Dynamic Geometry Processing

EG 2012 Tutorial
Dynamic Registration

Niloy J. Mitra
University College London
Scan Registration
Scan Registration

Solve for inter-frame motion:

\[ \alpha := (R, t) \]
Scan Registration

Solve for inter-frame motion:

$$\alpha_j := (R_j, t_j)$$
The Setup

Given:

A set of frames \( \{P_0, P_1, \ldots, P_n\} \)

Goal:

Recover rigid motion \( \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \) between adjacent frames
The Setup

Smoothly varying object motion

Unknown correspondence between scans

Fast acquisition →
motion happens between frames
Insights

Rigid registration $\rightarrow$ kinematic property of space-time surface (locally exact)

Registration $\rightarrow$ surface normal estimation

Extension to deformable/articulated bodies
Time Ordered Scans

\[ \tilde{P}^j \equiv \{ \tilde{p}^j_i \} := \{(p^j_i, t^j), p^j_i \in \mathbb{R}^d, t^j \in \mathbb{R}\} \]
Space-time Surface
Space-time Surface

\[ \tilde{p}_i^j \rightarrow \tilde{\alpha}_j(\tilde{p}_i^j) = (R_j p_i^j + t_j, t_j^j + \Delta t^j) \]
Space-time Surface

\[ \tilde{\alpha}_j (\tilde{p}_i^j) = \left( R_j p_i^j + t_j, t_j^j + \Delta t_j^j \right) \]

\[ \tilde{\alpha}_j = \arg\min_{i=1}^{P_j} d^2 (\tilde{\alpha}_j (\tilde{p}_i^j), S) \]
Spacetime Velocity Vectors

Tangential point movement $\rightarrow$ velocity vectors orthogonal to surface normals

$$\tilde{\alpha}_j = \arg\min_{\alpha_j} \sum_{i=1}^{P^j} d^2(\tilde{\alpha}_j(p^i), S)$$
Spacetime Velocity Vectors

Tangential point movement → velocity vectors orthogonal to surface normals
Final Steps

(rigid) velocity vectors \( \rightarrow \)

\[ \tilde{v}(\tilde{p}_i^j) = (c_j \times p_i^j + \bar{c}_j, 1) \]

\[
\min_{c_j, \bar{c}_j} \sum_{i=1}^{|P^j|} w_i^j \left[ (c_j \times p_i^j + \bar{c}_j, 1) \cdot \tilde{n}_i^j \right]^2
\]
Final Steps

(rigid) velocity vectors!

\[
\tilde{v}(\tilde{p}_i^j) = (c_j \times p_i^j + \bar{c}_j, 1)
\]

\[
\min_{c_j, \bar{c}_j} \sum_{i=1}^{|P^j|} w_i^j \left[ (c_j \times p_i^j + \bar{c}_j, 1) \cdot \tilde{n}_i^j \right]^2
\]

\[
Ax + b = 0
\]

\[
A = \sum_{i=1}^{|P^j|} w_i^j \begin{bmatrix} \tilde{n}_i^j \\ p_i^j \times \tilde{n}_i^j \end{bmatrix} \begin{bmatrix} \tilde{n}_i^j & (p_i^j \times \tilde{n}_i^j)^T \end{bmatrix}
\]

\[
b = \sum_{i=1}^{|P^j|} w_i^j n_i^j \begin{bmatrix} \tilde{n}_i^j \\ p_i^j \times \tilde{n}_i^j \end{bmatrix}
\]

\[
x = \begin{bmatrix} \bar{c}_j \\ c_j \end{bmatrix}
\]
Registration Algorithm

1. Compute time coordinate spacing ($\sigma$), and form space-time surface.

2. Compute space-time neighborhood using ANN, and locally estimate space-time surface normals.

3. Solve linear system to estimate ($c_j$, $\bar{c}_j$).

Normal Estimation: PCA Based

Plane fitting using PCA using chosen neighborhood points.
Normal Estimation: Iterative Refinement

Update neighborhood with current velocity estimate.
Normal Refinement: Effect of Noise

Stable, but more expensive.
Normal Estimation: Local Triangulation

Perform local surface triangulation (tetrahedralization).
Normal Estimation

Stable, but more expensive.
Comparison with ICP

ICP point-plane

Dynamic registration
Rigid: Bee Sequence (2,200 frames)

Bee Input frames (Selection)

2200 pointclouds scanned at 17 Hz transformations only for adjacent frames considered no global error correction no noise smoothing
Rigid: Coati Sequence (2,200 frames)

Coati

Input frames (Selection)

2200 pointclouds scanned at 17 Hz
transformations only for adjacent frames considered
no global error correction
no noise smoothing
Handling Large Number of Frames
Rigid/Deformable: Teapot Sequence (2,200 frames)

Teapot Input frames (Selection)

2200 pointclouds scanned at 17 Hz
transformations only for adjacent frames considered
no global error correction
no noise smoothing
Deformable Bodies

\[
\min_{c_j, \bar{c}_j} \sum_{i=1}^{|P^j|} w_i^j \left[ (c_j \times p_i^j + \bar{c}_j, 1) \cdot \tilde{n}_i^j \right]^2
\]

Cluster points, and solve smaller systems.

Propagate solutions with regularization.
Deformable: Hand (100 frames)

Hand

Input frames & registered result

100 pointclouds scanned at 17 Hz
transformations only for adjacent frames considered
no global error correction
no noise smoothing

first frame is tracked
defformation due to severely missing data (e.g. ring finger)
Deformable: Hand (100 frames)

scan #1 : scan #50

scan #1 : scan #100
Deformation + scanner motion: Skeleton (100 frames)

Grasp  Input frames & registered result

100 simulated scan data sets
simultaneous object deformation and camera motion
transformations only for adjacent frames considered
no global error correction, no noise smoothing

first frame is tracked
data completion (e.g. for middle finger)
Deformation + scanner motion: Skeleton (100 frames)

scan #1 : scan #50

scan #1 : scan #100
Deformation + scanner motion: Skeleton (100 frames)

rigid components
## Performance
(on 2.4GHz Athlon Dual Core, 2GB RAM)

<table>
<thead>
<tr>
<th>Model</th>
<th># scans</th>
<th># points/scan (in 1000s)</th>
<th>Time (mins)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bunny (simulated)</td>
<td>314</td>
<td>33.8</td>
<td>13</td>
</tr>
<tr>
<td>bee</td>
<td>2,200</td>
<td>20.7</td>
<td>51</td>
</tr>
<tr>
<td>coati</td>
<td>2,200</td>
<td>28.1</td>
<td>71</td>
</tr>
<tr>
<td>teapot (rigid)</td>
<td>2,200</td>
<td>27.2</td>
<td>68</td>
</tr>
<tr>
<td>skeleton (simulated)</td>
<td>100</td>
<td>55.9</td>
<td>11</td>
</tr>
<tr>
<td>hand</td>
<td>100</td>
<td>40.1</td>
<td>17</td>
</tr>
</tbody>
</table>
Conclusion

Simple algorithm using kinematic properties of space-time surface.

Easy modification for deformable bodies.

Suitable for use with fast scanners.
Limitations

Need more scans, dense scans, ...

Sampling condition $\rightarrow$ time and space
thank you