

Fast Registration of Articulated Objects from Depth Images

Sourabh Prajapati

Centre for Visual Information Technology,
International Institute of Information Technology,
Hyderabad - 500032
Email: srbhprajapati@gmail.com

P. J. Narayanan

Centre for Visual Information Technology,
International Institute of Information Technology,
Hyderabad - 500032
Email: pjn@iiit.ac.in

Abstract—We present an approach for fast registration of a Global Articulated 3D Model to RGBD data from Kinect. Our approach uses geometry based matching of rigid parts of the articulated objects in depth images. The registration is performed in a parametric space of transformations independently for each segment. The time for registering each frame with the global model is reduced greatly using this method. We experimented the algorithm with different articulated object datasets and obtained significantly low execution time as compared to ICP algorithm when applied on each rigid part of the articulated object.

I. INTRODUCTION

In recent years, cheap and fast scanning devices have been easily available in the market. Scanning devices like Microsoft Kinect provides us with plentiful of information about the surrounding environment. Using such vast information, it is possible to generate 3D Geometric model of the nearby environment. 3D Geometric Models provides better insight of the surface pattern, physical structure and realistic motion. With fast scanning devices like Kinect, it is viable to generate real-time 3D Geometric Model.

Range scan registration poses the problem of matching and estimation of rigid transformation of various scans of the same object. What makes the problem difficult is that correspondence between the point scans are unknown a priori. If the object is rigid, registration and tracking is relatively easier to solve. Motion of articulated objects is complex as they are continuously changing their shape during the motion. Although the overall structure is changing, the structure of each rigid part remains the same.

In this paper, we presents a fast algorithm for registration of articulated 3D global models to frames from RGBD images from devices like the Kinect. Our approach transforms the global model to align with successive depth images from sensors like Kinect. Our approach is fast and efficient to align the known global model to an unknown depth image frame. We perform geometry-based registration of articulated objects and perform the registration in a parametric space, inspired by Hough transforms.

A popular approach for registration of rigid surfaces is Iterative Closest Point (ICP) introduced by Besl [1] and Chen [2]. The technique is attractive because of its simplicity and its performance. Although initial estimate does need to be reasonably good, the algorithm converges relatively quickly. Other techniques were developed with a variation of ICP

for registration of Non rigid Surfaces([3], [4], [5]). These approaches are limited to local deformation of non rigid objects and will not work for significant changes in the pose. Kinect Fusion [6], developed in recent years, creates 3D models of real world objects or environments by combining a continuous stream of data from the Kinect. It enables 3D object model reconstruction, 3D augmented reality, and 3D measurements. However if the object in the environment is moving, it creates multiple deformed models.

Mitra et al. [7] performs shape registration as an application of symmetry detection. Sagawa et al. [8] rely on matching both texture and shape features to register a sequence of deforming texture scans. Other work on mesh morphing solves the correspondence problem by finding a common base parameterization across multiple meshes of a common topological type ([9], [10], [11]).

The Correlated correspondence algorithm developed by Angelov [12] is an unsupervised algorithm for registering 3D surface scans of an object undergoing significant deformations. It doesn't uses user-placed markers, or assumes temporal coherence or rough initial alignment. With the modest requirement of template shape, it finds a good correspondence assignment in spite of significant changes in the object. Chang and Zwicker [13] performed registration of articulated object to optimize the transformation assignment. Although the registration obtained is quite effective, it required large amount of time.

II. OUR APPROACH

This paper presents a fast algorithm to register an articulated 3D global model to its depth scans. The animation consist of motion of different rigid parts of an articulated object. The motion is tracked by comparing the feature all points in one frame to another. The feature geometry at any point of the object is described as a collection of several points around the point with the distances between them and information about their local geometry. Depth scans are searched for the geometry of each rigid part and the difference between the position of corresponding rigid parts provides the transformation of the articulated object. Final Transformation for the articulated objects consists of a set of transformation each of which representing the movements of one rigid part of the object.

The global model, G , is a known standard 3D model that can be rendered from all directions to depth images. The

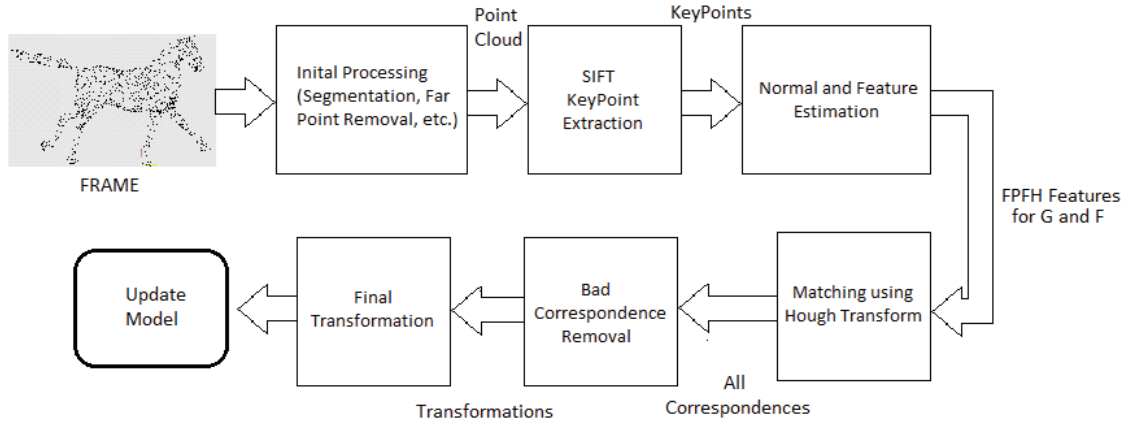


Fig. 1. Pipeline for the registration of Articulated Object Using Hough Transform

templateless articulated model consists of multiple rigid parts that are labelled such that each point in the point cloud can be labelled to one of the rigid parts. Our proposed pipeline herein takes input a global model and aligns it with the incoming depth frames of the same moving object. The pipeline uses Hough Transform for matching features of global model to incoming frames and calculate the transformation on a voting basis.

A. Pipeline

The execution of the algorithm pipeline starts with the preprocessed global model and the first frame as input. The frame is then processed and matched with the global model. The different steps of the pipeline are described below.

1) *Initial Processing*: Initial step of the pipeline consist of loading the point cloud from the sensor and segmenting the object from background. If the density for the point cloud is high to process then we downsample the point cloud using voxelgrid filter. Other filtering used are radius outlier removal, color filtering, etc. depending on the type of point cloud. After the Initial processing, we are left with a depth frame consisting of point cloud of the given object which is then forwarded to next block in the pipeline.

2) *SIFT KeyPoint detection*: Since the number of points in a point cloud is huge, we sample them to a subset of interest points that represents stable, distinctive and identifiable structure of the object. The number of interest points in a point cloud will be much smaller than the total number of points in the cloud. When used in combination with local feature descriptors at each keypoint, the keypoints and descriptors can be a compact yet descriptive representation of the original data. SIFT(Scale Invariant Feature Transform) [14] can be extended from images to 3D Point Clouds by using RGB values or point value or normal and curvature as replacement for “intensity” [15]. We detect SIFT keypoint for a given point cloud using normals and curvature on these points and match only these keypoints to incoming frame. This has two advantages: 1.) reducing the number of points to be processed and therefore reducing the time of processing non SIFT Keypoints and 2.) Since we are only matching keypoints of Global model to

keypoints of frame, we reduce the error of matching a keypoint to a non-keypoint of frame (as keypoints are only supposed to match corresponding keypoints). Here we are using SIFT only to detect the keypoints and not to define the features for matching. After the selection of keypoints, the next step is to calculate features of these keypoints.

3) *Normal and Feature Estimation*: We use the Fast Point Feature Histogram (FPFH) for matching [16], [17]. FPFH is a faster version of Point Feature Histogram (PFH) and retains most of the discriminative power of the PFH. Point Feature Histograms are robust multi-dimensional features which describe the local geometry around a point p for 3D point cloud datasets. We calculate FPFH features on each of the identified interest points of G . The algorithm is tested with various features including SIFT, PFH, Spin Images, FPFH. Out of these features FPFH gave the best result in execution time performance.

4) *Matching using Hough Transforms*: Hough transform is a technique to find structure in a parametric space even under incomplete data. It uses voting in the parametric space (here it is Transformation space) to accumulate evidence of the presence of a particular structure. We use Hough transform in the Transformation space after matching FPFH features from the observed frame to the known model for each interest point. We first match interest points using FPFH by searching in the known model. Since the object is articulated, the search space for registration of one rigid part is reduced from the position of nearby rigid part, thereby improving computational performance. Each match corresponds to a vote the transformation space for that part. When all the matching is completed, the distribution of votes in the transformation space represents multiple clusters, one for each joint. The final transformation is calculated by extracting clusters and evaluating the transformation for each. Figure 2 shows the implementation of Hough Transform.

5) *Bad Correspondence Rejection*: After matching points using Hough Transform, we can get multiple correspondences from one keypoint of global model to incoming frame. Due to local neighbourhood similarity, there will be several wrong matches. Since the frames are coming in continuously, we

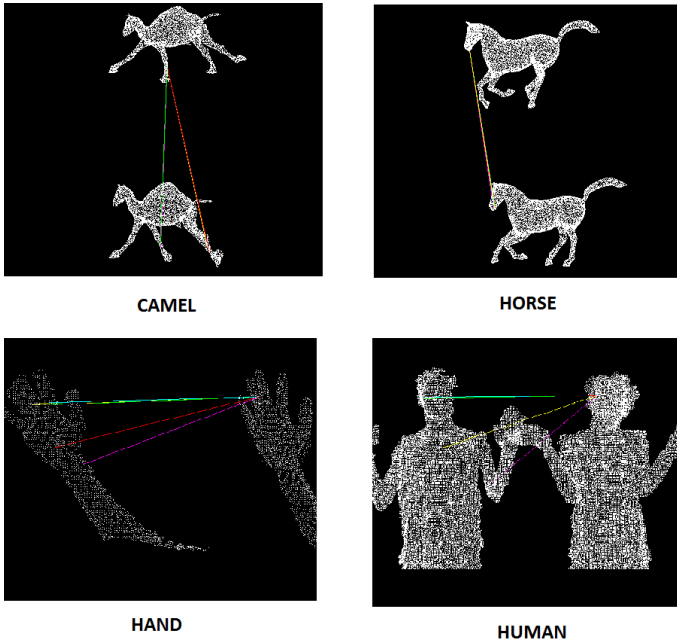


Fig. 2. Hough Transform for various datasets depicting multiple correspondences of the same point. Each correspondence is used as a vote in parametric space.

anticipate the motion to be continuous and reject the correspondences which are too far off. We remove correspondences if the difference between motion of a particular point is above a manually selected threshold.

6) *Final Transformation*: In the final stage of the pipeline, we calculate the final transformation by clustering the correspondences in transformation space and finding out the transformation that represents the cluster. For a most efficient transformation, all the correspondence will lie around a point in transformation space. We used a simple data clustering approach in an Euclidean sense which is implemented by making use of a 3D grid subdivision of the space using fixed width boxes, or more generally, an octree data structure. We made use of nearest neighbors and used a clustering technique that is essentially similar to a flood fill algorithm. Each cluster represents the transformation of corresponding rigid part of the object.

B. Preprocessing

In the preprocessing step, G is processed to label all the points to their corresponding rigid parts. G is computed so as to have the information of whole articulated object. We used multiple incomplete frames and combined them with ICP algorithm to get the global model with complete information of the articulated object. So for all the features in the incoming frames, there will be a feature in G . All the points in G is then labelled with their corresponding rigid parts. Therefore, for each correspondence from frame to G , we'll have the corresponding rigid part it matched with. After the labelling, we calculate the SIFT keypoints using normal and curvature for G which are to be used for matching with frames. For these set of points, Normals and FPFH features are calculated to describe the local geometry around them. After this, G is

given as initial input to the pipeline.

III. EXPERIMENTATION RESULTS AND DISCUSSION

We performed experiments on several object datasets for registration with multiple frames. For each dataset, a set of frames which describes motion of the articulated object, are taken to perform the registration. We took a hand model, a horse model, an arm model, a camel model and a human model. We have implemented the code in C++ and used Point Cloud Library(PCL) [15] for processing of incoming frames. For each object dataset, a sequence of 35 frames is used for experimentation. The experiments were intended to verify the computational efficiency and accuracy of the presented algorithm. The results are compared with ICP algorithm (provides registration of rigid objects with high accuracy) for each rigid part of the articulated object. The mesh data used in this project was made available by Robert Sumner and Jovan Popovic from the Computer Graphics Group at MIT [18].

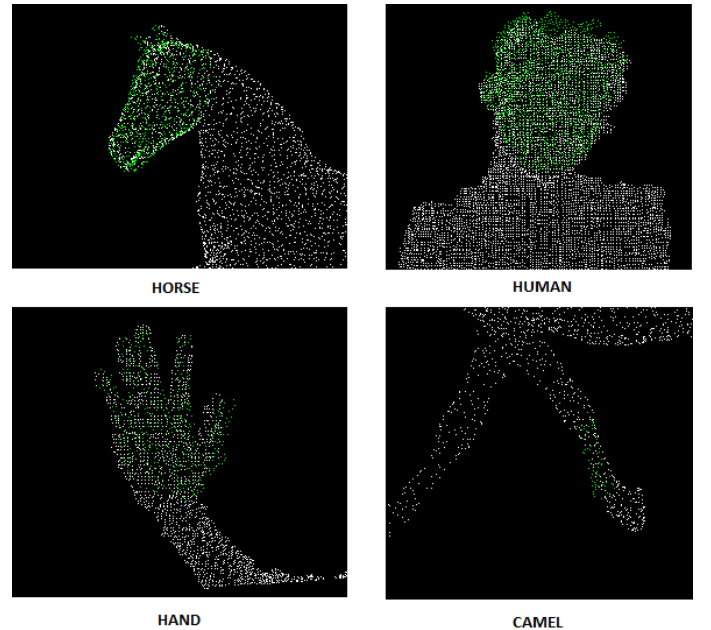


Fig. 3. Registration of various datasets. The white cloud represent the target point cloud of the incoming frame to be registered with and the green cloud represents a rigid part of the Global model.

Figure 4 shows the final average reprojection error of registration of all datasets. The Final Average Reprojection Error for the registration of frames varies from 1% to 9%. But the execution time per frame is reduced by more than 95% compared to ICP. The error in registering multiple frames in the proposed pipeline is not accumulative. Figure shows the registration between the various frames of the datasets. Since the matching is done in continuation one frame after another frame, the error of one frame is therefore included in the registration of next frame. Hence the error doesn't add up. The bulgy area represents the accumulation of error but the decline after the bulge shows that the error is reduced by including it in the registration of next frame. In Figure 4, we can see that there are many bulgy areas in between frames where the reprojection error is high (around 8-9 %) but while registering successive frames, this error is compensated in

TABLE I. AVERAGE PERFORMANCE AND TIMING STATISTICS FOR FIVE DATASETS. THE RUNNING TIME STATISTICS WERE GATHERED FROM TESTING OUR IMPLEMENTATION ON MULTIPLE CORES OF A CORE I3 2.3 GHZ INTEL PROCESSOR.

Object Model	#Points	#Joints	#KeyPoints	Preprocessing Time	#Frame Vertices	Ava Matching Time	Avg Clustering Time	Avg Transformation Time	ICP Time
Hand	2281	2	241	5.0 min	2100	0.9s	0.008s	0.01s	3.1 min
Horse	3490	14	442	8.2 min	3396	1.1s	0.01s	0.01s	8 min
Arm	3620	5	374	7.3 min	3507	1.0s	0.008s	0.01s	5.9 min
Camel	3500	14	426	9.0 min	3412	1.1s	0.01s	0.01s	8 min
Human	4563	8	577	10.8 min	4270	1.4s	0.009s	0.01s	11 min

the calculation of next transformation, hence the overall error for the next frames comes out to be better than the previous error. The execution and processing time are shown in I. In any frames, if a rigid part of the articulated object is occluded or partial data of the rigid part is present, then transformation of the articulated object will be performed using registration of partial data. Since some of the data is missing, the features that are calculated may have a little difference in the neighbourhood, therefore will have more registration error.

Parameters: The performance of the algorithm is dependent on parameters we choose for registration. Major parameters are Normal and Feature Radius, Votes per keypoint, Probabilistic Search Radius, Matching Threshold, Correspondence removal distance Threshold. Choosing them inappropriately will make the algorithm computationally inefficient.

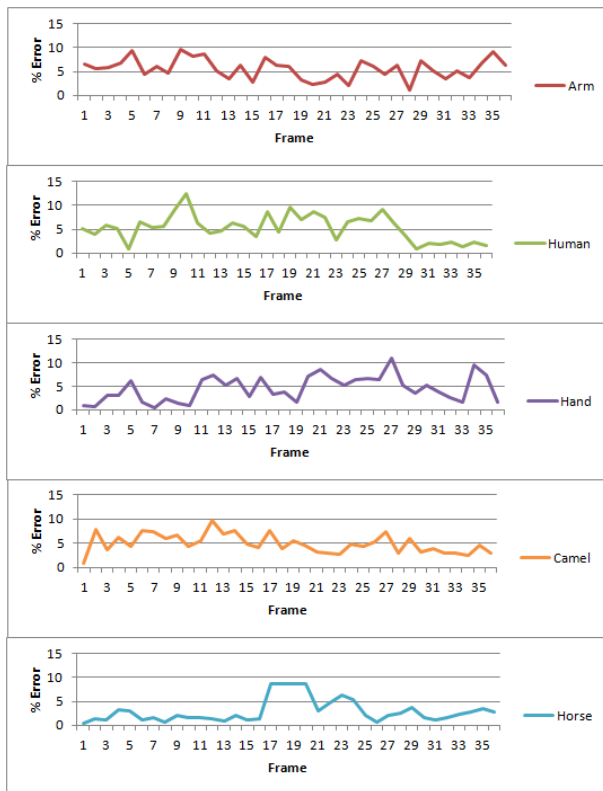


Fig. 4. Final Average Reprojection Error for datasets

IV. CONCLUSION

We presented a fast articulated object registration algorithm for aligning articulated 3D model to incoming depth frames. We formulated the registration problem as matching geometry

of rigid part of the articulated object. The algorithm finds out the geometry of rigid objects in depth frames. It then transforms the global model so as to align with the depth frames. With the final reprojection error for registration ranging from 1%-9%, the execution time is reduced by more than 95%.

REFERENCES

- [1] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.
- [2] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, April 1992.
- [3] B. J. Brown and S. Rusinkiewicz. Global non-rigid alignment of 3-d scans. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.
- [4] D. Hähnel, S. Thrun, and W. Burgard. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. 2003.
- [5] Brian Amberg. Optimal step nonrigid icp algorithms for surface registration. In *In CVPR07*, 2007.
- [6] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3d reconstr. and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on UI software and technology*, pages 559–568, NY, USA, 2011. ACM.
- [7] N. J. Mitra, S. Flory, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann. Dynamic geometry registration. In *Symposium on Geometry Processing*, pages 173–182, 2007.
- [8] R. Sagawa, N. Osawa, and Y. Yagi. Deformable registration of textured range images by using texture and shape features. In *Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling, 3DIM '07*, pages 65–72, Washington, DC, USA, 2007. IEEE Computer Society.
- [9] S. Takahashi, Y. Kokojima, and R. Ohbuchi. Explicit control of topological transitions in morphing shapes of 3d meshes. In *Proceedings of the 9th Pacific Graphics*, PG '01, pages 70–, Washington, DC, USA, 2001. IEEE Computer Society.
- [10] T. Kanai, H. Suzuki, and F. Kimura. Metamorphosis of arbitrary triangular meshes. *IEEE Comp. Graph. Appl.*, 20(2):62–75, March '00.
- [11] A. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. In *Proceedings of SIGGRAPH*, pages 343–350, August '99.
- [12] D. Anguelov, P. Srinivasan, H. Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS*, 2004.
- [13] W. Chang and M. Zwicker. Automatic registration for articulated shapes. *Computer Graphics Forum (Proceedings of SGP 2008)*, 27(5):1459–1468, 2008.
- [14] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.
- [15] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE ICRA*, Shanghai, China, May 9-13 2011.
- [16] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proceedings of the 2009 IEEE ICRA*, 2009, pages 1848–1853, Piscataway, NJ, USA, 2009. IEEE Press.
- [17] Radu Bogdan Rusu, Andreas Holzbach, Nico Blodow, and Michael Beetz. Fast geometric point labeling using conditional random fields. In *The 22nd IEEE/RSJ IROS*, St. Louis, MO, USA, 10/2009 2009.
- [18] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Trans. Graph.*, 23(3):399–405, August 2004.