the aperture problem), one possible solution is to use a large smoothing filter when the $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$ matrices are calculated. However, applying a large filter may result inaccurate local motion estimation. A solution, which we have used successfully with relatively rigid brain scans, is to modify $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$ slightly to become $\tilde{\mathbf{Q}}'_{\boldsymbol{\chi}}$, where

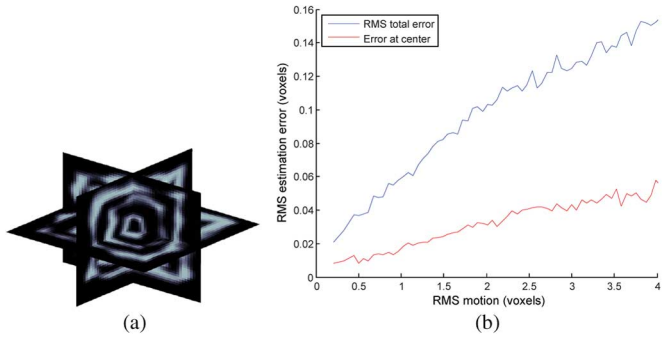
$$\tilde{\mathbf{Q}}'_{\boldsymbol{\chi}} = \tilde{\mathbf{Q}}_{\boldsymbol{\chi}} + \lambda\tilde{\mathbf{Q}}_{\text{mean}}. \tag{26}$$

Recall that $\tilde{\mathbf{Q}}_{\text{mean}}$ is the mean of $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$ over all localities, i.e., $\tilde{\mathbf{Q}}_{\text{mean}}$ represents the estimate of the global motion. λ in (26) should be a small constant, which we will discuss shortly. With the definition of $\tilde{\mathbf{Q}}'_{\boldsymbol{\chi}}$ in (26), the solution of the affine parameters is now expressed as follows:

$$\mathbf{a}_{\boldsymbol{\chi}} = -\tilde{\mathbf{Q}}'^{-1}_{\boldsymbol{\chi}}\mathbf{q}'_{\boldsymbol{\chi}} \tag{27}$$

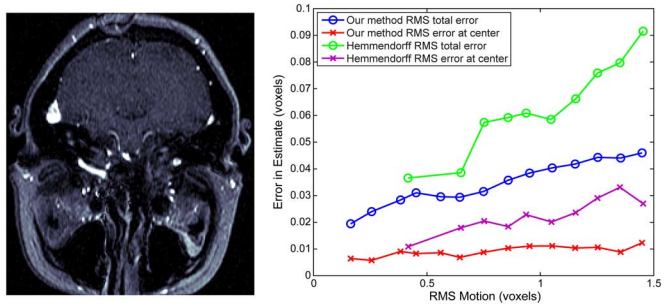
where $\mathbf{Q}'_{\boldsymbol{\chi}}$ and $\mathbf{q}'_{\boldsymbol{\chi}}$ come from the newly defined $\tilde{\mathbf{Q}}'_{\boldsymbol{\chi}}$ as follows:

$$\tilde{\mathbf{Q}}'_{\boldsymbol{\chi}} = \begin{bmatrix} \mathbf{Q}'_{\boldsymbol{\chi}} & \mathbf{q}'_{\boldsymbol{\chi}} \\ \mathbf{q}'^{T}_{\boldsymbol{\chi}} & q'_{0\boldsymbol{\chi}} \end{bmatrix} \tag{28}$$



(a) (b)

Fig. 7. (a) Intersection view of the 3-D shell pattern (64 × 64 × 64). (b) RMS motion estimation error versus the RMS value of the true motion.



(a) (b)

Fig. 8. (a) MRI data set (140 × 140 × 68, 2-mm slice sampling spacing). (b) RMS motion estimation error versus the RMS value of the true motion. Our RMS error curves are generated using 250 experimental results, whereas Hemmendorff's error curves are generated using the 124 data points presented in [21, Fig. 7].

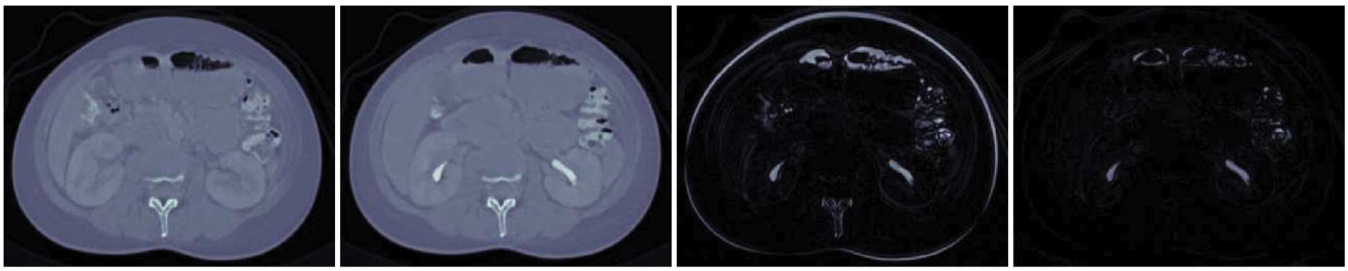
The modified $\tilde{\mathbf{Q}}'_{\boldsymbol{\chi}}$ can be regarded as the superposition of the local motion and a small fraction of global motion. This is equivalent to driving the local motion estimate slightly toward the global motion, and the amount is controlled by λ. The value of λ is not critical, and we express it as

$$\lambda = \sqrt{\frac{0.001.\sum_{\text{all}\boldsymbol{\chi}} \text{Energy of } \tilde{\mathbf{Q}}_{\boldsymbol{\chi}}}{(\#\text{ of localities})(\text{Energy of } \tilde{\mathbf{Q}}_{\text{mean}})}} \tag{29}$$

where the energy of a matrix is defined as the sum of all its elements squared. It is best to consider the meaning of $\lambda\tilde{\mathbf{Q}}_{\text{mean}}$ in order to interpret λ in the above equation. With the definition of λ in (29), the energy of $\lambda\tilde{\mathbf{Q}}_{\text{mean}}$ is just 0.001 of the mean energy of $\tilde{\mathbf{Q}}_{\boldsymbol{\chi}}$.

With the small amount of added global motion, $\tilde{\mathbf{Q}}'_{\boldsymbol{\chi}}$ no longer tends to be singular. This technique also tends to give a more regularized motion field, particularly in regions where the motion is ill-defined.

For example, consider typical medical images with human organs in the center being surrounded by dark backgrounds. The featureless background is usually not perfectly dark, and some noise exists in these regions. Our motion estimation gives a small weighting factor to the dark backgrounds, but at the localities far away from the features, it will still try to register the noise because motion from the feature-rich regions is not smoothed enough to affect these far-away localities. However, with the modified $\tilde{\mathbf{Q}}'_{\boldsymbol{\chi}}$, a small amount of global motion is added to every locality; therefore, it will tend to suppress noisy motion estimates in the background localities and make them look like



(a) (b) (c) (d)

Fig. 9. Registration of 3-D CT scans (384 × 512 × 128) with contrast agents where the venous phase data set is being registered to the delay phase data set. The registration is performed in 3-D, but only a single slice is displayed here for the convenience of visualization. (a) Venous phase. (b) Delay phase. (c) Difference between data sets before registration. (d) Difference between data sets after registration.

the global motion. Note that, although the motion estimates at the featureless background are not critically important, it is best to stop them from being irregular and introducing artifacts. This can have the added advantage of speeding up the 3-D spatial interpolation, which follows the motion estimation.

VI. Experimental Results

In this section, we show that our algorithm gives highly accurate estimation of motion on artificial and real-world data sets. The medical data sets in Sections VI-B and VI-C, together with the Matlab scripts for visualizing data sets, are available online.[1]

A. Registration of Synthetically Shifted Data Sets

An artificial 3-D shell pattern is generated, as shown in Fig. 7(a). It is produced by stacking several cubical shells together and warping them with a nonrigid motion field. The shape of the cubical shell is designed like this for two reasons: 1) Its asymmetric nature avoids ambiguity when analyzing the rotation components; and 2) the rather irregular spatial features excite all the DT-$\mathbb{C}$WT subbands; therefore, we can test the motion estimates from all subbands.

The synthetic 3-D shell pattern is warped by a single random affine motion. The motion between the original and the warped data sets is estimated and compared with the true motion. We have used translation, rotation, scaling, and shear components to produce true affine motion. The affine parameters are randomly generated from zero-mean Gaussian distributions (except for the scaling component distribution, whose mean is set to one). In order for all types of motion components to contribute almost equally to the combined motion, the rotation, scaling, and shear components are scaled such that their motion at a radius of $n/4$ (where n is the linear size of the n^3 data set) from the center is approximately equal to the translation motion.

To evaluate the accuracy of our registration algorithm, we follow the evaluation method, as described in [21], where the estimated motion RMS error is plotted against the ground-truth motion RMS value. To get a statistical measure of the registration accuracy, the registration results are averaged over a large number of experiments. The error curve plotted in Fig. 7(b) shows that our algorithm achieves very high accuracy.

We have also performed experiments to evaluate the algorithm performance on a real MRI data set of the human brain, as shown in Fig. 8(a). The testing conditions are the same as those of the synthetic 3-D shell pattern, and the results are shown in Fig. 8(b). Compared with Hemmendorff's method [21] on the same data set, our method achieves better accuracy and lower filter computational complexity. As shown in Appendix B, the DT-$\mathbb{C}$WT requires 75 operations per voxel compared with Hemmendorff's method, which requires 310 operations per voxel.

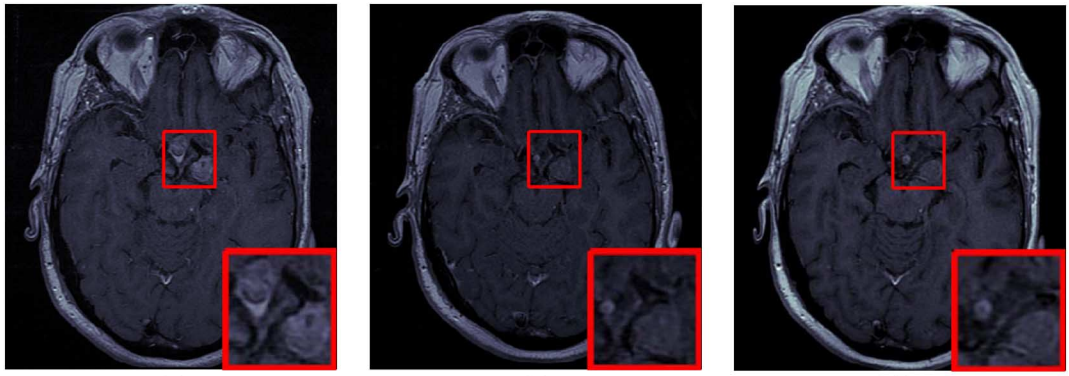
B. Registration of 3-D CT Scans With Contrast Agents

In medical radiology, contrast-agent injection techniques are commonly used to improve the visibility of internal body structures. Two 3-D abdominal CT scans [see Fig. 9(a) and (b)] are acquired during the venous phase and the delay phase of kidney contrast-agent injection. There is an interval of a few minutes between the acquisition of the two scans, and motion has occurred mainly due to patient movement and breathing. To show the effect of image registration, subtraction is performed on the data sets before and after registration. Before registration, the motion artifacts can be easily visualized in Fig. 9(c). We then apply our algorithm to align the venous-phase data set with the delay-phase data set, and the difference after registration is shown in Fig. 9(d). By comparing Fig. 9(c) and (d), it can be seen that the effects of motion at the outer boundary and the backbone regions are largely eliminated. On the other hand, the contrast agent in the kidneys as well as the gases in the stomach and the intestines are shown clearly. These observations are expected since the differences in these regions are not related to motion.

C. Registration of MRI Data sets

Fig. 10 shows the registration of paired MRI scans taken before and after the operation on a brain pituitary tumor. The middle part of the tumor was debulked during the operation, but the right part was left alone. In this scenario, the objects in the two 3-D MRI data sets are not the same since the tumor region has changed. Recall that we use weighted motion constraints to address the problem of registering nonidentical objects, as described in Section III-B1. The weighting factor of the motion constraint is small when the image features are dissimilar in the two data sets, which occurs in the middle tumor region where the features cannot be well matched. In Fig. 10, it can be seen that, apart from the middle tumor (in the red boxes), the other parts of the postoperative data set are well aligned with the preoperative data set.

[1]http://mars3.stanford.edu/hchen/imagereg.zip



(a) (b) (c)

Fig. 10. Registration of 3-D MRI scans (512 × 512 × 128) before and after operation. The tumor region in each MRI image is highlighted by the red boxes. (a) Preoperative data set (slice 92). (b) Postoperative data set (slice 99). (c) Postoperative-registered-to-preoperative data set (slice 92).

VII. Conclusion

We have shown how to perform accurate 3-D registration using the phase information of the DT-$\mathbb{C}$WT. Our algorithm adopts an efficient iterative coarse-to-fine approach, which estimates large motion first and then refines the motion field. It relies on shift-invariance and good directional filtering properties, which are key features of the DT-$\mathbb{C}$WT. Nonrigid motion is well modeled by a locally affine parametric model, whose parameters are obtained by minimizing the squared errors of the model. The weighting factors of the motion constraints are designed to reduce the perturbations of the motion estimates due to inconsistent features and noise. From the final estimated motion field, the sensed data set can be accurately registered to the reference data set by spatial-domain interpolation.

Appendix A
Interpolation of DT-$\mathbb{C}$WT Coefficients

The DT-$\mathbb{C}$WT coefficients cannot be directly interpolated since the transform produces bandpass filters, which introduce a relatively high phase rotation rate to the coefficients [1]. In order to center the passband of the DT-$\mathbb{C}$WT subband outputs on zero frequency, we set up the expected rotation rate for each subband to correspond to a frequency offset of $-1/4$ and $-3/4$ times the subband output sampling rate, which are the expected rotation rates for the low- and high-pass filters, respectively.

Note that $-1/4$ of the sampling frequency gives a phase increment of $-\pi/2$ between samples. Nevertheless, in practice, we reduce this value a little to $-\pi/2.1$, to model better the slight asymmetry of the scaling function and wavelet frequency responses. For ease of demonstration, denote $\omega_0 = -\pi/2.1$ and $\omega_1 = -3\pi/2.1$. The expected phase rotation rate for each of the 28 subbands can simply be calculated from the subband's corresponding filtering process. The rule is that high pass H, low pass L, conjugate high pass H*, and conjugate low pass L* correspond to phase rotations of ω_1, ω_0, $-\omega_1$, and $-\omega_0$, respectively. For example, for the H*LL band (i.e., remembering that this means conjugate high-pass filtering on rows and low-pass filtering on columns and slices), the expected phase rotation is $-\omega_1$, ω_0, and ω_0 in the x-, y-, and z-directions, respectively.

For each selected subband, when the expected phase rotation rate has been obtained, the DT-$\mathbb{C}$WT coefficients are derotated by multiplying by the following phase rotation terms:

$$\exp[-j\omega_x k_x - j\omega_y k_y - j\omega_z k_z]$$

where k_x, k_y, and k_z are the linear indices of the coefficients along the x-, y-, and z-directions and ω_x, ω_y, and ω_z are the expected phase rotation rates in these directions introduced by the H or L filtering processes of the subband.

Interpolation of the derotated subband coefficients can now be performed to shift the subband. Finally, the interpolated coefficients must be rotated back by multiplying them by

$$\exp\left[j\omega_x(k_x + v_x) + j\omega_y(k_y + v_y) + j\omega_z(k_z + v_z)\right]$$

where v_x, v_y, and v_z are the motion components in the x-, y-, and z-directions, respectively.

Appendix B
Algorithm Complexity

Assume that the input 3-D data set is of size N^3. Performing a 1-D filtering on this data set with a filter of length h requires N^3h operations, where 1 operation = 1 addition + 1 multiplication. For DT-$\mathbb{C}$WT filtering, we use $h = 9$-tap filter for level 1 and $m = 14$-tap filter for higher levels, as presented in [1]. Since we only keep the LLL band in the level-1 decomposition (as described in Section II-C), performing level-1 DT-$\mathbb{C}$WT needs $3N^3h$ operations on three dimensions. Level 2 requires $3N^3m$ operations, and DT-$\mathbb{C}$WT in each higher level requires 1/8 of the operations of the previous level since the filter output from the previous level is downsampled by 2 in each dimension. Consequently

$$\begin{aligned}\text{DT-}\mathbb{C}\text{WT complexity} &= 3N^3h + 3N^3m\left(1 + \frac{1}{8} + \frac{1}{64} + \cdots\right)\\ &= 75N^3 \text{ operations.}\end{aligned}$$

We perform similar analysis on the filter bank used in Hemmendorff's algorithm. Referring to [21, Fig. 4], each nine-tap real-valued low-pass filter requires $9N^3$ operations, whereas each nine-tap complex-valued quadrature filter requires $18N^3$

operations since it needs $9N^3$ for the real part and $9N^3$ for the imaginary part. Therefore, the complexity for level-1 decomposition, as illustrated in [21, Fig. 4] is $(9 \times 12 + 18 \times 9)N^3 = 270N^3$. To create a multiscale tree, an extra low-pass filter is needed in the final step of the level-1 decomposition to generate an LLL band. The LLL band can then be downsampled by 2 in each dimension; therefore, the next level of filtering requires 1/8 of the operations of the previous level. Hence

$$[21]\ \text{complexity} = \left[270+(9+270) \times \left(\frac{1}{8}+\frac{1}{64}+\cdots\right)\right] N^3 \simeq 310N^3 \text{ operations.}$$

REFERENCES

[1] N. G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *J. Appl. Comput. Harmon. Anal.*, vol. 10, no. 3, pp. 234–253, May 2001.

[2] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.

[3] H. Li, B. S. Manjunath, and S. K. Mitra, "A contour-based approach to multisensor image registration," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 320–334, Mar. 1995.

[4] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Unified real-time tracking and recognition with rotation-invariant fast features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 934–941.

[5] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, Jan. 2010.

[6] D. Ta, W. Chen, N. Gelfand, and K. Pulli, "SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2937–2944.

[7] R. Berthilsson, "Affine correlation," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 1998, vol. 2, pp. 1458–1460.

[8] A. Simper, "Correcting general band-to-band misregistrations," in *Proc. Int. Conf. Image Process.*, Sep. 1996, pp. 597–600.

[9] G. Wolberg and S. Zokai, "Image registration for perspective deformation recovery," in *Proc. SPIE Int. Symp. Aerosp., Defense Sens., Simul. Controls*, 2000, pp. 259–270.

[10] P. Viola and W. M. Wells, III, "Alignment by maximization of mutual information," *Int. J. Comput. Vis.*, vol. 24, no. 2, pp. 137–154, Sep. 1997.

[11] F. Macs, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 187–198, Apr. 1997.

[12] P. Thevenaz and M. Unser, "An efficient mutual information optimizer for multiresolution image registration," in *Proc. Int. Conf. Image Process.*, Oct. 1998, vol. 1, pp. 833–837.

[13] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment," *Pattern Recognit.*, vol. 32, no. 1, pp. 71–86, Jan. 1999.

[14] P. A. Van Den Elsen, E. J. D. Pol, and M. A. Viergever, "Medical image matching—A review with classification," *IEEE Eng. Med. Biol. Mag.*, vol. 12, no. 1, pp. 26–39, Mar. 1993.

[15] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Med. Image Anal.*, vol. 2, no. 1, pp. 1–36, Mar. 1998.

[16] L. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, Dec. 1992.

[17] J. M. Fitzpatrick, D. L. G. Hill, and C. R. Maurer, Jr., "Image registration," in *Handbook of Medical Imaging*. Bellingham, WA: SPIE, 2000, pp. 447–513.

[18] B. Zitová and J. Flusser, "Image registration methods: A survey," *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, Oct. 2003.

[19] A. Gholipour, N. Kehtarnavaz, R. Briggs, M. Devous, and K. Gopinath, "Brain functional localization: A survey of image registration techniques," *IEEE Trans. Med. Imag.*, vol. 26, no. 4, pp. 427–451, Apr. 2007.

[20] M. Hemmendorff, "Motion estimation and compensation in medical imaging," Ph.D. dissertation, Dept. Biomed. Eng., Linköpings Univ., Linköping, Sweden, 2001.

[21] M. Hemmendorff, M. T. Andersson, T. Kronander, and H. Kuntsson, "Phase-based multidimensional volume registration," *IEEE Trans. Med. Imag.*, vol. 21, no. 12, pp. 1536–1543, Dec. 2002.

[22] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 123–151, Nov. 2005.

[23] N. G. Kingsbury, "The dual-tree complex wavelet transform: A new technique for shift invariance and directional filters," presented at the IEEE Digital Signal Processing Workshop, Bryce Canyon, UT, 1998, Paper 86, unpublished.

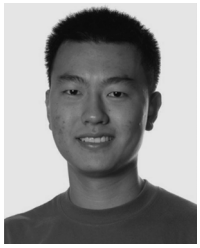
[24] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 587–607, Mar. 1992.

[25] R. Szeliski, "Image alignment and stitching: A tutorial," *Found. Trends Comput. Graph. Vis.*, vol. 2, no. 1, pp. 1–104, Jan. 2006.

[26] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature*, vol. 317, no. 6035, pp. 314–419, Sep. 1985.

[27] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1–3, pp. 185–203, Aug. 1981.

[28] J. F. A. Magarey and N. G. Kingsbury, "Motion estimation using a complex-valued wavelet transform," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 1069–1084, Apr. 1998.



Huizhong Chen (S'10) received the B.A. and M.Eng. degrees in electrical and information sciences from the University of Cambridge, Cambridge, U.K., in 2008 and 2009. He is currently working toward the Ph.D. degree in electrical engineering with Stanford University, Stanford, CA.

He worked with the Signal Processing and Communications Laboratory, Cambridge University, during his master's study, focusing on the applications of dual-tree complex wavelets. He also has past research experience on nanotube fabrication with the Nanomaterials Group, Cambridge University. His current research is include mobile visual search.

Mr. Chen was a recipient of the Kodak Fellowship and the Sir Ian McFarlane Scholarship.

Nick Kingsbury (M'87) received the Honor and Ph.D. degrees in electrical engineering from the University of Cambridge, Cambridge, U.K., in 1970 and 1974, respectively.

From 1973 to 1983, he was a Design Engineer and subsequently a Group Leader with Marconi Space and Defence Systems, Portsmouth, U.K., specializing in digital signal processing and coding, as applied to speech coders, spread-spectrum satellite communications, and advanced radio systems. Since 1983, he has been a Lecturer in communications systems and image processing with the University of Cambridge and a Fellow with Trinity College, Cambridge. In 2000, he was appointed as a Reader in signal processing with the University of Cambridge and, in 2007, to the position of a Professor of signal processing. He has developed the dual-tree complex wavelet transform and is particularly interested in the application of complex wavelets and related multiscale and multiresolution methods to the analysis of images and 3-D data sets. He is currently the Head of the Signal Processing and Communications Research Group. His current research interests include image analysis and enhancement techniques, object recognition, motion analysis, and registration methods.