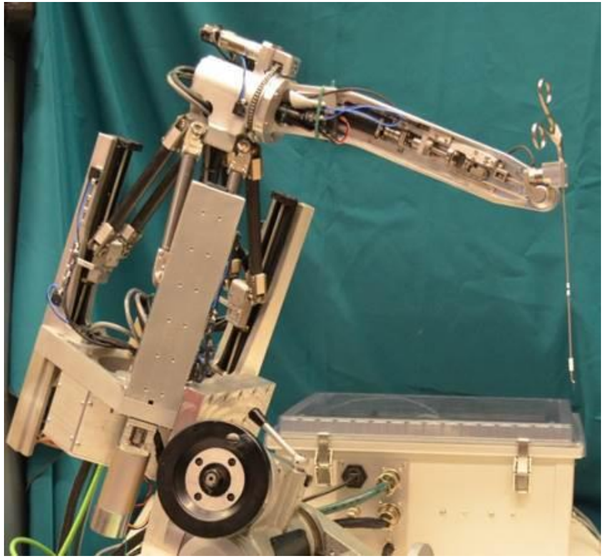
Iterative Most-Likely Point Registration (IMLP): A Robust Algorithm for Computing Optimal Shape Alignment

Seth D. Billings[1]*, Emad M. Boctor[2,1,3], Russell H. Taylor[1]

Zach Paine Sabin

REMS Registration Project

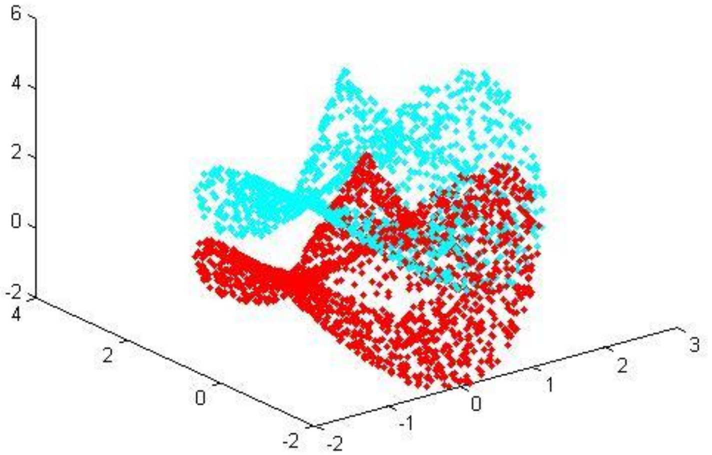


Motivation

We are using Seth Billings’ code, and this is an option for his registration

It could potentially be more accurate and speed is not a major concern

Iterative Closest Point



Algorithm 1. Iterative Closest Point (ICP)

input: Source shape as point cloud: $X = \{\vec{x}_i\}$

Target shape: Ψ

Initial transformation estimate: $[R_0, \vec{t}_0]$

output: Final transformation $[R, \vec{t}]$ that aligns the shapes X and Ψ

1 Initialize transformation: $[R, \vec{t}] \leftarrow [R_0, \vec{t}_0]$

2 while *not converged* **do**

3 Compute closest-point correspondences $y = \left\{\vec{Y}_i\right\}$:

$$\vec{y}_i \leftarrow \mathrm{C_{cp}}(R\vec{x}_i + \vec{t}, \Psi)$$

4 Update the transformation to minimize $E_{LS}(X, Y)$:

$$[R, \vec{t}\,] \leftarrow \underset{[R, \vec{t}\,]}{\operatorname{argmin}} \sum_{i=1}^{n} \| \vec{y}_i - R\vec{x}_i - \vec{t}\ \|_2^2$$

5 end

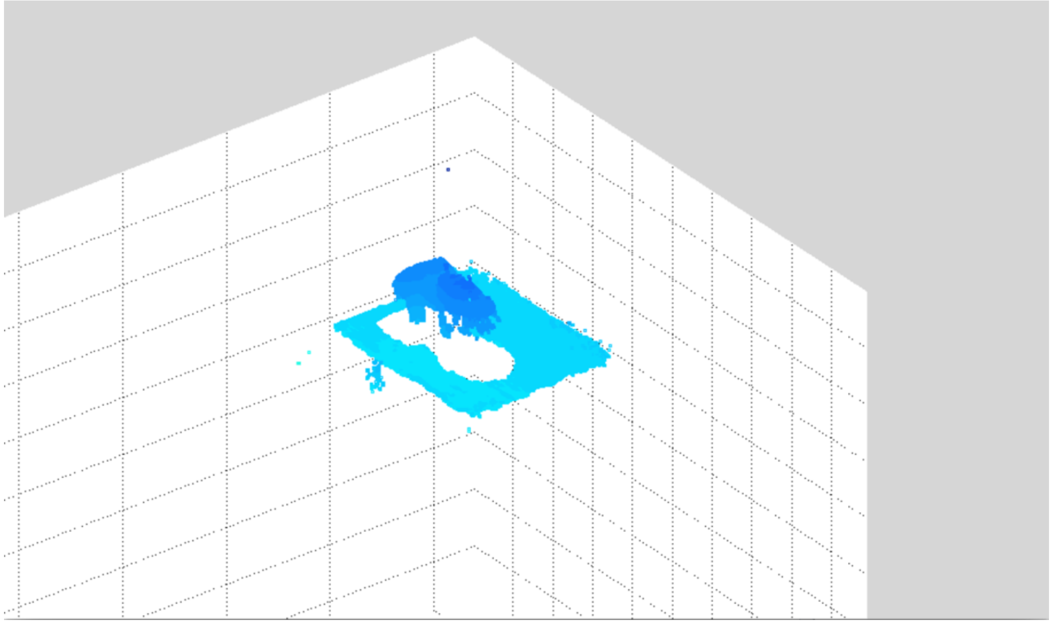
Problem

While ICP is fast, it often leads to high registration error and does not effectively account for noise in the data.

IMLP uses a noise model and finds the most likely correspondences of the points which allows it to account for noise and be more accurate.

Significance

While IMLP is slower than ICP, it allows for much greater accuracy, especially on noisy datasets.



Algorithm Overview

Quick discussion of each step of algorithm

Algorithm Overview

input: Source shape as point cloud: $X = \{\vec{x}_i\}$

Target shape: Ψ

Measurement-error covariances: $M_X = \{M_{Xi}\}$, M_Ψ

Surface-model covariances: $M_{SX} = \{M_{SXi}\}$, $M_{S\Psi}$

Initial transformation estimate: $[R_0, \vec{t}_0]$

Upper bound on match uncertainty: $\sigma^2_{\max}$ (default: ∞)

Chi-square threshold value for outliers: χ^2_{thresh} (default: 7.81)

output: Final transformation $[R, \vec{t}]$ that aligns the shapes X and Ψ

1 Initialize transformation: $[R, \vec{t}] \leftarrow [R_0, \vec{t}_0]$

2 Initialize noise model: $\sigma^2 \leftarrow 0$

3 Compute initial correspondences (Equ. 8):

$$[\vec{y}_i, M_{yi}, \mathrm{M}_{\mathrm{S}\,yi}] \leftarrow \mathrm{C}_{\mathrm{mlp}}(\vec{x}_i, \Psi, I, I, R, \vec{t},)$$

4 Skip to Step 6

5 Compute most-likely correspondences (Equ. 8):

$$[\vec{y}_i, M_{yi}, M_{\mathrm{S}yi}] \leftarrow \mathrm{C}_{\mathrm{mlp}}(\vec{x}_i, \Psi, M_{xi} + M_{\mathrm{S}xi} + \sigma^2 I, M_\Psi + M_{\mathrm{S}\Psi}, R, \vec{t},)$$

6 Update the match-uncertainty noise-model term (Equ. 4):

$$\sigma^2 \leftarrow \min\left(\frac{1}{N_{\text{inlier}}} \sum_{i \in \text{inliers}} \|\vec{y}_i - R\vec{x}_i - \vec{t}\,\|_2^2, \sigma^2_{\max}\right)$$

7 Identify outliers using a chi-square test (Equ. 6):

$$(\vec{x}_i, \vec{y}_i) \text{ is outlier if } \mathrm{E}_{\text{SqrMahalDist}}(\vec{x}_i, \vec{y}_i, M_{xi}, M_{yi} + \sigma^2 I, R, \vec{t}\,) > \chi^2_{\text{thresh}}$$

and update the outlier noise-model terms (Equ. 7):

$$\varphi_i = \begin{cases} 9\|\vec{y}_i - R\vec{x}_i - \vec{t}\,\|_2^2 & \text{if } (\vec{x}_i, \vec{y}_i) \text{ is an outlier} \\ 0 & \text{otherwise} \end{cases}$$

8 Set the noise-model covariances for the registration phase:

$$M^*_{xi} \leftarrow M_{xi} \qquad M^*_{yi} \leftarrow M_{yi} + M_{\mathrm{S}yi} + M_{\mathrm{S}xi} + \frac{\varphi_i}{2} I' + \frac{\varphi_i}{2} I + \sigma^2 I$$

9 Update the transformation to align the corresponding point sets by GTLS (Equ. 20):

$$[R, \vec{t}] \leftarrow \underset{[R, \vec{t}]}{\text{argmin}} \sum_{i=1}^{n} (\vec{y}_i - R\vec{x}_i - \vec{t}\,)^T (R M^*_{xi} R^T + M^*_{yi})^{-1} (\vec{y}_i - R\vec{x}_i - \vec{t}\,)$$

10 **if** *not converged* **then** goto Step 5

Algorithm Inputs

Target Shape

Covariances (measurement error and surface-model)

Transformation Estimate

Match Uncertainty

Outlier Threshold

Finding Correspondences

Probability that a transformed source point corresponds to a specific target point is

$$\mathrm{L_{match}}(\vec{x}, \vec{y}, M_{\mathrm{x}}, M_{\mathrm{y}}, R, \vec{t}) = \frac{1}{\sqrt{(2\pi)^3 |RM_{\mathrm{x}}R^T + M_{\mathrm{y}}|}} e^{-\frac{1}{2}(\vec{y}-R\vec{x}-\vec{t})^T (RM_{\mathrm{x}}R^T + M_{\mathrm{y}})^{-1} (\vec{y}-R\vec{x}-\vec{t})} \quad (3)$$

Finding Correspondences

input: Source point: $\vec{x}$

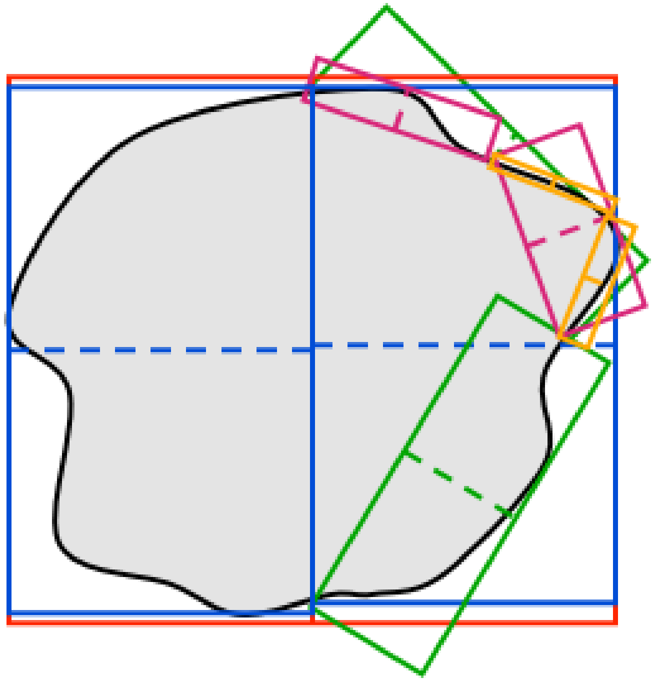
Source-point noise model: M_x

PD tree containing target shape (Ψ) and target noise model (M_Ψ): T

Current transformation: $[R, \vec{t}]$

Prior most-likely match for this source point: $(\vec{y}_{\text{pre}}, M_{\text{y\_pre}})$

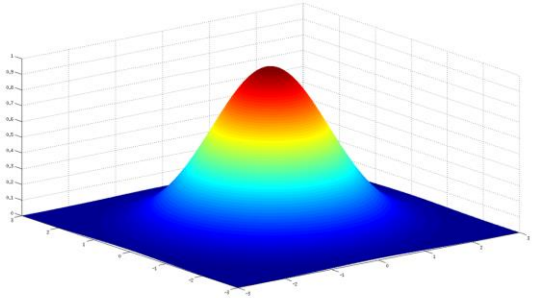
output: Most-likely match and its corresponding noise model: $(\vec{y}, M_{\text{y}})$

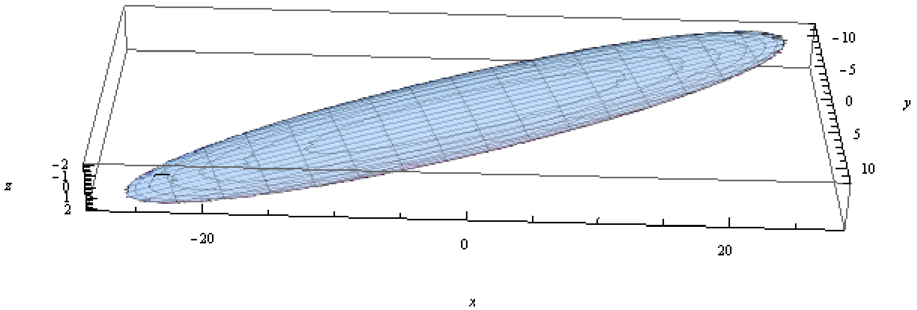


Dr. Taylor CIS 1 Lecture

Noise Model

$$\sigma^2 = \frac{1}{N_{\text{inlier}}} \sum_{i \in \text{inlier}} \|\vec{y}_i - R\vec{x}_i - \vec{t}\,\|_2^2$$



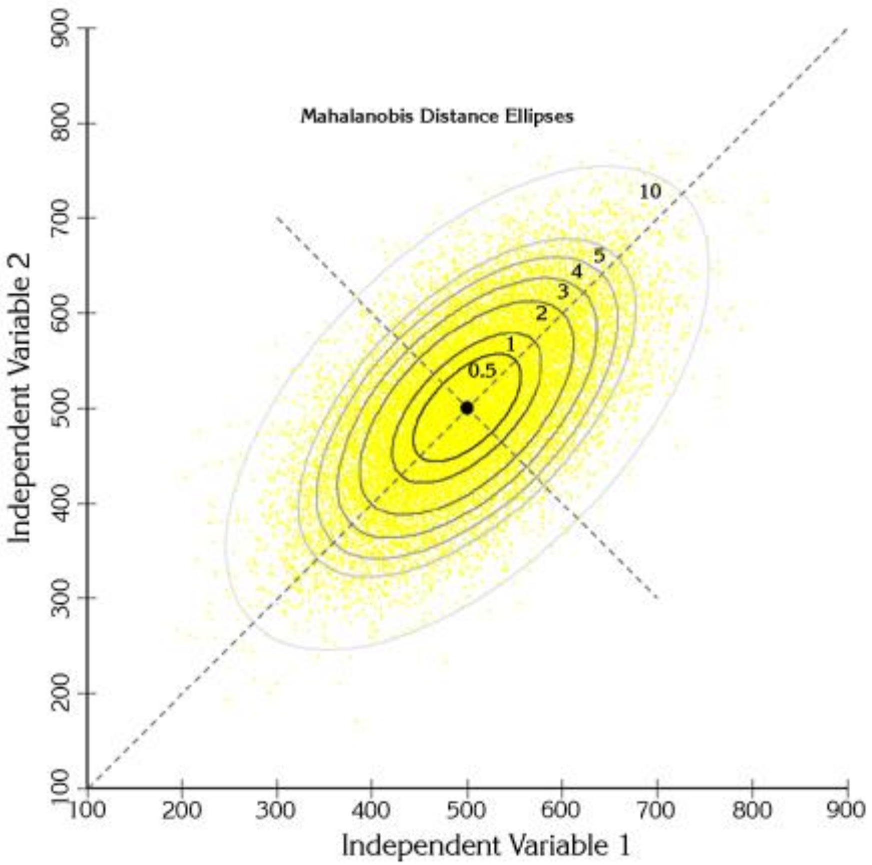


Outlier Identification

Mahalanobis distance

Number of standard deviations away on each Principal Component Axis

$$E_{\mathrm{SqrMahalDist}}(\vec{x}, \vec{y}, M_{\mathrm{x}}, M_{\mathrm{y}}, R, \vec{t}) > \mathrm{chi2inv}(p, 3) = \chi^2_{\mathrm{thresh}} \quad (6)$$



Weighing Outliers

What to do with outliers?

Increase the variance in their noise models so that they have less effect on Most Likely Point

$$\varphi_i = \begin{cases} 9\|\vec{y}_i - R\vec{x}_i - \vec{t}\,\|_2^2 & \text{if } (\vec{x}_i, \vec{y}_i) \text{ is an outlier} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Aligning Corresponding Points

$$L_{\mathrm{total}}(X, Y, M_{\mathrm{X}}, M_{\mathrm{Y}}) = \max_{[R, \vec{t}]} \prod_{i=1}^{n} \mathrm{L}_{\mathrm{match}}(\vec{x}_i, \vec{y}_i, M_{\mathrm{x}i}, M_{\mathrm{y}i})$$

$$\sum_{i=1}^{n} \log |R M_{\mathrm{x}i} R^T + M_{\mathrm{y}i}| + \sum_{i=1}^{n} (\vec{y}_i - R\vec{x}_i - \vec{t})^T (R M_{\mathrm{x}i} R^T + M_{\mathrm{y}i})^{-1} (\vec{y}_i - R\vec{x}_i - \vec{t}) \ . \qquad (19)$$

Aligning Corresponding Points

$$L_{\text{total}}(X, Y, M_X, M_Y) = \max_{[R,\vec{t}]} \prod_{i=1}^{n} \mathrm{L}_{\text{match}}(\vec{x}_i, \vec{y}_i, M_{xi}, M_{yi})$$

$$\cancel{\sum_{i=1}^{n} \log |RM_{xi}R^T + M_{yi}|} + \sum_{i=1}^{n} (\vec{y}_i - R\vec{x}_i - \vec{t})^T (RM_{xi}R^T + M_{yi})^{-1} (\vec{y}_i - R\vec{x}_i - \vec{t}) \ . \tag{19}$$

$$\mathrm{E}_{\text{GTLS}}(X, Y, M_X, M_Y) = \min_{[R,\vec{t}]} \sum_{i=1}^{n} (\vec{x}_i - \vec{x}_i^*)^T M_{xi}^{-1} (\vec{x}_i - \vec{x}_i^*) + \sum_{i=1}^{n} (\vec{y}_i - \vec{y}_i^*)^T M_{yi}^{-1} (\vec{y}_i - \vec{y}_i^*)$$

$$\text{subject to:} \quad \vec{y}_i^* = R\vec{x}_i^* + \vec{t} \tag{21}$$

Algorithm Overview

input: Source shape as point cloud: $X = \{\vec{x}_i\}$

Target shape: Ψ

Measurement-error covariances: $M_X = \{M_{Xi}\}$, M_Ψ

Surface-model covariances: $M_{SX} = \{M_{SXi}\}$, $M_{S\Psi}$

Initial transformation estimate: $[R_0, \vec{t}_0]$

Upper bound on match uncertainty: $\sigma^2_{\max}$ (default: ∞)

Chi-square threshold value for outliers: χ^2_{thresh} (default: 7.81)

output: Final transformation $[R, \vec{t}]$ that aligns the shapes X and Ψ

1 Initialize transformation: $[R, \vec{t}] \leftarrow [R_0, \vec{t}_0]$

2 Initialize noise model: $\sigma^2 \leftarrow 0$

3 Compute initial correspondences (Equ. 8):

$$[\vec{y}_i, M_{yi}, M_{S\,yi}] \leftarrow \mathrm{C_{mlp}}(\vec{x}_i, \Psi, I, I, R, \vec{t},)$$

4 Skip to Step 6

5 Compute most-likely correspondences (Equ. 8):

$$[\vec{y}_i, M_{yi}, M_{Syi}] \leftarrow \mathrm{C_{mlp}}(\vec{x}_i, \Psi, M_{xi} + M_{Sxi} + \sigma^2 I, M_\Psi + M_{S\Psi}, R, \vec{t},)$$

6 Update the match-uncertainty noise-model term (Equ. 4):

$$\sigma^2 \leftarrow \min\left(\frac{1}{N_{\text{inlier}}} \sum_{i \in \text{inliers}} \| \vec{y}_i - R\vec{x}_i - \vec{t} \|_2^2, \sigma^2_{\max}\right)$$

7 Identify outliers using a chi-square test (Equ. 6):

$$(\vec{x}_i, \vec{y}_i) \text{ is outlier if } \mathrm{E_{SqrMahalDist}}(\vec{x}_i, \vec{y}_i, M_{xi}, M_{yi} + \sigma^2 I, R, \vec{t}) > \chi^2_{\text{thresh}}$$

and update the outlier noise-model terms (Equ. 7):

$$\varphi_i = \begin{cases} 9\|\vec{y}_i - R\vec{x}_i - \vec{t}\,\|_2^2 & \text{if } (\vec{x}_i, \vec{y}_i) \text{ is an outlier} \\ 0 & \text{otherwise} \end{cases}$$

8 Set the noise-model covariances for the registration phase:

$$M^*_{xi} \leftarrow M_{xi} \quad M^*_{yi} \leftarrow M_{yi} + M_{Syi} + M_{Sxi} + \frac{\varphi_i}{2} I' + \frac{\varphi_i}{2} I + \sigma^2 I$$

9 Update the transformation to align the corresponding point sets by GTLS (Equ. 20):

$$[R, \vec{t}] \leftarrow \underset{[R,\vec{t}]}{\operatorname{argmin}} \sum_{i=1}^{n} (\vec{y}_i - R\vec{x}_i - \vec{t})^T (R M^*_{xi} R^T + M^*_{yi})^{-1} (\vec{y}_i - R\vec{x}_i - \vec{t})$$

10 **if** *not converged* **then** goto Step 5

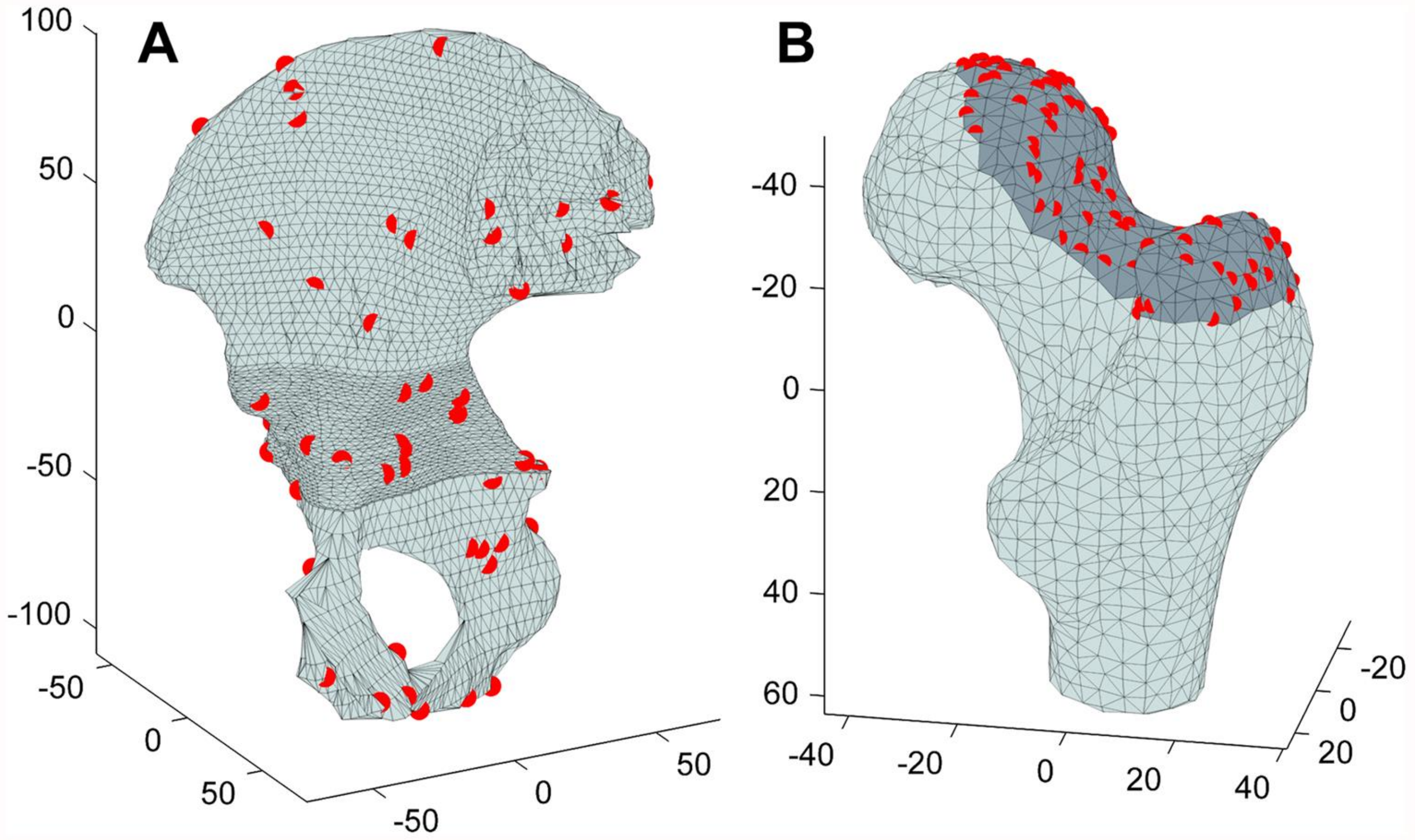
Experimental Overview

Ran 8 Different Experiments

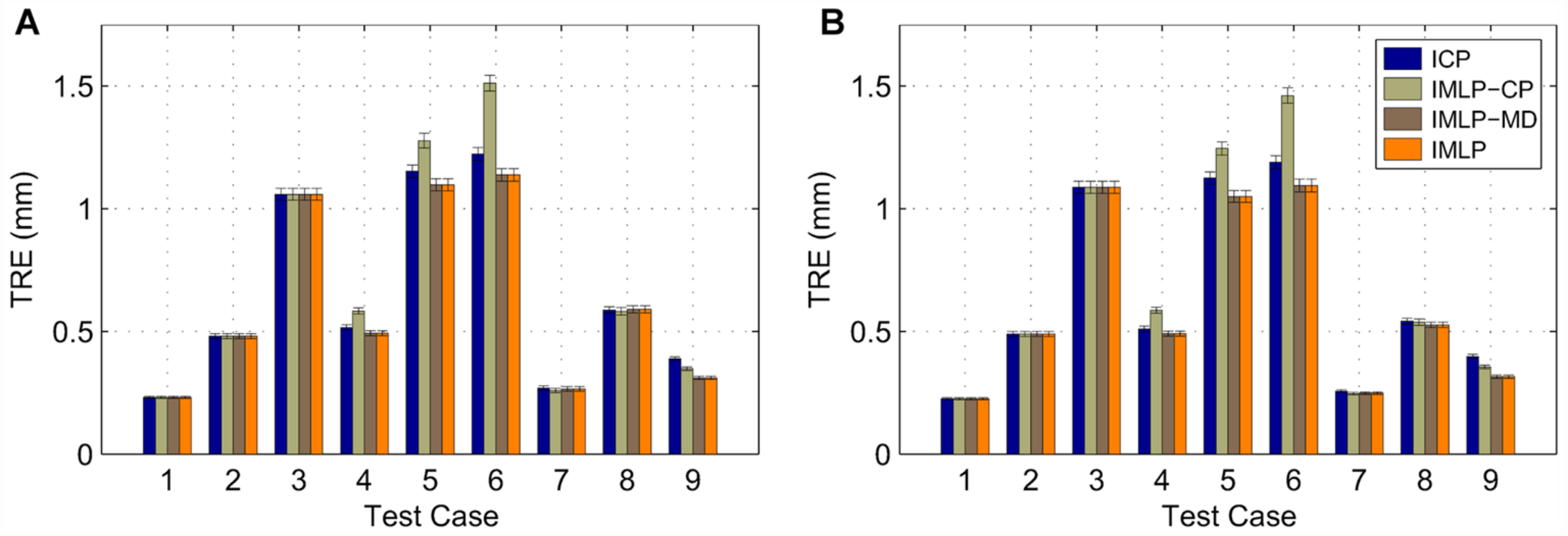
I’ll look at ones most relevant to our project

- Mesh Target
- Mesh Target with Outliers
- Sub Shape Registration

Mesh Target



Mesh Target



Mesh Target with Outliers

Moderate Misalignment

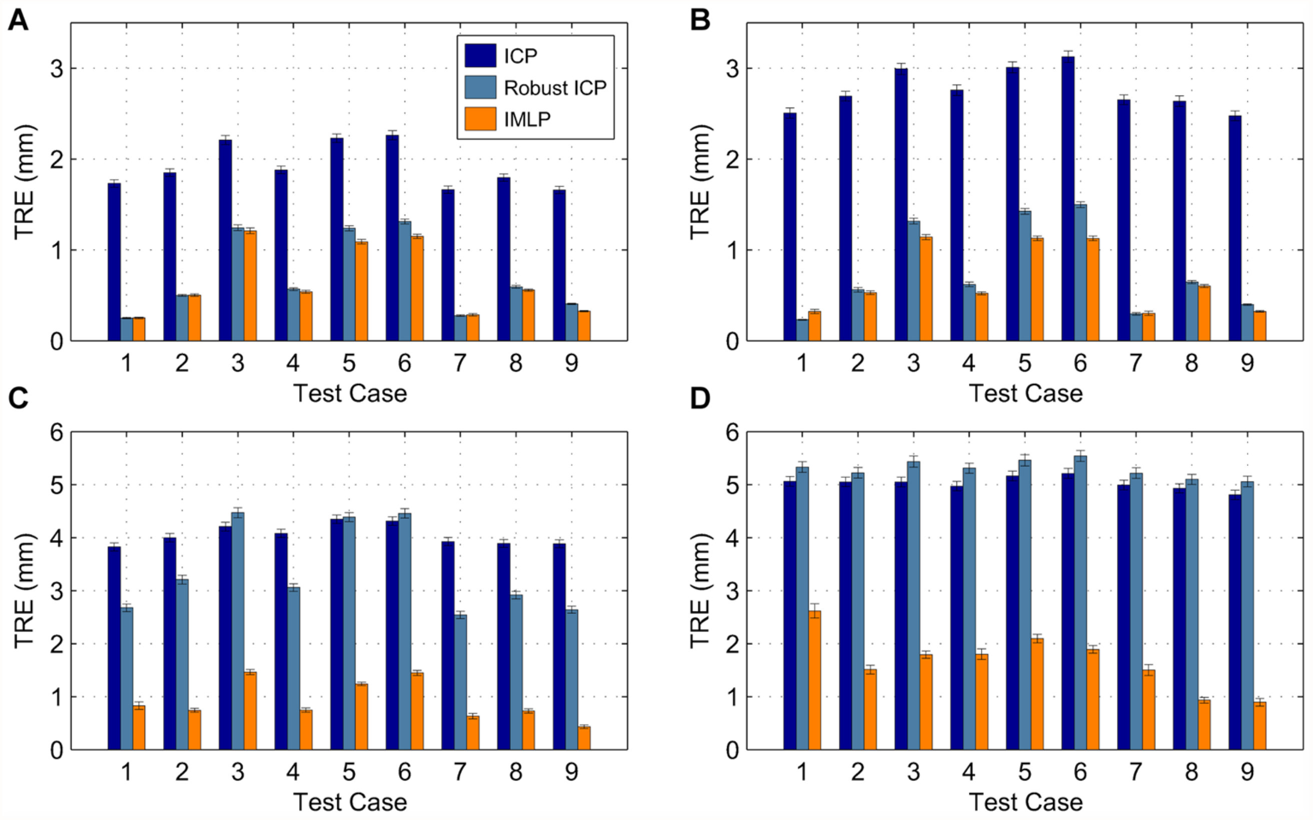
[15,30] mm/degı

A: 5%

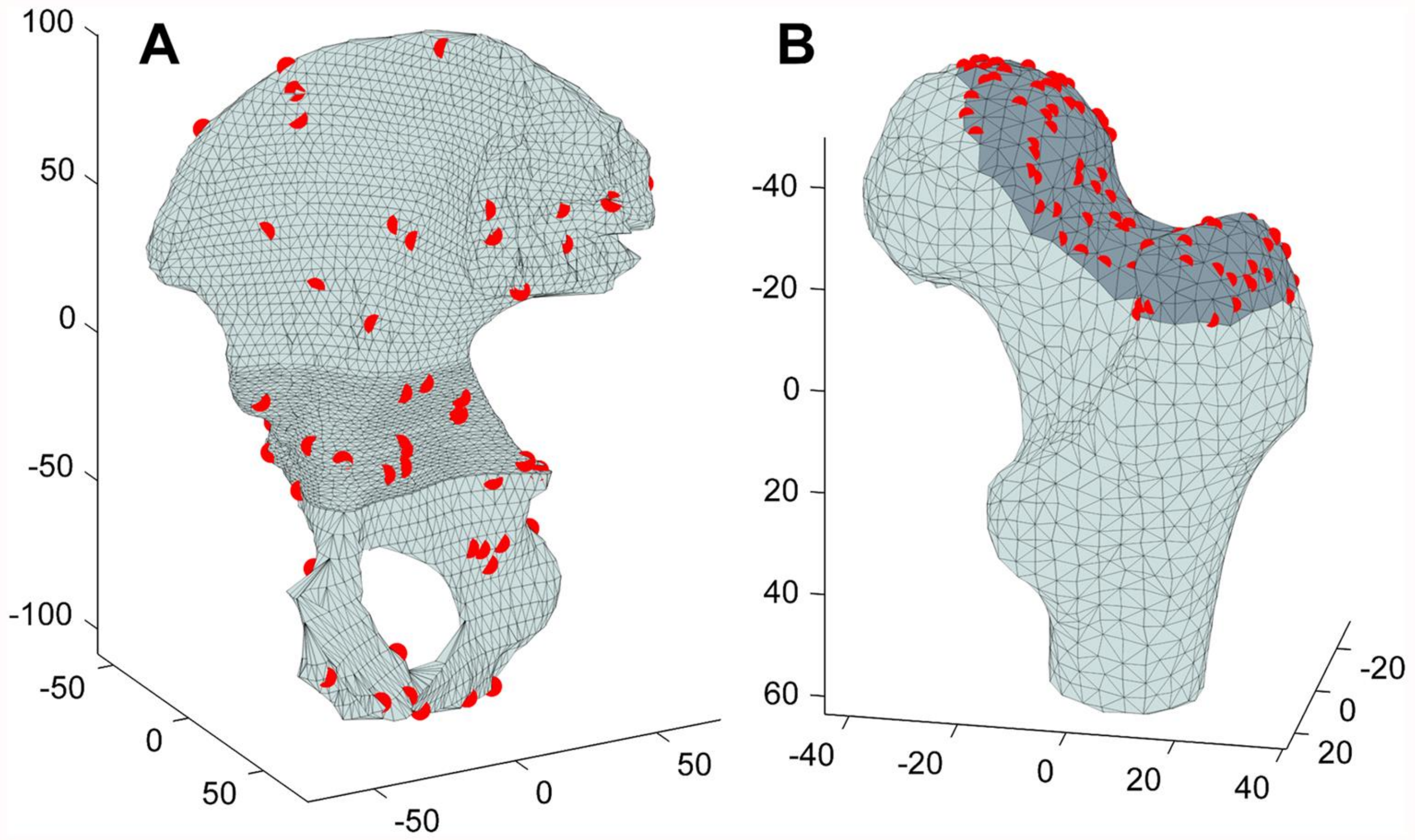
B: 10%

C: 20%

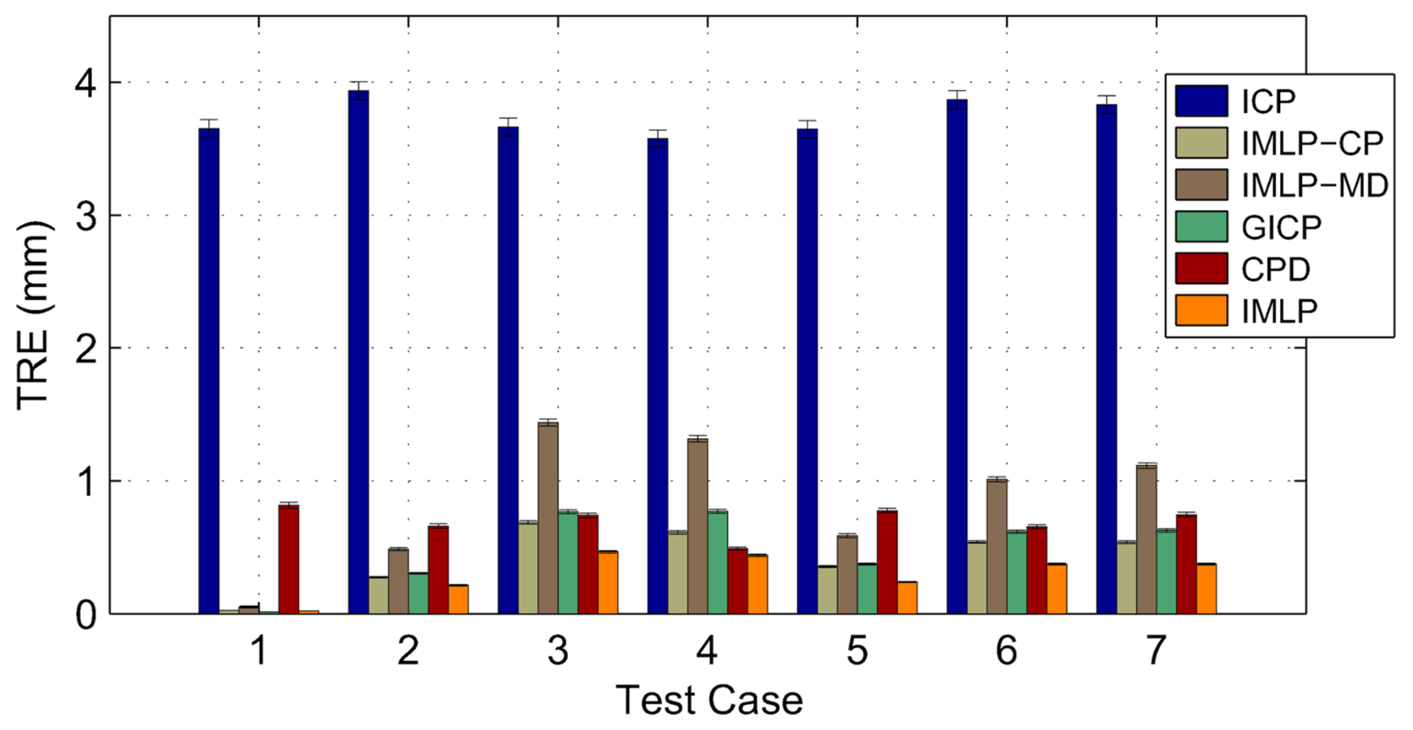
D: 30%



Sub Shape Registration



Sub Shape Registration



Different covariances (magnitude and direction)

Conclusion

Slower runtime

More accurate

Better with outliers

Higher failure rate

Assessment

Excellent algorithm

Did not discuss higher failure rate

Some hand wavy math