

# COMPARISON OF TWO DISTANCE BASED ALIGNMENT METHOD IN MEDICAL IMAGING

G. Bulan, C. Ozturk

Institute of Biomedical Engineering, Bogazici University, Istanbul, Turkey

**Abstract**—A k-dimensional (k-d) tree based alignment and its comparison with the standard distance map based alignment is presented. We first describe a brief outline of both distance based iterative alignment algorithms. The new k-d based technique uses modified Approximate Nearest Neighbor (ANN) Library, which is designed for both exact and approximate nearest neighbor searching in multidimensional space. We performed self-alignment tests of the k-d tree based alignment and compared two different alignment methods using a large 3D dataset of a rodent brain. The results indicate that the k-d based image alignment is highly effective, accurate, reliable, and provides compatible errors with the distance map based alignment method. On the other hand, as a big advantage, k-d tree alignment requires significantly less virtual or physical memory; a critical issue for large datasets.

**Keywords** – Registration, distance map, k-d tree, alignment.

## I. INTRODUCTION

There are two techniques discussed in this paper: distance map based alignment and k-dimensional (k-d) tree based alignment. The aim of both alignment algorithms is to compute the optimum parameters for a three-dimensional rotation and translation, which aligns one set of data points (test object) with another (reference object). Both methods can be very helpful to register large datasets of contours or surfaces, commonly encountered in medical imaging. They do not require special ordering or preprocessing of the data.

K-d tree based and distance map based methods perform multidimensional image alignment based on geometrical matching. The algorithm has mainly two parts: 1) k-d tree based or distance map based distance calculation, 2) Marquardt-Levenberg (M-L) algorithm for optimization. The two parts are complementary to each other. First part provides the distances and derivatives, and M-L algorithm efficiently minimizes the sum of the square distances. Both methods differ only on the initial stage: formation of the k-d tree or distance map using distance transformation.

## II. METHODOLOGY

A rigid body transformation in three dimensions can be written as  $p \in T(p, \Pi)$ , where  $p$  refers to coordinates of image point and  $\Pi$  are the transformation parameters. The shortest distances,  $d(p)$ , between the transformed points of the test object and reference object points are used during the iterative alignment. The sum of squares of distances indicates the goodness of the alignment [1]:

$$e(\Pi) = \sum_{i=1}^N d^2(T(p_i, \Pi)) \quad (1)$$

The best-fit transformation parameters are calculated by minimizing  $e(\Pi)$  over transformation parameters using Marquardt-Levenberg (M-L) algorithm for minimization [1]. The M-L method computes the rotation and translation parameters, which minimizes the sum of the squared distances between transformed and reference points. Each point in the test set is transformed by rotation and translation, and the algorithm iterates until the magnitude of rotation and translation are each less than a threshold. The distances that are used in equation 1 obtained differently for the two methods and will be explained next.

The distance map alignment algorithm operates with voxel-based descriptions of the object and the use of the vector valued Euclidean distance function [1]. The distance map is in the form of a three-dimensional array, whose entries are three-dimensional vectors from the closest points in the object to the given voxel [1]. The array dimensions have to be larger than the reference objects' dimensions. Distance map is calculated once at the beginning of the alignment. Later it is used as a look-up table to find the distance to the closest reference point for test points, each time they are transformed into a new position in space.

The k-d tree is data structure that is based on a recursive subdivision of the universe into subspaces by disjoint hyper-rectangular regions, called cells [2]. Each node of the tree is associated with such region, called box and is associated with a set of data points that lie within this box. The root node of the tree is linked to a bounding box that contains all the data points. As long as the number of data points related to this node is greater than a small quantity, called bucket size, the box split into two boxes by an axis-orthogonal hyper-plane that intersects this box.

Approximate Nearest Neighbor (ANN), which is a library of C++ objects and procedures [3], includes exact and approximate nearest neighbor algorithms, which supports two methods for standard tree-ordered search and priority search [4]. In this study, this ANN library has been applied to the alignment problem via an interface that transfers the closest points, their indices and distance vector values from ANN library into the alignment package written in Matlab. The k-d tree enables efficient resolve of the closest point [5], therefore, it is shown to significantly reduce the computation required [6].

Both distance map alignment programs were run on a PC Pentium-III/ 866 MHz computer with 128 Megabytes of RAM. The algorithms were tested by aligning data sets of

## Report Documentation Page

<b>Report Date</b> 25OCT2001	<b>Report Type</b> N/A	<b>Dates Covered (from... to)</b> -
<b>Title and Subtitle</b> Comparison of Two Distance Based Alignment Method in Medical Imaging	<b>Contract Number</b>	
	<b>Grant Number</b>	
	<b>Program Element Number</b>	
<b>Author(s)</b>	<b>Project Number</b>	
	<b>Task Number</b>	
	<b>Work Unit Number</b>	
<b>Performing Organization Name(s) and Address(es)</b> Department of Radiology Duke University Medical Center, NC	<b>Performing Organization Report Number</b>	
<b>Sponsoring/Monitoring Agency Name(s) and Address(es)</b> US Army Research, Development & Standardization Group (UK) PSC 802 Box 15 FPO AE 09499-1500	<b>Sponsor/Monitor's Acronym(s)</b>	
	<b>Sponsor/Monitor's Report Number(s)</b>	
<b>Distribution/Availability Statement</b> Approved for public release, distribution unlimited		
<b>Supplementary Notes</b> Papers from the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 25-28 OCT 2001, Istanbul, Turkey. See also ADM001351 for entire conference on cd-rom.		
<b>Abstract</b>		
<b>Subject Terms</b>		
<b>Report Classification</b> unclassified	<b>Classification of this page</b> unclassified	
<b>Classification of Abstract</b> unclassified	<b>Limitation of Abstract</b> UU	
<b>Number of Pages</b> 3		

mouse brain images (~700 000 points). The distance transform for this dataset are stored in a 320×220×210 binary voxel array. (320\*220\*210\*3\*2 bytes are needed for the distance map, approximately 89 Mbytes)

For the alignments, the test object was calculated by taking random  $N$  points from the reference object, transforming them with a known rotation and translation and converting coordinates of transformed points to voxel values.

The alignment accuracy was measured by two parameters. They are actual error and residual error. The actual error is defined as,

$$\hat{a}_m = \frac{1}{N} \sum_{i=1}^N \|p'_i - a_i\| \quad (2)$$

where  $p_i$  is the location  $i$ -th test point after alignment and  $a_i$  is the actual location of the this point in the reference object. Actual error evaluation can be computed when the true locations of the test points are known. If we align unknown objects, we use another measure of error, known as residual error. It is defined as the average value of the distance function, evaluated over the transformed test points,

$$\hat{a}_r = \frac{1}{N} \sum_{i=1}^N d(p'_i) \quad (3)$$

where  $p_i$  is the location of the  $i$ -th test point after matching, and  $d(\cdot)$  is the distance of those points to the reference object.

### III. RESULTS

We first examined the residual and actual errors as we changed the number of test points ( $N$ ) for both alignment methods. The results for k-d tree based alignment is given in Table I, for distance map based alignment in Table II. These results are obtained using the following translation and rotation parameters:  $T_x=20$ ,  $T_y=-30$ ,  $T_z=10$ ,  $\theta_x=0.3$ ,  $\theta_y=0.4$ ,  $\theta_z=0.5$ . The stopping criteria used by the algorithm determines the accuracy and running time. The alignment algorithm for both methods was stopped at the condition of whenever greater translations change in  $x$ ,  $y$ , and  $z$  during iteration was less than 0.1 pixel spacing, and greater rotation change was less than 0.003 radians.

Monte Carlo multiple alignment trials were performed also for different number of test points, but with the various random values of initial test object transformations. The range of translations was  $-20 \leq T_x, T_y, T_z \leq 20$  pixels. The range of angles was  $-0.4 \leq \theta_x, \theta_y, \theta_z \leq 0.4$  radians. The trials for each value of  $N$  were done 100 times. Actual error statistical events are listed on Table III and IV for k-d tree and distance map based alignment respectively.

TABLE I  
K-D TREE ALIGNMENT RESULTS WITH THE VARYING NUMBER OF TEST POINTS

Point Number	Performance Parameters			
	Actual Error	Residual Error	Iteration Number	Time (s)
20,536	0.053	0.053	12	62.5
10,268	0.038	0.038	11	28.78
5134	0.056	0.056	9	11.04
2567	0.135	0.135	9	4.39
1284	0.004	0.004	9	2.26
642	0.058	0.058	8	0.99
321	0.039	0.039	7	0.50
161	0.113	0.113	9	0.33
81	0.011	0.011	8	0.22
41	0.049	0.049	12	0.22
21	0.109	0.109	12	0.17
11	31.145	28.1449	12	0.22

TABLE II  
DISTANCE MAP ALIGNMENT RESULTS WITH VARYING NUMBER OF TEST POINTS

Point Number	Performance Parameters			
	Actual Error	Residual Error	Iteration Number	Time (s)
20,536	0.319	0.0	15	65.75
10,268	0.295	0.0	19	47.23
5134	0.503	0.0	16	34.05
2567	0.334	0.0	18	25.92
1284	0.215	0.0	17	20.76
642	0.115	0.0	12	19.50
321	0.452	0.0	12	18.95
161	0.669	0.463	15	19.17
81	0.775	0.442	14	19.88
41	0.702	0.473	19	19.67
21	0.781	0.476	16	21.42
11	17.21	15.379	16	21.64

TABLE III  
FOR K-D TREE ALIGNMENT, ACTUAL ERROR STATISTICAL RESULTS OBTAINED BY MULTIPLE ALIGNMENT TRIALS WITH VARYING NUMBER OF TEST POINTS

Point Number	Statistical Parameters			
	Average Distance	Minimum Distance ( $10^5$ )	Maximum Distance	Standard Deviation
10,268	0.0827	0.084	0.358	0.024
5134	0.073	0.47	0.286	0.019
2567	0.059	0.05	0.295	0.016
1284	0.059	0.31	0.333	0.015
642	0.187	3.1	63.27	0.085
321	0.538	2.6	96.68	0.537
161	3.164	8.5	89.78	1.437
81	4.667	111.3	92.03	2.095

TABLE IV  
FOR DISTANCE MAP ALIGNMENT, ACTUAL ERROR STATISTICAL RESULTS OBTAINED BY MULTIPLE ALIGNMENT TRIALS WITH VARYING NUMBER OF TEST POINTS

Point Number	Statistical Parameters			
	Average Distance	Minimum Distance	Maximum Distance	Standard Deviation
10,268	0.082	0.0	0.72	0.020
5134	0.081	0.0	0.70	0.203
2567	0.146	0.0	0.88	0.247
1284	0.211	0.0	0.91	0.396
642	0.412	0.0	1.03	0.821
321	0.301	0.0	1.05	0.386
161	2.873	0.0	119.21	1.632
81	3.436	0.0	158.24	1.757

#### IV. DISCUSSION

For distance map alignment, some of the residual errors in Table II and IV are zero because of rounding which is inevitable in voxel description of the space. The round off error of distance map alignment are not present in the k-d tree based methodology. Actual errors provide a better measure for the individual evaluation and comparison of the algorithms. Actual errors of kd tree methodology are less than distance map alignment. If we compare the results for N=1284 of Table III and IV; the actual error obtained by distance map alignment algorithm is about 0.2 pixels with standard deviation (over trials) of about 0.4 pixels and maximum error is about 0.9. For k-d tree based methodology, the actual error is about 0.06 pixels with the standard deviation (over trials) of about 0.015 pixels, and maximum error is about 0.33. The performance of the kd tree based algorithm is better than distance map alignment.

All tables show that large number of test points result in higher accuracy of the alignment. Total running time for distance map based alignment is the sum of times for vector distance calculation and optimization. In general, running times of kd tree based and distance map alignments are comparable. This has to be kept in perspective because demands on excessive RAM, and subsequent use of disk cache might slow the distance map based alignment for larger datasets.

Performance parameters, which are accuracy and running time, are also dependent on local minima in the alignment function. The local minima effect is more pronounced for small test number alignments. The poor performance occurs for both alignment methods, as we decrease the number of test points.

The distance map alignment creates distance map in the size of original reference data and it requires usually more physical and virtual memory allocation than kd tree. The time consuming effect of distance map alignment limiting its application occurs during the large number of data sets' alignment.

#### V. CONCLUSION

We have implemented a k-d tree based alignment method and showed its effectiveness in three-dimensional objects of natural data. The comparison of two methods indicates that k-d tree based alignment gives better results without the need for large storage spaces for distance maps. Since the kd based methodology includes ANN Library, it can be also applied to higher, more than three dimensions. Therefore, the k-d tree methodology should be the preferred choice in this type of alignment procedures.

We are currently running detailed noise sensitivity analysis of both methods and comparing the results of the alignments of rodent brains coming from different animals to each other.

#### REFERENCES

- [1] D. Kozinska, O. Tretiak, J. Nissanov, and C. Ozturk, "Multi-dimensional Alignment Using the Euclidean Distance Transform," in *Graphical Models and Image Processing*, vol 59, No.6, pp. 373-387, 1997.
- [2] D. Pelleg and A. Moore, "Accelerating exact kmeans Algorithms with Geometric Reasoning," in *School of Computer Science, Carnegie Mellon University, Pittsburgh*.
- [3] S. Arya, D. M. Mount and O. Narayan, "Accounting for boundary effects in nearest neighbor searching," *DCG*, vol 16, pp.155-176, 1996.
- [4] S. Arya and M. Mount, "ANN: Library for Approximate Nearest Neighbor Searching," in J. Clerk Maxwell, *ANN Programming Manual*.
- [5] M. Greenspan, G. Guder, J. Talbott, "Acceleration of Binning Nearest Neighbor Methods," in *Vision Interface 2000*, Montreal, Canada, May, pp.14-17, 2000.
- [6] E.T. Baumgartner, P.C. Leger, P.S. Schenker, and T.L. Huntsberg, "Sensor-Fused Navigation and Manipulation from a Planetary Rover," in the *Proceeding of SPIE*, vol. 13523, Boston, MA, 1998.