An Efficient Method for Surface Registration

Tomislav Pribanic[1], Yago Diez[2], Sergio Fernandez[2] and Joaquim Salvi[2]
[1]*Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, 10000 Zagreb, Croatia*
[2]*Institute of Informatics and Applications, University of Girona, 17071 Girona, Spain*

Keywords: Surface Registration, 3D Reconstruction, Structured Light, Inertial Sensor.

Abstract: 3D object data acquired from different viewpoints are usually expressed in different spatial coordinate systems where systems' spatial relations are defined by Euclidean transformation parameters: three rotation angles and a translation vector. The computation of those Euclidean parameters is a task of surface registration. In a nutshell all registration methods revolve around two goals: first how to extract the most reliable features for correspondence search between views in order to come up with the set of candidate solutions, secondly how to quickly pinpoint the best, i.e. satisfying, solution. Occasionally some registration method expects also other data, e.g. normal vectors, to be provided besides 3D position data. However, no method assumed the possibility that part of Euclidean parameters could be reliably known in advance. Acknowledging technology advancements we argue that it become relatively convenient to include in 3D reconstruction system some inertial sensor which readily provides info about data orientation. Assuming that such data is provided, we demonstrate a simple, but yet time efficient and accurate registration method.

1 INTRODUCTION

The task of a surface registration is to fuse 3D data originally acquired from different viewpoints (Salvi et al., 2007). To this end it is required to find the 6 degrees of freedom: three rotation angles and three dimensional translation vector that describe a spatial relationship between a pair of viewpoints. An alternative to readily solve the surface registration, using rotating tables and/or robot arms, is neither always available nor a feasible solution due to the number of degrees of freedom of the mechanics and particularly in the case of large surfaces, self-occlusion areas etc. Consequently software based, coarse and fine, surface registration methods have been investigated which operate on 3D data only. The former searches for a good enough initial estimate which is then usually refined by some fine registration method.

The majority of published algorithms, except for few PCA based exceptions (Chung and Lee, 1998), consist of two important phases: first a selection of candidate solutions, and second, the detection of the optimal candidate solution. The intrinsic combinatorial complexity of the problem causes that, for all algorithms, both phases are very likely to be memory and time demanding, as well as sensitive to outliers. Genetic (Santamaría et al., 2011) and RANSAC based algorithms (Diez et al., 2012) are known for its reliability to give eventually a good solution, but at the expense of a substantial computation time which can make them quite impractical. On the other hand, a ready to use PCA fast solutions are additionally sensitive to the object symmetries (otherwise common problem to all other methods too) and a requirement for a large overlap between views. Next, most methods require at least three correspondences between pair of views to define a candidate solution. Even if they seemingly require only a single correspondence (Feldmar and Ayache, 1994) then there is a substantial pre-processing involved during which a feature vector for every point is first found and afterwards used for the extensive comparisons. Finally, some methods (Makadia et al., 2006) expect as input from 3D reconstruction system not only 3D point position data, but also a normal vector in every 3D point. Somewhat surprisingly, it appears that neither method expected a possibility if a part of Euclidean parameters are known in advance, rather they try to solve all six Euclidean transformation unknowns.

In this work we propose a surface registration method assuming that rotation is provided by an inertial sensor (e.g. MTx, 2012), and translation

Pribanic T., Diez Y., Fernandez S. and Salvi J..
An Efficient Method for Surface Registration.
DOI: 10.5220/0004346505000503
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 500-503
ISBN: 978-989-8565-47-1

vector is still left to be found. We justify our assumption by the fact that nowadays technologies have made quite affordable a large pallet of various inertial devices which reliably outputs data about object orientation. This is not only present in popular smart phones and accessories for video games, but there are also smart cameras which have an embedded on-board inertial sensor for orientation detection in 3D space (Shirmohammadi and Taylor, 2011). Nevertheless, it seems that this fact has not been exploited in the context of surface registration.

2 METHOD DEFINTION

The proposed method consists of the following straightforward steps:

1 Given a pair of views **A** and **B** align them partially using the system provided (ideally from some embedded inertial sensors) orientation data (rotation angles).

2. Pick a single point $\mathbf{X_A}$ from view **A** for which a correspondence in the second view exists.

3. Sample (e.g. uniformly) a set of M points $\mathbf{X_{Bi}}$ from the second view **B**, where is typically $M \leq 100$.

4. Construct a candidate translation vector between point $\mathbf{X_A}$ and every point $\boldsymbol{i}$ from $\mathbf{X_{Bi}}$ set. For every translation vector candidate solution, translate views **A** and **B** into common reference system and evaluate the goodness of solution using some merit function. For a further consideration, keep only a single $\mathbf{X_B}$ from $\mathbf{X_{Bi}}$ set for which an extreme of merit function is achieved.

5. Take the closest N points (~100) to $\mathbf{X_B}$ and similar as in step 4, define N candidate translation vectors, for each of them translate **A** to **B** view, evaluate the goodness of potential solution using some merit function. At last, the one for which an extreme of merit function is achieved, take as a final solution.

3 EXPERIMENTS

In this work we have used 3D data from a structured light scanning of mannequin head from several different viewpoints (Figure 1). The white markers set on it (Figure 2) served as control points to compute registration data and are in the subsequent experiments considered as ground truth data. In fact, we use those control points to compute rotation data between views just as if they were provided by some inertial sensor.

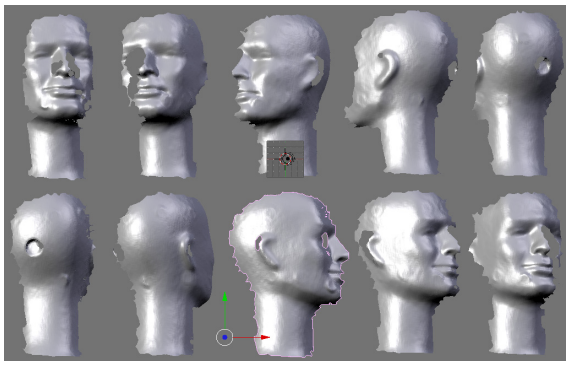

Figure 1: A screen snap from Blender software package where 3D raw data from ten viewpoints were input and surface meshes created.

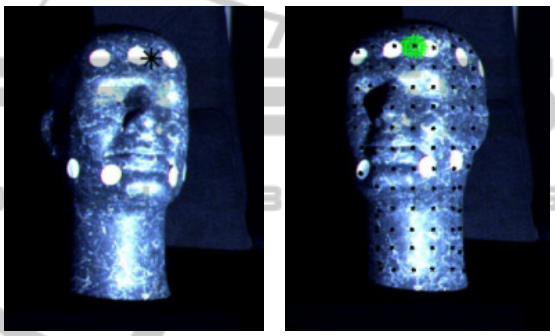

Figure 2: Camera images for a pair of views **A** and **B**. See text for more details.

The left image on Figure 2 is an example of view **A** and the black star on is a point chosen for which a correspondent one is searched in the view **B** (right image). Black dots on the view **B** image represent the actual candidate solutions sampled during step 3 and processed during step 4 of the proposed method. The green area is neighborhood within which a correct solution is, i.e. within which a final solution is searched (step 5).

Figure 3 shows an alignment between two views after applying a rotation. The right (blue) and left (red) point clouds are view **A** and **B** respectively. The black dots on the view **B** point cloud, the green area marked on it and the black star on the view **A** point cloud represent the same points as already introduced for Figure 2, but now in 3D space.

Moreover, black dots on Figure 3 shows that even a sparse consideration of candidate solution allows a pretty good estimate of the final one. Hence, a candidate solution providing an extreme of merit function after step 4 is actually a black dot within a green area. Thus, the green area on Figure 3 shows the neighborhood of points which is further evaluated in order to set a final solution (step five).

Table 1: Comparison of the ground truth data with the initially computed and final results.

View Pair Number of Points	Ground truth data[mm]			Error initial solution[mm]			Error final solution[mm]		
	T_x	T_y	T_z	ΔT_x	ΔT_y	ΔT_z	ΔT_x	ΔT_y	ΔT_z
(33110, 34109)	-75,72	-16,25	-86,57	-6,10	-5,12	-0,06	-0,98	-0,47	-0,38
(34109, 39255)	-119,04	-19,89	-104,45	-2,29	-7,33	0,14	-1,59	-1,97	-1,24
(39255, 43096)	-158,73	-37,78	-194,54	-8,46	1,98	-4,73	-1,36	0,21	-0,87
(43096, 38759)	-111,19	-8,00	-54,98	-4,10	7,40	4,63	-1,60	1,92	2,33
(38759, 35371)	-102,26	-27,49	-156,87	8,88	13,00	-4,64	-3,76	4,36	-3,86
(35371, 39647)	-133,69	-21,36	-104,36	-4,84	-10,29	0,17	-2,47	-4,18	0,44
(39647, 44440)	-133,58	-30,09	-155,76	0,55	-11,91	-4,17	-2,35	-1,27	-1,96
(44440, 41044)	-44,59	-8,93	-59,90	-1,10	5,82	-2,49	-0,10	0,23	0,09
(41044, 36357)	-82,26	-13,36	-65,99	-12,06	-0,17	-2,61	-1,65	0,08	-2,57
	Absolute average error[mm]			5,38	7,00	2,63	1,76	1,63	1,52

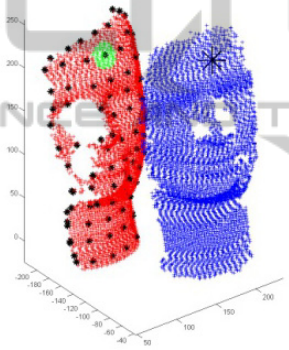

Figure 3: An alignment of 3D point clouds (downsampled for better visualization) from two views after applying a rotation. See text for more detail.

Table 1 is a representative example of extensive experimenting using real 3D noisy data and it demonstrates a very good agreement of ground truth data for translation vector components with the computed values. The central table rows resemble the registration of back of the mannequin head which due to symmetries (Figure 1) is harder to register and consequently a slight decrease in accuracy is witnessed.

Although the speed optimization was out of the scope of this work (Park et. al., 2011), for completeness we stress out that it took about 4 minutes in Matlab to align a pair of views (Intel Core 2 DUO 2.5GHz), where we used all available points (>35000 points, first column Table 1). We point out that such excessive number of points is normally not used in practice (Salvi et al., 2007). When we downsampled our 3D point clouds to be ~6000 points (note this is still a rather large number for registration) per view, we have acquired basically the same results as in Table 1 (not shown here for the lack of space). But in addition, the processing time significantly dropped to only 10-15 seconds per view pair, even using this noticeably suboptimal Matlab code version. Based on this timing and earlier experience, we estimate that even CPU optimized C like code (future work) would allow processing time in matter of seconds for point clouds of size 15000-20000 points.

4 DISCUSSION AND CONCLUSIONS

Unlike other methods the proposed one does not pre-process data in order to first compute certain Euclidean invariants and/or local features around candidate points (Chua, 1997). The proposed method has basically an inherent Euclidean invariant which allows that after rotation alignment is done then all points needs to be moved by the very same translation vector. Since for a some point in view **A** we know that there is its correspondent point in view **B** we simply search for the best candidate in view **B** which minimizes our merit function. However, we do not perform an exhaustive search, typical for some RANSAC based approaches, of all candidate solutions in view **B**. In our particular experiment it was sufficient to try altogether just a couple hundred of candidate solutions to reach a final one. But we note that we did essentially work all the time with all available reconstructed points (~30000 to 40000 points per view). It means that we have tested <1% of possible solutions. At the same time, no initial data downsampling (Trucco et al., 1999) was

performed to make the method actually applicable. Our method uses a point-to-point correspondences strategy, which is generally much simpler than curves or surface correspondence (Wyngaerd, 2002). The merit function used here to estimate candidate solution quality is very simple one: distance point-to-point. We do not require implementation of point-to-plane distances known to be more robust, but also harder to compute (Rusinkiewicz and Levoy, 2001).

As shown the method performs quite well even without removing any outliers (Dalley and Flynn, 2002) and we do not bother with the computing and assigning different weights during processing (Godin et al., 1994). In addition, our method does not require a priory any initial guess for the searched translation vector, but computes accurately even a final solution and therefore can be recognized as both coarse and final method in one. In terms of speed and simplicity our method resembles the character of PCA method, but it is also quite less subtle to the size of overlapping regions. Our method is general purpose one, meaning that to be successfully applied it does not require any typical environment, such as buildings where planar regions and straight lines are expected (Stamos and Leordeanu, 2003). Furthermore, no estimation of certain experimental parameters is needed, as usual in some genetic algorithms. We think that technology associated with inertial sensor has become mature enough to be more affordable, and therefore, additional cost justified. Particularly if we compare it with the alternative of using rotation tables and/or robot arms which can be also prohibitive in many practical situations. Our future course is the implementation of proposed idea using the actual inertial sensor which experimenting began during the submission of this work.

ACKNOWLEDGEMENTS

This work has been supported by the University of Zagreb Development Fund, Croatia. Additionally this work has been partially supported by the project ANDREA/RAIMON – Autonomous Underwater Robot for Marine Fish Farms Inspection and Monitoring (Ref CTM2011-29691-C02-02) funded by the Spanish Ministry of Science and Innovation.

REFERENCES

Chua, C. J. R., 1997. Point signatures: a new representation for 3d object recognition, Int. J. Comput. Vision 25 1, 63–85.

Chung, D., Lee, Y. D. S. 1998. Registration of multiple-range views using the reverse-calibration technique, Pattern Recogn. 31 (4) 457–464.

Dalley, G., Flynn, P., 2002. Pair-wise range image registration: a study in outlier classification, Comput. Vis. Image Und. 87 (1–3), 104–115.

Diez, Y., Martí, J., Salvi, J., 2012. Hierarchical Normal Space Sampling to speed up point cloud coarse matching Pattern Recognition Letters 33 (16), 2127–2133

Feldmar, J., Ayache, N., 1994. Rigid, affine and locally affine registration of free-form surfaces, Tech. rep., Technical Report of INRIA, Sophia Antipolis.

Godin, G., Rioux, M., Baribeau, R., 1994. Three-dimensional registration using range and intensity information, in: Proceedings of SPIE Videometric III, vol. 2350, pp. 279–290.

Makadia A., Patterson A., Daniilidis K., 2006. Fully automatic registration of 3D point clouds. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume I, pp. 1297-1304.

MTx Access date: November 2012. http://www.xsens.com

Park, S.-Y., Choi, S.-I., Kim, J., Chae, J. S., 2011. Real-time 3D registration using GPU. Machine Vision and Applications, 22 5, 837-850

Rusinkiewicz, S., Levoy, M., 2001. Efficient variant of the ICP algorithm, in: 3rd International Conference on 3-D Digital Imaging and Modeling, pp. 145–152

Salvi, J., Matabosch, C., Fofi, D., Forest, F., 2007. A review of recent range image registration methods with accuracy evaluation. Image and Vision Computing 25, pp. 578–596.

Santamaría, J., Cordon, O., Damas, S., 2011. A comparative study of state-of-the-art evolutionary image registration methods for 3D modelling. Computer Vision and Image Understanding. 115, Issue 9, 1340-1354

Shirmohammadi, B., Taylor, C. J., 2011. Self-localizing smart camera networks. ACM Transactions on Embedded Computing Systems, Vol. 8, pp. 1-26.

Stamos, I., Leordeanu, M., 2003. Automated feature-based range registration of urban scenes of large scale, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 555–561

Trucco, E., Fusiello, A., Roberto, V., 1999. Robust motion and correspondences of noisy 3-d point sets with missing data, Pattern Recogn. Lett. 20 (9) 889–898.

Wyngaerd, J. V., 2002. Automatic crude patch registration: Toward automatic 3d model building, Comput. Vis. Image Und. (87) 8–26.