Automatic Registration for Articulated Shapes

Problem Statement
- Solve pairwise registration problem
- Develop robust method independent of initial pose
- Do not require markers or a template
- Contributions:
  - Useful for initialization: used as preprocessing step
  - Focus on registration: does not solve for a reduced motion model

Related Work
- Correlated correspondence algorithm, requires a template (Anguelov et al. 2004)

Algorithm Overview
- Articulated motion → small set of transformations
- Predetermine a set of transformations describing the motion
- Optimize assignment of transformations to the points
Algorithm Description

Motion Sampling Illustration
- Find transformations that move parts of the source to parts of the target

Sampled Points

Source Shape  Target Shape

Source Shape  Target Shape
Motion Sampling Illustration

- Find transformations that move parts of the source to parts of the target

![Diagram showing motion sampling illustration with source shape, target shape, and transformation space.]

Limitations of Motion Sampling

- **Final Output:** finite set of rigid transformations
- If there are multiple similar parts
  - Does not figure out the correct part
  - Disambiguate in the optimization step

![Diagram showing source with selected region, candidate transformations, and visualized transformations.]

Algorithm Description

- Motion Sampling
- Global Motion Optimization

![Diagram showing algorithm description with motion sampling and global motion optimization.]
Global Motion Optimization

- Optimize an assignment from a finite set of transformations

\[ \text{argmin} \quad \text{Data Cost} + \text{Smoothness Cost} \]

- A discrete labelling problem → Graph Cuts for optimization

Data Term

- Move all points as close as possible to the target
- How to measure distance to target?
  - Apply selected transformation \( f_p \) for all \( p = f_p(p) \)
  - Measure distance to closest point \( u \) in target

Smoothness Term

- Preserve edge length between neighboring points
  \[ V(p, q, f_p, f_q) = \left| \|p - q\| - \|f_p(p) - f_q(q)\| \right| \]
- Disambiguates multiple possible mappings

Symmetric Cost Function

- Swapping source / target can give different results
  - Optimize assignment in both meshes (forward & backward)
  - Enforce consistent assignment: penalty when \( f_p \neq f_u \)
Optimization Using Graph Cuts

\[
\text{argmin} \quad \text{Data}_{\text{Source}} + \text{Smoothness}_{\text{Source}} + \\
\text{Data}_{\text{Target}} + \text{Smoothness}_{\text{Target}} + \\
\text{Symmetric Consistency}_{\text{Source} & \text{Target}}
\]

- Data and smoothness terms apply to both shapes
- Additional symmetric consistency term
- Weights to control relative influence of each term
- Use “graph cuts” to optimize assignment
  - [Boykov, Veksler & Zabih PAMI ’01]

Results

Horse Dataset
Arm Dataset
Hand Dataset

Horse Dataset Results

12 poses of galloping horse: total of 66 pairs, correct leg matched in 64 pairs

Histogram of Error in Galloping Horse Dataset (minimum over 3 trials)

Synthetic Dataset Example

1.5% Registration Error

Motion Segmentation (from Graph Cuts)
Synthetic Dataset w/ Holes

Source

Target

Aligned Result

Distance (from Target) to the closest point (% bounding box diagonal) 5.3%

0%

Arm Dataset Results

12 poses of arm scans: total of 66 pairs, arm & hand orientation matched in all pairs

Histogram of Error in the Arm dataset (1 trial)

Arm Dataset Example

Source

Noisy Target

Aligned Result

Distance (from Target) to the closest point (% bounding box diagonal) 5.4%

0%
Hand Dataset Example

Performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Points</th>
<th># Labels</th>
<th>Matching</th>
<th>Clustering</th>
<th>Pruning</th>
<th>Graph Cuts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horse</td>
<td>8431</td>
<td>1500</td>
<td>2.1 min</td>
<td>3.0 sec</td>
<td>(skip)</td>
<td>1.6 sec</td>
</tr>
<tr>
<td>Arm</td>
<td>11865</td>
<td>1000</td>
<td>55.0 sec</td>
<td>0.9 sec</td>
<td>12.4 min</td>
<td>1.2 sec</td>
</tr>
<tr>
<td>Hand (Front)</td>
<td>8339</td>
<td>1500</td>
<td>14.5 sec</td>
<td>0.7 sec</td>
<td>7.4 min</td>
<td>1.2 sec</td>
</tr>
<tr>
<td>Hand (Back)</td>
<td>6773</td>
<td>1500</td>
<td>17.3 sec</td>
<td>0.9 sec</td>
<td>9.4 min</td>
<td>1.6 sec</td>
</tr>
</tbody>
</table>

- Graph cuts optimization is most time-consuming step
  - Symmetric optimization doubles variable count
  - Symmetric consistency term introduces many edges

Limitations

- Errors in registration
  - Trade-off between data and smoothness costs
    - Data weight too high → May break smoothness
    - Smoothness weight too high → Prefer bad alignment
Limitations

- Errors in registration
  - Motion sampling: may fail to sample properly when too much missing data, non-rigid motion
  - Hard assignment of transformations

Conclusions

- Automatic method for registering articulated shapes
  - No template, markers, or manual segmentation needed
  - Explicitly sample a discrete set of motion
  - Optimize the assignment of transformations
  - Graph cut result gives intuitive segmentation

- Useful for obtaining a robust initialization of the registration
  - Does not provide an articulated motion model

Problem Statement

- Fit a model of the surface motion to a pair of scans
  - Articulated model (e.g. joints, smooth weights)
  - Serves as the basis for fitting on multiple frames

Range Scan Registration Using Reduced Deformable Models
Related Work

- User provided segmentation: Pekelny08
- Unsupervised pairwise registration: Lio8, Huango8

Problem Formulation

Model: Linear Blend Skinning

- Transformations (bones) and weights
  - Each point assigned weights in reference pose
  - Transformations move each point according to its weights

Shape

<table>
<thead>
<tr>
<th>Shape with Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bone 1</td>
</tr>
<tr>
<td>Bone 2</td>
</tr>
</tbody>
</table>

Weighted Blending Result

<table>
<thead>
<tr>
<th>Weighted Blending Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bone 1</td>
</tr>
<tr>
<td>Bone 2</td>
</tr>
<tr>
<td>[1,0]</td>
</tr>
<tr>
<td>[0.5, 0.5]</td>
</tr>
<tr>
<td>[0,1]</td>
</tr>
</tbody>
</table>
Weight Grid

- Define weights on grid enclosing surface
  - Covers small holes, reduces variables
  - Provides regular structure for optimization

LBS for scan registration

- Fit the transformations and weights to align a pair of range scans

Algorithm Description

Initialization → Main Optimization Loop → Weight Refinement → Final Result
Optimization strategy

Optimization overview

T-Step: Distance Term
- Fix weights & solve for transformations

T-Step: Optimize Alignment
- Distance Term
- Joint Constraint Term
T-Step: Distance Term

- Fix weights & solve for transformations
  - Use closest point correspondences

T-Step: Distance Term

- Fix weights & solve for transformations
  - Use closest point correspondences

T-Step: Distance Term

- Fix weights & solve for transformations
  - Use closest point correspondences
  - Iterate further until convergence

T-Step: Joint Constraint Term

- Prevent neighboring bones from separating
T-Step: Joint Constraint Term
- Prevent neighboring bones from separating
  - Constrain overlapping weight regions

Bone 1  Bone 2  Bone 3

Unwanted stretch

T-Step: Joint Constraint Term
- Prevent neighboring bones from separating
  - Constrain overlapping weight regions

Bone 1  Bone 2  Bone 3

T-Step: Optimization summary
- Like rigid registration
  - Except multiple parts & joint constraints
- Non-linear least squares optimization
  - Solving for a rotation matrix
  - Gauss-Newton algorithm
  - Solve by iteratively linearizing solution
- Few variables $\rightarrow$ Fast performance
  - $\#$ variables = 6 x $\#$bones
  - Typically 5-10 bones in our examples
Optimization overview

 Initialization → Main Optimization Loop → Weight Refinement → Final Result

Main Optimization Loop

- **W-Step**: Optimize Weights
  - Use Discrete Labelling
  - Continuous Weight Refinement

W-Step: Optimizing weights

- Fix transformations, solve for continuous weights

Correspondences from last T-Step

Bone 1 (Applied to entire shape)
W-Step: Optimizing weights

- Fix transformations, solve for continuous weights

Good Alignment

Bone 2
(Applied to entire shape)

W-Step: Optimizing weights

- Fix transformations, solve for continuous weights

Good Alignment

Bone 3
(Applied to entire shape)

W-Step: Optimizing weights

- Fix transformations, solve for continuous weights

Bone 1 & 3
Bone 2 & 3
Bone 1 & 2
Bone 1
Bone 2
Bone 3

“Ideal” solved result

W-Step: Optimizing weights

- Without additional constraints, problem is underconstrained

Bone 1 & 3
Bone 2 & 3
Bone 2 & 3

Typical solved result
Use discrete labeling

- **Our solution**: one transformation per location
  - Bones = labels
  - Becomes discrete labeling problem

W-Step: Optimization Summary

- Use “graph cuts” to optimally label grid cells
  - [Boykov, Veksler & Zabih PAMI ’01]
- Distance term + Smoothness term
  - Distance: measures alignment for a given label
  - Smoothness: penalizes different labels for adjacent cells
- Good Performance
  - Only ~ 1000 grid cells (graph nodes) in our examples
  - Fast performance for graph cuts

Results

- Robot, torso video
- Interactive posing video
- Additional results & statistics
### Torso video (2x speed recording)

- **Alignment Result**
  - 7 bones
  - 4890 cells

- **Solved Weights**

### Interactive posing (real-time recording)

- **Solved Weights**
  - (7 bones, 1598 cells)

- **Interactive Posing Result**

### Average performance statistics

<table>
<thead>
<tr>
<th></th>
<th>Car</th>
<th>Robot</th>
<th>Walk</th>
<th>Hand</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bones</strong></td>
<td>7</td>
<td>7</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td><strong>Corresp.</strong></td>
<td>1200</td>
<td>1200</td>
<td>1000</td>
<td>1500</td>
</tr>
<tr>
<td><strong>Vertices</strong></td>
<td>5389</td>
<td>9377</td>
<td>4502</td>
<td>34342</td>
</tr>
<tr>
<td><strong>Max Dist</strong></td>
<td>20</td>
<td>40</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td><strong>Grid Res</strong></td>
<td>60</td>
<td>65</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td><strong>Grid Cells</strong></td>
<td>1107</td>
<td>1295</td>
<td>1014</td>
<td>814</td>
</tr>
<tr>
<td><strong>Grid Points</strong></td>
<td>2918</td>
<td>3366</td>
<td>2553</td>
<td>1884</td>
</tr>
<tr>
<td>Setup</td>
<td>0.185 sec</td>
<td>0.234 sec</td>
<td>0.136 sec</td>
<td>0.078 sec</td>
</tr>
<tr>
<td>RANSAC</td>
<td>8.089 sec</td>
<td>20.001 sec</td>
<td>5.517 sec</td>
<td>N/A</td>
</tr>
<tr>
<td>Align</td>
<td>9.945 sec</td>
<td>19.644 sec</td>
<td>23.092 sec</td>
<td>49.918 sec</td>
</tr>
<tr>
<td>Weight</td>
<td>6.135 sec</td>
<td>10.713 sec</td>
<td>10.497 sec</td>
<td>3.689 sec</td>
</tr>
<tr>
<td><strong>Total Time</strong></td>
<td>24.355 sec</td>
<td>50.591 sec</td>
<td>39.242 sec</td>
<td>53.684 sec</td>
</tr>
</tbody>
</table>

### Limitations

- **Discussion**
  - Topology issues with grid
    - Improve in next section using graph-based approach
  - Limited to a pair of scans
    - Simultaneously register multiple frames in next section
  - Limitations with LBS
    - Optimize better model (e.g. DLB)
Conclusion

- A new algorithm to align range scans by modeling the motion with a reduced deformable model
  - Use LBS to represent the motion
  - Represent weight function using a 3D grid
  - Solve for the parameters using alternating optimization
  - No marker, template, segmentation information
  - Robust to occlusion & missing data

- Next: extend this method to handle multiple frames