LECTURE 08A: 3D REGISTRATION

Venu Madhav Govindu
Department of Electrical Engineering
Indian Institute of Science, Bengaluru

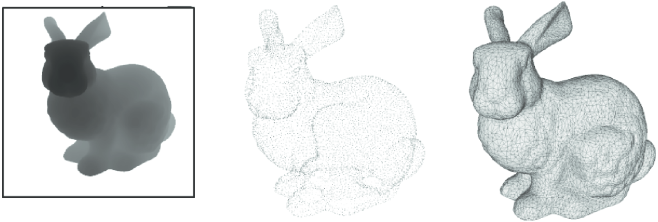
2023

3D Reconstruction

3D Reconstruction

- Stereo methods give us depth from one viewpoint
- Need to cover entire object/scene
- To build a full 3D model
  - Stereo/depth estimation method
  - Method to align/register individual scans
  - Method to create a single unified surface representation

3D Reconstruction

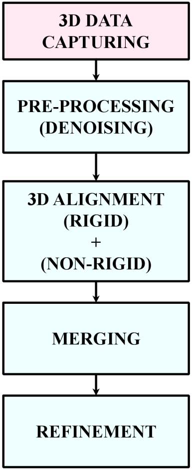


3D Representations

- Three primary representations
  - **Depth map** $Z(x, y)$
  - Unstructured **Point Cloud**
  - Triangulated **Mesh**
- Representation has consequences on processing
- Sparsity/density of measurements
- Mesh models extensively developed in graphics

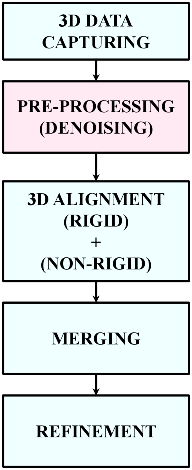
Figured adapted from Kaiser *et al.* 'A Survey of Simple ...'

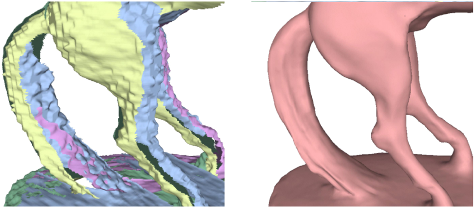
3D Reconstruction



Model from AIM Shape Repository

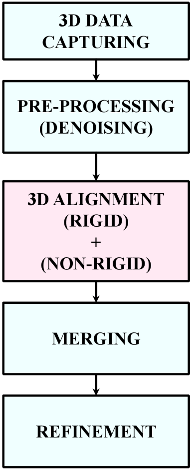
3D Reconstruction

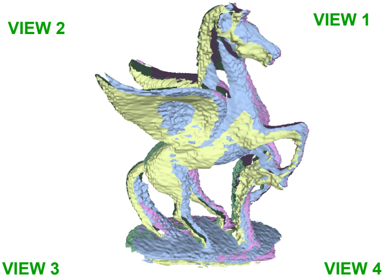




Model from AIM Shape Repository

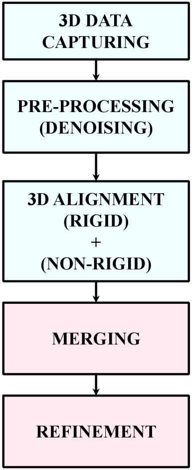
3D Reconstruction

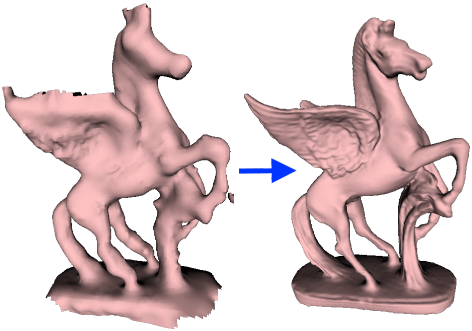




Model from AIM Shape Repository

3D Reconstruction



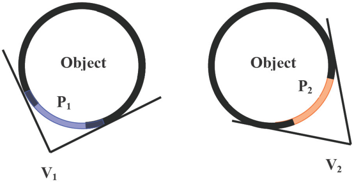


Model from AIM Shape Repository

3D Scan Registration

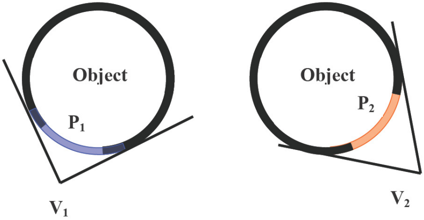
Scans from multiple orientations

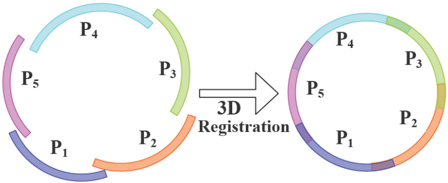
3D Scan Registration



- Each scan is a *partial* model
- Has own *local* frame of reference

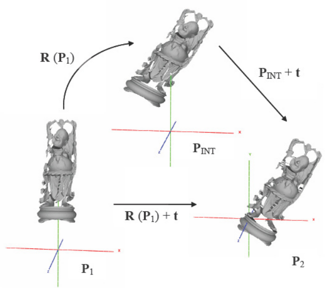
3D Scan Registration





- Need to **register** partial *overlapping* scans
- Need a **single** common frame of reference
- **Many** approaches to register scans
- We will focus on a classic approach
- Modern learning models widely applied

3D Scan Registration



3D Registration

- Motion model is 3D rotation $\boldsymbol{R}$ and translation $\boldsymbol{T}$
- $\boldsymbol{R} \in \mathbb{SO}(3)$ and $\boldsymbol{T} \in \mathbb{R}^3$, full motion $\in \mathbb{SE}(3)$
- Let two sets of corresponding points be $\boldsymbol{P}_1 \in \mathbb{R}^3$ and $\boldsymbol{P}_2 \in \mathbb{R}^3$
- Transformation model: $\boldsymbol{P}_2(i) = \boldsymbol{R}\boldsymbol{P}_1(i) + \boldsymbol{T}, \forall i$
- Consider cost : $\sum_{i=1}^{N} ||\boldsymbol{R}\boldsymbol{P}_1(i) + \boldsymbol{T} - \boldsymbol{P}_2(i)||^2$
- Makes sense with Gaussian noise model

Euclidean Transformation in $\mathbb{R}^3$

$$
\begin{aligned}
\boldsymbol{P}_2 &= \boldsymbol{R}\boldsymbol{P}_1 + \boldsymbol{T} \\
\boldsymbol{P}_2 &= \boldsymbol{R}(\boldsymbol{P}_1 + \boldsymbol{T})
\end{aligned}
$$

- First is rotate then translate
- Second is translate then rotate
- Both are valid representations
- We will prefer the first form over the second
- **Warning : Always understand which one is used!**

Cost function to be minimised:

$$\begin{aligned} \boldsymbol{P}_2(i) &= \boldsymbol{R}\boldsymbol{P}_1(i) + \boldsymbol{T}, \forall i \\ &\Rightarrow \min_{\{\boldsymbol{R},\boldsymbol{T}\}} \sum_i ||\boldsymbol{R}\boldsymbol{P}_1(i) + \boldsymbol{T} - \boldsymbol{P}_2(i)||^2 \end{aligned}$$

Least Squares Fitting

- Most common approach is to optimise least-squares cost function
- This is **not** a linear problem. Why ?
- Separability of translation and rotation estimation

Consider the correspondences:

$$
\begin{aligned}
\boldsymbol{P}_2(1) &= \boldsymbol{R}\boldsymbol{P}_1(1) + \boldsymbol{T} \\
\boldsymbol{P}_2(2) &= \boldsymbol{R}\boldsymbol{P}_1(2) + \boldsymbol{T} \\
&\cdots \\
\boldsymbol{P}_2(N) &= \boldsymbol{R}\boldsymbol{P}_1(N) + \boldsymbol{T}
\end{aligned}
$$

Consider the correspondences:

$$\begin{aligned}
\boldsymbol{P}_2(1) &= \boldsymbol{R}\boldsymbol{P}_1(1) + \boldsymbol{T} \\
\boldsymbol{P}_2(2) &= \boldsymbol{R}\boldsymbol{P}_1(2) + \boldsymbol{T} \\
&\cdots \\
\boldsymbol{P}_2(N) &= \boldsymbol{R}\boldsymbol{P}_1(N) + \boldsymbol{T}
\end{aligned}$$

Does this suggest something?

Consider the correspondences:

$$\begin{aligned}
\boldsymbol{P}_2(1) &= \boldsymbol{R}\boldsymbol{P}_1(1) + \boldsymbol{T} \\
\boldsymbol{P}_2(2) &= \boldsymbol{R}\boldsymbol{P}_1(2) + \boldsymbol{T} \\
& \cdots \\
\boldsymbol{P}_2(N) &= \boldsymbol{R}\boldsymbol{P}_1(N) + \boldsymbol{T}
\end{aligned}$$

Let's add them up

Consider the correspondences:

$$\begin{aligned}
\boldsymbol{P}_2(1) &= \boldsymbol{R}\boldsymbol{P}_1(1) + \boldsymbol{T} \\
\boldsymbol{P}_2(2) &= \boldsymbol{R}\boldsymbol{P}_1(2) + \boldsymbol{T} \\
&\cdots \\
\boldsymbol{P}_2(N) &= \boldsymbol{R}\boldsymbol{P}_1(N) + \boldsymbol{T}
\end{aligned}$$

$$\begin{aligned}
\sum_i \boldsymbol{P}_2 &= \boldsymbol{R}\sum_i \boldsymbol{P}_1 + N\boldsymbol{T} \\
\Rightarrow \frac{1}{N}\sum_i \boldsymbol{P}_2 &= \frac{1}{N}(\boldsymbol{R}\sum_i \boldsymbol{P}_1 + N\boldsymbol{T}) \\
\Rightarrow \mu_2 &= \boldsymbol{R}\mu_1 + \boldsymbol{T}
\end{aligned}$$

We have

$$\begin{aligned} \boldsymbol{P}_2(i) &= \boldsymbol{R}\boldsymbol{P}_1(i) + \boldsymbol{T} \\ \mu_2 &= \boldsymbol{R}\mu_1 + \boldsymbol{T} \end{aligned}$$

Remove translation component

- Subtracting centroids we get:

$$\boldsymbol{P}_1^{'} = \boldsymbol{P}_1 - \mu_1$$
$$\boldsymbol{P}_2^{'} = \boldsymbol{P}_2 - \mu_2$$
$$\boldsymbol{P}_2^{'} = \boldsymbol{R}\boldsymbol{P}_1^{'}$$

- New problem is independent of translation $\boldsymbol{T}$
- Need to solve for $\boldsymbol{R} \in \mathbb{SO}(3)$

3D Scan Registration

Cost function

$$P_2'(i) = RP_1'(i)$$
$$\Rightarrow \min_{R} \sum_i ||RP_1'(i) - P_2'(i)||^2$$

Optimal 3D Rotation

- Define $M = P_1 P_2^T$
- SVD: $M = U \sum V^T$
- Optimal 3D rotation $R = USV^T$
- $S = \text{diag}(1, 1, |UV^T|)$

Recovering 3D Rotation

- Problem only to recover R
- Least squares cost function minimisation
- Collect all correspondences in $3 \times N$ matrices P_1 and P_2
- Simple and elegant solution based on SVD

3D Scan Registration

$$\text{Cost}: \sum_{i=1}^{N} ||\boldsymbol{R}\boldsymbol{P}_1(\boldsymbol{i}) + \boldsymbol{t} - \boldsymbol{P}_2(\pi(\boldsymbol{i}))||^2$$

3D Registration : Known Correspondences

- Assume **known correspondence** : $\pi(.)$ is known
- Solving for motion is **easy**

3D Registration : Known Motion

- Need $\pi(.)$: $\boldsymbol{P}_1(\boldsymbol{i}) \leftrightarrow \boldsymbol{P}_2(\pi(\boldsymbol{i}))$
- Apply motion to $\boldsymbol{P}_1(\boldsymbol{i})$
- Find its closest point in $\boldsymbol{P}_2$
- Justified for Gaussian noise model

Iterative Closest Point Algorithm

- 3D Registration problem is
  - **Easy** for known **motion** or **correspondence**
  - **Hard** if *both* are unknown
- Search space grows enormously if neither known
- ICP provides a *greedy* solution
- Iterative method
- Converges to local minima
- Needs a good initial guess

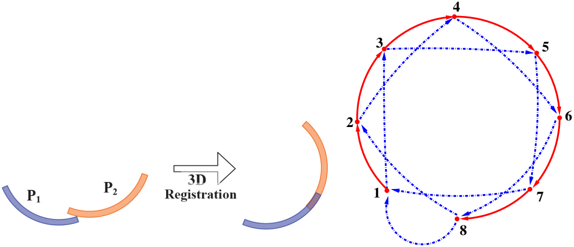
3D Scan Registration

Iterative Closest Point Algorithm

- Start at initial guess of motion
- Correspondence for **given** motion (correspondence step)
- Update motion using this correspondence (motion step)
- Iterate till convergence
- Many practical issues and refinements

ICP is the workhorse for 3D registration

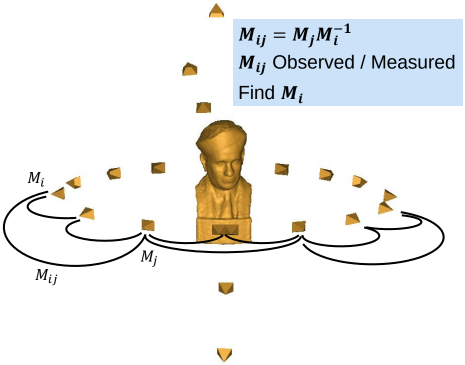
3D Scan Registration



Iterative Closest Point Algorithm

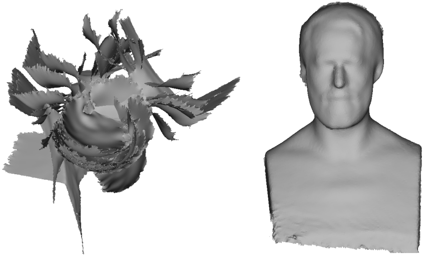
- ICP limited to 2 views at a time
- Does not carry out *simultaneous* registration of scans
- Incremental drift in the solution
- Fails to exploit available constraints like ‘closure’
- Global methods obviate such problems

3D Reconstruction



- Pairwise motions are estimated using ICP
- Problem: To find absolute motions from given pairwise relative motions

3D Reconstruction

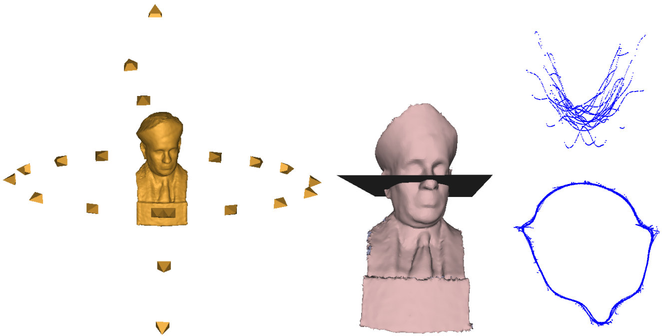


Alignment using ℓ_2 and ℓ_1 motion averaging

Robustness

- Robustness is a major issue
  - RANSAC-like approaches
  - IRLS for $\rho(.)$ loss functions
- Many heuristics in ICP implementations

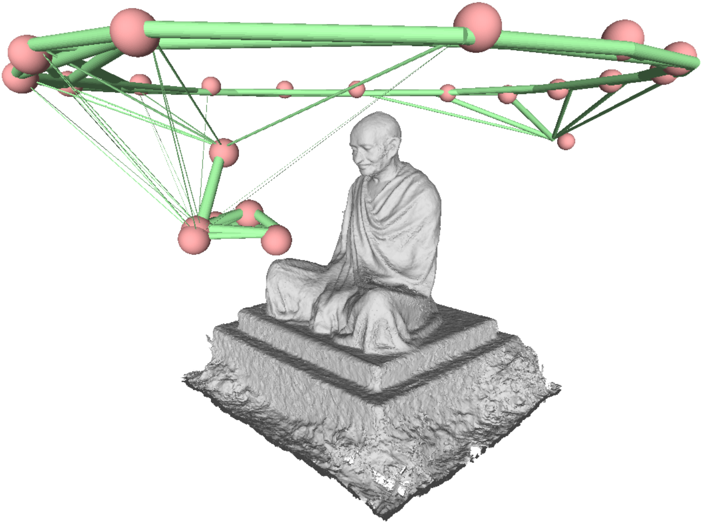
3D Reconstruction



Left: Final model with estimated scanner locations
Right: Cross-sections before and after alignment

Figure: Statue of Mahatma Gandhi at Sabarmati Ashram, Ahmedabad (90 cm height)

3D Reconstruction



Issues not considered

- Lots of engineering heuristics (Kinect Fusion)
- Methods to extract SIFT-like 3D features
- Learnt vs. Geometric 3D features
- Alternate methods to estimate $\boldsymbol{R}$
- Merge registered scans
  - Create single surface representation
  - Signed distance function
  - Find zero crossing
  - Fast methods
  - Occupancy data structures
  - Marching cubes
- Deep learning for surface representations
- End-to-end deep learning