3D Pipeline
Segmentation (planar)
Registration (using lines)

3D PHOTOGRAPHY EXAMPLE

Automatic registration.
Each scan has a different color.

Registration details
Problem

- Computer modeling
  - Serious bottleneck
- 3D details of existing structures
  - Virtually impossible to model by hand
- Existing methods are deficient
  - Require extensive human interaction

Planar Segmentation Result: From Points to 3D Planar Polygons

Intersection line.

Each Segmented Planar Area is shown with different color for clarity.

Range sensing direction.

Exterior border.

Segmented Planar Area. Exterior and Interior borders shown.

Interior border.
More Segmentation Results

Two segmented range scans of building

Approach

1. Acquire range scans of scene
   → each scan is an array of depth values
2. Recover planar segments and linear features
3. Align scans together (3D-3D registration)
   → form single point cloud of scene
4. Generate mesh model
5. Acquire color photos of scene
6. Register photos with mesh (2D-3D registration)
7. Perform texture mapping
Feature-based range-range registration

- Pairwise registration between two scans.
- Automated method uses linear features.
- Features extracted at boundaries of SPAs.
- Two correctly matched lines between scans provide solution.
- If scan A contains N lines and scan B contains M lines
  - Need to consider $O(N^2M^2)$ pairs.
  - For each pair verification of registration needed.
- Naïve method is time consuming.
- Two efficient novel algorithms developed.
- Problems also induced by scene symmetry.
- User-interface for smart user interaction developed.

One Pair of Correctly Matched Lines Provides Rotation

- Only orientation and location of lines used.
- Endpoints not-used.
- Closed-form solution provides rotation.
One Pair of Correctly Matched Lines Provides Rotation

- Only orientation and location of lines used.
- Endpoints not-used.
- Closed-form solution provides rotation.

• Right coordinate system is rotated about its origin to match left coordinate system.
One Pair of Correctly Matched Lines Provides Estimated Translation

- Estimated translation: vector that connects the midpoints of the two lines.
- It is not exact because endpoints are never accurately extracted.

Two Pairs of Correctly Matched Lines Provide Exact Translation

- Exact translation computation by using two pairs.
- Endpoints of lines are not used.
First efficient algorithm

- Problems to tackle:
  - Noise: Lines and normals do not match exactly.
  - Search space is large.
  - Verification of correct match expensive.

- Search for correct pairs of matched lines.
  - Search first for one pair.
  - Proceed to search for second pair.
  - Grade each computed transform: # of matches.
  - Keep the transform with the highest grade.
  - At the end refine best transform using all lines.

Exhaustive Search Approach

- Find the two pairs of corresponding lines that maximizes the total number of line matches
  - Consider two corresponding line pairs
  - Compute transformation
  - Grade of computed transform: total number of line matches
  - Keep the transform with the highest grade
  - Refine best transform using all matched lines

White lines (left scan)
Blue lines (right scan)
Red/Green lines (matches)
Flowchart

Pairwise Registrations: Line Matches

White lines (left scan)
Blue lines (right scan)
Red/Green lines (matches)
## Results

<table>
<thead>
<tr>
<th>Pair</th>
<th>Line Pairs</th>
<th>Pre (%)</th>
<th>S2 % (#)</th>
<th>S3 % (#)</th>
<th>Matches</th>
<th>t (sec)</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>306 x 306</td>
<td>16</td>
<td>1.7 (1555)</td>
<td>0.38 (346)</td>
<td>35</td>
<td>15</td>
<td>10.92mm</td>
</tr>
<tr>
<td>2</td>
<td>306 x 290</td>
<td>17</td>
<td>2.0 (2420)</td>
<td>0.04 (705)</td>
<td>25</td>
<td>20</td>
<td>0.22mm</td>
</tr>
<tr>
<td>3</td>
<td>290 x 317</td>
<td>21</td>
<td>2.8 (2572)</td>
<td>1.88 (1726)</td>
<td>36</td>
<td>52</td>
<td>2.77mm</td>
</tr>
<tr>
<td>4</td>
<td>317 x 190</td>
<td>19</td>
<td>3.4 (1355)</td>
<td>1.15 (656)</td>
<td>28</td>
<td>21</td>
<td>11.90mm</td>
</tr>
<tr>
<td>5</td>
<td>211 x 180</td>
<td>21</td>
<td>1.6 (1759)</td>
<td>2.1 (802)</td>
<td>31</td>
<td>19</td>
<td>9.20mm</td>
</tr>
<tr>
<td>6</td>
<td>180 x 274</td>
<td>17</td>
<td>2.6 (1900)</td>
<td>0.34 (168)</td>
<td>22</td>
<td>9</td>
<td>11.43mm</td>
</tr>
<tr>
<td>7</td>
<td>114 x 274</td>
<td>19</td>
<td>1.6 (507)</td>
<td>2.2 (894)</td>
<td>33</td>
<td>6</td>
<td>5.61mm</td>
</tr>
<tr>
<td>8</td>
<td>274 x 138</td>
<td>16</td>
<td>1.8 (667)</td>
<td>1.5 (557)</td>
<td>31</td>
<td>5</td>
<td>3.08mm</td>
</tr>
<tr>
<td>9</td>
<td>114 x 138</td>
<td>18</td>
<td>2.7 (423)</td>
<td>3.8 (593)</td>
<td>32</td>
<td>4</td>
<td>3.94mm</td>
</tr>
<tr>
<td>10</td>
<td>138 x 247</td>
<td>18</td>
<td>2.3 (794)</td>
<td>1.3 (429)</td>
<td>20</td>
<td>5</td>
<td>1.30mm</td>
</tr>
</tbody>
</table>

### Campus Building - Results (average error 7.4mm)

### Cathedral - Results (average error 17.3mm)
**Graph Search Global Registration**

- Create weighted graph of scans. Edges of graph are confidence in finding correct registration between pairs of scans.
- Confidence (cost) is number of correctly aligned lines after applying registration (R,T).
- Global Registration: find most robust path from pivot scan to each scan.

---

**Thomas Hunter Building: 10 scans**

Automatic Registration. Each scan has different color.

Registration Detail.
Problems with described algorithm

- Complexity is still high for
  - Large line-sets.
  - Symmetrical configurations.
- Symmetry can result to wrong registration.
New Framework

Lines and planes from segmentation

User interaction

Image1 line clustering

Rotation computation

Translation computation

Transformation refinement

Automated procedures

Image2 line clustering

Display registered pair

Wrong registration due to symmetry

Rotation adjustment

Translation adjustment

Next pair of scans

Correct registration

Exhaustive Search Approach

- No initial registration needed
- High computational complexity
- Symmetry problem unsolved

Improvements
- Extract object-based coordinate system
- Context-sensitive user interface
Framework of Interactive Solution

Lines and planes from segmentation

Image1 line clustering
Rotation estimation
Translation estimation
Transform refinement by ICP

Automated Registration

Image2 line clustering

Display registered pair
Rotation adjustment
Translation adjustment
Correct registration
Next pair of scans
Global stitching

User Interactions

Line Clustering

- Line clustering
  - Line directions
  - Plane normals

Building’s local coordinate system
Rotation Estimation

- Rotation estimation

\[ R = [x_2 \ y_2 \ z_2]^T \times [x_1 \ y_1 \ z_1] \]

24 possible \( R \)'s?

- Heuristic: eliminate candidates based on observations
  - Scanner moves on the ground plane: y axis not change much
  - Overlapping images from close by viewpoints: smallest rotation candidate is chosen

\[ x_1 \cdot R = x_2 \]
\[ y_1 \cdot R = y_2 \]
\[ z_1 \cdot R = z_2 \]

\( (0,1,0) \cdot R = R_{11} \) : projection of \( y_1 \) on \( y_2 \)

\[ R_{11} > \cos(45^\circ) = 0.7 \]

2 to 5 \( R \)'s

Return the \( R \) with the largest diagonal sum
Translation Estimation
- Translation estimation

- Left and right axes parallel accordingly after rotation
- Pick robust line pairs to estimate translation

Translation Estimation
- One pair of matched lines provides an estimated translation
- Two pairs with similar estimated translations provide translation candidate

Line pair 1
Line pair 2
Translation Estimation

- Two types of translation candidates

\[ T = \frac{(d_1 + d_2)}{2} \]

- Find the translation that maximizes the total number of line matches

- Cluster all estimated T's, pick 10 most frequently appeared T's

- For each T:
  - Find all matches, solve linear system to update R&T
  - Count matched line pairs again

- Choose the R&T with the most number of matched line pairs

- Refine transformation with ICP
Registration System Flowchart

start

Read in all image pairs, form the transformation graph

Read in one image pair

Last pair?

Automated registration

Display to user

Find pivot image, compute path

Compute transform from each image to pivot image

Global optimization

Global stitching

Display and save → exit

Rotation wrong by 90°

Display other rotations

Choose a new R, compute T

Correct Rotation & Translation

Rotation Symmetry by 90deg.

Need Manual Adjustment on R/T

Y

N

Manual adjustment

1. correct

2. Rotation wrong by 90°

Adjust R/T, optimize

Correct Rotation & Translation

Need Manual Adjustment on R/T

User Interface

Display window: Points and lines of registered two scans
User Interface

- Rotation wrong by 90 degrees: choose from other candidate rotations computed previously

User Interface

- Adjusting rotation and translation based on the building’s coordinate system
User Interface
Display window: Points and lines of registered two scans

Correct Rotation & Translation
Rotation Symmetry by 90deg.
Need Manual Adjustment on R/T

Exhaustive Search Approach
ICP Optimization

Clustering 3D lines provide major building’s directions

Two segmented scans with major directions shown
Utilizing user-interaction

Rotation Computation

• If we have correct match of axes:
  \[ R = [x_2 \ y_2 \ z_2]^T \ast [x_1 \ y_1 \ z_1] \]
• 24 candidate rotations exist if match of axes is unknown.
• Heuristic produces the more plausible candidate.
• If wrong the user can select from a set of candidates.
Translation Computation

- Search for two pairs of correctly matched lines.
- Clustering of lines utilizes search.
- Lines parallel to $x_1$ axis in scan 1 can only be matched with lines parallel to $x_2$ axis in scan 2, etc.

For all pairs of lines $(l_1, r_1)$ and $(l_2, r_2)$ such that:
- $(l_1, r_1)$ are parallel to the same axis ($y$-axis in this example).
- $(l_2, r_2)$ are parallel to the same axis ($z$-axis in this example).
- The distance $d_1$ is equal (within a threshold) to distance $d_2$.

Compute an exact the translation $T_{12}$ by solving a linear system of equations.
- Note that $d_1$ and $d_2$ are estimates that depend on endpoints.
Translation Computation

- Cluster all computed translations.
- Pick N most frequently appeared translations.
- For each of the N translations:
  - Apply an optimization routine on R and T.
  - Count matched line pairs with optimized T.
- Pick the T with the largest number of matches.

Overview of user-interface
Iterative Closest Point Algorithm

Before ICP

After ICP

Final result – 24 scan pairs
~600 lines per scan