ESTIMATION OF LINEAR DEFORMATIONS OF 3D OBJECTS

Attila Tanács [1], Joakim Lindblad [2], Nataša Sladoje [3], Zoltan Kato [1]

[1]Dept. of Image Processing and Computer Graphics, University of Szeged, P.O. Box 652, 6701 Szeged, Hungary
[2]Centre for Image Analysis, Swedish University of Agricultural Sciences, Uppsala, Sweden,
[3]Faculty of Technical Sciences, University of Novi Sad, Serbia

ABSTRACT

We propose a registration method to find affine transformations between 3D objects by constructing and solving an overdetermined system of polynomial equations. We utilize voxel coverage information for more precise object boundary description. An iterative solution enables us to easily adjust the method to recover *e.g.* rigid-body and similarity transformations. Synthetic tests show the advantage of the voxel coverage representation, and reveal the robustness properties of our method against different types of segmentation errors. The method is tested on a real medical CT volume.

Index Terms— 3D registration, affine transformation, system of polynomial equations, voxel coverage

1. INTRODUCTION

3D imaging in medical and industrial applications is common nowadays. Taking 3D images of the same or similar objects at different times raises the problem of *registration*, *i.e.* establishing the geometric correspondence between these images. Many approaches have been proposed for a wide range of problems in the past decades [1].

Classical methods solve the registration problem by either extracting *geometric features* or using the *image intensities* directly, and try to establish correspondences by usually applying an iterative technique. Geometric features can be *e.g.* points, surfaces [2] or skeletons. Speed is always an important factor; on-line registration may be required, e.g. during surgery, for registering pre-operative volumes to intra-operative ones. Zhang *et al.* give an overview of surface based registration techniques and propose a 15 times faster method than standard Iterative Closest Point (ICP) methods [2]. However, it still takes around one minute to register object models segmented from high resolution CT images. Intensity similarity methods are used mainly for non-binary single- and multi-modality medical registration problems, but can also be applied for registration of binary images [3]. Since the iterative search in these methods uses the features/intensities in each step, they suffer from considerably increased computational complexity in case of large amount of data. A direct method of Burel *et al.* uses spherical harmonics to recover the orientational differences between surfaces of 3D objects [4]. It provides fast registration in case of *rigid-body* problems.

In this paper we propose the extension of our 2D affine method [5] to 3D objects. The extension is not trivial; the corresponding polynomial system of equations consists of more equations and its exact solution does not exist, least-squares solution is found by a Levenberg-Marquardt solver. The system is generated by a single pass over the image, no correspondences are required, thus the solution does not depend on the size of the objects. This makes our method especially suitable for registering large volume images. Voxel coverage information further improves the registration accuracy.

2. REGISTRATION FRAMEWORK

Let us denote the object points of the *template* and the *observation* volume images by $\mathbf{x}, \mathbf{y} \in \mathbb{P}^3$, respectively in the projective space. Let $\mathbf{A}$ denote the unknown, non-singular 4×4 homogeneous matrix of the affine transformation that we want to recover. It relates the template and observation as

$$\mathbf{A}\mathbf{x} = \mathbf{y} \quad \Leftrightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{y}.$$

The above equations still hold when a properly chosen function $\omega : \mathbb{P}^3 \to \mathbb{P}^3$ is acting on both sides of the equations [6]:

$$\omega(\mathbf{A}\mathbf{x}) = \omega(\mathbf{y}) \quad \Leftrightarrow \quad \omega(\mathbf{x}) = \omega(\mathbf{A}^{-1}\mathbf{y}). \tag{1}$$

In order to avoid the need for point correspondences, we integrate over the foreground domains $\mathcal{F}_t$ and $\mathcal{F}_o$ of the *template* and the *observation*, respectively, yielding

$$|\mathbf{A}| \int_{\mathcal{F}_t} \omega(\mathbf{x})\, d\mathbf{x} = \int_{\mathcal{F}_o} \omega(\mathbf{A}^{-1}\mathbf{y})\, d\mathbf{y}\,, \quad \text{and} \tag{2}$$

$$\int_{\mathcal{F}_t} \omega(\mathbf{A}\mathbf{x})\, d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathcal{F}_o} \omega(\mathbf{y})\, d\mathbf{y}. \tag{3}$$

Authors from University of Szeged are supported by the Hungarian Scientific Research Fund (OTKA) Grant No. K75637. This research was also partially supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency.

N. Sladoje and J. Lindblad are financially supported by the Ministry of Science of the Republic of Serbia through Projects ON144018 and ON144029 of MI-SANU.

We thank N. Cornea for providing us some of the synthetic objects, and Á. Perényi and A. Palkó for providing a set of CT studies.

The Jacobian of the transformation is $|\mathbf{A}| = \int_{\mathcal{F}_o} d\mathbf{y} / \int_{\mathcal{F}_t} d\mathbf{x}$.

A 3D affine transformation is determined by 12 parameters, thus we need at least 12 equations. The basic idea of the proposed approach is to generate sufficiently many linearly independent equations by making use of the relations in Eq. (2)–(3). To achieve this goal, we select polynomial ω functions such that their k-th coordinate is of the form $\omega(\mathbf{x})^{(k)}_{f,g,h} = x_1^f \cdot x_2^g \cdot x_3^h$, where $f, g, h \in \mathbb{N}$, $f + g + h = d$, and $d \in \{1, 2, 3\}$. From Eq. (2) these functions generate the following polynomial equations:

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_a \, d\mathbf{x} = \sum_{i=1}^{4} q_{ai} \int_{\mathcal{F}_o} y_i \, d\mathbf{y} \, , \tag{4}$$

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_a x_b \, d\mathbf{x} = \sum_{i=1}^{4} \sum_{j=1}^{4} q_{ai} q_{bj} \int_{\mathcal{F}_o} y_i y_j \, d\mathbf{y} \, , \tag{5}$$

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_a x_b x_c \, d\mathbf{x} = \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{4} q_{ai} q_{bj} q_{ck} \int_{\mathcal{F}_o} y_i y_j y_k \, d\mathbf{y} \tag{6}$$

where $1 \le a, b, c \le 3$, $a \le b \le c$, and q_{ij} denote the unknown elements of the inverse transformation $\mathbf{A}^{-1}$. This gives $3 + 6 + 10 = 19$ equations. In order to increase numerical stability, we add another 19 similar equations using Eq. (3), *i.e.* by changing the role of the point sets $\mathbf{x}$ and $\mathbf{y}$. Note, that this step introduces no new unknown parameters, since $\mathbf{A}$ is uniquely determined by parameters q_{ij} in the non-singular case. The system of equations is up to third order and is overdetermined.

2.1. Voxel coverage object representation

Digital image space provides only limited precision, thus the integrals in Eq. (4)–(6) can only be *approximated* by a discrete sum over the voxels. To maximally utilize the information available in the data, we propose to use voxel coverage representations of the objects, where the value of a voxel is set to be proportional to its coverage by the imaged object. It was proved that such representation provides higher precision of feature estimates compared to binary representations at the same spacial resolutions [7]. Details about this type of representations can be found in [8].

Similarly as in [5], we are interested in high precision estimates of geometric moments, considering that the coefficients of the system of equations in Eq. (4)–(6) are (continuous) geometric moments of the *template* and *observation*. Analogously as in [7], the geometric moments of order $i + j + k$ of a continuous 3D object, *i.e.* the integrals in Eq. (4)–(6), are approximated by

$$\int_{\mathcal{F}_t} x_1^i x_2^j x_3^k \, d\mathbf{x} \approx \sum_{\mathbf{x} \in X_t} \mu_t(\mathbf{x}) \, x_1^i x_2^j x_3^k \, , \tag{7}$$

where $\mu_t(\mathbf{x})$ is the partial coverage of voxel $\mathbf{x}$ in the template image. Similar approximation holds for $\mathcal{F}_o$ using coverage μ_o. The Jacobian is approximated as

$$|\mathbf{A}| = \frac{\sum_{\mathbf{y} \in X_o} \mu_o(\mathbf{y})}{\sum_{\mathbf{x} \in X_t} \mu_t(\mathbf{x})} . \tag{8}$$

X_t and X_o are the discrete domains of the template and observation image, respectively. The approximating discrete system of polynomial equations can now be produced by inserting these approximations into Eq. (4)–(6).

Estimations of the coverage values μ can be obtained from a voxel coverage segmentation, such as *e.g.*, the 3D extension of the method presented in [8].

3. NUMERICAL SOLUTION

The system of polynomial equations of Eq. (4)–(6) can be solved in the least-squares sense (LSE) using a standard Levenberg-Marquardt (LM) method. Compared to methods that provide a closed form solution, a drawback of the iterative LM method is that it may get stuck in local minima and usually takes longer running time. On the other hand it has the advantage of being able to give a (local) solution even when the algebraic error of the system of equations is well above zero.

To increase numerical stability, the object coordinates are mapped into $[-0.5, 0.5]$ by applying normalizing transformations $\mathbf{N}_t$ and $\mathbf{N}_o$ (scaling and translations) before registration. Also notice that all unknowns are outside of the integrals in Eq. (4)–(6), thus integrals have to be evaluated only once. Its time complexity is $\mathcal{O}(N)$, where N is the number of the voxels of the object, since all the summations can be computed in a single pass over the volume image.

We found that the initial rotational parameters have a strong influence on the registration result. To reduce the risk of getting trapped in a sub-optimal local minimum, the LM solver is started from 27 different positions, corresponding to rotations of 120° around each spatial axis. Scaling, shearing and translational parameters are set to their identity values. When started near the optimal rotation, the algebraic error decreases fast, allowing to terminate the search after a few iterations; we terminate each initial search after 20 iterations. Having found the orientation that provided the smallest error, we continue with a full search. If, for any of the starting orientations, after 20 iterations, an algebraic error below a given threshold is found, checking other starting orientations is not necessary. We set this threshold to a constant value of 100. These parameters were determined by experimental tests.

A general transformation $\mathbf{A}$ may contain reflections too, which is undesirable in many practical applications. To avoid reflections we can impose the following condition during the search: if $|\mathbf{A}| < 0$, *i.e.* $\mathbf{A}$ contains reflections, then we assign a high value as algebraic error (10^{50} was used).

Since $\mathbf{A}$ can also define geometric transformations of lower degrees of freedom than affine, we can easily construct

methods for such, *e.g.* rigid-body or similarity problems. When computing the algebraic error, the 12 necessary parameters can be derived from the parameters of the transformation with a lower degree of freedom (*e.g.* shearing and scaling is set to identity for rigid-body transformations).

Algorithm 1 summarizes the steps of our proposed method.

Algorithm 1: Affine registration of 3D objects

Input: Template and observation volume images.
Output: Transformation parameters $\mathbf{A}$.

Step 1 Extract foreground voxels and apply normalizing transformations $\mathbf{N}_t$, $\mathbf{N}_o$
Step 2 Construct system of equations using Eq. (7)–(8)
Step 3 Least-squares solution
- Initialize LM solver (27 orientations, 20 iterations)
- At each iteration compute *algebraic error* E
  Apply constraints to the parameters if applicable
  If $E < 100$ is obtained, then skip other orientations
- $\mathbf{A}^* =$ Transformation parameters given by a full search starting from the best orientation

Step 4 Unnormalize: $\mathbf{A} = \mathbf{N}_o^{-1} \cdot \mathbf{A}^* \cdot \mathbf{N}_t$

4. EXPERIMENTAL RESULTS

The performance of the proposed algorithm is quantitatively evaluated on a database of 3D volume objects. The dataset consists of 15 different objects and their transformed versions, a total of 1500 objects. The objects typically consist of 200000–2 million voxels. The transformation parameters are randomly selected in the following ranges: Rotation angle: $[0, 2\pi)$; Scale factor: $[0.5, 1.5]$; Shear: $[-1, 1]$; Translation vector magnitude: $[0, 1]$ (larger translations are irrelevant since coordinates are transformed into $[-0.5, 0.5]$ before registration).

For each template, we apply 100 transformations, of which 25 are rigid-body, 25 with non-uniform scaling, and 50 full affine. Fig. 1 shows some examples from this database.

In order to quantitatively evaluate the results, we use the following two error measures:

$$\epsilon = \frac{1}{|T|} \sum_{\mathbf{p} \in T} \|(\mathbf{A} - \widehat{\mathbf{A}})\mathbf{p}\|, \quad \text{and} \quad \delta = \frac{|R \triangle O|}{|R| + |O|} \cdot 100\%,$$

where $\triangle$ denotes symmetric difference, while T, R and O are the sets of voxels of the *template*, *registered* object and *observation* respectively. The algorithm is implemented in Matlab 7.7 and is run on a desktop computer using Intel Core2 Duo processor at 2.4 GHz.

Voxel coverage representations of the observation objects are generated by using $n \times n \times n$ ($n \in \{1, 2, 4, 8\}$) supersampling of the voxels close to the object boundary and the coverage is approximated by the fraction of sub-voxels that fall inside the object. Before computing the δ errors the objects are binarized. Registration accuracies are shown in Table 1.

Fig. 1. Examples from the image database: template objects (top) and their affine deformed observations (bottom).

Table 1. Median error values for different supersampling levels n. The percentage of the registrations where the δ error is above 1% and 10% ($\approx$ visually acceptable) is also shown.

n	ϵ	δ	$\delta > 1\%$	$\delta > 10\%$	Time (sec)
1	0.0361	0.1555	10.67	2.13	1.54
2	0.0108	0.0627	3.13	2.27	1.56
4	0.0069	0.0470	2.47	2.20	1.54
8	0.0065	0.0402	2.47	2.20	1.52

The computing time does not include the time to build the system of equations, which currently takes around 1.5–2 seconds (can be further optimized). The results show that already the binary representation gives excellent results. However, even the lowest level of voxel coverage information brings clear improvement. Above $n = 4$ levels of supersampling we see little change. The computing time is independent of n.

In practice, images are always corrupted by various types of error, *e.g.* missing data, segmentation error or occlusion. To test the robustness of our method, we synthetically generate these types of errors on binary images. For the missing data case, a given percentage of the voxels are randomly removed. To simulate segmentation error, we randomly modified the observation boundary by adding or removing small clusters of voxels on the object boundary. Finally, to simulate occlusion, we randomly occluded a specified amount of foreground voxels. Table 2 shows the registration results.

Table 2. Registration errors induced by imaging errors

	Missing vxls.		Segm. error		Occlusion	
	50%	90%	5%	25%	1%	2%
ϵ (voxels)	0.22	0.65	0.30	1.79	1.77	3.82
$\delta(\%)$	0.83	2.20	1.15	5.76	5.22	9.92
$\delta > 5\%$	7.3%	25%	7.4%	61%	53%	84%
$\delta > 10\%$	3.3%	7.6%	2.8%	28%	23%	49%
Time (sec)	2.0	3.4	3.7	6.6	5.3	5.8

Uniformly removed object points cause no problems in general, even for 90% the results are acceptable. In case of segmentation errors, at 25% degradation level more than 70%

of the cases still provided good alignment. As any area-based method, our approach seems to be sensitive to occlusion, however. Above 1–2% of degradation, the results become unreliable. Notice that, since the algebraic error is higher for higher degradation levels, more starting orientations are taken into account which leads to increased computing time. We conclude that our method performs well as long as the geometric moments of the objects do not change dramatically.

4.1. Comparison with mutual information

We compare our results against a classic intensity similarity method based on mutual information (MI) utilizing a multiresolution pyramid scheme [3], that is adopted to binary objects. The capture range of MI is narrow, thus at a lower pyramid level the optimization is started from the same 27 orientations as our method. The optimal result is propagated to finer pyramid levels. Based on 200 rigid-body registrations, our method clearly outperforms MI. The average computation time of MI exceeds 2 minutes, compared to few seconds of our method. The median of δ errors are 0.42 for MI and 0.05 for our method. Additionally, in 40% of the cases MI provides poor or failed registrations (*i.e.* $\delta > 10\%$). Our method gives excellent results, for all cases $\delta < 1.14\%$.

4.2. Experiments with real CT volume images

We also test our method on objects extracted from real medical data. CT images of the pelvic area of the same person were acquired at different times. The spatial resolution of the CT is $0.8 \times 0.8 \times 5\,\text{mm}$, and the segmented objects contain 400–500 thousand voxels. Bone regions from the images are segmented by thresholding and removal of unwanted components.

The main challenges are poor image resolution, segmentation errors, and slightly different placement of the femural head. We use the rigid-body restriction in our method and since the orientations of the objects are close to each other, it is enough to use only one initial orientation. The construction of the system of equations takes around half a second, the optimization around 0.2 second. Visual inspection confirms satisfactory registration results and indicates applicability of the method on real medical data (see Fig. 2).

5. CONCLUSIONS

In this paper we propose a registration method to recover various linear transformations between 3D objects. The solution is obtained by solving a polynomial system of equations. The construction of the system of equations has linear time complexity, the LSE solution can be controlled by carefully selecting initial parameters, and is independent of the size of the data. Thus, registration problems of large volume images can be solved fast and efficiently. The method proves to be robust against various types of errors and shows applicability in medical image registration of bone structures. The experimental evaluation confirms that utilizing partial voxel coverage information enhances the registration.

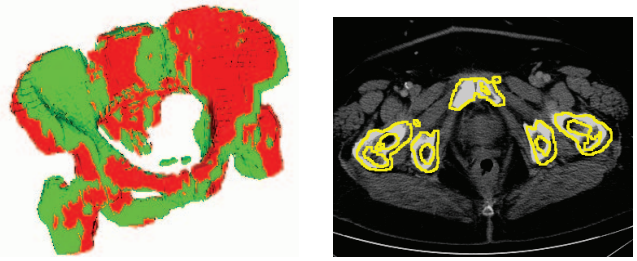

Fig. 2. Registration of real CT data: superimposed registered 3D bone models (left), and bone contours of the registered template (yellow) overlayed on a CT slice of the observation.

6. REFERENCES

[1] B. Zitová and J. Flusser, "Image registration methods: A survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[2] J. Zhang, Y. Ge, S. H. Ong, C. K. Chui, S. H. Teoh, and C. H. Yan, "Rapid surface registration of 3D volumes using a neural network approach," *Image and Vision Computing*, vol. 26, no. 2, pp. 201–210, 2008.

[3] J.P.W. Pluim, J.B.A. Maintz, and M.A. Viergever, "Mutual-information-based registration of medical images: a survey," *Medical Imaging*, vol. 22, no. 8, pp. 986–1004, 2003.

[4] G. Burel, H. Henocq, and J.-Y. Catros, "Registration of 3D objects using linear algebra," in *CVRMed*, 1995, pp. 252–256.

[5] A. Tanács, C. Domokos, N. Sladoje, J. Lindblad, and Z. Kato, "Recovering affine deformations of fuzzy shapes," in *Proceedings of SCIA*. 2009, vol. 5575 of *LNCS*, pp. 735–744, Springer.

[6] C. Domokos and Z. Kato, "Parametric estimation of affine deformations of planar shapes," *Pattern Recognition*, vol. 43, no. 3, pp. 569–578, 2010.

[7] N. Sladoje and J. Lindblad, "Estimation of moments of digitized objects with fuzzy borders," in *Proceedings of ICIAP*. 2005, vol. 3617 of *LNCS*, pp. 188–195, Springer.

[8] N. Sladoje and J. Lindblad, "Pixel coverage segmentation for improved feature estimation," in *Proceedings of ICIAP*. 2009, vol. 5716 of *LNCS*, pp. 929–938, Springer.