Automatic Estimation of 3D Transformations using Skeletons for Object Alignment

Tao Wang and Anup Basu

Department of Computing Science, University of Alberta, Edmonton, AB T6G 2E8, Email: {taowang, anup}@cs.ualberta.ca

Abstract

An algorithm for automatic estimation of 3D transformations between two objects is presented in this paper. Skeletons of the 3D objects are created with a fully parallel thinning algorithm and feature point pairs (land markers) are extracted from skeletons automatically, and least squares method is applied to solve an over-determined linear system to estimate the 3D transformation. Experiments show that this method works quite well with high accuracy when the translations and rotation angles are small, even when there is some noise. The estimation process requires about 900 ms on an Intel Centrino Laptop with 512 MB memory, for a complex model with about 37,000 object points and 500 object points for its skeletons.

1. Introduction

3D alignment or registration algorithms have many applications such as object matching and medical image processing. Consider two objects *O1* and *O2*, such that *O2=M*O1*, where *M* is the 3D transformation matrix. In this context, the objective of 3D alignment or registration is to estimate the 3D transformation matrix *M*.

Survey [8] of 3D alignment or registration methods is available in the literature. 3D alignment or registration algorithms are classified into *extrinsic* and *intrinsic* methods. *Extrinsic* algorithms rely on artificial objects (markers) attached to patients. Algorithms in this category are usually automatic and fast because the 3D transformation matrix can be computed explicitly. One drawback of extrinsic method is the invasive procedures are needed to place the markers. *Intrinsic* methods depend only on images generated from patients. Alignments are based on some salient point *land markers*, or on measures calculated from the gray scale values of images (*voxel property based*). *Land markers* are salient points that can be located on the object. The advantage of this technique is that the set of land markers is sparse compared to the original object, so that the optimization procedure is relatively fast. Optimization measures compute the mean square distance (MSD) [9] between each land marker and its closest counterpart. Segmentation is a necessary step for algorithms in this category. The main drawback is that user interaction is usually required for identifying the land markers. *Voxel property based* methods operate directly on the gray scales of images without segmentation. The advantage is that it can utilize all of the available image information so that it can be used in many different applications. However, this process is usually computationally expensive.

Our application is in 3D medical image processing. We are interested in defining and tracking volume changes of airways caused by surgery, such as removal of tonsils in children, which increases volume. Doctors need to track the volume changes of airways for administering effective treatments. In our application, the doctors can take scans of patients and save them in a DICOM [6] format. They use commercial software such as ScanIP/FE [7] to segment the airways and remove noises semi-automatically. The first stage of our research is to build up a model to depict the segmented airway. Since airways have an elongated or tree-like structure, we found in the literature that 3D skeletons [2-4] can provide a good description of it. The skeletons are created using a fully parallel 3D thinning algorithm, which improved Ma and Sonka's algorithm [3] to preserve connectivity of 3D objects. The second stage of our research is to align two models for comparison. We noticed that different scans of the same patient may have different orientations. These differences are not very significant. However, small misalignments can cause big mistakes. This is the motivation behind our work. In our experiments, we applied a heuristic algorithm to automatically extract feature point pairs (land markers) from skeletons. These feature point pairs are used to estimate the 3D transformation matrix in real time for a small part of the upper airway model with about 37,000 object points and 500 object points for its skeletons. The algorithm has 5 main steps: segmentation (off-line, semi-automatic), noise removal (off-line, automatic), thinning (off-line, automatic), finding feature point pairs (on-line and in real-time, automatic) and transformation estimation (on-line and in real-time, automatic).

In the following sections, the method and results will be described in detail. In Section 2, we will introduce the data collection and segmentation procedures. Section 3 will briefly discuss how to artificially create data for comparison. The fully parallel 3D thinning algorithm and the skeletonization process is summarized in Section 4. Section 5 focuses on the acquisition of feature point pairs (land markers), and the method to estimate the 3D transformation matrix. Results of experiments are presented in Section 6, before the work is concluded in Section 7.

2. Data collection, segmentation and noise removal

A small potion of the upper airway is scanned for a child and saved in the DICOM [6] format, which is the industry standard for medical images. In our experiment, we used ScanIP/FE [7] to segment airway semi- automatically from the DICOM format scans, remove the noise, and create an airway model in 3D images. A *3D image* is a mapping that assigns the value of 0 or 1 to each point in the 3D space. Points having the value of 1 are called *black (object)* points, while 0's are called *white (background)* ones. Black points form objects of the image. Our test data has about 37,000 object points. Figure 1 shows some image slices and the segmented airway (in red). The 3D models from different viewpoints are displayed in Figure 2.

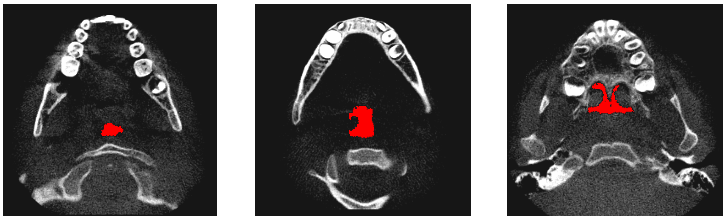

Figure 1: Some image slices and segmented airway (in red color) (Left) The first image slice (Mid) A center slice (Right) The last slice

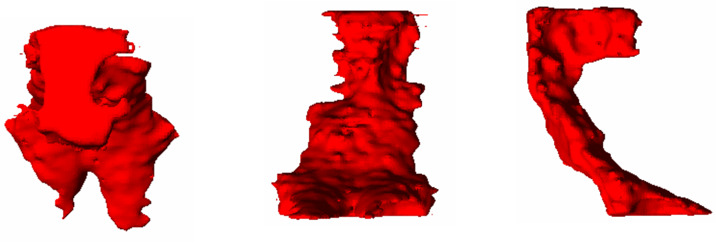

Figure 2: 3D airway model from 3 different points of view

3. Data creation for comparison

To verify our methodology, we artificially created some airway models for comparison. We applied small 3D rotations and translations to the 3D image of airway by Homogeneous Transformation [1]. We use (a , b , g , Δx , Δy , Δz) to represent the rotation angles and translations for x-, y- and z-axis of the Homogeneous Matrices. And we also added some noise to it, to create a new model for comparison.

4. Fully parallel 3D thinning algorithm and skeletonization

3D thinning for medial lines or surface approximation is a useful approach for many potential applications. The approach has been extensively researched in the last decade [2-4]. Among all 3D thinning and medial lines/surfaces approximation algorithms, 3D parallel thinning algorithms [2-4] have many advantages over others. Typically, they study a local neighborhood (3*3*3 or 5*5*5) of voxels in a parallel fashion, and iteratively change *object points* that satisfy some masks to *background points*. The first advantage is that they are quite efficient. Second, they are easy to understand and implement. Third, some parallel thinning algorithms [2, 4] had been proved to preserve the connectivity of 3D objects, which is the one of the most important concerns in a thinning algorithm. However, some constraints of this technique prevent it from being widely applied. The first constraint is that most parallel thinning algorithms are sensitive to rotations. Second, if the input 3D image is not well composed [5], it is impossible to guarantee that thinning algorithms given by some masks can produce unit width structure, which is required in some applications. This paper, however, will show how to apply this technique to build up a model to represent airways, and to estimate the 3D transformation matrix in real-time, even though the technique has some drawbacks.

In the experiment, we applied our fully parallel 3D thinning algorithm to extract the skeletons from 3D images after a noise removal step. Our algorithm is adapted from Ma and Sonka's algorithm [3], which is the only fully parallel 3D thinning algorithm in literature. It has a number of advantages. However, we found that it cannot preserve connectivity of 3D objects. Ashutosh *et al* [11] also found this problem. We improved [3] by changing some masks in Class D to preserve connectivity of 3D objects.

The skeletons are used to represent the airway models. The skeletons have about 500 object points, a large reduction from the 37,000 object points.

Figure 3 shows the skeletons of the original model and the artificially created model with (a , b , g , Δx , Δy , Δz) = (1, 1, 1, 0, 0, 0). We can know from the skeletons that:

1. The thinning algorithm is sensitive to rotations. Even when the rotation angles are so small (1 degree in each direction), the two skeletons look quite different.
2. The thinning algorithm cannot provide unit-width structures. We can see that some regions on the skeletons are quite dense.

However, we will show that these two drawbacks of the thinning algorithm do little harm to the estimation of a 3D transformation matrix. It means that we can still estimate the transformation matrix precisely based on the thinning algorithm.

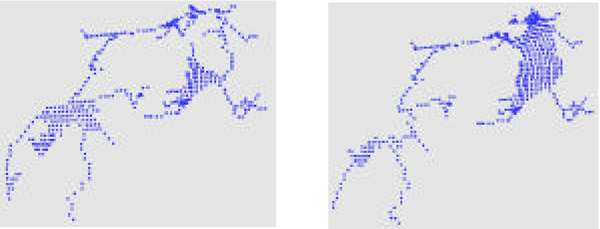

Figure 3: (Left) The skeleton of the original model. (Right) The skeleton of the transformed model with (1, 1, 1, 0, 0, 0)

5 Feature point pair acquisition and estimation the 3D transformation matrix

In general, the estimation of 3D transformation matrix is quite difficult and time-consuming [8, 9]. For our application, it is not possible to put landmarks on the airways of patients when taking scans. Our tool needs to be improved in the future. The parallel 3D thinning algorithm has two constrains as we mentioned above. One is that it is sensitive to rotation. The other is that it cannot generate unit-width structures. The first constraint can be relaxed because we only deal with objects having similar orientations. But it seems to be difficult to get feature point pairs from the skeletons because of the second constraint, especially for some regions with dense object points. However, this problem is not as difficult as it looks. In this section, we will show how to use a heuristic algorithm to get all the feature point pairs fully automatically from the skeletons in real time.

Definition 1: Connectivity number

*If P is an object point in a 3D image, the connectivity number of P is the number of object points except itself in P's (3 * 3 * 3) neighborhood.* In Figure 4, the connectivity number of *P* is 4.

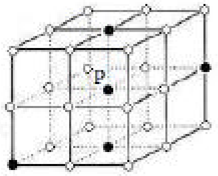

Figure 4: Connectivity number of point *P* is 4. A " ● " is an object point. A " ○ " is a background point.

Definition 2: Line point

A line point is an object point with connectivity number equal to 2.

In Figure 5, the line points are displayed in red color.

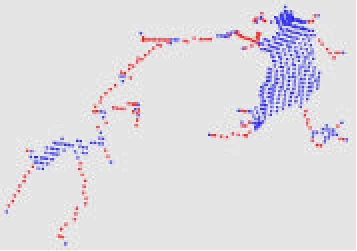

Figure 5: Line points (in red color)

Definition 3: Feature point (land marker) candidate

*A feature point (land marker) candidate is a line point with at least one non- line point in its (3 * 3 * 3) neighborhood.*

Definition 4: Feature point pair (land marker)

*I1 and I2 are two 3D image of the same object. I2=M*I1, where M is a 3D transformation matrix. A feature point pair (land marker) contains two object points P1 and P2, where P1 is on the skeleton of 3D image I1 and P2 is on the skeleton of 3D image I2, and P2=M*P1.*

5.1 The over-determined linear system

A 3D transformation can be represented by a 3 * 3 matrix *M*. For point *P1*= $(x1\ y1\ z1)^T$ in 3D space, *P2* = *M* * *P1*= $(x2\ y2\ z2)^T$. The goal of our work is to estimate the 9 variables of matrix *M* in certain conditions. Since we need to calculate 9 variables, mathematically, we need 9 equations for this linear system. For each feature point pair, we can have 3 equations for x, y, and z. So we need 3 feature point pairs in total to solve this problem. However, in most cases in our experiment, we can find more than 3 feature point pairs. This is a typical case of an over-determined linear system. An over-determined linear system is a well-solved problem [10].

5.2 Heuristic rule and algorithm

In our experiment, a heuristic rule is used to identify the feature point pairs. It requires two points in a feature point pair to have similar "configurations" in its 8 sub-neighborhood. That is, the local topologies of two points in a feature point pair should be similar. In the experiment, we used an adaptive method to search the best neighborhood scale.

Heuristic Rule:

If two individual object points on different skeletons have the same line-point number in all 8 sub-neighborhoods, these two points form a feature point pair.

The algorithm to find feature point pairs is described as follows:

INPUT: *S1* (skeleton of 3D image *I1*), *S2* (skeleton of 3D image *I2*)

OUTPUT: Estimation of 3D transformation matrix *M*

ALGORITHM

```
Load the S1 and S2
FOR (NB=MIN; NB<=MAX; NB+=2)
1.  Calculate the feature point candidate sets C1 from S1, C2 from S2,
    within current neighborhood scale NB
2.  Select feature point pairs P=(P1, P2) with the heuristic rule, where
    $P1 \subseteq C1$ and $P2 \subseteq C2$. If P has less than 3 pairs of points,
    CONTINUE
3.  Use least square method to solve the over-determined linear system
    with P, and get the 3D transformation matrix M
4.  Create a new point set P3, where P3=M*P1
5.  Calculate the mean square distance (MSD) between P2 and P3
END FOR
The estimate of the 3D transformation is the M with smallest MSD
```

In our experiments, MIN=3 and MAX=21. It should be noticed that in the skeleton representation, the number of object points in a given neighborhood is quite small, even though the neighborhood is large (for instance, *NB*=21). And we use the mean square distance (*MSD*) [9] as the metric. Our algorithm is implemented in C++. It can estimate the 3D transformation matrix in about 900 ms on an Intel Centrino Laptop with 512 MB memory. The test data has about 37,000 object points and its skeleton has 500 object points or so.

Figure 6 shows the feature points (in red) in the original 3D image and their pairs in the test case $(a, b, g, \Delta x, \Delta y, \Delta z)$ = (1, 1, 1, 0, 0, 0). We can see that each feature point pair is located very precisely. The estimated and the real transformation matrix are

$$\begin{bmatrix} 1.0011 & -0.0224 & -0.0167 \\ 0.0081 & 0.9980 & -0.0172 \\ 0.0163 & 0.0124 & 0.9998 \end{bmatrix} \quad and \quad \begin{bmatrix} 0.9997 & -0.0174 & -0.0175 \\ 0.0171 & 0.9997 & -0.0174 \\ 0.0178 & 0.0171 & 0.9997 \end{bmatrix}$$

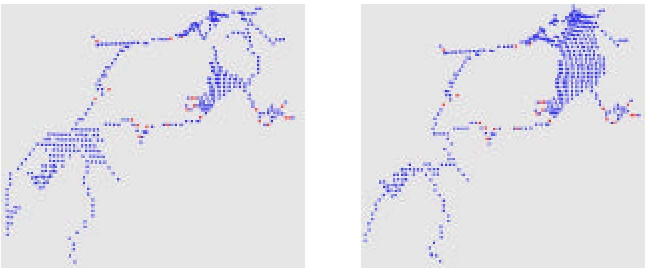

Figure 6: Feature points (in red) in the original 3D image and in test case $(a, b, g, \Delta x, \Delta y, \Delta z)$ = (1, 1, 1, 0, 0, 0)

6. Experiments and the results

Our experiment has three steps. First of all, we extract the feature point pairs (*P1, P2*) from skeletons *S1* of 3D image *I1* and *S2* of 3D image *I2*, and estimate the 3D transformation matrix *M* with smallest *E*. Then, we create a new 3D image *I3*, where *I3=M*I1*. In the last step, we compute the *MSD* between *I3* and *I2*, as the metric. $(a, b, g, \Delta x, \Delta y, \Delta z)$ is represented by (alpha, beta, gama, delta_x, delta_y, delta_z) in the figures.

We first applied our algorithm to estimate the 3D translations. There are three test sets in our experiments: Translation in x-axis only, Translations in x- and y-axis, and Translations in x-, y- and z-axis. All the translations are positive integers in [0, 5]. Figure 7 shows the *MSD* for 3D translations. Result shows that the estimates are quite precise. Even when $(a, b, g, \Delta x, \Delta y, \Delta z)$ = (0, 0, 0, 5, 5, 5), the *MSD* is only about 0.26, which means the mean distance between each object point in the translated image *I2* and estimated image *I3* is only 0.26 (better than half-voxel precise). In fact, if we increase the *NB* parameter, we can estimate larger translations precisely. We notice that when a = 5, the *MSD* is about 0.15 and when $a + b$ =5, the MSD is very close to 0.00. That is because for the second case, we randomly generated some a s and b s and their sum is 5. And for most cases (for instance, a =2 and b =3), the MSD is very close to 0.00. So that the average MSD for $a + b$ =5 is very close to 0.00. This is true for all the experiments.

Next, we used our algorithm to estimate the 3D rotations. There are three test sets in our experiment: Rotations in x-axis only, Rotations in x- and y-axis, and Rotations in x-, y- and z-axis. All the rotation angles are positive integers in [0, 5]. Figure 8 shows the *MSD* for 3D rotations. Result shows that when the rotation angles are smaller than 4 degrees ($a \le 3$, $a + b \le 6$ and $a + b + g \le 9$), the estimations are quite accurate (*MSD* ≤ 1.0) with small rotation angles. When the rotation angles increase, the performance decreases dramatically because the 3D thinning process is sensitive to rotations. Thus, the feature point pairs matching algorithm does not work well, and this leads to large estimation errors.

We also applied our method to estimate the combinations of translations and rotations with some random noises (Figure 9). All the translations and rotation angles are positive integers in [0, 5]. The random noise is in the range of [0%, 10%]. When there is no noise, we can achieve reasonable performance (*MSD* ≤ 1.0) when $a + b + g + \Delta x + \Delta y + \Delta z \le 19$. When the noise is not very significant (1% and 2%), we can achieve performance with *MSD* ≤ 1.0 when $a + b + g + \Delta x + \Delta y + \Delta z \le 15$. With larger noise (5% and 10%), we can achieve performance with *MSD* ≤ 1.0 when $a + b + g + \Delta x + \Delta y + \Delta z \le 12$. This result means the algorithm can estimate the 3D transformations with small rotation angles and small translations accurately in noisy conditions.

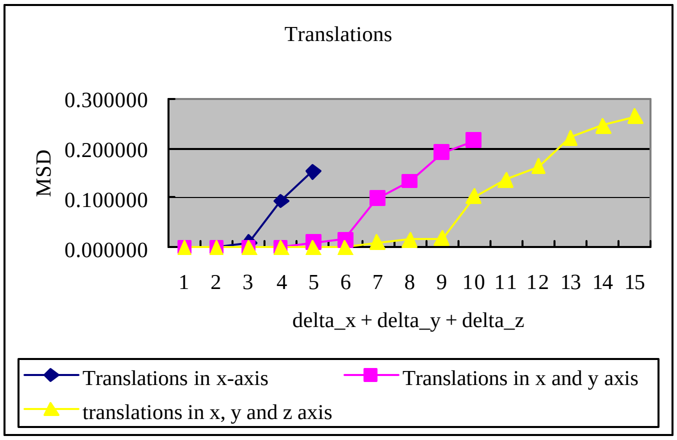

Figure 7: MSD for 3D translations

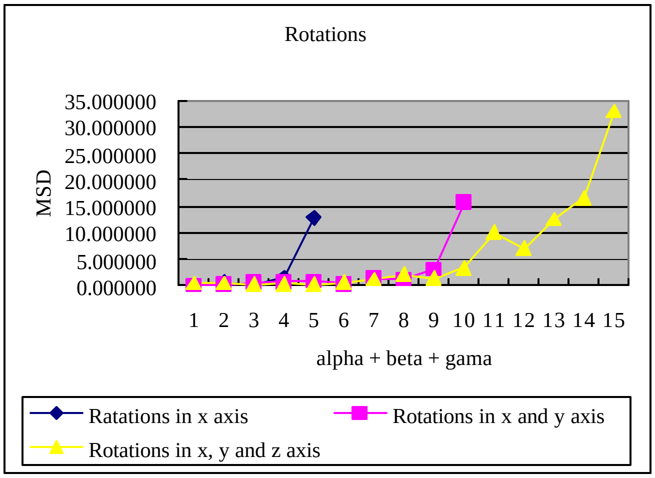

Figure 8: MSD for 3D rotations

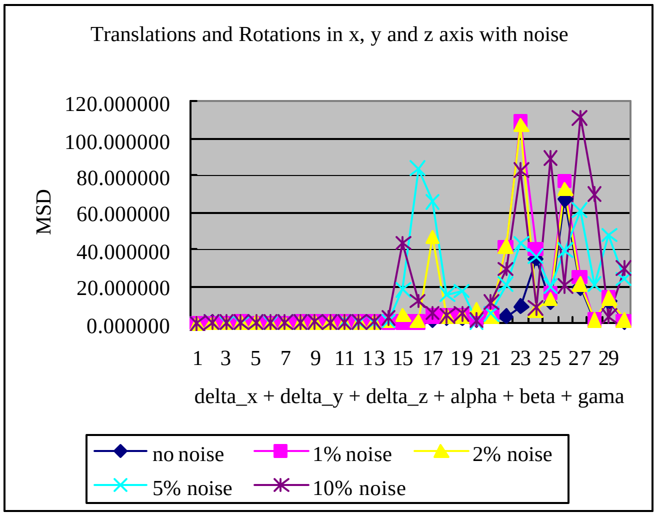

Figure 9: MSD for 3D translations and rotations with random noise

7. Conclusions and feature work

In this paper, we demonstrated how to use 3D thinning technique to extract the skeletons of the segmented airway model. Then, some feature point pairs were automatically extracted from the skeletons as land markers. Finally, we used these feature point pairs to estimate the 3D transformation matrix. Result shows that this method works quite well when the differences in orientation are small.

To the best of our knowledge, this is the first attempt to estimate 3D transformations using the skeleton representation. This method is novel, fast, and accuracy is good for small rotations and translations. The approach has a number of advantages. First of all, the skeleton is a compact representation of 3D objects, which makes fast optimization possible. Second, the feature point pair detection and estimation procedures are automatic and real-time. This algorithm is an *intrinsic* method depending on land markers. It avoids the invasive procedures to place the markers, which is necessary for *extrinsic* methods. And it runs faster than the *voxel property based intrinsic* methods. Also, it is different from most *land markers based intrinsic* algorithms in that it can find land markers automatically without user interactions. The third advantage is that no initial information is needed for estimation. Result shows that this method works quite well in real time when the objects for comparison only having small differences in orientations. The algorithm fails when the differences are large due to the limitation of the 3D parallel thinning algorithm.

This work is a preliminary approach for defining and tracking the volume changes of airways with surgery. There are a number of improvements that need to be made in the feature. First, since the parallel thinning algorithm is sensitive to rotations, we will study and apply some dstance-based skeletonization or voronoi-skeletonization methods which are rotation-invariant. Secondly, because of constraints on time, we do not have airway model after surgery. Therefore, we had to create some models artificially to test our algorithm. In the near feature, we will obtain real models after surgery and then compare our results using before and after data. Third, we assumed rigid transformation in the modeling. We will extend our algorithm to deal with non-rigid transformation.

Acknowledgements

The authors thank Manuel Langravere and Yi Fang for pre-processing the airway data, Dr. Irene Cheng and Meghna Singh for discussion on 3D alignment algorithms, Dr. Paul Major for providing the test data, and Lihnag Ying and Xiaodong Wen for helpful discussions and assistance in program debugging.

References

[1] D. Hearn and M. P. Barker, Computer Graphics (second edition), Pearson Education.

[2] K. Palagyi, A. Kuba, A 3D 6-subiteration thinning algorithm for extracting medial lines, Pattern Recognition Letters, 19 (7), May 1998, pp. 613-627.

[3] C. M. Ma, M. Sonka, A fully parallel 3D thinning algorithm and its applications, Computer Vision And Image Understanding, 64 (3), November 1996, pp. 420-433.

[4] K. Palagyi, A. Kuba, A parallel 3D 12-subiteration thinning algorithm, Graphical Models and Image Processing, 61 (4), July 1999, pp. 199-221.

[5] L. J. Latecki, 3D well-composed pictures, Graphical Models and Image Processing 59 (3), May 1997, pp. 164-172.

[6] http://www.sph.sc.edu/comd/rorden/dicom.html

[7] http://www.simpleware.com/software.php

[8] J.B.A. Maintz and M.A. Viergever, A Survey of Medical Image Registration, Medical Image Analysis, vol.2, pp.1-36, 1998

[9] P. J. Besl and N. D. Mckay, A method for registration of 3D shapes, IEEE Trans. Pattern Anal. Machine Intell. 14(2), 239–256 (1992)

[10] J. J. Leader, Numerical analysis and scientific computation, Pearson Addison Wesley, Boston, 2004.

[11] A. Chaturvedi, Z. Lee, Three-dimensional segmentation and skeletonization to build an airway tree data structure for small animals, 50 (7), April 2005, Physics in Medicine and Biology, pp. 1405-1419.