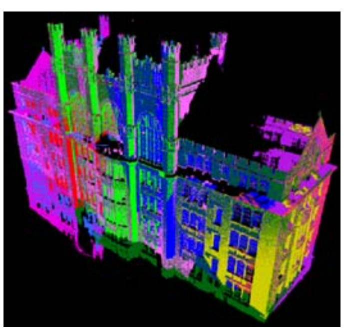
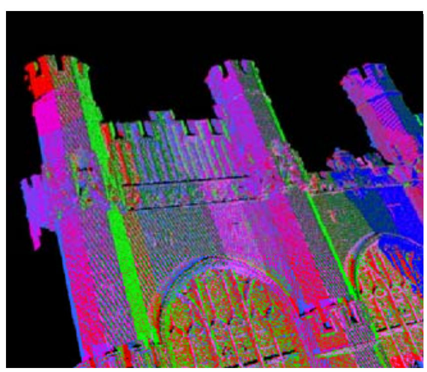
3D Pipeline
Segmentation (planar)
Registration (using lines)

3D PHOTOGRAPHY EXAMPLE



Automatic registration.
Each scan has a different color.

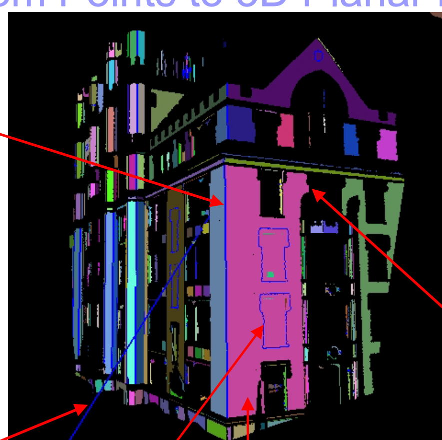


Registration details

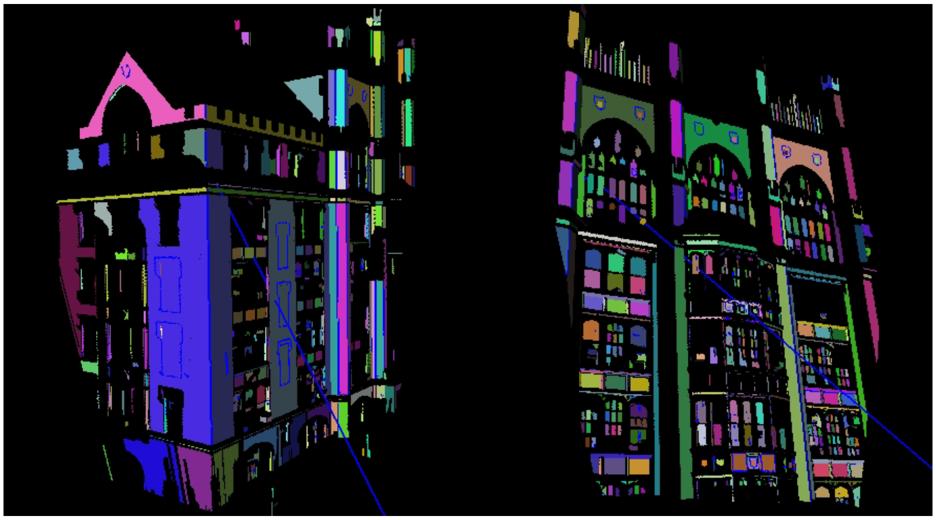
Problem

- 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



More Segmentation Results



Two segmented range scans of building

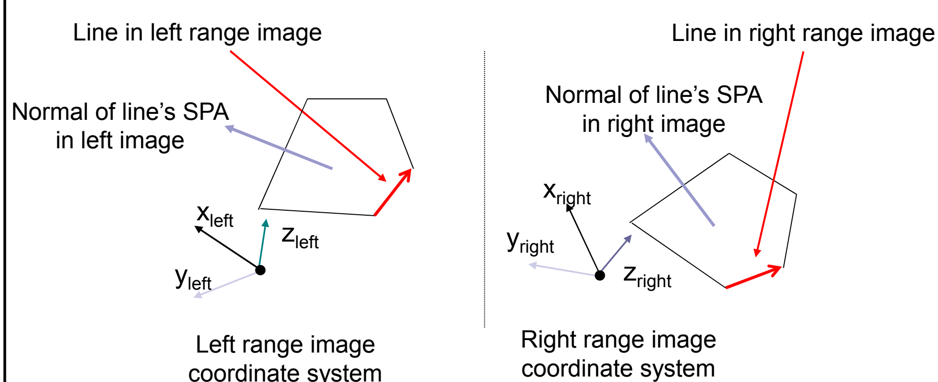
Approach

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

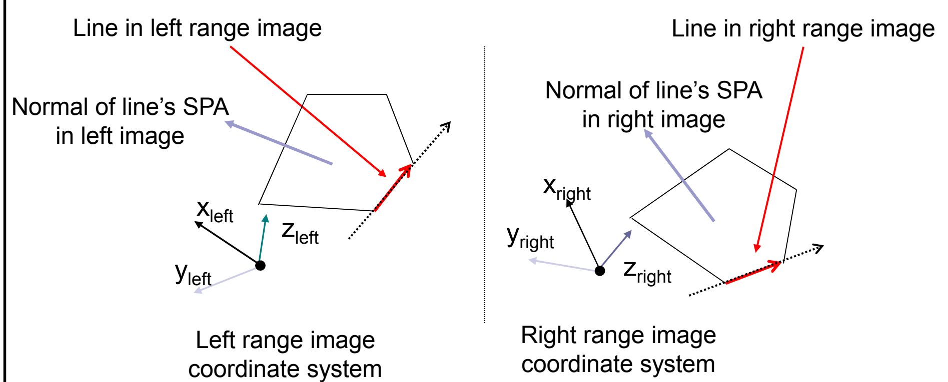
- Pairwise registration between two scans.
- Automated method uses linear features.
- Features extracted at boundaries of SPAs.
- Two correctly matched lines between scans provide solution.
- If scan A contains N lines and scan B contains M lines
  - Need to consider $O(N^2M^2)$ pairs.
  - For each pair 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



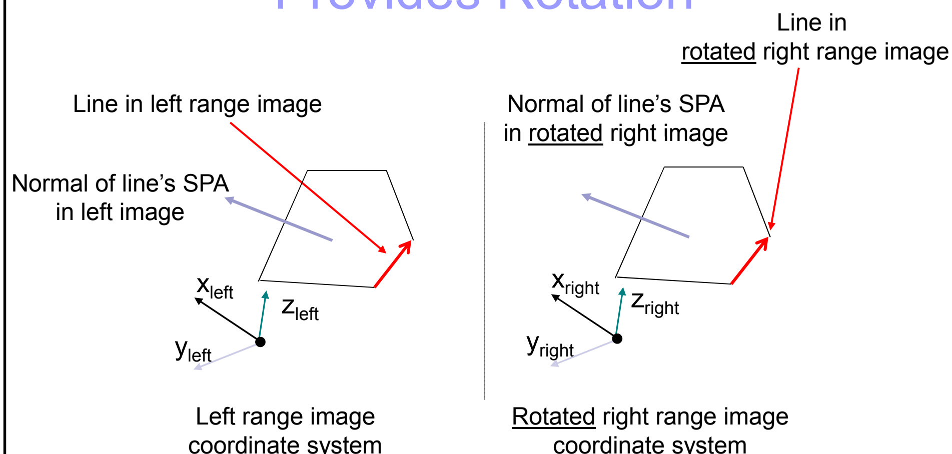
- Only orientation and location of lines used.
- Endpoints not-used.
- Closed-form solution provides rotation.

One Pair of Correctly Matched Lines Provides Rotation



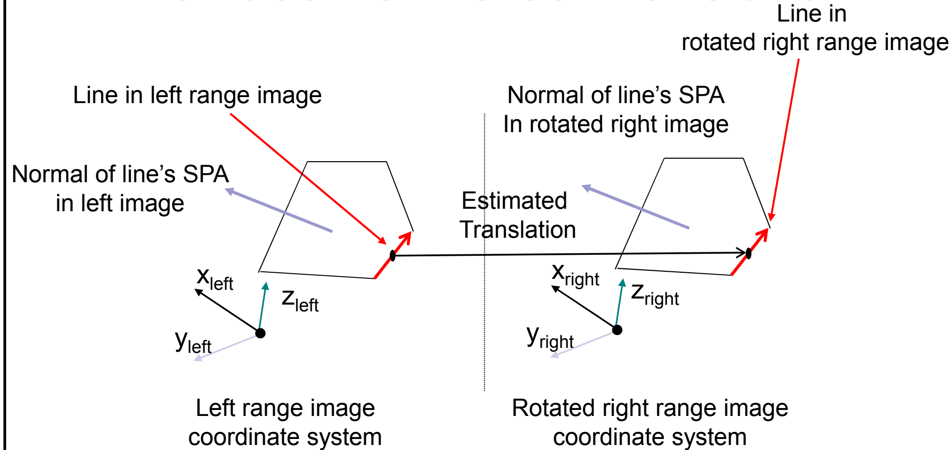
- Only orientation and location of lines used.
- Endpoints not-used.
- Closed-form solution provides rotation.

One Pair of Correctly Matched Lines Provides Rotation



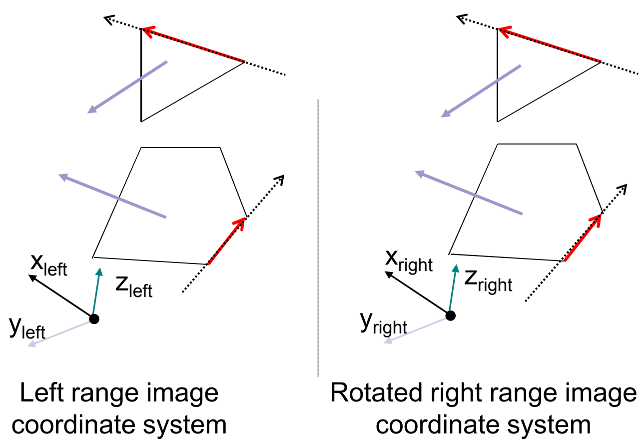
- Right coordinate system is rotated about its origin to match left coordinate system.

One Pair of Correctly Matched Lines Provides Estimated Translation



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

Two Pairs of Correctly Matched Lines Provide Exact Translation



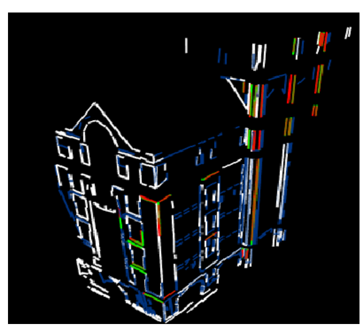
- <u>Exact</u> translation computation by using two pairs.
- Endpoints of lines are not used.

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.

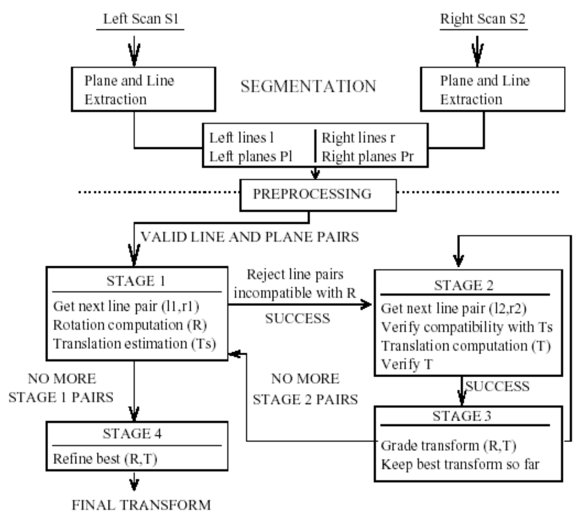
Exhaustive Search Approach

- 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

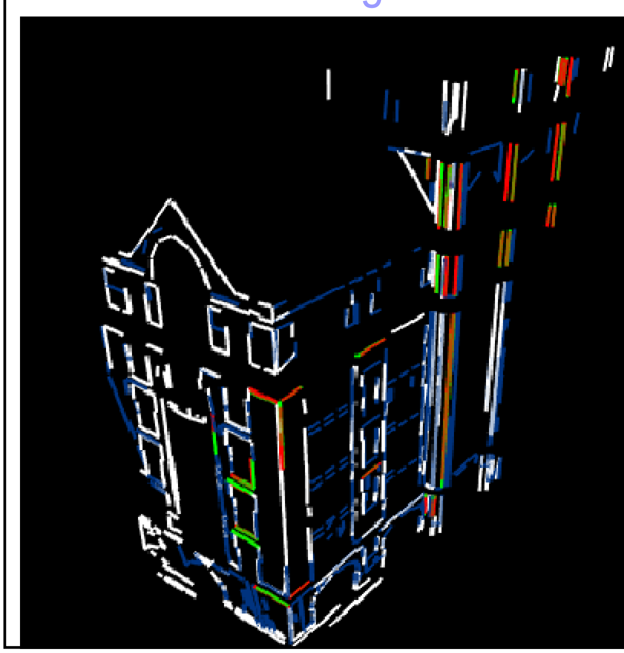


White lines (left scan)
Blue lines (right scan)
Red/Green lines (matches)

Flowchart



Pairwise Registrations:Line Matches



White lines (left scan)

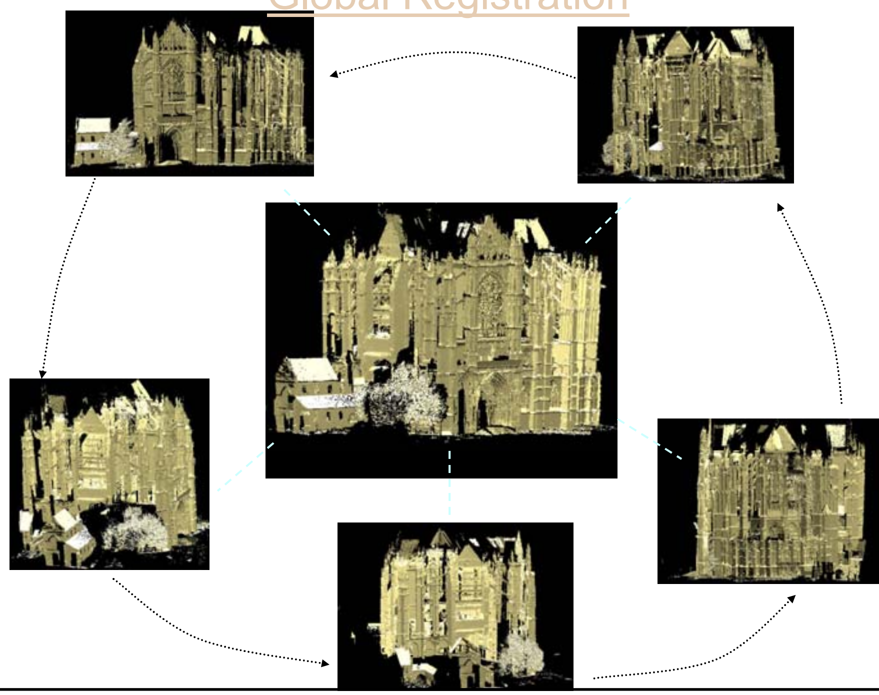
Blue lines (right scan)

Red/Green lines (matches)

Results

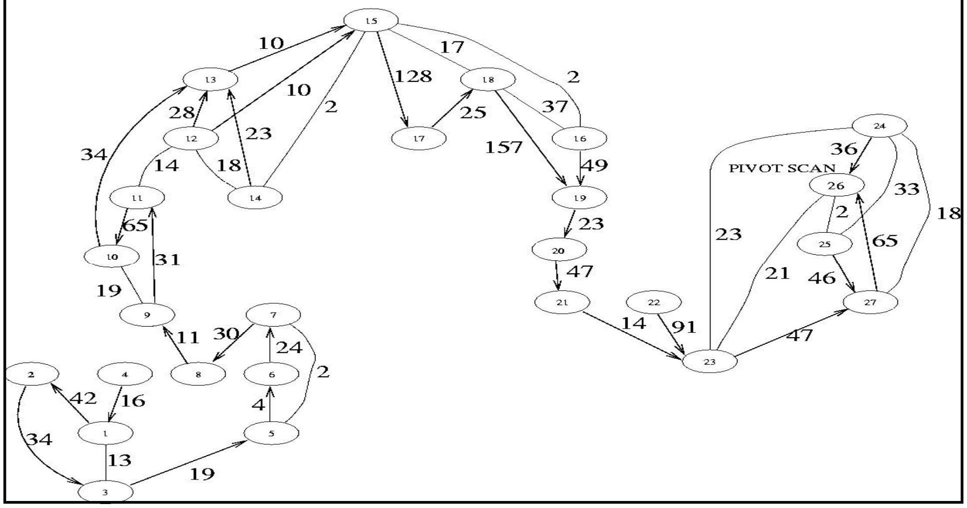
Campus Building - Results (average error 7.4mm)							
Pair	Line Pairs	Pre (%)	S2 % (#)	S3 % (#)	Matches	t (sec)	Error
1	301 × 303	16	1.7 (1555)	0.38 (346)	35	15	10.99mm
2	303 × 290	17	2.8 (2429)	0.84 (735)	25	29	6.28mm
3	290 × 317	21	2.8 (2572)	1.88 (1728)	36	52	2.77mm
4	317 × 180	19	3.4 (1955)	1.15 (656)	28	21	14.96mm
5	211 × 180	21	4.6 (1759)	2.1 (802)	31	19	9.26mm
6	180 × 274	17	2.6 (1306)	0.34 (168)	22	9	11.42mm
7	114 × 274	19	1.6 (507)	2.2 (894)	33	6	5.61mm
8	274 × 138	16	1.8 (667)	1.5 (557)	31	5	3.08mm
9	114 × 138	18	2.7 (423)	3.8 (593)	32	4	3.94mm
10	138 × 247	18	2.3 (791)	1.3 (429)	20	5	1.36mm
Cathedral - Results (average error 17.3mm)							
1	406 × 464	7	0.9 (1650)	0.3 (615)	42	39	9.37mm
2	464 × 269	7	0.7 (888)	0.3 (443)	34	16	16.9mm
3	406 × 269	11	0.7 (794)	0.1 (104)	13	9	56.08mm
4	151 × 406	21	1.1 (668)	0.8 (480)	16	7	5.34mm
5	269 × 387	11	0.7 (702)	0.4 (369)	19	9	15.8mm
6	326 × 197	10	0.9 (597)	0.1 (49)	24	4	11.68mm
7	197 × 143	15	1.0 (290)	0.3 (82)	30	3	6.44mm
8	143 × 194	16	1.9 (520)	0.1 (31)	11	3	29.24mm
9	194 × 356	15	2.0 (1429)	0.1 (93)	19	11	30.82mm

Global Registration

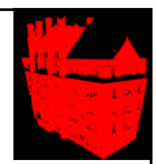


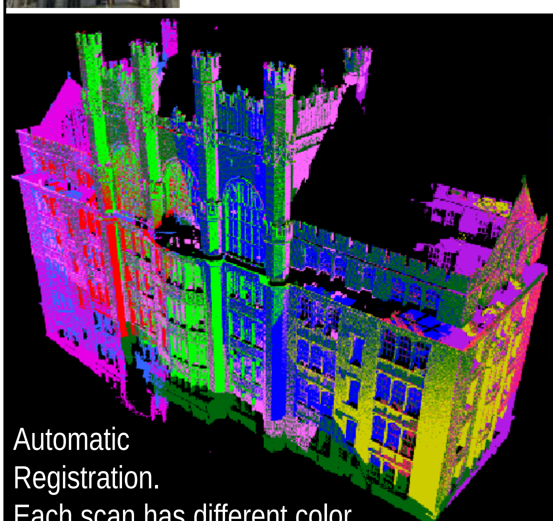
Graph Search Global Registration

- 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

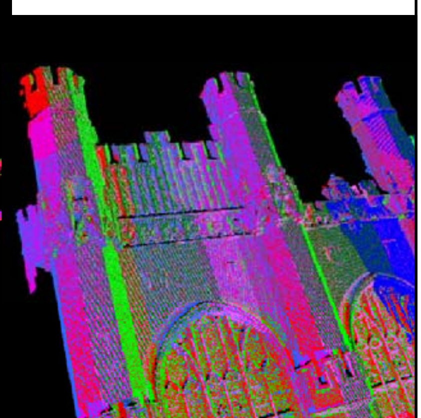


Thomas Hunter Building: 10 scans





Automatic Registration.
Each scan has different color.



Registration Detail.

St. Pierre Cathedral, Beauvais France

World Monuments Fund's Most Endangered List.

Photograph

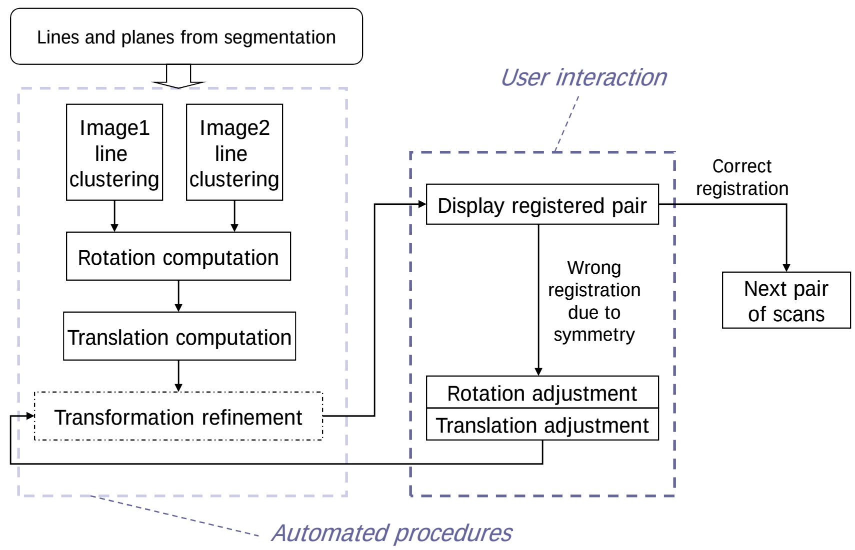
Combined 3D mesh

(with P. K. Allen, Columbia University)

Problems with described algorithm

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

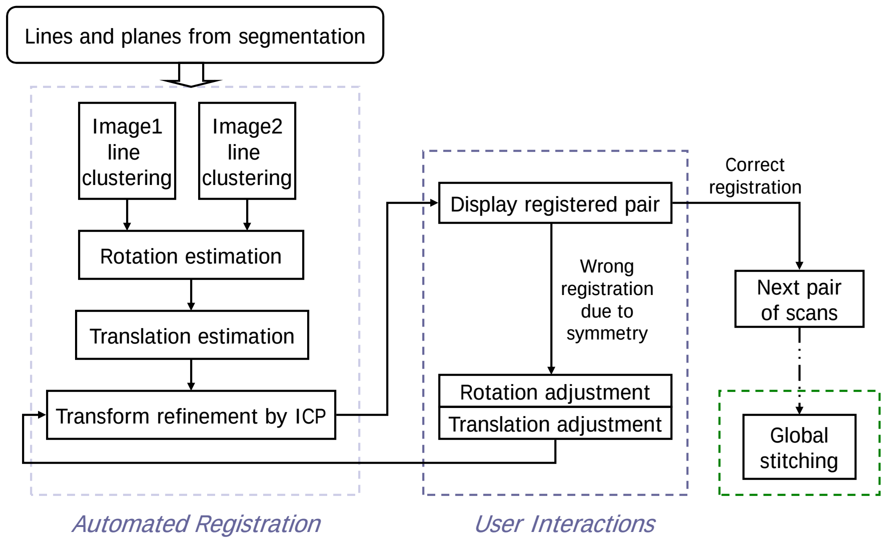
New Framework



Exhaustive Search Approach

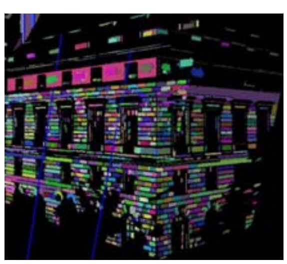
- No initial registration needed
- High computational complexity
- Symmetry problem unsolved
- Improvements
  - Extract object-based coordinate system
  - Context-sensitive user interface

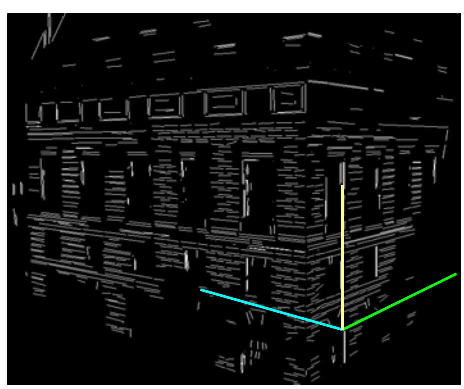
Framework of Interactive Solution



Line Clustering

- Line clustering
  - Line directions
  - Plane normals

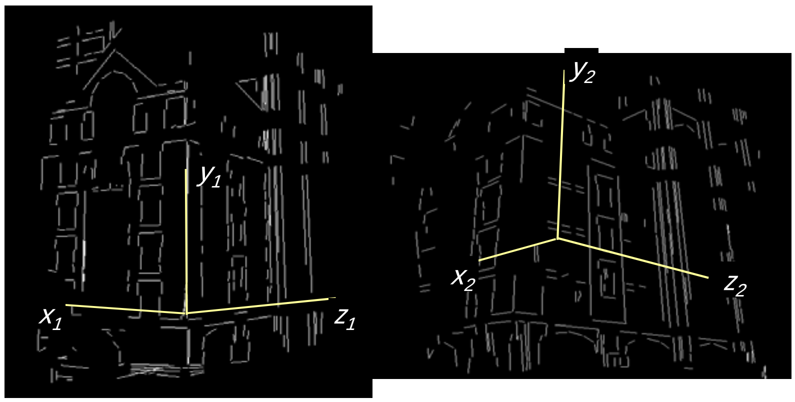




Building's local coordinate system

Rotation Estimation

- Rotation estimation

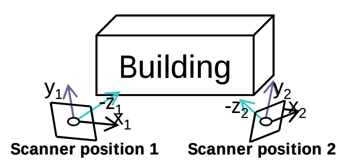


$R = [x_2\ y_2\ z_2]^T * [x_1\ y_1\ z_1]$

24 possible R's?

Rotation Estimation

- 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



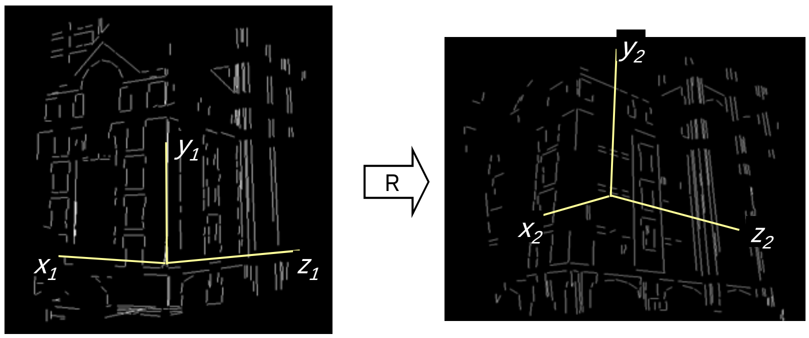
$x_1 * R = x_2$
$y_1 * R = y_2$ ⇒ $(0,1,0) * R = R_{11}$: projection of y1 on y2
$z_1 * R = z_2$

⇒ $R_{11} > \cos(45°) = 0.7$ ⇒ 2 to 5 R's

⇒ Sort by $R_{00}+R_{11}+R_{22}$ ⇒ *Return the R with the largest diagonal sum*

Translation Estimation

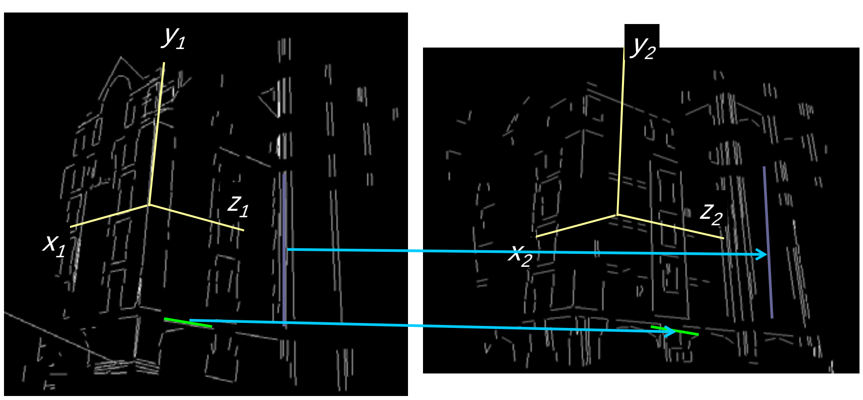
- Translation estimation



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

Translation Estimation

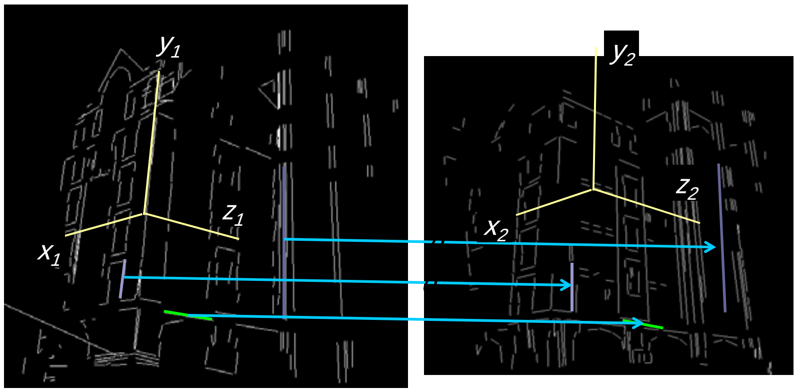
- One pair of matched lines provides an estimated translation
- Two pairs with similar estimated translations provide translation candidate



Line pair 1 *Line pair 2*

Translation Estimation

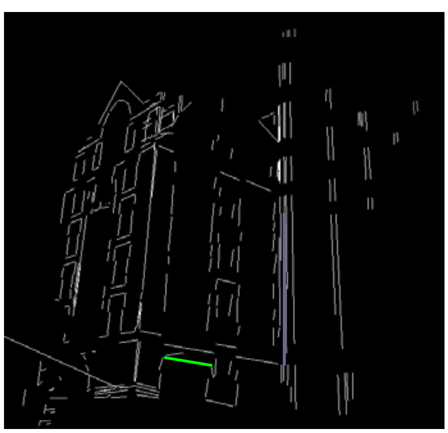
- Two types of translation candidates

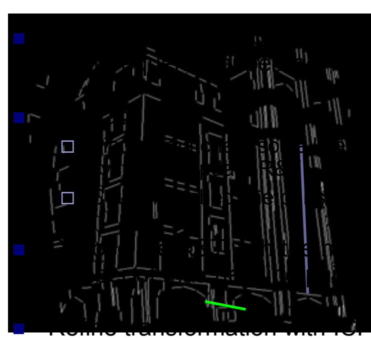


$T = (d_1 + d_2) / 2$ $\quad$ $T \leftarrow$ linear system

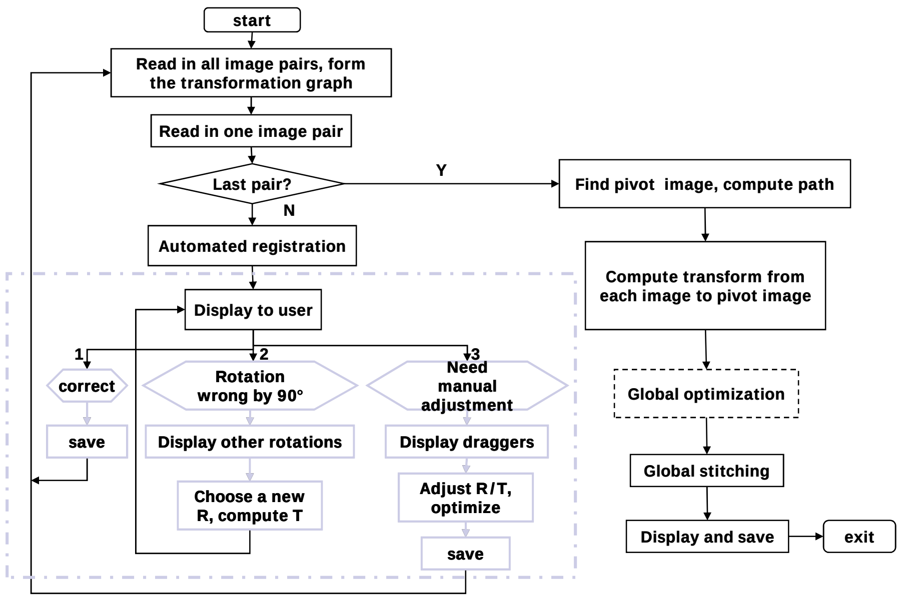
Translation Estimation

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



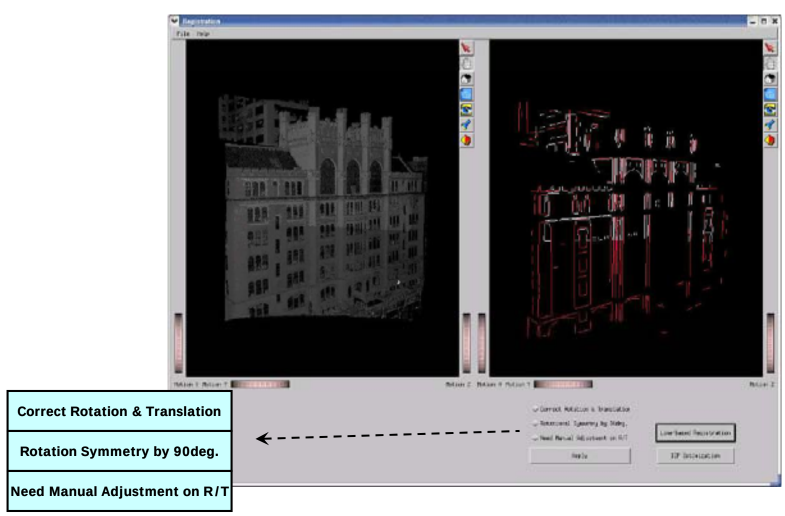


Registration System Flowchart



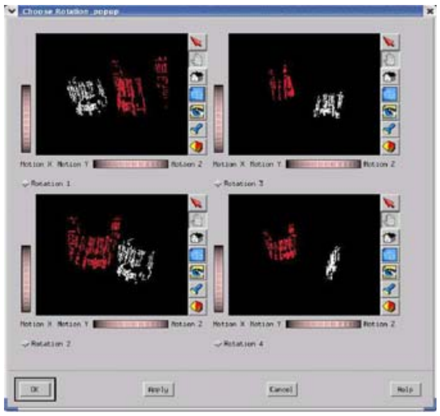
User Interface

Display window: Points and lines of registered two scans



User Interface

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

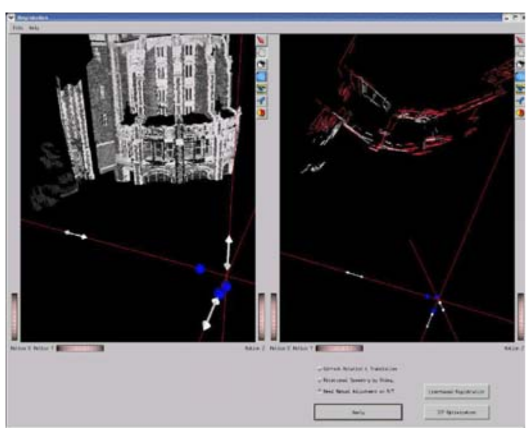


User Interface

- Adjusting rotation and translation based on the building's coordinate system

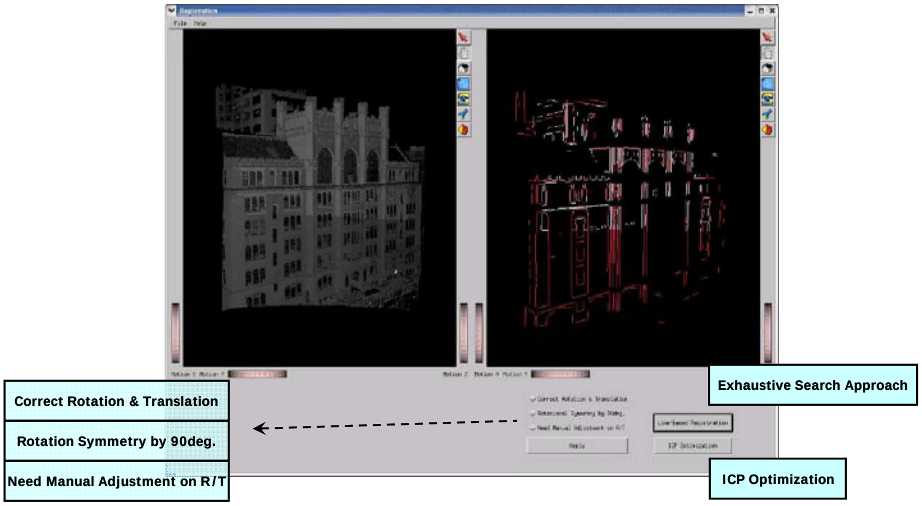
White draggers for translation

Blue spheres for rotation

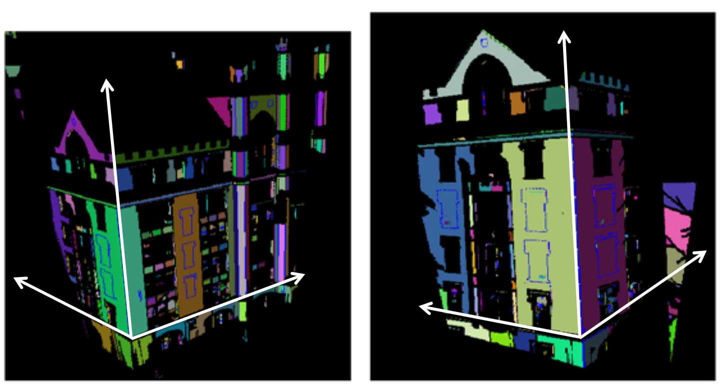


User Interface

Display window: Points and lines of registered two scans



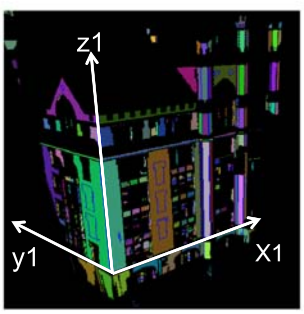
Clustering 3D lines provide major building's directions

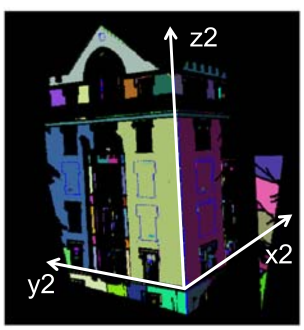


Two segmented scans with major directions shown

Utilizing user-interaction

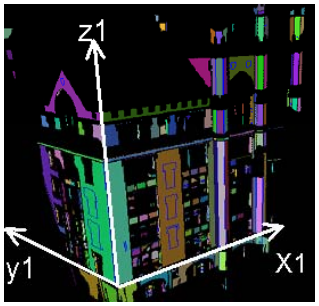
Rotation Computation

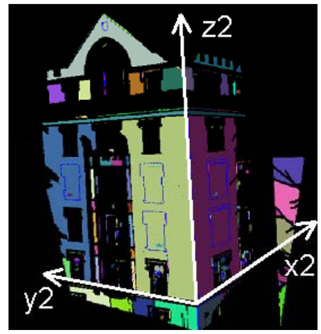




- If we have correct match of axes:
  $R = [x_2\ y_2\ z_2]^T * [x_1\ y_1\ z_1]$
- 24 candidate rotations exist if match of axes is uknown.
- Heuristic produces the more plausible candidate.
- If wrong the user can select from a set of candidates.

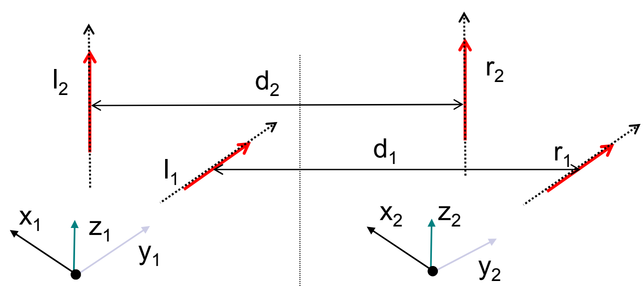
Translation Computation





- Search for two pairs of correctly matched lines.
- Clustering of lines utilizes search.
- Lines parallel to x1 axis in scan 1 can only be matched with lines parallel to x2 axis in scan 2, etc.

Translation Computation



Left range image coordinate system

Rotated right range image coordinate system

For **all** pairs of lines (l1, r1) and (l2, r2) such that:

- (l1, r1) are parallel to the same axis (y-axis in this example).
- (l2, r2) are parallel to the same axis (z-axis in this example).
- The distance d_1 is equal (within a threshold) to distance d_2.

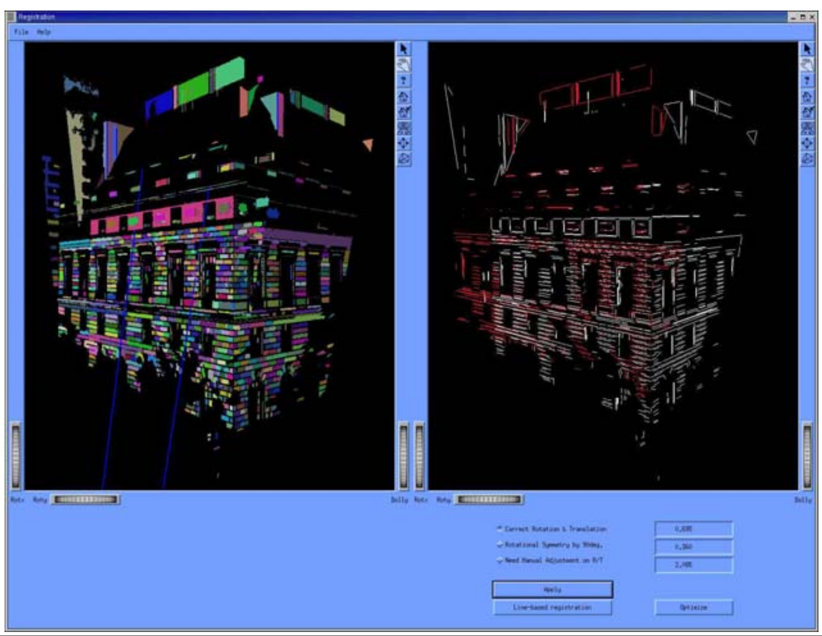
Compute an exact the translation T_{12} by solving a linear system of equations.

- Note that d_1 and d_2 are estimates that depend on endpoints.

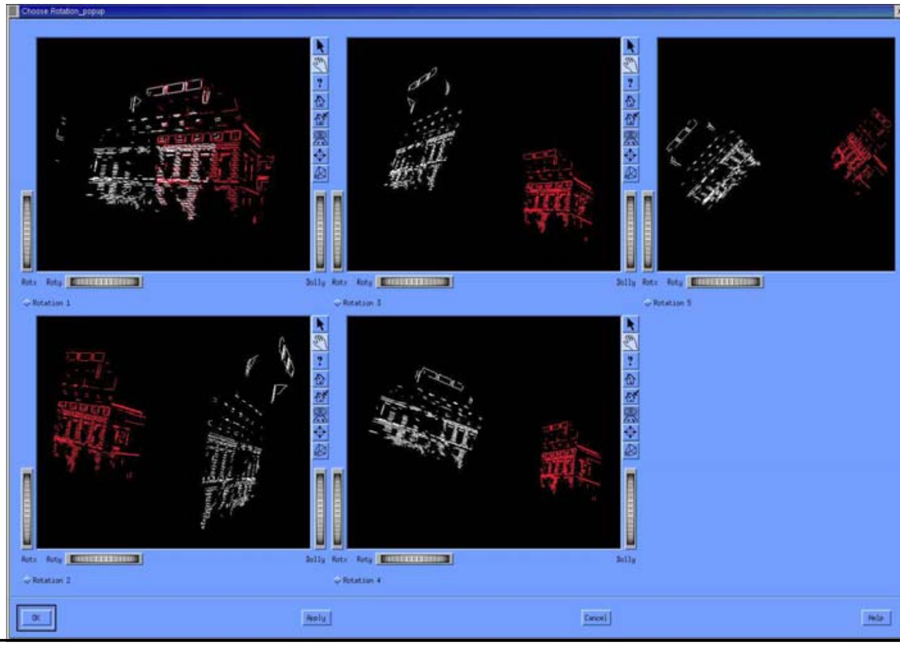
Translation Computation

- 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.

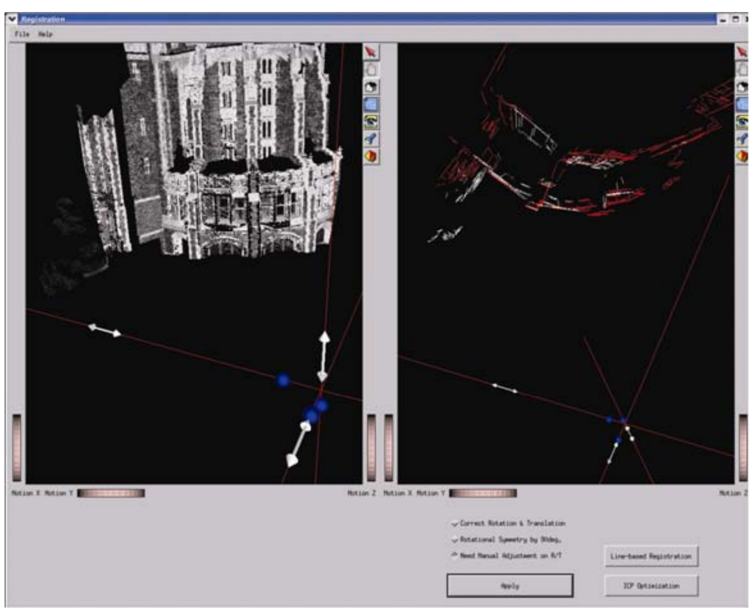
Overview of user-interface



Overview of user-interface

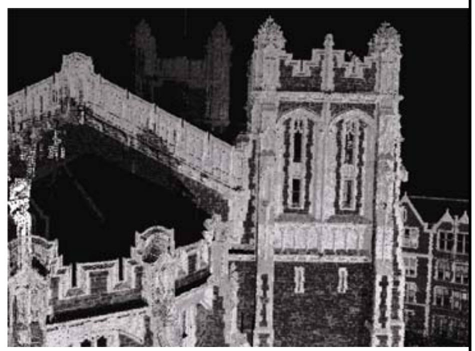


Overview of user-interface



Iterative Closest Point Algorithm

Before ICP



After ICP

Final result – 24 scan pairs
~600 lines per scan