Fast Point Feature Histograms (FPFH) for 3D Registration

Dan Adler
Dr. Nassir Navab, Bernhard Fuerst, Sing Chun Lee, Javad Fotouhi, Tiffany Chung
Project 6
Project Background and Objective

Orthopaedic surgeries are time intensive and require multiple images to ensure correct placement and direction of tools.

Research has been done to create a manual calibration algorithm, that creates an intra-operative mixed-reality visualization.

Goal is to automate the calibration process between CBCT scanner and RGBD camera.
Paper Selection - from PCL Library


RGBD and CBCT calibration fails to converge to global minimum

Look for a different registration approach (feature based)

Manual calibration algorithm developed for this project uses this algorithm

Improve usage of this algorithm, or find a comparable algorithm for an initial registration
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 (rotation and translation).

Many typical optimization techniques can be utilized to find the transformation that minimizes the distance between respective points in the two clouds.

Often times these algorithms fail because they become trapped in a local minimum and converge even though a better solution exists.
Key Result and Significance

This paper proposes a new algorithm using fast point feature histograms (FPFH) to perform an initial alignment that places a registration into the correct global minimum space.

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).

FPFH computations are faster compared than PFH computations ($O(k)$ vs. $O(k^2)$) to build per point.

Provide implementation results for 3D point-cloud to point-cloud registration, using a sample consensus initial alignment algorithm (SAC-IA).
Point Feature Histograms (PFH)

Point Feature Histograms (PFH) are previously proposed pose-invariant local features

Point, \( p \) in a cloud and the surface normal, \( n \), at that point, as well as relevant information from \( p \)'s nearest neighbors

Radius \( r \) is chosen, and all neighbors to a point \( p \) with distance \( \leq r \) are selected. Suppose uniform density, and thus about \( k \) neighbors per point

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:

\[
\begin{align*}
u &= n_i \\
v &= (p_j - p_i) \times u \\
w &= u \times v
\end{align*}
\]
Point Feature Histograms (PFH)

From these the following **angular variations** of the surface normals are found:

\[
\begin{align*}
  u &= n_i \\
  v &= (p_j - p_i) \times u \\
  w &= u \times v \\
  \alpha &= v \cdot n_j \\
  \phi &= \frac{u \cdot (p_j - p_i)}{\|p_j - p_i\|} \\
  \theta &= \arctan(w \cdot n_j, u \cdot n_j)
\end{align*}
\]
Persistence

$O(k^2)$ pieces of information per point $p$

Disregard points with similar PFHs

Finding the mean PFH of a dataset, $\mu$ and only selecting less common points outside of some range $\mu \pm \beta \sigma$ for some user-defined $\beta$

Density variations in points might affect how a PFH performs over different radii $r_i, r_j$

If $P_{f_i}$ signifies the set of persistent points at radii $r_i$, the full set of persistent points can be written as:

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

PFH for different geometric surfaces (synthetic data)

Histogram Intersection Kernel
Caching and Point Ordering
Fast Point Feature Histograms

$n$ points in a cloud, each with about $k$ neighbors: $O(n*k^2)$ to $O(n*k)$ computation

SPFH = Simplified Point Feature Histogram

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

Contribution from $p$ and neighbors $\leq r$

Contribution from $p$'s neighbors (points $\leq 2r$)
FPFH Geometric Signatures

[Graph showing FPFH for different geometric surfaces (synthetic data).]
Persistence Analysis

Increasing radii from 0.003 - 0.005 cm
Online Implementation

Scanline:  

\[ r = 1 \]
Online Implementation

Scanline: 1 2 ... r = 1

Queue: ...
Online Implementation

Scanline:

1 2 ... r = 1

Queue:

Neighbor Lists:
Online Implementation

Scanline: 1 2 ... r = 1

Neighbor Lists:

New Scanline:

Neighbor list of  will not be modified!
Sample Consensus Initial Alignment: SAC-IA

For point cloud initial registration

1. Select sample points $s$ from a point cloud $P$ that are at least a certain distance from each other, specified by a user as $d_{\text{min}}$
2. For each $s$, find a list of points in another cloud $Q$ with a similar FPFH
3. Calculate a rigid transformation between points

The error metric defined is often used for the error between surfaces, and it is the **Huber Penalty** measure, $L_h$. The transformation with the lowest $L_h$ is kept, and after a set number of iterations the algorithm terminates.

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

Not specified how to choose $t_e$ linear/quadratic boundary
Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>GIA - run 1</th>
<th>GIA - run 2</th>
<th>SAC-IA</th>
</tr>
</thead>
<tbody>
<tr>
<td>run time</td>
<td>&gt; 17 min</td>
<td>&gt; 43 min</td>
<td>34 sec</td>
</tr>
<tr>
<td># points considered</td>
<td>200</td>
<td>250</td>
<td>10462</td>
</tr>
<tr>
<td># of combinations</td>
<td>37186</td>
<td>58300</td>
<td>1000</td>
</tr>
</tbody>
</table>

Datasets of outdoor scenes from a previous paper

GIA = Greedy Alignment Algorithm from previous work with FPFH
   Searches through all possible correspondence pairs

SAC-IA: Considered more points, less combinations and in faster time
Conclusions

Good solution to finding an initial alignment to two point clouds, and can thus be refined using another algorithm (paper uses Levenberg-Marquadt from CIS 1)

Authors propose that they will investigate the robustness of the algorithm with noisier data in the future

Other uses for FPFH, like fast scene segmentation
Thoughts on Paper

Pros

- Presents a solid conceptual understanding of using FPFH, and its advancements compared to its slower predecessor, PFH
- Proves why using features might be a good place for an initial alignment for two points clouds

Cons

- Quick comparison of an old algorithm (GIA) compared to a new algorithm (SAC-IA) using FPFH
- 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
- Some confusing notation (utilizing SPF vs. SPFH)
- Too much reliance on old papers (decorrelation, histograms, half the paper was background)
Relevance
Questions?
Reading List


