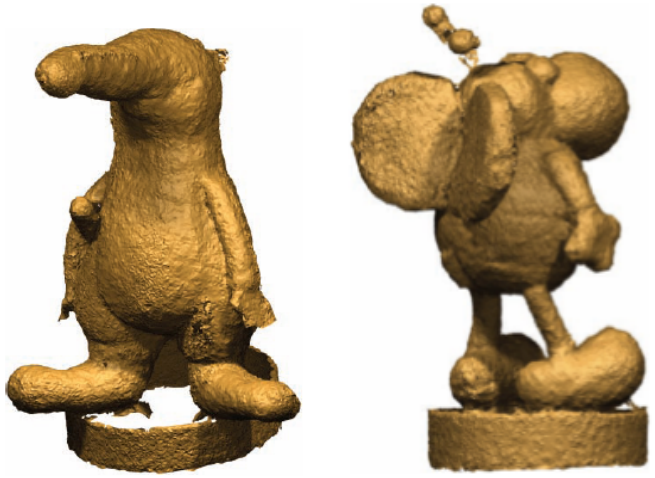
Dynamic Geometry Processing

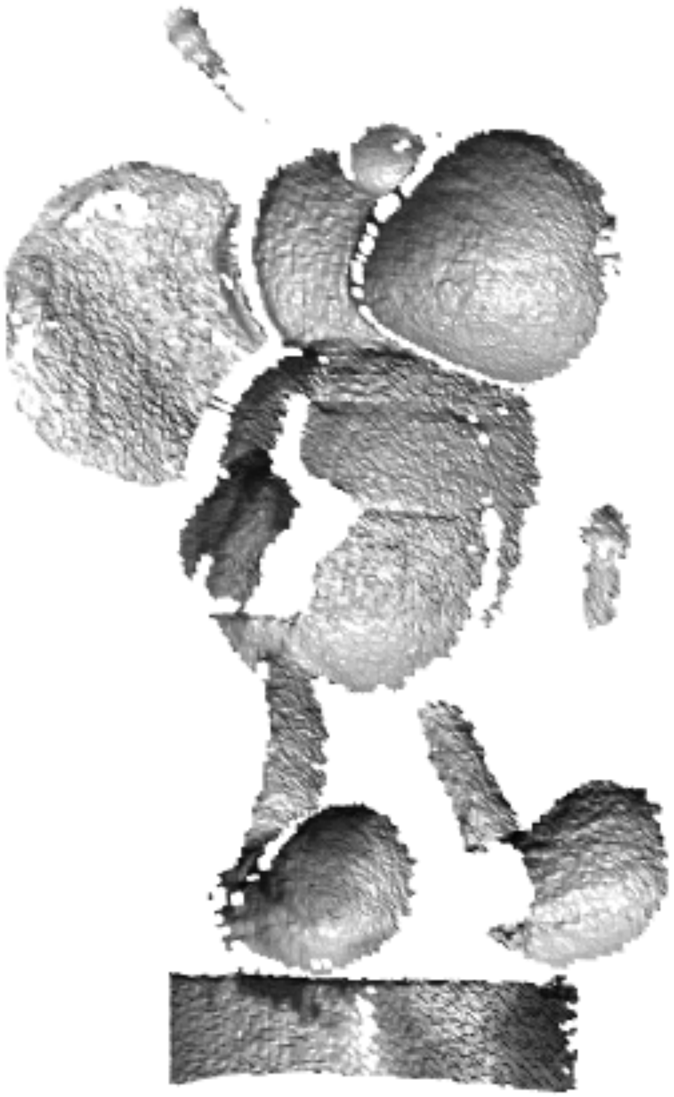
EG 2012 Tutorial

Dynamic Registration

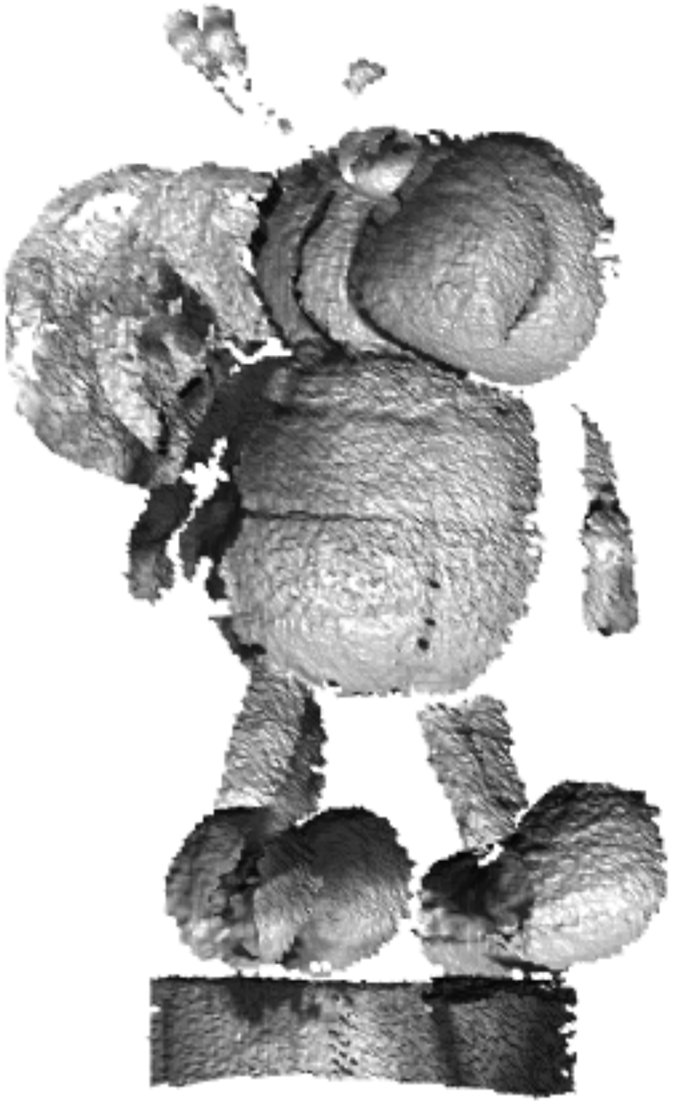


Niloy J. Mitra
University College London

Scan Registration



Scan Registration



Solve for inter-frame motion:

$$\alpha := (\mathbf{R}, \mathbf{t})$$

Scan Registration

Solve for inter-frame motion:

$$\alpha_j := (\mathbf{R}_j, \mathbf{t}_j)$$

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

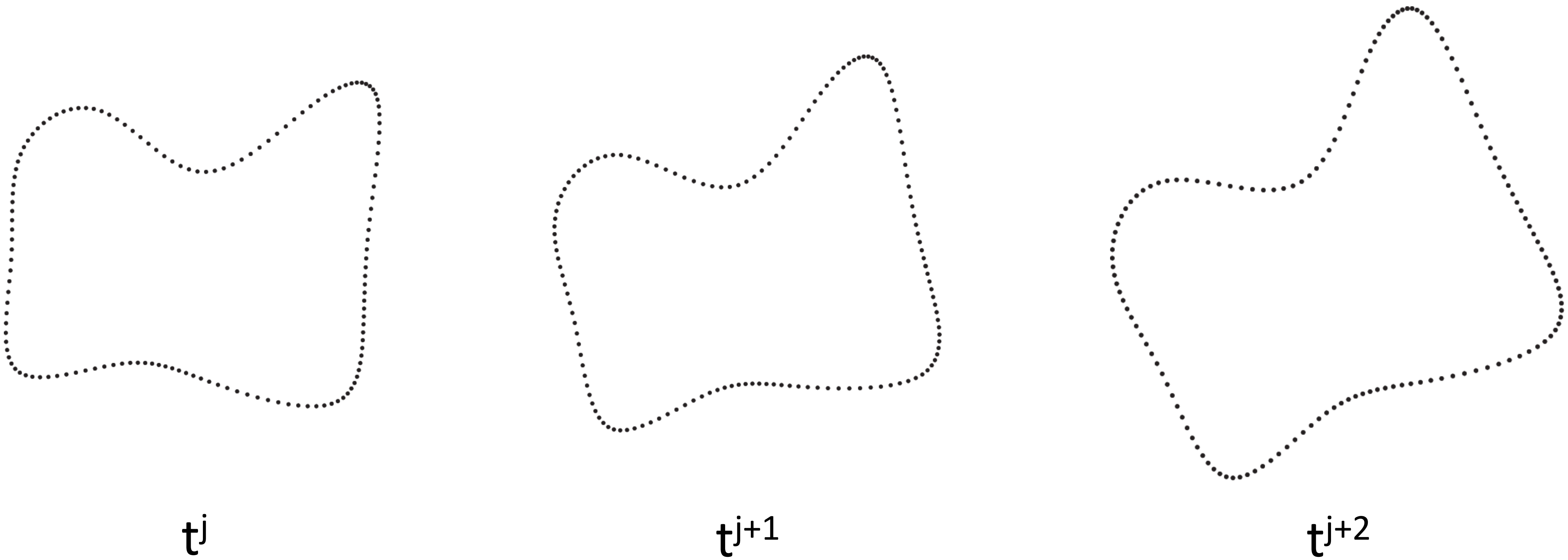
Insights

Rigid registration → kinematic property of space-time surface (locally exact)

Registration → surface normal estimation

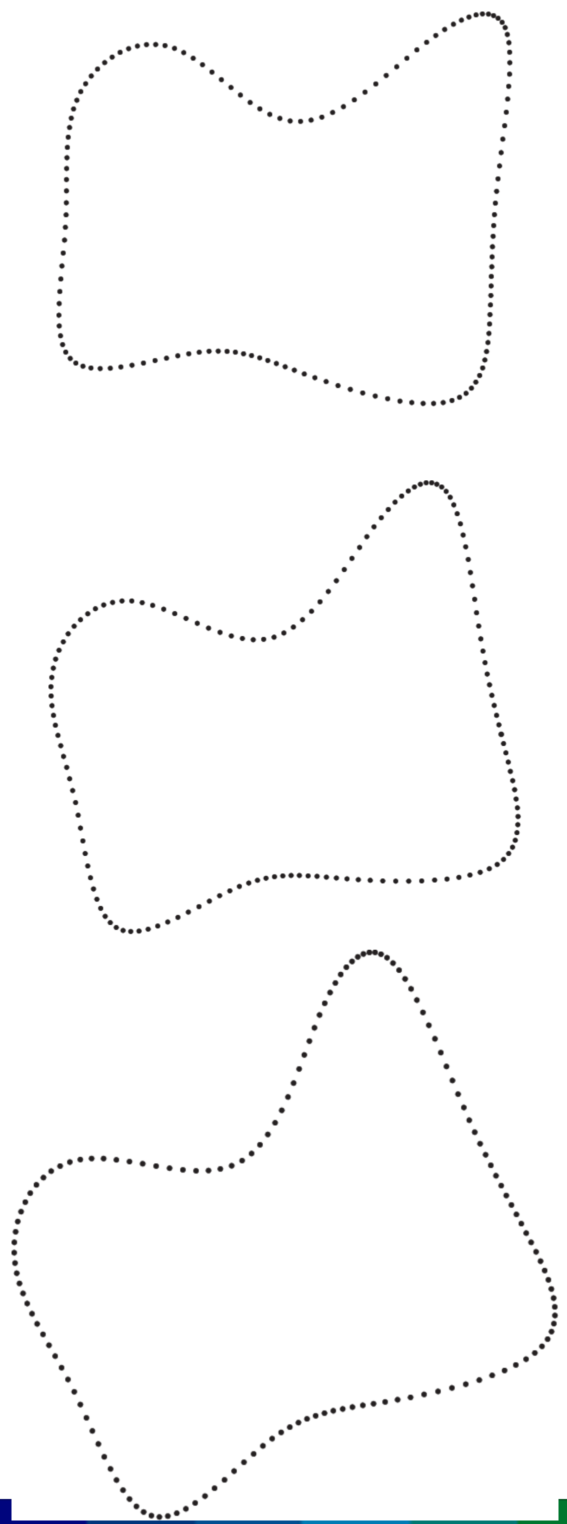
Extension to deformable/articulated bodies

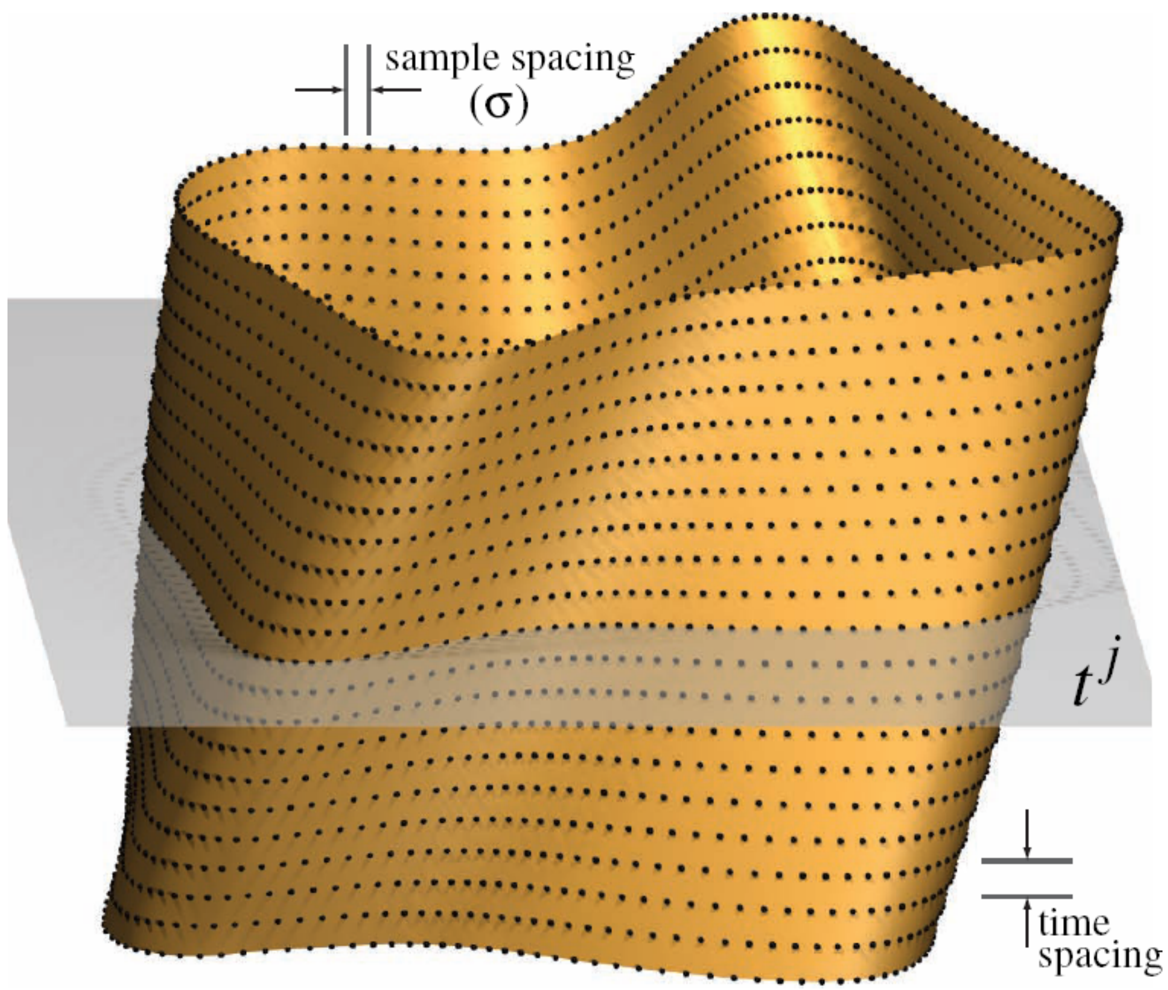
Time Ordered Scans



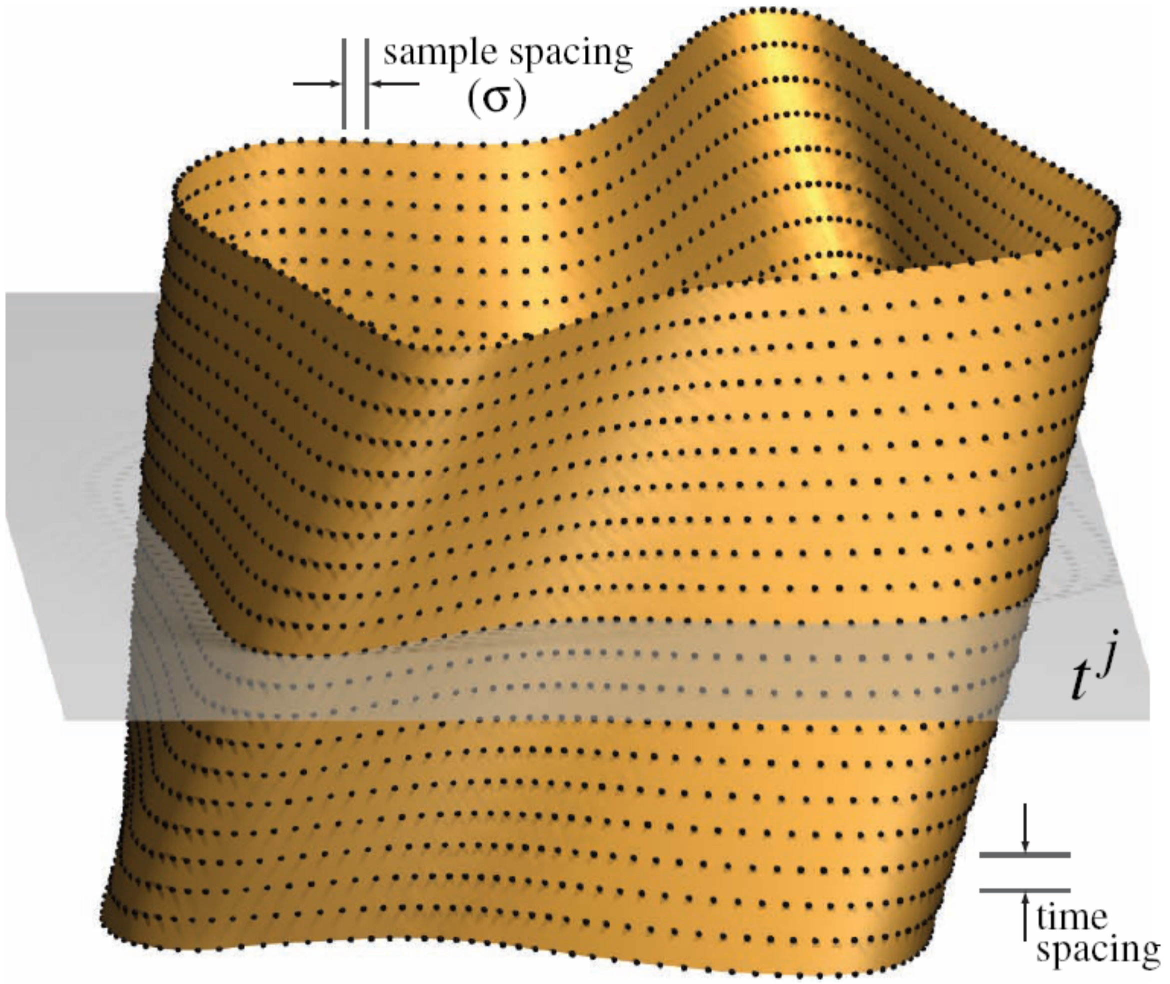
$$\widetilde{P}^j \equiv \{\widetilde{\mathbf{p}}_i^j\} := \{(\mathbf{p}_i^j, t^j), \mathbf{p}_i^j \in \mathbb{R}^d, t^j \in \mathbb{R}\}$$

Space-time Surface





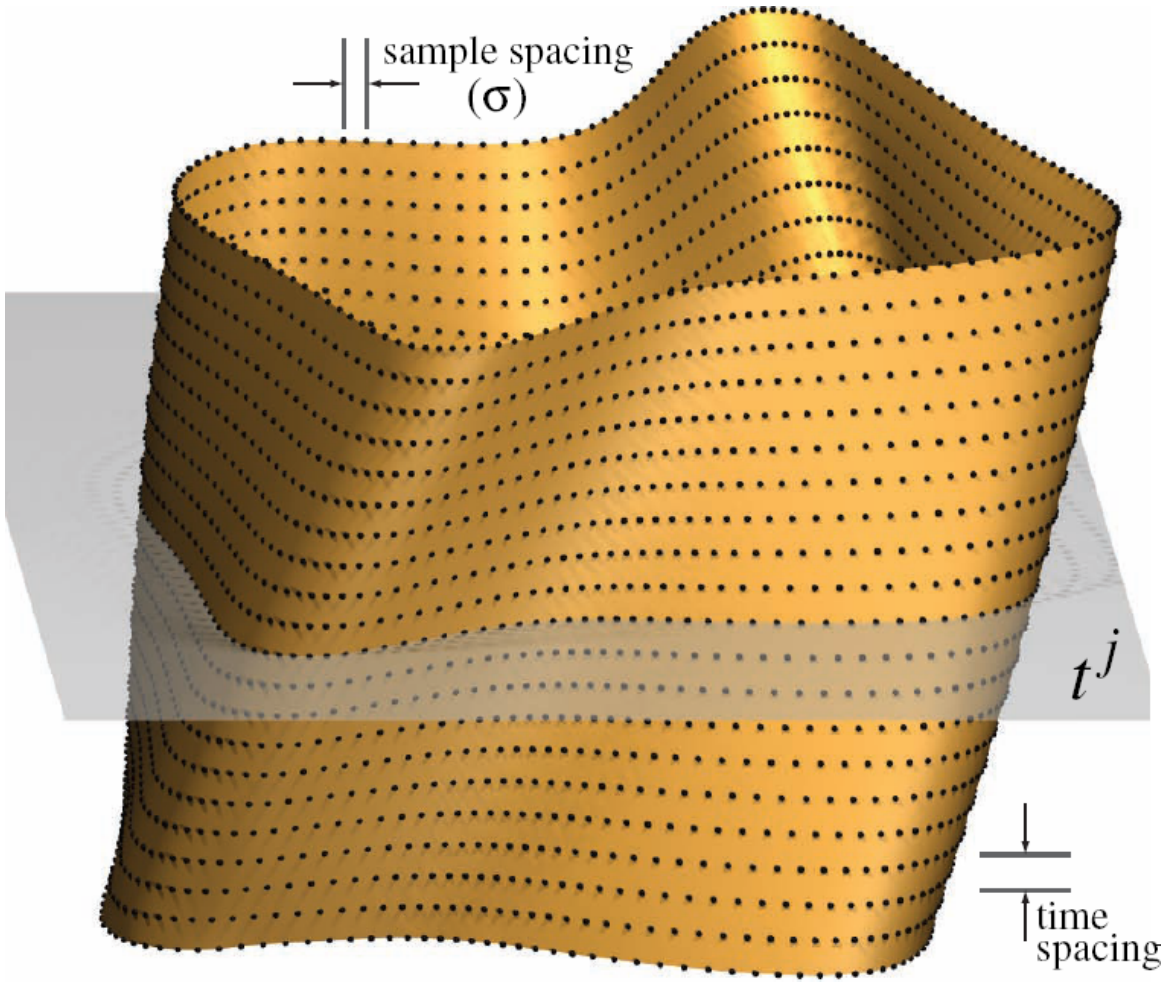
Space-time Surface



Δt^j

$\tilde{\mathbf{p}}_i^j \quad \rightarrow \quad \widetilde{\alpha}_j(\tilde{\mathbf{p}}_i^j) = \left(\mathbf{R}_j \mathbf{p}_i^j + \mathbf{t}_j, t^j + \Delta t^j\right)$

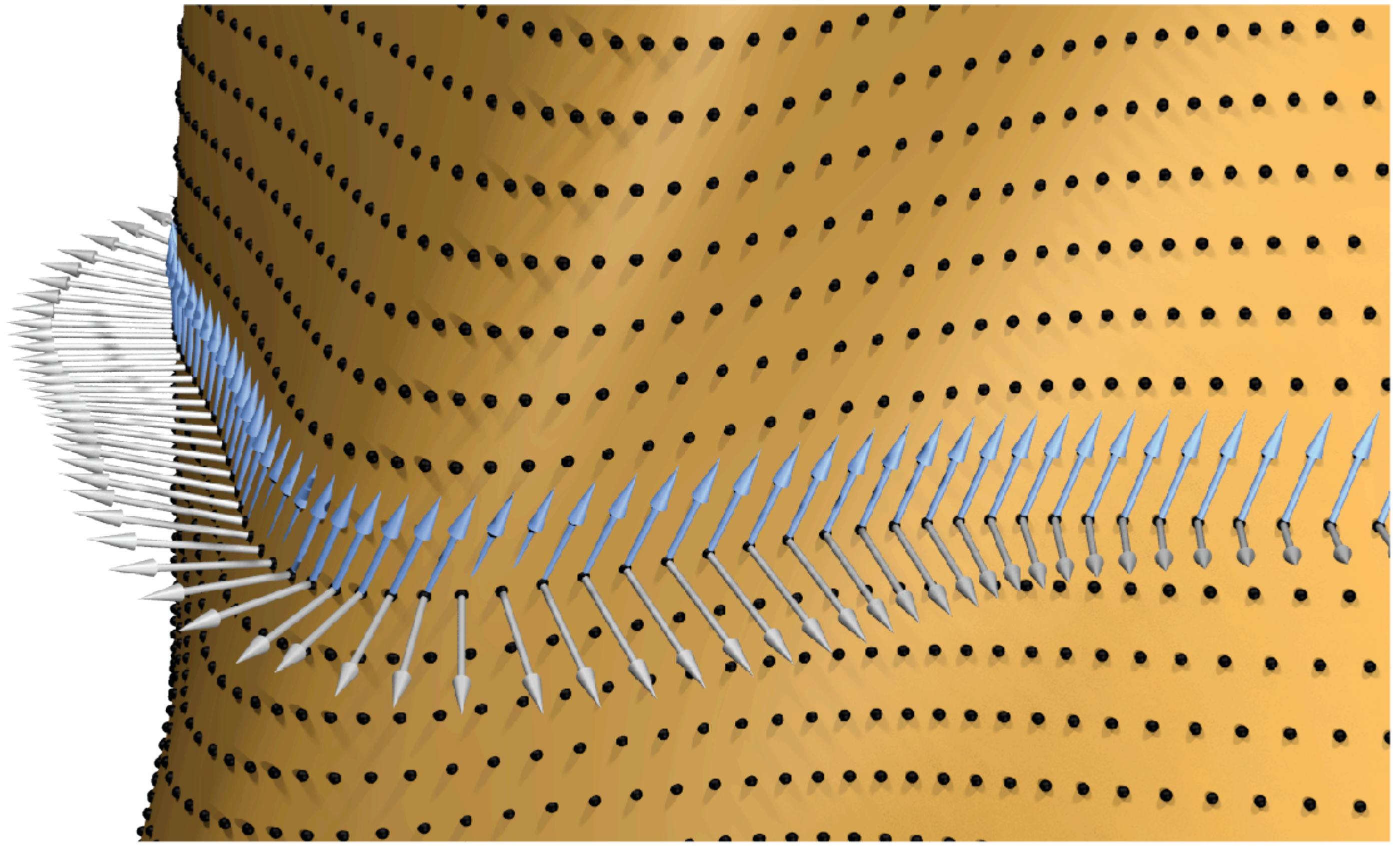
Space-time Surface



$$\widetilde{\mathbf{p}}_i^j \quad \rightarrow \quad \widetilde{\alpha}_j(\widetilde{\mathbf{p}}_i^j) = \left(\mathbf{R}_j \mathbf{p}_i^j + \mathbf{t}_j, t^j + \Delta t^j\right)$$

$$\widetilde{\alpha}_j = \operatorname{argmin} \sum_{i=1}^{|P^j|} d^2(\widetilde{\alpha}_j(\widetilde{\mathbf{p}}_i^j), S)$$

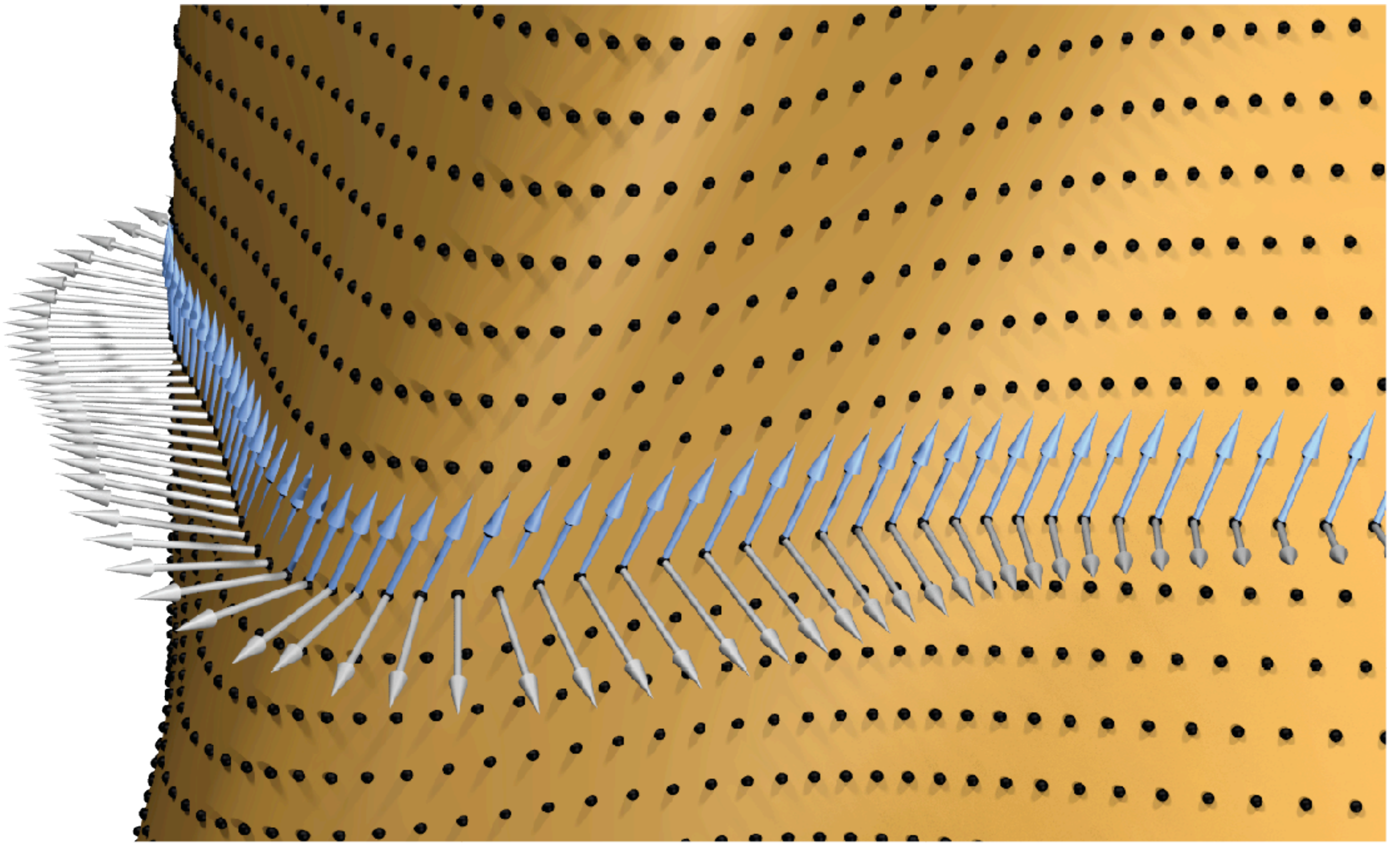
Spacetime Velocity Vectors



Tangential point movement → velocity vectors orthogonal to surface normals

$$\widetilde{\alpha}_j = \operatorname{argmin} \sum_{i=1}^{|P^j|} d^2(\widetilde{\alpha}_j(\widetilde{\mathbf{p}}_i^j), S)$$

Spacetime Velocity Vectors



Tangential point movement → velocity vectors orthogonal to surface normals

Final Steps

(rigid) velocity vectors → $\widetilde{\mathbf{v}}(\widetilde{\mathbf{p}}_i^j) = (\mathbf{c}_j \times \mathbf{p}_i^j + \overline{\mathbf{c}}_j, 1)$

$$\min_{\mathbf{c}_j, \overline{\mathbf{c}}_j} \sum_{i=1}^{|P^j|} w_i^j \left[(\mathbf{c}_j \times \mathbf{p}_i^j + \overline{\mathbf{c}}_j, 1) \cdot \widetilde{\mathbf{n}}_i^j \right]^2$$

Final Steps

(rigid) velocity vectors !

$$\widetilde{\mathbf{v}}(\widetilde{\mathbf{p}}_i^j) = (\mathbf{c}_j \times \mathbf{p}_i^j + \bar{\mathbf{c}}_j, 1)$$

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

$$A\mathbf{x} + \mathbf{b} = 0$$

$$A = \sum_{i=1}^{|P^j|} w_i^j \begin{bmatrix} \bar{\mathbf{n}}_i^j \\ \mathbf{p}_i^j \times \bar{\mathbf{n}}_i^j \end{bmatrix} \begin{bmatrix} \bar{\mathbf{n}}_i^{j\,T} & (\mathbf{p}_i^j \times \bar{\mathbf{n}}_i^j)^T \end{bmatrix}$$

$$\mathbf{b} = \sum_{i=1}^{|P^j|} w_i^j n_i^j \begin{bmatrix} \bar{\mathbf{n}}_i^j \\ \mathbf{p}_i^j \times \bar{\mathbf{n}}_i^j \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} \bar{\mathbf{c}}_j \\ \mathbf{c}_j \end{bmatrix}$$

Registration Algorithm

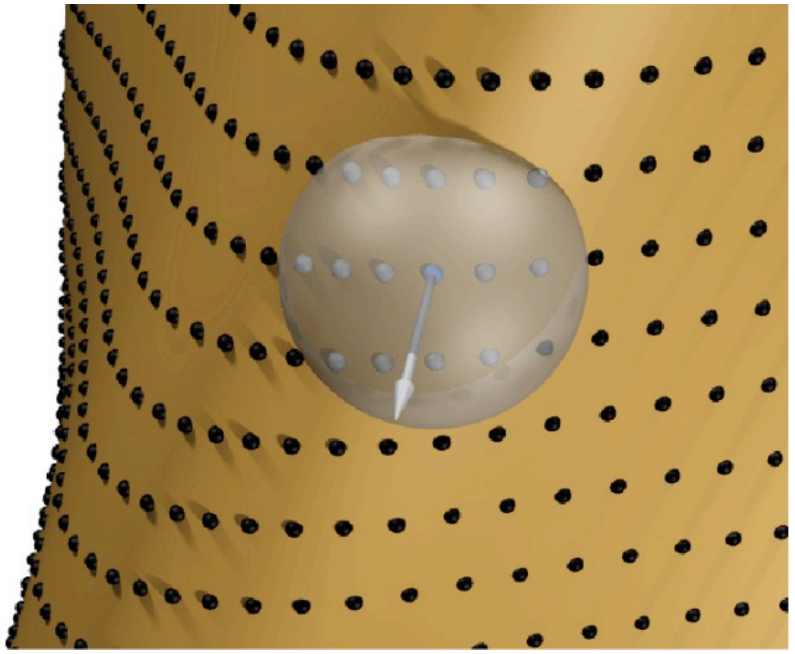
1. **Compute time coordinate spacing (σ), and form space-time surface.**

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

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

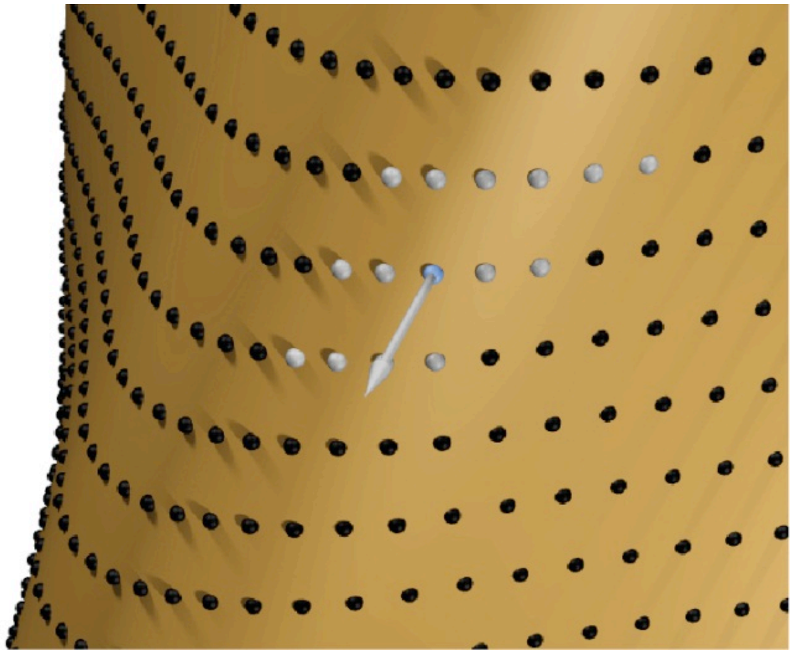
4. **Convert velocity vectors to rotation matrix + translation vector using Plücker coordinates and quarternions.**

Normal Estimation: PCA Based



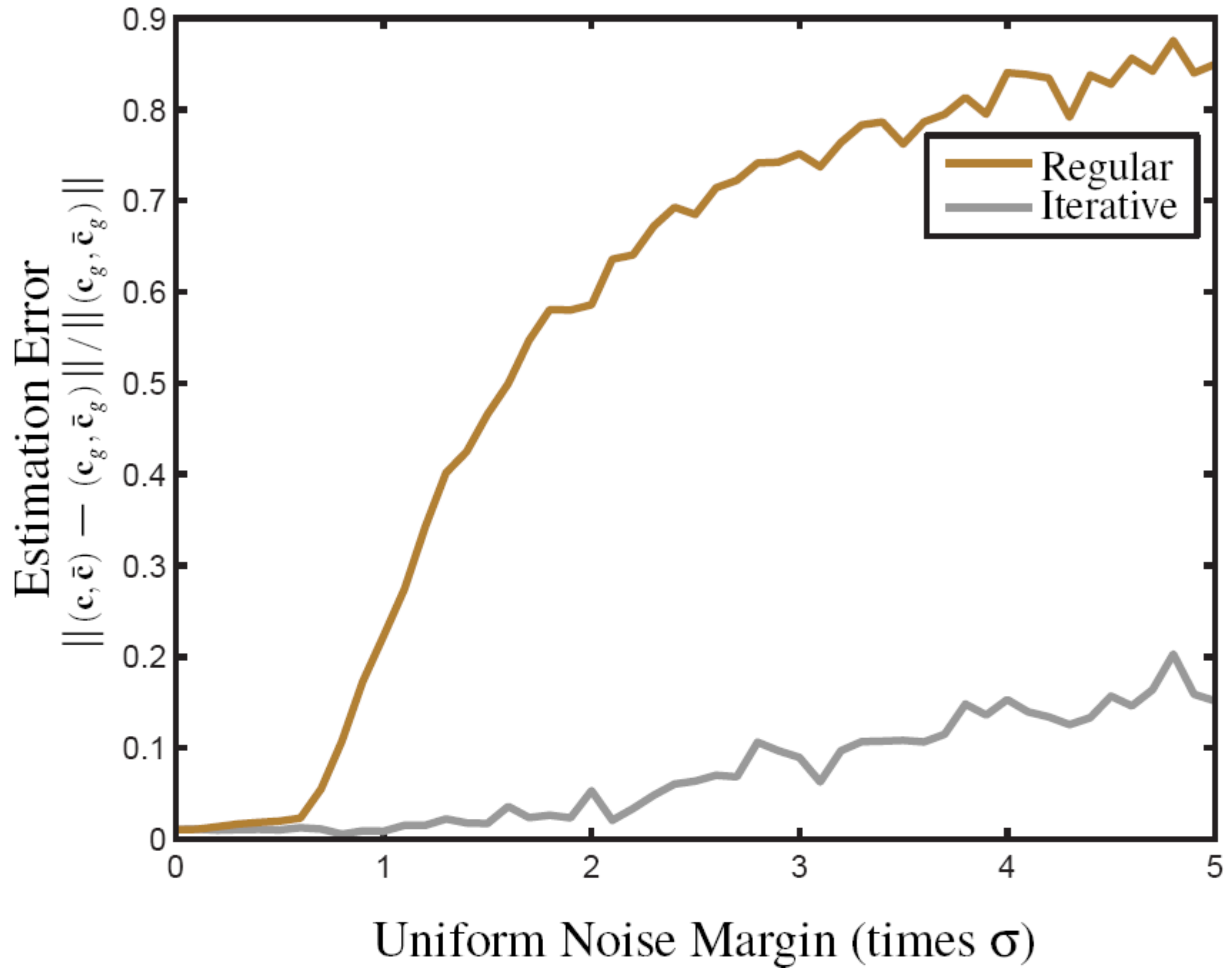
Plane fitting using PCA using chosen neighborhood points.

Normal Estimation: Iterative Refinement



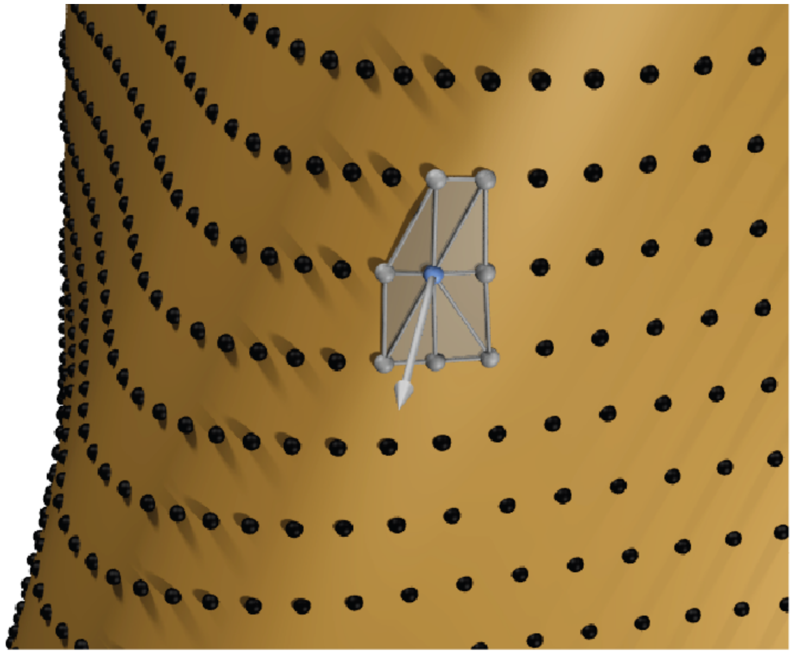
Update neighborhood with current velocity estimate.

Normal Refinement: Effect of Noise



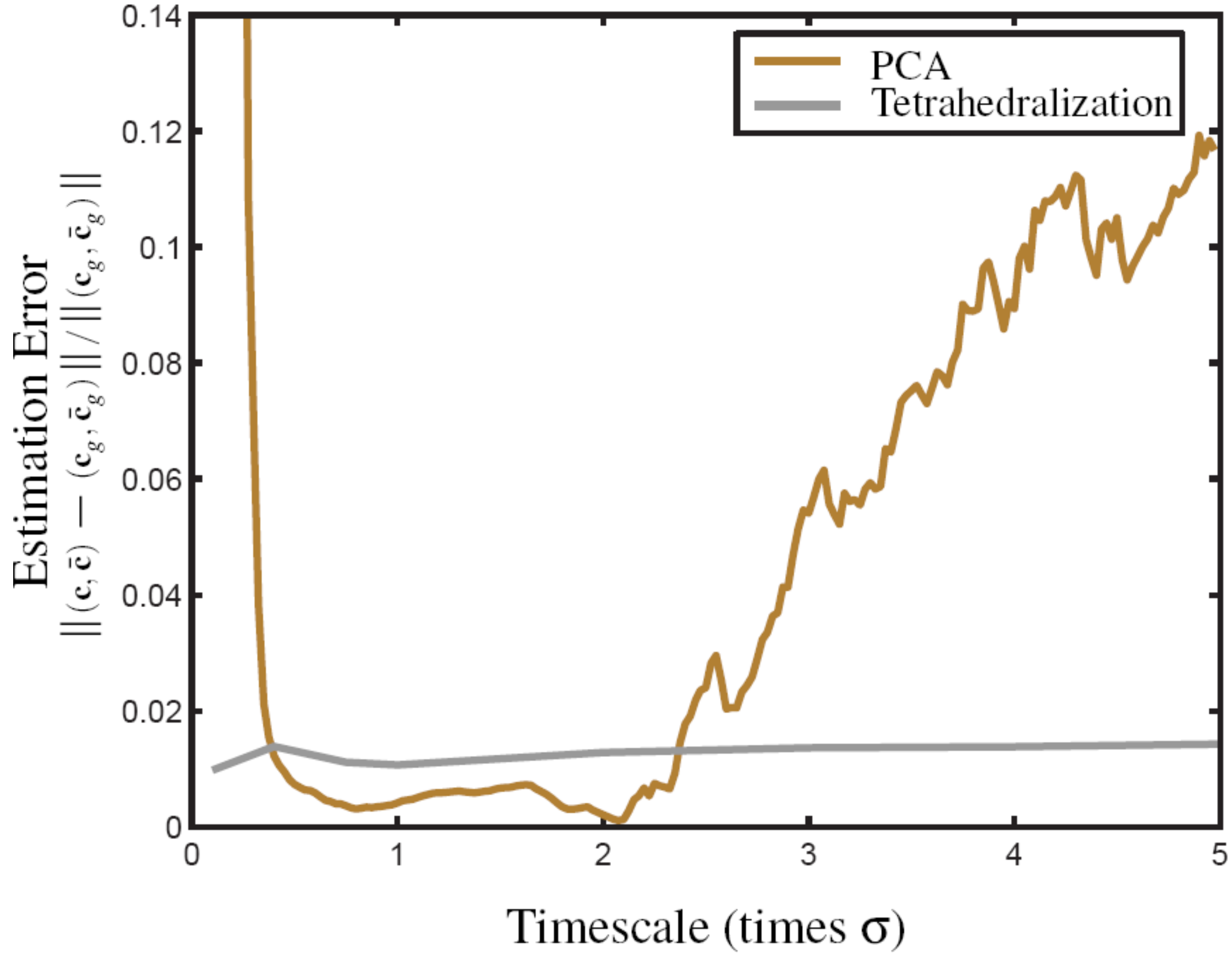
Stable, but more expensive.

Normal Estimation: Local Triangulation



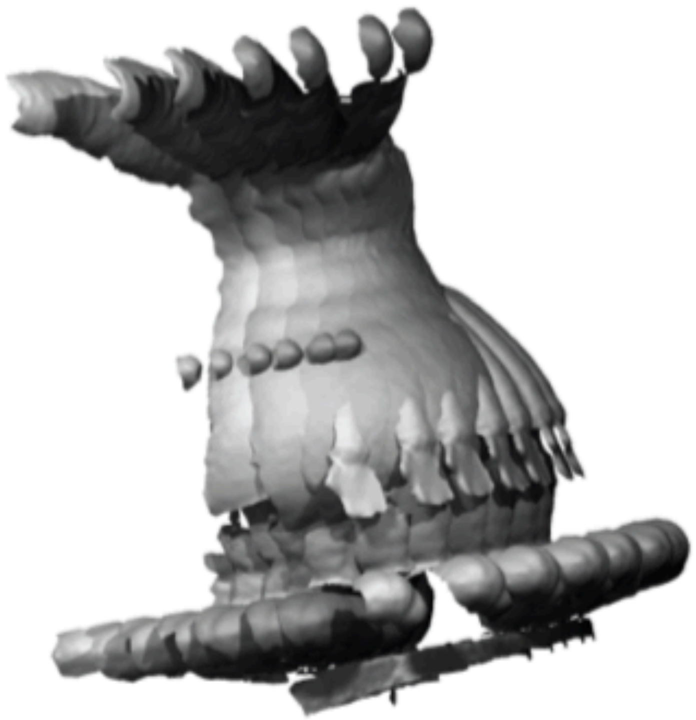
Perform local surface triangulation (tetrahedralization).

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)

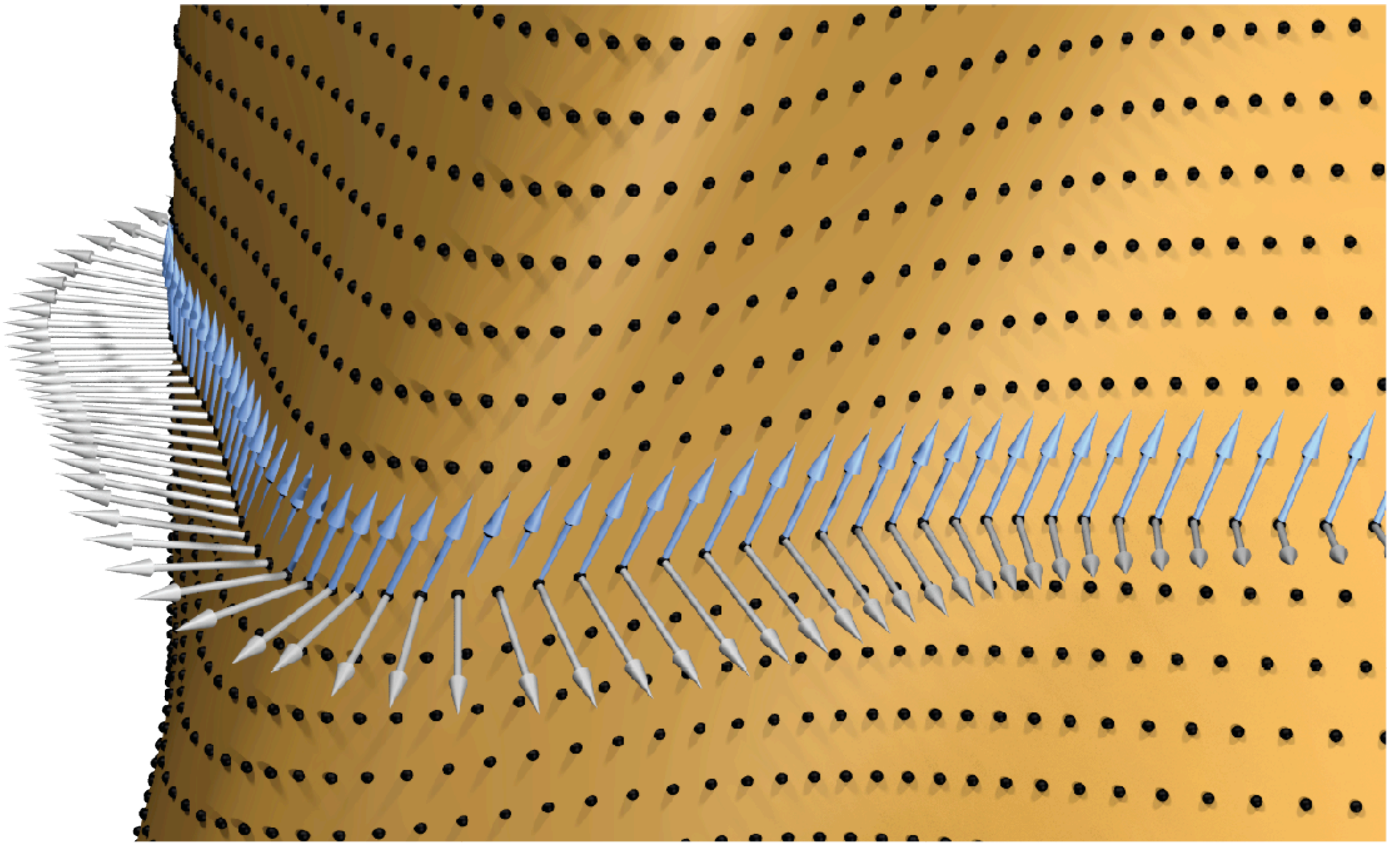
2200 pointclouds scanned at 17 Hz

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

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

Cluster points, and solve smaller systems.

Propagate solutions with regularization.

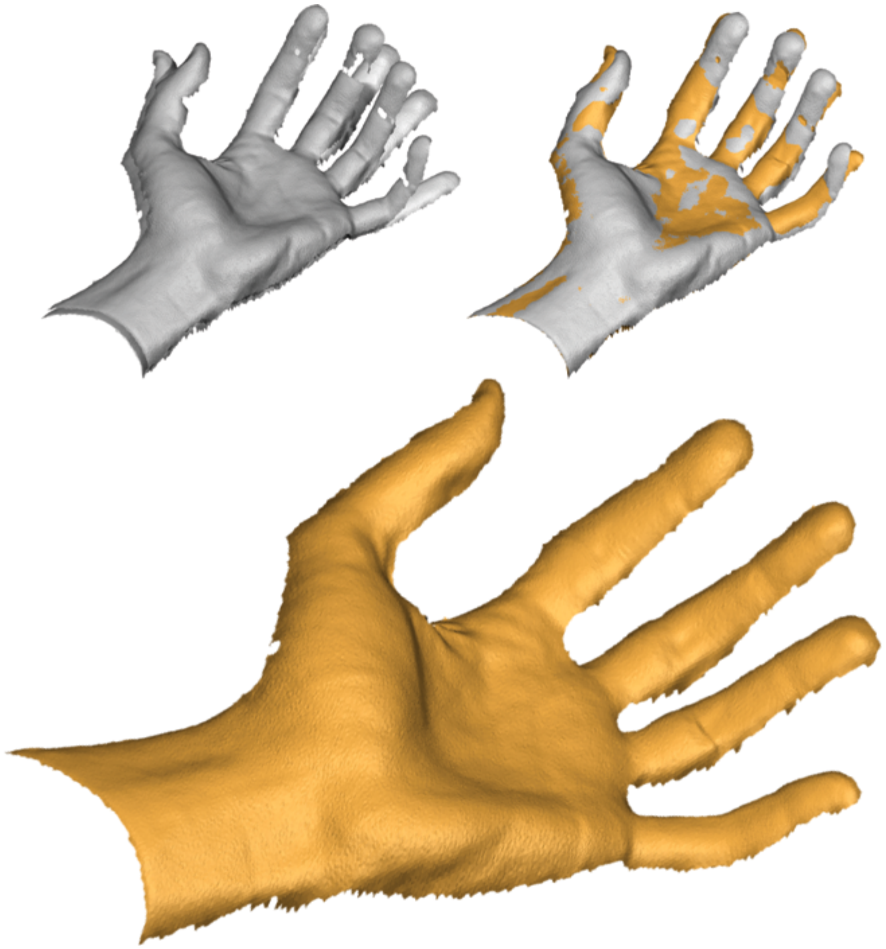
Deformable: Hand (100 frames)

Hand Input frames & registered result

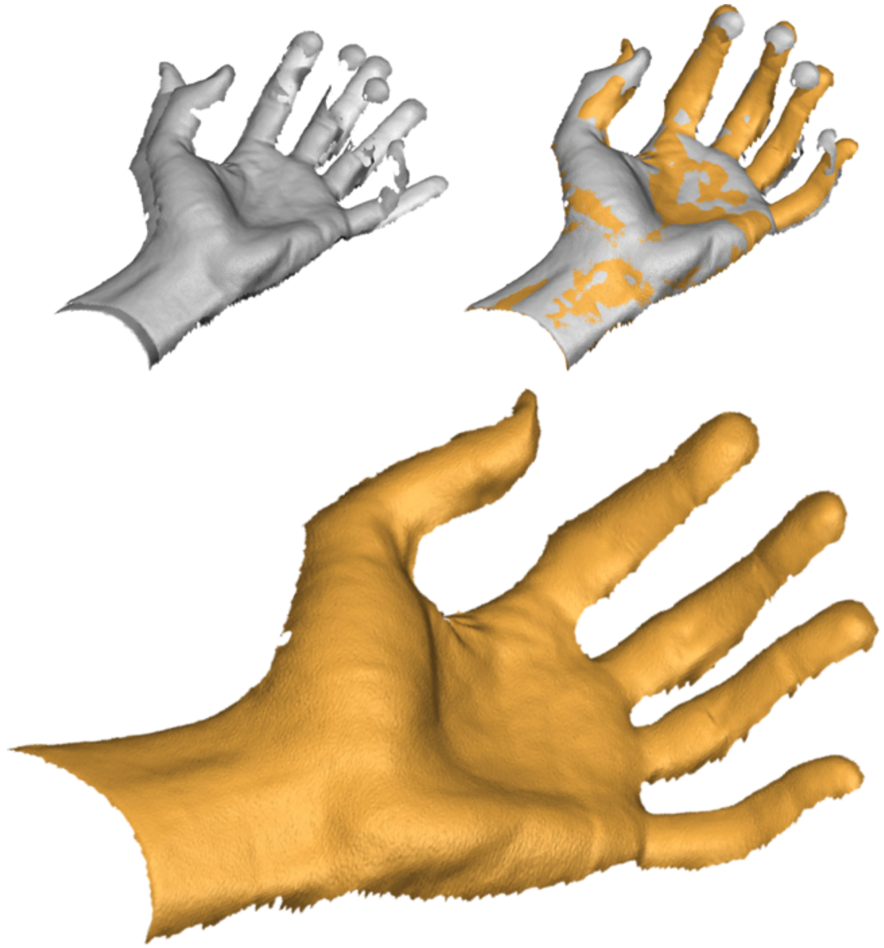
100 pointclouds scanned at 17 Hz
transformations only for adjacent frames considered
no global error correction
no noise smoothing

first frame is tracked
deformation due to severly missing data (e.g. ring finger)

Deformable: Hand (100 frames)



scan #1 : scan #50



scan #1 : scan #100

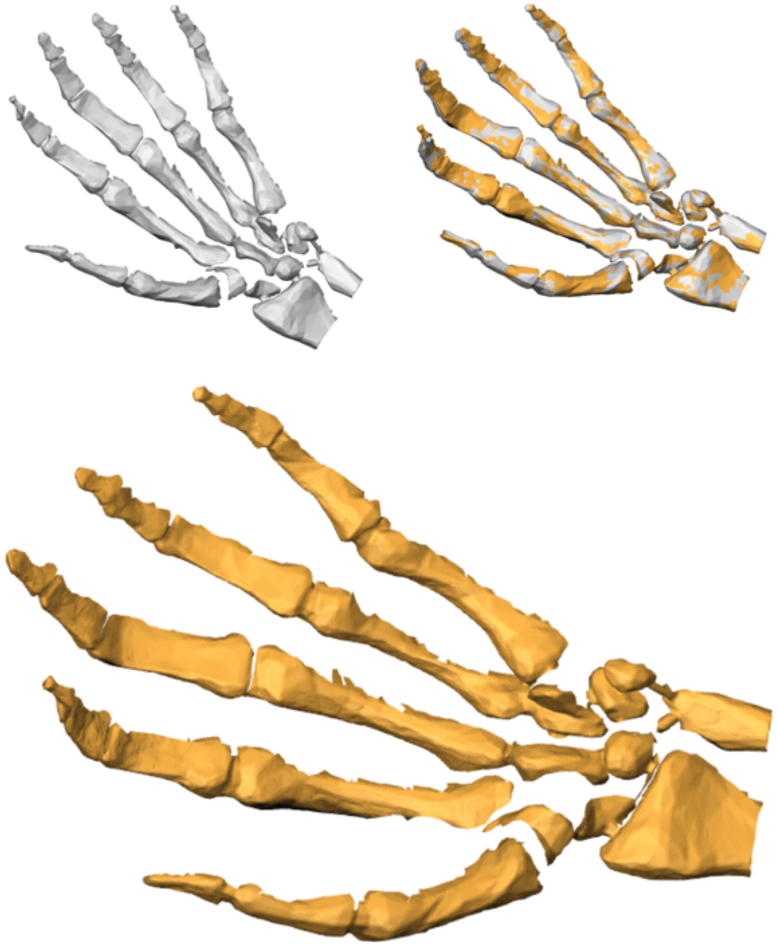
Deformation + scanner motion: Skeleton (100 frames)

Grasp   Input frames & registered result

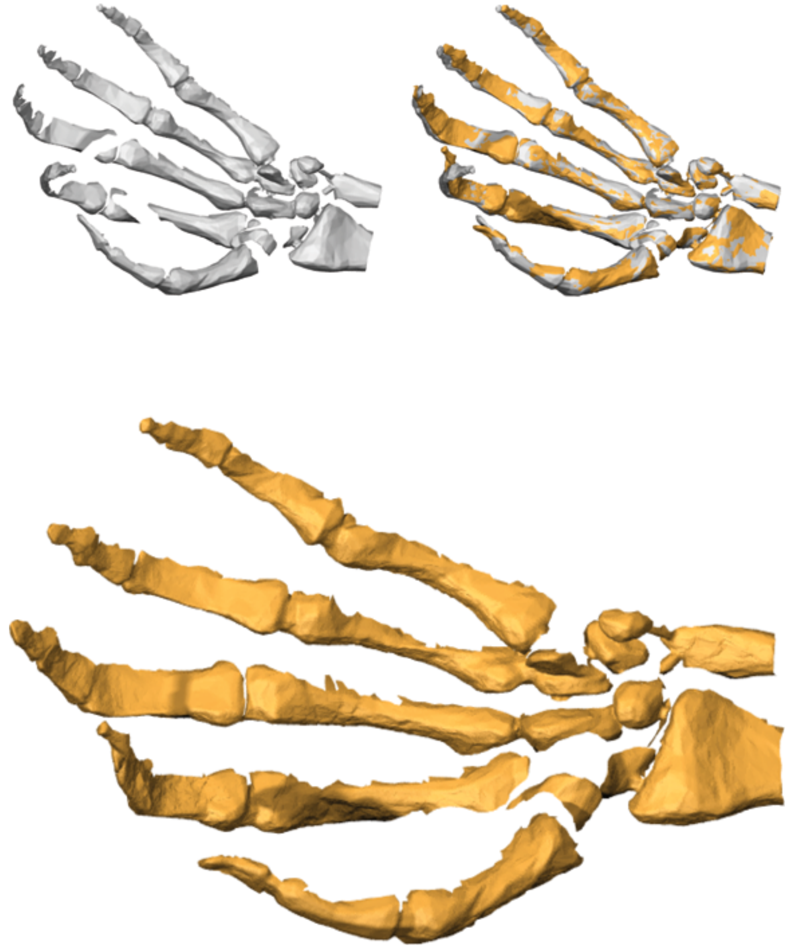
100 **simulated** scan data sets
simultaneous object deformation and camera motion
transformations only for adjacent frames considered
no global error correction, no noise smoothing

first frame is tracked
data completion (e.g. for middle finger)

Deformation + scanner motion: Skeleton (100 frames)

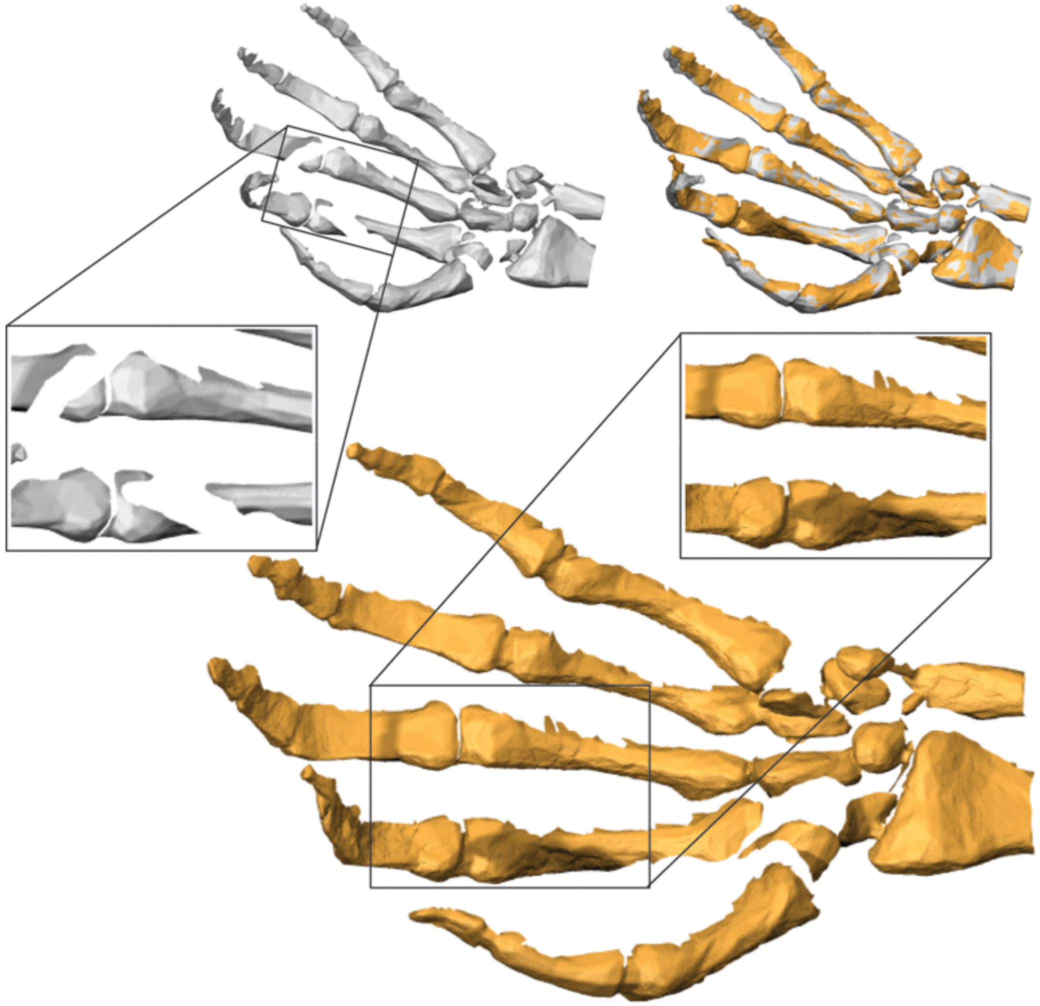


scan #1 : scan #50



scan #1 : scan #100

Deformation + scanner motion: Skeleton (100 frames)



rigid components

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

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

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 → time and space

thank you