REVIEW: FAST POINT FEATURE HISTOGRAMS (FPFH) FOR 3D REGISTRATION

Group 6: Augmented Reality for Orthopedic and Trauma Surgery

Daniel Adler

April 21, 2016

1 Project

Background: Orthopedic surgeries are often time intensive and require the correct placement of different tools inside of the human body. Within these surgeries, surgeons use cone beam computer tomography (CBCT) to ensure that they are correctly guiding objects into the damaged area of the human body. This process often requires hundreds of scans, which lengthens procedural time and exposes the physician and patient to high radiation dosages.

Work has been done to develop a mixed-reality visualization that allows physicians to see in real-time if they are correctly placing objects. A manual calibration algorithm has been developed to create a mixed-reality visualization by performing a one-time calibration between a 3D Red-Green-Blue Depth (RGBD) camera and the CBCT scanner [2]. It is this project's goal to create a standalone automated calibration algorithm between the CBCT and RGBD scanner with minimal dependencies.

Paper Selection: During the registration between the RGBD and CBCT point clouds, common algorithms such as Iterative Closest Point (ICP), do not converge to the global error minimum and instead converge to a local minimum. The following paper details an algorithm that utilizes geometric features to find an initial registration (rotation matrix and translation vector) between two 3D Point Clouds that places the registration in the global minimum area. This algorithm is currently utilized in the manual calibration algorithm developed for this project, but it is time intensive in the current pipeline, and studying this algorithm might lead to future improvements.

2 Significance

Problem: 3D point cloud to point cloud registration algorithms are used to align two different views of an object in space. This alignment is performed by finding the best transformation, i.e. a rotation and translation, to transform one point cloud into another. Many optimization techniques can be utilized to find the transformation that minimizes the distance between respective points within two clouds. Often times these algorithms fail because they become trapped in a local minimum and converge when a better solution exists. One example of this is the popular Iterative Closest Point Algorithm [1]. ICP has had many improvements, but still often falls into local minimum trap.

Key Result: This paper proposes a new algorithm using fast point feature histograms (FPFH) to perform an initial alignment that places a registration into the correct local minimum space, and thus the registration can be fine-tuned with an algorithm such as ICP. The algorithm builds upon previous work of the author that proposed a novel way to analyze geometric features of a point cloud, called point feature histograms (PFH) [4].

3 Background

Point Feature Histograms (PFH): Point Feature Histograms (PFH) are previously proposed pose-invariant local features that are based upon surface model properties of a point, p in a cloud. Their computation is based upon the 3D point coordinates (x, y, z) and the surface normal at that point, as well as relevant information from p's k nearest neighbors. To compute the PFH of a point, a radius r is chosen, and all neighbors to a point p within distance $\leq r$ are selected. For each pair of points $p_i, p_j, i \neq j$ in this neighborhood with normals n_i, n_j, the following frame (u, v, n) is calculated as follows:

$$u = n_i$$

$$v = (p_j - p_i) \times u$$

$$w = u \times v$$

and from these the following angular variations of the surface normals are found:

$$\alpha = v \cdot n_j$$

$$\phi = \frac{(u \cdot (p_j - p_i))}{||p_j - p_i||}$$

$$\theta = arctan(w \cdot n_j, u \cdot n_j)$$

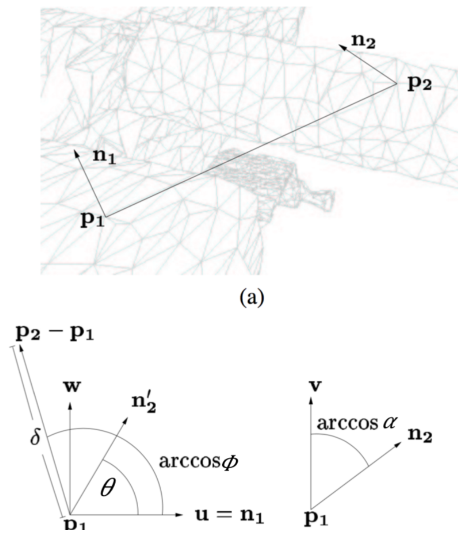

Figure 1: Angle metrics that are invariant under rigid transformations

Persistence: A large issue with using PFH is that for a point cloud P a lot of memory is required to extract the PFH of each point in the cloud. The paper chooses to concentrate on prominent points throughout a cloud, and disregard points that are dominant throughout an entire dataset (i.e. points that have many similar PFH to it). This analysis is called a *persistence analysis*, and it is performed by finding the mean PFH of a dataset, μ and only selecting less common points outside of some range $\mu \pm \beta\sigma$ for some user-defined parameter β.

It is possible that density variations in points might affect how a PFH performs over different radii r_i, r_j. Thus, points are chosen such that given two different radii, r_i, r_{i+1}, the points appear persistent over both radii. Thus, if P_{f_i} signifies the set of persistent points at radii r_i, the full set of persistent points can be written as:

$$P_f = \bigcup_{i=1}^{n-1} [P_{f_i} \cap P_{f_{i+1}}]$$

Figure 2: PFH persistence of varying radii from r = 0.003-0.005 cm as well as overall points

Geometric Signatures: The PFH space can be analyzed in two ways. (1) it can dictate informative information about the specific point cloud of a certain geometric object, and (2) it can be shown to have discriminatory power between different geometric objects. To prove the discriminatory power of PFH, five different convex surfaces were analyzed. The following confusion matrices showed the different mean histograms of the different shapes:

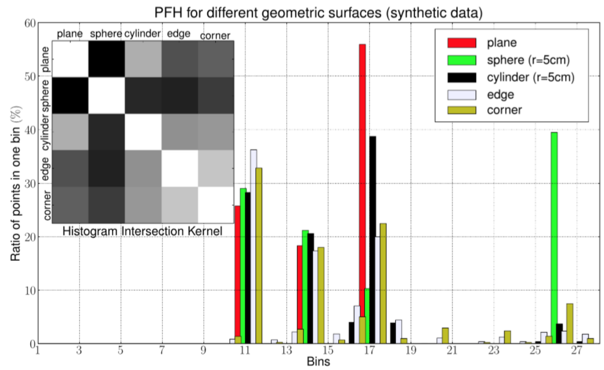

Figure 3: PFH and discrimination between different geometric shapes

Caching and Point Ordering: In addition to using persistence analysis, other tools can be utilized to reduce the amount of memory to compute a point's PFH. Note that while calculating the PFH of two neighboring points p and q, it is likely that many neighbors of p are also neighbors of q. Thus, a cache can be utilized to help with fast recall of recently computed PFHs. Using a cache and nearest neighbor tree to sort points actually reduced runtime to 75% of its original value.

4 Methods

Fast Point Feature Histograms (FPFH): In a point cloud with n points, each with k neighbors, the computational complexity of creating a PFH is $O(n \cdot k^2)$. This, the paper proposes using a simplified histogram computation, called a Fast Point Feature Histogram (FPFH) to reduce computational space to $O(n \cdot k)$.

The basic idea for computing an FPFH for a given point p in a cloud is to compute the relationships just between a point p and its neighbors, not every pair of neighbors. This will be called a Simplified Point Feature Histogram (SPFH). The paper then presents weighting the final features for a point p by using the SPFH and the corresponding distance, w_k, between a point p and its neighbor's SPFH p_k:

$$FPFH(p) = SPFH(p) + \frac{1}{k}\sum_{i=1}^{k} \frac{1}{w_k} \cdot SPFH(p_k)$$

Persistence: A persistence analysis was conducted using FPFHs on the same phantom as before. Most of the discriminating power was retained, but some small details were lost. Using the same geometric figures as before, the power of the FPFH was then tested. The uncorrelated histograms lost their informative power, but the confusion matrix showed that the FPFH could still discriminate between objects.

Figure 4: FPFH persistence of varying radii from r = 0.003-0.005 cm as well as overall points

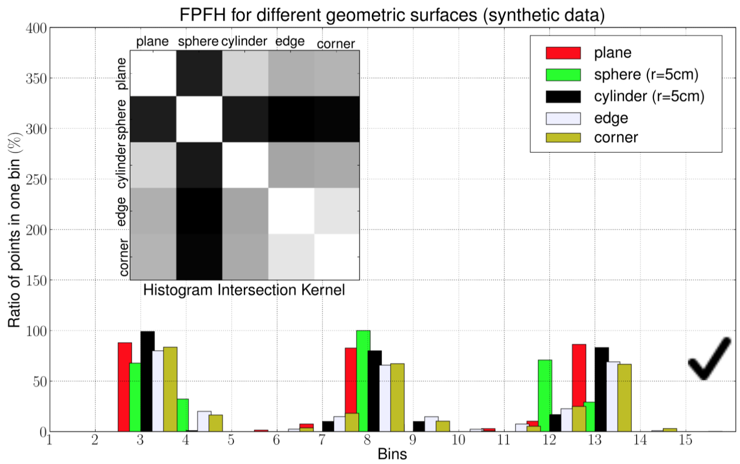

Figure 5: FPFH and discrimination between different geometric shapes

Online Implementation: The paper then describes one useful online implementation of the algorithm - for single sweep medical scans, like a 2D X-Ray. The algorithm simply maintains a list of points in a queue, for each a point a list of its k nearest neighbors. As soon as a new scan does not affect a certain point's list of neighbors, the point is processed.

4.1 Sample Consensus Initial Alignment (SAC-IA)

Lastly, FPFH's are applied to an algorithm for 3D-3D registration, called the Sample Consensus Initial Alignment (SAC-IA) method. The algorithm essentially:

1. Selects sample points s from a point cloud P that are at least a certain distance from each other, specified by a user as d_{min}.
2. For each sample point, a list of points in another cloud Q are selected that are similar to this point's histogram.
3. Rigid transformations are calculated between the sample points and their corresponding points in the other cloud. The error metric (Huber Penalty, L_h) defined below is utilized.

$$L_h(e_i) = \begin{cases} \frac{1}{2}e_i^2 & ||e_i|| \leq t_e \\ \frac{1}{2}t_e(2||e_i|| - t_e) & ||e_i|| > t_e \end{cases}$$

5 Results

Experimental Results: The paper compares the results of SAC-IA with a previously proposed Greedy Initial Alignment (GIA) algorithm that searches for every point in a dataset. SAC-IA outperformed the GIA with respect to speed and the number of points considered in this time period. No metrics were provided on the error of using SAC-IA with FPFH.

Conclusions: FPFH proves a good solution to finding an initial alignment of two point clouds, and can thus be refined using another algorithm after the initial alignment is found. The authors propose that they will investigate the robustness of the algorithm with noisier data in the future, and to learn about other uses for FPFH, like fast scene segmentation.

	GIA - run 1	GIA - run 2	SAC-IA
run time	> 17 min	> 43 min	34 sec
# points considered	200	250	10462
# of combinations	37186	58300	1000

Figure 6: GIA vs. SAC-IA runtime and points considered using FPFH

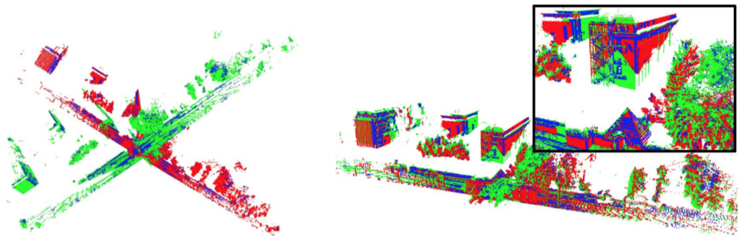

Figure 7: Alignment of an outdoor dataset using SAC-IA with FPFH

6 Review

Thoughts: The paper provided an adequate conceptual understanding of using FPFH, and its advancements compared to its slower predecessor, PFH. It proves why using features might be a good place for an initial alignment for two points clouds, and thus offers slight comparison to other existing alignment algorithms that have been traditionally used.

Where the paper lacks information is in the results section. The paper does a quick comparison of an old algorithm (GIA) compared to a new algorithm SAC-IA using FPFH. It does not compare the accuracy and precision of using FPFH for alignment, which is a large contributor for determining whether FPFH is a good algorithm to use. Thus, there is really only a minimal and non-convincing validation for using FPFH for image registration. It also is not consistent with its acronyms (SPFH vs SPF) and spent about half of the paper describing background knowledge instead of proving uses for its methodology.

Project Relevance: In our maximum deliverable, our goal is to find an alternative to using FPFH. Therefore, the purpose of studying this paper was to learn why FPFH might not be a good enough choice for our project. Unfortunately, the paper only mentioned a single example to when FPFH was utilized, and this example had a short runtime. Thus, more experiments will have to be conducted to see why SAC-IA is slow.

References

[1] Besl PJ, McKay ND. A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, February 1992.

[2] Lee SC, Fuerst B, Fotouhi J, Fischer M, Osgood G, Navab N. Calibration of RGBD Camera and Cone-Beam CT for 3D Intra-operative Mixed Reality Visualization. International Journal of Computer Assisted Radiology and Surgery / International Conference on Information Processing in Computer-Assisted Interventions (IPCAI), Heidelberg, June 2016.

[3] Rusu RB, Cousins S. 3D is here: Point Cloud Library (PCL). IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 2011.

[4] Rusu, RB, Marton ZC, Blodow N, Beetz M. Peristent Point Feature Histograms for 3D Point Clouds. Intelligent Autonomous Systems, Munich, Germany, December 2007.

[5] Rusu RB, Blodow N, Beetz M. 2009. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the 2009 IEEE international conference on Robotics and Automation (ICRA'09). IEEE Press, Piscataway, NJ, USA, 1848-1853.

[6] Wahl E, Hillenbrand U, Hirzinger G. Surflet-Pair-Relation Histograms: A Statistical 3D-Shape Representation for Rapid Classification. IEEE 3-D Digital Imaging and Modeling, Banff, Alta, October 2003.