

A Novel Approach to Automatic Registration of Point Clouds

Rui Liu, Darius Burschka, Gerd Hirzinger
Institute of Robotics and Mechatronics,
German Aerospace Center (DLR)
Muenchner Str. 20, 82234 Wessling, Germany
Rui.Liu@dlr.de

Abstract—For the 3D reconstruction inside historic buildings, we need a marker-free automatic registration approach to align different views together, because GPS does not work indoors and markers are not allowed to paste on the walls. This paper presents an automatic matching process, which employs a novel algorithm, Dynamic Matching Tree technique, for a fast and stable coarse-matching to achieve the automatic pre-alignment of two point clouds and uses modified ICP to do a fine matching efficiently.

The whole process can be divided in the following stages: preprocessing, 2-View matching and N-View matching. To validate our method, various experiments has been done on reconstruction of historic sites and industrial objects.

Keywords—range image registration, segmentation and analysis

I. INTRODUCTION

Matching of multiple views is often addressed in 3D-model generation and is normally a two-stage process consisting of a coarse and a fine matching stage. Coarse matching, that is the pre-alignment of the surfaces for the complex forms, which can be positioned far away from each other in 3D space, is a difficult problem to solve. Fine matching on the other hand can be performed accurately using either the ICP (iterative closest point) [1] method or the least square surface matching [2] method. Nevertheless, ICP involves an iterative solution which consumes much computing time, and it requires models with a considerable degree of overlap at the starting position. This is because it treats the closest point in the other model as the corresponding point and updates the corresponding relationship in each iterative step. If the models have insufficient overlap, ICP will converge to a false result. Consequently, a good coarse matching is a precondition for a successful ICP.

Fully automatic matching of multiple range images is an hard problem for the reconstruction of historic buildings. Because of the demand of high resolution, the processing time of each scan is quite long. Therefore the relative distance between the scans is usually large. Additionally, many different objects appear in views. Due to these difficulties, the existing matching techniques are insufficient for automatic matching of such scans for building reconstruction. Therefore, in industry, the typical solution to this problem is using artificial markers or

involving a manually interactive pre-aligning to solve the coarse matching problem.

We solve this problem by combine dynamic programming with “matching tree” structure [3] to pre-align two scans fast and automatically and uses a modified ICP to do fine-matching with a high accuracy.

Experiments have been performed by reconstructing various historic buildings in Bavaria, Germany, e.g. the Neuschwanstein Castle, the church in Seefeld and the church in Wies, etc. All of the scan data are acquired with the advanced laser scanner of the high tech company Zoller & Froehlich.

This paper is organized as follows. In section 2, previous work is briefly summarized. Then, section 3, describes the details of the automatic matching process, which is divided into three stages: preprocessing, 2-view matching and N-view matching. After this, experimental results are shown in section 4. For the more, the possible improvements in the future are addressed in section 5. Finally, the related acknowledgement is announced in the end section.

II. RELATED WORK

Automatic Matching without an additional tracking system has always been a hot topic in the 3D modelling field. The research can be categorized into: coarse matching and fine matching, two views matching and multiple matching.

Coarse matching, namely pre-aligning, is usually the precondition of fine matching. For single-object views, man normally utilizes “principal axis transformation” [4]. For multi-object views, the objects’ global registration problem must be solved. A possible technique is skeleton based matching [5], which encodes the geometrical and topological information in form of a skeletal. However, the construction of skeleton is sophisticated. “Matching Tree” [3] is a relative simple and effective structure to solve the global maximum matching in 3D space based on local m:n corresponding relationships. It provides a solution in a closed form with polynomial runtime, while the original solution based on interpretation tree is theoretical NP-complete. To reduce the searching space, Rabbini and Heuvel [6] used constrained search for finding correspondent objects. Kang and Zlatanova [7] used corner detector to extract feature points from reflectivity images of

scans and by constructing a triangle mesh to reduce the search space.

Fine matching is a fairly adult field in science. The most robust and frequently used method is ICP (Iterative Closest Point) [1] and numerous variants of it: ICCP (Iterative Closest Compatible Point) [8], ICPIF (Iterative Closest Points using Invariant Features) [9] etc. The basic idea of ICP is treating the nearest point in the other view as correspondent point. SVD (Singular Value Decomposition) is used to deduce the transformation-matrix during iterative steps to align the views to each other. The nearest point can in distance field (original ICP), or in diverse feature field (ICPIF), and by weighted correspondent pair or by reducing the search space (ICCP) to accelerate the convergence. The limit of these ICPs is that they only reach local minima. That is, if most of the nearest points just lie in the false direction of the true correspondent points, they will converge to a false result. This problem can be solved by a reasonable pre-alignment.

Adaptive least square matching [2] is an effective method for matching of 3D surface patches. However, for multi-object scenes, it needs an initial approximation.

Multi-view matching can be solved either incrementally or simultaneously. The interested reader is referred to the article of Cunningham and Stoddart [10], who gave a comparison of three n-view point set registration algorithms.

III. MAIN METHOD

In this section, the proposed method is described in detail. The whole process can be divided into three steps:

- Preprocessing
- 2-View matching
- N-View matching

A. Preprocessing

In this stage, we do a segmentation of each point cloud firstly. And then we use a form-descriptor to generate corresponding segment-pairs.

By the characteristics of the surfaces, the model in one view is segmented into diverse objects. Here, the difference between normal vectors between adjacent points is treated as segmenting criteria. If the difference between the normal vectors of point p and q is fairly large, then the boundary between two objects will be set through here (See Fig. 1).

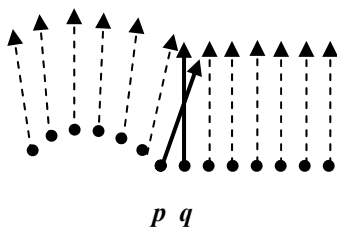


Figure 1. Criterion for segmentation.

The estimation of the normal vector is done by least squares fitting of a tangent plane through the point neighborhood. For the 2.5 D range image, the information from the points can be stored in a matrix, so the access time to the neighborhood is $O(1)$.

The whole matrix will be treated as a graph. Each point in the matrix is like the vertex of the graph, and an edge is constructed if the adjacent points fit the segmenting criteria. By BFS (breadth first search), the whole graph is segmented. The runtime of BFS is $O(m + n)$, in which m is the number of the edges and n is the number of the vertices. Because the edges exist only between adjacent points, the maximal number of edges on a vertex is 8. So the runtime of segmentation is linear, i.e. $O(n)$.

We define a form-descriptor as a two-attribute vector consisting of the sinus value of the angle between the normal vectors and the radius of the surface curvature, and then we generate a list of correspondent segment-pairs with a certain tolerance according to the form-descriptor.

B. 2-View matching

This is a coarse to fine matching process, which consists of two steps: the first is coarse matching, which solves the pre-alignment problem automatically; and the second is fine matching, which aligns two views accurately.

1) Coarse matching

A novel algorithm “Dynamic Matching Tree (DMT)” will be used in this stage. DMT applies the dynamic programming technique to “Matching Tree (MT)” structure. The original implementation of MT had a runtime complexity of $O(n^4)$, where n is the number of corresponding pairs [3]. By use of dynamic programming, DMT reduces the complexity to $O(IJ)$, where I and J are the numbers of feature-objects in two models. It is equivalent to $\Theta(n^{3/2})$.

Matching Tree is designed for 3D global registration out of local correspondent relationships. Every node contains a corresponding pair, which can be valid or invalid. The root contains three non-collinear self-validated nodes. Matching Tree is a tree, and it can be treated as a bipartite matching if we delete its edges, split every tree-node to two object-nodes and add an edge between corresponding object-pair in each valid tree-node (See Fig. 2).

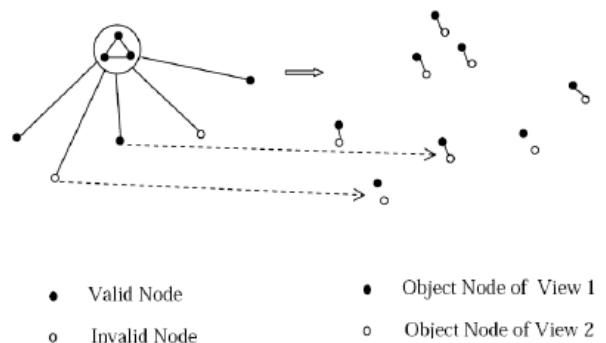


Figure 2. Matching Tree(links) and Matching(rights)

Validation of the none-root-nodes is through comparing the directed relative distances to the root-nodes (See Fig. 3). That means, we need not only to compare the distances from the corresponding pair in this node to the corresponding pairs in the three root-nodes, but also the position of them, because one object-node can be above or under the root-plane with the same distances to the three root-nodes. As showed in Fig. 3, (p, p') is a mirror-pair, which cannot be matched by pure translation and rotation. And (p, p'') is the right corresponding pair. We can check it by comparing the direction of the normal vector formed by the three root-nodes with the direction from the center of the root-nodes (O, O') to the node (p, p').

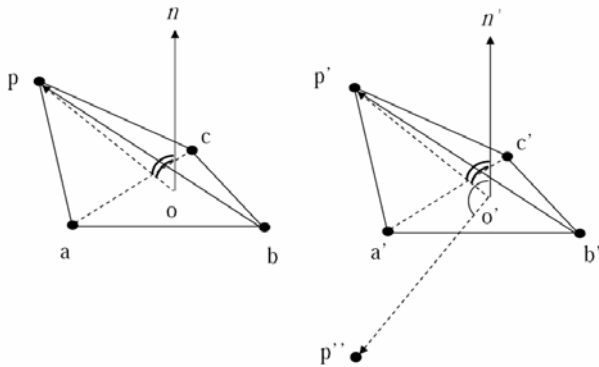


Figure 3. Validation of none-root node

If we assign a matching weight to each node, the goal is to get a matching tree with maximal weight. The basic form to solve the sub-problem of dynamic programming is:

$$M_{i,j} = \max \{M_{i-1,j-1} \oplus w_{i,j}, M_{i-1,j}, M_{i,j-1}\} \quad (1)$$

$M_{i,j}$ is the maximal matching formed by the first i object-nodes (representative points for objects) from the one view $\{X_1, X_2, \dots, X_i\}$ and the first j object-nodes from the other view $\{Y_1, Y_2, \dots, Y_j\}$. And $w_{i,j}$ is the weight of the correspondent pair (X_i, Y_j) . $w_{i,j} = 0$ means, there is no corresponding relationship between X_i and Y_j . The goal of us is to find $M_{I,J}$, if there are I objects in one view and J objects in the other. Then we make a table and file the table from left to right, top to down.

TABLE I. TABLE OF DYNAMIC MATCHING TREE ALGORITHM

X \ Y	0	1	...	j	...	J
0	0	0	...	0	...	0
1	0					
⋮	⋮					
i	0			$M_{i,j}$		⋮
⋮	⋮					
I	0			...		$M_{I,J}$

The value of $M_{i,j}$ is chosen as the maximum from the three items: $M_{i-1,j-1} \oplus w_{i,j}$, $M_{i-1,j}$ and $M_{i,j-1}$. The item $M_{i-1,j-1} \oplus w_{i,j}$ implies, that we should not add the matching weight $w_{i,j}$ of the node pair (X_i, Y_j) to $M_{i-1,j-1}$ directly. Validation to the root-nodes should be done.

If (X_i, Y_j) is accepted by the root of $M_{i-1,j-1}$, we get the equation (2), if

$$M_{i-1,j-1} \oplus w_{i,j} = M_{i-1,j-1} + w_{i,j} \quad (2)$$

Otherwise, we get the equation (3).

$$M_{i-1,j-1} \oplus w_{i,j} = \max \{M_{i-1,j-1}, M'_{i-1,j-1} + w_{i,j}\} \quad (3)$$

A "split-tree" action is involved, if the correspondent pair (X_i, Y_j) is rejected by the root of $M_{i-1,j-1}$. And $M'_{i-1,j-1}$ is the split part from $M_{i-1,j-1}$, which forms a new matching tree with the node (X_i, Y_j) .

In fact, if a node is rejected by the root, it will not be accepted by any three none-collinear valid tree-nodes. Because, these three none-collinear valid tree-nodes form another valid root of this matching tree.

To control the split-tree function in linear time $O(I)$ or $O(J)$, we do a linearity test before the creation of DMT-Table. That means: for each view, we find all 3er-lines and make a pointer from one point to each 3er-line through it. "3er-line" means, there are more than 2 points on this line. Obviously, every point has maximal $(I/2 - 1)$ 3er-lines through it, if there are totally I points in one view. This test has a runtime complexity of $O(I^3)$, which is equivalent to $\Theta(n^{3/2})$.

And we do a split-node action, if the node $P(X_i, Y_j)$ is accepted by the root of tree $M_{i-1,j-1}$ after validation and two nodes on a 3er-line of P are already in tree $M_{i-1,j-1}$. That mean, by adding an accepted node $P(X_i, Y_j)$, we do the following:

If P has 3er-lines $//X_i$ has 3er-lines and Y_j has 3er-lines

Color all tree-nodes of $M_{i-1,j-1}$ green;

For each 3er-line L_P of P do

If two nodes V, W of L_P are green

Split P onto V and W :

$$V_{new} = V + \frac{|WP|}{|VP| + |WP|} \bullet P$$

$$W_{new} = W + \frac{|VP|}{|VP| + |WP|} \bullet P$$

If P was not split

Add P as a tree-node to $M_{i-1,j-1}$.

The above split-node action has runtime complexity of $O(I)$, and it guarantees that there are no three tree-nodes in the

matching tree collinear. So, we can say, if a node is rejected by the root, then it will be not accepted by any three tree-nodes in this matching tree.

So, if a node $P(X_i, Y_j)$ is rejected by the root, we find all the nodes $N(X_k, Y_h)$ of $M_{i-1,j-1}$, which fit the distance constraint of P, that means $|X_i X_k| = |Y_j Y_h|$. And choose the heaviest two from them as the split-part $M'_{i-1,j-1}$, which may form a new tree with P. This is the calculation of the second item $M'_{i-1,j-1} + w_{ij}$ in equation (3). It has a runtime complexity of $O(I)$, for there are maximal $(i-1)$ nodes in matching tree $M_{i-1,j-1}$.

Because the DMT-Table has $(I+1) \times (J+1)$ items and filling of each item cost run time of $O(I)$, the whole runtime complexity of DMT-Algorithm is $O(I^2 J)$, which is equivalent to $\Theta(n^{3/2})$.

And out of the corresponding pairs in maximum matching tree $M_{I,J}$, we use the SVD method to calculation the transformation between two views [3].

2) Fine matching

As the correct corresponding relationships between segments are found after the coarse matching, we can generate an arbitrary number of control-point-pairs by projecting the sampling points of one segment to its correspondent segment. And by the use of iterative actions, the result will be refined.

C. N-View matching

After the fine two-view matching, all the scans are in a common coordinate system of. So we can generate the corresponding point pairs of the multiple views, and make a bundle adjustment to homogenize the whole point clouds.

IV. EXPERIMENTS AND RESULTS

We have verified our method by the reconstruction of various historical buildings and industrial objects. Because of the limit of pages, only two examples are showed below.

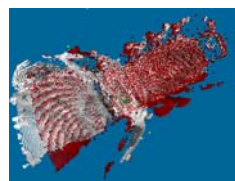
I. Reflectivity Images



II. 3D Point-clouds

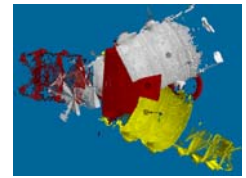


a) before matching

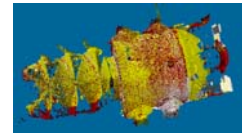


b) after matching

Figure 4. Two scans of Wies-Church



a) before matching



b) after matching

Figure 5. Three scans of Seefeld-Church

V. CONCLUSION

By changing the number of root-nodes, Matching Tree can be adapted for registration in n dimensional space.

ACKNOWLEDGMENT

Thanks to Bernhard Strackenbrock, Johann Heindl and Michael Suppa for the useful discussions, delivery of related input data and testing of our implementations.

REFERENCES

- [1] Z. Zhang, "Iterative Point Matching for Registration of Free-Form Curves and Surfaces," International Journal of Computer Vision 13(2), pp. 119-152, 1994.
- [2] A. Gruen and D. Akca, "Least Squares 3D Surface Matching," in Proc. of 1st Panoramic Photogrammetry Workshop, Dresden, Germany, Feb. 2004.
- [3] R. Liu and G. Hirzinger, "Marker-free Automatic Matching of Range Data," in Proc. of 2nd Panoramic Photogrammetry Workshop, Berlin, Germany, 2005.
- [4] M. Mayr, "Matching zweier Punktwolken zur intraoperativen Patientenregistrierung," Diplomarbeit für Institut für Robotik und Mechatronik, German Aerospace Center (DLR), Wessling, Germany, April 2004.
- [5] H. Sundar, D. Silver, N. Gagvani and S. Dickinson, "Skeleton Based Shape Matching and Retrieval," International Conference on Shape Modeling and Applications 2003, Seoul, Korea. pp. 130, May 2003.
- [6] T. Rabbani, Frank van den Heuvel: Automatic Point Cloud Registration Using Constrained Search for Corresponding Objects, in Proc. of 7th Conference on Optical 3-D Measurement Techniques, Part 1, pp. 177-186, Vienna, 2005.
- [7] Z. Kang and S. Zlatanova: Automatic Registration of Terrestrial Scan Data Based on Matching Corresponding points from Reflectivity Images, in Proc. of Urban Remote Sensing Joint Event, Paris, 2007.
- [8] G. Godin, M. Rioux and R. Baribeau, "Three-dimensional registration using range and intensity information," in Proc. Of SPIE vol.2350, Videometrics III, pp. 279-290, 1994.
- [9] G.C. Sharp, S.W. Lee and D.K. Wehe, "ICP registration using invariant features," IEEE Pattern Analysis and Machine Intelligence, 24(1), pp. 90-102, 2002.
- [10] S. J. Cunningham and A. J. Stoddart, "N-View Point Set Registration: A Comparison," In Proc. of BMVC, pp. 234-244, 1999.