Robust Multiview Registration of Point Clouds

Dhanya S Pankaj
Department of Earth and Space Sciences
Indian Institute of Space Science and Technology
Thiruvananthapuram, Kerala, India
Email: dhanyaspankaj@gmail.com

Rama Rao Nidamanuri
Department of Earth and Space Sciences
Indian Institute of Space Science and Technology
Thiruvananthapuram, Kerala, India
Email: rao@iist.ac.in

***Abstract*—Multiview registration is an important part of the 3D modeling pipeline and it aims to bring all the partial views of a model in to a common co-ordinate system. In case of availability of redundant overlap area among the partial point clouds, motion averaging provides an efficient solution to the multiview registration problem. The averaging of underlying relative motions is performed in the corresponding Lie-algebra elements of the SE(3) transformation matrices. However, in the presence of outliers in the set of relative motions this method is non-robust. We present a graph-based algorithm to filter out the relative motion outliers before performing motion averaging. The relative motions are assigned weights based on their agreement with global motions and other relative motions. The results indicate that our approach can introduce robustness to the motion averaging method of multiview registration by efficiently filtering out the outliers.**

I. Introduction

The 3D modelling of objects using 3D scanning devices finds application in many fields including reverse engineering, rapid prototyping, quality control etc. The partial 3D views of an object are obtained in multiple views and are registered together to form a single point cloud. This is usually performed in multiple stages viz. 'pairwise registration' and 'multi-view registration' [1]. The pair-wise registration align partially overlapping scan pairs and involves coarse [2] and fine registration stages [3].

Multiview registration aims to bring all of the partial scans into a common co-ordinate system and can be achieved by sequentially propagating the pairwise registration results [4]. This results in the accumulation of errors towards the end [5]. In the case of scan sequences where the first and last scans overlap, the error in loop closing can be distributed across the scans to get multiview registration [6]. The global registration approach of incrementally registering the newly acquired scans to a partially registered model [7], [4], does not make use of the information provided by the later scans, which may improve the registration of the previous scans [8]. This limitation can be overcome by the simultaneous registration methods which consider all scans at the same time and register them simultaneously [9], [10]. A comparison of different approaches for multiview registration is presented in [11].

By utilising the Lie group structure of 3D rigid body transformations, an approach to global registration of 3D point cloud is presented in [5]. An iterative algorithm which combines the correspondence estimation and motion estimation step, for computing the global transformations when the relative transformations are available, is presented. The constraints obtained from the additional relative motion estimates are utilized to average out the errors in relative motion computation than to simply distribute them. However, the algorithm does not consider the possibility of outliers in the set of relative motions. If a corrupt relative motion is present, then the result of the motion averaging will be affected since $L2$ averaging cannot handle outliers. There have been attempts in the literature to deal with outliers in motion averaging problem and the approaches can be classified into two categories. The first approach is to identify the outliers before averaging and exclude them from the averaging step [12]. The second approach is to deal with outliers using more robust averaging techniques [13]. The works in [12] and this paper deal with the first approach. The work in [12] presents a graph-based sampling scheme using Random Sample Consensus (RANSAC) to identify the inliers. The limitation of this approach is the exhaustive search of the view graph required for obtaining a consistent minimum spanning tree (MST). Also, in the case of a large graph, the convergence can be slow since the number of RANSAC iterations required depends on the percentage of inliers, the size of the sampling set etc. The motion averaging algorithm assumes a good initialization of global motions and we argue that the relative motions which are consistent with it can be estimated in a single step. We present a graph-based approach to identify the inliers in the set of relative motions. Our solution is better than the existing approach [12] because the inliers can be obtained in a single step instead of performing an exhaustive search of the view graph. Since the inliers can be found in advance, the need for a costly L1 averaging step as in [13] can be avoided.

II. Method

The global motion estimates are denoted by M_i and represent the 3D transformation from each of the scans $S_i, i = 1 : N$ to the initial scan S_1. The relative motion estimates are denoted by M_{ji} and represent the 3D transformation from scan j to scan i. Assume that there are n relative motions. Let $G = (V, E)$ be the view graph where the vertices V represent the N partial 3D views. The edges E represent the availability of a relative motion estimate between a pair of views. Our goal is to identify the correct relative motion estimates. We propose an algorithm to estimate the weights of each of the edges. An estimate of the global motions is computed by finding the shortest path from each of the scan to the scan S_1 by using Dijkstra's single source shortest path algorithm (SSSP) [14]. Then, the inlier edges are identified and the motion averaging is performed using the inliers. The steps involved in the proposed algorithm are detailed in Algorithm 1.

Algorithm 1 Proposed algorithm

1. Form the graph $G = (V, E)$ with $V = \{$ scan views, $S_1 : S_N\}$ and $E = \{$ relative motions, $M_{ji}\_k,\ k = 1 : n\}$
2. Find an initial estimate of the global motions $M_i,\ i = 1 : N$
3. Find edge weights for the relative motion estimates.
4. Find shortest path from each scans to scan S_1 by Dijkstra's SSSP algorithm.
5. Find global motion estimates by sequentially registering along the shortest path.
6. Identify the inliers among the relative motion estimates
7. Perform motion averaging to refine the global motion estimates using the inlier relative motion estimates (Algorithm 2).

A. Initialize Global Motion Estimates

The sequential registering of scan views has the drawback of errors getting accumulated towards the end. To minimize the propagation of errors at least by half, we propose to register the scan pairs in two directions if the relative motion from scan N to the first scan is available. The scan pairs from S_1 to $S_{N/2}$ are registered sequentially from the first scan and those from S_N to $S_{N/2+1}$ are registered in the reverse direction starting from S_1. Here large accumulation of errors towards the end of the scans is reduced. However, there will be a shift near to $S_{N/2}$, which will be usually less compared to the error towards the scan S_N in sequential registration.

B. Weight calculation

Relative motion weights are assigned based on their agreement with respect to the estimates of global motions and other relative motions. Consider another graph $G_1 = (V_1, E_1)$ where the vertices represent relative motion estimates (M_{ji}). Edges are present if there is a common scan view in both the vertices. i.e. if (M_{ji}) and (M_{lm}) are the two relative motion estimates, then there is an edge connecting them if $j = m$ or l or $i = m$ or l. The distance between relative and global motions is calculated and assigned as the vertex weight.

1) Distance between motion matrices: In order to find the difference between two motion matrices, we consider the fact that the group of rigid transformations is a Linear Lie group $SE(3)$ [15]. The distances between the elements of $SE(3)$ can be either intrinsic or extrinsic [16]. The intrinsic distance (Riemannian) is the length of the shortest geodesic curve between the elements [16]. Please refer to [17], [18] for details on Lie Groups and $SE(3)$ group. For $X,\ Y\ \epsilon\ SE(3)$, the Riemannian distance is defined by Eq.1 [18] where log refers to the logarithmic map that maps Lie group to the corresponding Lie algebra [19].

$$d(X, Y) = \|log(YX^{-1})\| \approx \|log(Y) - log(X)\| = \|y - x\| \tag{1}$$

The approximation is obtained by the BCH formula (Eq.2) where x and y are the corresponding Lie algebra elements of X and Y. $\|.\|$ is the Frobenius norm. Thus by tangent space approximation, the Riemannian distance between two elements in $SE(3)$ group is now given by the 'Euclidean distance' in its Lie algebra [12]. This is the first order linear approximation of the Riemannian distance between the motion matrices.

$$BCH(x,\, y) = x + y + \frac{1}{2}[x, y] + \frac{1}{12}[x - y, [x, y]] + \mathcal{O}(|(x, y)|^4) \tag{2}$$

$$\tilde{M_{ji}} = M_i^{-1} * M_j \tag{3}$$

The distance between the input relative motion estimates M_{ji} and the relative motions computed from the global motions (Eq.3) are computed and assigned as vertex weights of G_1 (v_i). This distance is given by Eq.4. The vertex weights indicate the agreement of the calculated relative motion with the initial global estimates. Lesser weights indicate more agreement as the weight is a measure of distance or error.

$$d(\tilde{M_{ji}}, M_{ji}) = \|log(\tilde{M_{ji}}) - log(M_{ji})\| \tag{4}$$

The correctness of a relative motion estimate is further affirmed by checking its compatibility with other relative motion estimates. Two relative motion estimates can be compared if there is an edge (common scan view) between them. The global motion estimates of the common scan view are computed from both of the relative motion estimates using one of the equations in Eq.5. Then the distance between them is calculated using Eq.2 and assigned as the edge weight e_j.

$$\begin{aligned} M_j &= M_i * M_{ji} \\ M_i &= M_j * M_{ji}^{-1} \end{aligned} \tag{5}$$

Once the edge weights (e_j) are computed, the vertex weights (v_i) are updated to $v_i^{'}$ as in Eq.6 where n_i is the number of edges at vertex i.

$$v_i^{'} = \frac{1}{n_i} * \sum_{j=1}^{n_i} (v_i * e_j) \tag{6}$$

The average of the edge weights at each vertex is calculated as all vertices may not be having the same number of edges and this depends on the input (edges in the view graph) data. The average edge weights are weighted by the initial vertex weights as the compatibility with global motion estimates is more indicative than that with other relative motion estimates, which may contain outliers as well.

C. Calculation of inliers

Once the weights for the relative motion estimates are obtained, this can be assigned to the view graph $G = (V, E)$ where V represents the scan views and E, the relative motions. The weights computed in section II-B act as the edge weights. The weights assigned to the outlier edges will be high because of their disagreement with the global motion estimates as well as to the other relative motion estimates whereas the weights assigned to the inlier edges will be less. From the view graph, the shortest path to the scan S_1 is computed from other scan views using Dijkstra's SSSP algorithm and this excludes the outlier relative motions which have higher assigned weights. The global motion estimates can then be calculated by registering relative motions sequentially along the computed path. Once the global motions are obtained, the inliers to the global motions are found out. The distance between the initial relative motions and the relative motions obtained from the updated global motions (as in Eq.5) are computed using the Riemannian distance measure in Eq.2. The distance is compared against a threshold (computed empirically) to identify the inliers to the model.

D. Motion Averaging

The global motions are estimated from the inlier relative motions using motion averaging [5]. The correspondence and relative motion estimation stage of ICP are combined with the motion averaging in an iterative algorithm. The steps involved are briefly outlined in Algorithm 2 [5], [20].

Algorithm 2 Motion Averaging

1. Compute the incremental update to the relative motions using ICP and update the relative motions
2. Initialize global motions from the relative motions (recalculate registration path)
3. Motion averaging
$\Delta M_{ji} = M_i^{-1} * M_{ji} * M_j$
$\Delta m_{ji} = log(\Delta M_{ji})$
$\Delta v_{ji} = vec(\Delta m_{ji})$
$\Delta \mathcal{V} = D^{\dagger} \Delta \mathcal{V}_{ji}$
$\forall k \,\epsilon\, [2, N], M_k = exp(\Delta m_k) * M_k$
$Repeat\, until \|\Delta \mathcal{V}\| < \varepsilon$
4. Update relative motions from global motions $M_{ji} = M_i^{-1} * M_j$
5. Repeat steps 1 to 4 until convergence

Δm_{ji} indicate the corresponding Lie algebra element of ΔM_{ji}, and since Δm_{ji} is a skew-symmetric matrix, it can be uniquely expressed by a 3-element vector given by Δv_{ji} which is obtained by the operation $vec(.)$. The relation in Eq.7 is derived from Eq.1, where $\mathcal{V}$ is obtained by stacking the vectors v_i in a column. $D_{ji} = [\ldots, -\mathbf{I}_{6\times6}, \ldots, \mathbf{I}_{6\times6}]$ is a matrix with 6 rows and $6N$ columns, where $-\mathbf{I}_{6\times6}$ appears at $6i$ th column $\mathbf{I}_{6\times6}$ appears at the $6j$th column. $\Delta \mathcal{V}_{ji} = [\Delta v_{ji1}, \Delta v_{ji2}, \ldots, \Delta v_{jin}]^T$, $D = [D_{ji1}, D_{ji2}, \ldots, D_{jin}]^T$ and $D^{\dagger}$ is the pseudo inverse of D. The $log()$ and $exp()$ operations are as defined in [19].

$$\begin{aligned} m_{ji} &\approx m_j - m_i \\ \implies v_{ji} = v_j - v_i &= [...{-\mathbf{I}}...\mathbf{I}...]\mathcal{V} \end{aligned} \tag{7}$$

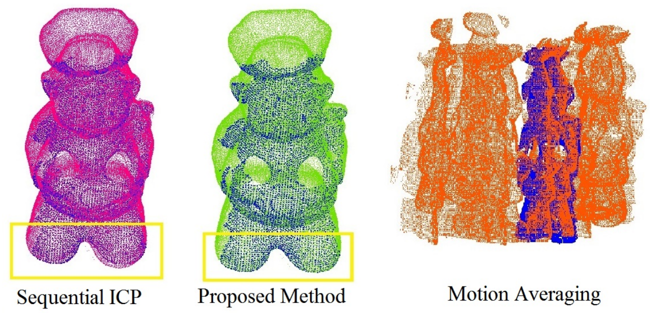
The global motion estimates in step 2 can be estimated by finding out the minimum spanning tree of the view graph (Prim's minimum spanning tree algorithm) [14]. In order to introduce more robustness to the motion averaging, we recalculate the registration path at each step. The proposed weight calculation algorithm and Dijkstra's SSSP algorithm is employed for estimating global motion. This is introduced to keep the registration path up to date according to the modified relative and global motion estimates.

III. Experiments

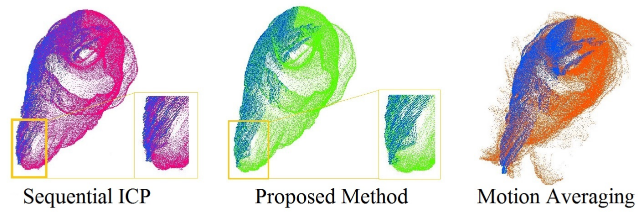
Partially overlapping point clouds of 3D models from the Stanford dataset [21], acquired using Cyberware 3030 MS scanner (Buddha) and from the University of Western Australia (UWA) dataset [22], acquired by Minolta scanner (Chef and Chicken) were tested. A set of pairs of overlapping scan views was considered for registration, which might contain outliers. The relative motion estimates of these pairs were obtained by performing the coarse [2] and fine registration(ICP). The inliers from relative motions were identified and the motion averaging was performed using them. We compared the registration results of the proposed algorithm with the results obtained with a) sequential registration by ICP [4] and b) motion averaging without the removal of outliers. In the case of Stanford model, the actual transformations in terms of 3D translation units and the 3D rotation unit quaternions [23] were available. Hence the obtained transformation matrices after global registration were converted to x, y, z translation units and the rotation unit quaternions for comparison. The dot product between the actual and computed rotation unit quaternions, which is used as a measure of similarity between the rotation unit quaternions, was computed. The greater the dot product value, the better is the similarity. The difference between estimated and actual global motions was calculated using the Riemannian distance measure. The quality of the registration was identified visually from the registered model when actual motion estimate was not available. The cross sections of the models also give evidence on the quality of registration. The proposed algorithm was tested on different sets of input relative motion pairs. The result for only a single set for each model is shown here for want of space; although the results obtained supported the proposed method. The algorithm is implemented in C++ with the help of PCL library [24].

IV. Results and Discussion

The set of overlapping scan pairs was passed as input to the algorithm, along with their relative motions. The global motions were initialized as in section II-A. The method of sequential registration of the scans in one direction, based on the results obtained by ICP in the pairwise registration, is referred to as sequential ICP hereafter in this document. The method of registration by motion averaging using the entire set of input relative motions is referred to as Motion Averaging [5] in this document.



(a) Chef



(b) Chicken

Fig. 1. Registered models obtained by multiview registration for the UWA dataset

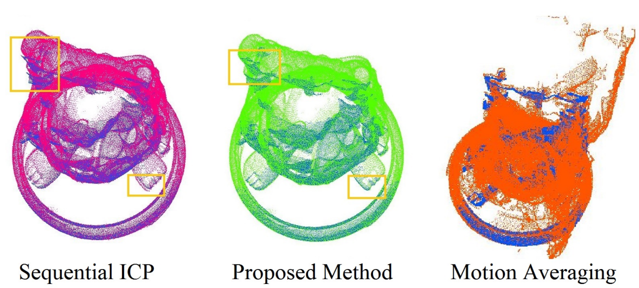
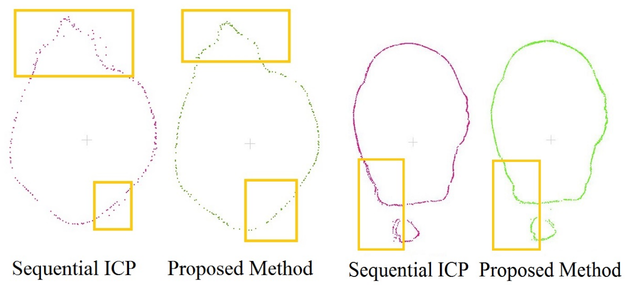


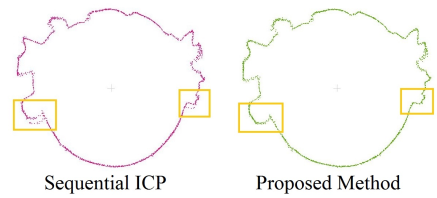
Fig. 2. Registered Stanford Buddha model obtained by multiview registration

The registration results of the UWA dataset models Chef and Chicken using a) Sequential ICP, b) Proposed method and c) Motion Averaging methods are shown in Fig.1. For clarity, the first scan of the model (in blue) is overlapped with the results. From the result of sequential ICP method, we can see that the errors had accumulated towards the end and this is highlighted in the figure for clarity. The initial scan is not matching with the registered final scan due to the accumulated error. The motion averaged results show that the outliers had corrupted the result to a great extent. The results by the proposed method indicate that the outliers were removed successfully and averaging by the inliers resulted in a properly merged point cloud.

Multiview alignment was performed on the scan views of Stanford Buddha model. The registered Buddha model obtained by a) Sequential ICP, b) Proposed method and c) Motion averaging are shown in Fig.2. The outcomes from registering this model also support our previous argument that the proposed method correctly identifies the inliers in the input relative motions and performs motion averaging to obtain a good global alignment. The initial scan is shown in blue, overlapped with the results, for reference.

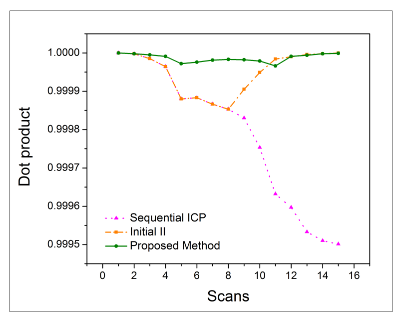


(a) Chef (b) Chicken

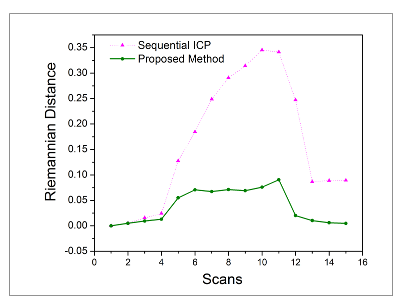


(c) Buddha

Fig. 3. Cross sections of the registered models



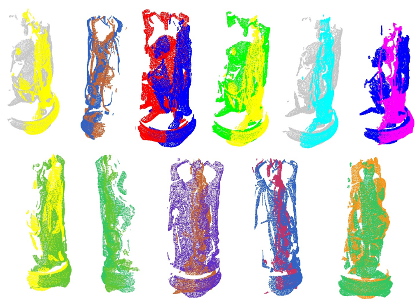
(a) Dot product of actual and estimated global rotations



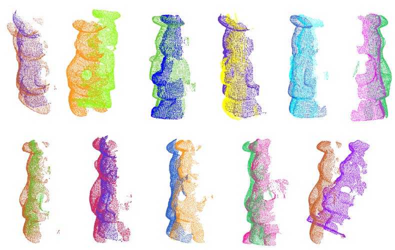
(b) Distance between estimated and actual global motions

Fig. 4. Buddha model - Comparison of actual and estimated global motions

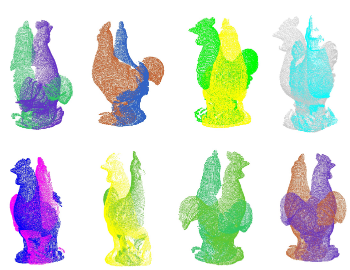
The cross sections of the registered models were extracted to highlight the loop closing error by the sequential ICP. The cross section of the Chef, Chicken and Buddha models obtained by sequential ICP and by the proposed method are shown in Fig.3. We can clearly see that the error in loop closing caused by the sequential ICP was rectified to result in a smooth cross section by the proposed method. For the Chicken model, the number of inlier redundant scan pairs available for averaging was less and hence the smoothing achieved was also less compared to other models.



(a) Stanford model - Buddha



(b) UWA model - Chef



(c) UWA model - Chicken

Fig. 5. Outliers in the relative motions identified by the proposed algorithm

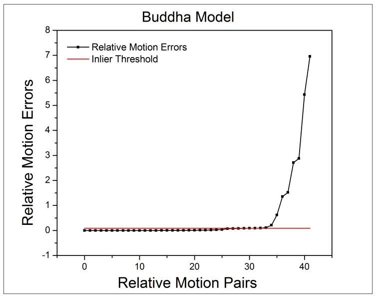
For the Stanford model, the actual global motions are available in the form of unit quaternions and 3D translation vectors. We converted the global transformation (M) obtained to quaternions and 3D translation vectors for comparison. The rotations were compared by taking the dot product of the actual and estimated unit quaternions and the results for the scan pairs of Buddha model is provided in Fig.4(a). The dot product was calculated for the results obtained by the sequential ICP method, by the initialization method mentioned in section II-A and by the proposed method. From the figure, we can observe that the accuracy of rotation estimation obtained by the sequential ICP method deteriorates as the number of scans increases, due to the accumulation of registration error. From the result of proposed initialization method mentioned in section II-A, hereafter called as Initial II, we can see that by registering around the middle scan view, we can reduce the propagation of errors all the way to the last scan. The dip in the middle portion is due to the accumulated error from both directions. The proposed method is initialized using the result of Initial II. From the figure, we can see that the proposed method is able to achieve reasonably accurate global rotation estimates by averaging the relative motions after performing outlier rejection. We have not included the results by motion averaging here as the merged cloud is largely corrupted by outliers. The distance between the actual global motions and the estimated global motions were computed using the Riemannian distance measure. The results are provided in Fig.4(b). Here we can see that the proposed method was able to achieve a good estimate of global motions by selecting the valid inliers and averaging them using motion averaging. The outliers present in the set of relative motions identified by the proposed approach are shown in Fig.5 and shows that the proposed method filters out the true outliers. The scans are aligned according to the input pairwise relative motion estimates.

Fig.6 shows the inlier selection by the proposed method for the various models tested. The Riemannian distances between input relative motions and estimated relative motions from the global motions as in II-C were computed and compared against a threshold value (which was calculated empirically and was guided by the resolution of the point cloud). The relative motion was considered an inlier if this distance is below the threshold value. In the Fig.6, the red line shows the threshold selected. We can see that the inliers and outliers are separated by a relatively large margin which makes the selection of threshold easier. The distances are sorted in the graph for clarity. Small refinement errors in relative motion estimates can be handled by the motion averaging step which iteratively refines the relative and global motion estimates. Hence threshold may be selected in such a way as to avoid only large outliers.

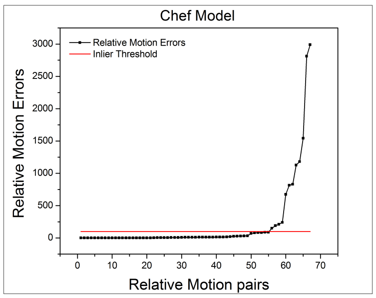
The vertex weight of graph G_1 in section II-B is a measure of the agreement of a particular relative motion with the rest of the relative motion estimates as well as with the initial global motion estimates. By considering the agreement with the other relative motions also, we ensure that the outliers get large vertex weights compared to the inliers and hence get excluded from the succeeding global motion estimation (by Dijkstra's SSSP) algorithm. The calculation of global motion estimates using the shortest path algorithm helps in considering the edges which are not included in the initial global estimation. The recalculation of the registration path within the motion averaging iteration helps in updating the path according to the latest estimates.

The proposed algorithm for robust motion averaging is better compared to the existing approaches [12], [13] from a computational point of view. The runtime of RANSAC loop in [12] is less predictable and data dependent compared to the single step calculation of inliers in the proposed algorithm.

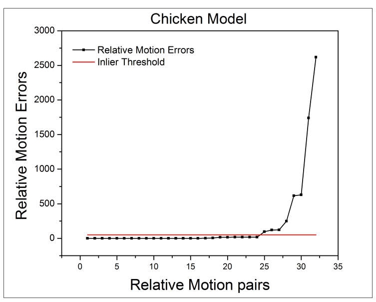
The L1 averaging step in [13] and the robust averaging is not required in the proposed approach as the inliers are identified before the motion averaging step.



(a) Stanford model - Buddha



(b) UWA model - Chef



(c) UWA model - Chicken

Fig. 6. Inlier calculaion by the proposed algorithm

V. Conclusion

A novel algorithm for introducing robustness in the multi-view registration of 3D point clouds using motion averaging is presented. The proposed algorithm is simple and efficiently filters outliers using a graph-based approach and the Lie Group structure of the rigid body motions. By making use of the Riemannian distance measure to form the edge weights in the view graph, the algorithm identifies the outliers in relative motion estimates. The results are compared with the existing approaches and are shown to be robust and accurate.

Acknowledgment

The work of the first author is supported by a Doctoral Fellowship provided by IIST, Department of Space, Government of India. The authors would like to thank the Stanford University and the University of Western Australia for providing the datasets freely available online.

References

[1] C. Dorai, G. Wang, A. K. Jain, and C. Mercer, "From images to models: Automatic 3d object model construction from multiple views," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 1. IEEE, 1996, pp. 770–774.

[2] D. S. Pankaj and R. R. Nidamanuri, "A robust estimation technique for 3d point cloud registration," *Image Analysis & Stereology*, vol. 35, no. 1, pp. 15–28, 2016.

[3] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, 2001, pp. 145–152.

[4] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[5] V. M. Govindu and A. Pooja, "On averaging multiview relations for 3d scan registration," *Image Processing, IEEE Transactions on*, vol. 23, no. 3, pp. 1289–1302, 2014.

[6] J. Sprickerhof, A. Nüchter, K. Lingemann, and J. Hertzberg, "An explicit loop closing technique for 6d slam." in *ECMR*, 2009, pp. 229–234.

[7] K. Pulli, "Multiview registration for large data sets," in *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*. IEEE, 1999, pp. 160–168.

[8] S.-W. Shih, Y.-T. Chuang, and T.-Y. Yu, "An efficient and accurate method for the relaxation of multiview registration error," *Image Processing, IEEE Transactions on*, vol. 17, no. 6, pp. 968–981, 2008.

[9] P. J. Neugebauer, "Geometrical cloning of 3d objects via simultaneous registration of multiple range images," in *Shape Modeling and Applications, 1997. Proceedings., 1997 International Conference on*. IEEE, 1997, pp. 130–139.

[10] D. F. Huber and M. Hebert, "Fully automatic registration of multiple 3d data sets," *Image and Vision Computing*, vol. 21, no. 7, pp. 637–650, 2003.

[11] S. Cunnington and A. J. Stoddart, "N-view point set registration: A comparison." in *BMVC*, 1999, pp. 1–11.

[12] V. M. Govindu, "Robustness in motion averaging," in *Computer Vision–ACCV 2006*. Springer, 2006, pp. 457–466.

[13] A. Chatterjee and V. Govindu, "Efficient and robust large-scale rotation averaging," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 521–528.

[14] A. Gibbons, *Algorithmic graph theory*. Cambridge University Press, 1985.

[15] J.-L. Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, 2010.

[16] P. T. Fletcher, C. Lu, and S. Joshi, "Statistics of shape via principal geodesic analysis on lie groups," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1. IEEE, 2003, pp. I–95.

[17] K. Kanatani, *Group-theoretical methods in image understanding*. Springer Science & Business Media, 2012, vol. 20.

[18] V. M. Govindu, "Lie-algebraic averaging for globally consistent motion estimation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I–684.

[19] X. Duan, H. Sun, and L. Peng, "Riemannian means on special euclidean group and unipotent matrices group," *The Scientific World Journal*, vol. 2013, 2013.

[20] Z. Li, J. Zhu, K. Lan, C. Li, and C. Fang, "Improved techniques for multi-view registration with motion averaging," in *3D Vision (3DV), 2014 2nd International Conference on*, vol. 1. IEEE, 2014, pp. 713–719.

[21] M. Levoy, J. Gerth, B. Curless, and K. Pull, "The stanford 3d scanning repository," *URL http://www-graphics. stanford. edu/data/3dscanrep*, 2005.

[22] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *International Journal of Computer Vision*, vol. 89, no. 2-3, pp. 348–361, 2010.

[23] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[24] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–4.