manuscript No.
(will be inserted by the editor)

Three-Dimensional Object Registration Using Wavelet Features

Julie S. Chalfant · Nicholas M. Patrikalakis

Received: date / Accepted: date

Abstract Recent developments in shape-based modeling and data acquisition have brought three-dimensional models to the forefront of computer graphics and visualization research. New data acquisition methods are producing large numbers of models in a variety of fields. Three-dimensional registration (alignment) is key to the useful application of such models in areas from automated surface inspection to cancer detection and surgery.

The algorithms developed in this research accomplish automatic registration of three-dimensional voxelized models. We employ features in a wavelet transform domain to accomplish registration. The features are extracted in a multi-resolutional format, thus delineating features at various scales for robust and rapid matching. Registration is achieved by using a voting scheme to select peaks in sets of rotation quaternions, then separately identifying translation. The method is robust to occlusion, clutter and noise. The efficacy of the algorithm is demonstrated through examples from solid modeling and medical imaging applications.

Keywords 3D Object Registration · Shape-Based Matching · Wavelets

J. S. Chalfant
Department of Mechanical Engineering
Massachusetts Institute of Technology
77 Massachusetts Ave., Cambridge, MA 02139, USA
E-mail: chalfant@mit.edu

N. M. Patrikalakis
Department of Mechanical Engineering
Massachusetts Institute of Technology
77 Massachusetts Ave., Cambridge, MA 02139, USA

1 Introduction

Three-dimensional models are key components in many disciplines of computational research. Registration of these models is imperative in applications ranging from cancer detection to manufacturing and inspection. Data collection and storage techniques have advanced to the point that researchers today generate overwhelmingly large sets of data.

A current area of very active research involves registration of 3D medical images obtained through methods such as MRI or Computed Tomography (CT), which produce voxelized three-dimensional models. Advances in speed and accuracy of such matching, especially given the large size of the data sets, could lead to further advances in areas such as computer-aided surgery, cancer detection, disease diagnosis, and prosthesis fitting.

Two main approaches to 3D object registration exist: registration using geometric features and registration using voxel comparison. In general, registration using geometric features is faster but less accurate, while registration using voxel comparison is very accurate but requires an initial positioning as the methods tend to settle into local minima. In addition, most geometric feature methods are not designed for use on voxelized data. We present a fast, automatic, rigid-registration method using wavelet features. This method is designed specifically for voxelized data and provides an excellent initial positioning for further non-rigid registration using a voxel comparison method.

While specifically developed with the medical imaging application in mind, the methods developed in this work are equally applicable to voxelized models in areas such as manufacturing, in which inspection procedures may include CT scanning. Indeed, any voxelized model

or even a polyhedral model which has been voxelized can be addressed using these methods.

This algorithm has been previously introduced in [1]; this paper elaborates on the details of the algorithm and provides more examples that demonstrate robustness despite noise, clutter and occlusion, and that include real-world applications.

This paper is organized as follows. Section 2 reviews recent research in 3D object registration. Section 3 introduces basic wavelet theory, describes recent research in wavelets, and describes the dual-tree complex wavelet transform. The object matching algorithms developed by this research are described in Section 4. The algorithms are applied to several solid modeling and biomedical examples in Section 5. Section 6 draws conclusions and makes recommendations for future work.

2 Review of Previous Work

Three-dimensional object registration is a very active area of research. Two major methodologies used in registration of volumetric models include registration using geometric features and registration using voxel comparison. Registration using geometric features is generally fast and accurate but is not usually designed to operate on voxelized models. Registration using voxel comparison is much slower but more accurate, and requires an initial positioning to avoid settling into an incorrect local minimum. Our method is a geometric feature method which is designed to operate on voxelized models, and is fast and accurate enough to act as a preprocessing step for voxel comparison methods.

Registration using geometric features. These methods use geometric features such as points, curvatures, ridges, or segments to align objects. The gold standard of these methods is the Iterative Closest Point (ICP) algorithm presented by Besl and McKay [2], which compares each point in the sample to the closest point in the model and finds a registration by minimizing the average Euclidean distance between points. Numerous modifications to this method incorporate factors such as local shape [3], outliers and occlusion [4], and partial matching [5].

Registration methods in general expand on this idea by extracting and matching like features. Many of these methods find a coarse registration which is then refined using ICP. Some examples of geometric features used in matching include surface curvature [6], geodesic rings [7], iso-curvature lines [8], curvature maxima [9], umbilics [10], height function critical points [11], or a combination of features [12, 13].

Our research builds upon work by members of the MIT Ocean Engineering Design Laboratory who developed a procedure for shape-based matching using shape intrinsic fingerprints of 3D models in NURBS representation [10]. However, the algorithms and methods developed in [10] depend upon the NURBS format of the data and cannot be applied to polyhedral or voxelized models, which do not have the mathematical representation required for the classification of umbilics.

Registration using voxel comparison. These methods involve the alignment and comparison of objects based on voxel representations such as intensity, color, illumination and reflectance. The primary method used is mutual information, which is a statistical measure of the amount of information in data; examples include [14, 15, 16]. Modifications include hardware acceleration [17], cumulative distribution [18] and gradient-based terms [19]. Many other voxel comparison methods exist, including cross-correlation, Fourier domain based cross-correlation, minimization of variance of grey values, histogram clustering, minimization of intensity differences, and maximization of zero crossings, as described in the review by Maintz and Viergever [20]. All share the advantage of using all available information throughout the registration process instead of selecting discrete features, but share the disadvantage of considerable computational costs. In addition, most methods require initial positioning information as they tend to settle into local minima.

3 Theoretical Background

3.1 Review of the Wavelet Transform

The wavelet transform [21, 22, 23, 24] decomposes an image according to scale while maintaining localized information in a multiresolution format, thus making it a useful tool for global and partial matching of objects. The wavelet transform is the convolution of a wavelet function, $\psi(t)$, with a signal, $x(t)$ [21]:

$$T(a,b) = \int_{-\infty}^{\infty} x(t)\psi^*_{a,b}(t)dt \tag{1}$$

at various locations, b, and scales, a:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right). \tag{2}$$

The wavelet function, $\psi(t)$, must meet the admissibility conditions of zero mean (no DC component), finite energy (localized), and finite admissibility coefficient (inverse transform exists). The Discrete Wavelet Transform (DWT) allows numerical implementation of equation (1) with an efficient algorithm of complexity $O(n)$,

where n is the number of data points in the signal [21]. The most common form of the DWT is discretized on a dyadic scale [21] where the size of the steps between locations, $(n2^m)$, is directly linked to the scale, (2^m). Thus, equation (1) becomes:

$$T_{m,n} = \frac{1}{\sqrt{2^m}} \int_{-\infty}^{\infty} x(t)\psi^* \left(\frac{t - n2^m}{2^m} \right) dt. \quad (3)$$

Dyadic discrete wavelets have associated scaling functions, ϕ, of the same form as the wavelets:

$$\phi_{m,n}(t) = \frac{1}{\sqrt{2^m}} \phi \left(\frac{t - n2^m}{2^m} \right), \quad (4)$$

but which have a mean equal to 1, instead of zero. The convolution of the scaling function with the signal produces a smoothed, or averaged, version of the signal, represented by approximation coefficients, $S_{m,n}$:

$$S_{m,n} = \int_{-\infty}^{\infty} x(t)\phi_{m,n}(t)dt. \quad (5)$$

The wavelet coefficients, $T_{m,n}$, from equation (3), are also known as detail coefficients, since they provide the detail required to obtain the original signal from the smoothed signal. The original signal can be reconstructed from the approximation coefficients at a scale m_0 plus all detail coefficients from m_0 back to the original signal.

Higher-dimensional DWTs can be formed by taking tensor products of the one-dimensional DWT; these are termed separable wavelets. It is also possible to construct non-separable wavelets which are designed in multiple dimensions and use non-rectangular grids [25].

3.2 Translation and Rotation Invariance

The discrete wavelet transform can vary greatly with both translation and rotation. A small change in the location of an object can result in a large change in the corresponding wavelet coefficient values. By the nature of the object registration problem, the objects are located in different places, thus rendering the DWT coefficients nearly useless as features unless the translation and rotation variance is overcome.

Shift or translation variance is caused by the downsampling of data at each level of wavelet decomposition, with the result that the data points to which the wavelet is applied are different when the signal is shifted, unless it is shifted only by a multiple of the coarsest scale used in the transform. Thus the wavelet coefficients for signals that differ only by translation can vary significantly.

Rotation variance in tensor product wavelets is due to the strong coupling of the wavelets with the orientation of the axes, highlighting features aligned with the axes and at diagonals. Therefore, as an object is rotated, its features will align with different axes and be highlighted in a different sub-band of the transform. Since discrete non-separable wavelets are not symmetric, even non-separable wavelets vary with rotation

Much recent research has been conducted in an attempt to develop invariant versions of the discrete wavelet transform [26, 27, 28, 29, 30]. Many of the solutions produce extremely redundant representations, or solve either shift or rotational invariance but not both.

3.3 Dual-Tree Complex Wavelet Transform

Kingsbury [31, 32, 33] proposed the dual-tree complex wavelet transform (DTCWT) which achieves approximate shift invariance by omitting the downsampling at the first level and interleaving samples at lower levels. At the first level of decomposition, this method produces two parallel trees which use opposite samples from the signal. Below the first level, the samples of the two trees are interleaved by using q-shift filters designed to have a group delay of one quarter sample; the reverse filter has a group delay of three-quarters sample, thus producing a half-sample difference between the two trees. The q-shift wavelet allows perfect reconstruction using orthonormal wavelets such that the reconstruction wavelets are the exact reverse of the analysis wavelets. Details of the design of the q-shift filter can be found in [32, 33]. Kingsbury stores the outputs of the two trees as the real and imaginary parts of complex wavelet coefficients; thus, the method is labeled the dual-tree *complex* wavelet transform.

This method is much less data intensive than the continuous wavelet transform: 2^d redundant trees are retained, where d is the dimensionality. In many cases, the highest resolution details contain mostly noise; if this level is discarded, then the redundancy is removed. Of course, this also makes perfect reconstruction impossible as well.

The DTCWT, although not completely rotation invariant, is much more resistant to rotation than the DWT. In two dimensions, the DTCWT produces subimages strongly oriented in the following six directions: $\pm 15^o$, $\pm 45^o$ and $\pm 75^o$ [32], as compared to real wavelets which produce three sub-images: two strongly oriented horizontally and vertically, and the third including components from both diagonals. DTCWTs at higher dimensions produce $(4^d - 2^d)/2$ directional sub-bands at each level [32], where d is the dimension.

The low levels of redundancy combined with the greatly reduced translation and rotation variance make the Kingsbury DTCWT an excellent candidate for use in object registration.

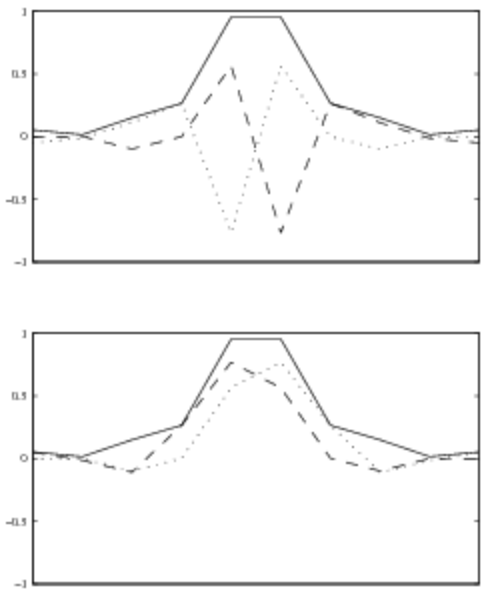

Fig. 1 A q-shift wavelet (top) and scaling function (bottom). The two symmetric lines are trees a (dotted line) and b (dashed line); the upper solid line is $|a + jb|$.

4 Object Registration

In this section we present an algorithm that employs the wavelet transform to accomplish automatic, global, rigid registration of three-dimensional voxelized objects which have undergone translation and rotation. The method is successful despite noise, occlusion and clutter which can be induced in models through such actions as transformation, voxelization, and the sensing process itself.

Given two voxelized models, A and B, of the same size and resolution with cubical voxels, we determine the rotation and translation such that:

$$A = \mathbf{R}B + \mathbf{T} \tag{6}$$

where $\mathbf{R}$ is the orthogonal rotation matrix and $\mathbf{T}$ is the translation vector.

An overview of the algorithm is as follows:

1. Apply a wavelet transform to each object.
2. Select the large magnitude wavelet coefficients in A and B as feature points.
3. Search within the lists of feature points for matching triplets of points. For each matching set of triplets, determine the rotation.
4. Use a voting scheme to select the best rotation.
5. Using the selected rotation, determine the translation.

The steps of the algorithm and the design choices developed therein are described in the following sections.

4.1 Wavelet Transform

A transformation that is invariant to rotation and translation (or nearly so) is desired for object registration. If the transformation results vary greatly with rotation and/or translation of the object, then corresponding feature points in each model will have completely different signatures and will be useless for registration purposes.

At the same time, a transformation that is computationally simple with minimal data requirements is desired, especially when working on the very large data sets that are common today. For example, the Mexican Hat Wavelet is a symmetric, 2D wavelet which is thus invariant to rotation; however, it is a continuous wavelet transform with the corresponding huge data sets and slow computation, which greatly reduces its utility.

Kingsbury's DTCWT [32] is nearly invariant to translation and is much more resistant to rotation than the discrete wavelet transform (DWT), while maintaining the $O(n)$ computation of the DWT with only a 2^d increase in data size, where d is the dimensionality. The increase in data size can be eliminated by not retaining the first-level wavelet coefficients. For these reasons, we use the DTCWT as the wavelet transform of choice in this algorithm.

Within the DTCWT family, there are a number of filters among which we can choose. For the first-level discrete wavelet filter, we select the nearly symmetric filter with the shortest length. For the second- and greater-level filter, we select the shortest-length q-shift filter. The short-length filters were selected so that abrupt changes in the composition of the object would be identified. Longer filters incorporate voxels over a greater range, giving weight to areas beyond edges and thus blurring the distinct change found at a feature. This adversely affects feature point selection.

4.2 Feature Points

Applying the three-dimensional DTCWT to each model produces, at each level of resolution, approximation (scaling function) and detail (wavelet) coefficients. There are several possibilities for sources of feature points, listed below and shown in Figure 2:

- The original object, with no wavelet transform applied.
- The scaling function coefficients at some specific level of resolution.
- The wavelet coefficients at one or more levels of resolution.
- The gradient of the wavelet coefficients.

In the two-dimensional example shown in Figure 2, the images containing the scaling coefficients and the gradient of the wavelet coefficients contain large swaths with identical extreme values; note the large contiguous areas of black or white. This is especially true in 3D for surface models with a constant value for the entire interior. Alternatively, the wavelet coefficients exemplify more discrete extrema because they highlight changes in intensity. Therefore, the wavelet coefficient extrema are the best choice for feature points.

This was confirmed through heuristic tests upon many three-dimensional models at many different orientations. The wavelet coefficients produced the highest percentage of feature points that matched in location from one model to another and were thus the most useful for object registration.

The three-dimensional DTCWT produces 28 complex sub-bands of wavelet coefficient information; instead of using 28 separate data sets, we reconstruct one level of wavelet coefficients, excluding the average coefficients, in order to produce a single set of real wavelet coefficient data. Details on this process can be found in [32].

Invariance. Although the DTCWT is nearly invariant to translation and is much less affected by rotation than other methods, it is completely invariant to neither rotation nor translation. Therefore, even though many of the DTCWT wavelet coefficient extrema identify the same features in each object, not all features are identified in both objects. This results in a set of feature points with many outliers.

As an example, we separately apply the DTCWT and the DWT to a two-dimensional image that has been translated by 42 pixels in the x and y direction, and to another that has been rotated by 30 degrees counterclockwise. We then compare the number of corresponding maxima of the wavelet coefficients at the third level of decomposition; 67-87% of the DTCWT feature points correspond, while only 47-60% of the DWT feature points correspond. See Figure 3 for a visual comparison and Table 1 for a numerical comparison.

It is unlikely that the feature points correspond by chance. Table 1 includes the probability of randomly selected feature points matching by chance given the size of the objects and the number of feature points selected in the examples shown in Figure 3.

Wavelet Coefficient Magnitude. A one-to-one correspondence of wavelet coefficients based on coefficient value cannot be used to pinpoint the location of a feature. The magnitude of the wavelet coefficient is different at corresponding points in two objects, and the order of

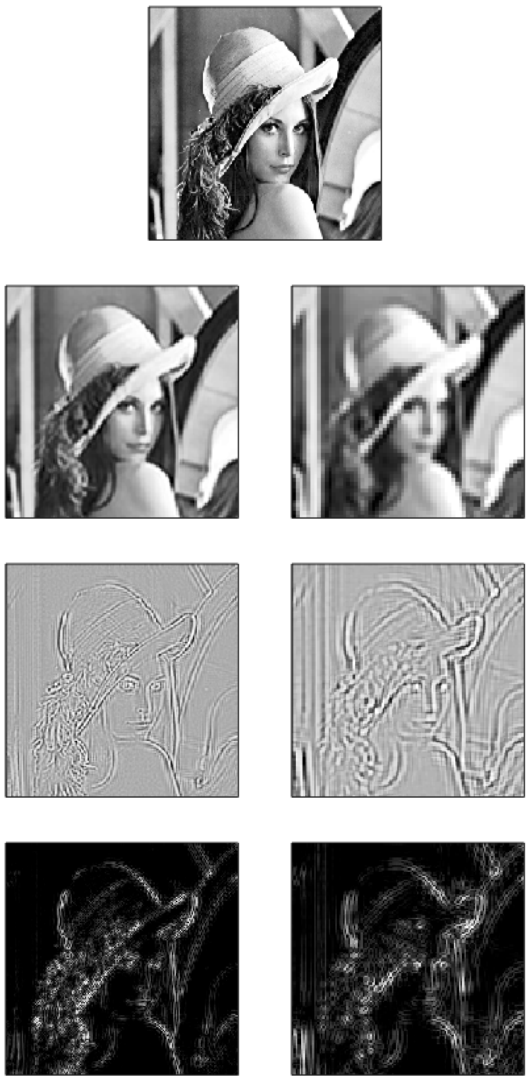

Fig. 2 A two-dimensional example of the DTCWT. The top figure is the original image. The three subsequent rows are the average (scaling function) coefficients, detail (wavelet) coefficients, and the gradient of the wavelet coefficients, in that order, with the second-level decomposition on the left and the third-level decomposition on the right.

Table 1 Comparison of feature point correspondence for DWT and DTCWT in translation and rotation. Number of corresponding feature points, percent of total feature points, and probability of matching that number of points by chance are shown for the third level DWT or DTCWT decomposition of two translated or rotated versions of the birthday image, shown in Figure 3. Fifteen total maxima were selected in each image for comparison.

	DWT	DTCWT
Translation		
number	7	10
percent	47	67
probability	$1.42\text{x}10^{-8}$	$1.58\text{x}10^{-14}$
Rotation		
number	9	13
percent	60	87
probability	$2.28\text{x}10^{-12}$	$2.32\text{x}10^{-22}$

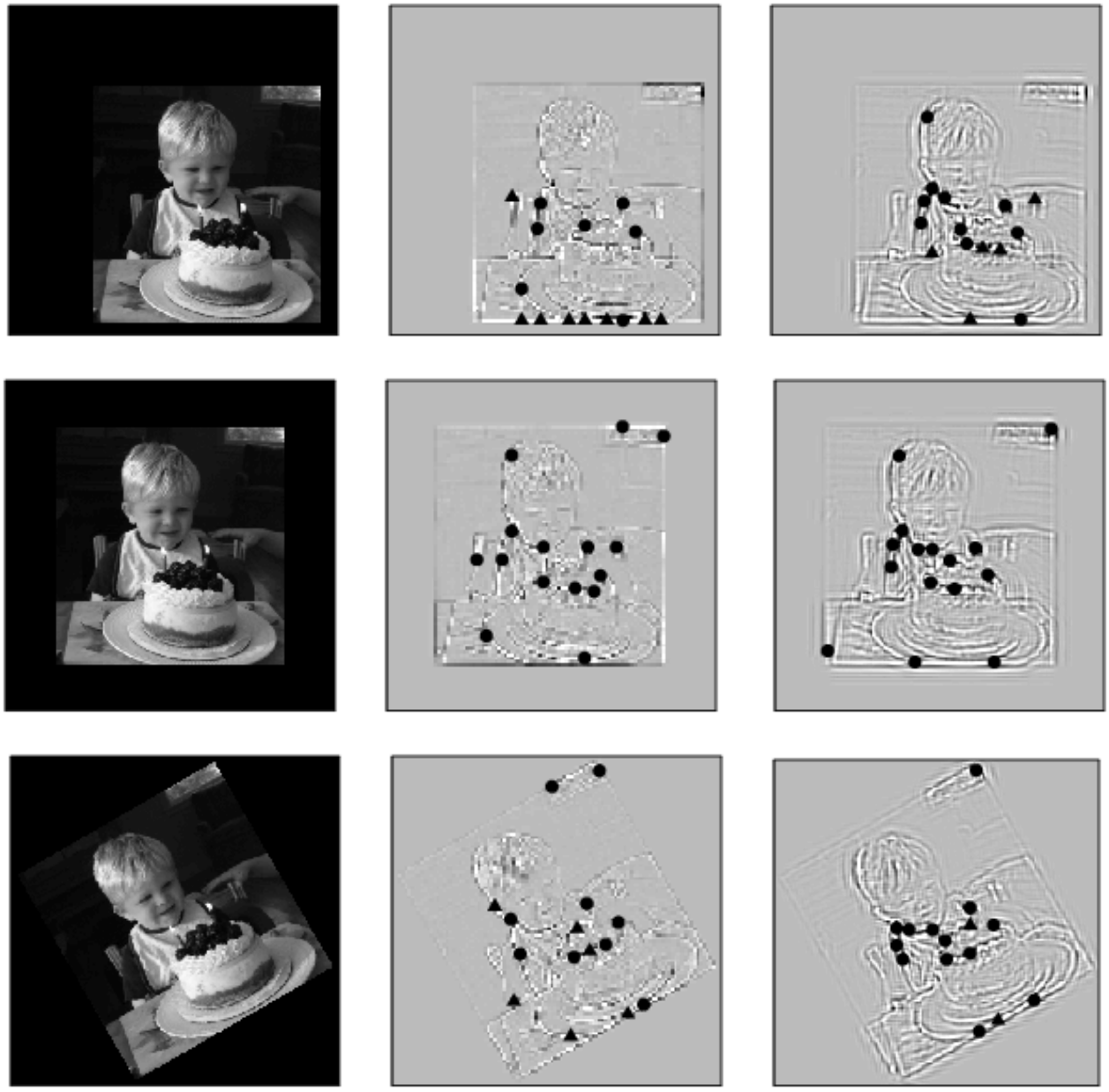

Fig. 3 Comparison of DWT and DTCWT extrema in rotation and translation. The center row is the image in the original location. The top row was translated 42 pixels in the i and j direction prior to application of the transform. The bottom row was rotated 30 degrees counterclockwise prior to application of the transform. The left-hand column is the original image, the center column is the DWT, and the right-hand column is the DTCWT. The 15 wavelet coefficients with the greatest value (positive only) in each image are marked. In the top and bottom rows, those feature points that correspond in location to the unrotated or untranslated image are marked with circles; those that do not correspond in location are marked with triangles.

the points differs when listed by magnitude, i.e., the feature with the greatest coefficient magnitude may not coincide with the feature of the greatest coefficient magnitude in the rotated and translated object. However, both of the features denoted with greatest magnitude in each object are likely to be represented in a group of points of large magnitude for each object.

As an example, in Figure 4, the six wavelet coefficients with the largest magnitudes are marked with colored dots on the level three reconstructed wavelet coefficients of two versions of the Lena picture, one translated from the other. The location and coefficient value are shown in Table 2. Note that although five of the six points match in location, only one matching pair holds the same ranking in each list, and points which seem to be very close in value do not necessarily correspond in position.

Therefore, the values of the wavelet coefficients are used to screen the image for feature points, but the particular value at each point is not used for locating matching points. Instead, a representation is used in which the top maxima are equally weighted, the bottom minima are equally weighted, and no other points are considered.

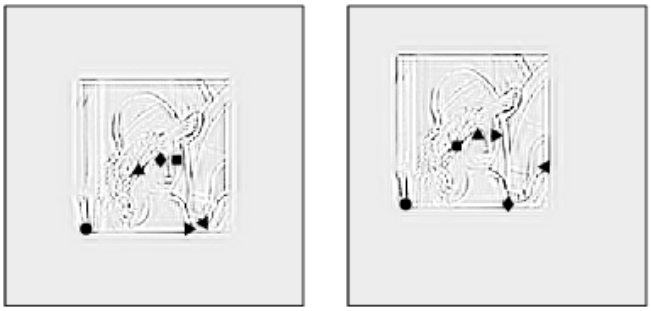

Fig. 4 The third level DTCWT decomposition of two translated versions of the Lena image. The six points with highest magnitude of coefficients are marked with symbols. In order of descending magnitude, the symbols are: circle, diamond, right-pointing triangle, left-pointing triangle, upward-pointing triangle, and square. The coefficient magnitudes and correspondences are listed in Table 2.

Table 2 Coefficient values of corresponding points in the third level DTCWT decomposition of two translated versions of the Lena image shown in Figure 4. The points from the left-hand image, denoted A, are listed in descending order of magnitude in the left-hand column. The points that correspond in location in the right-hand image are listed in the right-hand column. Note that the order of magnitude is different, and that for the most part, the magnitudes are quite different.

A			B		
Order	Symbol	Coeff. Mag.	Order	Symbol	Coeff. Mag.
1	circle	224.7	1	circle	206.3
2	diamond	177.3	5	up tri.	164.7
3	right tri.	170.2	2	diamond	180.7
4	left tri	159.8	n/a		
5	up tri.	159.7	6	square	159.3
6	square	151.1	3	right tri.	173.1
n/a			4	left tri.	171.2

Quantity of Feature Points. Another consideration is the proper number of feature points. Selecting too many points causes the algorithm to run slowly and increases the number of false positives by increasing the likelihood of random correspondence, while too few points will not achieve registration.

One upper limit on feature point quantity is introduced by the potential of saturating the object such that randomly matching feature points cause false positive rotations. This is discussed more in the discussion on levels of resolution below.

Another consideration in the upper limit on the number of feature points is the computational complexity of the algorithm, which depends both upon the size of the model and the number of feature points. The computational complexity is the greater of $O(n)$, where n is the number of voxels in the model, and $O\binom{m}{3}^2$, where m is the number of feature points. This is described in more detail in Section 4.6. Choosing 30 feature points yields $\binom{30}{3}^2 = 16$ million as a worst case. In actuality, a quarter or less of these become viable triangles for testing. Depending upon the number of outliers, the speed of the registration portion of the algorithm runs from 18 to 56 seconds. Increasing the number of feature points greatly slows the algorithm.

The lower limit on the number of feature points was first determined using the following line of reasoning, then was confirmed heuristically. Given m feature points, the total number of triangles is $\binom{m}{3}$. Assuming three quarters of the feature points are outliers (an assumption on the high end), only $\binom{m/4}{3}$ are potentially matching. Of those, some will be eliminated because they are collinear or too small; we assume this is 20% (a number that is not extreme based on heuristics). There are four sets of feature points: positive and negative coefficients at both the second and third levels of resolution. This calculation yields 4 matching triplets for 10 feature points, 32 matching triplets for 20 points, 180 matching triplets for 30 points, and 384 matching triplets for 40 points. The many examples that were run in the process of generating and verifying this algorithm support this line of reasoning. Ten to fifteen feature points often produced a correct rotation but occasionally did not. Twenty-five to thirty-five points reliably produced the correct rotation.

Levels of Resolution. The number of levels of resolution of the wavelet transform and which levels of resolution are selected for feature point identification can be varied as well. It is noted that the number of points from which feature points are selected reduces by 2^d at each level of resolution, where d is the dimensionality. Therefore, at some point there are too few data points and the object becomes saturated with feature points. On the examples we are running, we reach saturation at the fourth or fifth level of resolution, so we select our lowest level of resolution to be three. See Table 3.

Through testing on a number of models, it was also determined that very few of the successfully matching triangles were produced by the first level of resolution. By eliminating the first level wavelet coefficients, we

Table 3 The saturation level of resolution for various models. Level indicates the level of resolution (note that levels 1 and 2 have the same number of points and are therefore listed together). Column n indicates the number of voxels in each level of resolution. Column p indicates the probability of randomly matching 7 of 20 feature points within 4 voxels. Column f indicates the percentage of voxels in the entire image that fall within a 4-voxel ring of 20 feature points, assuming no voxel rings overlap each other or the edges of the model. Note that each model becomes saturated at the fourth to fifth level of resolution.

Model	Level	n	p	f
Brain	1 and 2	2,457,600	3.52e-09	0.89
MRI	3	307,200	0.0020	7.12
	4	38,400	≈1	56.95
Knee	1 and 2	7,077,888	2.41e-12	0.31
	3	884,736	3.24e-06	2.47
	4	110,592	0.14	19.78
Gooch	1 and 2	11,534,336	8.10e-14	0.19
	3	1,441,792	1.29e-07	1.52
	4	180,224	.0273	12.13
	5	22,528	≈1	97.08
Shrek	1 and 2	1,741,916	3.63e-08	1.26
	3	217,739	.0116	10.04
	4	27,217	≈1	80.35

save time and memory space in the algorithm with no penalty in the quality of the registration.

4.3 Feature Point Matching

Beginning with all possible combinations of three feature points in models A and B, we reduce the number used by selecting only corresponding triplets with essentially the same distances and angles between the three points, after first eliminating collinear triplets and very small triangles.

For potentially matching triplets, we calculate a rotation using the closed form solution found by Horn [34], using unit quaternions. The quaternion representation has an ambiguity in that q and -q describe the same rotation; we eliminate this by restricting one quaternion coefficient, q_1, to be greater than zero.

4.4 Voting Scheme

Potentially corresponding triplets that are paired correctly will generate transformations that are close in value to one another, differing only by noise and error in the calculation of the rotation. The outliers, which generate incorrectly corresponding triplets, will produce widely varying transformations that are unlikely to correspond. Therefore, calculating rotations for each set of potentially corresponding triplets generates a set of potential rotations with a cluster at the desired rotation and noise elsewhere. We use a histogram voting scheme to locate the cluster and thus the desired rotation. This voting scheme is based loosely on the ideas found in the Hough transform [35] and pose clustering [36], but is tailored to this specific application.

Rotation Representation. The voting scheme could be based on any of the possible representations of rotation, such as the orthogonal rotation matrix, Euler angles, unit quaternions, or the axis-angle representation. Using the rotation matrix would require searching for a peak in 9-space, which is more cumbersome than other choices. Euler angles suffer from gimbal lock problems and are therefore not a stable representation upon which to base the voting scheme. Two solid choices, both of which would require searching for a peak in 4-space, are quaternions and the axis-angle representation. The conversion from one format to the other is:

$$q_0 = \cos(\alpha/2)$$
$$q_1 = \sin(\alpha/2)\cos(\beta_x)$$
$$q_2 = \sin(\alpha/2)\cos(\beta_y)$$
$$q_3 = \sin(\alpha/2)\cos(\beta_z),$$

where $[q_0\ q_1\ q_2\ q_3]$ are the coefficients of the quaternion, and α, β_x, β_y and β_z are the rotation angle and direction cosines of the axis-angle representation. Note that this conversion is nonlinear; evenly spaced bins in one representation do not correspond to evenly spaced bins in the other. Also note that the degrees of freedom in each representation are reduced by one: $\beta_x^2+\beta_y^2+\beta_z^2 = 1$, and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$.

If we use equally spaced histogram bins for voting based on quaternions, then a bin for a single quaternion coefficient will correspond to a greater arc length in the bins near -1 and 1 than in the bins near zero. However, the bins of the remaining coefficients will correspond to much shorter arc lengths; if one quaternion value is close to one, the remaining values must be close to zero to maintain the unit magnitude requirement. Thus the overall size of the 4-space bin does not vary greatly.

As an alternative, one could base the voting scheme on the axis-angle representation. This has the seeming advantage that equally spaced bins correspond to a linear division of degrees in real space. However, bins of equal numbers of degrees at the equator of a sphere are much larger than those at the poles.

The voting scheme based on quaternions is the most viable of the alternatives and has the added advantage that we calculate our rotation in the quaternion format and therefore require no conversion before voting.

During the voting process, the quaternion coefficier bins are equally spaced from -1 to 1 for each coefficien except q_1 which runs from 0 to 1 to prevent redundanc This means that many of the bins will be empty sinc only unit quaternions are used.

Histogram Bin Size. A proper bin size is required fc determining the peak. If the bins are too large, the there may be sufficient outliers in an incorrect bin t obscure the desired result. In the extreme, a single bi would be of no assistance at all. Conversely, bins tha are too small may split the correct transformation pea into separate bins, again obscuring the peak. In th extreme, each bin would have one or zero items in tl bin and no peak would be found.

Many treatises on correct bin size for a histogra [37, 38, 39] discuss matching the underlying statistic of the data. In our application, we have a single larg peak and much surrounding noise as shown in Figu 5. We are interested only in finding the peak and not i any analysis of items not in the peak; therefore, stati tical analysis of non-peak bins is not required, and tl number of samples in non-peak bins is not relevant.

The goal bin size is one that allows us to quick identify the peak in the data, and to pinpoint it wit sufficient accuracy for our desired application. Our al gorithm achieves global registration of voxelized objects explicitly enough to provide an initial estimate for fine registration systems such as ICP or voxel intensity measures to be successfully applied without settling into local minima. We define sufficient accuracy as less than two degrees error in each Euler angle. Within this defined goal, a bin size that is too small increases computational time, and a bin size that is too large includes outliers which can mask the proper result.

We use the quaternion coefficients in our histogram, while our definition of sufficient accuracy is based on Euler angles. The nonlinear conversion is as follows:

$$\phi = \arcsin\big(-2(q_2q_4 + q_1q_3)\big)$$
$$\theta = \arctan\left(\frac{-2(q_3q_4 - q_1q_2)}{q_1^2 - q_2^2 - q_3^2 + q_4^2}\right)$$
$$\psi = \arctan\left(\frac{-2(q_2q_3 - q_1q_4)}{q_1^2 + q_2^2 - q_3^2 - q_4^2}\right).$$

A bin size of 0.1 for the quaternion coefficients corresponds to an Euler angle bin size ranging from approximately 2 to 10 degrees, with some ranging up to 15 degrees, depending upon which quaternion coefficient and where in the range from -1 to 1 the bin occurs.

Histogram Origin. Identifying a peak in data with a histogram leads to the problem of proper origin placement in order to avoid splitting the peak. If the bin edge occurs exactly upon the peak, the peak will be split with only half the number of samples appearing in two adjacent bins, which may obscure the peak from detection.

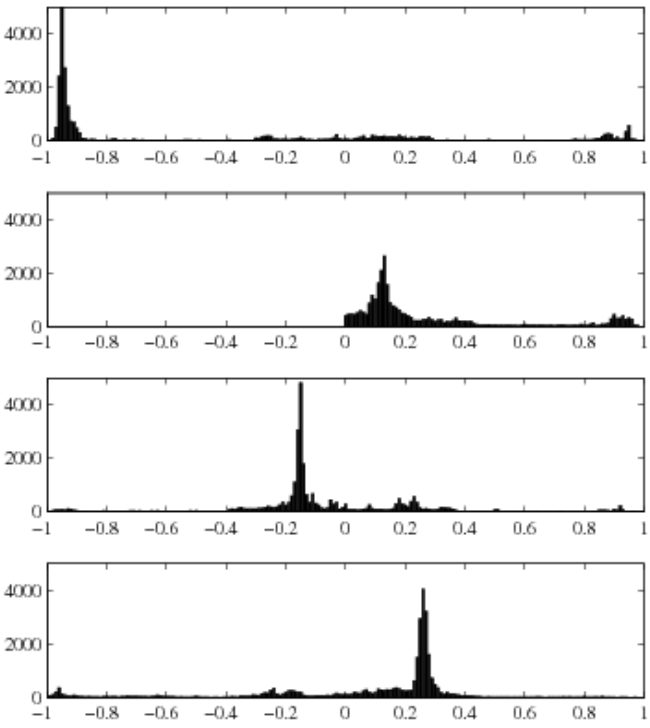

Fig. 5 A histogram of each coefficient in the rotation quaternion voting scheme. A significant peak occurs at the proper rotation with noise elsewhere. Note that this is a one-dimensional histogram of each coefficient and does not show the interrelation between coefficients; the actual histogram performed is four-dimensional.

Scott [38] developed a method called an averaged shifted histogram (ASH) which eliminates this problem by averaging multiple histograms of equal bin widths but shifted bin locations. The method is statistically comparable to kernel estimators with much reduced computational complexity. The method is implemented by selecting a bin size, h, and a number of histograms to be averaged, m. Next, a histogram is performed on the set of data using a small bin size, $\delta = h/m$. Adjacent bins are then averaged. In three dimensions, the histogram value at sample $\mathbf{x}$ is [38]:

$$\hat{f}(\mathbf{x}) = \frac{1}{h_1h_2h_3}\sum_{i_1}\sum_{i_2}\sum_{i_3} wn_{k_1+i_1,k_2+i_2,k_3+i_3},$$

where $w = (m - |i_1|)(m - |i_2|)(m - |i_3|)$, i_j runs from $1-m_j$ to m_j-1, and $\mathbf{x}$ is in bin I_{k_1,k_2,k_3} which contains n_{k_1,k_2,k_3} values.

Through testing it was verified that the identification of the peak is fairly robust to variations in bin size. Bin width options from 0.001 to 0.2 and numbers of averaged histograms from 1 to 20 were tested; a range of bin widths from 0.001 to 0.05 with 2 to 10 histograms averaged produced acceptable results. We selected a bin width of 0.02 and two averaged histograms.

Locating the Peak Quaternion. By using the averaged shifted histogram method, we identify a small histogram bin at the center of the peak. We use the values for the rotation quaternion from this small bin instead of some combination of the averaged values in the larger shifted bin. This has the effect of pinpointing samples in the center of the peak, thus reducing the variation of the samples used to determine the final (normalized) rotation quaternion.

4.5 Translation

We then take the top few rotations (those with the greatest number of votes), and calculate a translation. We multiply the list of points from model B by the selected rotation, then determine the single translation that aligns the greatest number of points from B with A. Note that the translations determined by lower levels of resolution are half the length of the next higher level.

4.6 Analysis

Computational Complexity. The complexity of the algorithm depends upon the size of the objects and the number of feature points. There are three main parts of the algorithm which make significant contributions to computational complexity.

The first is the application of the wavelet transform and reconstruction of the wavelet coefficients. Per Kingsbury [32], the computational complexity of the DTCWT is $O(n)$, where n is the number of voxels in the object.

The second major contributor is the screening of all possible triplets of feature points, which is $O\binom{m}{3}$.

The third major contributor is the comparison of the two lists of potentially matching triplets. These triplets of feature points are pared down by eliminating collinear points, small triangles, and triplets without matching triplets in the other model. In the worst case, if no triplets were eliminated and the perimeter and angles of all triplets in one model matched those of all triplets in the second model, the complexity of the algorithm would be $O\binom{m}{3}^2$.

Speed. As stated above, the speed of the algorithm depends both upon the size of the object and the number of feature points selected. Table 4 shows speed of the algorithm in two sections; the first is the wavelet transform and feature point selection which depends upon the size of the object. The second is the point matching, voting, and registration which depends upon the number of feature points. The algorithm is run on a Core 2 Duo processor. Times are elapsed time.

The first time, which measures the part of the algorithm that is of complexity $O(n)$, is very consistent. The second time, which measures the part of the algorithm that is of worst case complexity $O\binom{m}{3}^2$, varies greatly depending upon the orientation of the model, as would be expected. As the rotation and translation from one model to the other changes, the number of outliers changes, and thus the number of matching triangles changes.

The algorithm compares favorably with other algorithms of its type. State of the art geometric feature registration algorithms run in the neighborhood of 70 to 300 seconds. This algorithm runs in the neighborhood of 20 to 150 seconds, depending upon the size of the object.

One possible use of the algorithm is as an initial processing step for voxel intensity methods which accomplish very fine elastic registration but which require initial coarse registration in order to avoid incorrect local minima. State of the art elastic registration methods run many minutes up to hours; the speed of this algorithm indicates its suitability for such an application.

5 Examples

In this section we show examples from a variety of applications. Both non-homogeneous models with continuously changing intensities throughout such as MRI scans and homogeneous models with a constant value in the interior such as voxelized surface models are included. We show two examples with artificially added Gaussian noise, one application of temporal matching, and one application with occlusion and clutter.

An important note is that the algorithm is not tuned to specific types of models. All examples were run using the second and third resolution wavelet coefficients with 30 feature points selected. The voting scheme is a four-dimensional averaged shifted histogram with bin size of 0.02 for each quaternion coefficient and two bins averaged.

Table 4 Speed of the wavelet registration algorithm. n is the number of voxels in the object. m is the number of feature points. The algorithm is divided into two parts. Time 1, with complexity $O(n)$, includes the wavelet transform and feature point selection. Time 2, with complexity $O\binom{m}{3}^2$, includes point matching, voting, and registration. Times are elapsed time.

Model	n	Time 1	m	$\binom{m}{3}^2$	Time 2
Knee MRI	7,077,888	14.4	30	16,483,600	18.2
Shrek Solid Model	3,932,160	12.8	30	16,483,600	33.1
Large Brain MRI	25,186,304	108.9	30	16,483,600	42.1
Brain MRI	7,614,464	7.0	15	207,025	1.9
Brain MRI	7,614,464	7.0	20	1,299,600	4.9
Brain MRI	7,614,464	7.0	25	5,290,000	13.6
Brain MRI	7,614,464	7.0	30	16,483,600	56.5
Brain MRI	7,614,464	7.0	35	42,837,025	374.2

5.1 Non-Homogeneous Objects

These objects are fully three dimensional, varying in intensity throughout the object. Both examples shown here are Magnetic Resonance Imaging (MRI) data. However, this description fits many types of data such as Computed Tomography (CT), Ultrasound (US), and functional MRI (fMRI).

These examples were formed by taking an object, then randomly selecting and applying an Euler angle rotation and rectangular translation to produce a second object. These two objects were registered using the wavelet extrema algorithm and the resulting registration is compared to the induced rotation and translation.

Knee MRI. The first example is an MRI scan of Paul Debevec's knee, which is available online courtesy of a torn ligament; see Figure 6. The original object was rotated by pitch-roll-yaw Euler angles of -20, 4, and 41 degrees, then translated 13, 8 and 11 voxels in the i, j and k directions respectively.

Registration was performed in a total elapsed time of 44 seconds; the results are presented in Table 5, showing the transformations with the eight highest votes in descending order. Rotation angles are the pitch, roll and yaw Euler angles, θ, ϕ and ψ, required for registration as determined by the algorithm. Δ_r is a pseudo-Manhattan distance which is the sum of the magnitude of the distance between each recommended angle and the inverse of the induced rotation. Vote is the number of votes received in the averaged shifted histogram voting scheme. Similarly, the translation is the number of voxels in the i, j and k directions required for registration as determined by the algorithm, and Δ_t is the Manhattan distance between the recommended translation and the inverse of the induced translation. Results are rounded to the nearest degree or voxel.

The top eight results are presented. The top result is the recommended answer; the rest are provided only for comparison and discussion. Note that the top seven results are very close to one another. The eighth, which is far from the desired result, received significantly fewer votes.

Table 5 Registration results for Knee MRI in descending order of vote. The correct rotation in pitch-roll-yaw Euler angles is 20, -4, and -41 degrees, with a translation of -13, -8 and -11 voxels.

Rotation			Δ_r	Vote	Translation			Δ_t
θ	ϕ	ψ			i	j	k	
20	-4	-40	1	5582	-12	-9	-10	2
20	-4	-42	1	4513	-12	-9	-10	2
21	-3	-40	3	4012	-12	-8	-10	2
18	-5	-40	3	3082	-13	-8	-10	1
20	-3	-43	2	2859	-12	-9	-10	3
18	-4	-42	3	2743	-13	-9	-10	2
20	-4	-38	3	2609	-12	-4	-11	5
22	5	-31	20	591	-14	3	-2	20

Brain MRI. The second example is an MRI scan of a brain (from MATLAB); see Figure 7. The original model was rotated 28, 37 and -34 degrees in a pitch-roll-yaw Euler angle scheme, then translated -14, 12 and -3 voxels to create the second model. Total elapsed time was 63 seconds, yielding Δ_r of 4 and Δ_t of 2. Results are displayed in Table 6.

5.2 Homogeneous Objects

The objects in this section are solid models which do not vary in the interior. If not already in voxelized form, mesh or other surface model types can be voxelized. We used Binvox, a binary voxelization tool produced by Patrick Min (http://www.cs.princeton.edu/∼min/binvox) based on the work of Nooruddin and Turk [40].

Shrek. The 3D mesh model of Shrek shown in Figure 8 was produced by Maurizio Pocci in 3D Valley. Two

Table 6 Registration results for various examples. For each model, the first line is the inverse of the induced rotation/translation, the second line is the algorithm's recommended rotation/translation for re-alignment. Note that the last two examples (Gooch Brain 1 and 2) have an initial misalignment, so the algorithm's results should not exactly match the induced amounts.

Model		Rotation			Δ_r	Translation			Δ_t
		θ	ϕ	ψ		i	j	k	
Knee	Induced	20	-4	-41		-13	-8	-11	
	Algorithm	20	-4	-40	1	-12	-9	-10	2
Brain	Induced	-28	-37	34		14	-12	3	
	Algorithm	-27	-36	33	4	14	-13	2	2
Full Shrek	Induced	-12	-15	14		-11	3	-11	
	Algorithm	-12	-14	14	1	-12	3	-10	2
Cropped Shrek	Induced	-10	-20	-30		-14	6	-3	
	Algorithm	-10	-21	-30	1	13	-7	4	4
Noise SNR 4.74	Induced	-10	-20	-30		-3	7	0	
	Algorithm	-10	-21	-30	1	-3	8	0	2
Noise SNR 0.67	Induced	-10	-20	-30		-3	7	0	
	Algorithm	-10	-20	-30	1	-3	8	0	1
Clutter	Induced	-6	-8	7		5	-3	2	
	Algorithm	-6	-6	6	3	5	-3	1	2
Gooch Brain 1	Induced	-5	-5	-70		0	0	0	
	Algorithm	1	-3	-73	10	8	4	-2	14
Gooch Brain 2	Induced	0	0	0		0	0	0	
	Algorithm	3	2	-3	8	7	-4	0	11

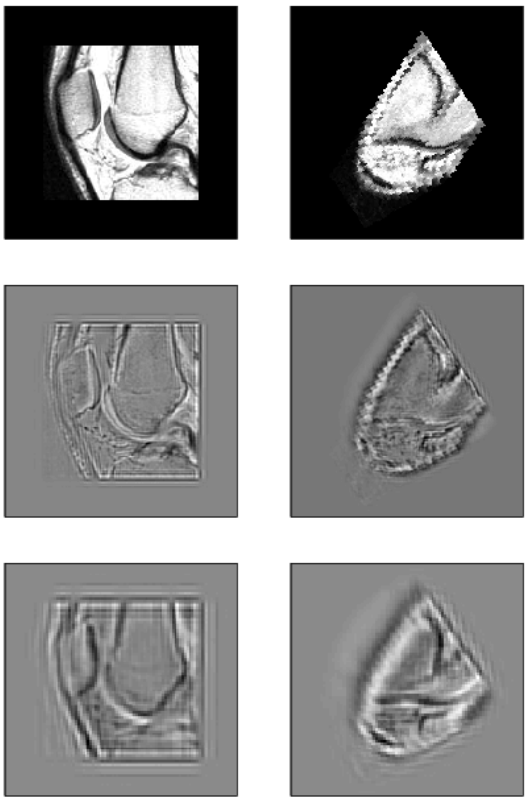

Fig. 6 Knee MRI scan. The top row is the original object. The middle and bottom rows are the reconstructed wavelet coefficients at the second and third levels of resolution respectively. The left-hand column is model A. The right-hand column is model B, which has been rotated and translated.

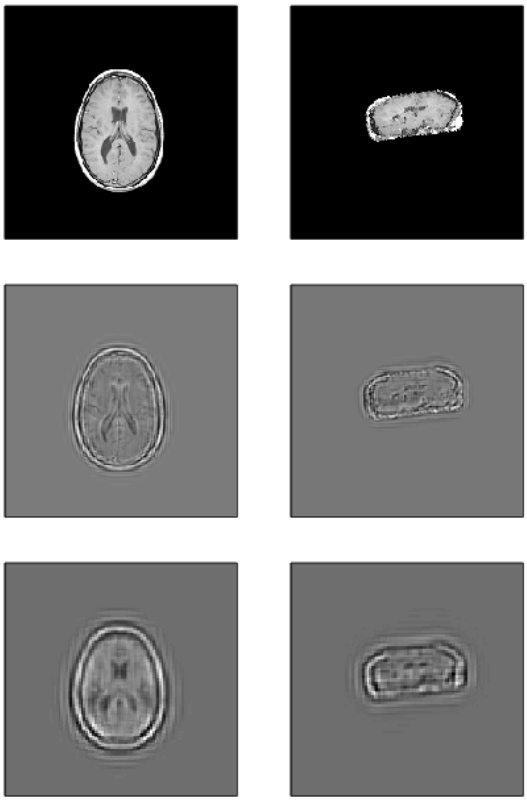

Fig. 7 Brain MRI scan (from MATLAB). The top row is the original object. The middle and bottom rows are the reconstructed wavelet coefficients at the second and third levels of resolution respectively. The left column is model A. The right column is model B, which has been rotated and translated.

tests were run. The first was a registration of the full model and the second a cropped portion of the model.

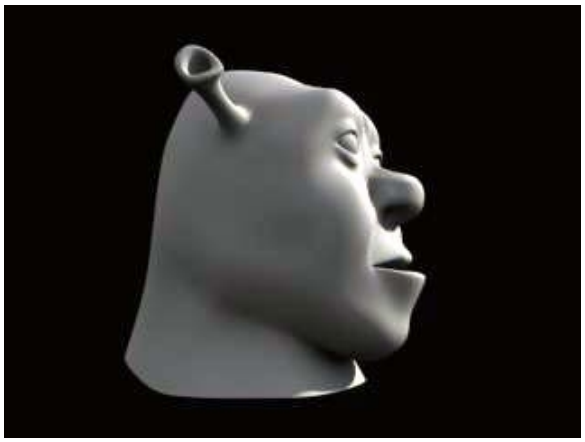

Fig. 8 Shrek model produced by Pocci and voxelized using Binvox.

The first test, using the full Shrek model with an induced rotation in pitch-roll-yaw Euler angles of 12, 15 and -14 degrees and an induced translation of 11, -3 and 11 voxels, had an elapsed time of 262 seconds, with a Δ_r of 1 degree and a Δ_t of 2 pixels. The second test, which used the cropped model, had an elapsed registration time of 67 seconds and yielded a Δ_r of 1 degree and a Δ_t of 4 pixels. See Table 6 and Figure 9.

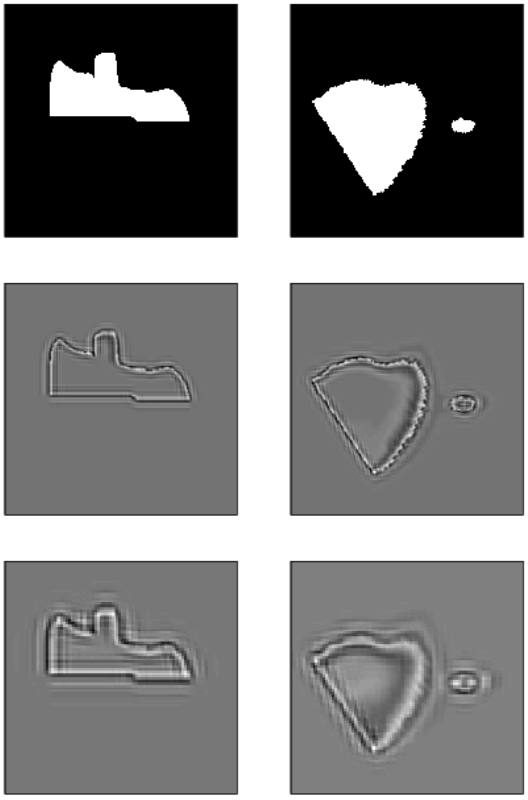

Fig. 9 Cropped Shrek solid model. The top row is the original object. The middle and bottom rows are the reconstructed wavelet coefficients at the second and third levels of resolution respectively. The left column is model A. The right column is model B, which has been rotated and translated.

5.3 Noise

Noise can be induced in 3D models in many ways. In medical imaging, noise is induced by the patient, either through movement or over time through growth and change. All sensing methods induce some type of noise. Manipulating the objects in their electronic form also induces noise. To test our method for noise sensitivity in a controlled manner, we added Gaussian noise to an object to obtain varying signal to noise ratios, then applied the algorithm to attempt registration.

Gaussian noise was added only to the rotated and translated object, which was then registered to the original object without noise. The Gaussian noise was generated using a normally distributed set of random numbers with a mean of zero, variance of one, and standard deviation of one. This set of randomly generated numbers was then multiplied by some greater standard deviation to achieve the desired signal to noise ratio (SNR). The SNR was calculated using the ratio of root mean square amplitudes of the signal and the noise [41]:

$$SNR = \left(\frac{A_{signal}}{A_{noise}}\right)^2 = \frac{\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} k_i^2}$$

where A is the root mean square amplitude, x_i is the voxel intensity at each point in the object, and k_i is the intensity of the noise added at each point in the object. The SNR can also be expressed in decibels:

$$SNR_{dB} = 10 \log_{10}\left(\frac{\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} k_i^2}\right)$$

As the test model, we used the knee MRI object with a rotation of 10, 20 and 30 degrees and a translation of -3, 7 and 0 voxels. The first test used a variance of 100 to achieve a SNR of 4.74, or 13.50 dB. Successful registration occurred in an elapsed time of 34 seconds. The second test used a variance of 5000 to achieve a SNR of 0.67, or -3.48 dB. Successful registration occurred in an elapsed time of 46 seconds. Results are shown in Table 6.

Note that there is noise induced in each of these tests through the initial rotation process in addition to the Gaussian noise induced.

This algorithm uses the second- and third-level resolution wavelet coefficients for registration. The first-level resolution wavelet coefficients contain information relating to the smallest scale details; much of the noise is contained in this level. This makes the algorithm

quite robust to noise, as demonstrated by the examples herein.

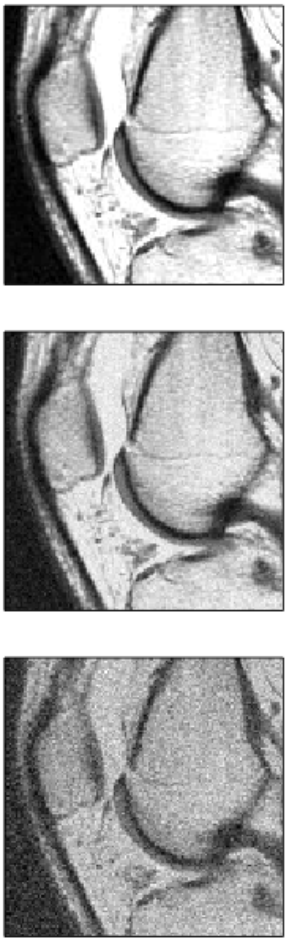

Fig. 10 Knee MRI scan with Gaussian noise added. Top image without noise, center image with SNR 4.74, bottom image with SNR 0.67.

5.4 Occlusion and Clutter

This example was constructed using Paul Debevec's knee MRI. A full model of the knee was rotated and translated in three dimensions, then the original and rotated versions were cropped so the majority of the object was the same but the edges contained different parts of the knee; some portions were rotated out of the box while others were rotated in. You can see in Figure 11, for example, that the top right edge of the knee is visible in the left column but not in the right column. Similarly, variations in the tibia are visible at the bottom edge in the right column but not in the left. Successful registration was achieved in 18 seconds; results are shown in Table 6.

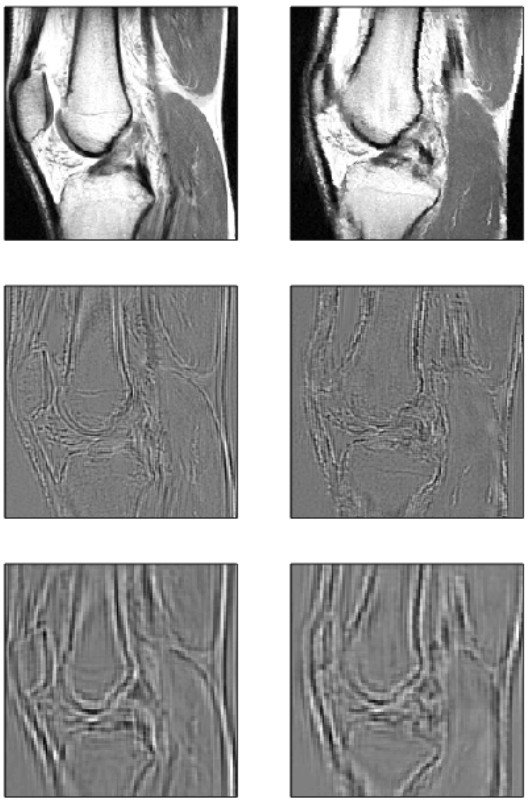

Fig. 11 Occlusion and clutter. The top row is the original object. The middle and bottom rows are the reconstructed wavelet coefficients at the second and third levels of resolution respectively. The left column is model A which has been cropped. The right column is model B, which was rotated before cropping.

5.5 Temporal

The final example consists of two MRI scans taken of Bruce Gooch at different times, one on June 5 and one on July 13. The other examples in this section are concocted by rotating and translating an object then comparing to a copy of its original self; therefore, the rotation and translation are known exactly. In this example, we are comparing two different objects of unknown initial orientation.

For the first test, we move the June 5 scan through a large rotation, then attempt registration with the July 13 scan. Results are shown in Table 6. The resulting rotation and translation are not exactly equal to the induced ones; in fact, this is correct. It can be seen in Figure 12 that the post-alignment registration is improved over the initial slightly unaligned version.

For the second test, the June 5 and July 13 scans are registered as is, with no induced rotation or translation. Results are shown in Table 6. Note that the recommended rotation and translation are very close to the difference between the induced and recommended rotation and translation in the first test.

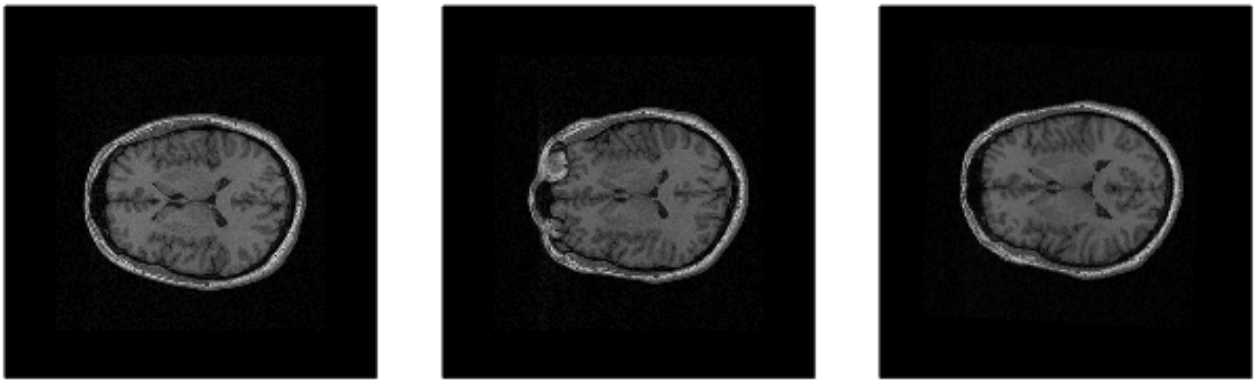

Fig. 12 Temporal brain MRI scan comparison. The left image was taken on July 13. The center image was taken on June 5. The right hand image is the June 5 image after a large rotation is induced then removed through registration with the July 13 image; note that the alignment is improved after registration despite the large initially induced angle. For example, the top of the eye is no longer visible after registration.

6 Conclusions and Recommendations

We have demonstrated an algorithm to conduct rigid registration of voxelized three-dimensional objects using the wavelet transform. The method performs registration of objects that are grossly misaligned and brings them closely into alignment. Global registration is successfully achieved without need of initial alignment information. Further fine registration may be conducted at this point using elastic registration methods or other fine alignment methods such as ICP.

We explored the effects of variations in the algorithm such as choice of feature point type and quantity, rotation representation, and voting scheme. Choices are explained and quantified.

We presented several applications and demonstrated efficacy on both fully three dimensional models in which the intensity varies throughout and on surface models in which the interior is a constant value. The algorithm is robust to noise, occlusion and clutter, and can operate on models that vary in time.

The method is as fast or faster than other geometric feature registration methods and is significantly faster than voxel intensity registration methods, thus making its use as a pre-processing step for voxel intensity methods feasible. It operates on voxelized objects, which many geometric feature registration methods do not, and solves the problem of most voxel intensity methods in that it achieves global registration without settling into a local minimum.

The directions in which this research could go next are numerous and interesting. A partial matching application could allow the alignment of smaller data sets to larger ones. Adding scale variation could allow the alignment of higher resolution data sets in areas of interest. Within the algorithm itself, the exploration of different size filters or different shape wavelets that still meet the DTCWT criteria is possible and may lead to improvements in precisely locating feature points. Cross modal registration would allow the comparison of objects from different types of scans, such as MRI and CT data for medical applications or acoustical and optical data for security applications such as mine detection.

Acknowledgements This work was supported in part by the National Defense Science and Engineering Graduate (NDSEG) Fellowship to the first author, by U.S. National Science Foundation grant DMI-062933 and by the SMART Center for Environmental Sensing and Modeling (CENSAM) sponsored by the Singapore National Research Foundation. The authors thank Drs. D. C. Gossard, V. Goyal, W. Cho, and M. K. Reuter for their comments on earlier versions of this work.

References

1. Chalfant JS, Patrikalakis NM (2008) Three-dimensional object registration using the wavelet transform. In: Proceedings of the Spring Conference on Computer Graphics, pp 196–203
2. Besl PJ, McKay ND (1992) A method for registration of 3D shapes. IEEE T Pattern Anal 14(2):239–256
3. Chen Y, Medioni G (1991) Object modeling by registration of multiple range images. In: Proceedings of the IEEE International Conference on Robotics and Automation, Sacramento, California, pp 2724–2729
4. Zhang Z (1994) Iterative point matching for registration of free-form curves and surfaces. Int J Comput Vis 13(2):119–152
5. Pajdla T, Van Gool L (1995) Matching of 3-D curves using semi-differential invariants. In: 5th In-

ternational Conference on Computer Vision, Cambridge, MA, pp 390–395
6. Barequet G, Sharir M (1997) Partial surface and volume matching in three dimensions. IEEE T Pattern Anal 19(9):929–948
7. Sun Y, Paik J, Koschan A, Page DL, Abidi MA (2003) Point fingerprint: A new 3-D object representation scheme. IEEE T Syst Man Cy B 33(4):712–717
8. Ko KH, Maekawa T, Patrikalakis NM (2003) An algorithm for optimal free-form object matching. Comput Aided Design 35(10):913–923
9. Mokhtarian F, Khalili N, Yuen P (2001) Multi-scale free-form 3D object recognition using 3D models. Image Vision Comput 19(5):271–281
10. Ko KH, Maekawa T, Patrikalakis NM, Masuda H, Wolter FE (2003) Shape intrinsic properties for free-form object matching. J Comput Inf Sci Eng 3(4):325–333
11. Dey TK, Giesen J, Goswami S (2003) Shape segmentation and matching with flow discretization. In: Dehne F, Sack JR, Smid M (eds) Proceedings of the 8th International Workshop on Algorithms and Data Structures (WADS), Ottawa, Canada, Lect Notes Comput Sc, vol 2748, pp 25–36
12. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110
13. Urschler M, Bauer J, Ditt H, Bischof H (2006) SIFT and shape context for feature-based nonlinear registration of thoracic CT images. Comput Vis Appr Med Image Anal 4241:73–84
14. Cole-Rhodes AA, Johnson KL, LeMoigne J, Zavorin I (2003) Multiresolution registration of remote sensing imagery by optimization of mutual information using a stochastic gradient. IEEE T Image Process 12:1495–1511
15. Fookes C, Bennamoun M (2003) Rigid medical image registration and its association with mutual information. Int J Pattern Recogn 17(7):1167–1206
16. Maes F, Vandermeulen D, Suetens P (2003) Medical image registration using mutual information. P IEEE 91:1699–1722
17. Hastreiter P, Ertl T (1998) Integrated registration and visualization of medical image data. In: Proceedings of Computer Graphics International (CGI), IEEE, Hannover, Germany, pp 78–85
18. Wang F, Vemuri B, Rao M, Chen Y (2003) A new & robust information theoretic measure and its application to image alignment. In: Taylor C, Noble JA (eds) Proceedings of the Conference on Information Processing in Medical Imaging : IPMI03, Ambleside, UK, Lect Notes Comput Sc, vol 2732, pp 388–400
19. Xu R, Chen YW (2007) Wavelet-based multiresolution medical image registration strategy combining mutual information with spatial information. Int J Innov Comput Inf 3:285–296
20. Maintz JBA, Viergever MA (1998) A survey of medical image registration. Med Image Anal 2:1–36
21. Addison PS (2002) The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance. Institute of Physics Publishing, Bristol
22. Daubechies I (1992) Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania
23. Stollnitz EJ, DeRose TD, Salesin DH (1996) Wavelets for Computer Graphics. Morgan Kaufmann Publishers, Inc., San Francisco, California
24. Strang G, Nguyen T (1996) Wavelets and Filter Banks. Wellesley-Cambridge Press, Wellesley, Massachusetts
25. de Zeeuw PM (2002) A toolbox for the lifting scheme on quincunx grids (lisq). Centrum voor Wiskunde en Informatica (CWI) Reports of: Probability, Networks and Algorithms (PNA) 24:1–23
26. Karasaridis A, Simoncelli EP (1996) A filter design technique for steerable pyramid image transforms. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, IEEE, Atlanta, GA, pp 2389–2392
27. Simoncelli EP, Freeman W (1995) The steerable pyramid: A flexible architecture for multiscale derivative computation. In: Proceedings of the International Conference on Image Processing (ICIP), IEEE, Washington, DC, pp 444–447
28. Hill PR, Bull DR, Canagarajah CN (2000) Rotationally invariant texture features using the dual-tree complex wavelet transform. In: Proceedings of the International Conference on Image Processing (ICIP), IEEE, Vancouver, BC, Canada, pp 901–904
29. Van De Ville D, Blu T, Unser M (2005) Isotropic polyharmonic B-splines: Scaling functions and wavelets. IEEE T Image Process 14(11):1798–1813
30. Tymczak CJ, Niklasson AMN, Röder H (2002) Separable and nonseparable multiwavelets in multiple dimensions. J Comput Phys 175(2):363–397
31. Kingsbury N (1999) Image processing with complex wavelets. Philos T R Soc A 357:2543–2560
32. Kingsbury N (2001) Complex wavelets for shift invariant analysis and filtering of signals. Appl Computat Harmon A 10(3):234–253
33. Kingsbury NG (2003) Design of Q-shift complex wavelets for image processing using frequency

domain energy minimisation. In: Proceedings of the IEEE Conference on Image Processing (ICP), Barcelona, paper 1199
34. Horn BKP (1987) Closed-form solution of absolute orientation using unit quaternions. J Opt Soc Am A 4(4):629–642
35. Ballard DH (1981) Generalizing the Hough transform to detect arbitrary shapes. Pattern Recogn 13(2):111–122
36. Stockman G (1987) Object recognition and localization via pose clustering. Comput Vision Graph 40:361–387
37. Jenkinson M, Smith S (2001) A global optimisation method for robust affine registration of brain images. Med Image Anal 5:143–156
38. Scott DW (1985) Averaged shifted histograms: Effective nonparametric density estimators in several dimensions. Ann Stat 13(3):1024–1040
39. Shimazaki H, Shinomoto S (2007) A method for selecting the bin size of a time histogram. Neural Comput 19:1503–1527
40. Nooruddin FS, Turk G (2003) Simplification and repair of polygonal models using volumetric techniques. IEEE T Vis Comput Gr 9:191–205
41. Rabiner LR, Rader CM (1972) Digital Signal Processing. IEEE Press, New York