NSF Engineering Research Center
for Computer Integrated Surgical
Systems and Technology

LABORATORY FOR
Computational
Sensing + Robotics
THE JOHNS HOPKINS UNIVERSITY

Registration

600.445/645 Computer Integrated Surgery

100 YEARS
JOHNS HOPKINS ENGINEERING

WHITING
SCHOOL OF
ENGINEERING
THE JOHNS HOPKINS UNIVERSITY

Russell H. Taylor

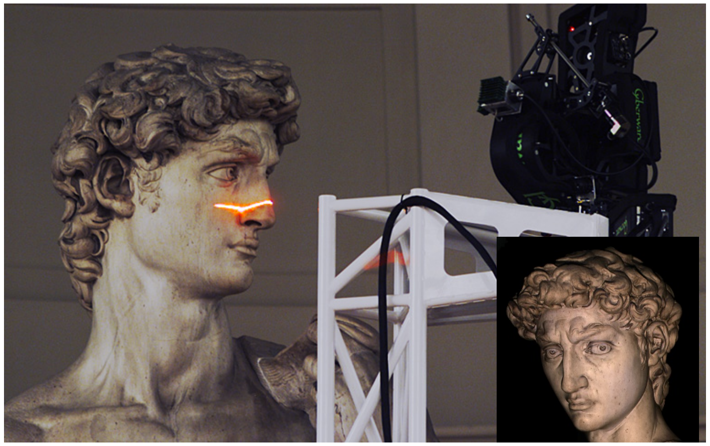
John C. Malone Professor of Computer Science,
with joint appointments in Mechanical Engineering, Radiology & Surgery

Director, Laboratory for Computational Sensing and Robotics

The Johns Hopkins University

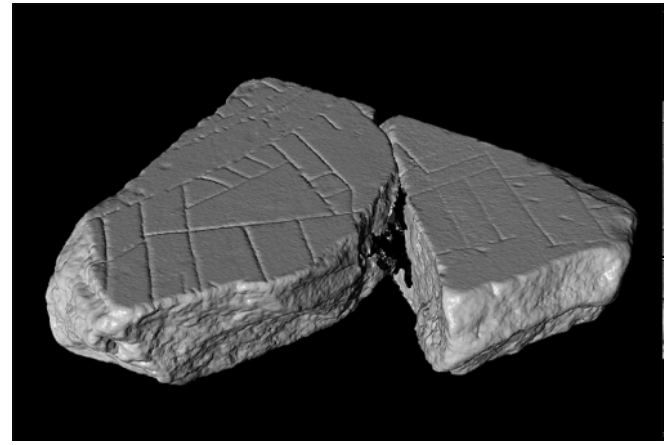
rht@jhu.edu

Why is Registration Important?



Digitize important cultural artifacts

Medical interventions

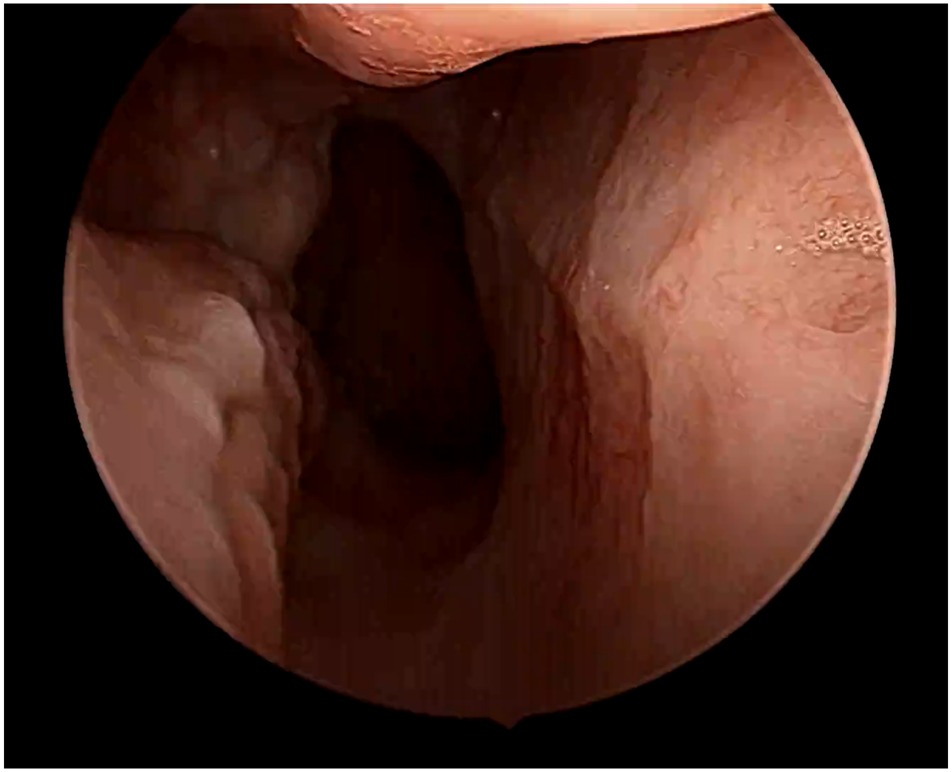


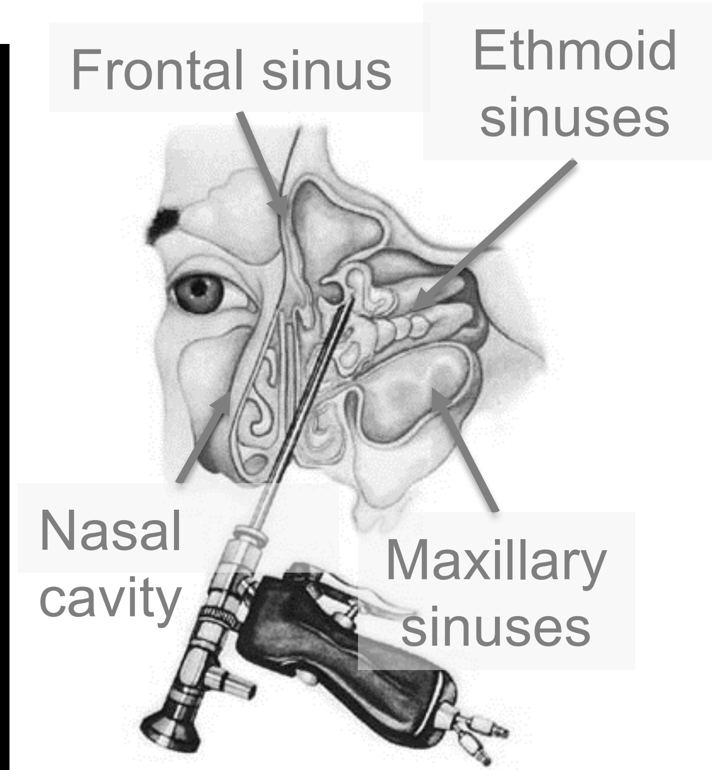
Archeology

And many more applications…

Slide Credit: Ayushi Sinha

Why is Registration Important?

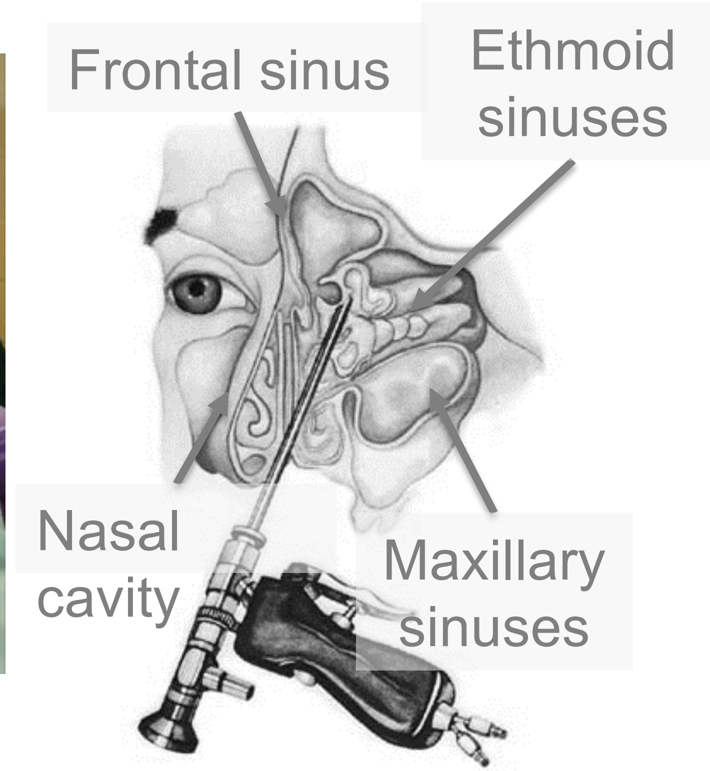




Typical Example: Sinus Endoscopy. The surgeon can only see video from the endoscope. But crucial data is in the CT about structures that cannot be seen.

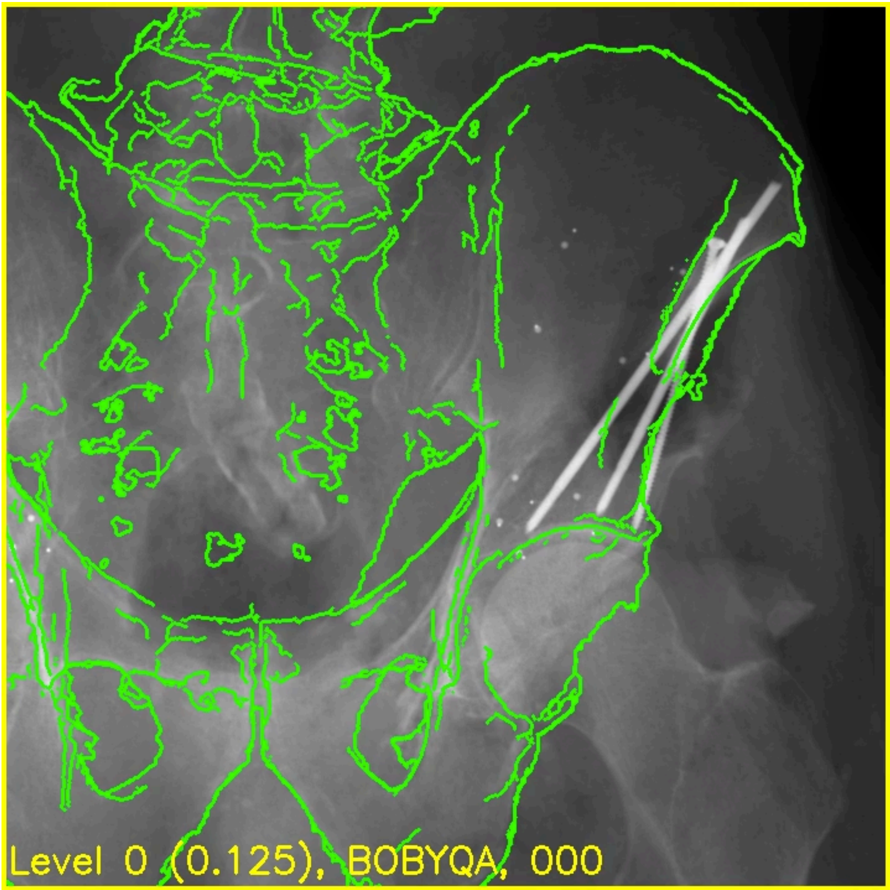
Slide Credit: Ayushi Sinha

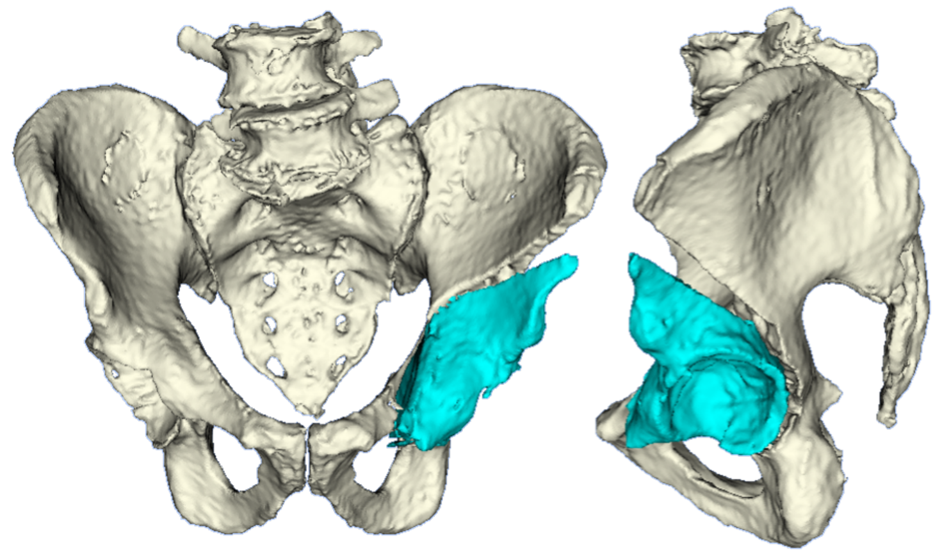
Why is Registration Important?



Typical Example: Sinus Endoscopy. After registration, the computed can create video overlays, help guide a robot, or provide other assistance.

Why is Registration Important?



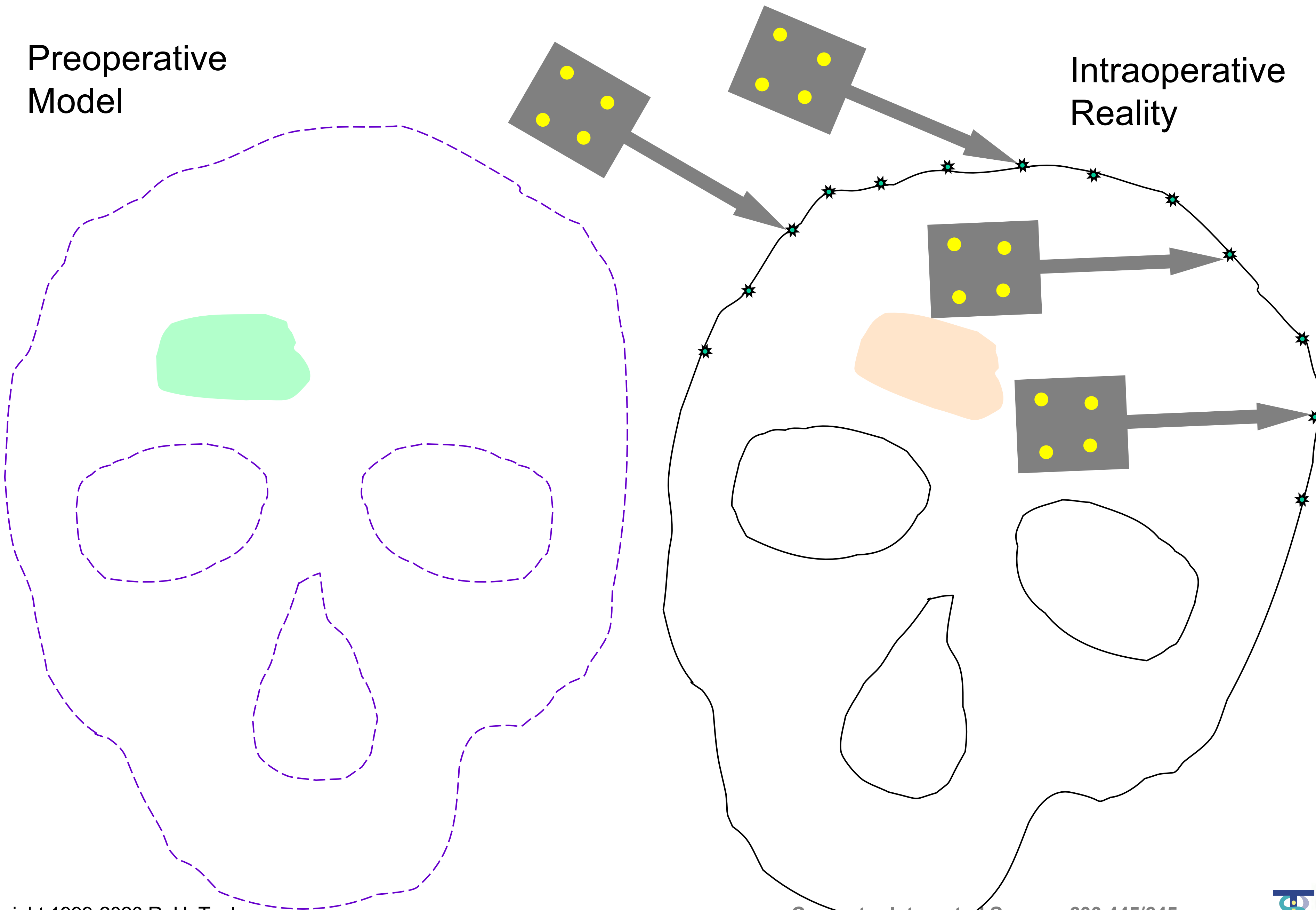


Typical Example: Osteotomies. Surgeon needs to know the position and orientation of bone fragment relative to pelvis, based on x-ray images.

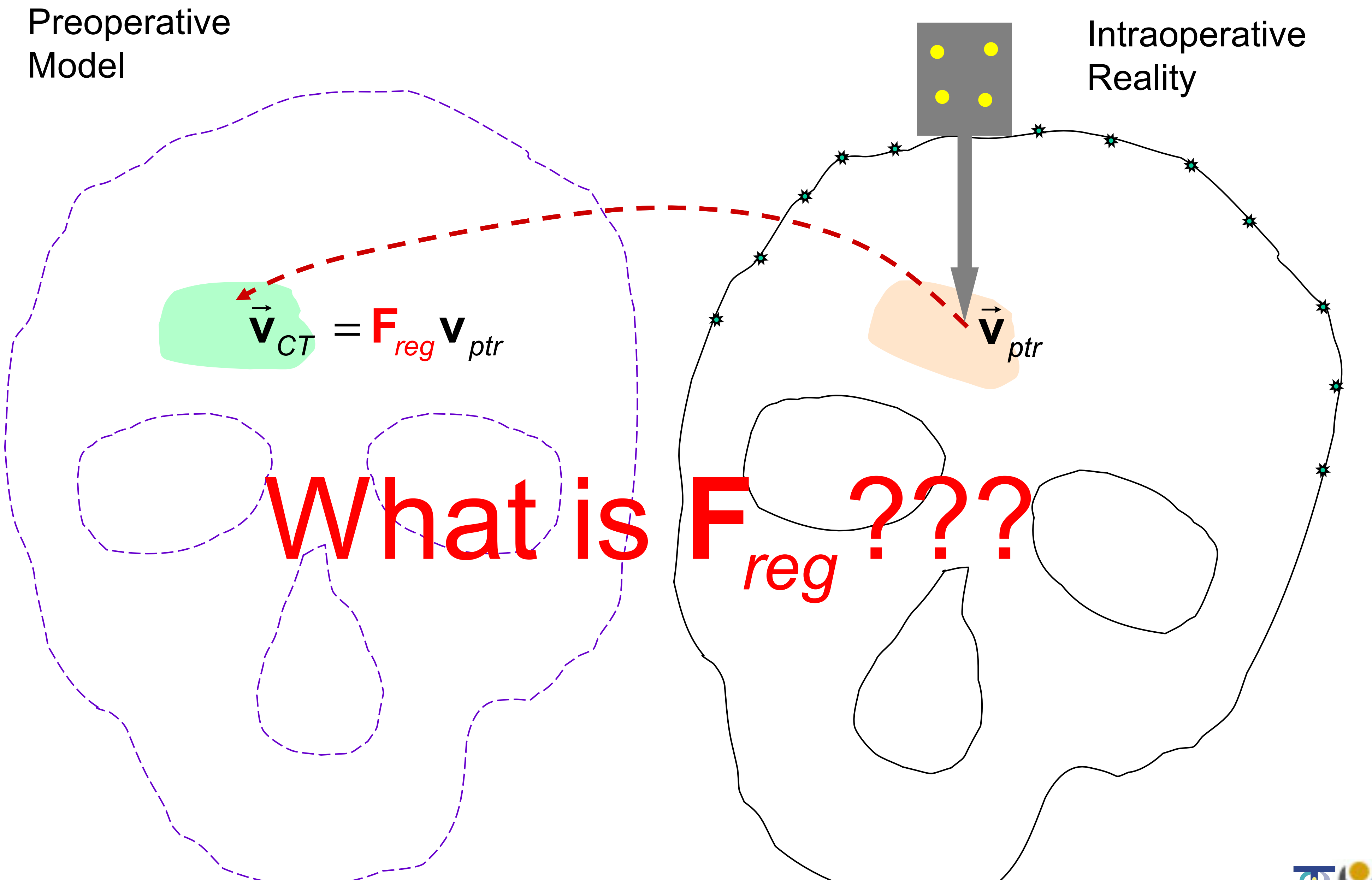
What needs registering?

- **Preoperative Data**
  - 2D & 3D medical images
  - Models
  - Preoperative positions
- **Intraoperative Data**
  - 2D & 3D medical images
  - Models
  - Intraoperative positioning information
- **The Patient**

A typical registration problem



A typical registration problem



Taxonomy of methods

- Feature-based
- Intensity-based

Framework for feature-based methods

- Definition of coordinate system relations
- Segmentation of reference features
- Definition of disparity function between features
- Optimization of disparity function

Definitions

Overall Goal: Given two coordinate systems,

$$\mathbf{Ref}_{\mathbf{A}} \ \& \ \mathbf{Ref}_{\mathbf{B}}$$

and coordinates

$$\mathbf{x}_{\mathbf{A}} \ \& \ \mathbf{x}_{\mathbf{B}}$$

associated with corresponding features in the two coordinate systems, the general goal is to determine a transformation function T that transforms one set of coordinates into the other:

$$\mathbf{x}_{\mathbf{A}} = \mathbf{T}(\mathbf{x}_{\mathbf{B}})$$

Definitions

- **Rigid Transformation:** Essentially, our old friends 2D & 3D coordinate transformations:

  $$\mathbf{T(x) = R \bullet x + p}$$

  The key assumption is that deformations may be neglected.
- **Similarity Transformation:** Essentially, rigid+scale change. Preserves angles and shape, but not size

  $$\mathbf{T(x) =} s\mathbf{R \bullet x + p}$$
- **Elastic Transformation:** Cases where must take more general deformations into account. Many different flavors, depending on what is being deformed

Uses of Rigid Transformations

- Register (approximately) multiple image data sets
- Transfer coordinates from preoperative data to reality (especially in orthopaedics & neurosurgery)
- Initialize non-rigid transformations

Uses of Elastic Transformations

- Register different patients to common data base (e.g., for statistical analysis)
- Overlay atlas information onto patient data
- Study time-varying deformations
- Assist segmentation

Typical Features

- Point fiducials
- Point anatomical landmarks
- Ridge curves
- Contours
- Surfaces
- Line fiducials

Distance Functions

Given two (possibly distributed) features *Fi* and *Fj*, need to define a distance metric distance (Fi, Fj) between them. Some choices include:

- Minimum distance between points
- Maximum of minimum distances
- Area between line features
- Volume between surface features
- Area between point and line
- etc.

Distance Functions Between Feature Sets

Let $\mathcal{F}_A = \{\ldots F_{Ai} \ldots\}$ and $\mathcal{F}_B = \{\ldots F_{Bi} \ldots\}$ be corresponding sets of features in $\mathbf{Ref}_A$and $\mathbf{Ref}_B$, respectively. We need to define an appropriate disparity function $D(\mathcal{F}_A, \mathcal{F}_B)$ between feature sets. Some typical choices include:

$$D = \sum_i w_i [distance(F_{Ai}, \mathbf{T}(F_{Bi}))]^2$$

$$D = \max_i distance(F_{Ai}, \mathbf{T}(F_{Bi}))$$

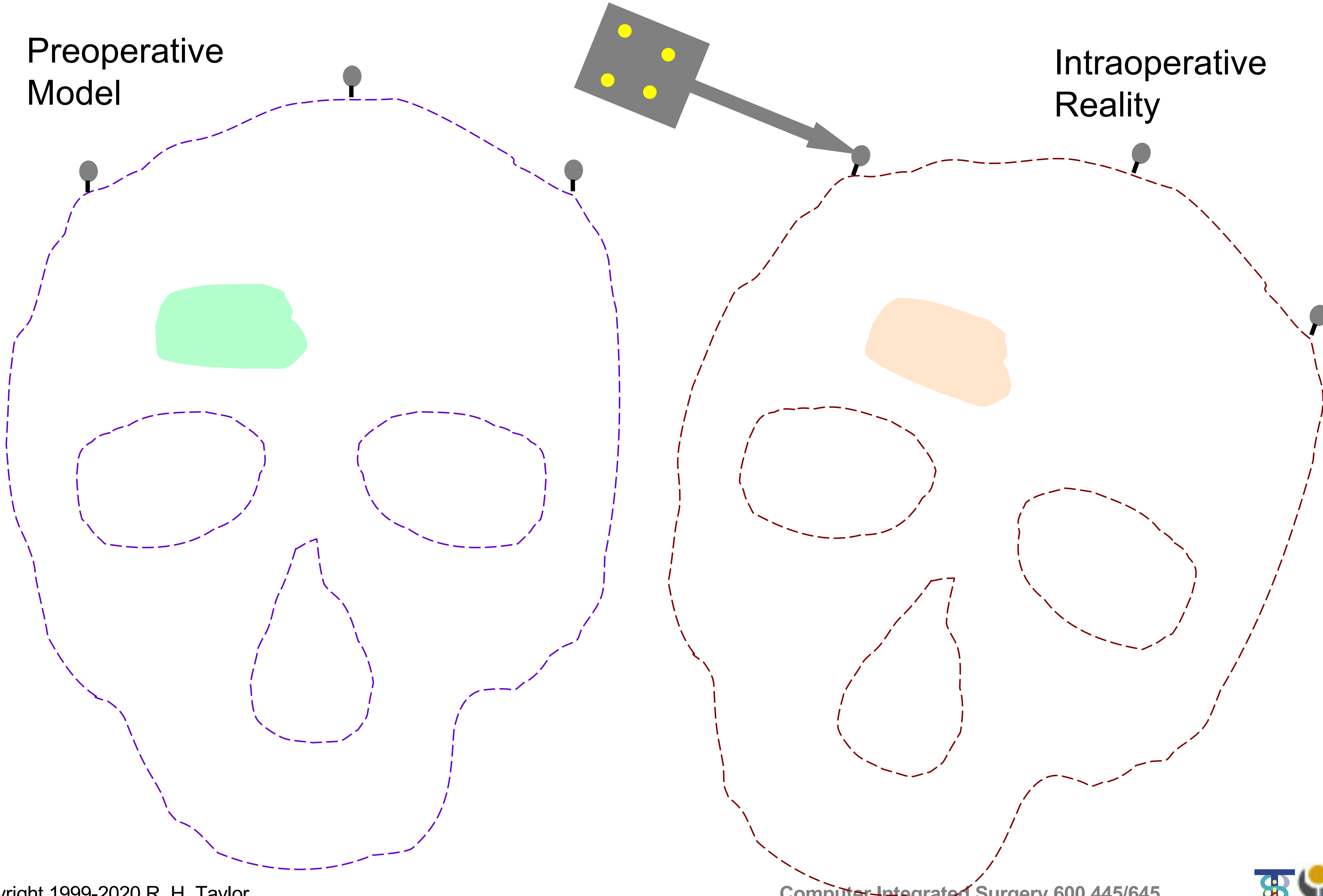
$$D = \underset{i}{\text{median}}\, distance(F_{Ai}, \mathbf{T}(F_{Bi}))$$

$$D = Cardinality\{i | distance(F_{Ai}, \mathbf{T}(F_{Bi})) > threshold\}$$

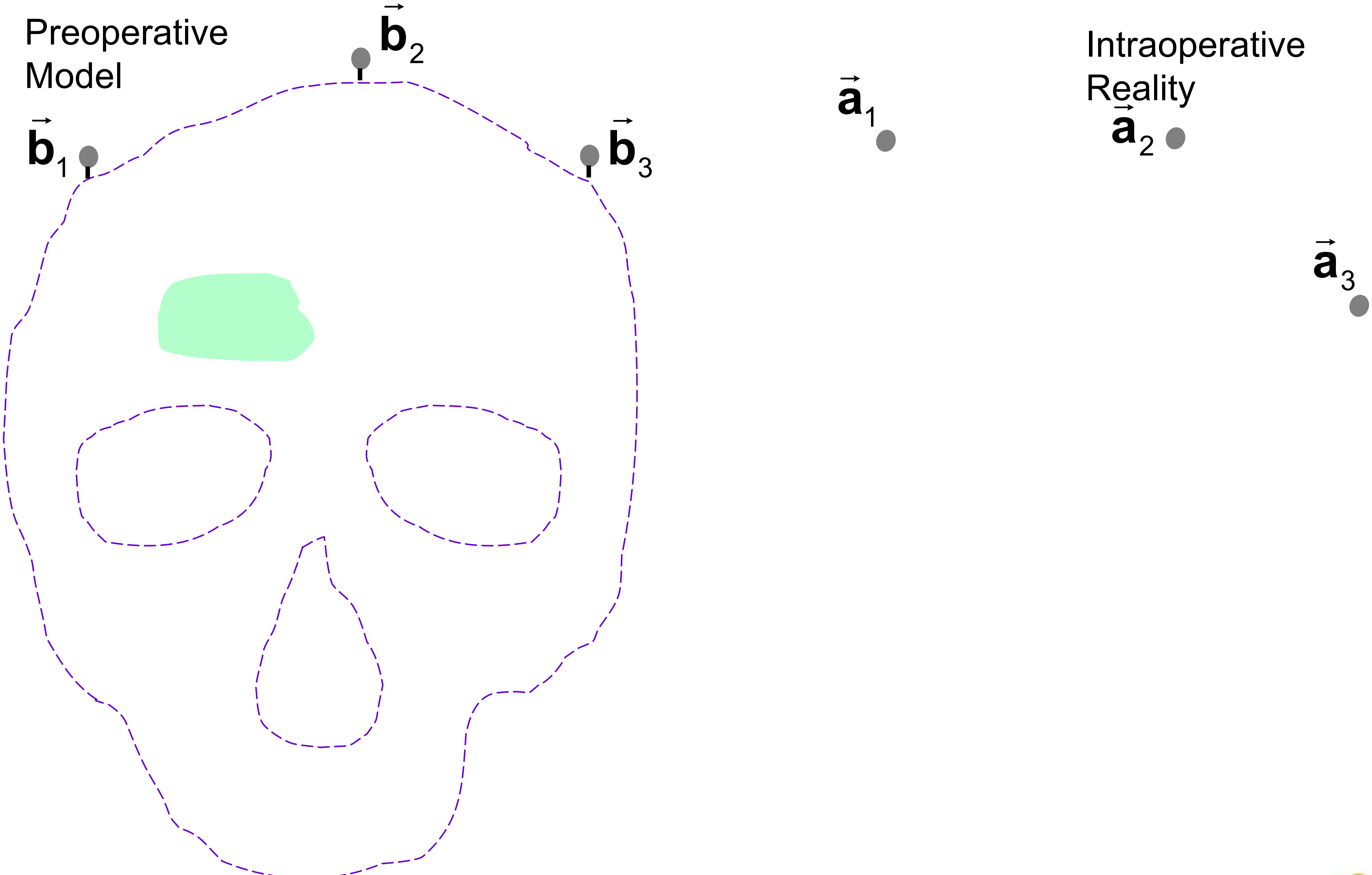
Optimization

- Global *vs* Local
- Numerical *vs* Direct Solution
- Local Minima

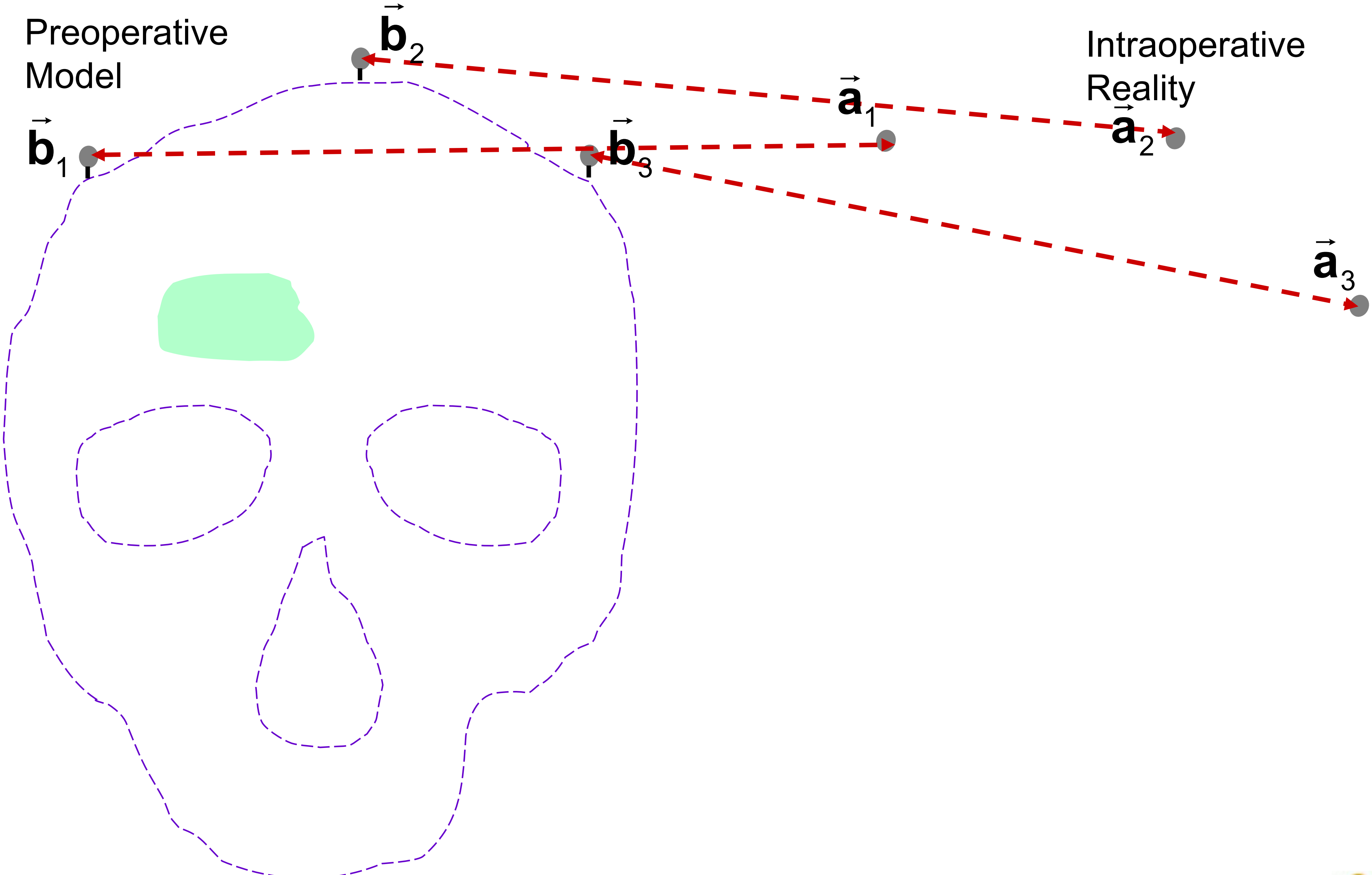
A typical fiducial-based registration problem



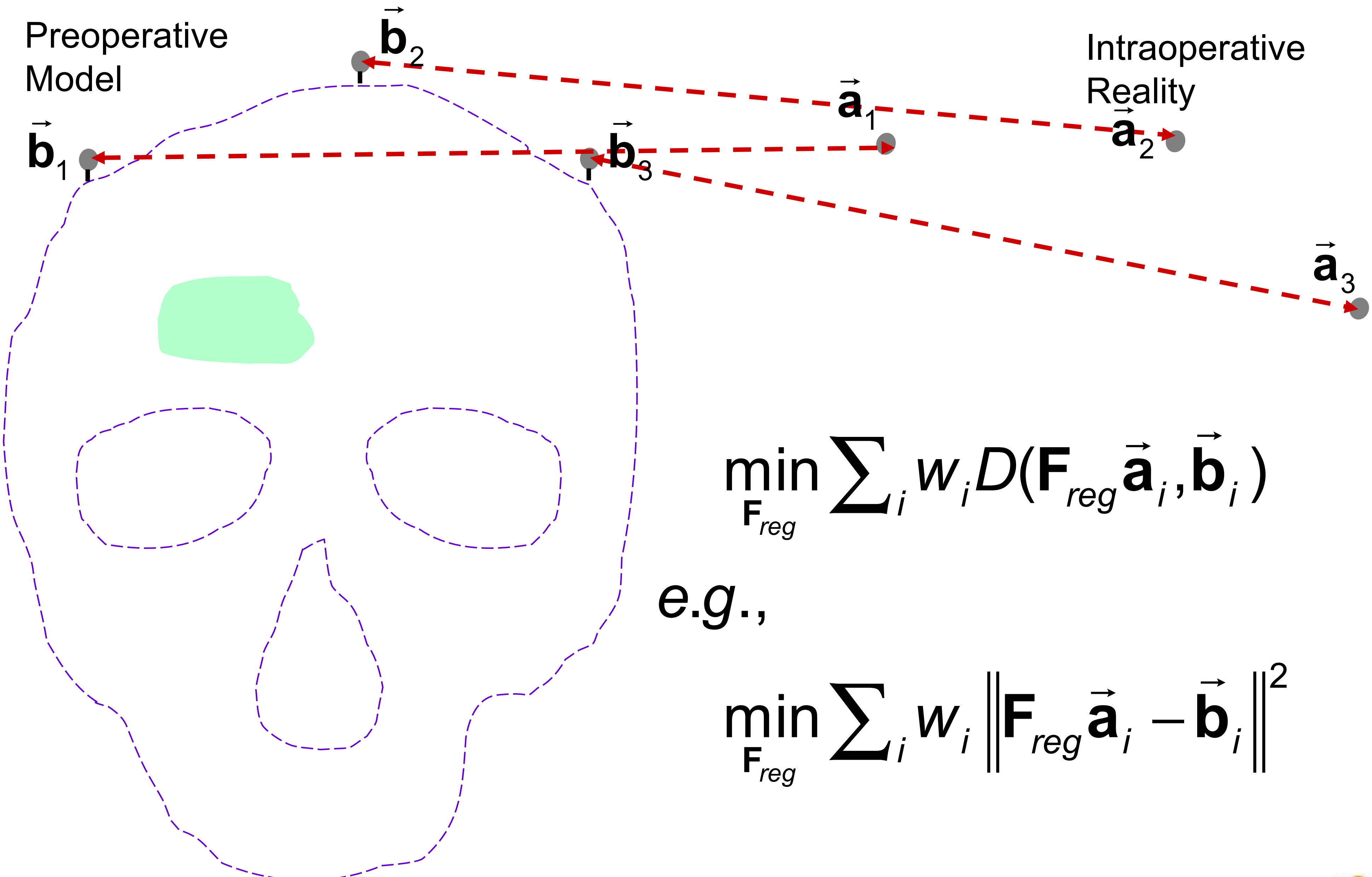
What the computer knows



Identify corresponding points



Find best rigid transformation!

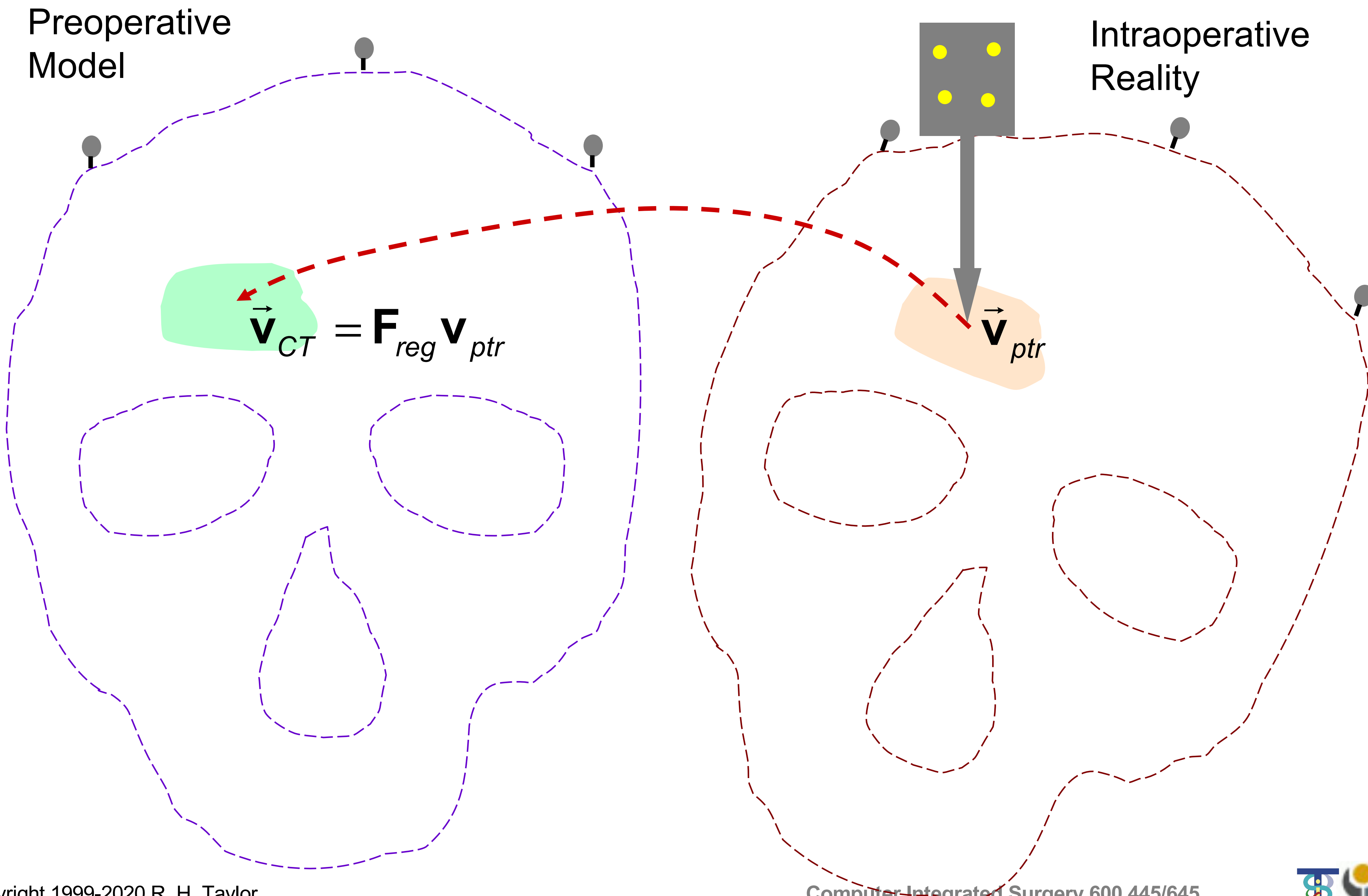


$$\min_{\mathbf{F}_{reg}} \sum_i w_i D(\mathbf{F}_{reg}\vec{\mathbf{a}}_i, \vec{\mathbf{b}}_i)$$

e.g.,

$$\min_{\mathbf{F}_{reg}} \sum_i w_i \left\| \mathbf{F}_{reg}\vec{\mathbf{a}}_i - \vec{\mathbf{b}}_i \right\|^2$$

Navigate


Preoperative
Model
Intraoperative
Reality
$\vec{v}_{CT} = \mathbf{F}_{reg}\mathbf{v}_{ptr}$
$\vec{v}_{ptr}$

Sampled 3D data to surface models

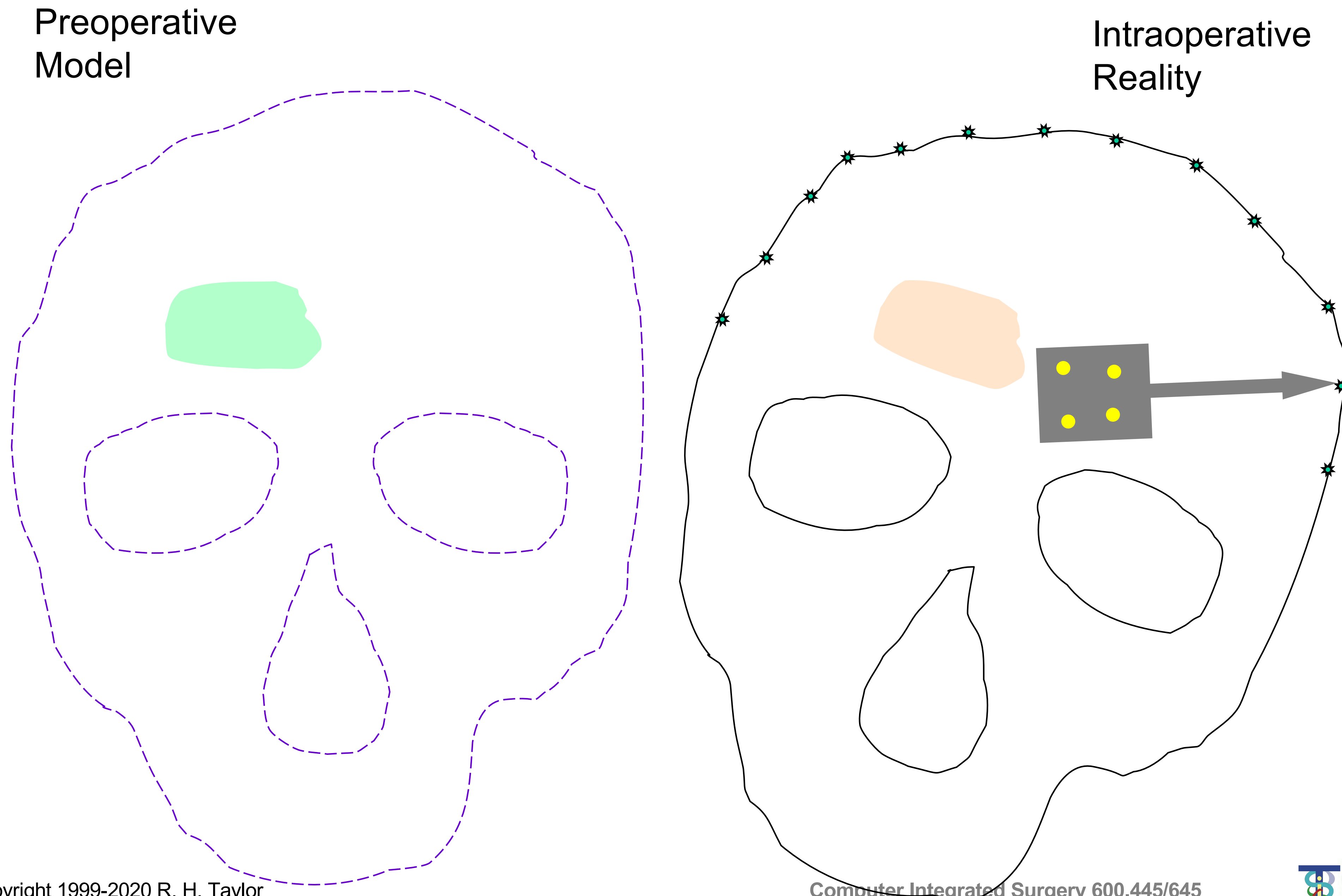
Outline:

- Select large number of sample points
- Determine distance function $d_S(\mathbf{f}, \mathcal{F})$ for a point $\mathbf{f}$ to a surface feature $\mathcal{F}$.
- Use d_S to develop disparity function D.

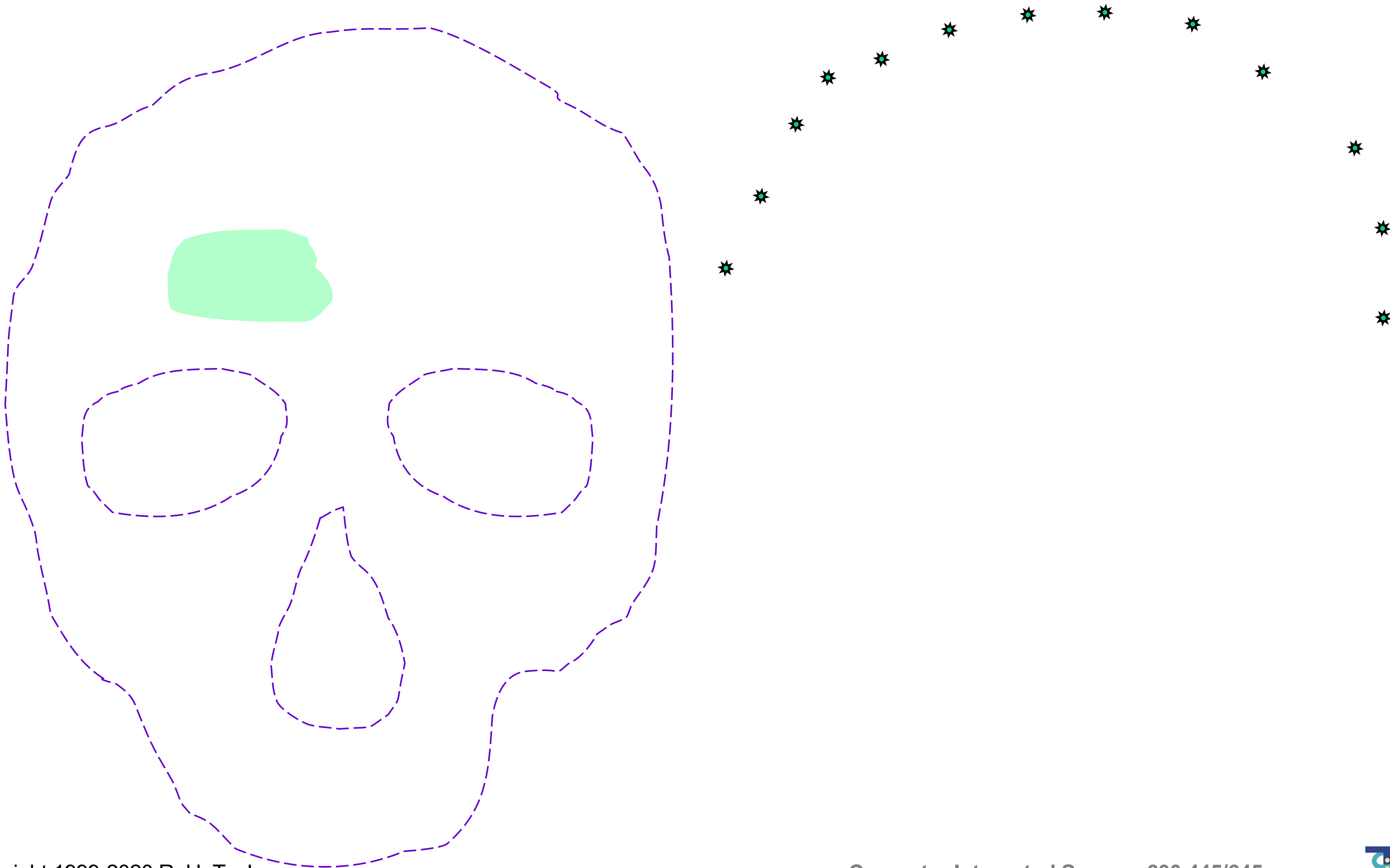
Examples

- Head-in-hat algorithm [Levin et al., 1988; Pelizzari et al., 1989]
- Distance maps [e.g., Lavallee et al]
- Iterative closest point [Besl and McKay, 1992]

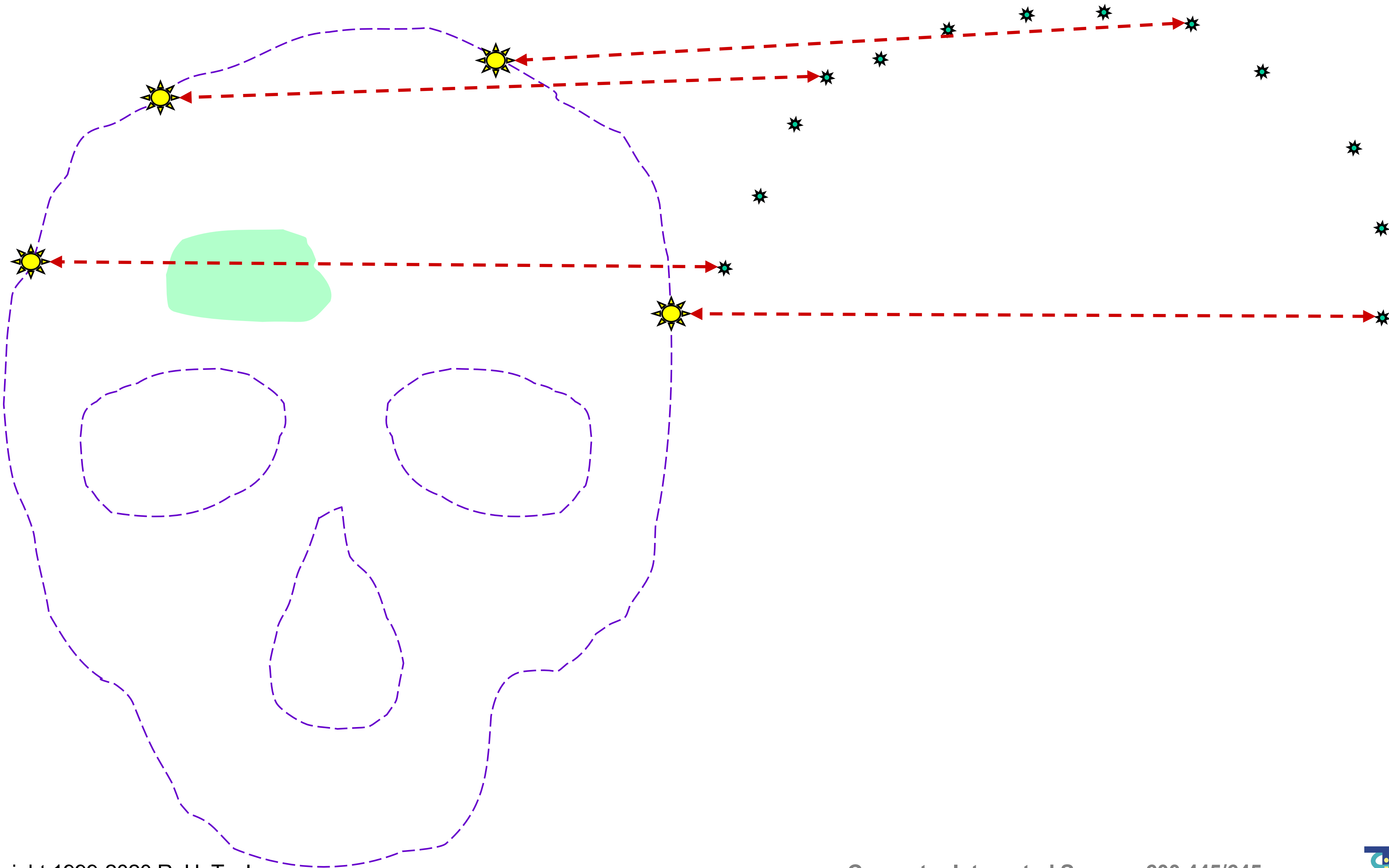
A typical surface registration problem



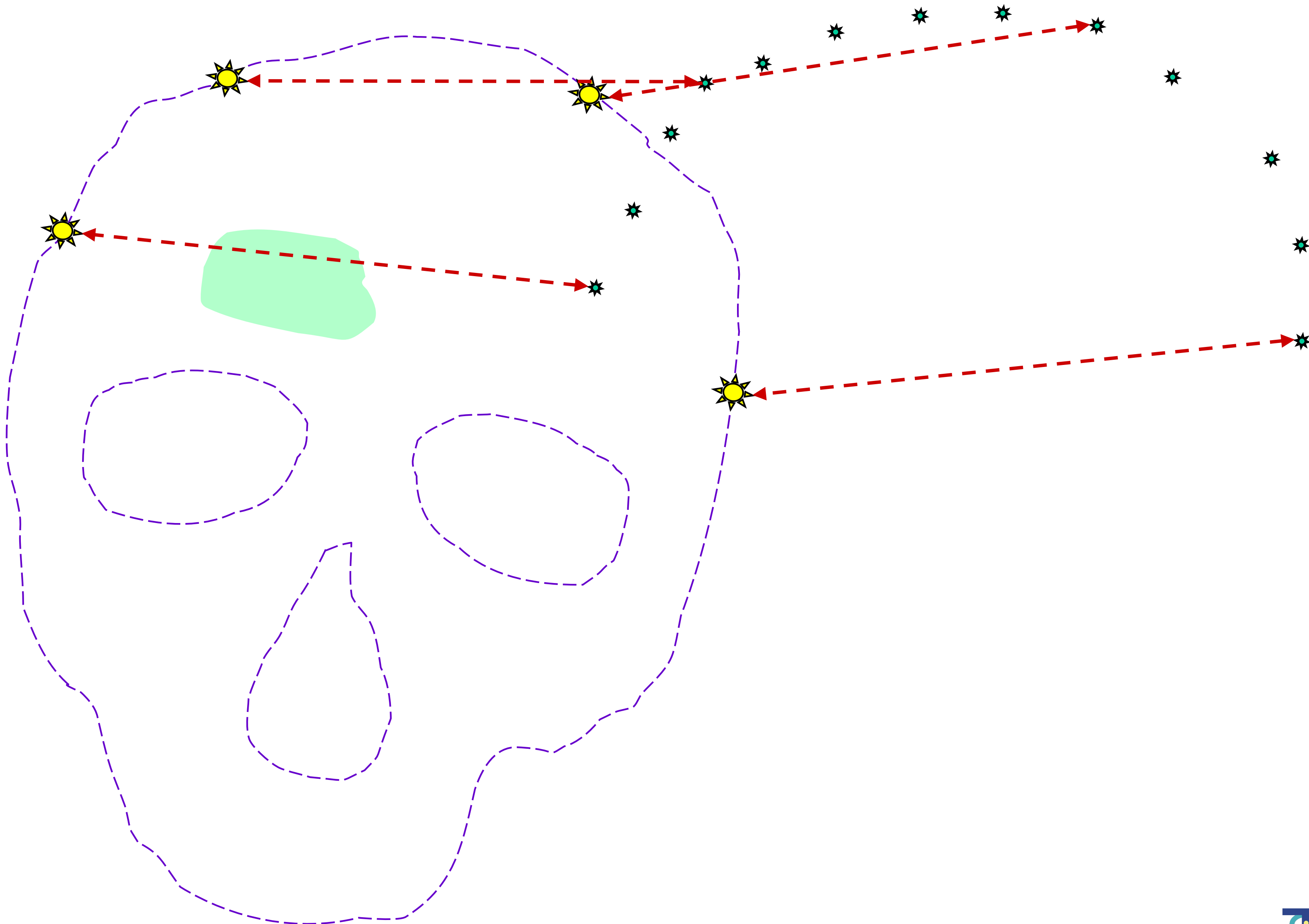
What the computer knows



Find corresponding points & pull!



Find corresponding points & pull!

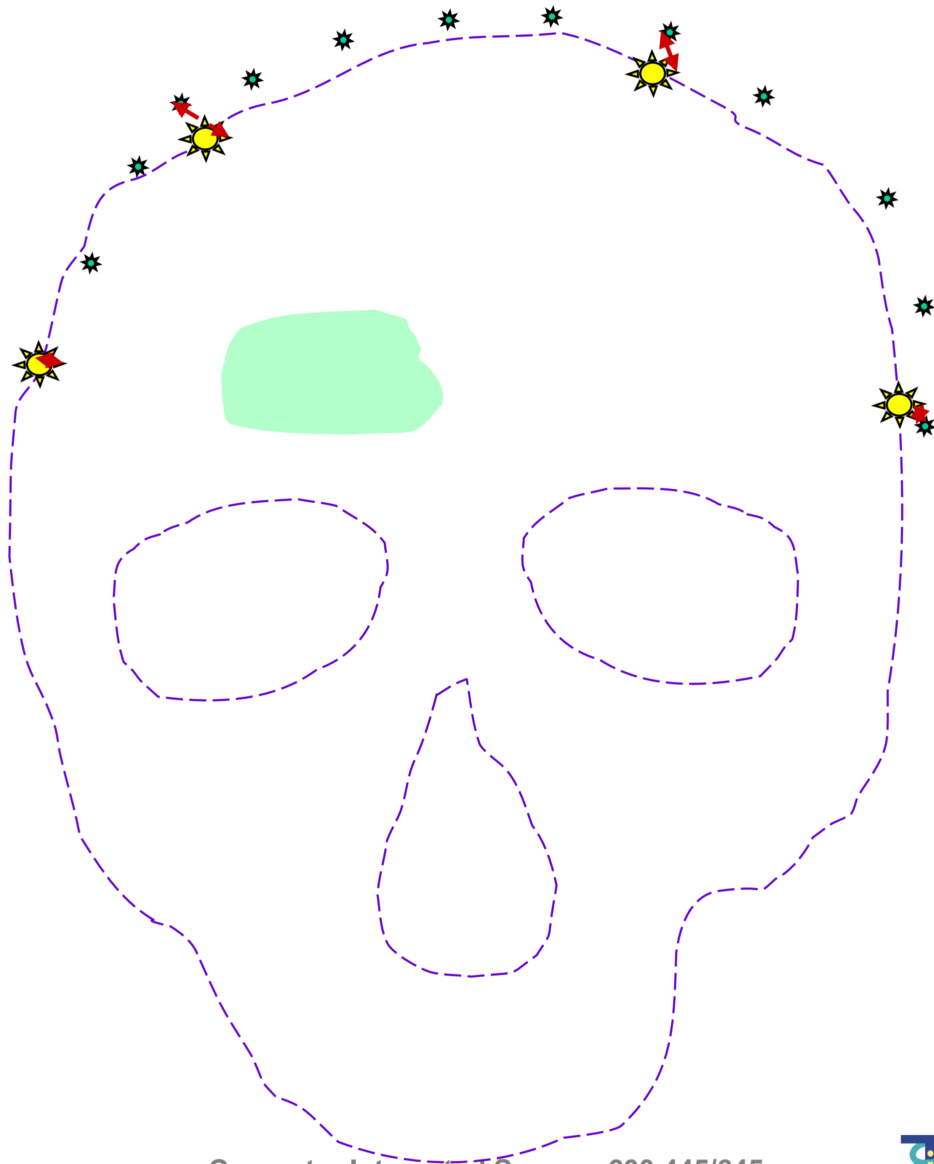


Find corresponding points & pull!

Iterate this until converge

Find new point pairs every iteration

Key challenge is finding point pairs efficiently.



Head in Hat Algorithm

- Levin et al, 1988; Pelizzari et al, 1989
- Origially used for Pet-to-MRI/CT registration
- Given $\mathbf{f}_i \in \mathcal{F}_A$, and a surface model $\mathcal{F}_B$, computes a rigid transformation $\mathbf{T}$ that minimizes

$$D = \sum_i [d_S(\mathcal{F}_B, \mathbf{T} \cdot \mathbf{f}_i)]^2$$

  where d_S is defined below, given a good initial guess for $\mathbf{T}$.

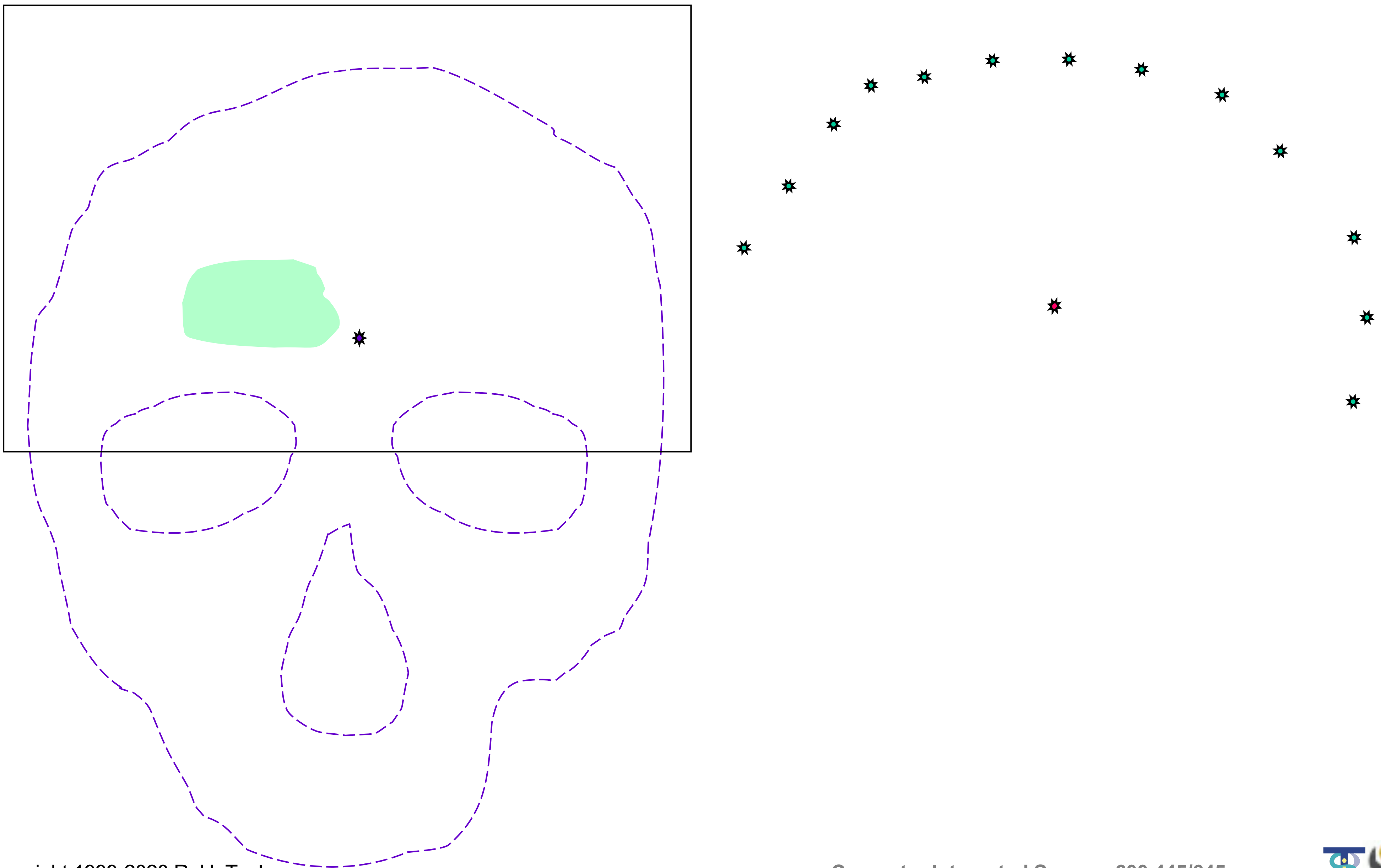
- Optimization uses standard numerical method (steepest gradient descent [Powell]) to find six parameters (3 rotations, 3 translations) defining $\mathbf{T}$.

Head in Hat Algorithm

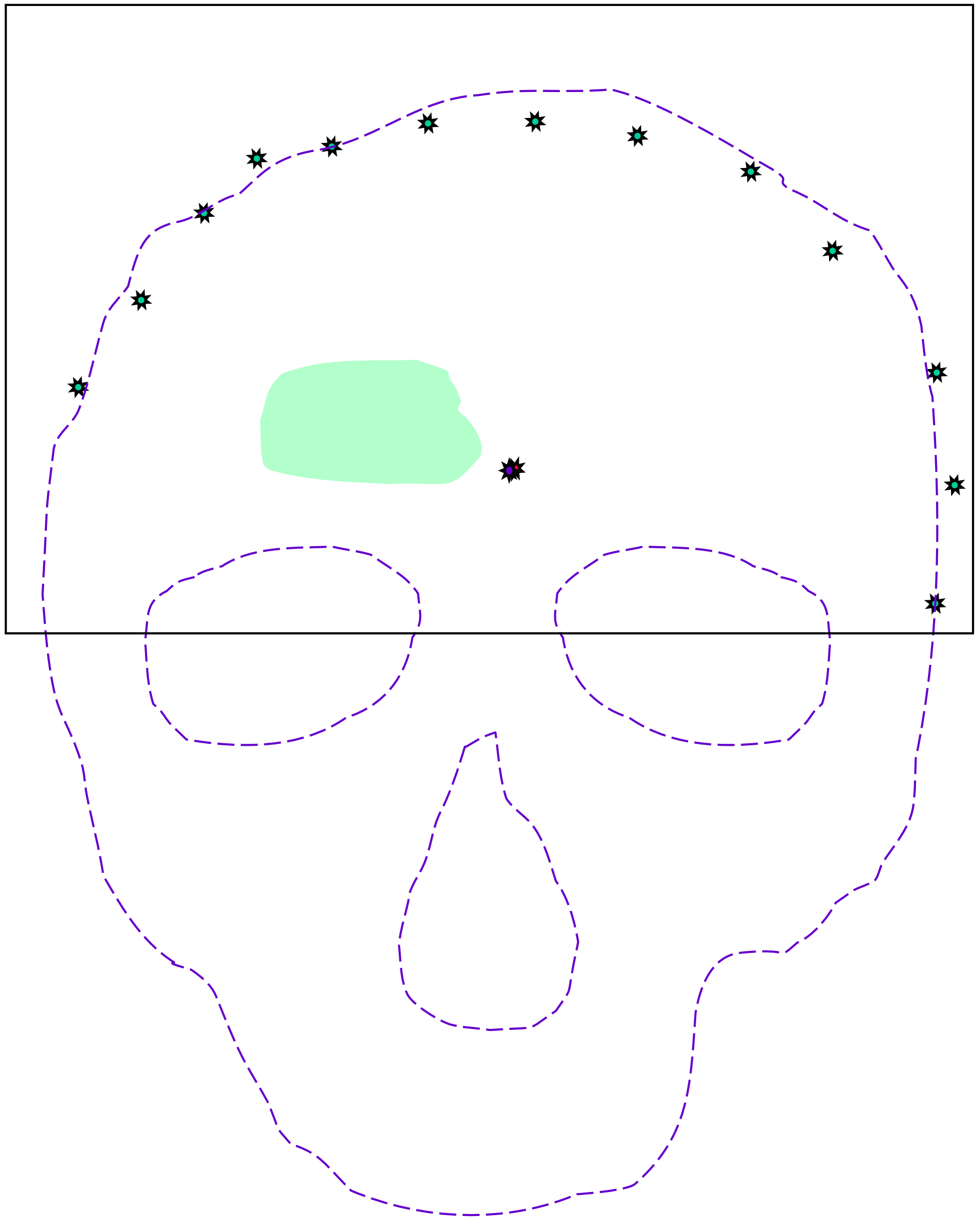
Definition of $d_S(\mathcal{F}_B, \mathbf{f}_i)$

1. Compute centroid $\mathbf{g}_B$ of surface $\mathcal{F}_B$.
2. Determine a point $\mathbf{q}_i$ that lies on the intersection of the line $\mathbf{g}_B - \mathbf{f}_i$ and $\mathcal{F}_B$.
3. Then, $d_S(\mathcal{F}_B, \mathbf{f}_i) = ||\mathbf{q}_i - \mathbf{f}_i||$

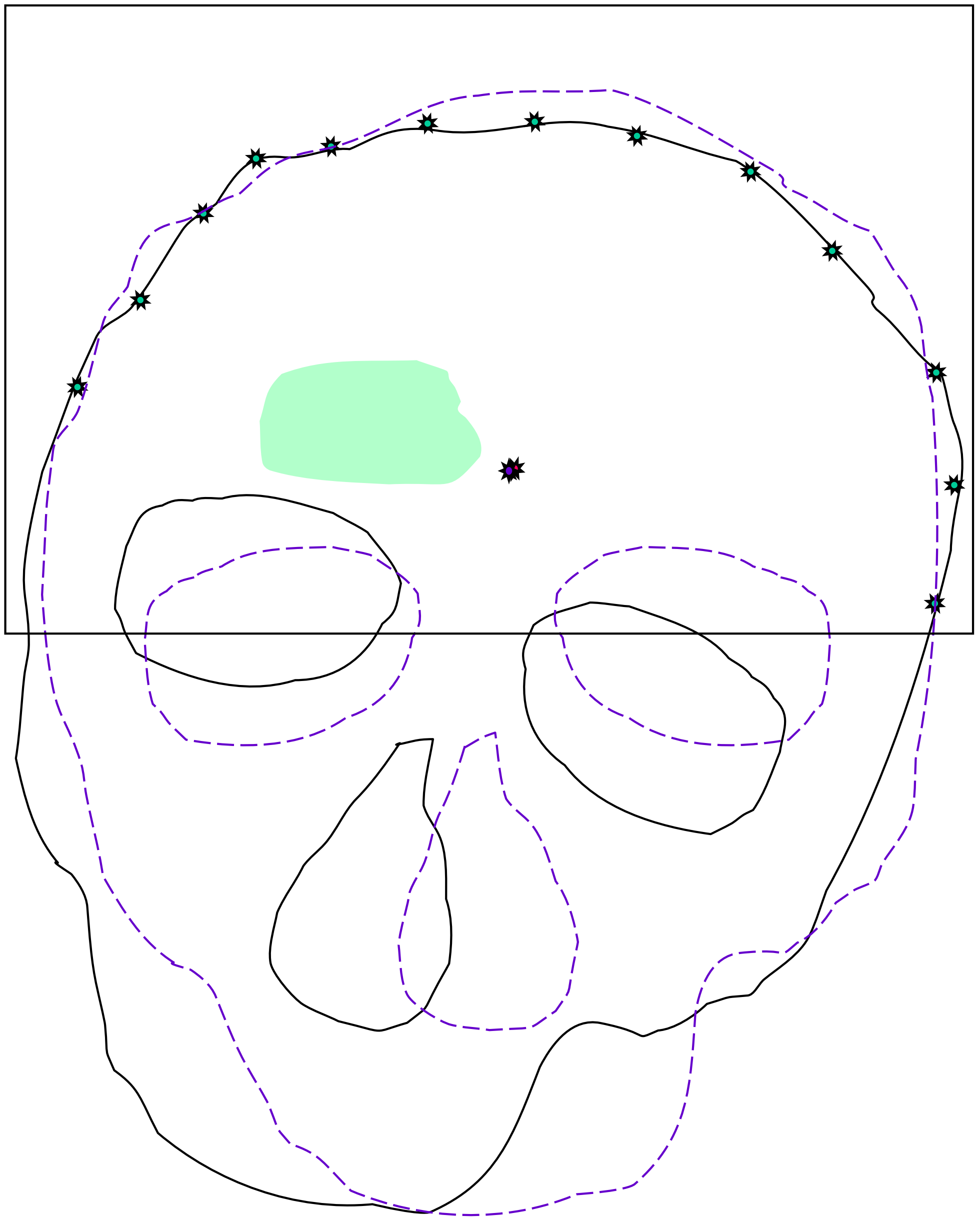
Head-in-hat algorithm: step 0



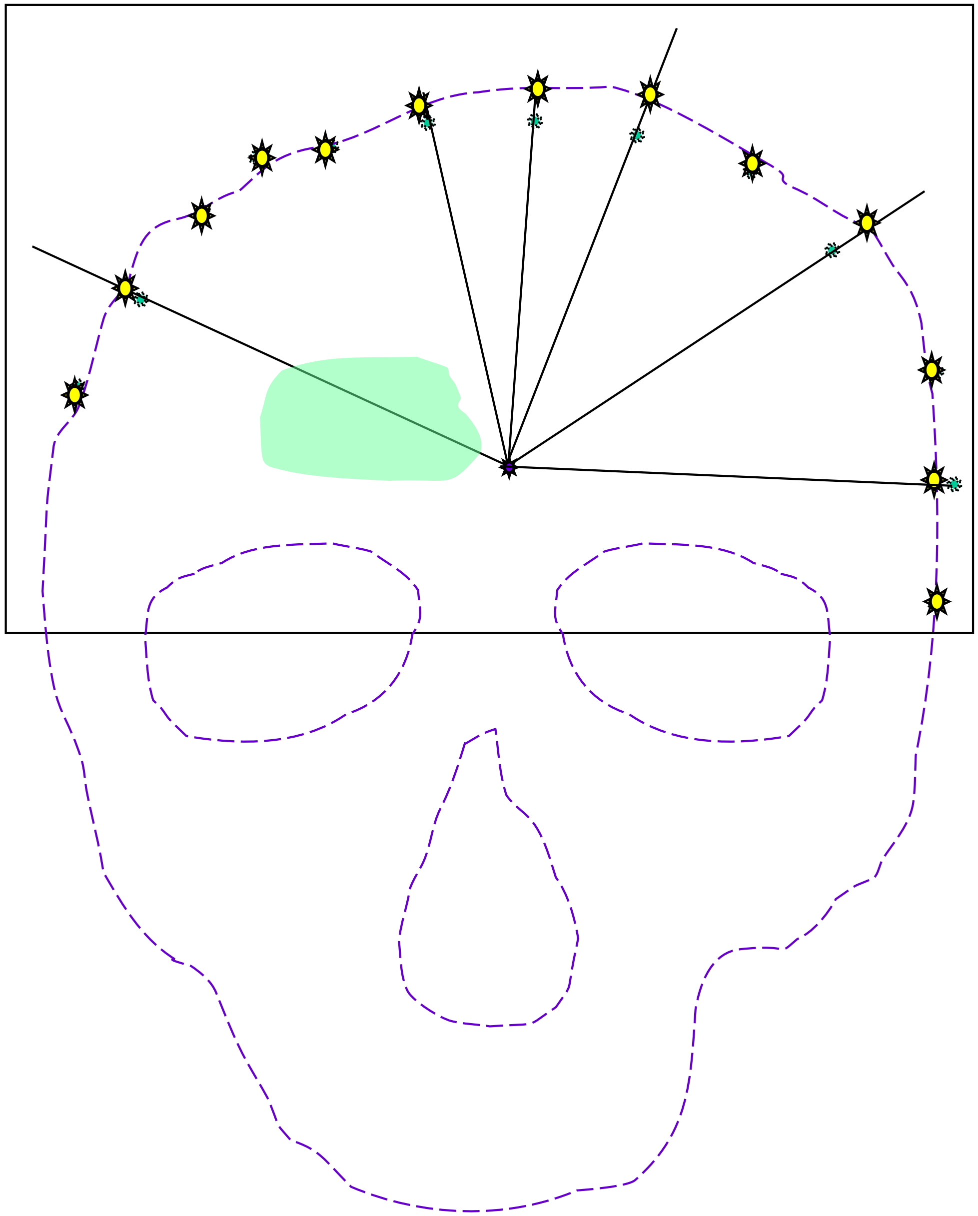
Head-in-hat algorithm: step1



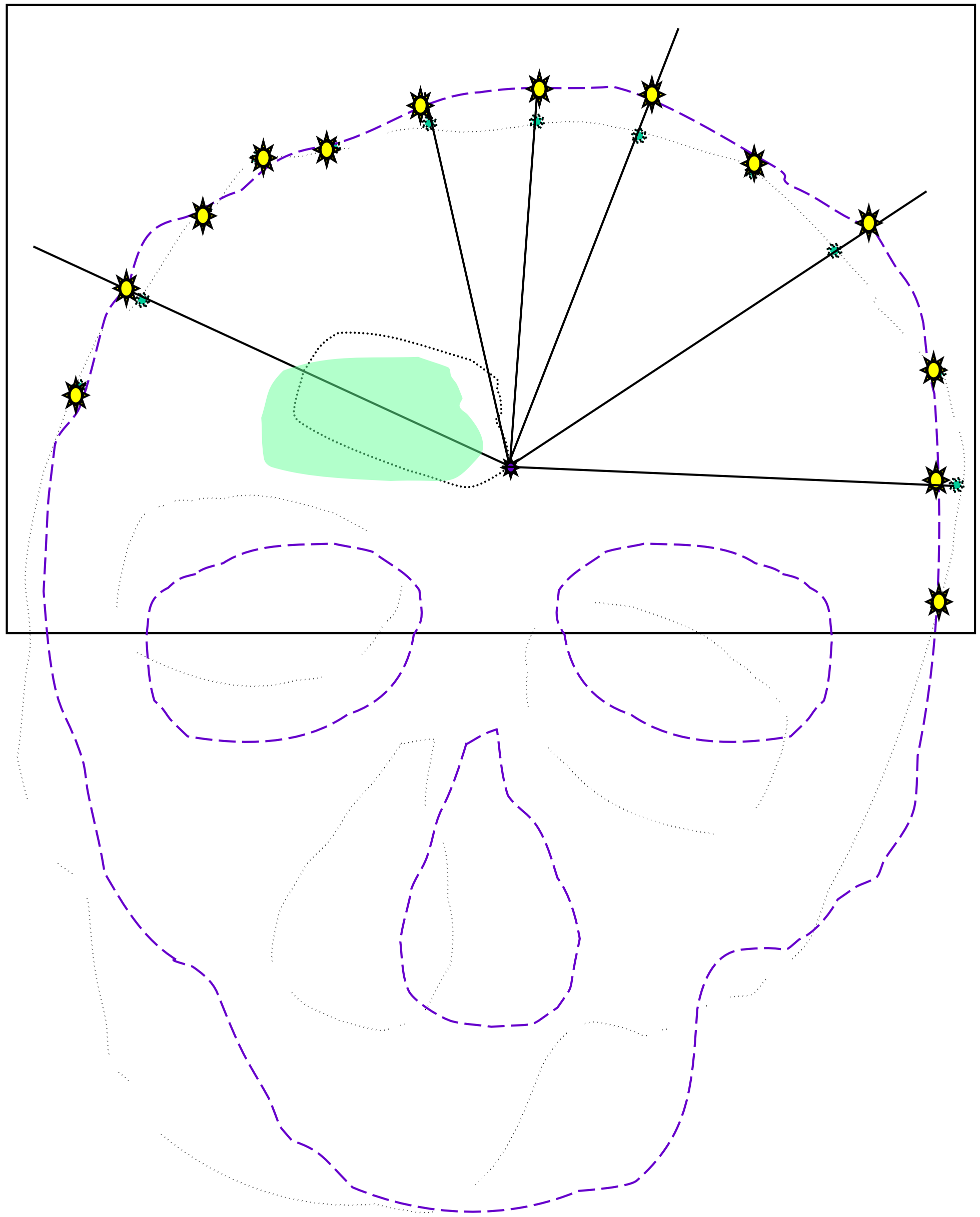
Head-in-hat algorithm: step1



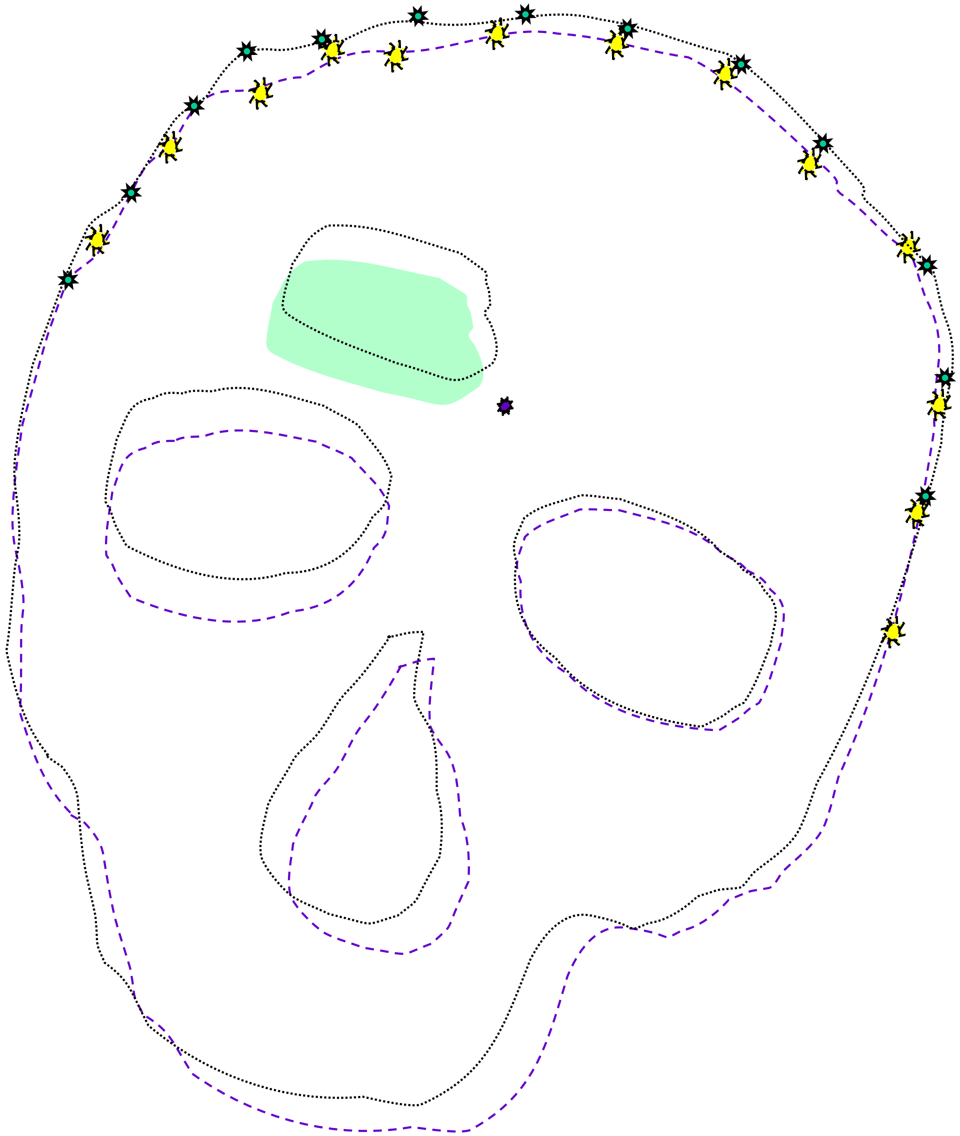
Head-in-hat algorithm: step 2



Head-in-hat algorithm: step 2



Head-in-hat algorithm: step 3



Head in Hat Algorithm

- Strengths
  - Moderately straightforward to implement
  - Slow step is intersecting rays with surface model
  - Works reasonably well for original purpose (registration of skin of head) if have adequate initial guess
- Weaknesses
  - Local minima
  - Assumptions behind use of centroid
  - Requires good initial guess and close matches during convergence

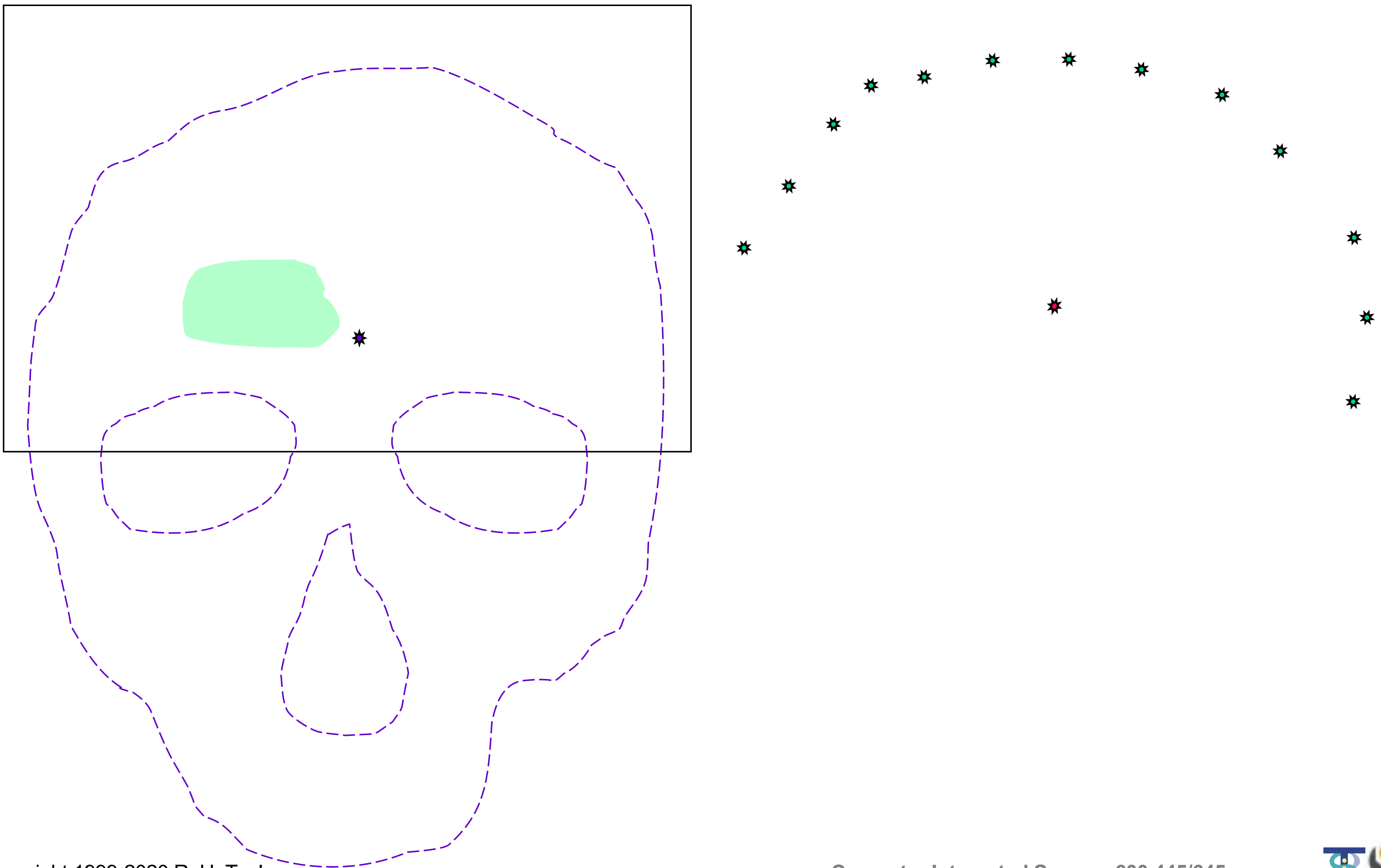
Iterative Closest Point

- Besl and McKay, 1992
- Start with an initial guess, $\mathbf{T}_0$, for $\mathbf{T}$.
- At iteration k
  1. For each sampled point $\mathbf{f}_i \in \mathcal{F}_A$. find the point $\mathbf{v}_i \in \mathcal{F}_B$ that is closest to $\mathbf{T}_k \cdot \mathbf{f}_i$.
  2. Then compute $\mathbf{T}_{k+1}$ as the transformation that minimizes

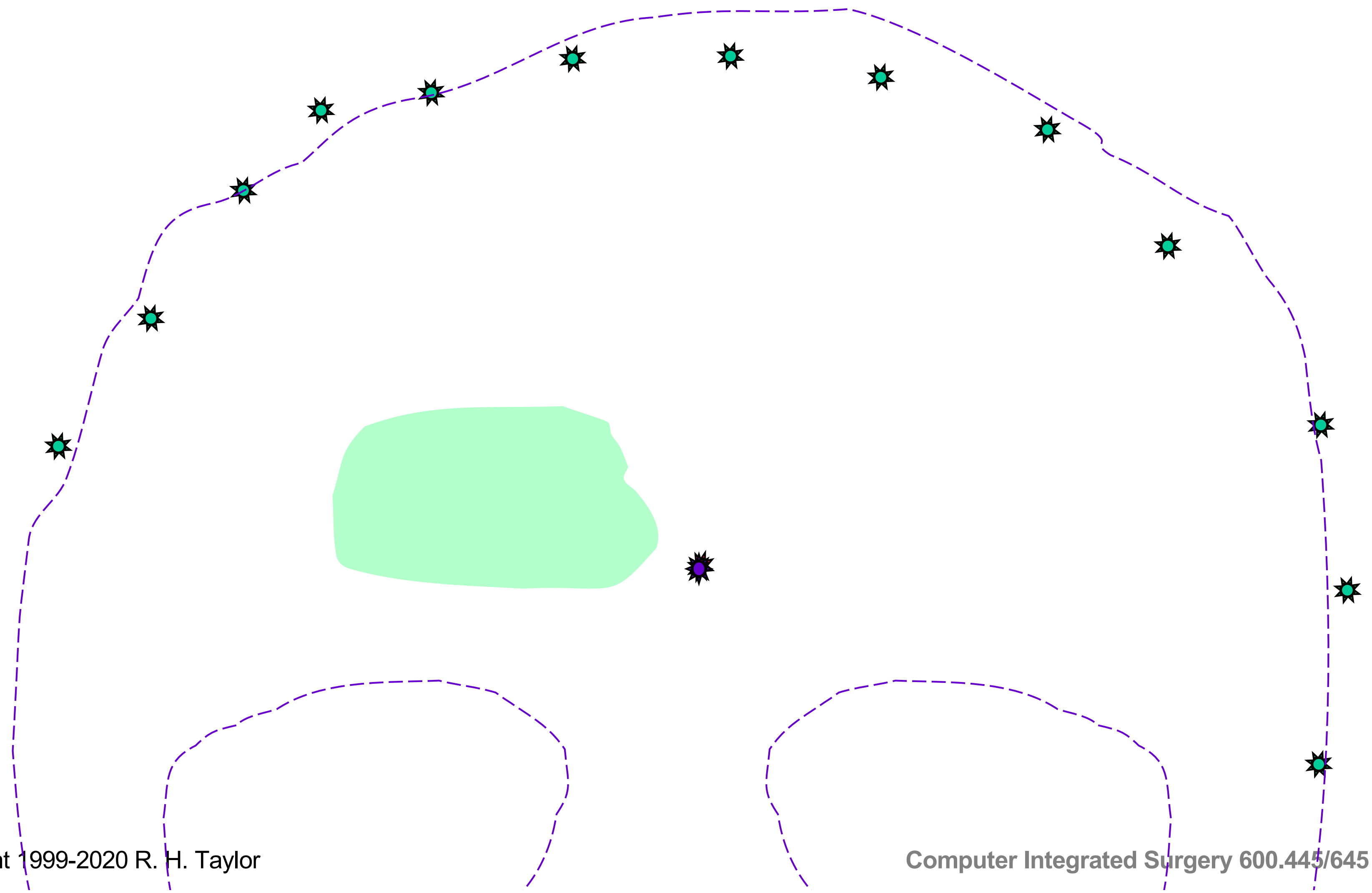
$$D_{k+1} = \sum_i \|\mathbf{v}_i - \mathbf{T}_{k+1} \cdot \mathbf{f}_i)\|^2$$

- Physical Analogy

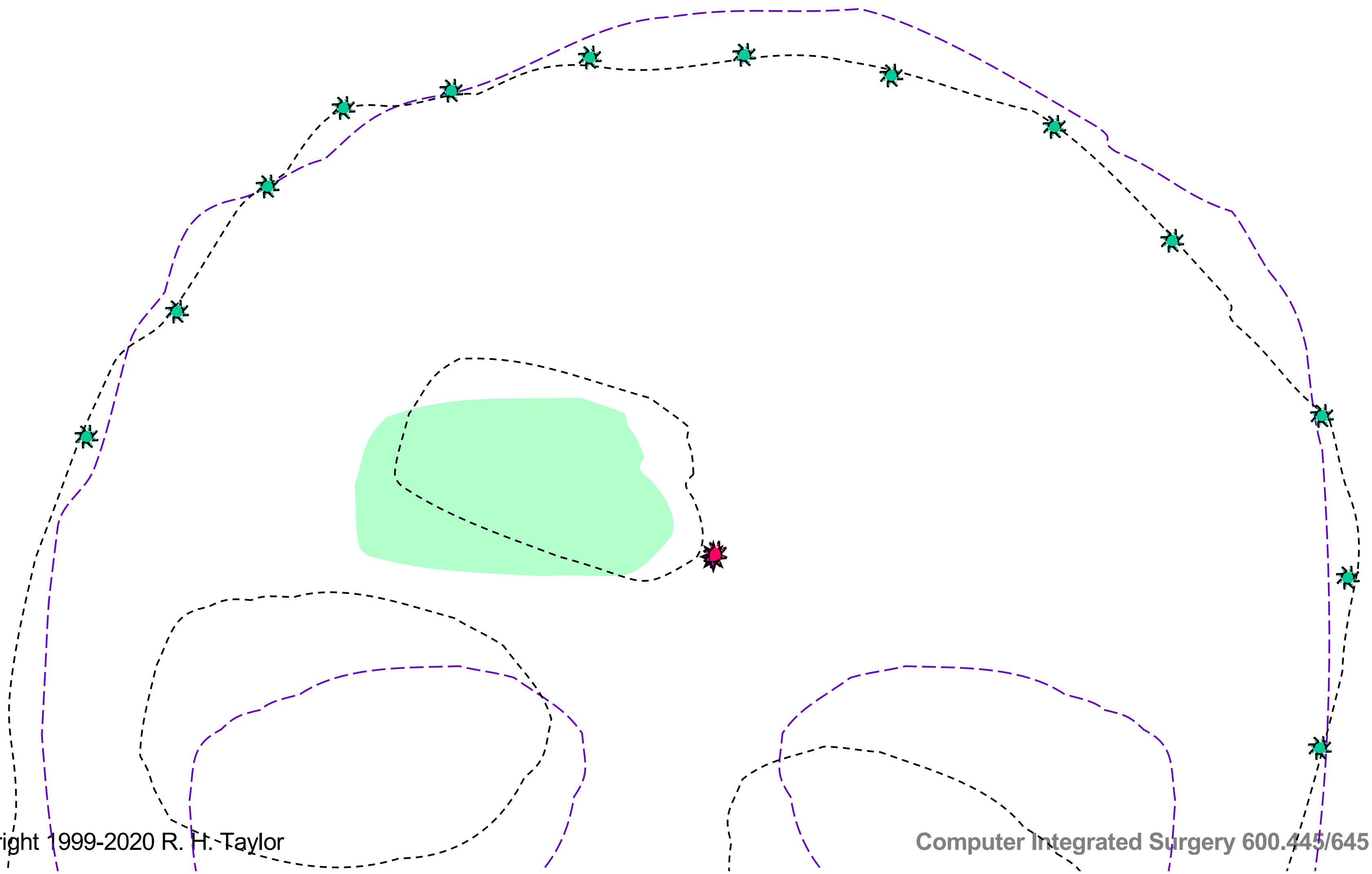
Iterative Closest Point: step 0



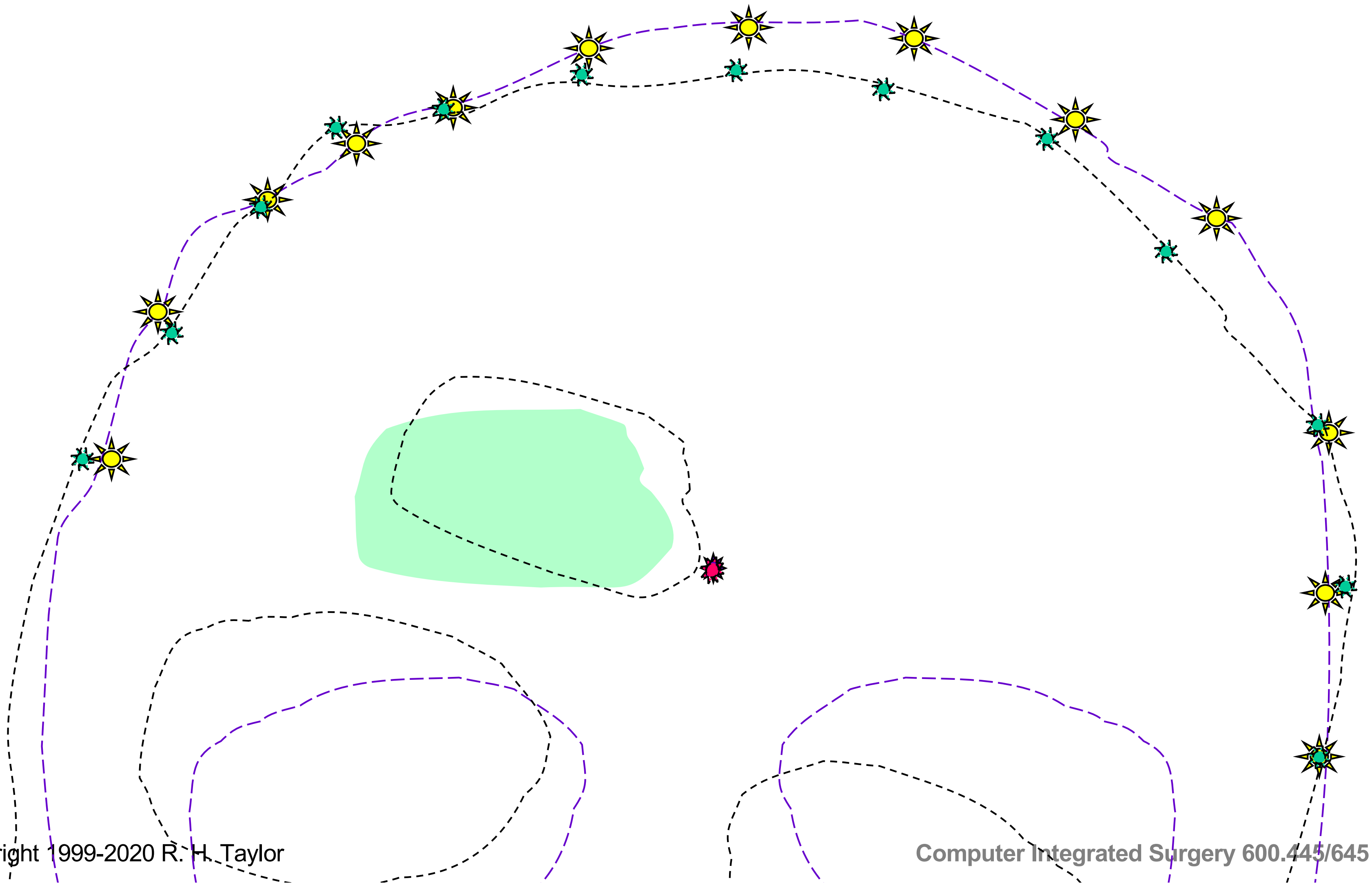
Iterative Closest Point: step1



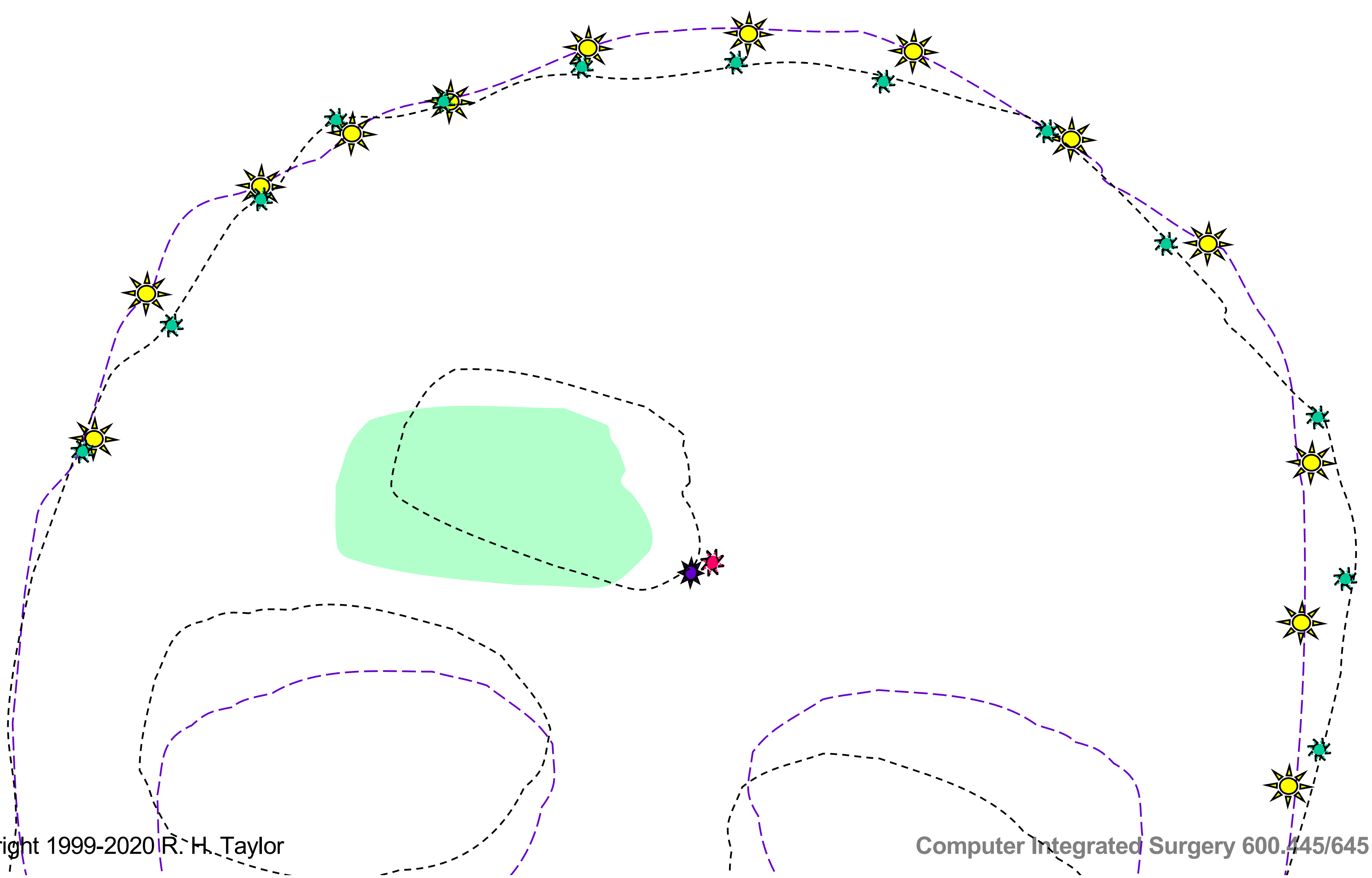
Iterative Closest Point: step1



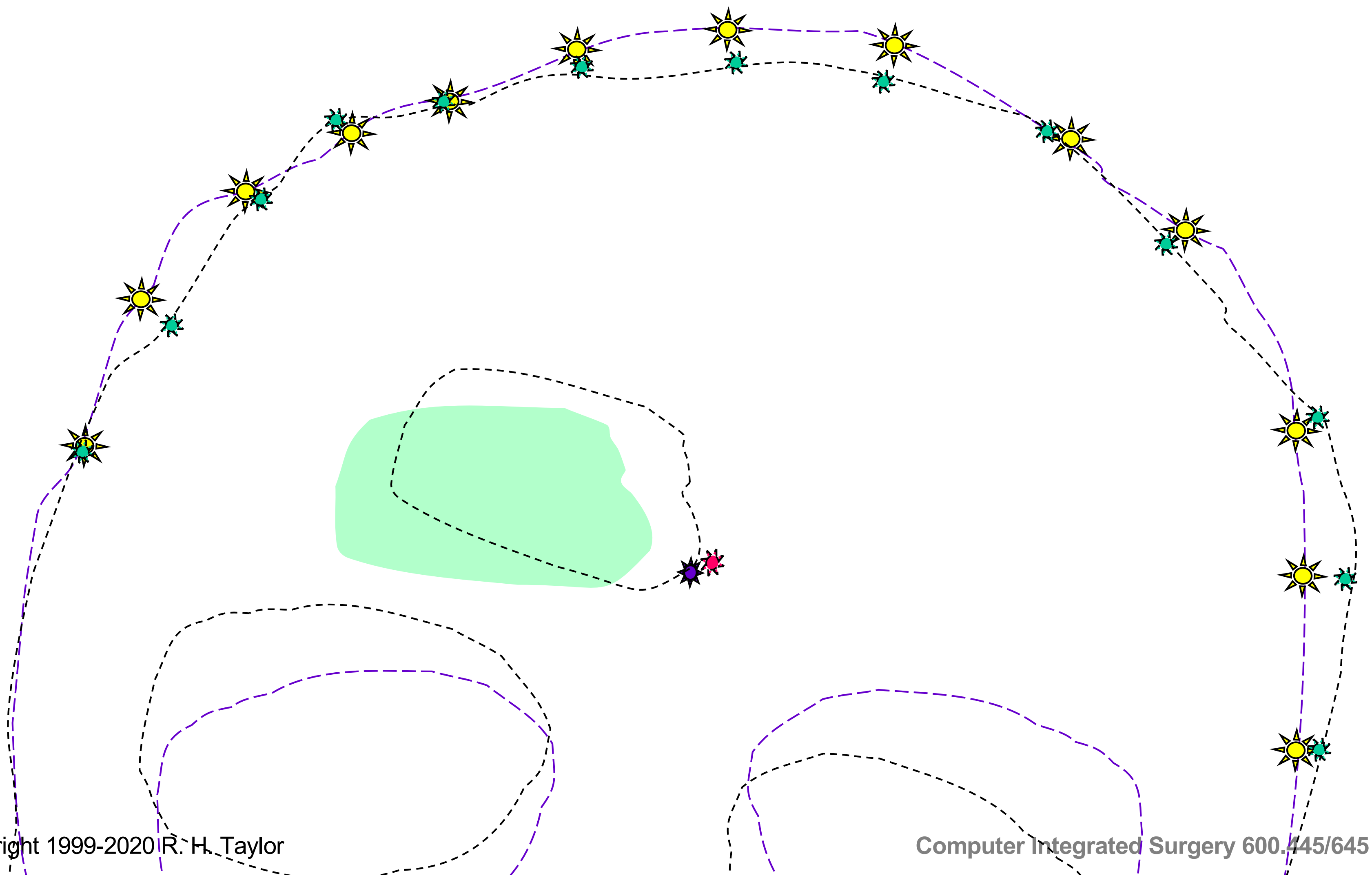
Iterative Closest Point: step2



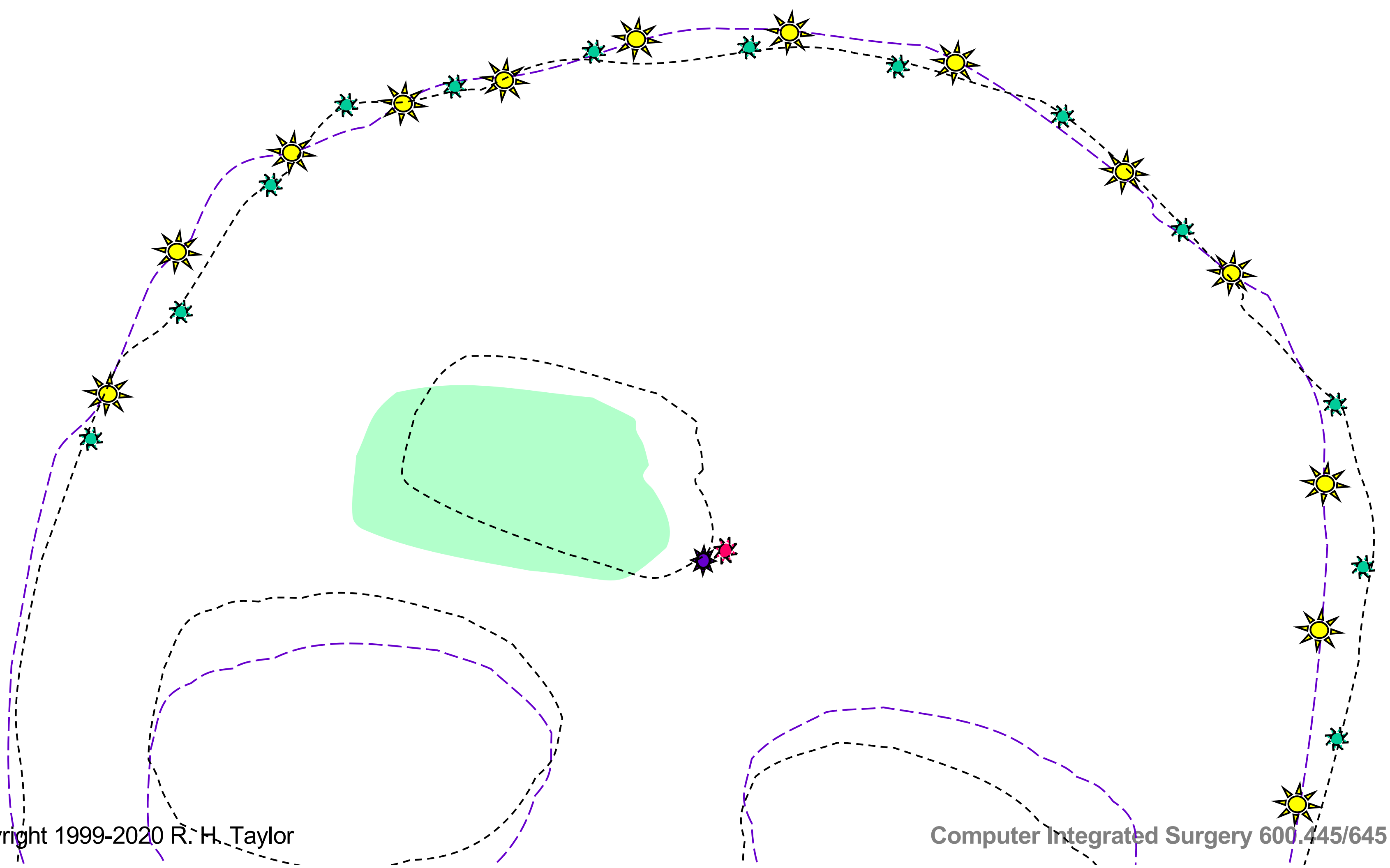
Iterative Closest Point: step 3



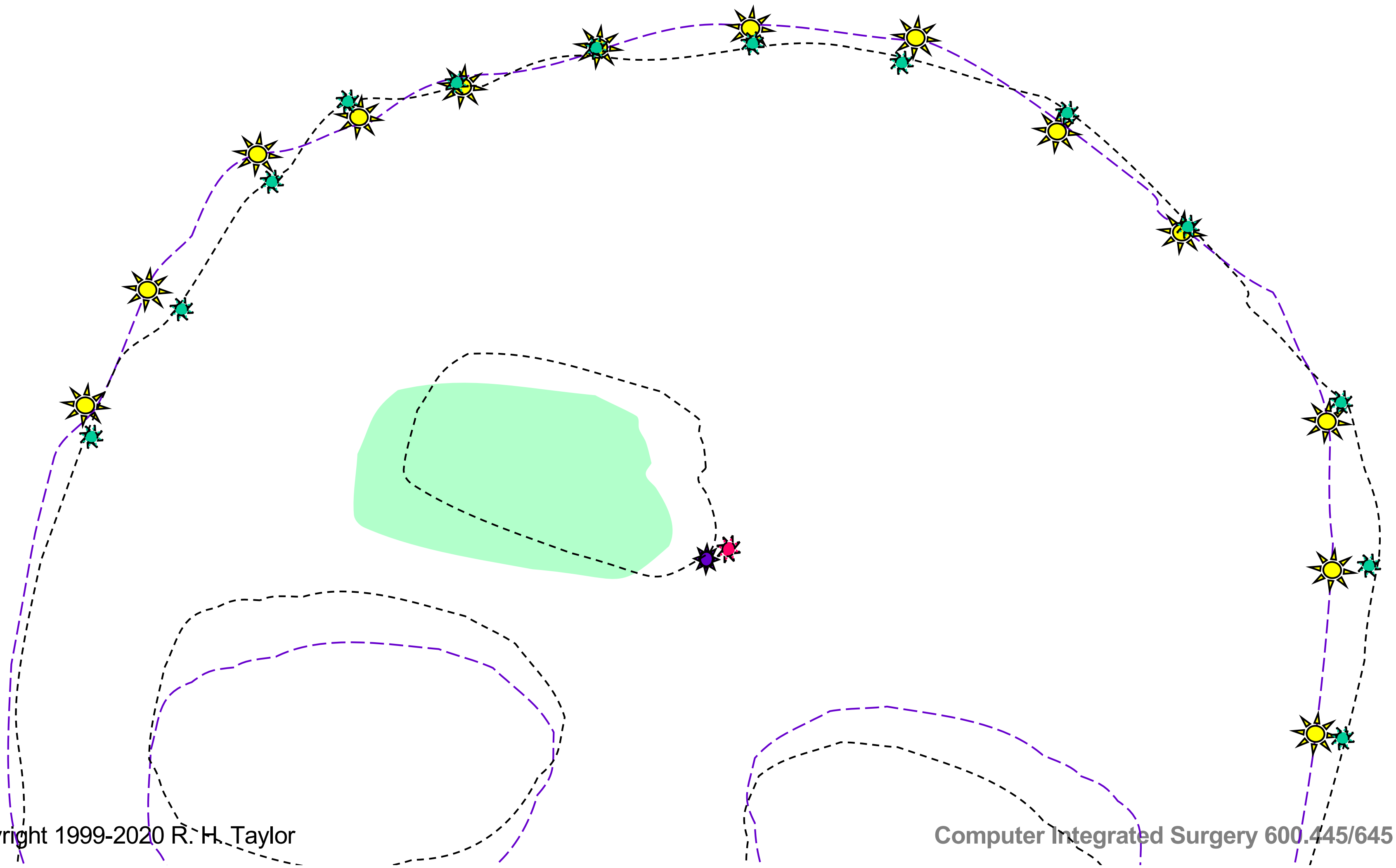
Iterative Closest Point: step 2 interation 2



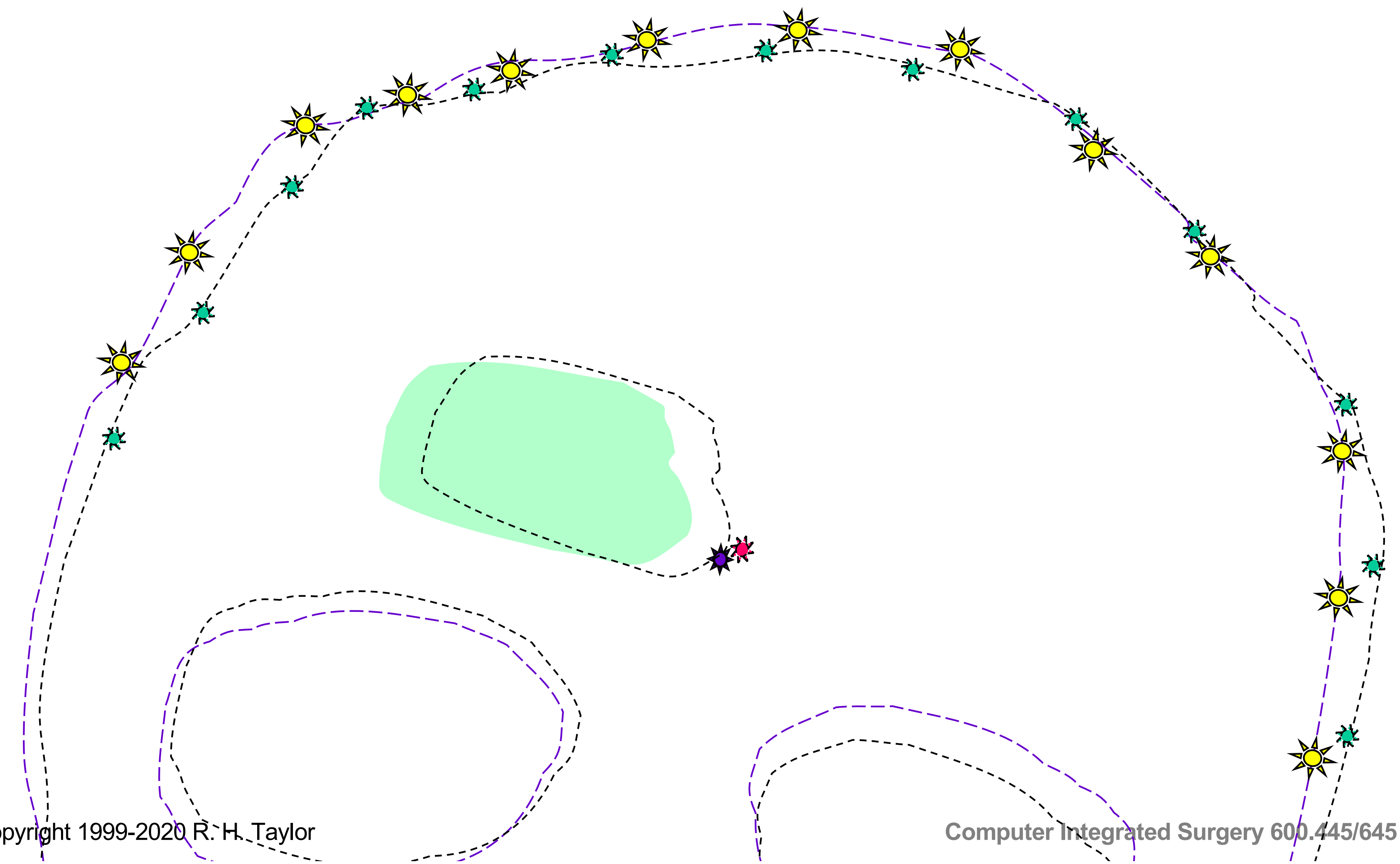
Iterative Closest Point: step 3 interation 2



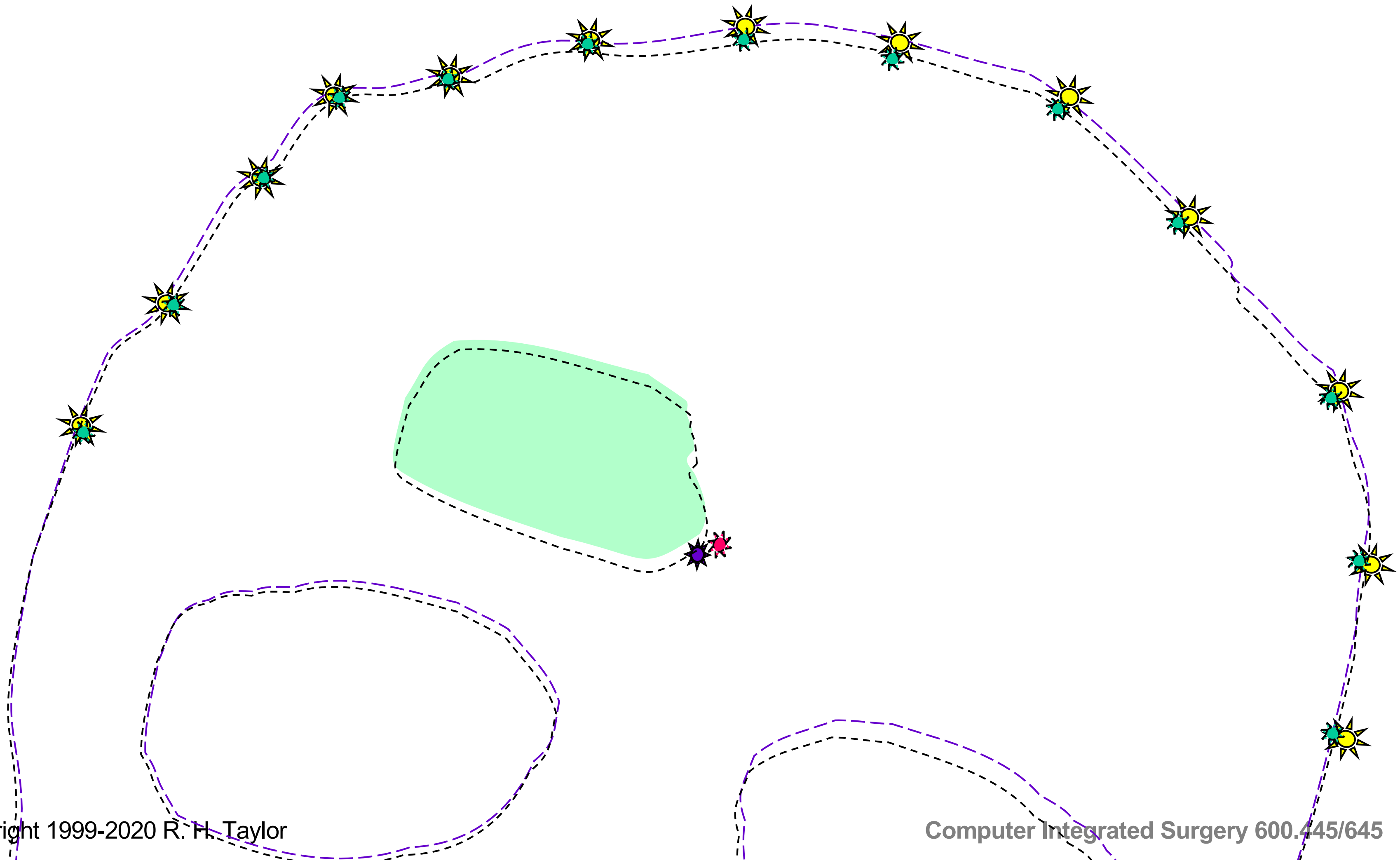
Iterative Closest Point: step 2 interation 3



Iterative Closest Point: step 3 interation 3



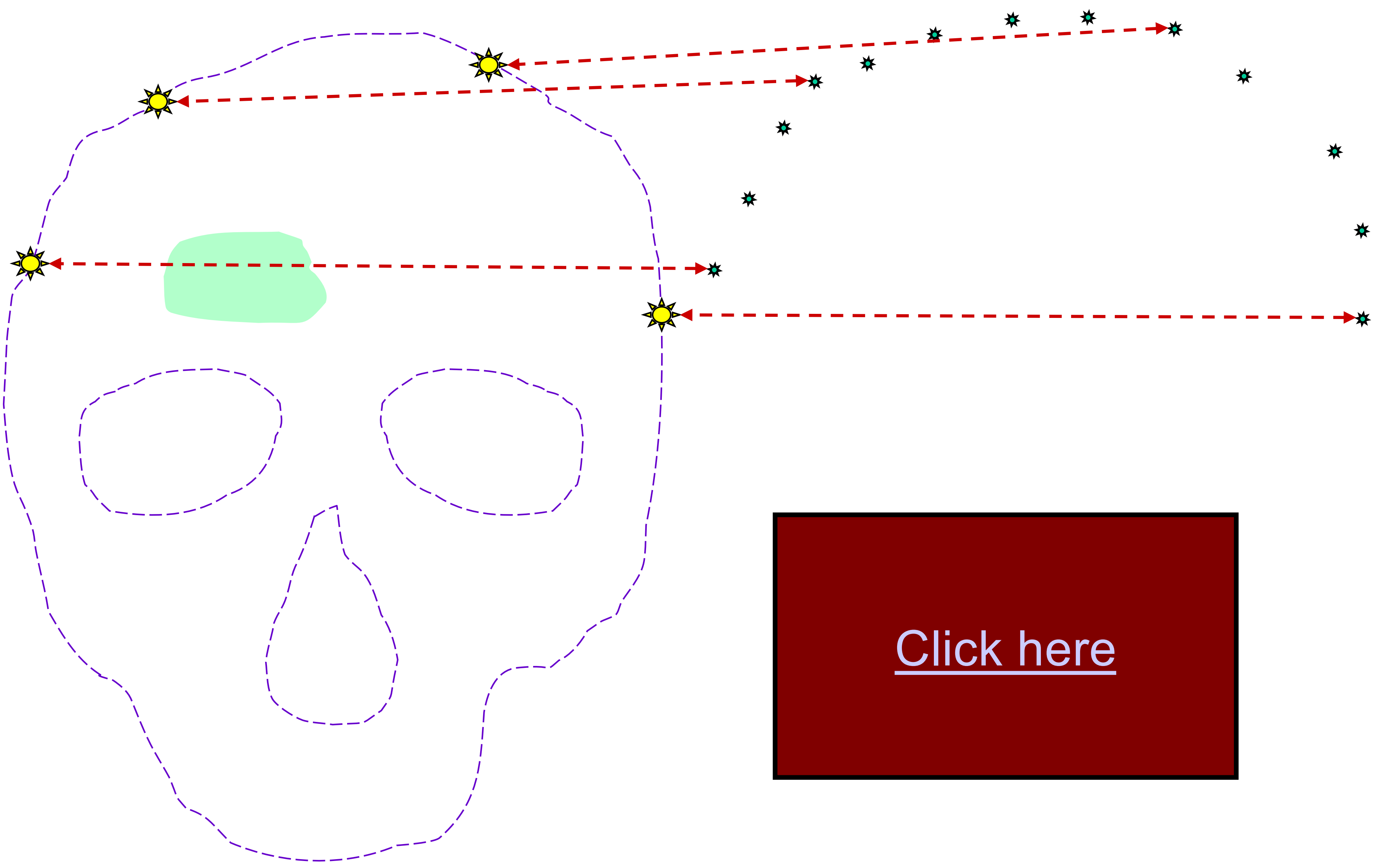
Iterative Closest Point: step 3 interation N



Iterative Closest Point: Discussion

- Minimization step can be fast
- Crucially requires fast finding of nearest points
- Local minima still an issue
- Data overlap still an issue

Digression: Finding Point Pairs



Click here

Outline of a practical ICP code

Given

1. Surface model $\mathtt{M}$ consisting of triangles $\{\mathtt{m}_i\}$
2. Set of points $\mathtt{Q} = \{\vec{\mathbf{q}}_1, \cdots, \vec{\mathbf{q}}_N\}$ known to be on $\mathtt{M}$.
3. Initial guess $\mathbf{F}_0$ for transformation $\mathbf{F}_0$ such that the points $\mathbf{F} \bullet \vec{\mathbf{q}}_k$ lie on $\mathtt{M}$.
4. Initial threshold η_0 for match closeness

Outline of a practical ICP code

Temporary variables

n	Iteration number
$\mathbf{F}_n = [\mathbf{R}, \vec{\mathbf{p}}]$	Current estimate of transformation
η_n	Current match distance threshold
$\mathtt{C} = \{\cdots, \vec{\mathbf{c}}_k, \cdots\}$	Closest points on $\mathtt{M}$ to $\mathtt{Q}$
$\mathtt{D} = \{\cdots, d_k, \cdots\}$	Distances $\mathrm{d}_\mathrm{k} = \left\Vert \vec{\mathbf{c}}_k - \mathbf{F}_n \cdot \vec{\mathbf{q}}_k \right\Vert$
$\mathtt{I} = \{\cdots, i_k, \cdots\}$	Indices of triangles $\mathtt{m}_{i_k}$ corresp. to $\vec{\mathbf{c}}_k$
$\mathtt{A} = \{\cdots, \vec{\mathbf{a}}_k, \cdots\}$	Subset of $\mathtt{Q}$ with valid matches
$\mathtt{B} = \{\cdots, \vec{\mathbf{b}}_k, \cdots\}$	Points on $\mathtt{M}$ corresponding to $\mathtt{A}$
$\mathtt{E} = \{\cdots, \vec{\mathbf{e}}_k, \cdots\}$	Residual errors $\vec{\mathbf{b}}_k - \mathbf{F} \cdot \vec{\mathbf{a}}_k$
$\sigma_n,\ (\varepsilon_{\mathrm{max}})_n, \bar{\varepsilon}_n$	$\dfrac{\sum_\mathrm{k} \vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}{NumElts(\mathtt{E})}$; $\max\limits_\mathrm{k} \sqrt{\vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}$; $\dfrac{\sum_\mathrm{k} \sqrt{\vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}}{NumElts(\mathtt{E})}$

Outline of a practical ICP code

Step 0 : (initialization)

Input surface model $\mathtt{M}$ and points $\mathtt{Q}$.

Build an appropriate data structure (e.g., octree, kD tree) $\mathtt{T}$ to facilitate finding the closest point matching search.

$$n \leftarrow 0; \quad \eta_n \leftarrow \text{large number}$$

$$\mathtt{I} \leftarrow \{\cdots, 1, \cdots\}$$

$$\mathtt{C} \leftarrow \{\cdots, \text{point on } \mathtt{m}_1, \cdots\}$$

$$\mathtt{D} \leftarrow \{\cdots, \left\|\vec{\mathbf{c}}_k - \mathbf{F}_0 \bullet \vec{\mathbf{q}}_k\right\|, \cdots\}$$

Outline of a practical ICP code

Step 1: (matching)

$\mathtt{A} \leftarrow \varnothing; \ \ \mathtt{B} \leftarrow \varnothing$

For $k \leftarrow 1$ step 1 to N do

begin

$bnd_k = \left\| \mathbf{F}_n \bullet \vec{\mathbf{q}}_k - \vec{\mathbf{c}}_k \right\|$

$\left[\vec{\mathbf{c}}_k, i, d_k \right] \leftarrow \text{FindClosestPoint}\left(\mathbf{F}_n \bullet \vec{\mathbf{q}}_k, \vec{\mathbf{c}}_k, i_k, bnd_k, \mathtt{T} \right);$

// Note: develop first with simple
// search. Later make more
// sophisticated, using $\mathtt{T}$

if ($d_k < \eta_n$) then { put $\vec{\mathbf{q}}_k$ into $\mathtt{A}$; put $\vec{\mathbf{c}}_k$ into $\mathtt{B}$; };

// See also subsequent notes

end

Outline of a practical ICP code

Step 1: (matching)

$\mathtt{A} \leftarrow \emptyset; \;\; \mathtt{B} \leftarrow \emptyset$

For $k \leftarrow 1$ step 1 to N do

begin

$bnd_k = \left\| \mathbf{F}_n \bullet \vec{\mathbf{q}}_k - \vec{\mathbf{c}}_k \right\|$

$\left[\vec{\mathbf{c}}_k, i, d_k \right] \leftarrow$ Fin

//

//

//

if ($d_k < \eta_n$) then

//

end

Note: If using a tree search, you can use previous match to get a reasonable initial bound. E.g.,

$$bnd_k = \left\| \vec{\mathbf{c}}_k - \mathbf{F}_n \bullet \vec{\mathbf{q}}_k \right\|$$

and then pass that to the tree search. Alternatively, you can find the closest point on triangle i_k and use that to get an initial bound bnd_k for the search

Outline of a practical ICP code

Step 2 : (transformation update)

$n \leftarrow n+1$

$\mathbf{F}_n \leftarrow \text{FindBestRigidTransformation}(\mathtt{A},\mathtt{B})$

$$\sigma_n \leftarrow \frac{\sqrt{\sum_k \vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}}{NumElts(\mathtt{E})}; \quad (\varepsilon_{\text{max}})_n \leftarrow \max_k \sqrt{\vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k};\ \bar{\varepsilon}_n \leftarrow \frac{\sum_k \sqrt{\vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}}{NumElts(\mathtt{E})}$$

Step 3 : (adjustment)

Compute η_n from $\{\eta_0,\cdots,\eta_{n-1}\}$ // see notes next page

// May also update $\mathbf{F}_n$ from $\{\mathbf{F}_0,\cdots,\mathbf{F}_n\}$ (see Besl & McKay)

Step 4 : (iteration)

if TerminationTest($\{\sigma_0,\cdots,\sigma_n\},\{(\varepsilon_{\text{max}})_0,\cdots,(\varepsilon_{\text{max}})_n, \{\bar{\varepsilon}_0,\cdots,\bar{\varepsilon}_n\}\}$)

then stop. Otherwise, go back to step 1 // see notes

Outline of practical ICP code

Threshold η_n update

The threshold η_n can be used to restrict the influence of clearly wrong matches on the computation of $\mathbf{F}_n$. Generally, it should start at a fairly large value and then decrease after a few iterations. One not unreasonable value might be something like $3\overline{\varepsilon}_n$. If the number of valid matches begins to fall significantly, one can increase it adaptively. Too tight a bound may encourage false minima

Also, if the mesh is incomplete, it may be advantageous to exclude any matches with triangles at the edge of the mesh.

Outline of practical ICP code

Termination test

There are no hard and fast rules for deciding when to terminate the procedure. One criterion might be to stop when $\sigma_n, \overline{\varepsilon}_n$ and/or $(\varepsilon_{max})_n$ are less than desired thresholds and $\gamma \le \overline{\varepsilon}_n / \overline{\varepsilon}_{n-1} \le 1$ for some value γ (e.g., $\gamma \cong .95$) for several iterations.

Short further note: ICP related methods

- There is an extensive literature on methods based on ideas similar to ICP. Surveys and tutorials describing some of them may be found at
  - http://www.cs.princeton.edu/~smr/papers/fasticp/fasticp_paper.pdf
  - http://www.mrpt.org/Iterative_Closest_Point_%28ICP%29_and_other_matching_algorithms
- There are also a number of methods that incorporate a probabilistic framework. One example is the "Generalized-ICP" method of Segal, Haehnel, and Thrun
  - Aleksandr V. Segal, Dirk Haehnel, and Sebastian Thrun, "Generalized-ICP", in *Robotics: Science and Systems,* 2009.
  - http://www.robots.ox.ac.uk/~avsegal/resources/papers/Generalized_ICP.pdf

Typical Generalized ICP Algorithm

Outline below is based mostly on from paper by A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP", in *Robotics: Science and Systems*, 2009.

$n \leftarrow 0$;initialize $\mathbf{F}_0$, threshold value η_0, distribution parameters Φ

Step 1: (matching)

$\mathtt{A} \leftarrow \varnothing$; $\mathtt{B} \leftarrow \varnothing$

For $k \leftarrow 1$ step 1 to N do

begin

$\left[\vec{\mathbf{c}}_k, i_k, d_k\right] \leftarrow \text{FindClosestPoint}\left(\mathbf{F}_n \bullet \vec{\mathbf{q}}_k, \vec{\mathbf{c}}_k, i_k, \mathtt{T}\right)$;

if ($d_k < \eta_n$) then { put $\vec{\mathbf{q}}_k$ into $\mathtt{A}$; put $\vec{\mathbf{c}}_k$ into $\mathtt{B}$; };

\\ alternative: test if $prob(\vec{\mathbf{q}}_k \sim \vec{\mathbf{c}}_k) > \eta_n$

end

Step 2: (transformation update)

$n \leftarrow n+1$

$$\mathbf{F}_n \leftarrow \underset{\mathbf{F}}{\arg\max}\; prob(\mathbf{F} \bullet A \sim B; \Phi) = \underset{\mathbf{F}}{\arg\max} \prod_i prob(\mathbf{F} \bullet \vec{\mathbf{a}}_i \sim \vec{\mathbf{b}}_i; \Phi)$$

$$= \underset{\mathbf{F}}{\arg\min} \sum_i -\log prob(\mathbf{F} \bullet \vec{\mathbf{a}}_i \sim \vec{\mathbf{b}}_i; \Phi)$$

Step 3: (adjustment)

update threshold η_n and distribution parameters Φ

Step 4: (iteration)

if TerminationTest($\cdots$) then stop. Otherwise, go back to step 1 // see notes

Related concept: Estimation with Uncertainty

Suppose you know something about the uncertainty of the sample data at each point pair (e.g., from sensor noise and/or model error). I.e.,

$$\vec{\mathbf{a}}_k \in \boldsymbol{A}_k; \;\; \vec{\mathbf{b}}_k \in \boldsymbol{B}_k; \;\; \mathrm{cov}(\boldsymbol{A}_k, \boldsymbol{B}_k) = \mathbf{C}_k = \mathbf{Q}_k \Lambda_k \mathbf{Q}_k^T$$

Then an appropriate distance metric is the Mahalabonis distance

$$\mathrm{D}(\vec{\mathbf{a}}_k, \vec{\mathbf{b}}_k) = (\vec{\mathbf{a}}_k - \vec{\mathbf{b}}_k)^T \mathbf{C}_k^{-1} (\vec{\mathbf{a}}_k - \vec{\mathbf{b}}_k) = \vec{\mathbf{d}}_k^T \Lambda_k^{-1} \vec{\mathbf{d}}_k$$

where

$$\vec{\mathbf{d}}_k = \mathbf{Q}_k^T (\vec{\mathbf{a}}_k - \vec{\mathbf{b}}_k)$$

This approach is readily extended to the case where the samples are not independent.

Distance Maps

- Many authors
- Somewhat related to ICP and also to level sets
- Basic idea is to precompute the distance to the surface for a dense sampling of the volume.
- Then use the gradient of the distance map to compute an incremental motion that reduces the sum of the distances of all the moving points to the surface.
- Then iterate

Distance Maps

There are a number of very fast algorithms for computing the Euclidean Distance Transform (distance to surface of each point in an image at each point in a 3D volume grid). One example is:

> J. C. Torelli, R. Fabbri, G. Travieso, and O. Bruno, "A High Performance 3D Exact Eeuclidean Distance Transform Algorithm for Distributed Computing", *International Journal of Pattern Recognition and Artificial Intelligence, vol. 24- 6, pp. 897-915, 2010.*

But a web search will disclose many others, together with open source code

Distance Maps

Given

a current registration transformation $\mathbf{F}$

Euclidean distance map $d(\vec{\mathbf{p}})$

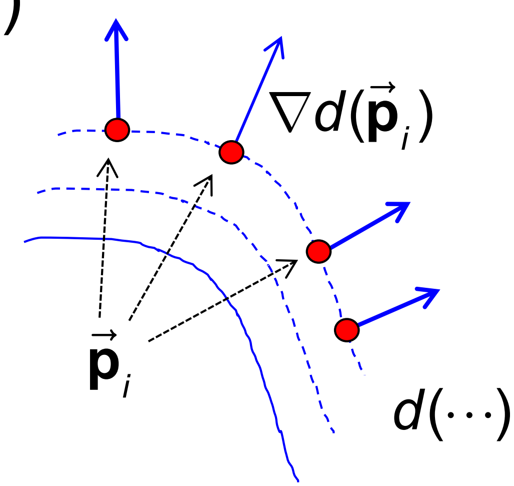
For each sample point $\vec{\mathbf{f}}_i$ compute $\vec{\mathbf{p}}_i = \mathbf{F} \bullet \vec{\mathbf{f}}_i$

Compute a small motion $\Delta\mathbf{F}$

$$\Delta\mathbf{F} = \underset{\Delta\mathbf{F}}{\arg\min} \sum_i (\Delta\mathbf{F} \bullet \vec{\mathbf{p}}_i - \vec{\mathbf{p}}_i) \bullet \nabla d(\vec{\mathbf{p}}_i)$$

Update $\mathbf{F} \leftarrow \Delta\mathbf{F} \bullet \mathbf{F}$

Iteate



Distance Maps

Given

a current registration transformation **F**

Euclidean distance map $d(\vec{\mathbf{p}})$

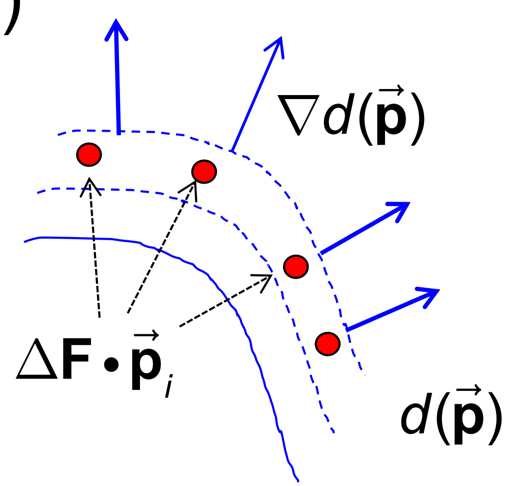
For each sample point $\vec{\mathbf{f}}_i$ compute $\vec{\mathbf{p}}_i = \mathbf{F} \bullet \vec{\mathbf{f}}_i$

Compute a small motion $\Delta\mathbf{F}$

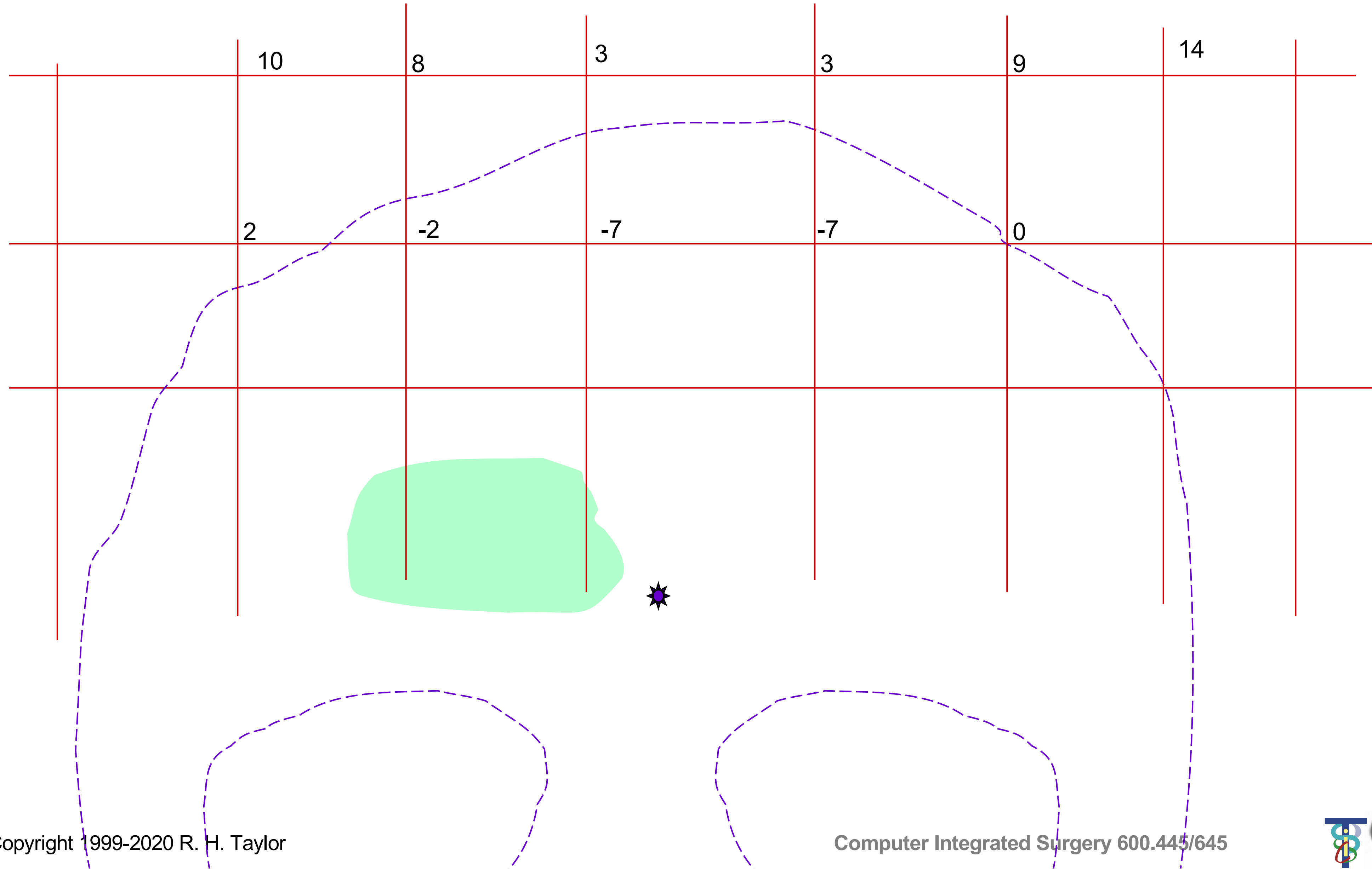
$$\Delta\mathbf{F} = \underset{\Delta\mathbf{F}}{\arg\min} \sum_i (\Delta\mathbf{F} \bullet \vec{\mathbf{p}}_i - \vec{\mathbf{p}}_i) \bullet \nabla d(\vec{\mathbf{p}}_i)$$

Update $\mathbf{F} \leftarrow \Delta\mathbf{F} \bullet \mathbf{F}$

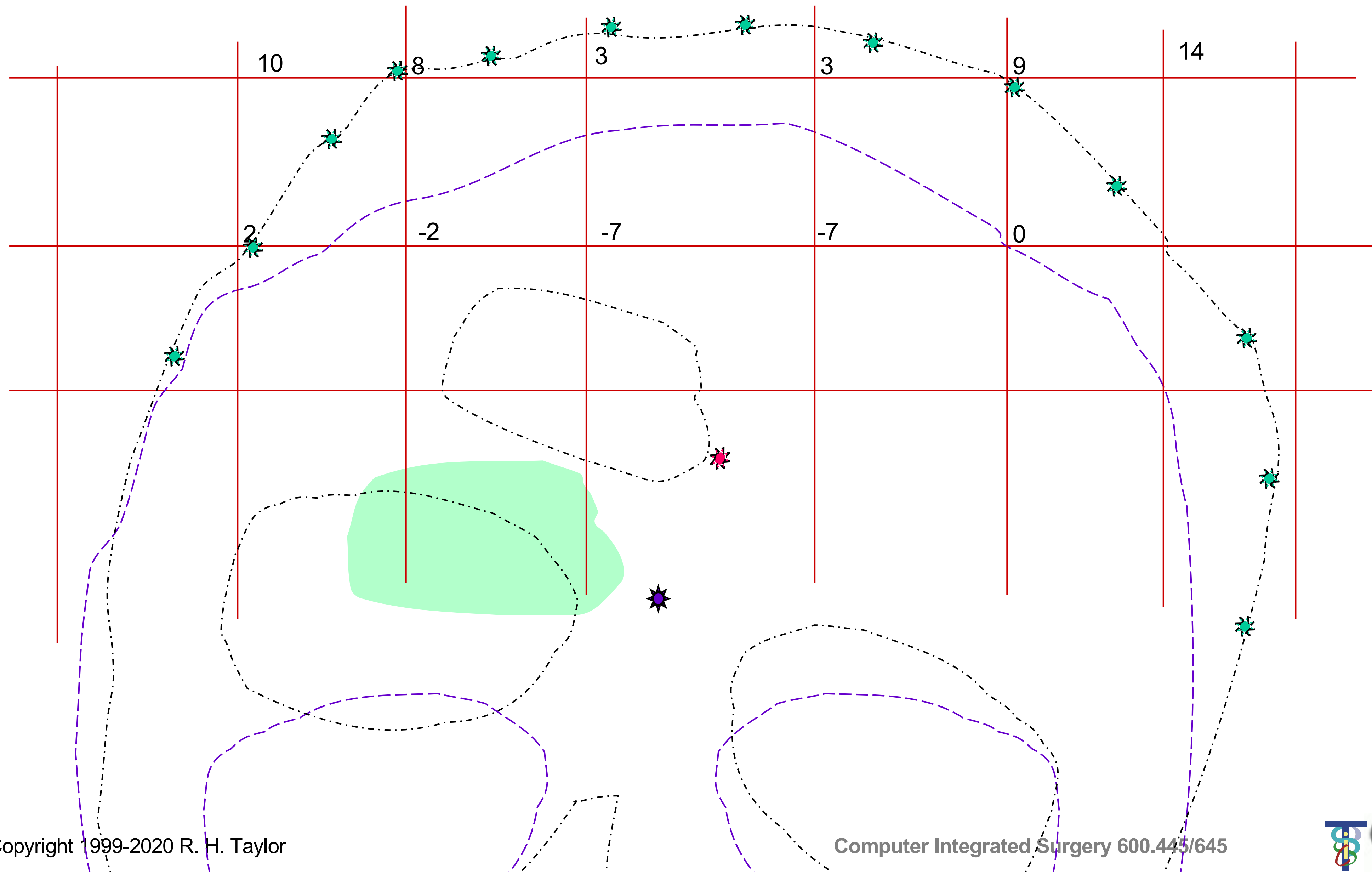
Iteate



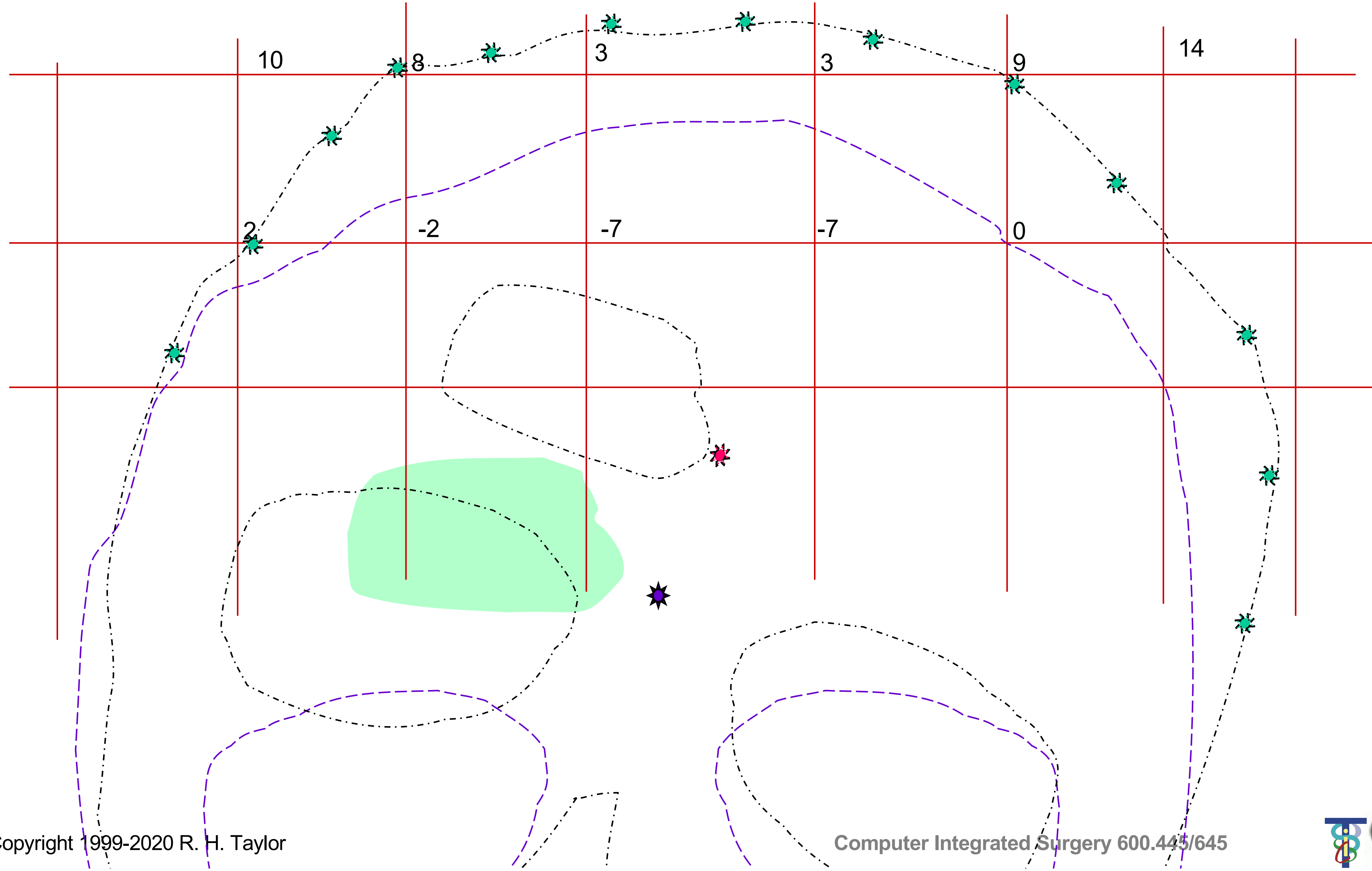
Distance Maps: step 0



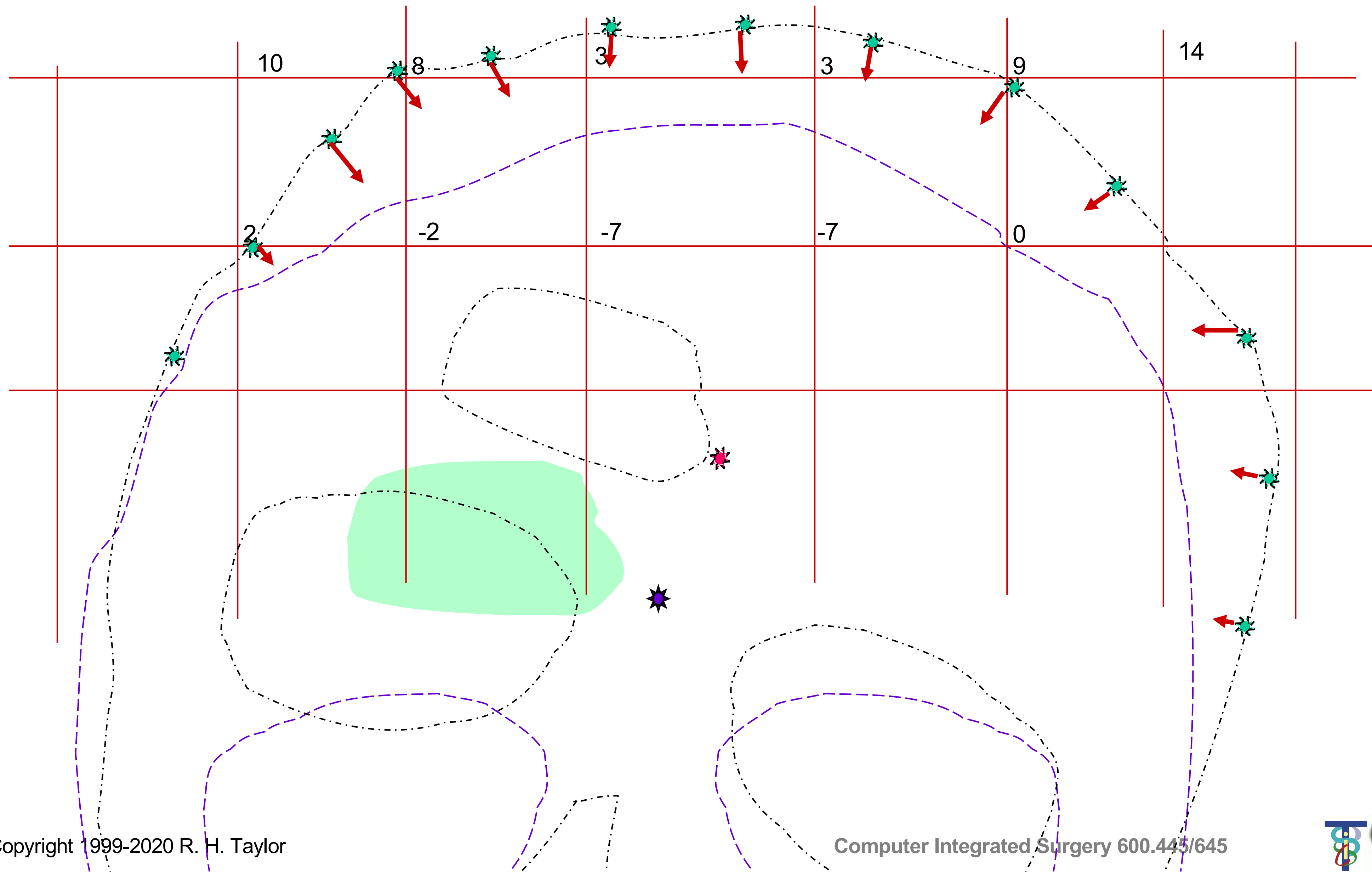
Distance Maps: step 1



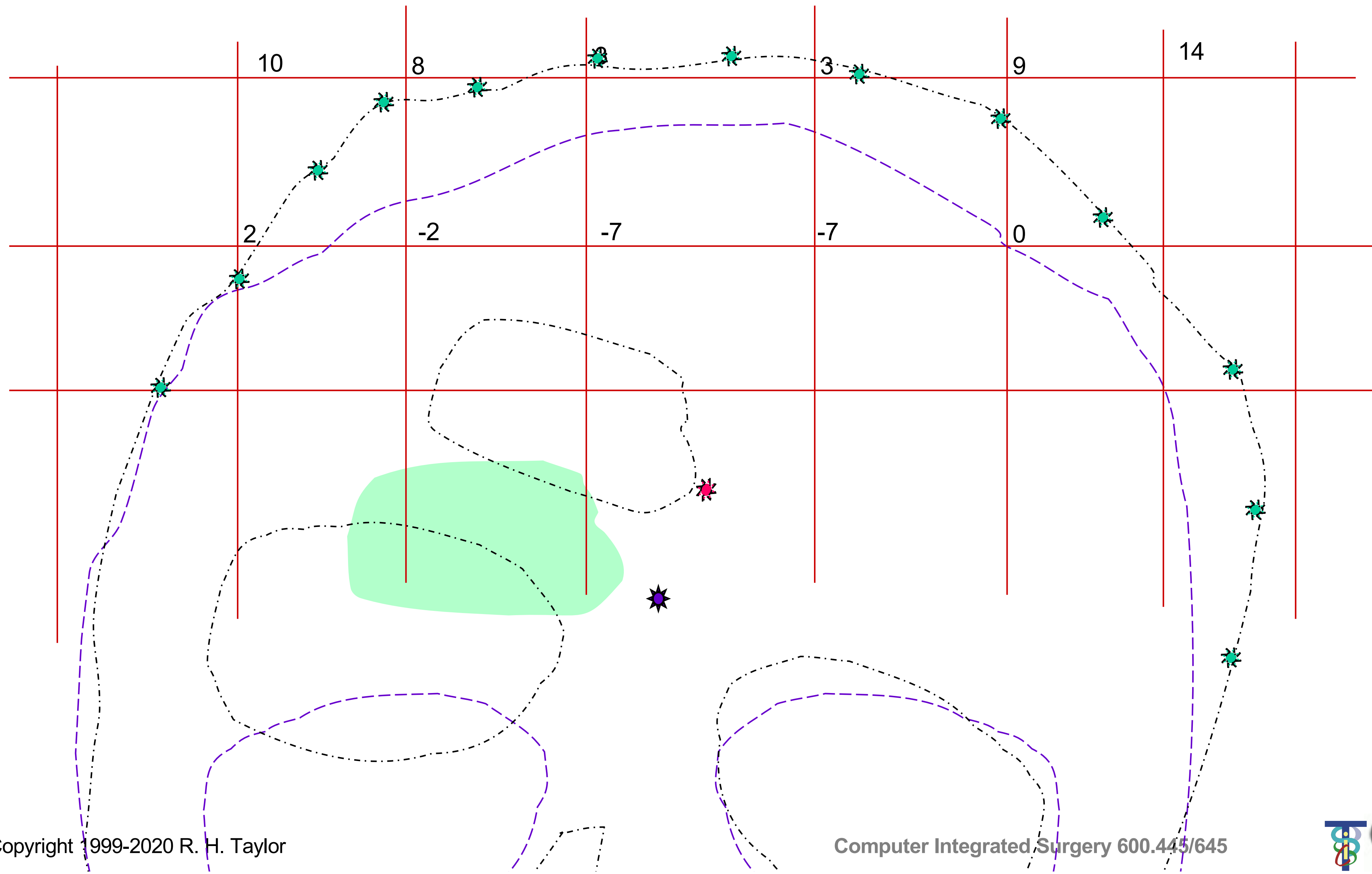
Distance Maps: step 1



Distance Maps: step 2



Distance Maps: step 3



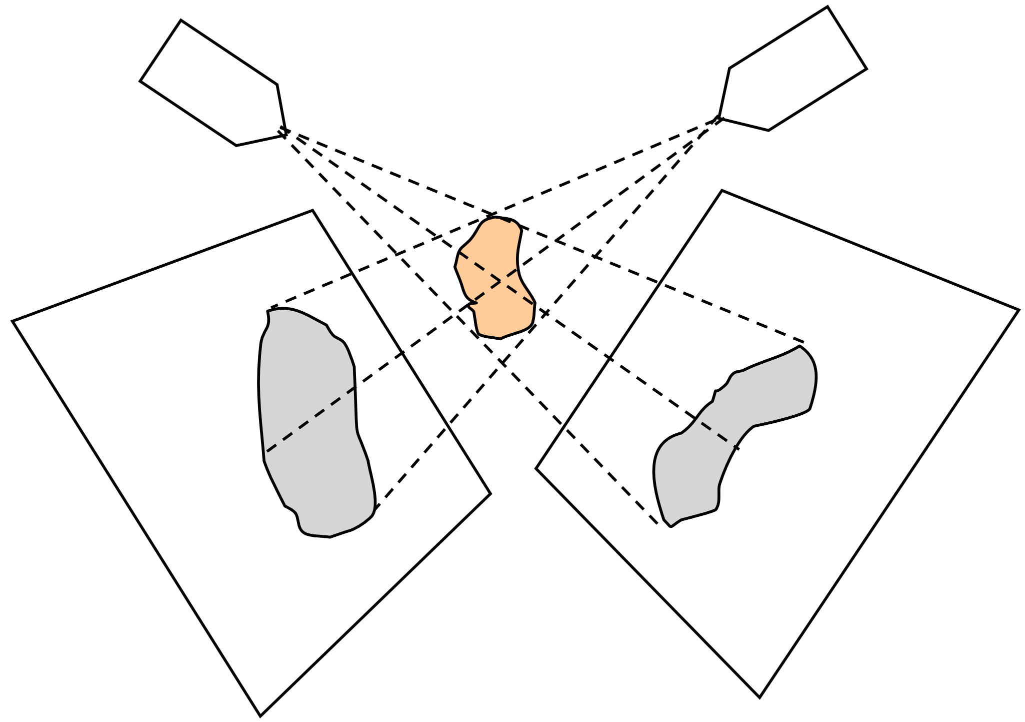
Feature-Based 2D-3D Registration

Given

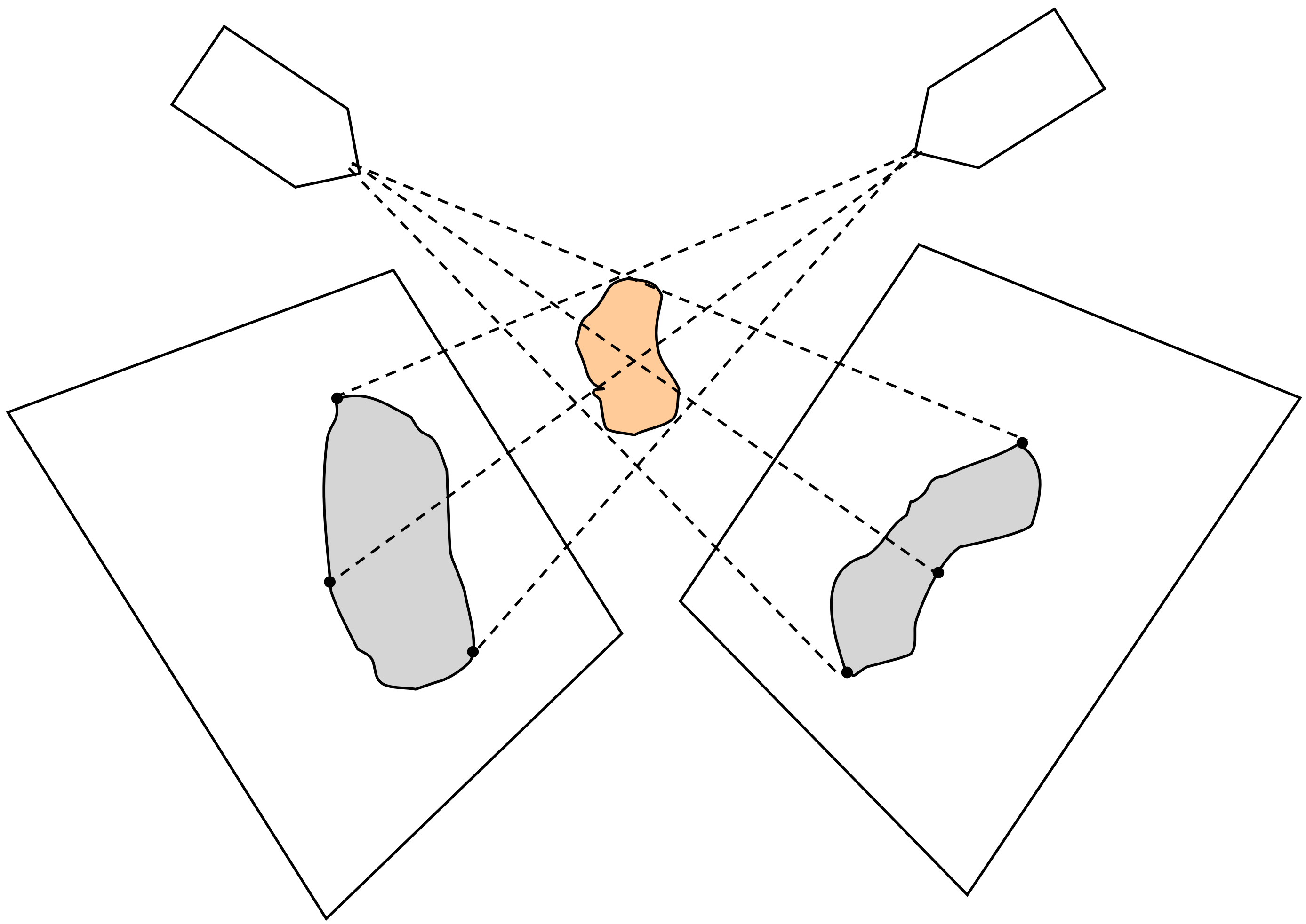
- 3D surface model of an anatomic structure
- Multiple 2D x-ray projection images taken at known poses relative to some coordinate system C
- Initial estimate of the pose **F** of the anatomic object relative to the x-ray imaging coordinate system C

Goal

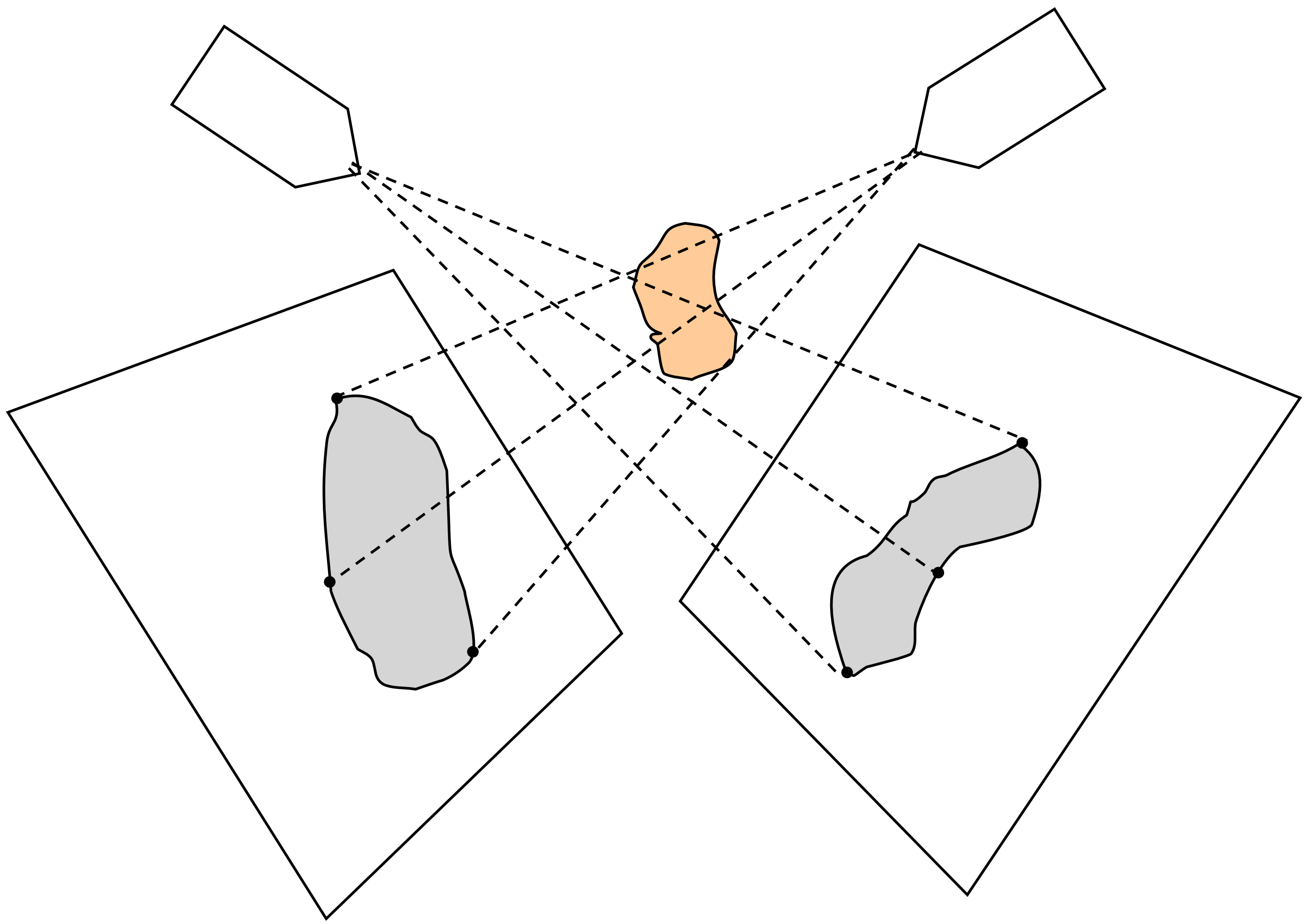
- Compute an accurate value for **F**



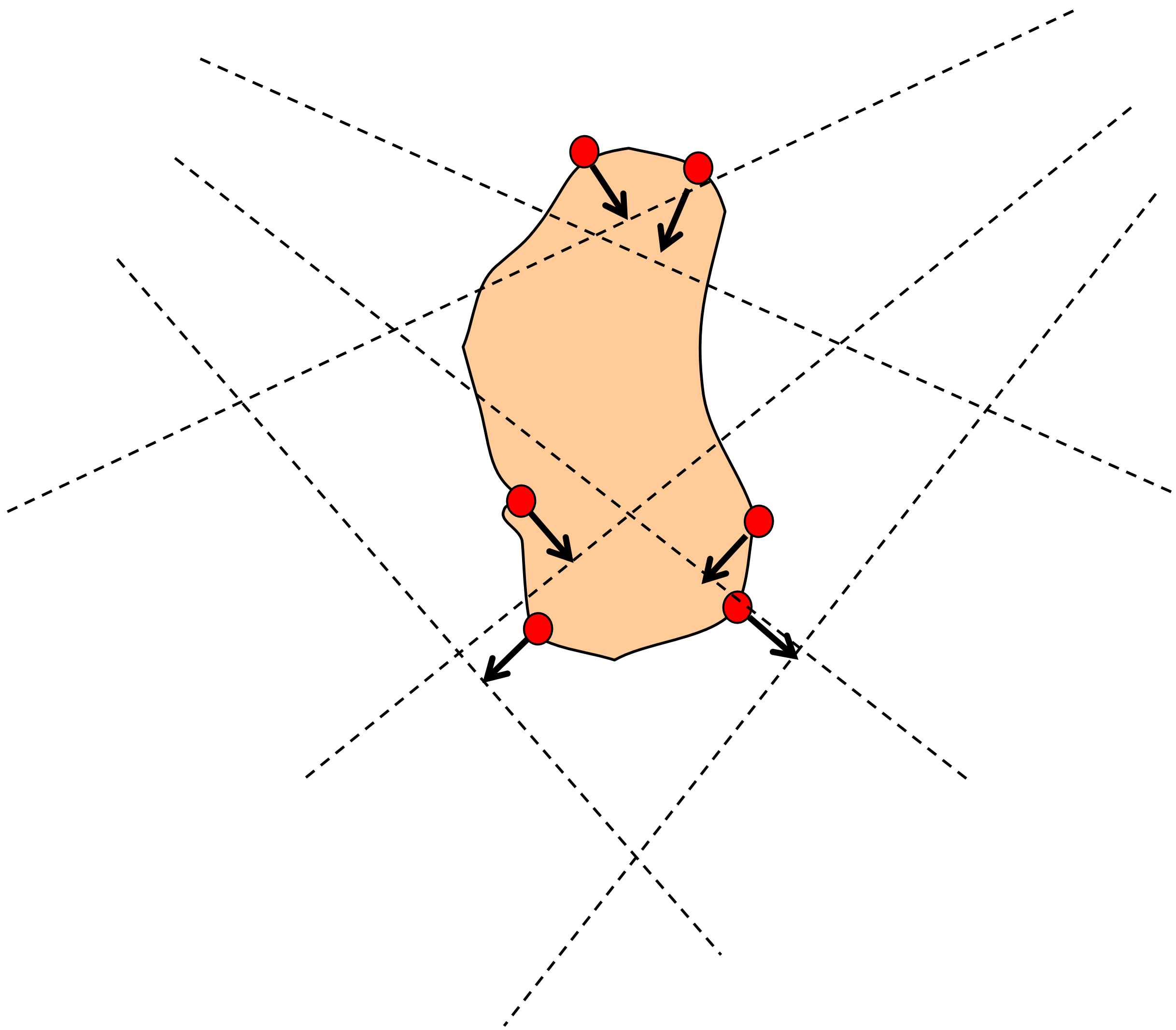
Feature-Based 2D-3D Registration



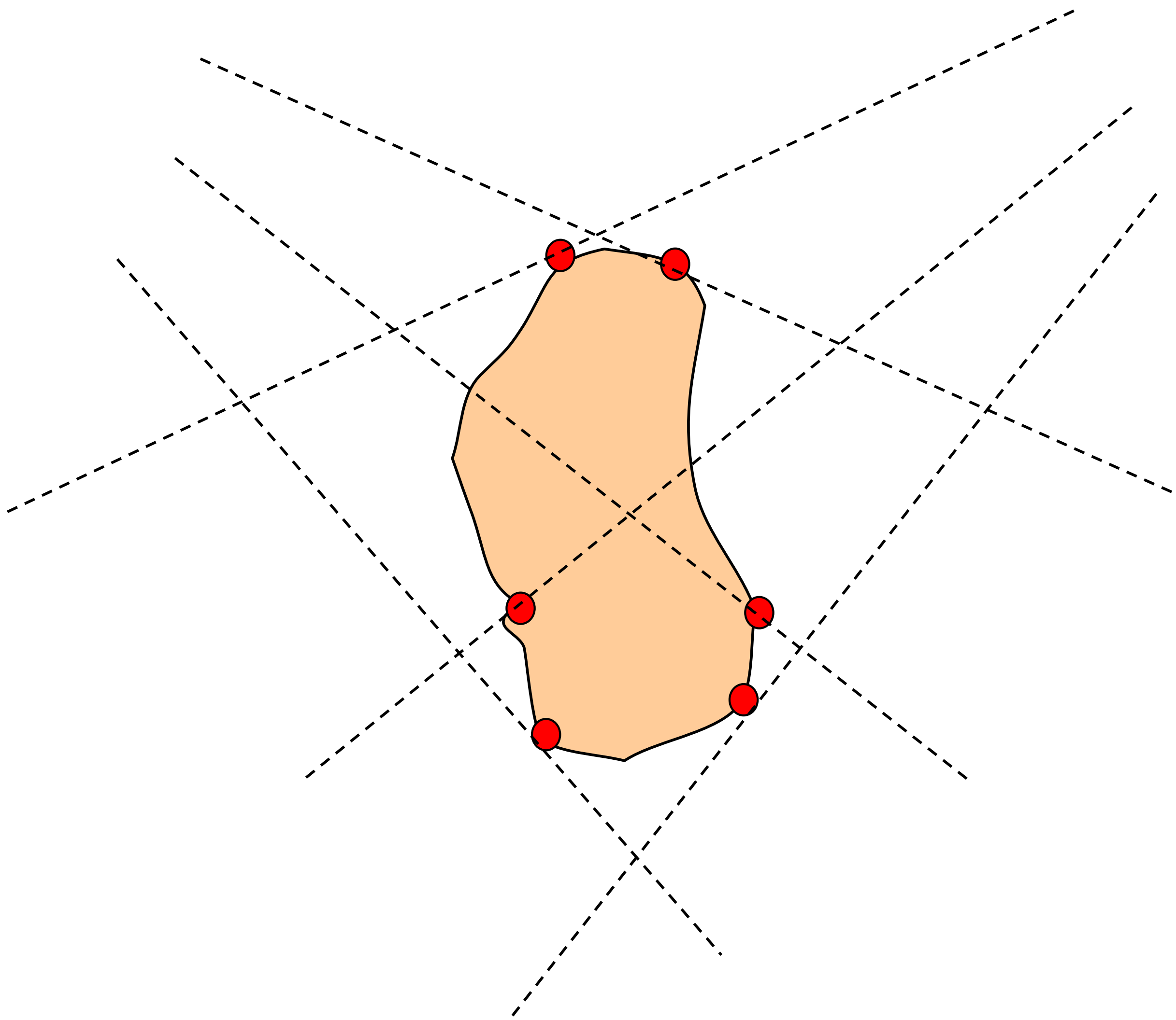
Feature-Based 2D-3D Registration



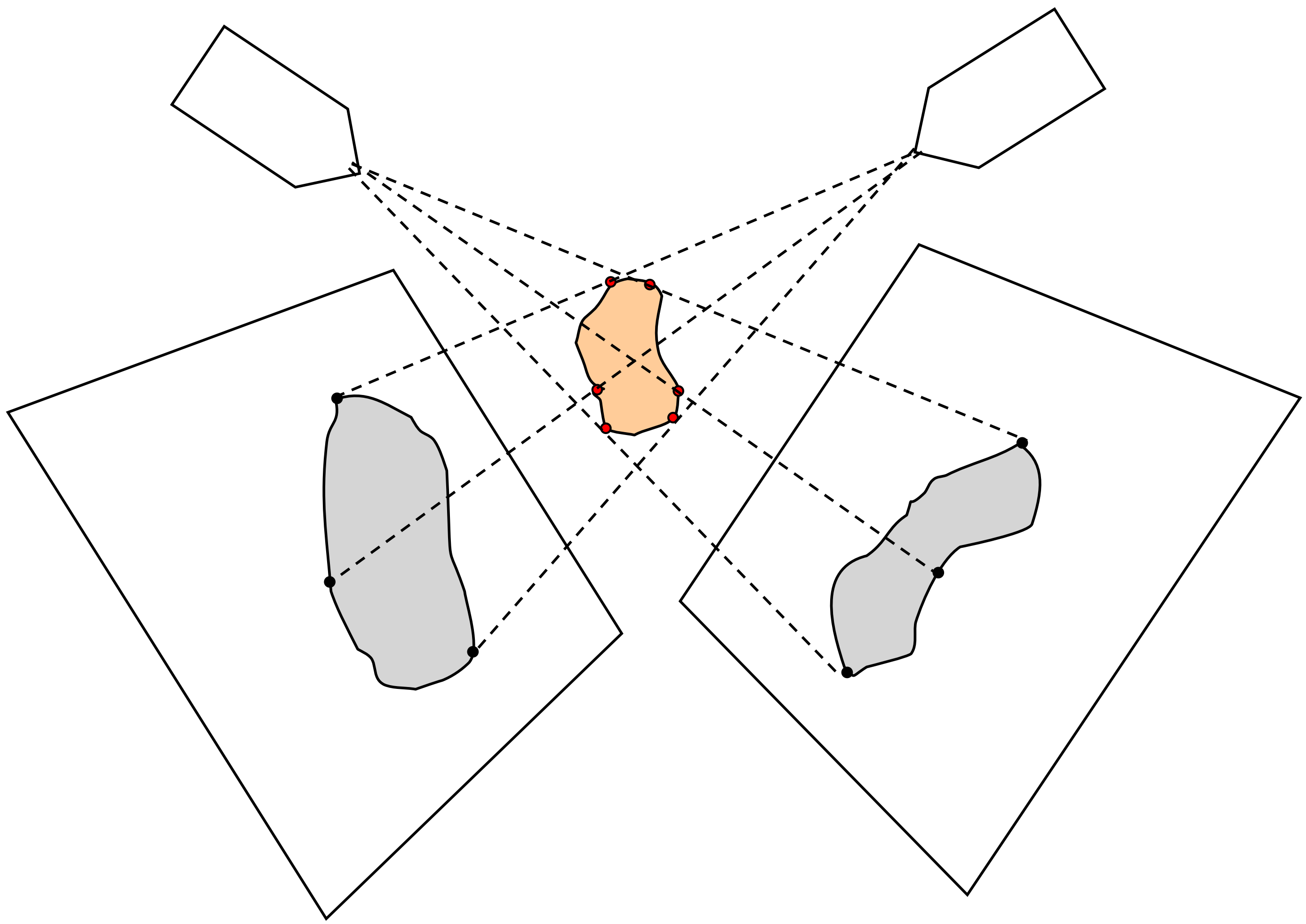
Feature-Based 2D-3D Registration



Feature-Based 2D-3D Registration



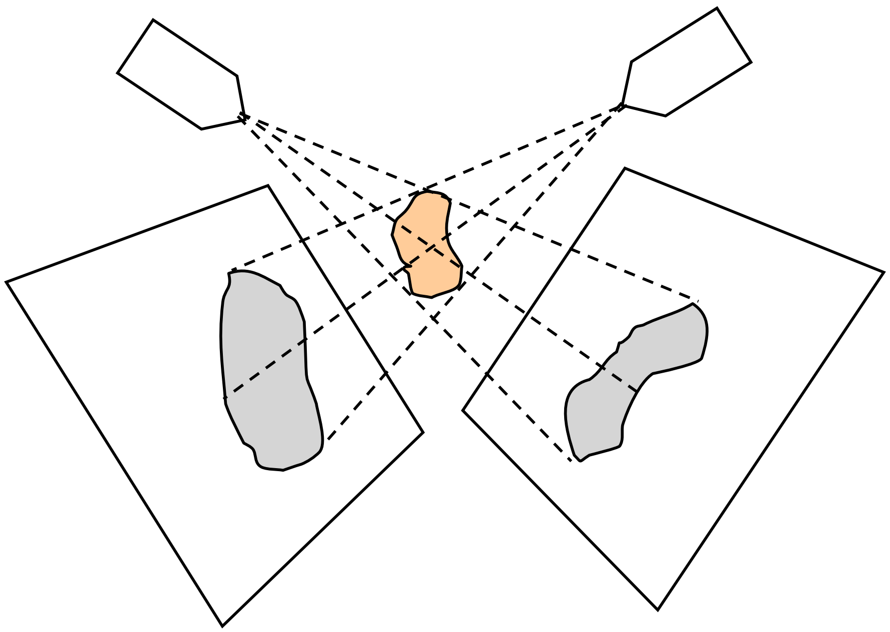
Feature-Based 2D-3D Registration

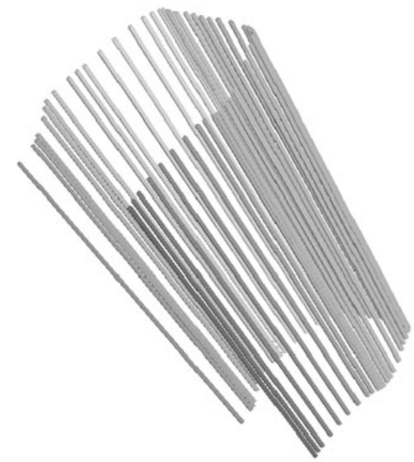


A contour-based 2D-3D method …

Gueziec *et al., 1998*

Step 0: Extract contours from x-ray images and compute corresponding lines between source and detector





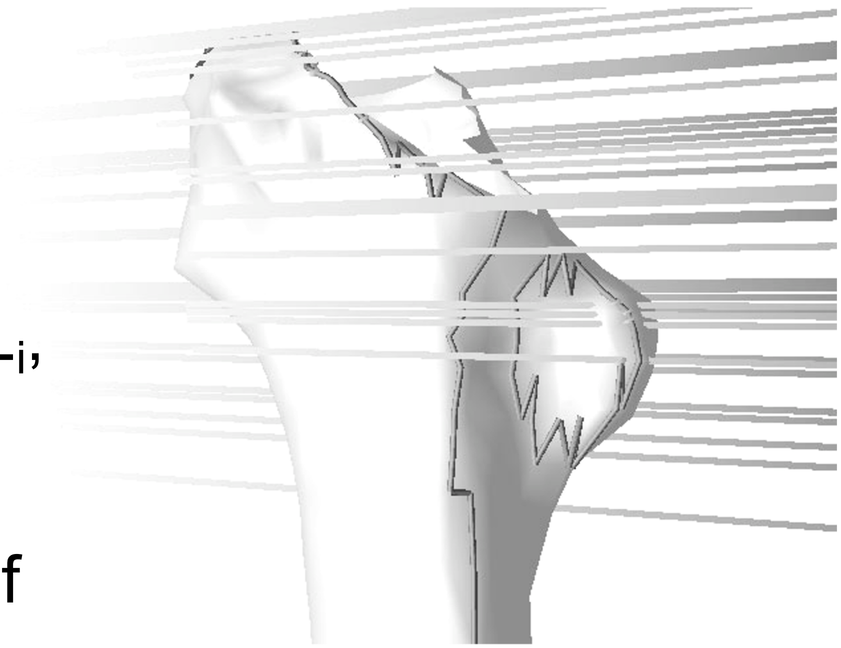
A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

A contour-based 2D-3D method …

Gueziec *et al., 1998*

Step 1: Given the current estimate for $F = [R,t]$, compute the apparent projection contours of the model for each viewing direction.

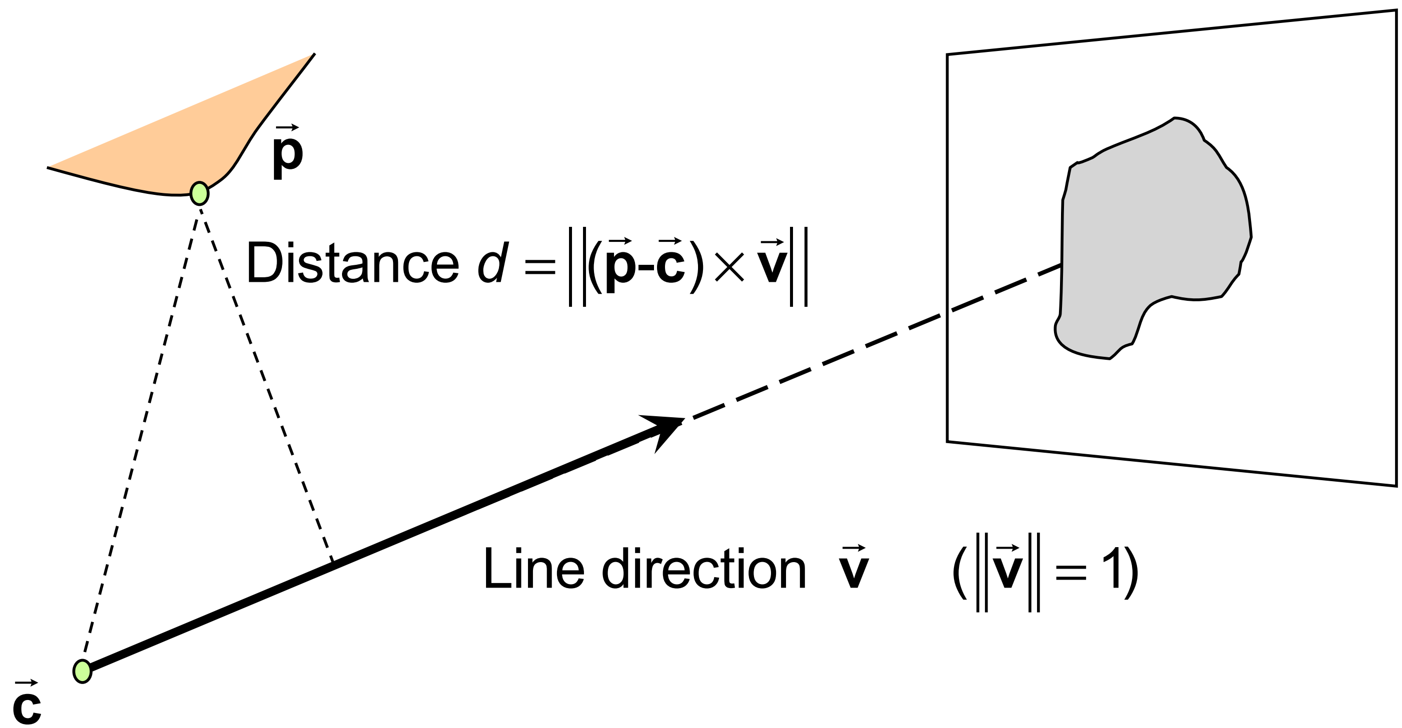
Step 2: For each x-ray path line line L_i, identify the closest point p_i on an apparent projection contour. This will give a set of points on the body surface to be moved toward the corresponding x-ray lines



A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

A contour-based 2D-3D method …

Gueziec *et al., 1998*

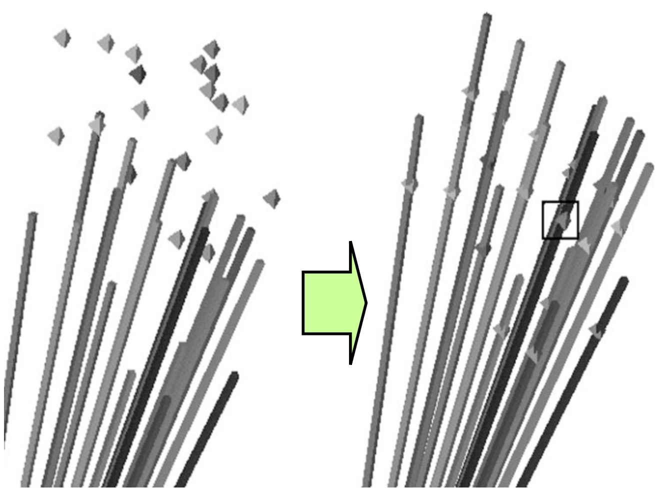


Note: It is convenient to use the x-ray source position (i.e., the center of convergence for a bundle of x-ray projection lines) as the value for $\vec{\mathbf{c}}$.

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

A contour-based 2D-3D method …

Gueziec *et al., 1998*



Step 3: Solve an optimization problem to compute a value of F that minimizes the distance between the p_i and the L_i.

$$\min_{\mathbf{R},\vec{\mathbf{t}}} \sum_i {d_i}^2 = \min_{\mathbf{R},\vec{\mathbf{t}}} \sum_i \left\| \vec{\mathbf{v}}_i \times \left(\mathbf{c}_i - \left(\mathbf{R}\vec{\mathbf{p}}_i + \vec{\mathbf{t}} \right) \right) \right\|^2$$

$$= \min_{\mathbf{R},\vec{\mathbf{t}}} \sum_i \left\| skew\left(\vec{\mathbf{v}}_i \right) \bullet \left(\mathbf{c}_i - \left(\mathbf{R}\vec{\mathbf{p}}_i + \vec{\mathbf{t}} \right) \right) \right\|^2$$

Step 4: Iterate steps 1-3 until reach convergence

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

Computational Note

Gueziec uses the Cayley parameterization for rotations:

$$\mathbf{R}(\vec{\mathbf{u}})=\left(\mathbf{I}\text{-skew}(\vec{\mathbf{u}})\right)\left(\mathbf{I}\text{+skew}(\vec{\mathbf{u}})\right)^{-1}$$

This leads to the approximation

$$\mathbf{R}(\vec{\mathbf{u}}) \approx \mathbf{I}\text{+skew}(2\vec{\mathbf{u}})$$

which is similar to our familiar $\mathbf{R}(\vec{\alpha}) \approx \mathbf{I}\text{+skew}(\vec{\alpha})$.

He also uses the notation $\mathbf{U}$=skew($\vec{\mathbf{u}}$). So $\mathbf{R}(\vec{\mathbf{u}}) = (\mathbf{I}-\mathbf{U})(\mathbf{I}+\mathbf{U})^{-1}$

Similarly, we will see $\mathbf{V}$=skew($\vec{\mathbf{v}}$), etc.

A countour-based 2D-3D method …

Gueziec et al., 1998

Gueziec compared three different methods for performing the minimization in Step 3:

- Levenberg Marquardt (LM) nonlinear minimization.
- Linearization and constrained minimization
- Use of a Robust M-Estimator

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

Levenberg-Marquardt …
(Following development in Gueziec et al., 1998)

Define $f_i(\vec{x}) = \left\| \mathbf{V}_i \left(\vec{\mathbf{c}}_i - \mathbf{R}(\vec{\mathbf{u}})\vec{\mathbf{p}}_i - \vec{\mathbf{t}} \right) \right\|$ where $\vec{x}^t = [\vec{\mathbf{u}}^t, \vec{\mathbf{t}}^t], \mathbf{V}_i = skew(\vec{\mathbf{v}}_i)$

Our goal is to minimize

$$\varepsilon(\vec{x}) = \sum_i f_i(\vec{x})^2 = \sum_i \left\| \mathbf{V}_i \left(\vec{\mathbf{c}}_i - \mathbf{R}(\vec{\mathbf{u}})\vec{\mathbf{p}}_i - \vec{\mathbf{t}} \right) \right\|^2$$

We note that $\varepsilon(\vec{x})$ is nonlinear. Levenberg-Marquardt is a widely used optimization method for problems of this type. However, it requires us to evaluate the partial derivitives $\partial f_i / \partial x_j$. Gueziec worked these out symbolically for his problem

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

Levenberg-Marquardt …

(Following development in Gueziec et al., 1998)

Define $f_i(\vec{x}) = \left\| \mathbf{V}_i \left(\vec{\mathbf{c}}_i - \mathbf{R}(\vec{\mathbf{u}})\vec{\mathbf{p}}_i - \vec{\mathbf{t}} \right) \right\|$ where $\vec{x}^t = [\vec{\mathbf{u}}^t, \vec{\mathbf{t}}^t], \mathbf{V}_i = skew(\vec{\mathbf{v}}_i)$

$$\mathbf{J} = \begin{bmatrix} \cdots & \dfrac{\partial f_i}{\partial \vec{x}} & \cdots \end{bmatrix} = \begin{bmatrix} \cdots & \begin{matrix} \dfrac{\partial f_i}{\partial \vec{\mathbf{u}}} \\ \dfrac{\partial f_i}{\partial \vec{\mathbf{t}}} \end{matrix} & \cdots \end{bmatrix}$$

$$\frac{\partial f_i}{\partial \vec{\mathbf{t}}} = \frac{\mathbf{V}_i^t \mathbf{V}_i (\mathbf{R}\vec{\mathbf{p}}_i - \mathbf{c} + \vec{\mathbf{t}})}{f_i}$$

$$\frac{\partial f_i}{\partial \vec{\mathbf{u}}} = \left(\frac{\partial \mathbf{R}\vec{\mathbf{p}}_i}{\partial \vec{\mathbf{u}}} \right)^t \frac{\mathbf{V}_i^t \mathbf{V}_i (\mathbf{R}\vec{\mathbf{p}}_i - \mathbf{c} + \vec{\mathbf{t}})}{f_i}$$

Details on this may be found in reference [45] of Gueziec's paper

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

Levenberg-Marquardt …

(Following development in Gueziec et al., 1998)

Step 1: Pick λ = a small number; pick initial guess for $\vec{x}$

Step 2: Evaluate $f_i(\vec{x})$ and $\mathbf{J}$ and solve the least squares problem

$$\begin{bmatrix} \vdots \\ (\mathbf{J}^t\mathbf{J} + \lambda\mathbf{I})\Delta\vec{x} - \mathbf{J}^t f_i \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ 0 \\ \vdots \end{bmatrix}$$

for $\Delta\vec{x}$.

Step 3: $\vec{x} \leftarrow \vec{x} + \Delta\vec{x}$; update λ.

Step 4: Evaluate termination condition. If not done, go back to to step 2

Note: Usually λ starts small and grows larger. Consult standard references (e.g., Numerical Recipes) for more information.

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

Constrained Linearized Least Squares …

(Following development in Gueziec et al., 1998)

Step 0: Make an initial guess for $\mathbf{R}$ and $\vec{\mathbf{t}}$

Step 1: Compute $\vec{\mathbf{p}}_{\mathrm{i}} \leftarrow \mathbf{R}\vec{\mathbf{p}}_i + \vec{\mathbf{t}}$

Step 2: Define $\mathbf{P}_{\mathrm{i}} = skew(\vec{\mathbf{p}}_i)$, $\mathbf{V}_{\mathrm{i}} = skew(\vec{\mathbf{v}}_i)$

Step 3: Solve the least squares problem:

$$\varepsilon^2 = \min \left\| \begin{bmatrix} \vdots & \vdots \\ 2\mathbf{V}_{\mathrm{i}}\mathbf{P}_i & \mathbf{V}_i \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vec{\mathbf{u}} \\ \Delta\vec{\mathbf{t}} \end{bmatrix} - \begin{bmatrix} \vdots \\ \mathbf{V}_i(\vec{\mathbf{c}}_i - \vec{\mathbf{p}}_i) \\ \vdots \end{bmatrix} \right\|^2 \quad \text{subject to } \left\| \vec{\mathbf{u}} \right\| \leq \rho$$

where ρ is sufficiently small so that **I**+2**U** approximates a rotation

Step 4: Compute $\Delta\mathbf{R} = (\mathbf{I} - \mathbf{U})(\mathbf{I} + \mathbf{U})^{-1}$

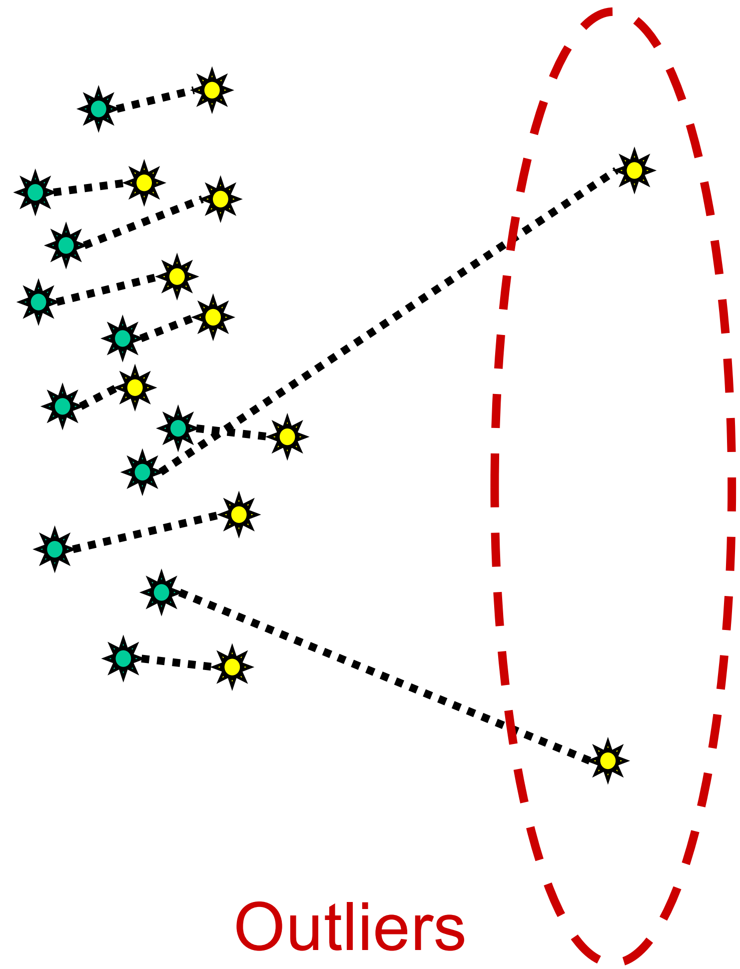
Update $\mathbf{p}_i \leftarrow \Delta\mathbf{R}\mathbf{p}_i + \Delta\vec{\mathbf{t}}$; $\mathbf{R} \leftarrow \Delta\mathbf{R}\mathbf{R}$; $\vec{\mathbf{t}} \leftarrow \Delta\mathbf{R}\vec{\mathbf{t}} + \Delta\vec{\mathbf{t}}$

Step 5: If ε is small enough or some othe termination condition is met, then stop. Otherwise go back to Step 2.

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

Robust Pose Estimation …

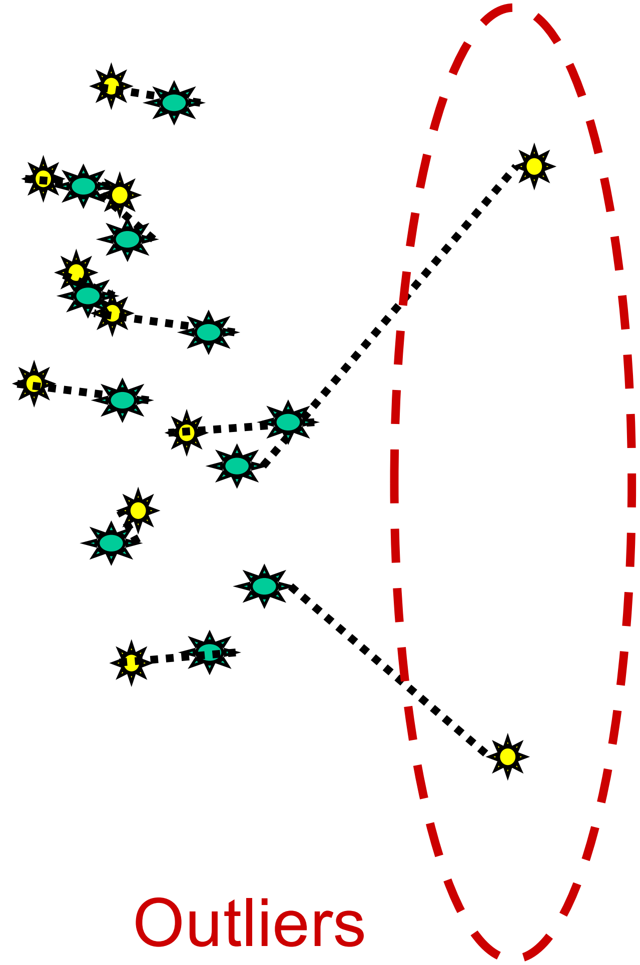
- Basic idea is to identify outliers and give them little or no weight.



R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," Comput. Vision, Graphics, Image Processing-IU, vol. 60, no. 3, pp. 313–342, 1994.

Robust Pose Estimation …

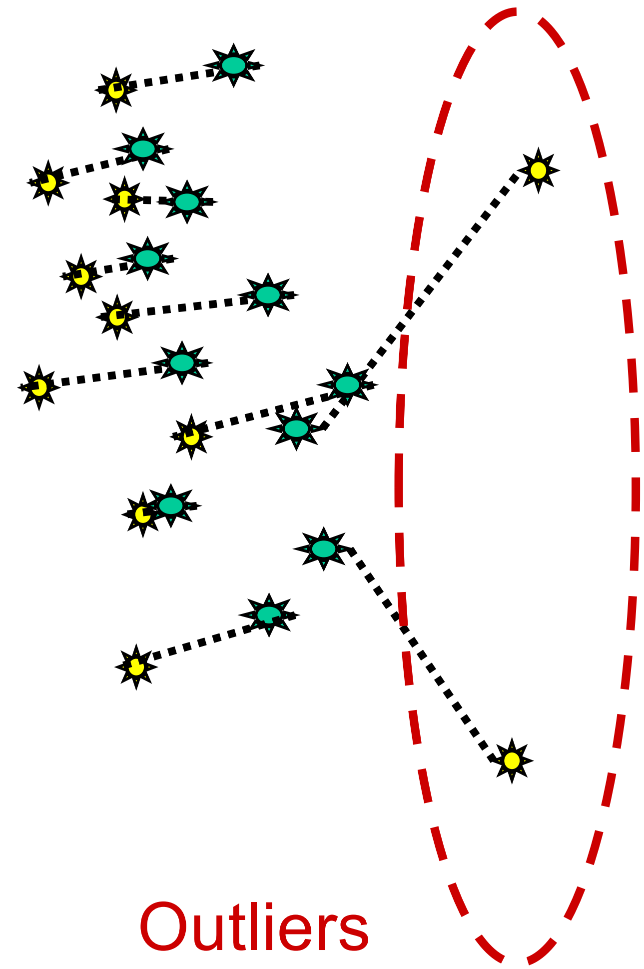
- Basic idea is to identify outliers and give them little or no weight.



R. Kumar and A. R. Hanson, “Robust methods for estimating pose and a sensitivity analysis,” Comput. Vision, Graphics, Image Processing-IU, vol. 60, no. 3, pp. 313–342, 1994.

Robust Pose Estimation …

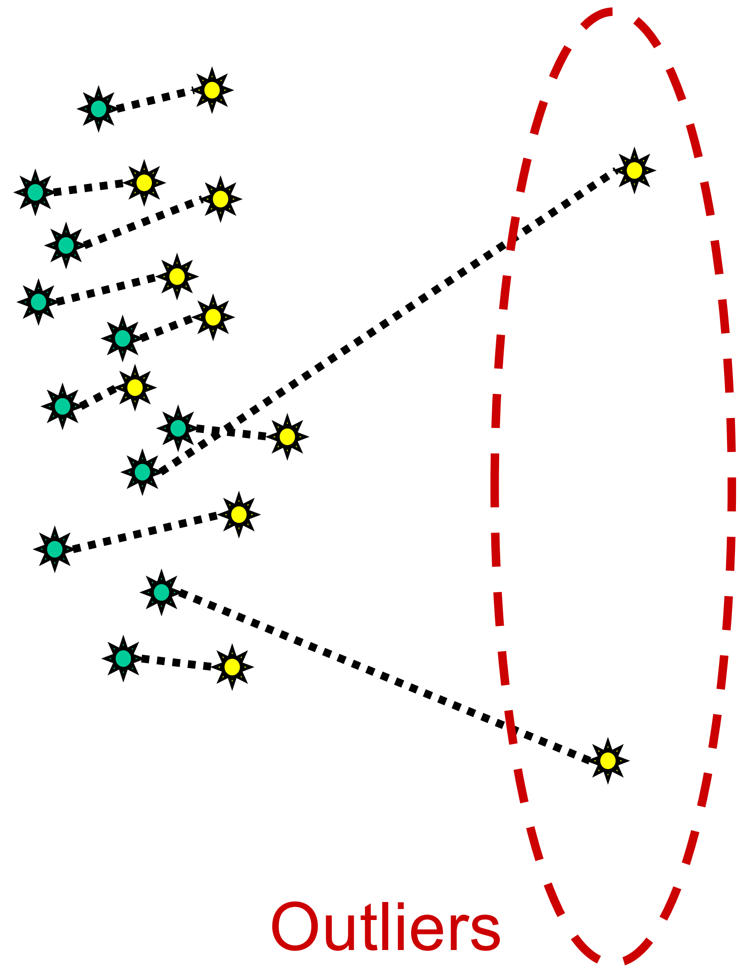
- Basic idea is to identify outliers and give them little or no weight.



R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," Comput. Vision, Graphics, Image Processing-IU, vol. 60, no. 3, pp. 313–342, 1994.

Robust Pose Estimation …

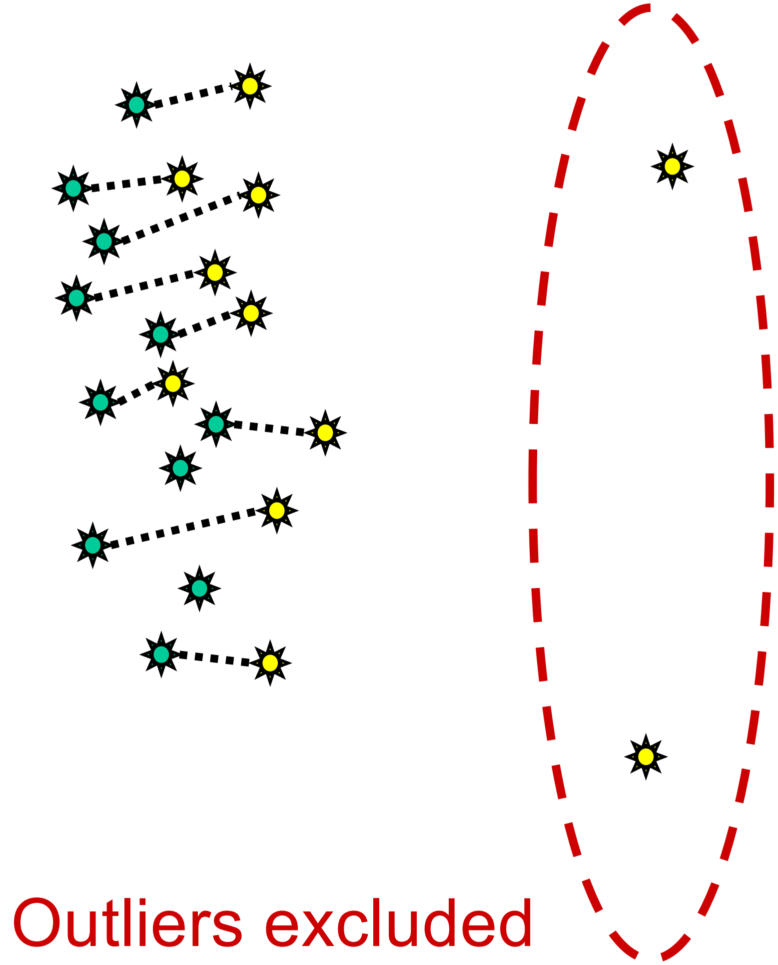
- Basic idea is to identify outliers and give them little or no weight.



R. Kumar and A. R. Hanson, “Robust methods for estimating pose and a sensitivity analysis,” Comput. Vision, Graphics, Image Processing-IU, vol. 60, no. 3, pp. 313–342, 1994.

Robust Pose Estimation …

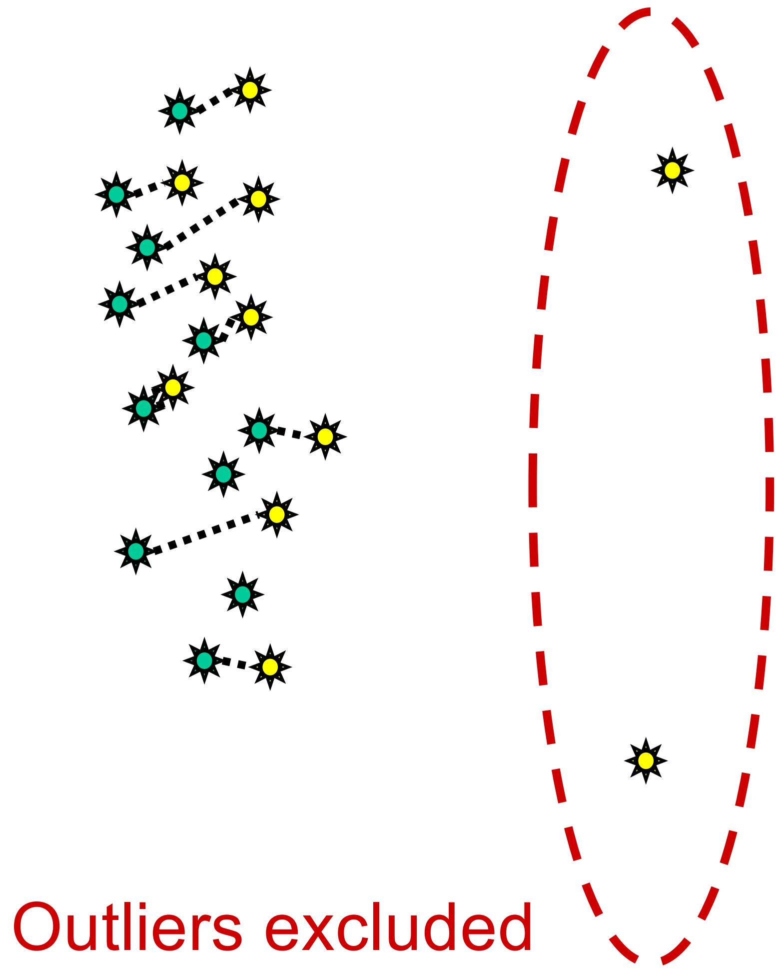
- Basic idea is to identify outliers and give them little or no weight.



R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," Comput. Vision, Graphics, Image Processing-IU, vol. 60, no. 3, pp. 313–342, 1994.

Robust Pose Estimation …

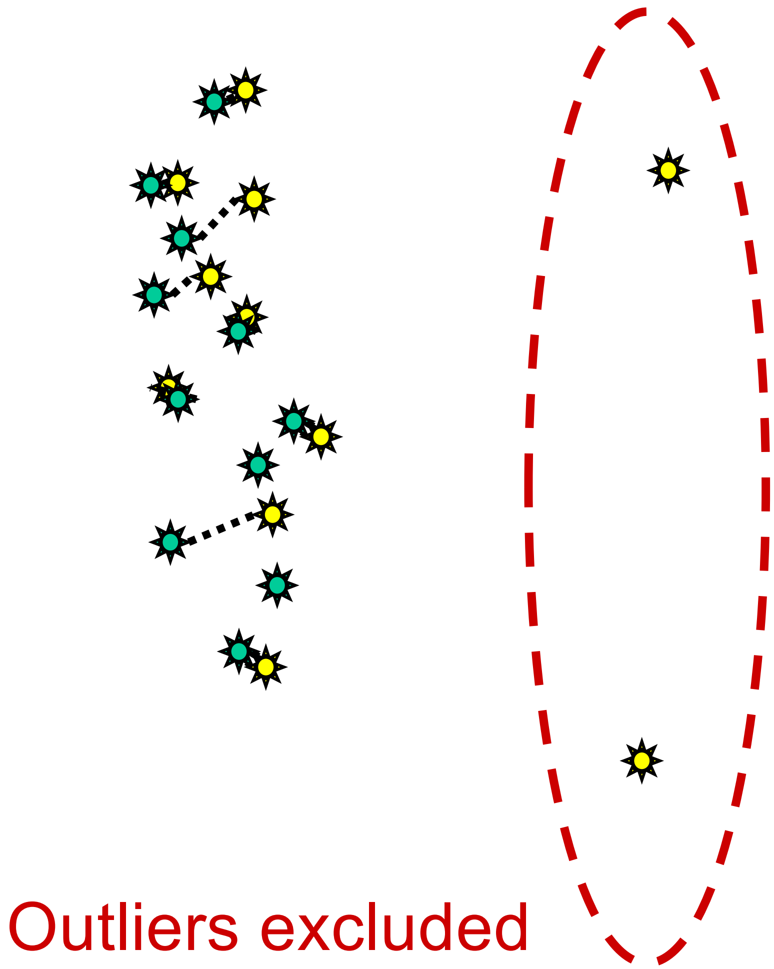
- Basic idea is to identify outliers and give them little or no weight.



R. Kumar and A. R. Hanson, “Robust methods for estimating pose and a sensitivity analysis,” Comput. Vision, Graphics, Image Processing-IU, vol. 60, no. 3, pp. 313–342, 1994.

Robust Pose Estimation …

- Basic idea is to identify outliers and give them little or no weight.



R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," Comput. Vision, Graphics, Image Processing-IU, vol. 60, no. 3, pp. 313–342, 1994.

Robust M-Estimator …
(Following development in Gueziec et al., 1998)

Step 0: Make an initial guess for $\mathbf{R}$ and $\vec{\mathbf{t}}$

Step 1: Compute $\vec{\mathbf{p}}_i \leftarrow \mathbf{R}\vec{\mathbf{p}}_i + \vec{\mathbf{t}}$

Step 2: Define $\mathbf{P}_i = skew(\vec{\mathbf{p}}_i)$, $\mathbf{V}_i = skew(\vec{\mathbf{v}}_i)$,

Step 3: Solve a robust linearized problem

$$\varepsilon = \underset{\vec{\mathbf{u}},\Delta\vec{\mathbf{t}}}{\arg\min} \sum_i \rho\left(\frac{0.6745\ e_i}{median(\{e_i\})}\right) \quad \text{where } e_i = \left\| \mathbf{V}_i(\vec{\mathbf{p}}_i - \mathbf{c}_i + 2\mathbf{P}_i\vec{\mathbf{u}} + \Delta\vec{\mathbf{t}}) \right\|$$

(See next slide)

Step 4: Compute $\Delta\mathbf{R} = (\mathbf{I} - \mathbf{U})(\mathbf{I} + \mathbf{U})^{-1}$

Update $\mathbf{p}_i \leftarrow \Delta\mathbf{R}\mathbf{p}_i + \Delta\vec{\mathbf{t}}$; $\mathbf{R} \leftarrow \Delta\mathbf{R}\mathbf{R}$; $\vec{\mathbf{t}} \leftarrow \Delta\mathbf{R}\vec{\mathbf{t}} + \Delta\vec{\mathbf{t}}$

Step 5: If ε is small enough or some othe termination condition is met, then stop. Otherwise go back to Step 2.

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

Robust M-Estimator …

(Following development in Gueziec et al., 1998)

Step 3.0: Set $\vec{\mathbf{u}} = \vec{\mathbf{0}}$, $\Delta\vec{\mathbf{t}} = \vec{\mathbf{0}}$

Step 3.1: Compute $e_i = \left\| \mathbf{V}_i(\vec{\mathbf{p}}_i - \vec{\mathbf{c}}_i + 2P_i\vec{\mathbf{u}} + \Delta\vec{\mathbf{t}}) \right\|$, $s = median(\{\cdots, e_i, \cdots\}) / 0.6745$,

Step 3.2: Solve $\mathbf{C}\vec{\mathrm{x}} = \vec{\mathbf{d}}$, where $\vec{\mathrm{x}}^t = [\vec{\mathbf{u}}^t, \vec{\mathbf{t}}^t]$

$$\mathbf{C} = \sum_i \Psi\left(\frac{e_i}{s}\right) \begin{bmatrix} 2\mathbf{P}_i\mathbf{W}_i\mathbf{P}_i & \mathbf{P}_i\mathbf{W}_i \\ 2\mathbf{P}_i\mathbf{W}_i & \mathbf{W}_i \end{bmatrix} \text{ and } \vec{\mathbf{d}} = \sum_i \Psi\left(\frac{e_i}{s}\right) \begin{bmatrix} \mathbf{P}_i\mathbf{W}_i(\vec{\mathbf{c}}_i - \vec{\mathbf{p}}_i) \\ \mathbf{W}_i(\vec{\mathbf{c}}_i - \vec{\mathbf{p}}_i) \end{bmatrix}$$

$$\text{where } \mathbf{W}_i = \mathbf{V}_i^t\mathbf{V}_i = \mathbf{I} - \vec{\mathbf{v}}_i\mathbf{v}_i^t \quad \Psi(\mu) = \begin{cases} \mu\left(1 - \mu^2/\alpha^2\right)^2 & \text{if } \|\mu\| \le \alpha \\ 0 & \text{otherwise} \end{cases}$$

(**Note**: We use α=2)

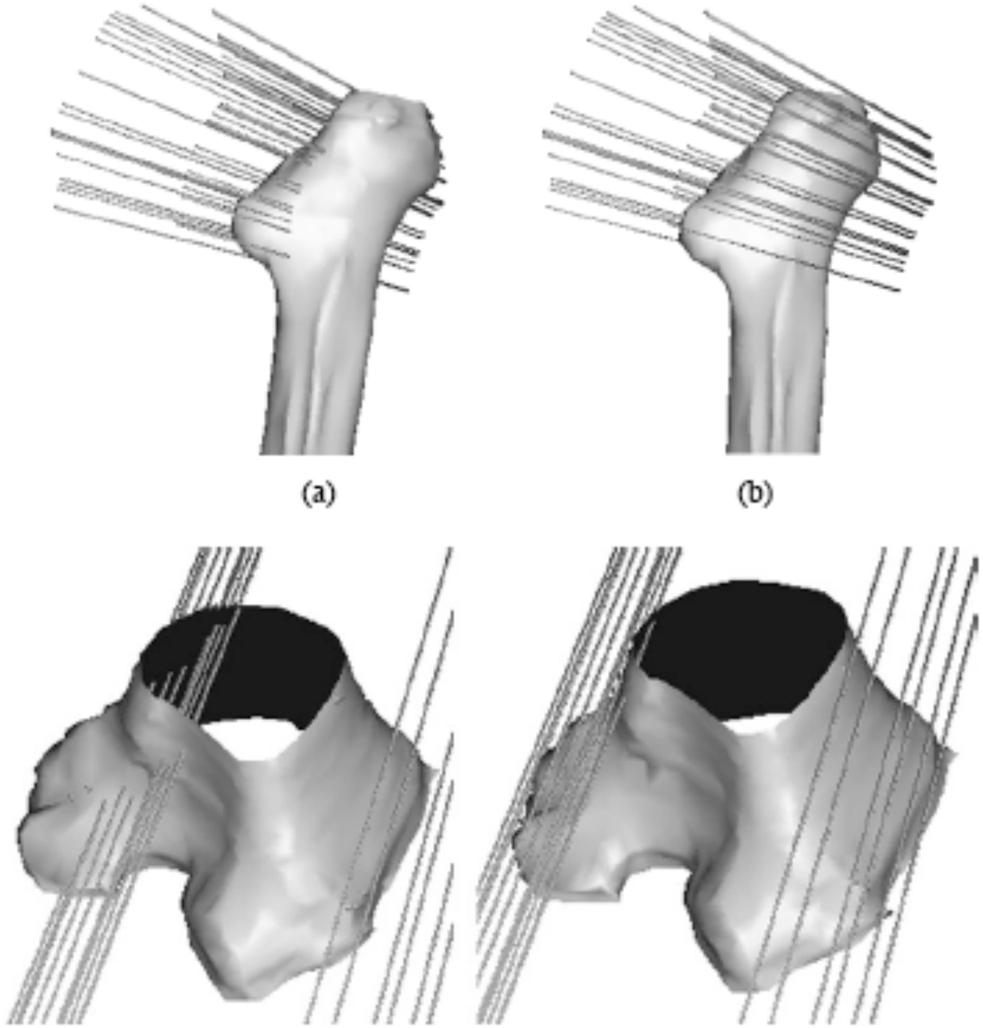
Step 3.3: Iterate steps 3.1 and 3.2 until a suitable termination condition is reached.

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

A countour-based 2D-3D method … results

Gueziec *et al., 1998*

Before



(a) (b)

(c) (d)

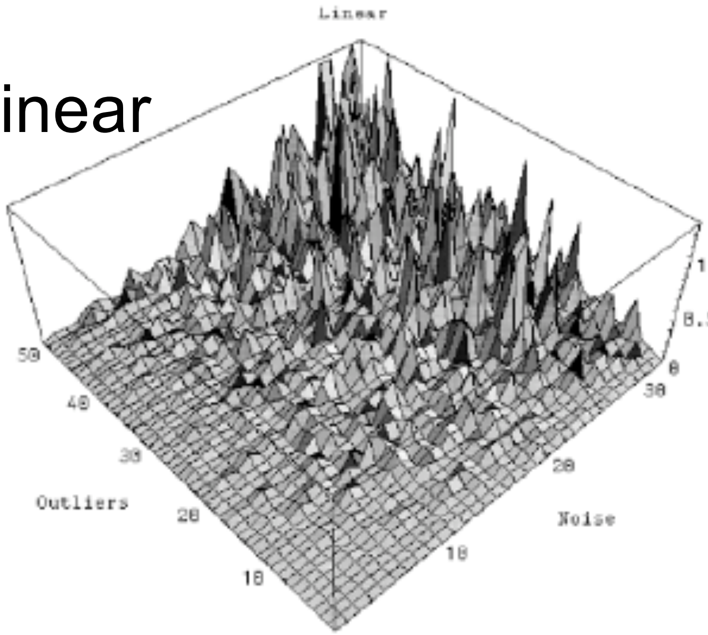
After

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

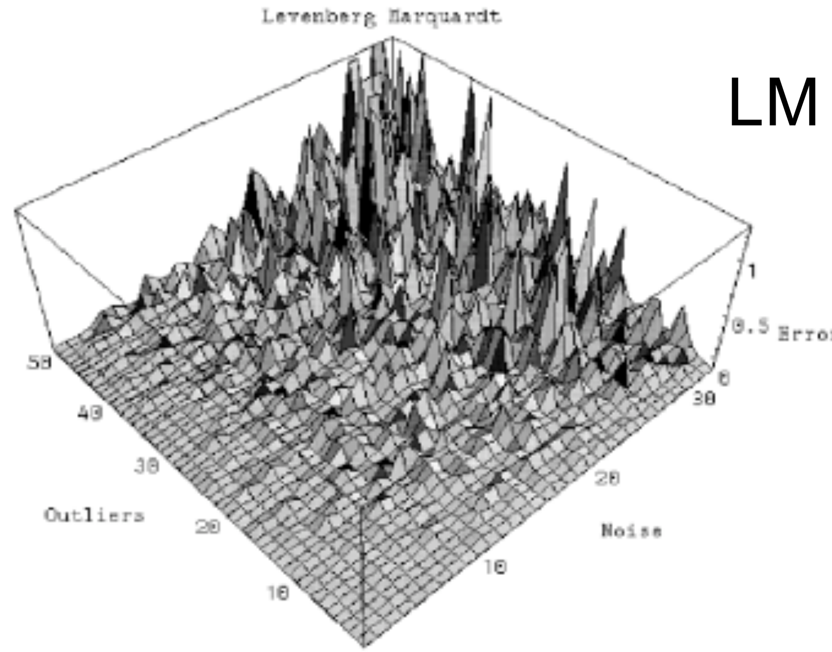
A countour-based 2D-3D method … results

Gueziec *et al., 1998*

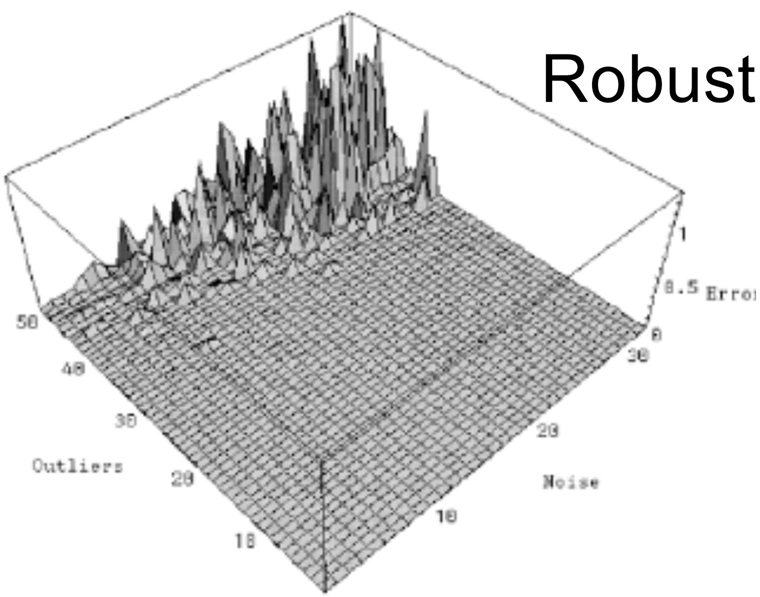
Linear



LM



Robust



Error *vs* noise and outliers

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

A contour-based 2D-3D method … times

Gueziec *et al., 1998*

TABLE I

AVERAGE EXECTUTION TIMES IN MS FOR THE THREE REGISTRATION METHODS APPLIED TO DATA SETS THAT COMPRISE 100 POINTS (TOP) AND 20 POINTS (BOTTOM)

Number Points/Method	LM	Linear	Robust
100 points (CPU time)	790	690	28
20 points (CPU time)	200	42	9.6

A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

Sample Set Analysis

- **Question:** How good is a particular set of 3D sample points for the purpose of registration to a 3D surface?

- Long line of authors have looked at this question
- Next few slides are based on the work of David Simon, et al (1995)

Sample Set Analysis: Distance Estimates

Let

$$F(\mathbf{x}) = 0$$

be the implicit equation of a surface, then one good estimate of the distance of a point $\mathbf{x}$ to the surface is

$$D(\mathbf{x}) = \frac{F(\mathbf{x})}{\|\nabla F(\mathbf{x})\|}$$

Sample set analysis: sensitivity

Let $\mathbf{x}_s$ be a point on the surface, and let $T(\overline{\eta})$ represent a small perturbation with parameters $\overline{\eta}$ with respect to the surface of point $\mathbf{x}_s$:

$$\mathbf{x}'_s = T(\overline{\eta})\mathbf{x}_s$$

Then we define $\mathbf{V}(\mathbf{x}_s)$ to be

$$\mathbf{V}(\mathbf{x}_s) = \frac{\partial D(T(\overline{\eta})\mathbf{x}_s)}{\partial \overline{\eta}} = \begin{bmatrix} \mathbf{n}_s \\ \mathbf{x}_s \times \mathbf{n}_s \end{bmatrix}$$

where $\mathbf{n}_s$ is the unit normal to the surface at $\mathbf{x}_s$. So,

$$D(\mathbf{T}(\overline{\eta})\mathbf{x}_s)) \simeq \mathbf{V}^T(\mathbf{x}_s)\overline{\eta}$$

Squaring this gives

$$\begin{aligned} D^2(\mathbf{T}(\overline{\eta})\mathbf{x}_s)) &\simeq \overline{\eta}_T \mathbf{V}(\mathbf{x}_s)\mathbf{V}^T(\mathbf{x}_s)\overline{\eta} \\ &= \overline{\eta}^T \mathbf{M}(\mathbf{x}_s)\overline{\eta} \end{aligned}$$

Note that $\mathbf{M}$ is 6×6 positive, semi-definite, symmetric matrix.

Sample set analysis: sensitivity

For a region $\mathcal{R}$, define

$$
\begin{aligned}
E_R(\overline{\eta}) &= \overline{\eta}^T \left[\sum_{\mathbf{x}_s \in \mathcal{R}} \mathbf{M}(\mathbf{x}_s) \right] \overline{\eta} \\
&= \overline{\eta}^T \mathbf{\Psi}_{\mathcal{R}} \overline{\eta} \\
&= \overline{\eta}^T \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T \overline{\eta} \\
&= \sum_{1 \le i \le 6} \lambda_i \left(\overline{\eta}^T \cdot \mathbf{q}_i \right)^2
\end{aligned}
$$

- Note that the eigenvectors $\mathbf{q}_i$ correspond to small differential transformations $\mathbf{T}(\mathbf{q}_i)$. and can sort eigenvalues so that

$$\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_6$$

- Note that eigenvector $\mathbf{q}_1$ corresponds to direction of greatest constraint.
- Similarly, can also think of $\mathbf{q}_6$ as the least constrained direction.

Sample Set Analysis: Goodness Measures

- Magnitude of smallest eigenvalue (Simon)
- (Kim and Khosla)

$$\frac{\sqrt[6]{\lambda_1 \cdot \ldots \cdot \lambda_6}}{\lambda_1 + \ldots + \lambda_6}$$

- Nahvi

$$\frac{\lambda_6^2}{\lambda_1}$$

Sample Set Selection

- One blind search method (similar to Simon, 1995) is:
  - Randomly select sample points on surface
  - (prune for reachability)
  - evaluate goodness of sample set using some criterion
  - repeat many times and choose the best one found

Sample Set Selection

- Refinement of blind search (hill climbing):
  - Randomly select sample points on surface
  - (prune for reachability)
  - evaluate goodness of sample set using some criterion
  - replace a point from sample set with a randomly selected point
  - evaluate goodness
  - if better, keep it
  - else revert to original point and try again
- Variations include simulated annealing, “genetic” algorithms

Sample Set Selection: Another Alternative

- Select large number of random points $\mathbf{x}_s$
- Prune for reachability
- For each point, compute constraint direction $\mathbf{V}_s = \mathbf{V}(\mathbf{x}_s)$. To a first approximation, a measurement at $\mathbf{x}_s$ with accuracy ϵ_s constrains $\overline{\eta}$ by

$$|\mathbf{V}_s\overline{\eta}| \leq \epsilon_s$$

- Now select subset of the $\mathbf{x}_s$ that minimizes, e.g.,

$$\min_{\delta_s} \max \overline{\eta}^T \mathbf{S} \overline{\eta}$$

subject to

$$\begin{aligned} \{\delta_s &\in \{0,1\} \\ |\delta_s \mathbf{V}_s \overline{\eta}| &\leq \epsilon_s \\ \sum_s \delta_s &\leq \text{subsetsize} \end{aligned}$$

There are various ways to do this.

Sample Set Selection: Another Alternative (con'd)

- One can also minimize other forms, e.g.,

$$\min_s \max_i |\sigma_i \eta_i|$$

  subject to similar constraints

- An alternative is to minimize the number of sample points required to ensure that some constraints on $\overline{\eta}$ are guaranteed to be met. E.g.,

$$\min_{\delta_s} \sum \delta_s$$

  such that

$$\begin{aligned} \delta_s &\in \{0,1\} \\ \xi &\leq \xi_{limit} \end{aligned}$$

  where

$$\xi = \max_{\overline{\eta}} \overline{\eta}^T \mathbf{S} \overline{\eta}$$

  or some other form subject to

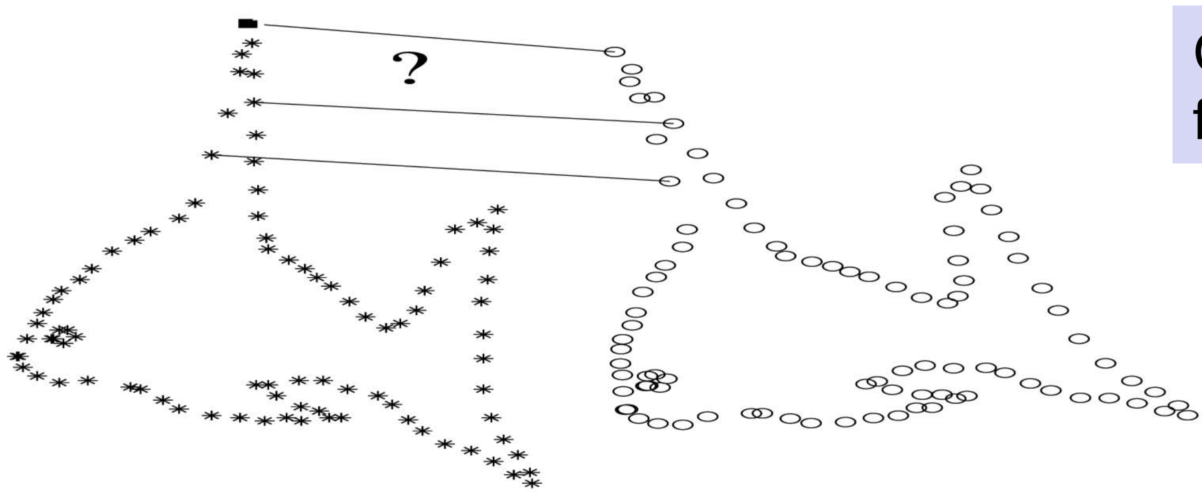
$$|\delta_s \mathbf{V}_s \overline{\eta}| \leq \epsilon_s$$

Probabilistic Registration

- Registration methods typically use some optimization algorithm to find a “best” transformation between one data set and the other.
- It makes sense to try to find the “most likely” registration transformation.
- ICP minimizes sum-of-squares distances.
- This is equivalent to assuming that point-pair match probabilities are independent and symmetric Gaussian distributions based on distances
- But there are a number of other methods that explicitly consider probabilities …

Coherent Point Drift

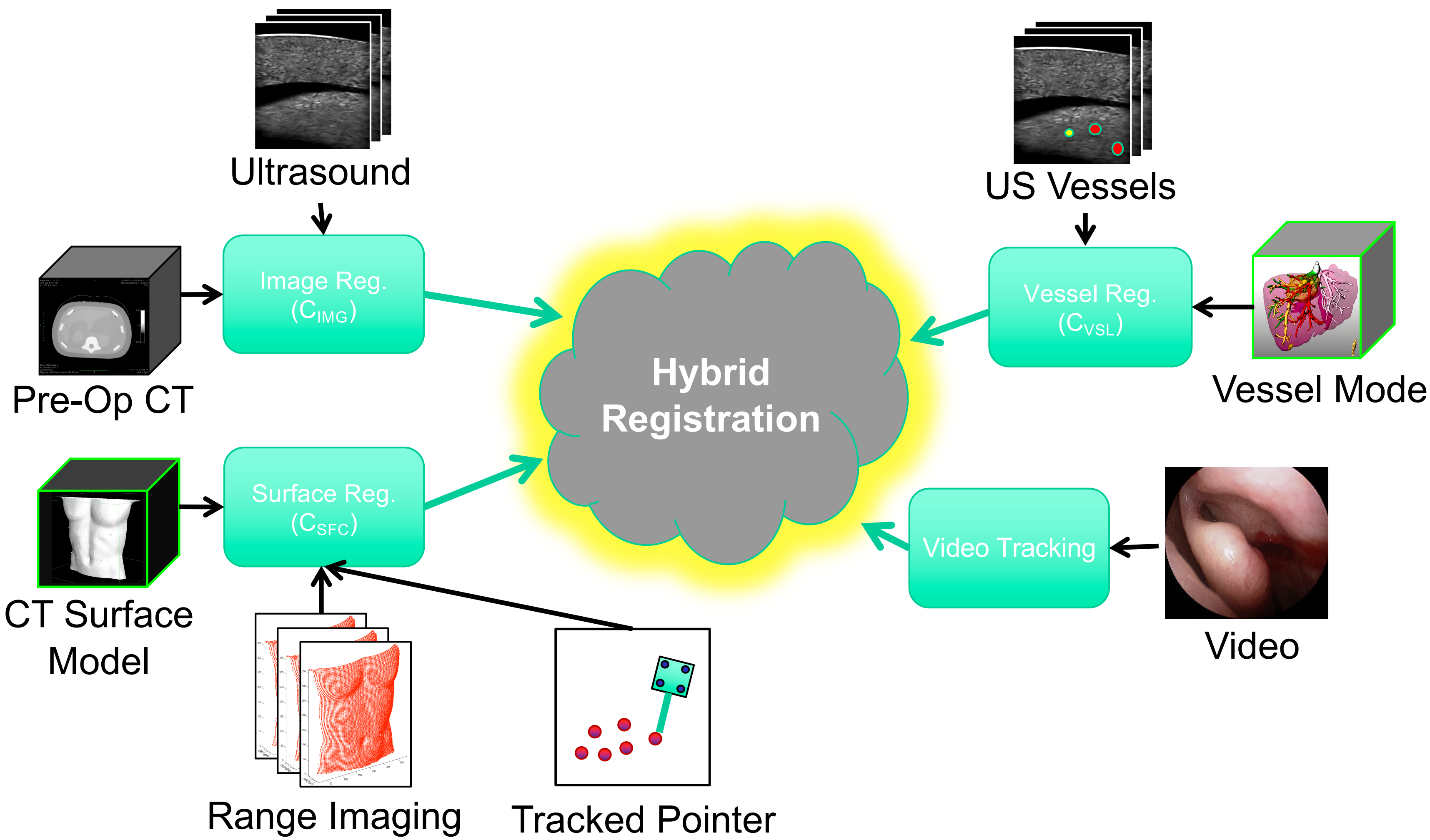
- A. Myronenko and X. Song, "Point-Set Registration: Coherent Point Drift", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 32- 12, pp. 2262-2275, 2010.
- Alignment of point clouds
  - Fast method follows "EM" paradigm
  - Tolerates outliers and noise
  - Transformations: Rigid, affine, general deformable



Click here for slides

Multi-Modal Feature-Based Registration

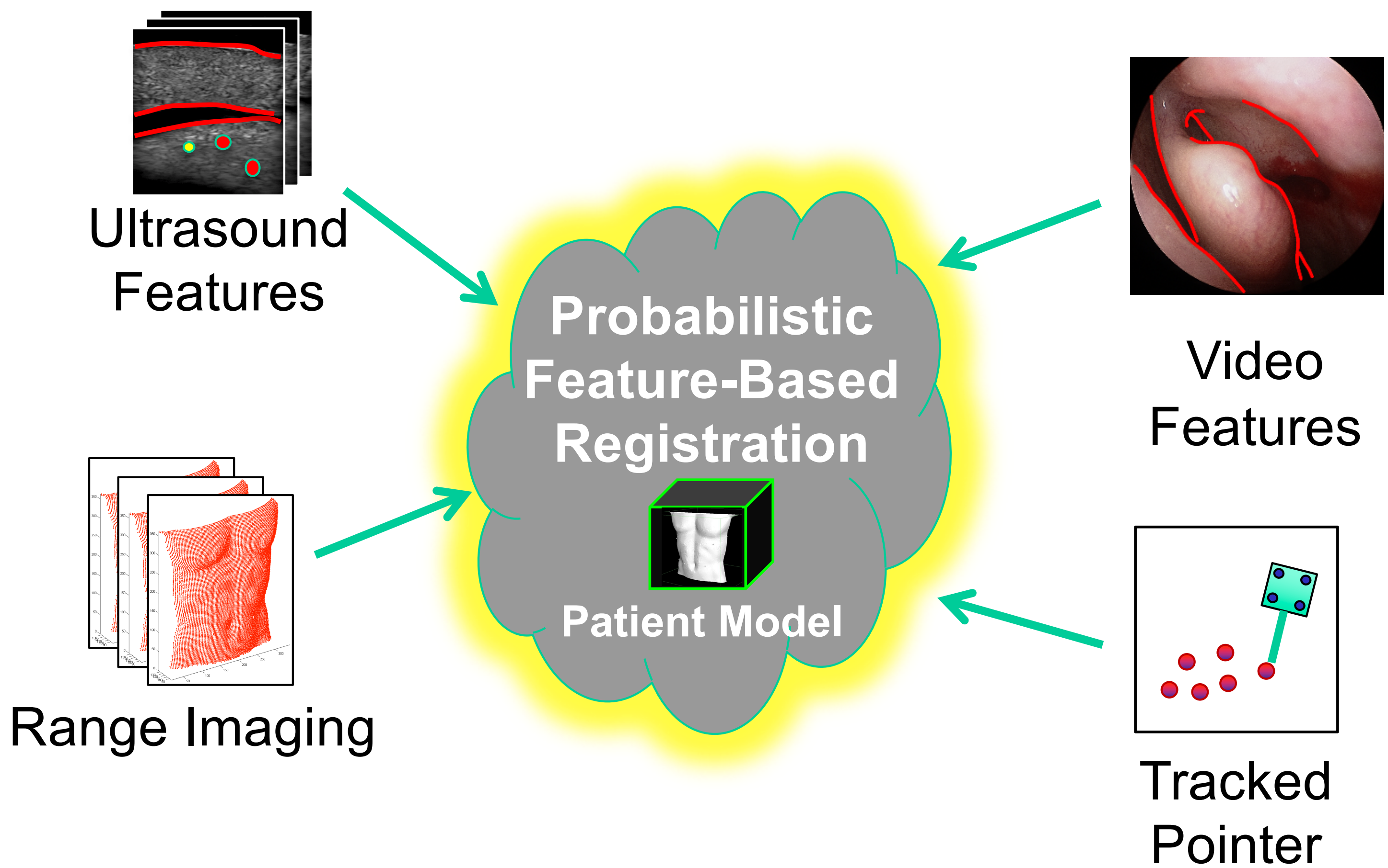
Question: How to combine multiple data sources, in order to improve the accuracy and robustness of registration outcomes?



Billings S, Kapoor A, Keil M, Wood BJ, Boctor E (2011) A Hybrid Surface/Image-Based Approach to Facilitate Ultrasound/CT Registration. In: *SPIE, Medical Imaging 2011: Ultrasonic Imaging, Tomography, and Therapy*

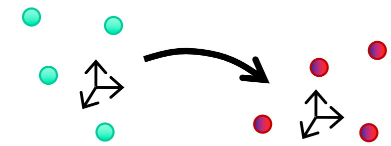
Multi-Modal Feature-Based Registration

Question: How to combine multiple data sources, in order to improve the accuracy and robustness of registration outcomes?

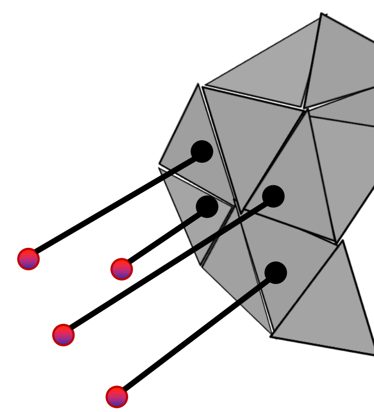


Credit: Seth Billings

Iterative Closest Point (ICP) Revisited



- Widely popular and useful method for point cloud to surface registration introduced by Besl & McKay in 1992
- Many variants proposed since its inception affecting all aspects of the algorithm (robustness, matching criteria, match alignment, etc.)



➢ **Matching Phase**:

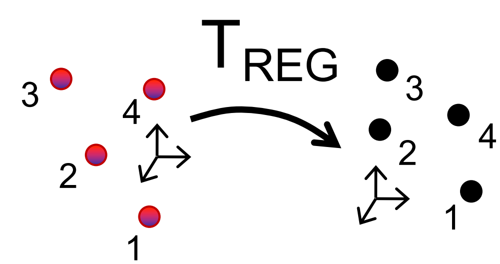
for each point in the source shape, find the closest point on the target shape

$$\boldsymbol{y_i} = \mathrm{C}_{\mathrm{CP}}(T(\boldsymbol{x_i}), \Psi) = \underset{\boldsymbol{y}\in\Psi}{\operatorname{argmin}} \|\boldsymbol{y} - T(\boldsymbol{x_i})\|_2$$

➢ **Registration Phase:**

compute transformation to minimize sum of square distances between matches

$$T = \underset{T}{\operatorname{argmin}} \sum_{i=1}^{n} \|\boldsymbol{y_i} - T(\boldsymbol{x_i})\|_2^2$$



S. Billings and R. H. Taylor, "Iterative Most Likely Oriented Point Registration", ***in Medical Image Computing and Computer-Assisted Interventions (MICCAI)*****, Boston, October, 2014.**

 Credit: Seth Billings

Most-Likely Point Paradigm Illustrated with ICP

1. **Probability Model:** isotropic Gaussian

$$f_{\mathrm{match}}(\mathbf{x} \mid \mathbf{y}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{3/2}} \cdot e^{-\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{x}\|^2}$$

2. **Match Phase:**

$$\begin{aligned}\mathbf{y}_i &= \underset{\mathbf{y}_i \in \Psi}{\operatorname{argmax}}\, f_{\mathrm{match}}(\mathrm{T}(\mathbf{x}_i) \mid \mathbf{y}_i, \sigma^2) \\ &= \underset{\mathbf{y}_i \in \Psi}{\operatorname{argmax}}\, \frac{1}{(2\pi\sigma^2)^{3/2}} \cdot e^{-\frac{1}{2\sigma^2}\|\mathbf{y}_i-\mathrm{T}(\mathbf{x}_i)\|^2} \\ &\rightarrow \underset{\mathbf{y}_i \in \Psi}{\operatorname{argmin}}\, \|\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i)\|\end{aligned}$$

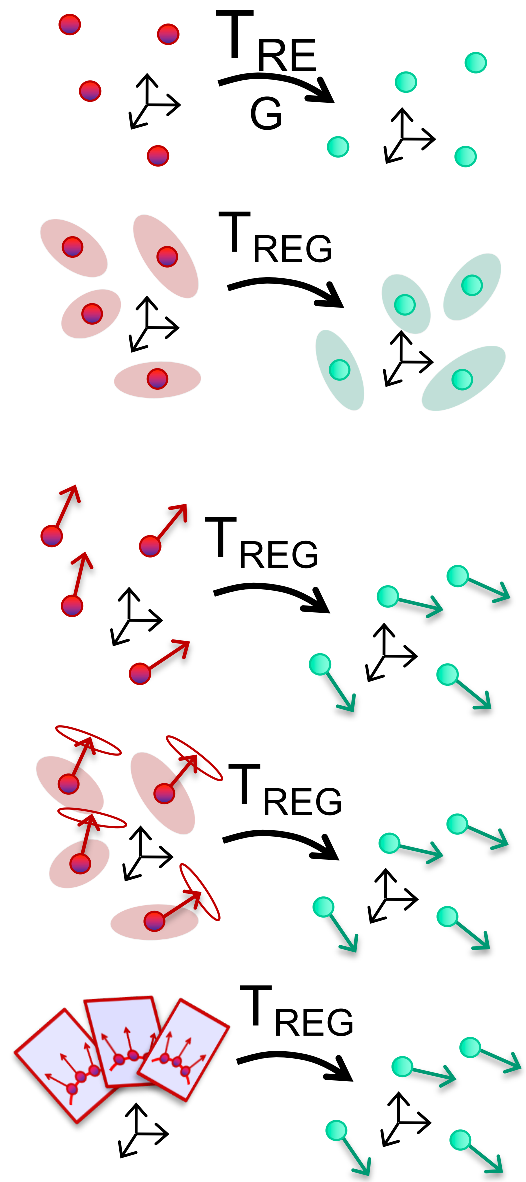
3. **Registration Phase:**

$$\begin{aligned}\mathbf{T} &= \underset{\mathbf{T}}{\operatorname{argmax}} \prod_i^n f_{\mathrm{match}}(\mathrm{T}(\mathbf{x}_i) \mid \mathbf{y}_i, \sigma^2) \\ &= \underset{\mathbf{T}}{\operatorname{argmax}} \prod_i^n \frac{1}{(2\pi\sigma^2)^{3/2}} \cdot e^{-\frac{1}{2\sigma^2}\|\mathbf{y}_i-\mathrm{T}(\mathbf{x}_i)\|^2} \\ &\rightarrow \underset{\mathbf{T}}{\operatorname{argmax}} \left[-n \log\left((2\pi\sigma^2)^{3/2} \right) - \frac{1}{2\sigma^2} \sum_i^n \|\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i)\|^2 \right] \\ &\rightarrow \underset{\mathbf{T}}{\operatorname{argmin}} \sum_i^n \|\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i)\|^2\end{aligned}$$

Credit: Seth Billings

Outline of Registration Algorithms

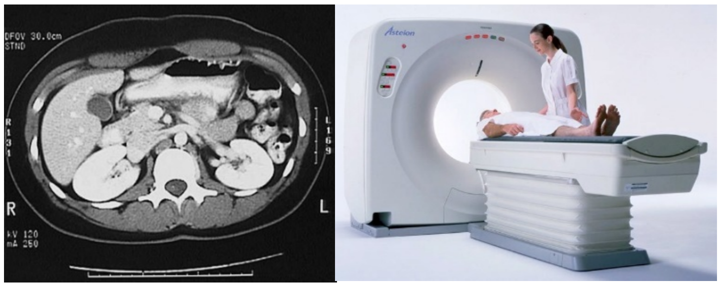
- ICP - Iterative Closest Point
  - isotropic position data
- IMLP - Iterative Most Likely Point
  - anisotropic position data
  - robust to outliers
- IMLOP - Iterative Most Likely Oriented Point
  - isotropic position & orientation data
- G-IMLOP - Generalized IMLOP
  - anisotropic position & orientation data
- P-IMLOP - Projected IMLOP
  - anisotropic position & projected orientation data



Credit: Seth Billings

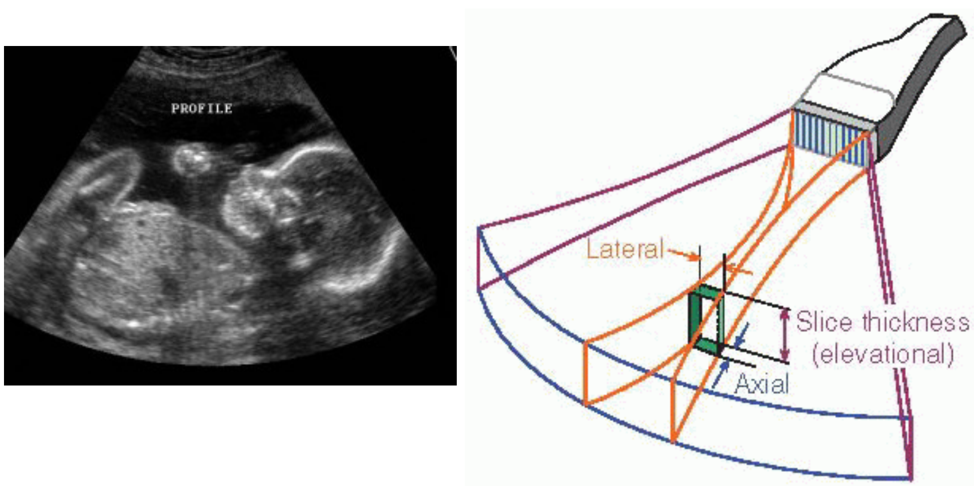
Sources of Anisotropic Uncertainty

Tomographic



Stereo Vision

Ultrasound



Figures: http://www.ndigital.com/wp-content/uploads/2013/09/4col_polarisvicra1.png; *The Essential Physics of Medical Imaging*, 3rd ed.; http://www.infotech.edu/wp-content/uploads/2013/01/L.A.B.-Look-At-Baby-3D-Ultrasound-Tests-Ultrasound-Technicians-bw-300x225.jpg; http://i00.i.aliimg.com/photo/v0/105832128/CT_Scan_equipment.jpg

Prior Work: Anisotropic Registration

- Generalized Total Least Squares ICP (GTLS-ICP)

  Estépar RSJ, Brun A, Westin C-F (2004) Robust generalized total least squares iterative closest point registration. In: *MICCAI 2004*
  - Registration Phase
    - anisotropic noise model
    - ad-hoc implementation less accurate / efficient; can be unstable
  - Match Phase
    - isotropic (i.e. closest-point matching)

- Generalized ICP (G-ICP)

  Segal A, Haehnel D, Thrun S (2009) Generalized-ICP. In: *Robotics: Science and Systems V*
  - Registration Phase
    - anisotropic noise model limited to model locally-linear surface regions surrounding each feature point of a point cloud shape
    - uses off-the-shelf conjugate gradient solver
  - Match Phase
    - isotropic (i.e. closest-point matching)

Prior Work: Anisotropic Registration

- Anisotropic ICP (A-ICP)

  Maier-Hein L, Franz AM, Dos Santos TR, Schmidt M, Fangerau M, et al. (2012) Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error. *IEEE Trans Pattern Anal Mach Intell* 34: 1520–1532.

  – Registration Phase
    - anisotropic noise model
    - ad-hoc implementation does not fully account for noise in both shapes (i.e., lacks ability to reorient the data-shape covariances during optimization)

  – Match Phase
    - anisotropic noise model with non-optimal matching (finds minimal Mahalanobis distance match rather than most-likely match)
    - inefficient implementation; also cannot guarantee that the "best" match is found

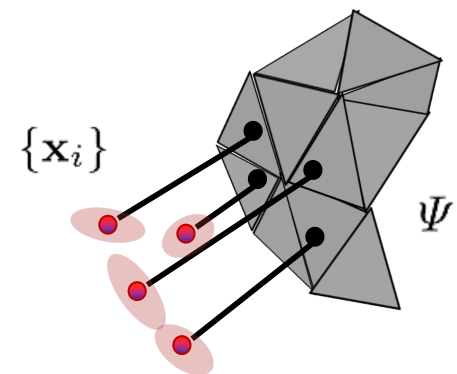
  – Initializes registration by ICP (due to inefficient match phase)

Iterative Most Likely Point (IMLP)

Probability Model: anisotropic Gaussian

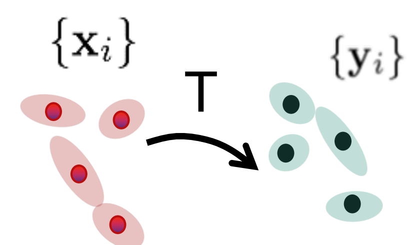
$$f_{\mathrm{match}}(\mathbf{x}\,|\,\mathbf{y},\Sigma_{\mathrm{x}},\Sigma_{\mathrm{y}}) = \frac{1}{(2\pi)^{3/2}|\Sigma_{\mathrm{x}}+\Sigma_{\mathrm{y}}|^{1/2}} \cdot e^{-\frac{1}{2}(\mathbf{y}-\mathbf{x})^T(\Sigma_{\mathrm{x}}+\Sigma_{\mathrm{y}})^{-1}(\mathbf{y}-\mathbf{x})}$$

Match Phase:



$$[\mathbf{y}_i,\mathbf{\Sigma}_{\mathrm{y}i}] = \underset{[\mathbf{y}_i,\mathbf{\Sigma}_{\mathrm{y}i}]\in\Psi}{\operatorname{argmin}} \Big[\log(|\mathbf{R}\mathbf{\Sigma}_{\mathrm{x}i}\mathbf{R}^T+\mathbf{\Sigma}_{\mathrm{y}i}|) + (\mathbf{y}_i-\mathrm{T}(\mathbf{x}_i))^T(\mathbf{R}\mathbf{\Sigma}_{\mathrm{x}i}\mathbf{R}^T+\mathbf{\Sigma}_{\mathrm{y}i})^{-1}(\mathbf{y}_i-\mathrm{T}(\mathbf{x}_i))\Big]$$

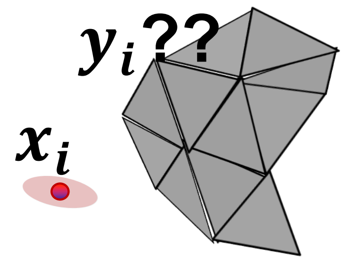
Registration Phase:



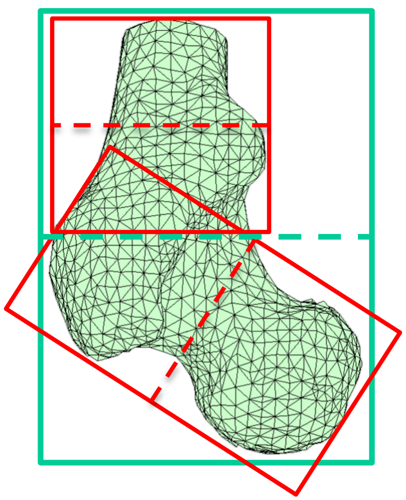
$$\mathbf{T} = \underset{\mathbf{T}=[\mathbf{R},\mathbf{t}]}{\operatorname{argmin}} \sum_i^n (\mathbf{y}_i-\mathrm{T}(\mathbf{x}_i))^T(\mathbf{R}\mathbf{\Sigma}_{\mathrm{x}i}\mathbf{R}^T+\mathbf{\Sigma}_{\mathrm{y}i})^{-1}(\mathbf{y}_i-\mathrm{T}(\mathbf{x}_i))$$

Billings SD, Boctor EM, Taylor RH (2015) Iterative Most-Likely Point Registration (IMLP): A Robust Algorithm for Computing Optimal Shape Alignment. *PLoS One* 10: e0117688

IMLP: Match Phase



- Due to anisotropic distance metric, standard KD-tree search techniques do not apply.
- **Approach**: PD-tree search with modified node test



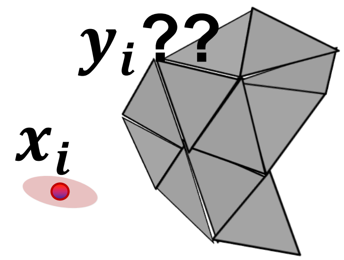
PD Tree Constructed by Datum Positions

Constructing the PD tree:

1. Add all datums to a root node
2. Compute covariance of datum positions within the node
3. Create minimally-sized bounding box aligned to the covariance eigenvectors
4. Partition node along the direction of greatest extent
5. Form left and right child nodes from the datums in each partition
6. Repeat from Step 2 for left and right child nodes until # datums in node < threshold or node size < threshold

Credit: Seth Billings

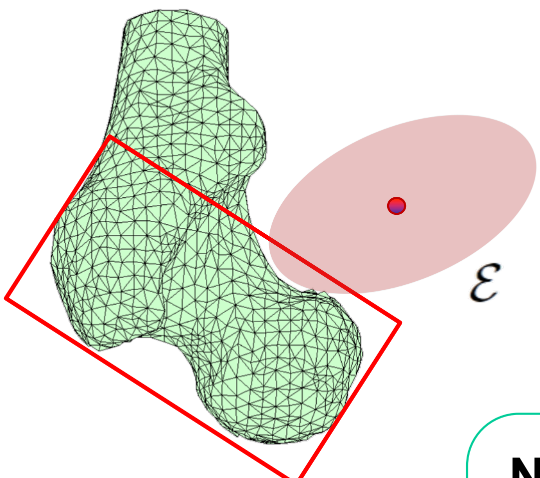
IMLP: Match Phase



Searching the PD tree:

Assume the current match candidate has a match error equal to E_{best}

Question: can any feature in this node possibly provide a match error less than E_{best}?



Node of the PD Tree

$$[\mathbf{y}_i, \boldsymbol{\Sigma}_{\mathrm{y}i}] = \underset{[\mathbf{y}_i, \boldsymbol{\Sigma}_{\mathrm{y}i}] \in \Psi}{\operatorname{argmin}} \Big[\log(|\mathbf{R}\boldsymbol{\Sigma}_{\mathrm{x}i}\mathbf{R}^T + \boldsymbol{\Sigma}_{\mathrm{y}i}|) + (\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i))^T (\mathbf{R}\boldsymbol{\Sigma}_{\mathrm{x}i}\mathbf{R}^T + \boldsymbol{\Sigma}_{\mathrm{y}i})^{-1} (\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i)) \Big]$$

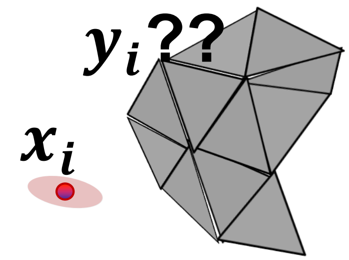
True if: $(\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i))^T (\mathbf{R}\boldsymbol{\Sigma}_{\mathrm{x}i}\mathbf{R}^T + \boldsymbol{\Sigma}_{\mathrm{node}})^{-1} (\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i)) < E_{\mathrm{best}} - log_{\min}$

Node Test: if the **ellipsoid**

$$\mathcal{E} = \{\mathbf{y} \mid (\mathbf{y} - \mathrm{T}(\mathbf{x}_i))^T (\mathbf{R}\boldsymbol{\Sigma}_{\mathrm{x}i}\mathbf{R}^T + \boldsymbol{\Sigma}_{\mathrm{node}})^{-1} (\mathbf{y} - \mathrm{T}(\mathbf{x}_i)) \leq E_{\mathrm{best}} - log_{\min}\}$$

intersects the bounding box of the node, then search the node

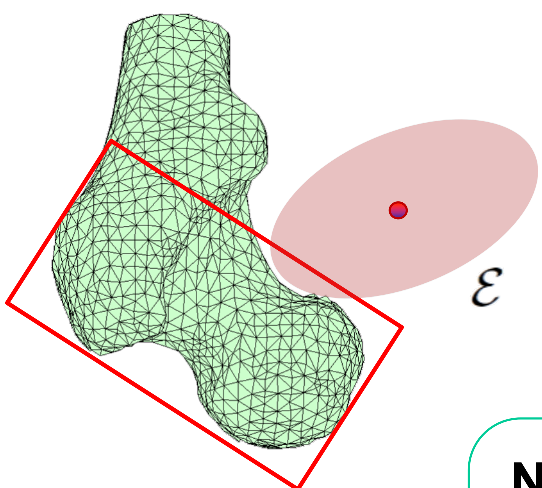
IMLP: Match Phase



Searching the PD tree:

Assume the current match candidate has a match error equal to E_{best}

Question: can any feature in this node possibly provide a match error less than E_{best}?



Node of the PD Tree

$$[\mathbf{y}_i, \mathbf{\Sigma}_{yi}] = \underset{[\mathbf{y}_i, \mathbf{\Sigma}_{yi}] \in \Psi}{\operatorname{argmin}} \Big[\log(|\mathbf{R}\mathbf{\Sigma}_{xi}\mathbf{R}^T + \mathbf{\Sigma}_{yi}|) + (\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i))^T (\mathbf{R}\mathbf{\Sigma}_{xi}\mathbf{R}^T + \mathbf{\Sigma}_{yi})^{-1} (\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i)) \Big]$$

True if: $(\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i))^T (\mathbf{R}\mathbf{\Sigma}_{xi}\mathbf{R}^T + \mathbf{\Sigma}_{node})^{-1} (\mathbf{y}_i - \mathrm{T}(\mathbf{x}_i)) < E_{best} - log_{min}$

Details in Billings' Thesis

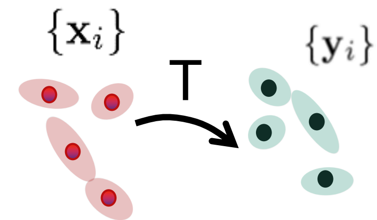
Node Test: if the **ellipsoid**

$$\mathcal{E} = \{\mathbf{y} \mid (\mathbf{y} - \mathrm{T}(\mathbf{x}_i))^T (\mathbf{R}\mathbf{\Sigma}_{xi}\mathbf{R}^T + \mathbf{\Sigma}_{node})^{-1} (\mathbf{y} - \mathrm{T}(\mathbf{x}_i)) \leq E_{best} - log_{min}\}$$

intersects the bounding box of the node, then search the node

Credit: Seth Billings

IMLP: Registration Phase



1. Re-formulate the cost function from an unconstrained optimization

$$\mathbf{T} = \underset{[\mathbf{R},\mathbf{t}]}{\text{argmin}} \sum_{i=1}^{n} (\mathbf{y}_i - \mathbf{R}\mathbf{x}_i - \mathbf{t})^T (\mathbf{R}\boldsymbol{\Sigma}_{\mathrm{x}i}\mathbf{R}^T + \boldsymbol{\Sigma}_{\mathrm{y}i})^{-1} (\mathbf{y}_i - \mathbf{R}\mathbf{x}_i - \mathbf{t})$$

to a constrained optimization

$$\mathbf{T} = \underset{[\mathbf{R},\mathbf{t}]}{\text{argmin}} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{x}_i^*)^T \boldsymbol{\Sigma}_{\mathrm{x}i}^{-1} (\mathbf{x}_i - \mathbf{x}_i^*) + \sum_{i=1}^{n} (\mathbf{y}_i - \mathbf{y}_i^*)^T \boldsymbol{\Sigma}_{\mathrm{y}i}^{-1} (\mathbf{y}_i - \mathbf{y}_i^*)$$

$$\text{subject to:} \quad \mathrm{F}_i(\mathbf{x}_i^*, \mathbf{y}_i^*, \mathbf{R}, \mathbf{t}) = \mathbf{y}_i^* - \mathbf{R}\mathbf{x}_i^* - \mathbf{t} = 0$$

Generalized Total Least Squares (GTLS)

- true (unknown) data-point position
- true (unknown) model-point position

2. Linearize the constraints with a Taylor series centered at the measured (known) data

$$\mathrm{F}_i(\mathbf{x}_i^*, \mathbf{y}_i^*, \mathbf{R}, \mathbf{t}) \approx \mathrm{F}_{\mathrm{L}i}^k(\mathbf{x}_i, \mathbf{y}_i, \mathbf{d}\alpha, \mathbf{dt})$$
$$= \mathrm{F}_i^0(\mathbf{x}_i, \mathbf{y}_i, \mathbf{R}_k, \mathbf{t}_k) - \mathbf{r}_{\mathrm{y}i} + \mathbf{R}_k\mathbf{r}_{\mathrm{x}i} + \mathrm{skew}(\mathbf{R}_k\mathbf{x}_i)\mathbf{d}\alpha - \mathbf{dt} = 0$$

Note using: $\boldsymbol{\Delta}\mathbf{R} \approx \mathbf{I} + \mathrm{skew}(\mathbf{d}\alpha)$ $\quad \mathbf{r}_{\mathrm{x}i} = \mathbf{x}_i - \mathbf{x}_i^*$ $\quad \mathbf{r}_{\mathrm{y}i} = \mathbf{y}_i - \mathbf{y}_i^*$

Credit: Seth Billings

IMLP: Registration Phase

3. Apply the method of Lagrange multipliers to solve constrained optimization.

   3a. Form the Lagrange function using the linearized constraints

$$\mathcal{L}(\mathbf{d}\alpha, \mathbf{dt}, \lambda) = \sum_{i=1}^{n} \mathbf{r}_{\mathrm{x}i}^T \mathbf{\Sigma}_{\mathrm{x}i}^{-1} \mathbf{r}_{\mathrm{x}i} + \sum_{i=1}^{n} \mathbf{r}_{\mathrm{y}i}^T \mathbf{\Sigma}_{\mathrm{y}i}^{-1} \mathbf{r}_{\mathrm{y}i} + \sum_{i=1}^{n} \lambda_i^T \, \mathrm{F}_{\mathrm{L}i}^k(\mathbf{x}_i, \mathbf{y}_i, \mathbf{d}\alpha, \mathbf{dt})$$

   3b. Solve zero gradient w.r.t. the optimization parameters and the Lagrange multipliers

$$\mathbf{J}^T \mathbf{\Sigma}^{-1} \mathbf{J} \mathbf{dp} = -\mathbf{J}^T \mathbf{\Sigma}^{-1} \mathbf{f}^0$$

modified Gauss-Newton

$$\mathbf{dp} = \begin{bmatrix} \mathbf{d}\alpha \\ \mathbf{dt} \end{bmatrix} \quad \mathbf{f}^0 = \begin{bmatrix} \mathbf{f}_1^0 \\ \vdots \\ \mathbf{f}_n^0 \end{bmatrix} \quad \mathbf{J} = \begin{bmatrix} \mathrm{skew}(\mathbf{R}_k \mathbf{x}_1) & -\mathbf{I} \\ \vdots & \vdots \\ \mathrm{skew}(\mathbf{R}_k \mathbf{x}_n) & -\mathbf{I} \end{bmatrix} \quad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{F}_x^0 \mathbf{\Sigma}_{\mathrm{x}} \mathbf{F}_x^{0\,T} + \mathbf{\Sigma}_{\mathrm{y}} \end{bmatrix}$$

$$\mathbf{F}_x^0 = \begin{bmatrix} -\mathbf{R}_k & & \\ & \ddots & \\ & & -\mathbf{R}_k \end{bmatrix} \quad \mathbf{\Sigma}_{\mathrm{x}} = \begin{bmatrix} \mathbf{\Sigma}_{\mathrm{x}1} & & \\ & \ddots & \\ & & \mathbf{\Sigma}_{\mathrm{x}n} \end{bmatrix} \quad \mathbf{\Sigma}_{\mathrm{y}} = \begin{bmatrix} \mathbf{\Sigma}_{\mathrm{y}1} & & \\ & \ddots & \\ & & \mathbf{\Sigma}_{\mathrm{y}n} \end{bmatrix}$$

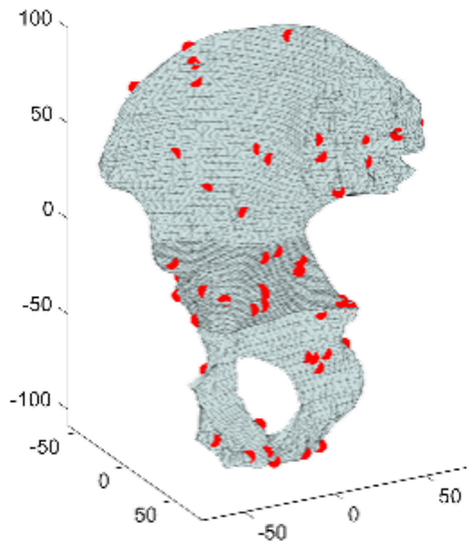
4. Iteratively solve 3b by linear least squares until convergence.

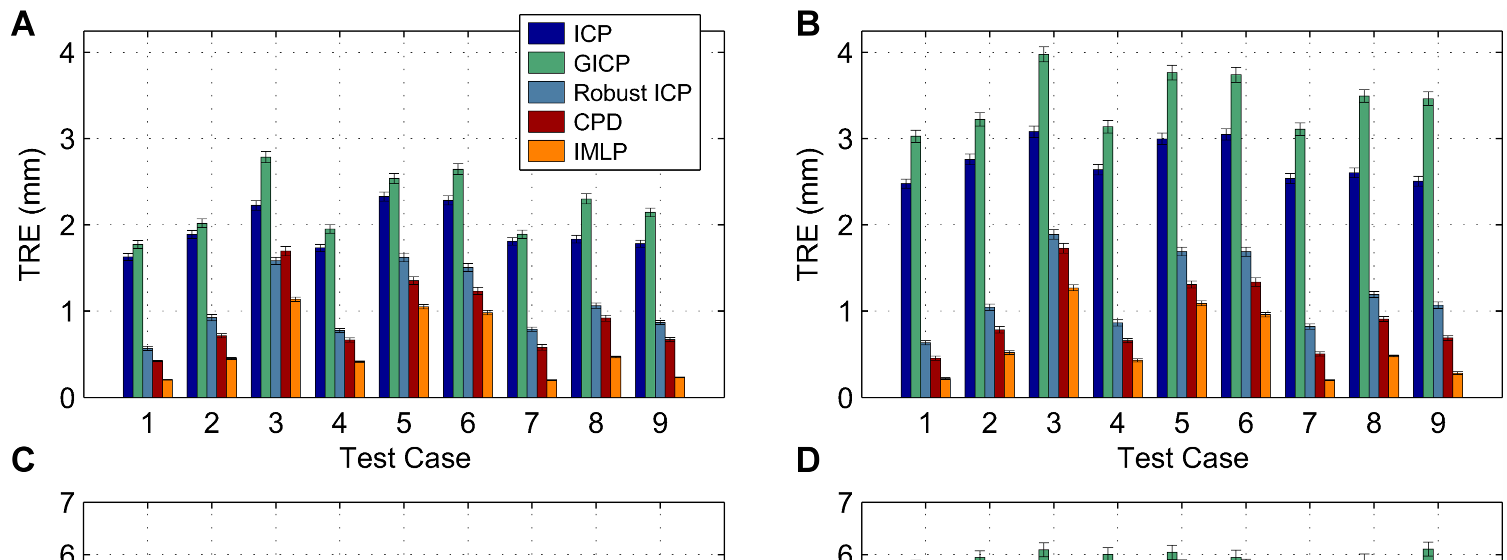
$$\mathbf{R}_{k+1} = \mathbf{R}(\mathbf{d}\alpha)\mathbf{R}_k \,, \quad \mathbf{t}_{k+1} = \mathbf{t}_k + \mathbf{dt}$$

IMLP: Experiments

Credit: Seth Billings

- *Data Shape*: 100 noisy points + *outliers* simulated from a mesh model of a human hip
- *Model Shape*: point-cloud formed from the center points of the mesh triangles
- Random *initial misalignments* [30,60] mm and [30,60] degrees
- Target registration error (TRE) averaged over 300 *randomized trials* for each test case



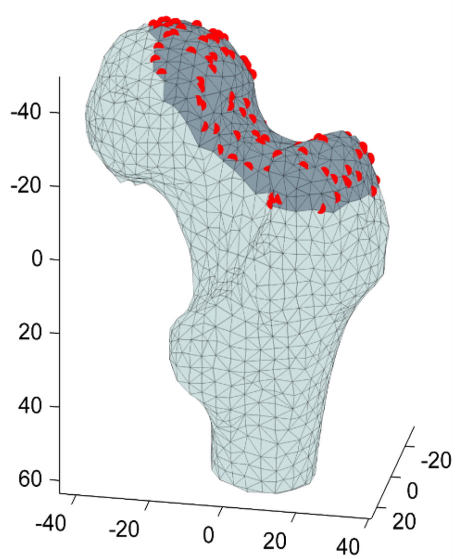


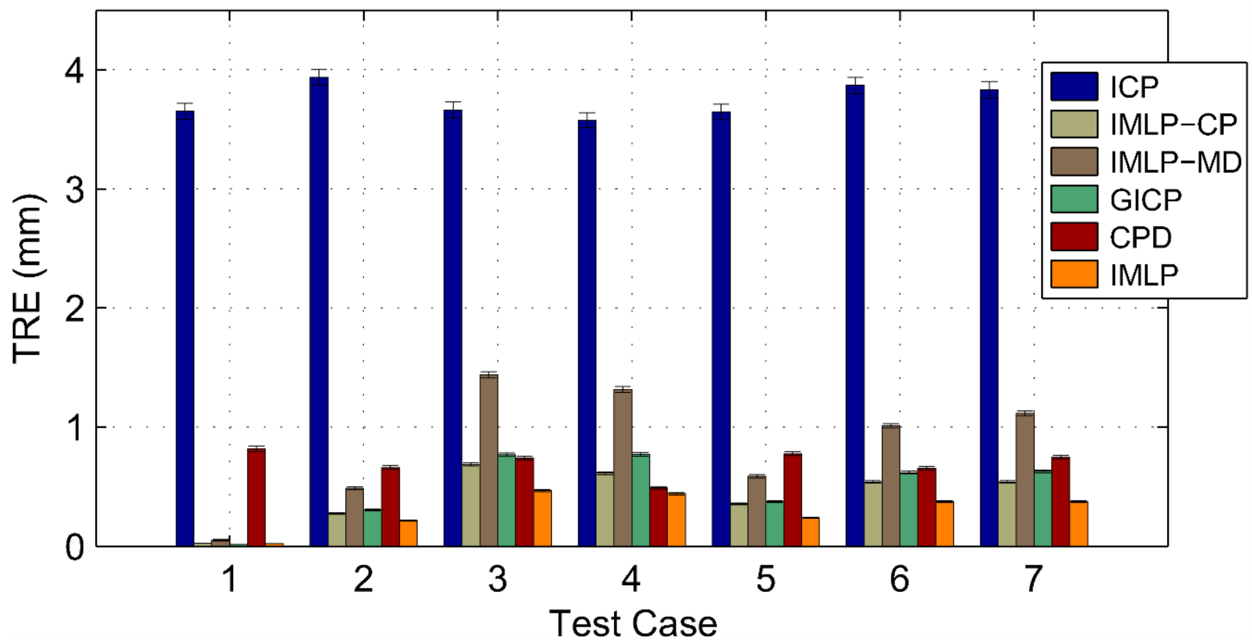
Alg	Average Runtime (sec.) by Test Case								
	1	2	3	4	5	6	7	8	9
ICP	0.009	0.009	0.010	0.009	0.010	0.009	0.009	0.009	0.009
IMLP-CP	0.015	0.016	0.019	0.016	0.020	0.019	0.015	0.017	0.015
IMLP-MD	0.068	0.078	0.093	0.079	0.097	0.093	0.067	0.079	0.069
GICP	-	-	-	-	-	-	-	-	-
CPD (2 cores)	3.465	4.346	4.336	3.864	4.340	4.374	4.238	4.650	4.484
IMLP	0.068	0.082	0.102	0.078	0.103	0.099	0.067	0.084	0.073

IMLP: Experiments

Credit: Seth Billings

- Data Shape: 100 noisy points simulated from a mesh model of a human femur
- Model Shape: point-cloud formed from the center points of the mesh triangles
- Random initial misalignments [10,20] mm and [10,20] degrees
- Target registration error (TRE) averaged over 300 randomized trials for each test case





Alg.	Failure Rate (%) by Test Case						
	1	2	3	4	5	6	7
ICP	15.0	10.7	17.3	13.7	14.7	18.7	16.3
IMLP-CP	4.7	2	5.3	4.3	4.3	4.3	4.0
IMLP-MD	6.0	3.3	7.3	5.3	7.0	6.7	5.3
GICP	6.0	4.3	8.3	6.3	6.0	5.3	4.7
CPD	0.0	0.0	0.0	0.0	0.0	0.3	0.3
IMLP	6.0	3.0	7.0	5.0	6.0	6.3	5.0

Credit: Seth Billings

IMLP: Experiments

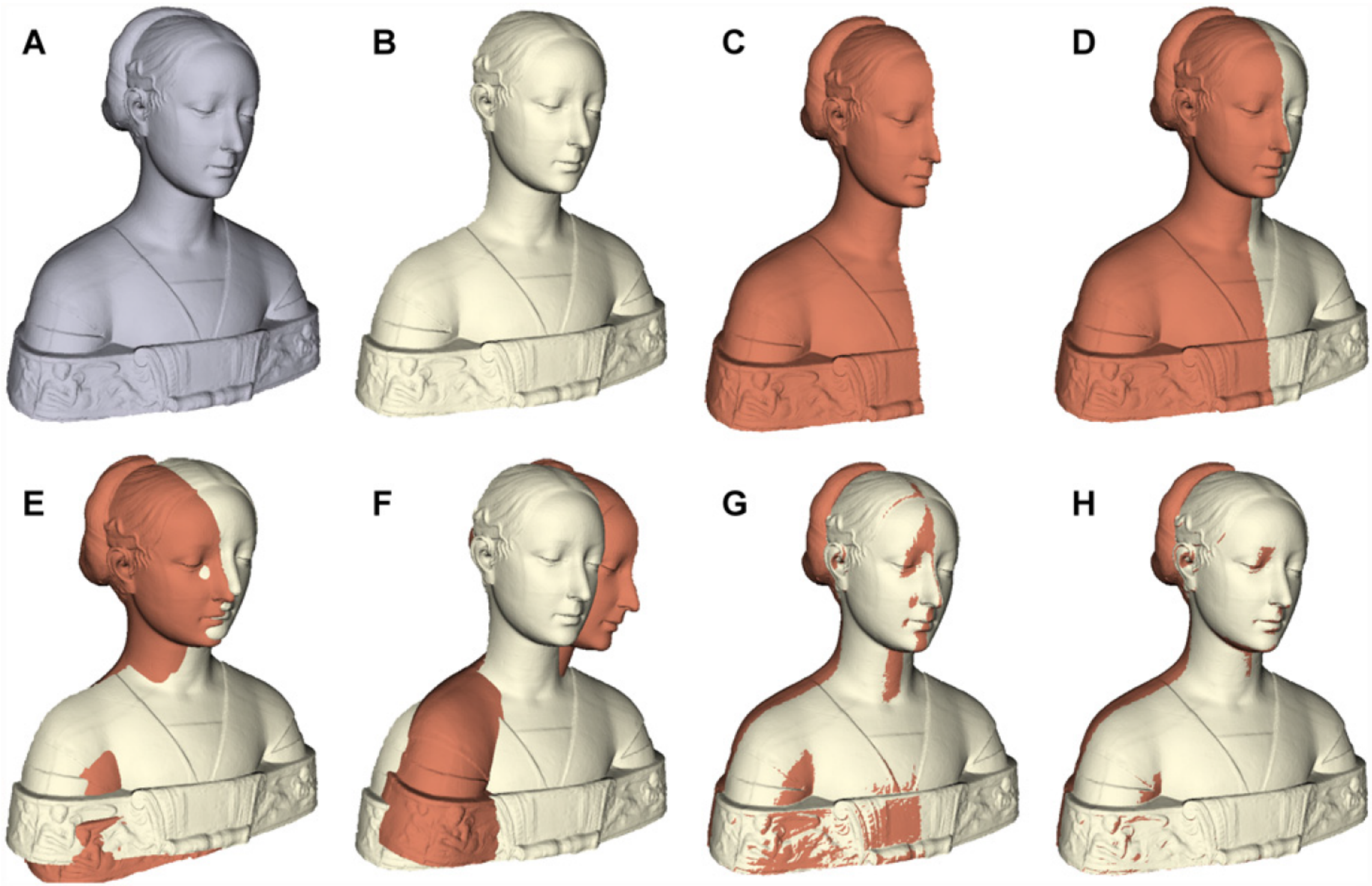
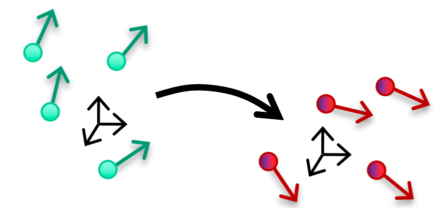


Fig 9. Registration of shapes having partial overlap. (Experiment 7). (A): The statue Laurana sub-divided into (B): front and (C): right half-sections, such that (D): a 50% overlap exists between the two sub-shapes. The sub-shapes were (E): misaligned by 10 mm and 10 degrees in a random direction and then registered using (F): CPD [20], (G): GICP [11], and (H): the proposed IMLP algorithm. Sub-figures (E-H) show the initial misalignment and the final registered alignments of the two shapes for the 6th randomized trial of Experiment 7, which involved 10 randomized trials in total.

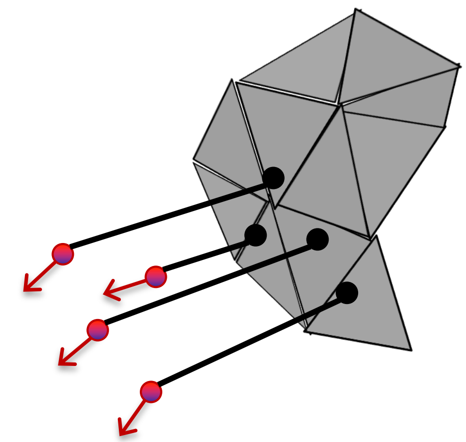
doi:10.1371/journal.pone.0117688.g009

Iterative Most Likely Oriented Point (IMLOP)



- **Matching Phase**:

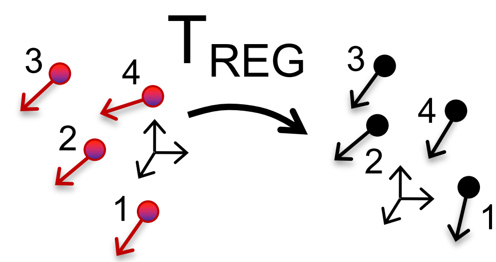
  for each oriented point in the source shape, find the most likely oriented point on the target shape



$$\boldsymbol{y}_i = \mathrm{C}_{\mathrm{MLP}}(T(\boldsymbol{x}_i),\Psi) = \underset{\boldsymbol{y}\in\Psi}{\operatorname{argmax}}\, f_{\mathrm{match}}(T(\boldsymbol{x}_i),\boldsymbol{y})$$

- **Registration Phase:**

  compute transformation to maximize the likelihood (i.e. minimize negative log-likelihood) of oriented point matches

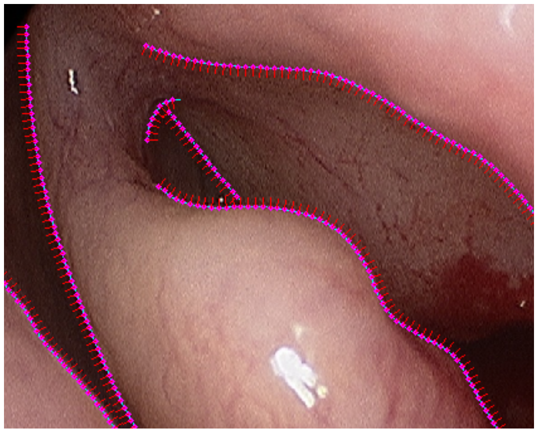


$$T = \underset{T}{\operatorname{argmin}} \left(\frac{1}{2\sigma^2}\sum_{i=1}^{n} \|\boldsymbol{y}_{\mathrm{p}i} - T(\boldsymbol{x}_{\mathrm{p}i})\|_2^2 - k\sum_{i=1}^{n} \boldsymbol{y}_{\mathrm{n}i}{}^{T} R\boldsymbol{x}_{\mathrm{n}i}\right)$$

S. Billings and R. H. Taylor, "Iterative Most Likely Oriented Point Registration", ***in Medical Image Computing and Computer-Assisted Interventions (MICCAI)*****, Boston, October, 2014. (accepted).**

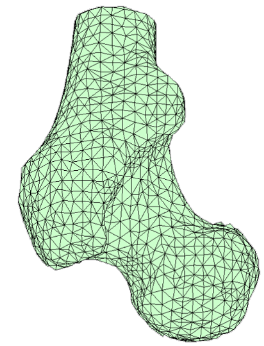
Sources of Orientation Data

Video

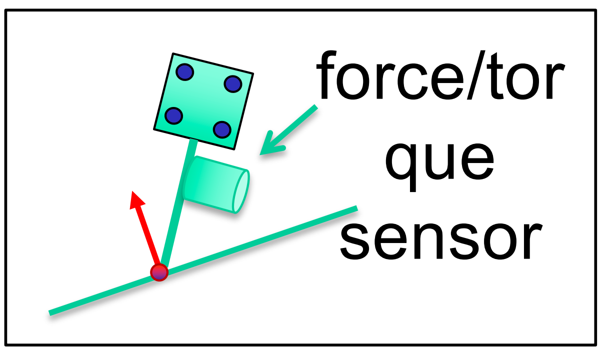


X-Ray

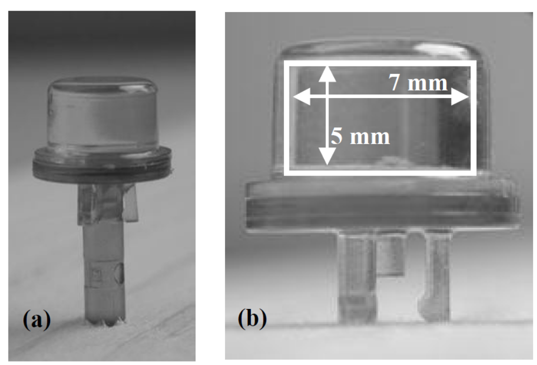
Shape Models



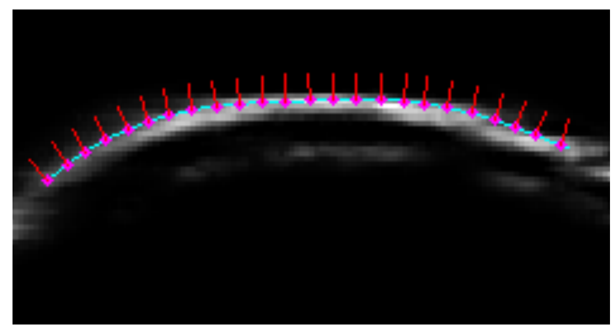
Tracked Pointer



Oriented Fiducials



Ultrasound

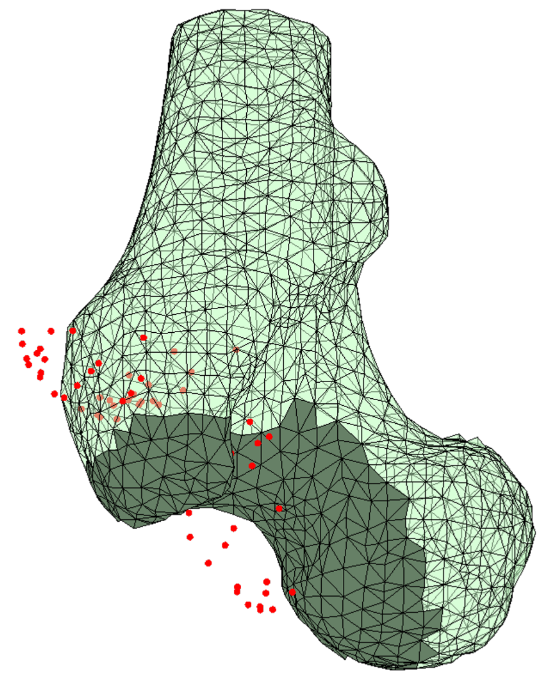


Figures: http://www.ndigital.com/wp-content/uploads/2013/09/4col_polarisvicra1.png; *The Essential Physics of Medical Imaging, 3rd ed.*; *http://www.infotech.edu/wp-content/uploads/2013/01/L.A.B.-Look-At-Baby-3D-Ultrasound-Tests-Ultrasound-Technicians-bw-300x225.jpg*; *http://i00.i.aliimg.com/photo/v0/105832128/CT_Scan_equipment.jpg*; *http://goodpixgallery.com/dl/?i=835588*; Liu X, Cevikalp H, Fitzpatrick JM (2003) Marker orientation in fiducial registration. In: Sonka M, Fitzpatrick JM, editors. SPIE, Medical Imaging 2003: Image Processing. Vol. 5032. pp. 1176–1185.

Experiments

Performance comparison of IMLOP vs. ICP was made through a simulation study using a human femur surface mesh segmented from CT imaging.

- source shape created by randomly sampling points from the mesh surface (10, 20, 35, 50, 75, and 100 points tested)
- Gaussian [wrapped Gaussian] noise added to the source points (0, 0.5, 1.0, and 2.0 mm [degrees] tested)
- Applied random misalignment of [10,20] mm / degrees
- 300 trials performed for each sample size / noise level
- Registration accuracy (TRE) evaluated using 100 validation points randomly sampled from the mesh
- Registration failures automatically detected using threshold on final residual match errors



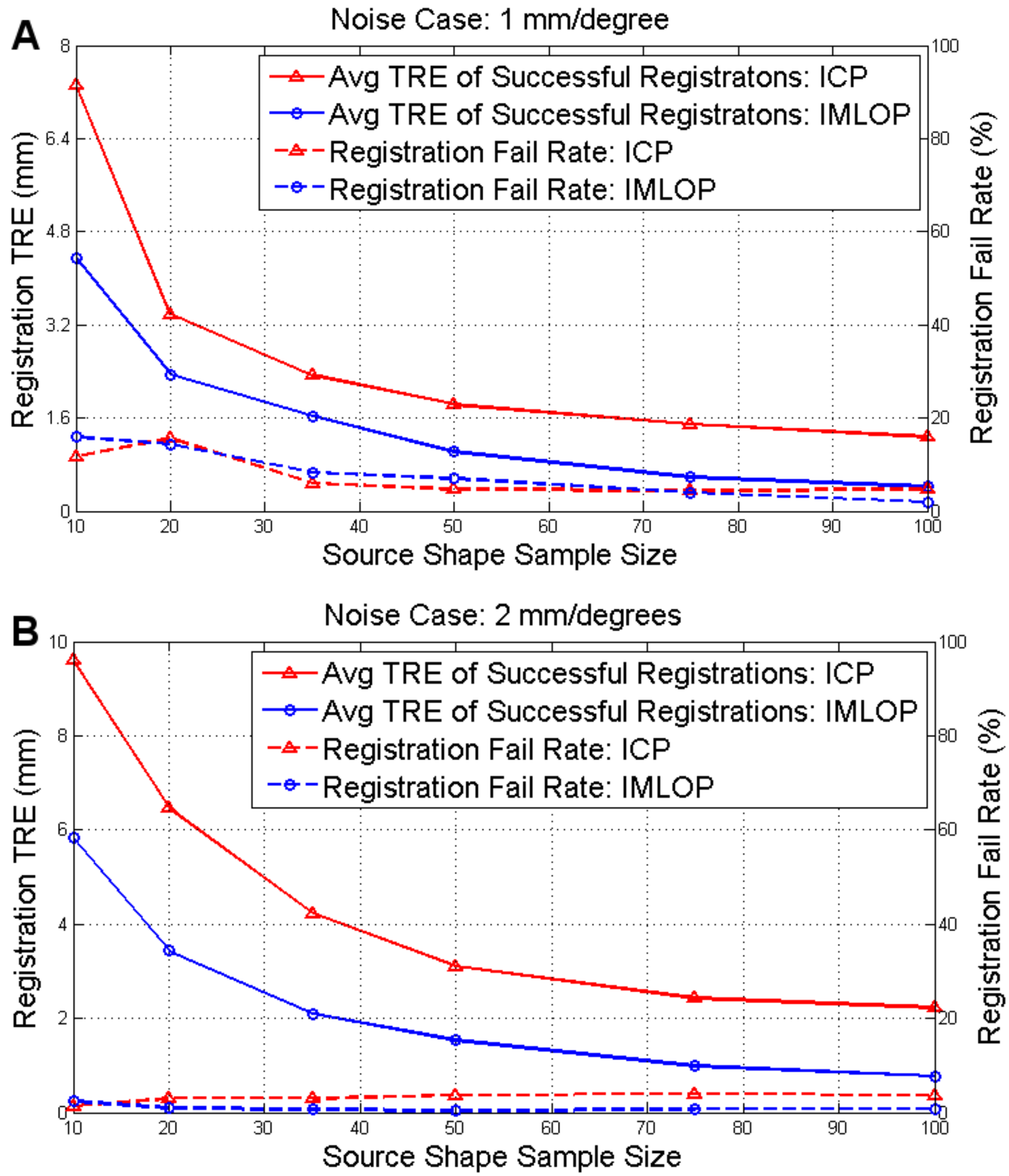
Example source point cloud sampled from dark region of target mesh.

ICP: threshold on position residuals only
IMLOP: threshold on position & orientation residuals

S. Billings and R. H. Taylor, "Iterative Most Likely Oriented Point Registration", *in Medical Image Computing and Computer-Assisted Interventions (MICCAI)*, Boston, October, 2014.

Average TRE of successful registrations and registration failure rates across all sample sizes for noise levels of 1 (A) and 2 (B) mm [degrees].

Registration failure threshold set to twice the noise level for both position and orientation.

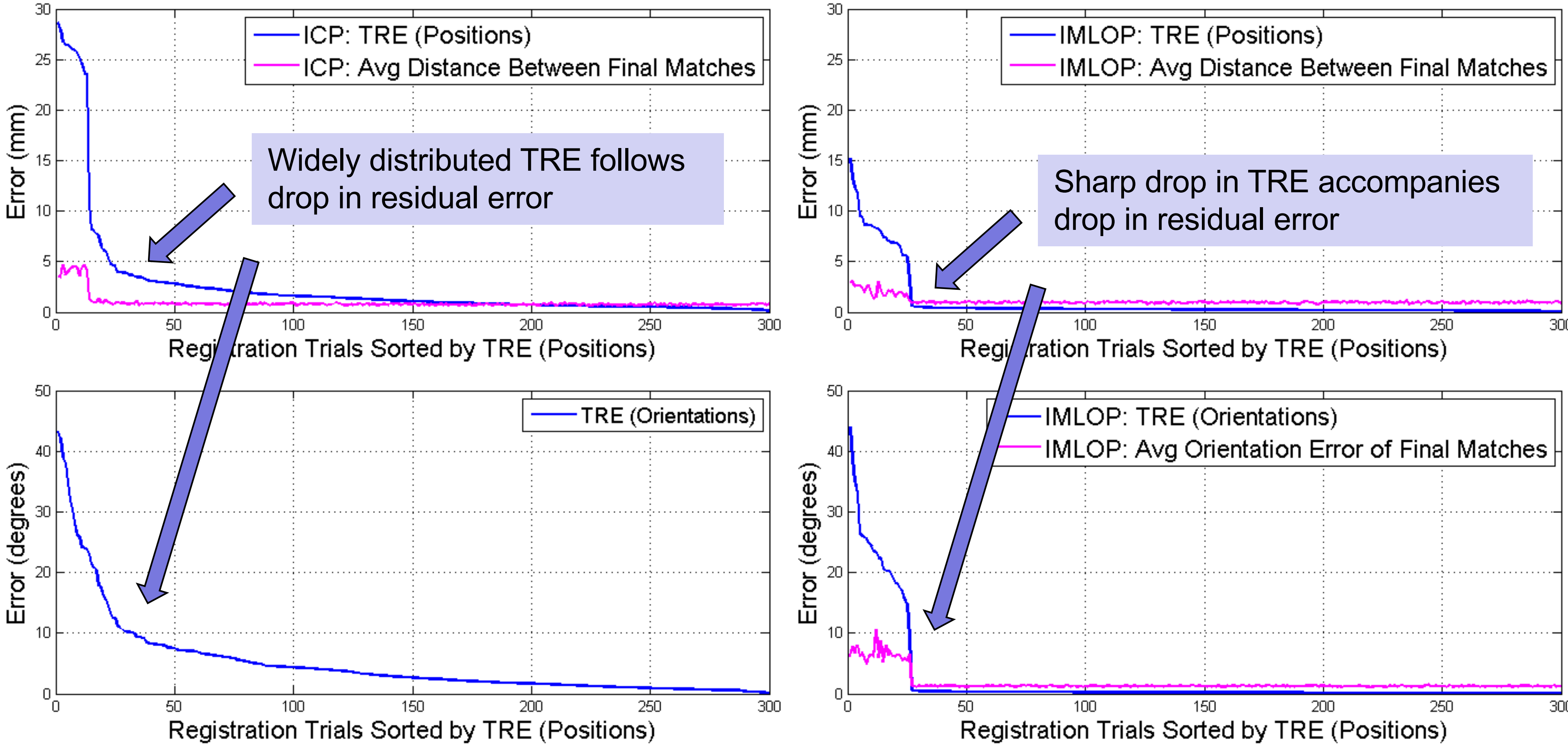


S. Billings and R. H. Taylor, "Iterative Most Likely Oriented Point Registration", ***in Medical Image Computing and Computer-Assisted Interventions (MICCAI)*****, Boson, October, 2014. (accepted).**

Experiments

Results from 300 trials within a single sample size (75 points) and noise level (1.0 mm [degree]). NOTE: improved accuracy and failure detection capability for IMLOP.

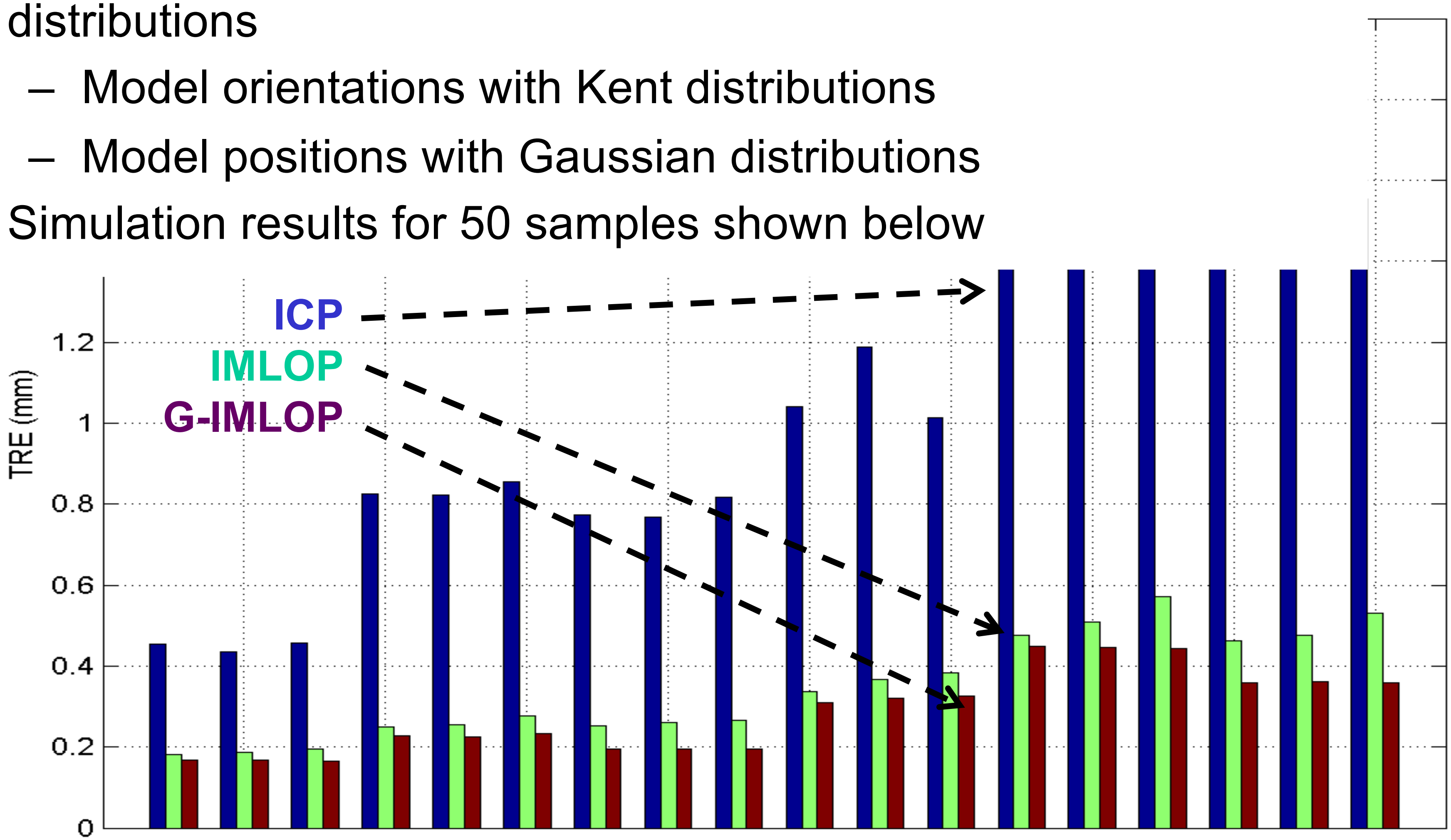
ICP **IMLOP**



S. Billings and R. H. Taylor, "Iterative Most Likely Oriented Point Registration", ***in Medical Image Computing and Computer-Assisted Interventions (MICCAI)*****, Boston, October, 2014. (accepted).**

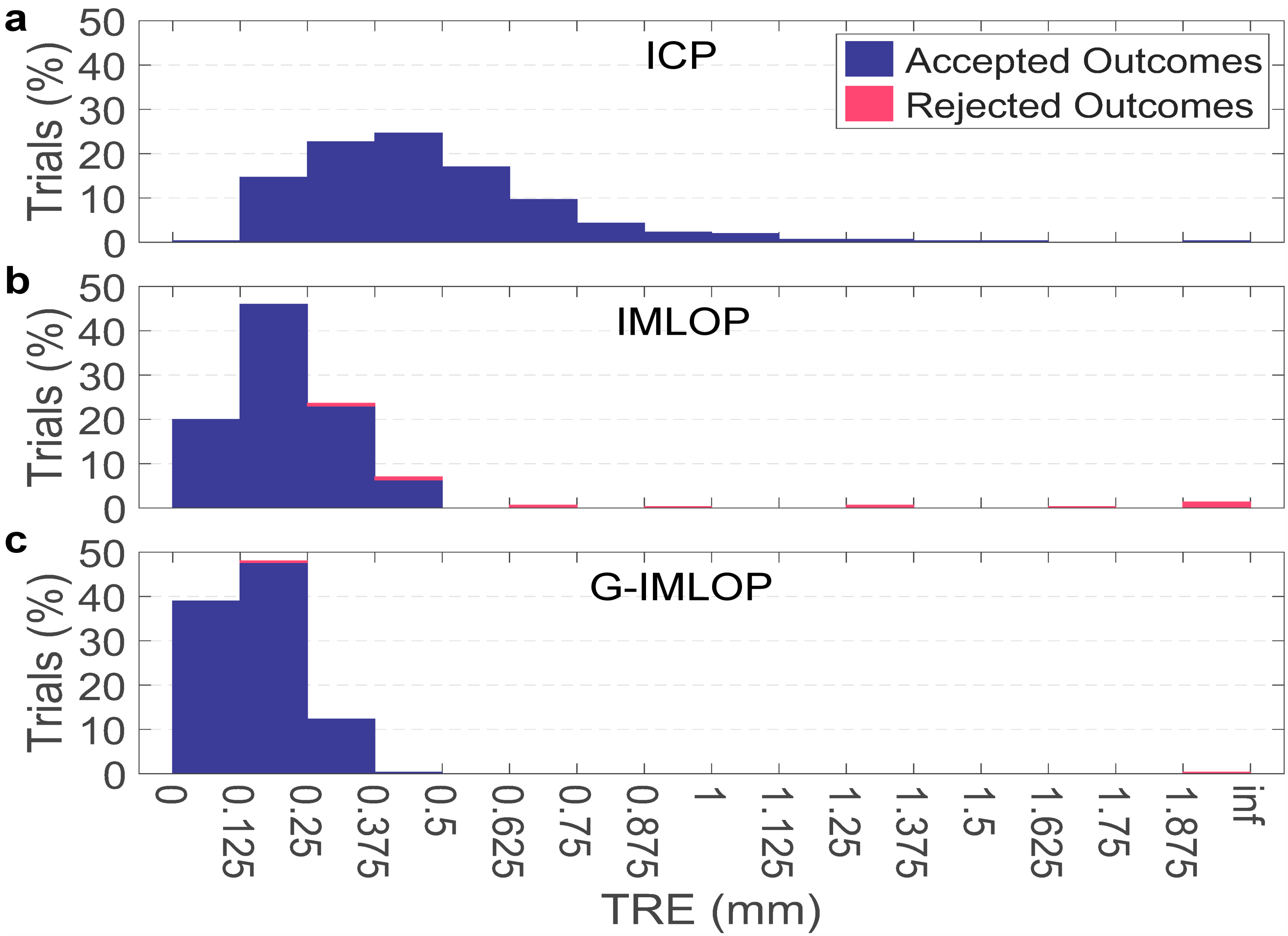
Generalized IMLOP Results

- Extends IMLOP to account for anisotropic measurement error distributions
  - Model orientations with Kent distributions
  - Model positions with Gaussian distributions
- Simulation results for 50 samples shown below



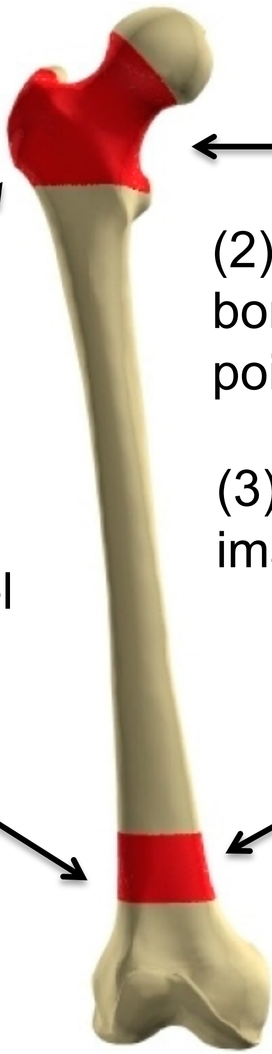
S. Billings and R. Taylor, "Generalized Iterative Most-Likely Oriented Point (G-IMLOP) Registration", ***Int. J. Computer Assisted Radiology and Surgery*****, 8(10) p.1213-1226, 2015.**

Experiments: TRE for Rejected and Non-Rejected Registrations



[S. Billings and R. Taylor, "Generalized Iterative Most-Likely Oriented Point (G-IMLOP) Registration", Int. J. Computer Assisted Radiology and Surgery, p. Accepted 2015.

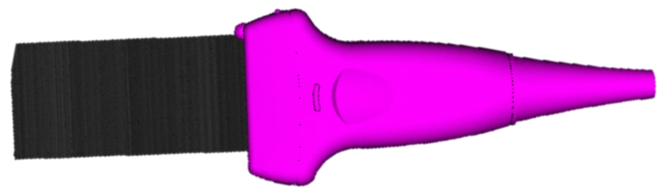
Ultrasound-assisted Registration



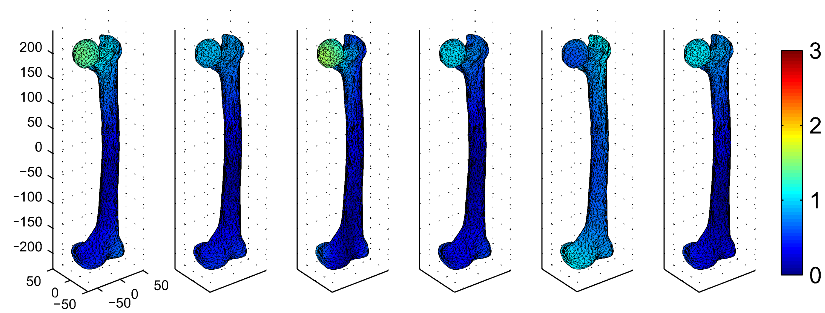
(1) Generate surface model from CT

(2) Digitize proximal bone using tracked pointer

(3) Collect tracked US images of distal bone

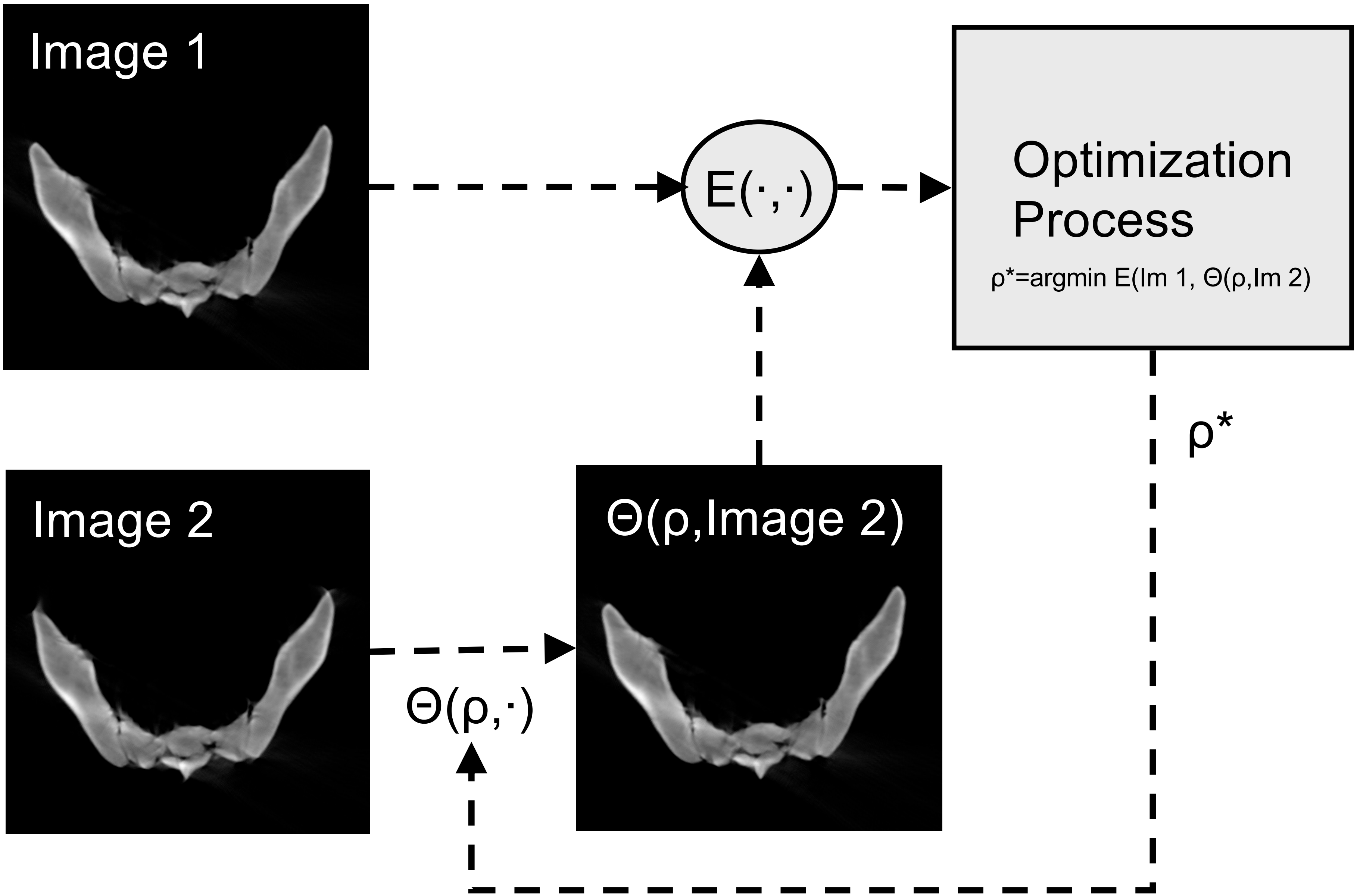


(4) Register points/contours to surface model



S. Billings, H. J. Kang, A. Cheng, E. Boctor, P. Kazanzides, and R. Taylor, "Minimally invasive registration for computer-assisted orthopedic surgery: combining tracked ultrasound and bone surface points via the P-IMLOP algorithm", Int. J. Computer Assisted Radiology and Surgery, p. (epub ahead of print), 2015. http://dx.doi.org/10.1007/s11548-015-1188-z DOI 10.1007/s11548-015-1188-z

Intensity-based methods



Intensity-based methods

- Typically performed between images
- The "features" in this case are the intensities associated with pixels (2D) or voxels (3D) in the images.
- General framework:

$$\vec{\rho}^* = \min_{\vec{\rho}} E\left(Image_1, \Theta\left(\vec{\rho}, Image_2\right)\right)$$

- Methods differ mostly in choice of transformation function *Θ(·)* and Energy function *E(·,·)*,

Typical energy functions
(not an exhaustive list)

Normalized image subtraction

$$E(\mathrm{Im}_1,\mathrm{Im}_2)=\sum_{\vec{k}}\frac{\left|\mathrm{Im}_1\left[\vec{k}\right]-\mathrm{Im}_2\left[\vec{k}\right]\right|}{\max_{\vec{j}}\left(\left|\mathrm{Im}_1\left[\vec{j}\right]-\mathrm{Im}_2\left[\vec{j}\right]\right|\right)}$$

Normalized cross correlation (NCC)

$$E(\mathrm{Im}_1,\mathrm{Im}_2)=\frac{\sum_{\vec{k}}\left(\mathrm{Im}_1\left[\vec{k}\right]-avg(\mathrm{Im}_1)\right)\left(\mathrm{Im}_2\left[\vec{k}\right]-avg(\mathrm{Im}_2)\right)}{\sqrt{\sum_{\vec{k}}\left(\mathrm{Im}_1\left[\vec{k}\right]-avg(\mathrm{Im}_1)\right)^2}\sqrt{\sum_{\vec{k}}\left(\mathrm{Im}_2\left[\vec{k}\right]-avg(\mathrm{Im}_2)\right)^2}}$$

Mutual information

$$E(\mathrm{Im}_1,\mathrm{Im}_2)=\sum_{p\in\mathrm{Im}_1\, q\in\mathrm{Im}_2}\Pr(p,q)\log\Pr(p,q)-\Pr_{\mathrm{Im}_1}(p)\log\Pr_{\mathrm{Im}_1}(p)-\Pr_{\mathrm{Im}_2}(q)\log\Pr_{\mathrm{Im}_2}(q)$$

Mutual Information

- First proposed independently in 1995 by Collignon and Viola & Wells.
- Very widely practiced
- Is able to co-register images with very different sensor modalities so long as there is a stable relationship between intensities in one modality with those in another
- Many “flavors” and variations

Mutual Information

Entropy

$$H(a) = \Pr(a)\log\Pr(a)$$

$$H(a,b) = \Pr(a,b)\log\Pr(a,b)$$

Mutual Information (Viola & Wells '95, Colligen '95)

$$\mathit{Similarity}(A,B) = H(A) + H(B) - H(A,B)$$

Normalized mutual information (Maes *et al.* '97)

$$\mathit{Similarity}(A,B) = \frac{H(A) + H(B)}{H(A,B)}$$

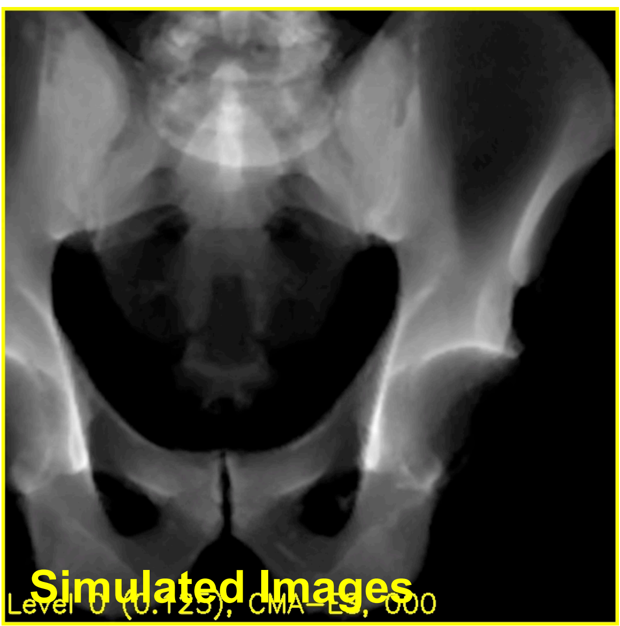
Objective function

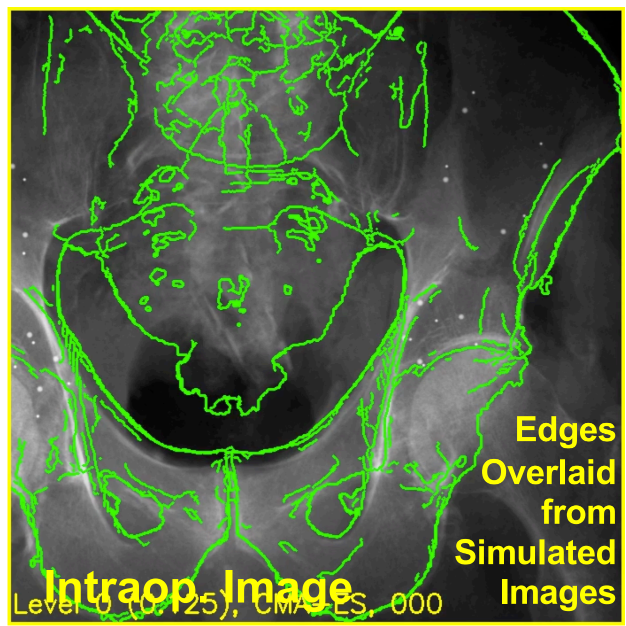
$$\mathrm{E}(\mathrm{Im}_1,\mathrm{Im}_2) = -\mathit{Similarity}(\mathrm{Im}_1,\mathrm{Im}_2)$$

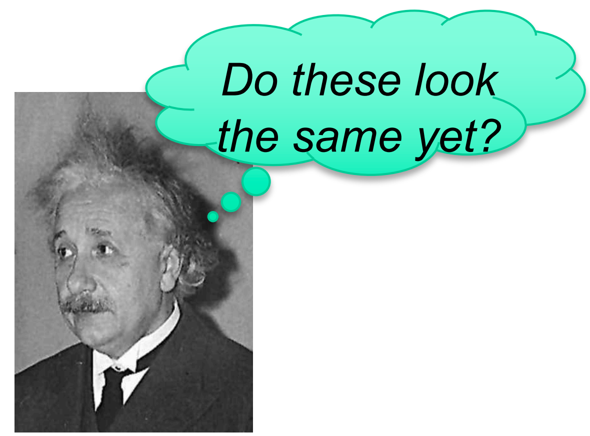
Basic Idea of Intensity-Based 2D/3D Registration

- Assumes a pre-op CT is available
- Simulate many C-Arm images and choose the most similar to the intraoperative image
- Solves the following optimization problem:

$$\operatorname*{argmin}_{\theta \in SE(3)} \mathcal{S}\left(I_{\text{Intra-Op}}, \mathcal{P}(\theta, I_{\text{CT}})\right)$$



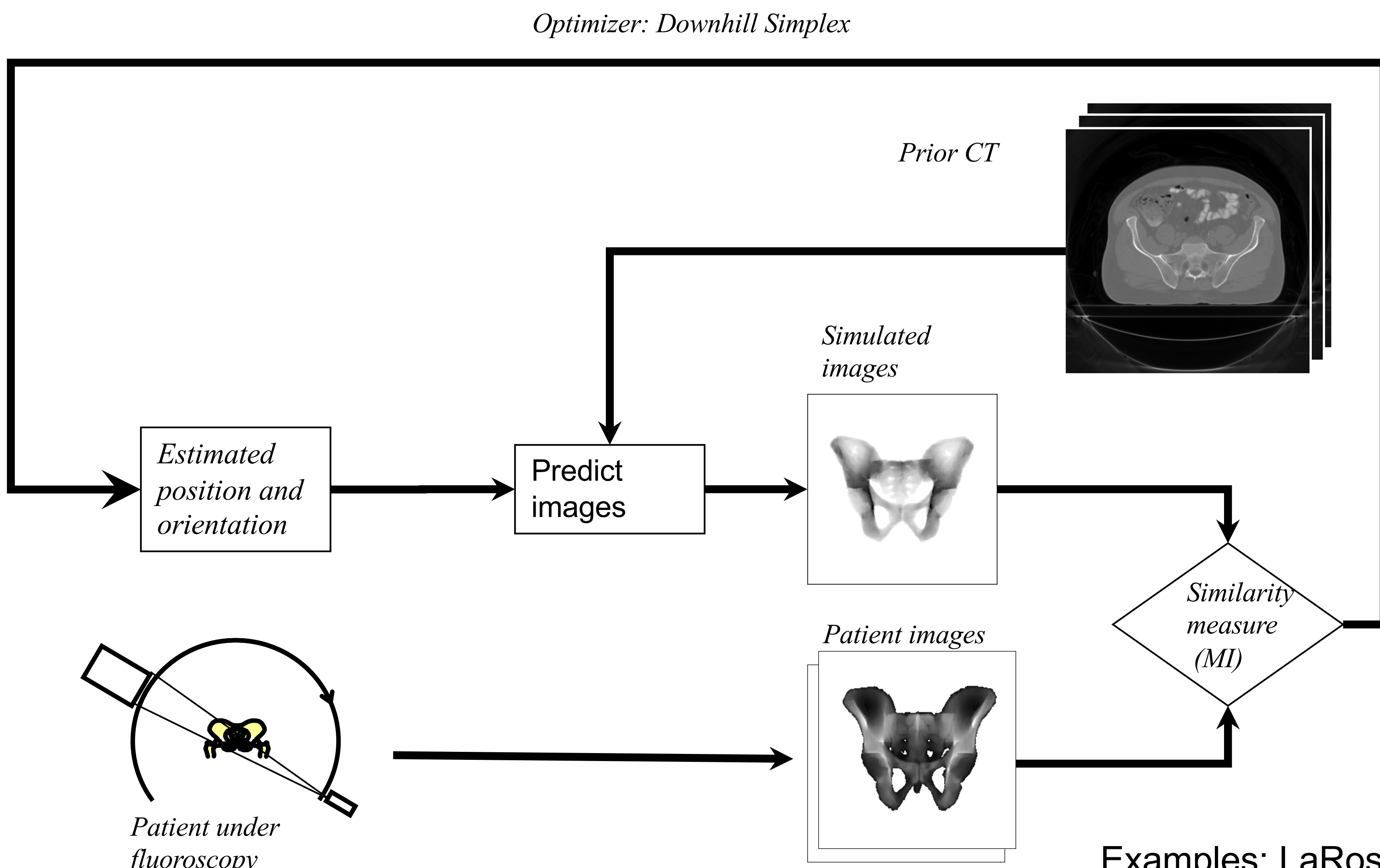




Slide credit: Robert Grupp

Rigid 3D/2D Registration

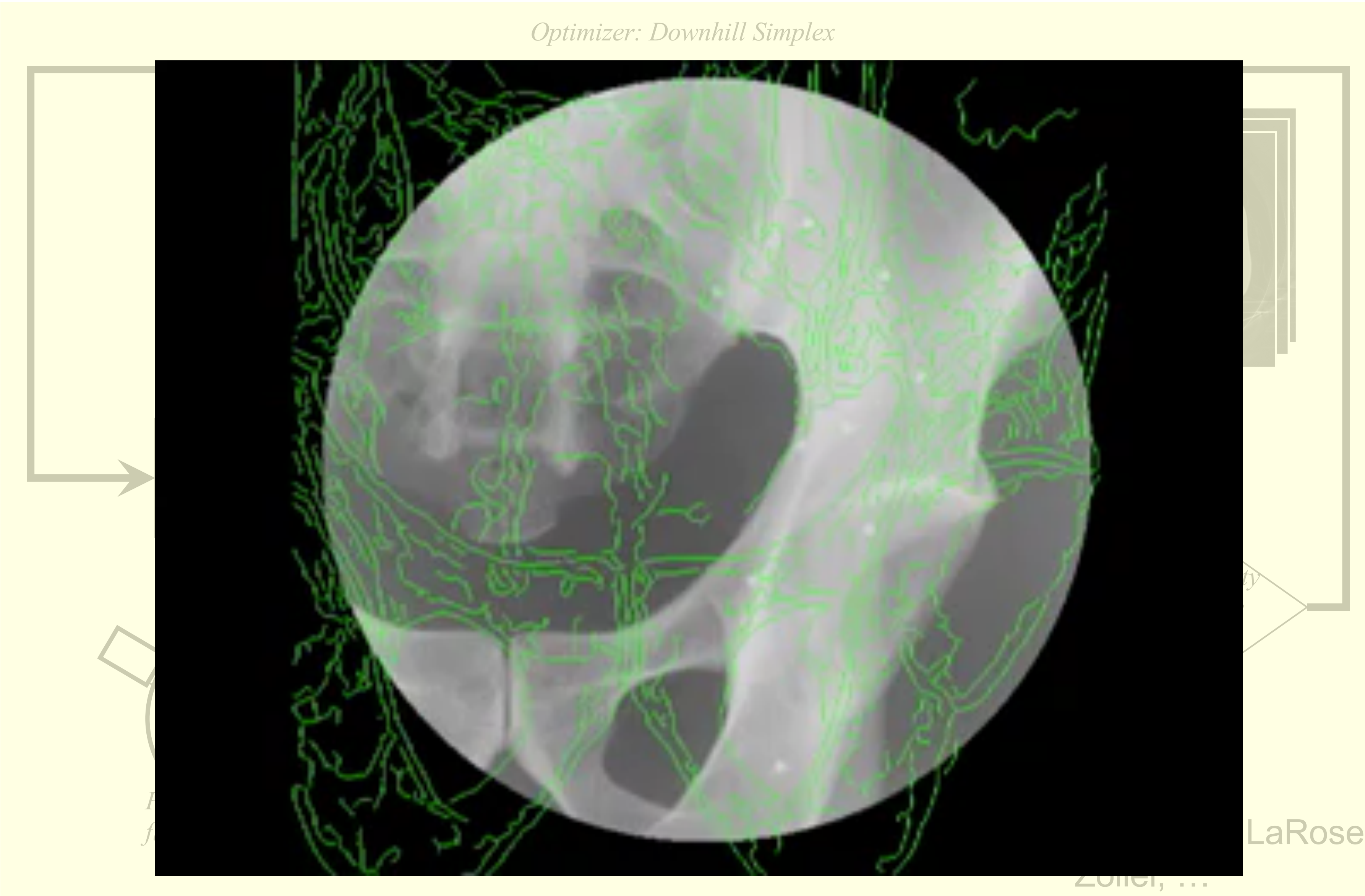
Ofri Sadowsky



Examples: LaRose, Zollei, …

Rigid 3D/2D Registration

Ofri Sadowsky



A clinical example (periacetublar osteotomy)

Problem: Acetabular Dysplasia

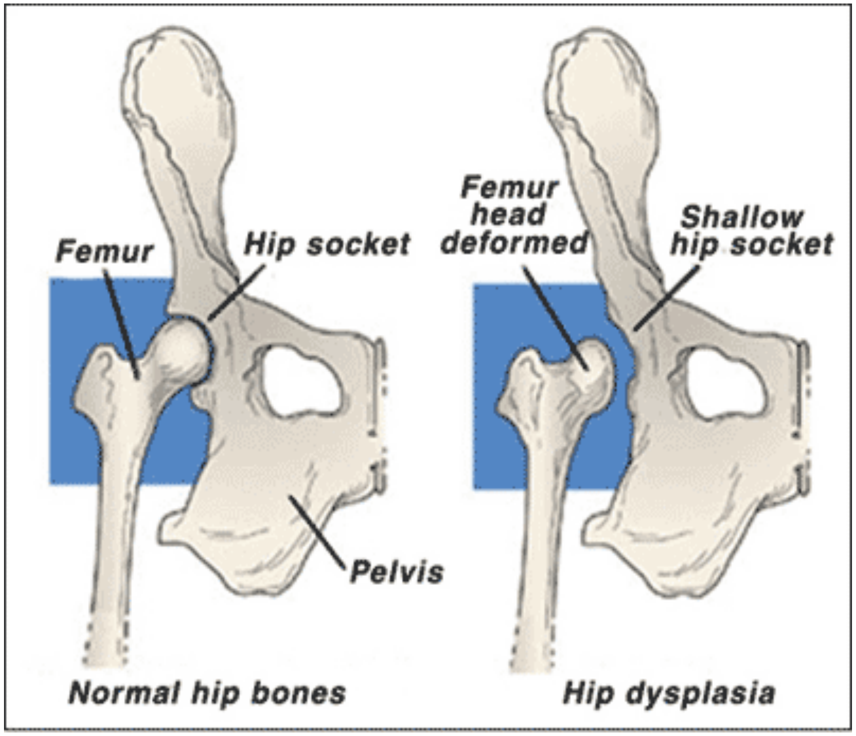


Image Source: ouh.nhs.uk

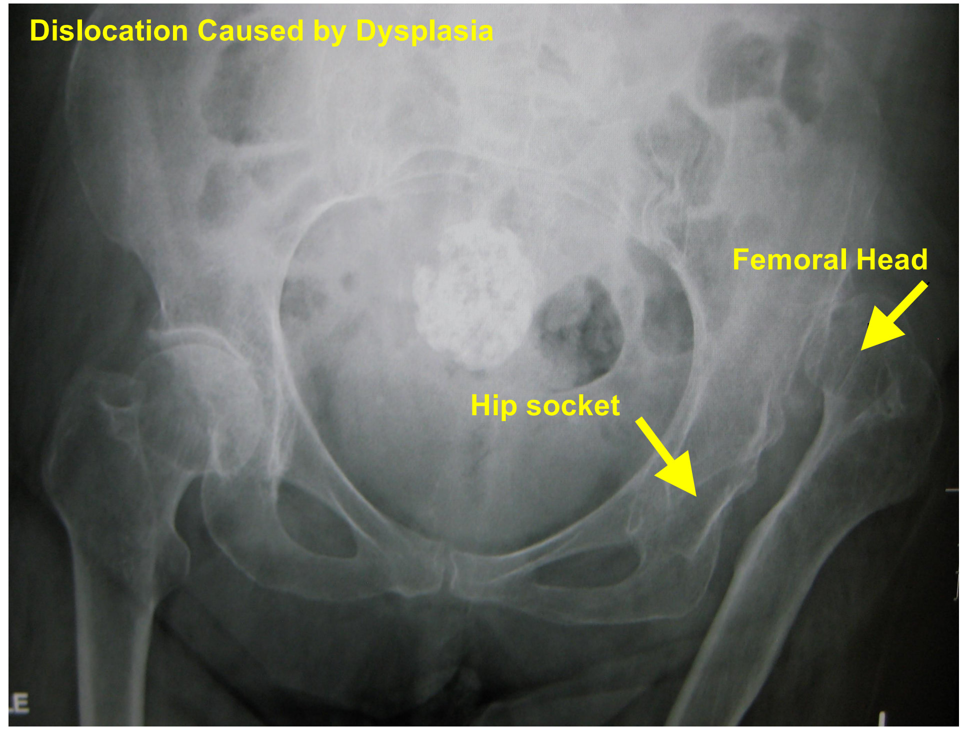


Image Source: James Heilman, MD

A clinical example (periacetublar osteotomy)

One Solution: Periacetabular Osteotomy (PAO)

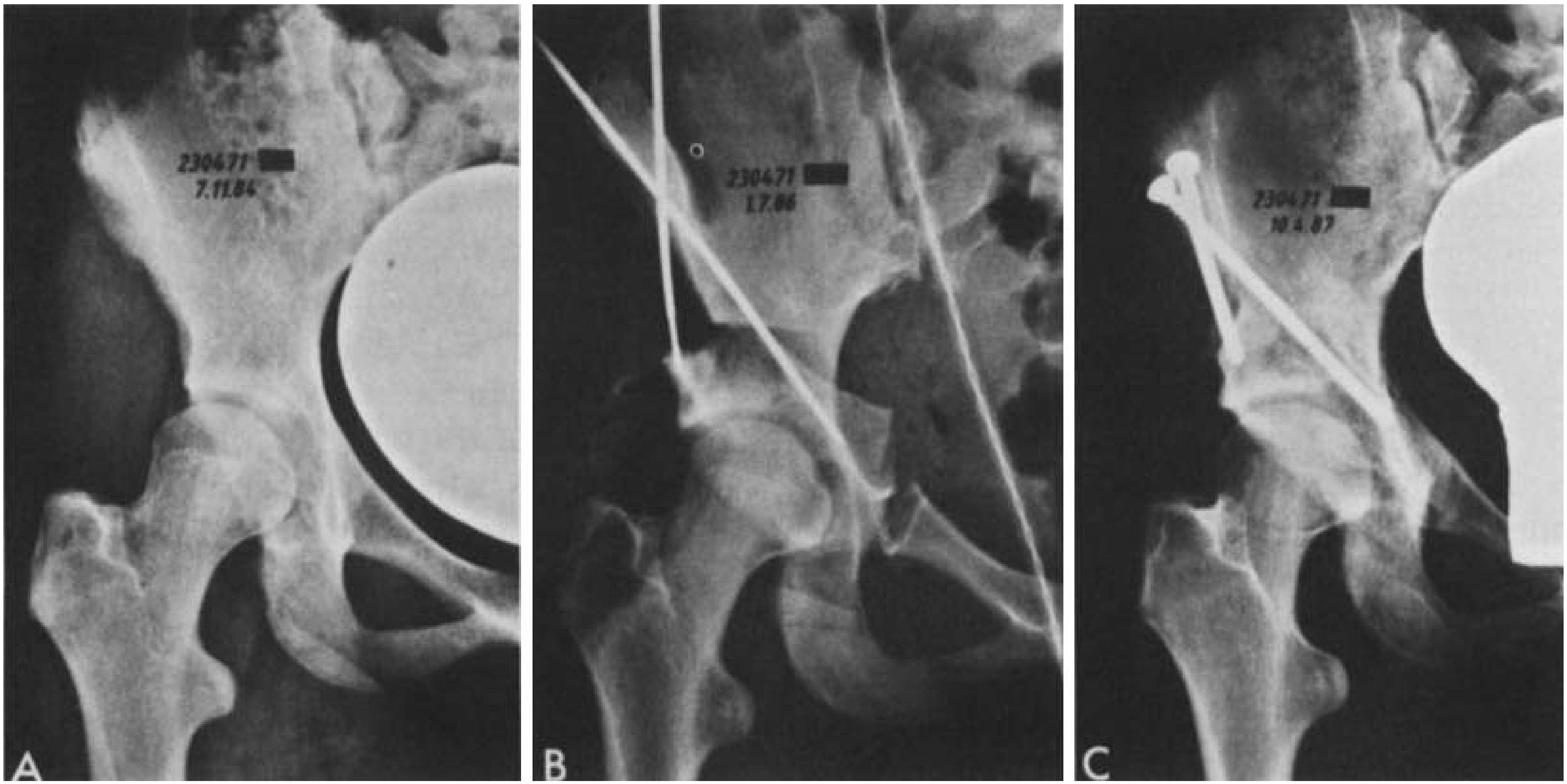
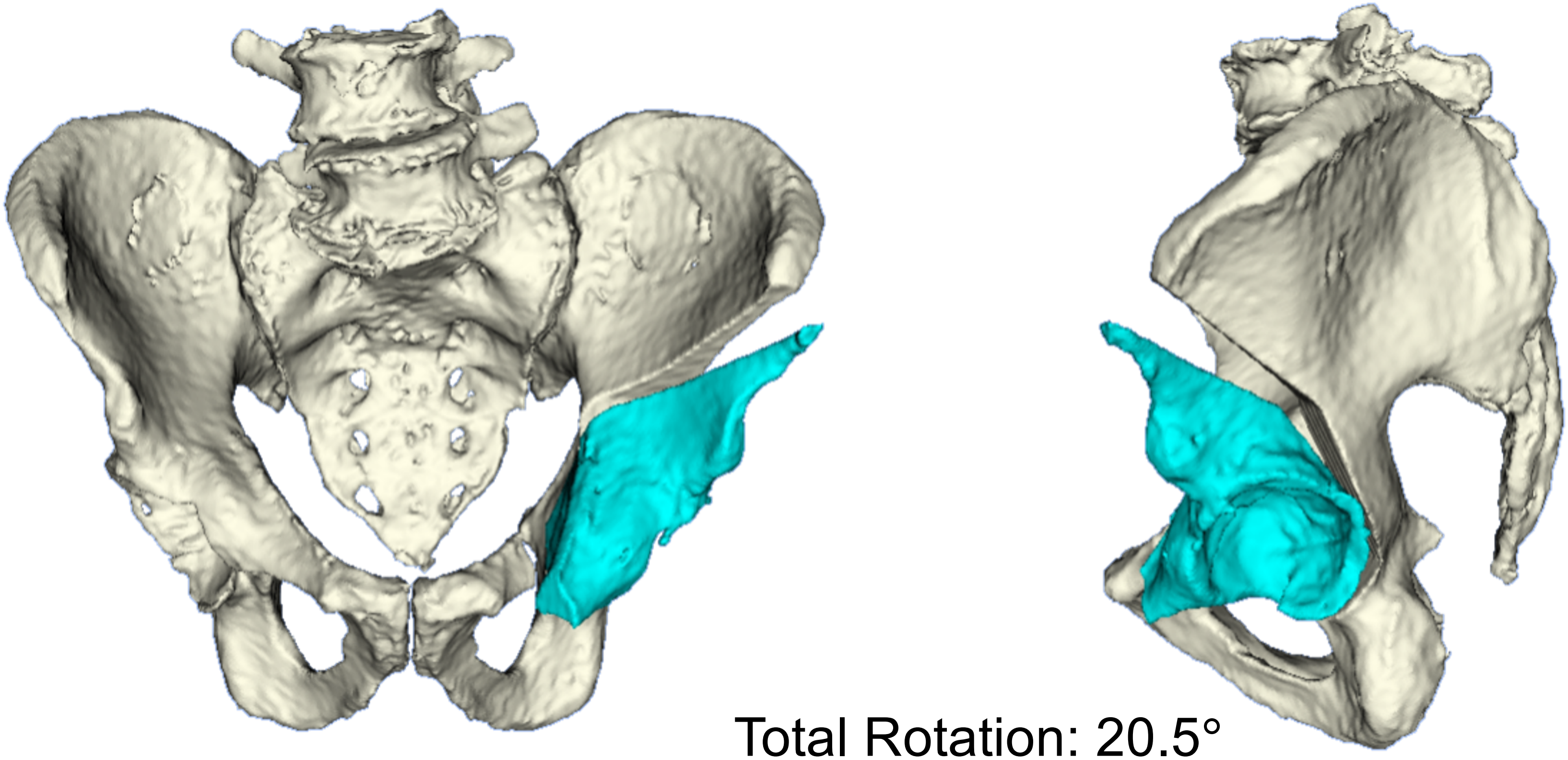


Image Source: Ganz 1988

A clinical example (periacetublar osteotomy)

Goal: Automatic visualization and guidance



Total Rotation: 20.5°
Anterior/Posterior Rotation: 3.7°
Left/Right Rotation: 16.3°
Inferior/Superior Rotation: 12.5°

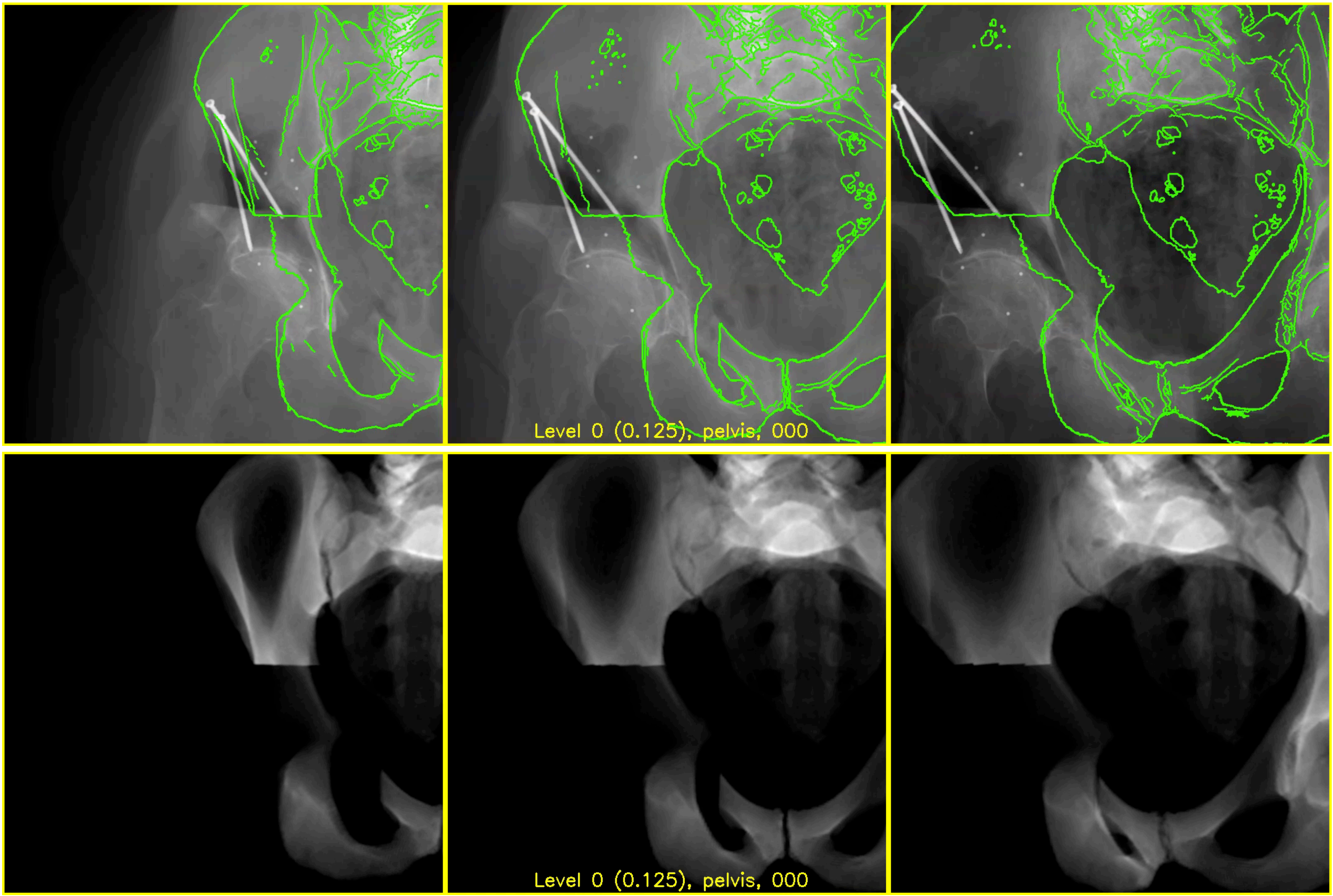
Slide credit: Robert Grupp

3D-2D Registration of Osteotomy Fragments

$$\underset{\theta_1,\ldots,\theta_N \in SE(3)}{\arg\min} \sum_{m=1}^{M} \mathcal{S}\left(I_m, \sum_{n=1}^{N} \mathcal{P}_m\left(I_{CT}; \theta_n\right)\right)$$

Fixed Images with Moving Image Edges

Moving Images



R. Grupp, R. Murphy, M. Armand, R. Taylor

Slide credit: Robert Grupp

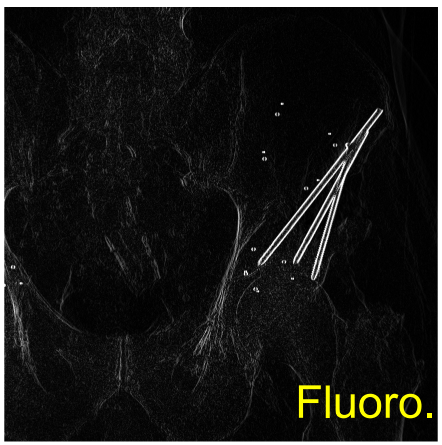
3D-2D Registration of Osteotomy Fragments

- Compute the Sobel derivatives in the X and Y directions of the two input images:

$$\nabla_X I_1,\ \nabla_X I_2,\ \nabla_Y I_1,\ \nabla_Y I_2$$

- Compute NCC between the corresponding gradient images:

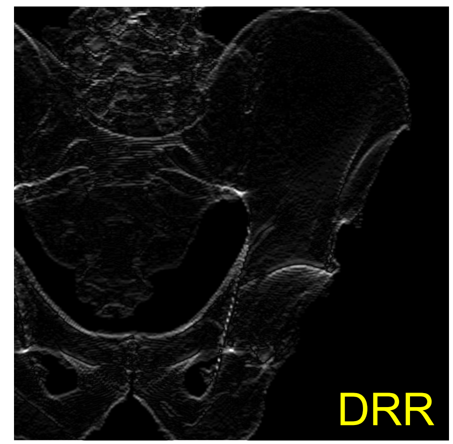
$$\mathcal{S}(I_1, I_2) = NCC(\nabla_X I_1, \nabla_X I_2) + NCC(\nabla_Y I_1, \nabla_Y I_2)$$



$\nabla_X I_1$

$\nabla_X I_2$

$\nabla_Y I_1$

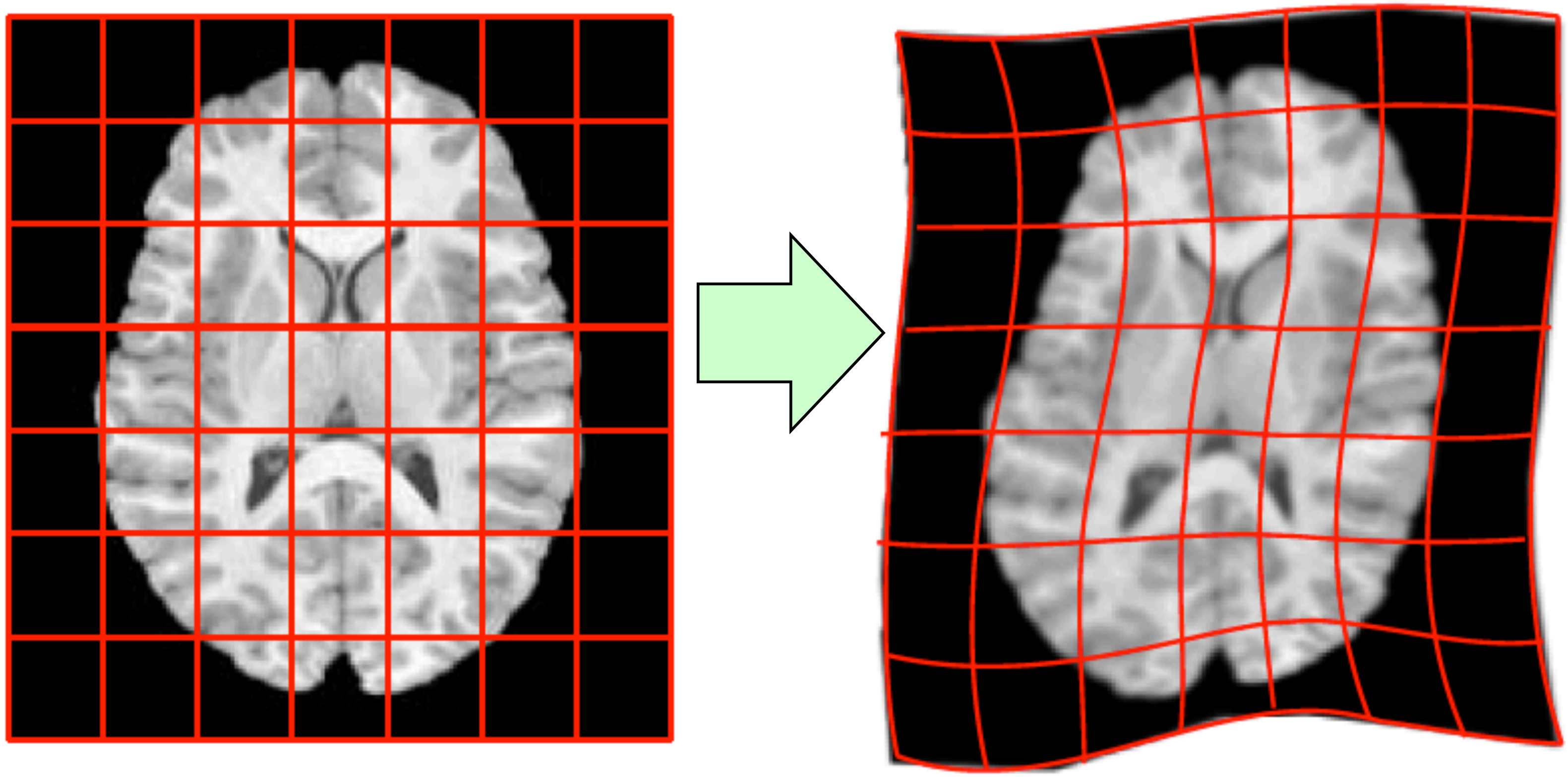


$\nabla_Y I_2$

R. Grupp, R. Murphy, M. Armand, R. Taylor

Slide credit: Robert Grupp

Deformable Registration

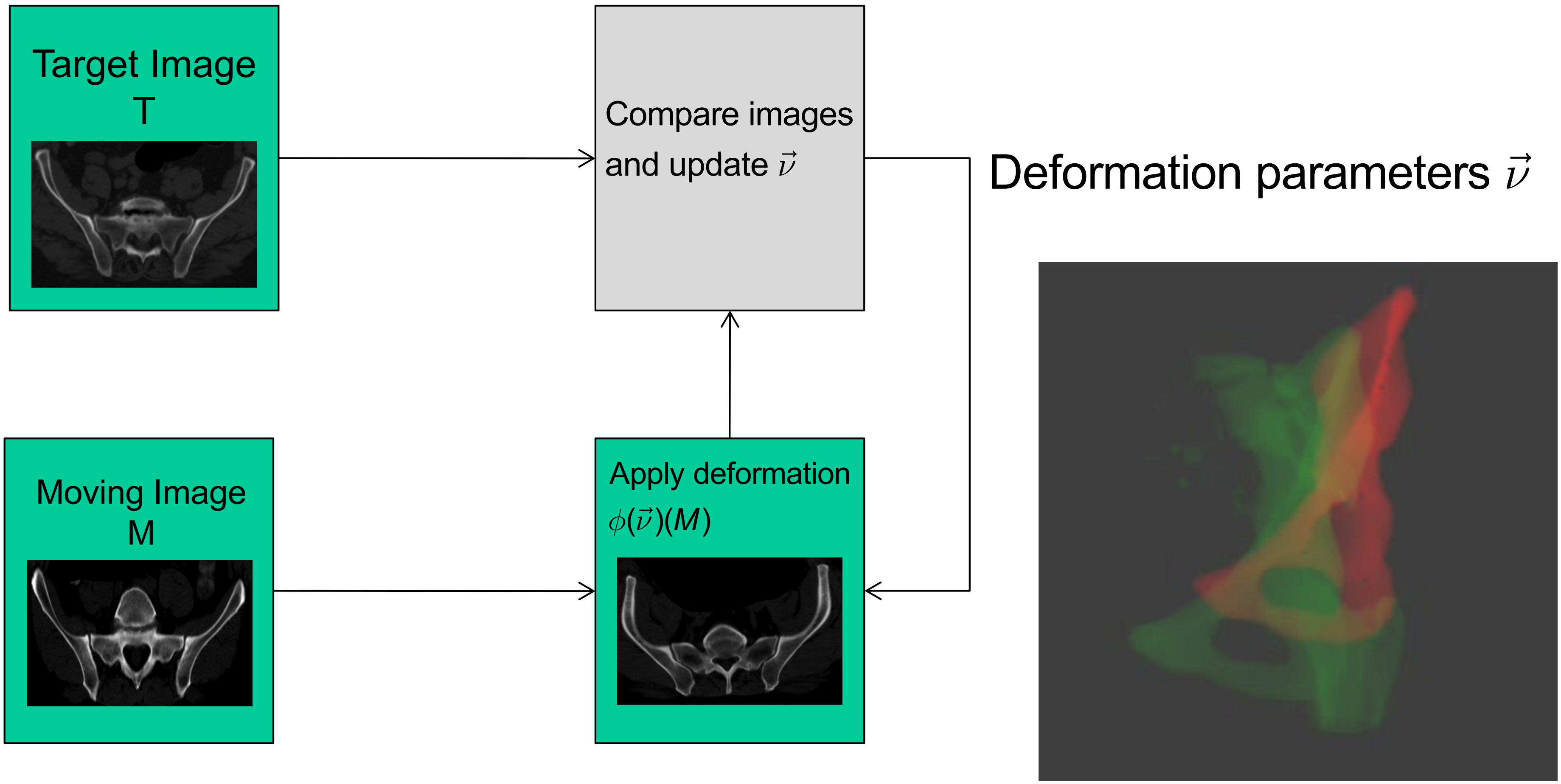


$\mathrm{Im}(\vec{\mathbf{x}})$ $\qquad$ $\mathrm{Im}(\Phi(\vec{\rho},\vec{\mathbf{x}}))$

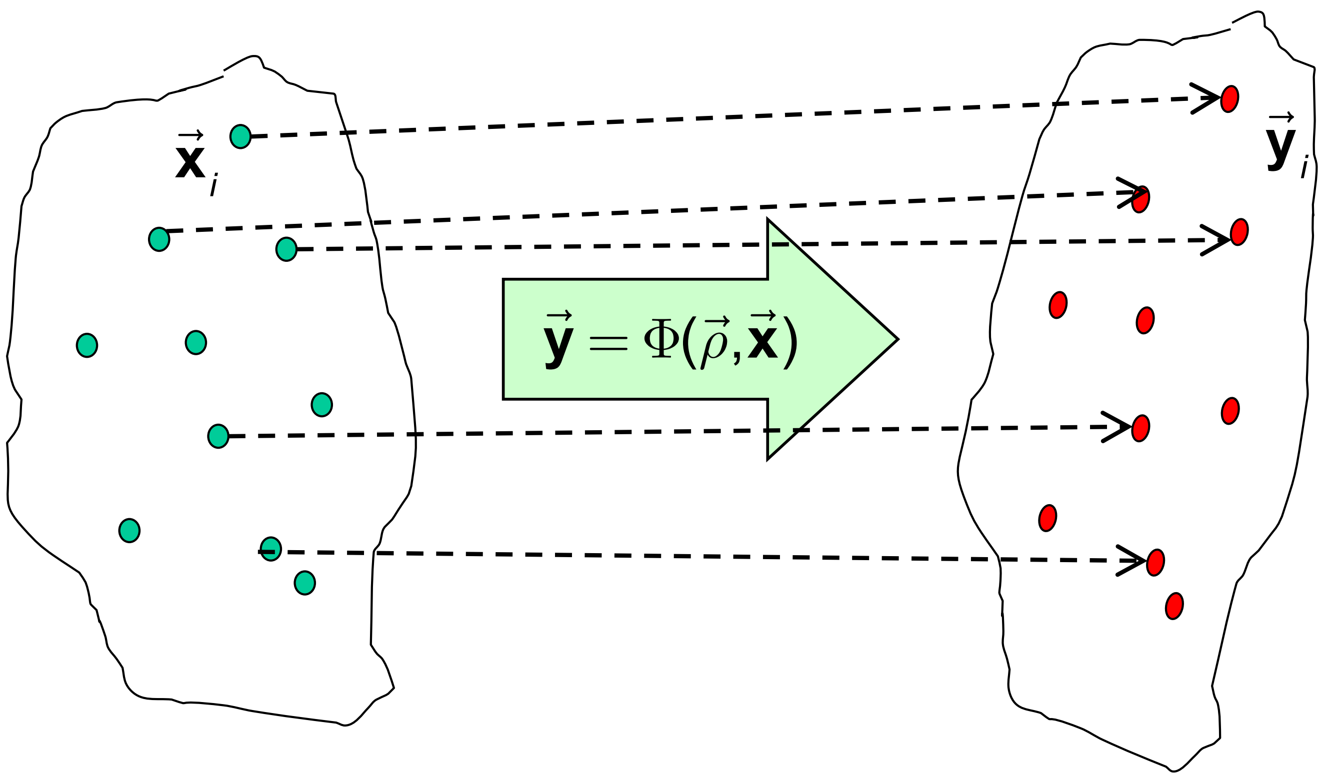
- Many different ways to parameterize the deformation function
  - Typically some version of a spline or radial basis function
- One desirable (though not universal) property: diffeomorphism
  - A function Φ is diffeomorphic if $\Phi$is bijective an both Φ and Φ^{-1} are smooth

Images: Tom Fletcher

Deformable Registration



Deformable Registration from Point Cloud Matches



Suppose that we have a bunch of corresponding point locations between an initial shape and a deformed shape. How can we use these point matches to compute a general deformation?

Images: Tom Fletcher

Deformable warping from point cloud matches

- One answer might make use of what we learned in programming assignments
  - E.g., fit Bernstein or B-spline polynomials to determine distortion.

$$\vec{\mathbf{u}} = TrimToBox(\vec{\mathbf{x}})$$

$$\vec{\mathbf{y}} = \sum_{i,j,k} \vec{\mathbf{c}}_{i,j,k} B_i(u_x) B_j(u_y) B_k(u_z)$$

or

$$\vec{\mathbf{y}} = \sum_{i,j,k} \vec{\mathbf{c}}_{i,j,k} N_i(u_x) N_j(u_y) N_k(u_z)$$

  - Note: In this case, the coefficients will also parameterize the "shape"

Deformable warping from point cloud matches

- Another answer might use something like “thin plate splines” (e.g. Bookstein)

$$TPS(\vec{\mathbf{v}};\vec{\mathbf{a}},\mathbf{B},\mathbf{C},\mathbf{P}) = \vec{\mathbf{a}} + \mathbf{B}\bullet\vec{\mathbf{v}} + \sum_i \vec{\mathbf{c}}_i U(\|\vec{\mathbf{v}} - \vec{\mathbf{p}}_i\|)$$

$$\text{where } U(r) = r^2 \log(r)$$

- Thin plate splines are multidimensional analogues of 1-dimensional spline curves.
- **NOTE:** One might also use other radial basis functions. For compact support, one example* could be

$$\Psi(r,\sigma) = \begin{cases} \mathbf{I}\left(1-\frac{r}{\sigma}\right)^{k+1+\lfloor d/2 \rfloor} & \text{if } 0 \le r \le \sigma \\ 0 & \text{otherwise} \end{cases}$$

* See: M. Fornefett, K. Rohr, and H. S. Stiehl, "Radial basis functions with compact support for elastic registration of medical images", *Image and Vision Computing,* vol. 19- 1,Äì2, pp. 87-96, 2001. http://www.sciencedirect.com/science/article/pii/S0262885600000573 http://dx.doi.org/10.1016/S0262-8856(00)00057-3

Radial Basis Functions

Given a scalar function $\phi(\bullet)$ and a set of sample points $\vec{\mathrm{p}}_k$ with associated deformations $\vec{\mathbf{d}}_k$, one can represent the deformation Φ at a point $\vec{\mathbf{x}}$ by

$$\Phi(\vec{\mathbf{x}}) = \sum_k \vec{\mathbf{d}}_k \phi_k\left(\left\|\vec{\mathbf{x}} - \vec{\mathbf{p}}_k\right\|\right)$$

- Many possible functions to use for φ
  - Common choices include Gaussians and "thin plate splines", which have non-compact support (i.e., Φ(y)>0 for arbitrarily large y)
  - Others have compact support (i.e., Φ(y)=0 for |y|> some value)*

* See: M. Fornefett, K. Rohr, and H. S. Stiehl, "Radial basis functions with compact support for elastic registration of medical images", *Image and Vision Computing,* vol. 19- 1,Äì2, pp. 87-96, 2001. http://www.sciencedirect.com/science/article/pii/S0262885600000573 http://dx.doi.org/10.1016/S0262-8856(00)00057-3

Thin Plate Splines Digression

- Some citations (from G. Donato and S. Belongie, "Approximation Methods for Thin Plate Spline Mappings and Principal Warps", 2002;

  http://www.cs.ucsd.edu/Dienst/UI/2.0/Describe/ncstrl.ucsd_cse/CS2003-0764)

[1] C. T. H. Baker. *The numerical treatment of integral equations*. Oxford: Clarendon Press, 1977.

[2] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Proc. 8th Int'l. Conf. Computer Vision*, volume 1, pages 454–461, July 2001.

[3] F. L. Bookstein. Principal warps: thin-plate splines and decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

[4] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 44–51, June 2000.

[5] M.H. Davis, A. Khotanzad, D. Flamig, and S. Harms. A physics-based coordinate transformation for 3-d image matching. *IEEE Trans. Medical Imaging*, 16(3):317–328, June 1997.

[6] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

[7] M. J. D. Powell. A thin plate spline method for mapping curves into curves in two dimensions. In *Computational Techniques and Applications (CTAC95)*, Melbourne, Australia, 1995.

[8] A.J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *ICML*, 2000.

[9] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[10] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 520–526, 1997.

[11] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, pages 682–688, 2001.

M-dimensional Thin Plate Spline Summary

Given

$$TPS(\vec{\mathbf{v}};\vec{\mathbf{a}},\mathbf{B},\mathbf{C},\mathbf{P}) = \vec{\mathbf{a}} + \mathbf{B}\bullet\vec{\mathbf{v}} + \sum_i \vec{\mathbf{c}}_i U(\|\vec{\mathbf{v}}-\vec{\mathbf{p}}_i\|)$$

where

$$U(r) = r^2\log(r) \quad \text{for 2D}$$

$$= r^2\log(r^2) \quad \text{for 3D}$$

$$\vec{\mathbf{v}} = [v_1,\cdots,v_M]^T$$

$$\vec{\mathbf{p}}_i = [p_1,\cdots,p_M]_i^T$$

$$\mathbf{P} = [\vec{\mathbf{p}}_1,\cdots,\vec{\mathbf{p}}_N]^T$$

$$\mathbf{C} = [\vec{\mathbf{c}}_1,\cdots,\vec{\mathbf{c}}_N]$$

$$\mathbf{B} = [\vec{\mathbf{b}}_1,\cdots,\vec{\mathbf{b}}_M]$$

M-dimensional Thin Plate Spline Fitting

Given

$$\mathbf{V} = \left[\vec{\mathbf{v}}_1, \cdots, \vec{\mathbf{v}}_N\right] \quad \mathbf{F} = \left[\vec{\mathbf{f}}_1, \cdots, \vec{\mathbf{f}}_N\right]$$

find $\vec{\mathbf{a}}$, $\mathbf{B}$,$\mathbf{C}$ such that

$$\vec{\mathbf{f}}_i = TPS(\vec{\mathbf{v}}_i;\ \vec{\mathbf{a}}, \mathbf{B}, \mathbf{C}, \mathbf{V})$$

To do this, solve the linear system

$$\begin{bmatrix} \mathbf{K}_{[NxN]} & \vec{\mathbf{1}}_{[N\times 1]} & \mathbf{V} \\ \vec{\mathbf{1}}_{[1\times N]} & 0 & 0 \\ \mathbf{V}^T & 0 & \mathbf{0}_{[M\times M]} \end{bmatrix} \begin{bmatrix} \mathbf{C}^T \\ \vec{\mathbf{a}}^T \\ \mathbf{B}^T \end{bmatrix} = \begin{bmatrix} \mathbf{F}^T \\ 0 \\ \mathbf{0}_{[M\times 1]} \end{bmatrix}$$

where

$$\mathbf{K}_{i,j} = \mathbf{K}_{j,i} = U\left(\left\|\vec{\mathbf{v}}_i - \vec{\mathbf{v}}_j\right\|\right) \quad \text{with } U(r) = r^2 \log r \text{ or } U(r) = r^2 \log r^2$$

$$\mathbf{K}_{i,j} = \left(\vec{\mathbf{v}}_i - \vec{\mathbf{v}}_j\right) \bullet \left(\vec{\mathbf{v}}_i - \vec{\mathbf{v}}_j\right) \log\left(\sqrt{\left(\vec{\mathbf{v}}_i - \vec{\mathbf{v}}_j\right) \bullet \left(\vec{\mathbf{v}}_i - \vec{\mathbf{v}}_j\right)}\right)$$

TPS 2D case

Given a set of points $\vec{\mathbf{p}}_i = \left[x_i, y_i\right]$ and corresponding points $\vec{\mathbf{p}}_i{}^* = \left[x_i{}^*, y_i{}^*\right]$,

we want to find TPS parameters such that $\vec{\mathbf{p}}_i{}^* = TPS(\vec{\mathbf{p}}_i; \vec{\mathbf{a}}, \mathbf{B}, \mathbf{C}, \mathbf{P})$

To do this, we solve the least squares problem

$$
\begin{bmatrix}
0 & \cdots & U_{1,k} & \cdots & U_{1,N} & 1 & x_1 & y_1 \\
\vdots & \ddots & & U_{ij} & & \vdots & \vdots & \vdots \\
U_{k,1} & \cdots & 0 & \cdots & U_{k,N} & 1 & x_k & y_k \\
\vdots & U_{ij} & & \ddots & \vdots & \vdots & \vdots & \vdots \\
U_{N,1} & \cdots & U_{N,k} & \cdots & 0 & 1 & x_N & y_N \\
1 & \cdots & 1 & \cdots & 1 & 0 & 0 & 0 \\
x_1 & \cdots & x_k & \cdots & x_N & 0 & 0 & 0 \\
y_1 & \cdots & y_k & \cdots & y_N & 0 & 0 & 0
\end{bmatrix}
\bullet
\begin{bmatrix}
\vec{\mathbf{c}}_1 \\ \vdots \\ \vdots \\ \vdots \\ \vec{\mathbf{c}}_N \\ \vec{\mathbf{a}} \\ \vec{\mathbf{b}}_x \\ \vec{\mathbf{b}}_y
\end{bmatrix}
=
\begin{bmatrix}
\vec{\mathbf{p}}_1{}^* \\ \vdots \\ \vec{\mathbf{p}}_k{}^* \\ \vdots \\ \vec{\mathbf{p}}_N{}^* \\ \vec{\mathbf{0}} \\ \vec{\mathbf{0}} \\ \vec{\mathbf{0}}
\end{bmatrix}
$$

where $U_{i,j} = U_{j,i} = U(\|\vec{\mathbf{p}}_i - \vec{\mathbf{p}}_j\|)$

M-dimensional Thin Plate Spline Fitting

Define

$$\mathbf{L}_{[M+N+1\times M+N+1]} = \begin{bmatrix} \mathbf{K}_{[NxN]} & \vec{\mathbf{1}}_{[N\times 1]} & \mathbf{V} \\ \vec{\mathbf{1}}_{[1\times N]} & 0 & 0 \\ \mathbf{V}^T & 0 & \mathbf{0}_{[M\times M]} \end{bmatrix}$$

If there are many points, this matrix may be expensive to invert or even pseudo-invert. There are various methods to deal with this problem. These include

- Use a random sample of the $\vec{\mathbf{v}}_i$ to approximate the solution
- Use a random sample of the basis functions & all data to solve problem in least squares sense
- Use matrix approximation methods

See

http://www.cs.ucsd.edu/Dienst/UI/2.0/Describe/ncstrl.ucsd_cse/CS2003-0764

Further Digression: Radial Basis Functions

Note that the function $U(r)$ in the previous discussion is a an example of a more general class of "radial basis functions". These functions can be used in deformable registration in much the same way as the TPS function used above. Other commonly used radial basis functions include

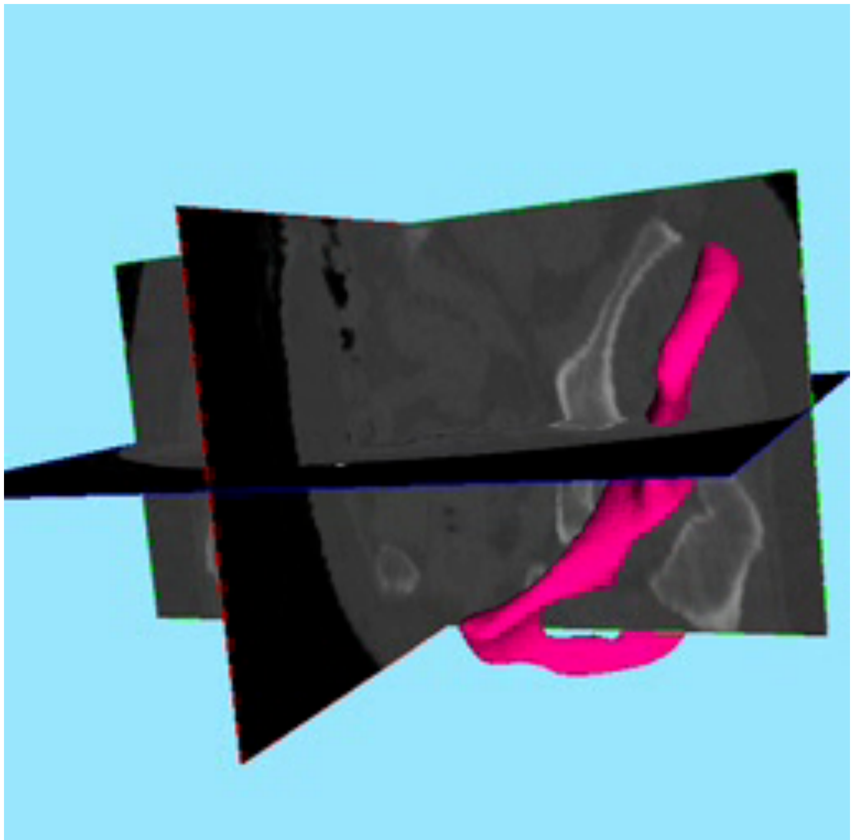
$$U(r) = (r^2 + c^2)^{\mu} \text{ for } \mu \in \mathbb{R}_+$$

$$U(r) = (r^2 + c^2)^{-\mu} \text{ for } \mu \in \mathbb{R}_+$$

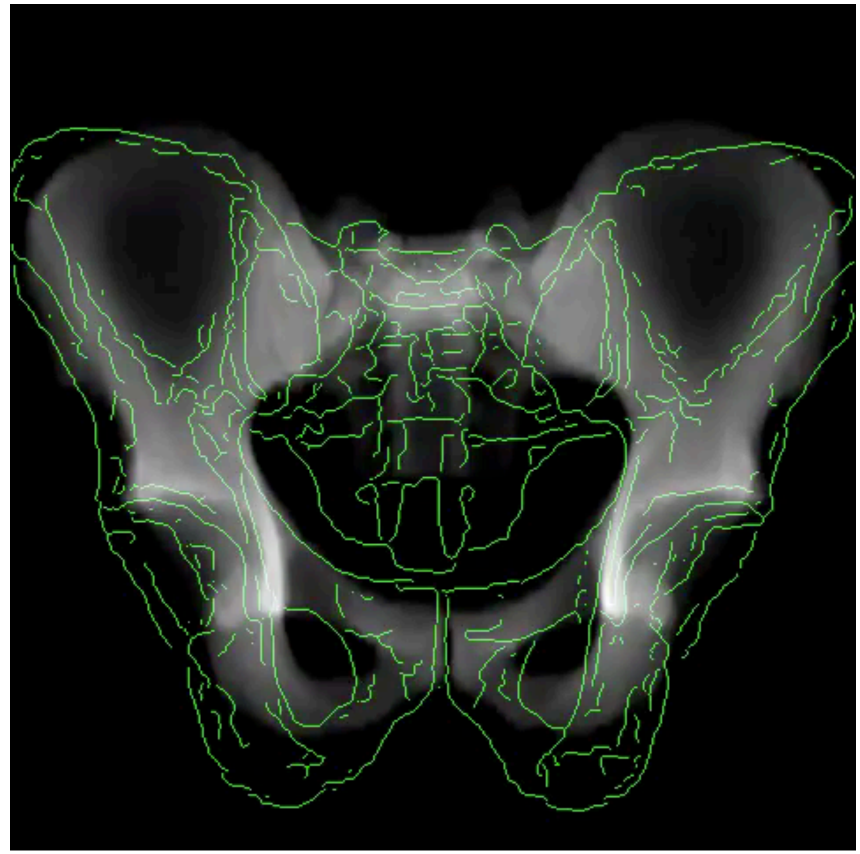
$$U(e) = e^{-r^2/2\sigma^2}$$

The last one is probably the most popular

Deformable Registration to Statistical “Atlases”



Deformable 3D/3D
Jianhua Yao



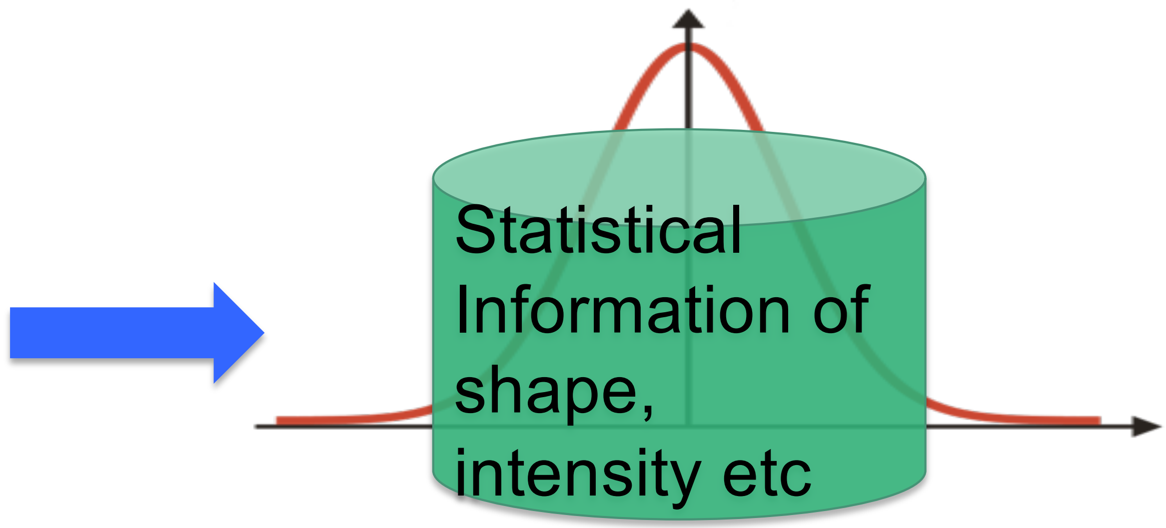
Deformable 2D/3D
Ofri Sadowsky

Deformable Altas-based Registration

- Much of the material that follows is derived from the Ph.D. thesis work of J. Yao, Ofri Sadowsky, and Gouthami Chintalapani:
  - J. Yao, “Statistical bone density atlases and deformable medical image registrations”, Ph. D. Thesis, Computer Science, The Johns Hopkins University, Baltimore, 2001.
  - O. Sadowsky, "Image Registration and Hybrid Volume Reconstruction of Bone Anatomy Using a Statistical Shape Atlas,” Ph.D. Thesis, Computer Science, The Johns Hopkins University, Baltimore, 2008
  - G. Chintalapani, Statistical Atlases of Bone Anatomy and Their Applications, Ph.D. thesis in Computer Science, The Johns Hopkins University, Baltimore, Maryland, 2010.
- A number of other authors, including
  - Cootes et al. 1999 – “Active Appearance Models”
  - Feldmar and Ayache 1994
  - Ferrant et al. 1999
  - Fleute and Lavallee 1999
  - Lowe 1991
  - Maurer et al. 1996
  - Shen and Davatzikos 2000

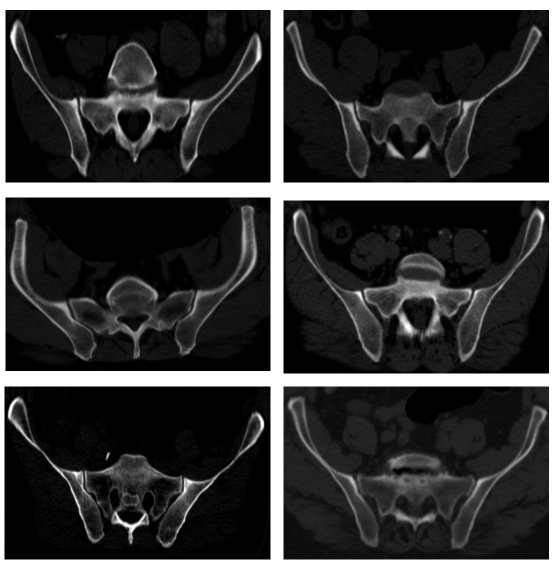
What is a "Statistical Atlas" ?

- An atlas that incorporates **statistics of anatomical shape and intensity variations** of a given population



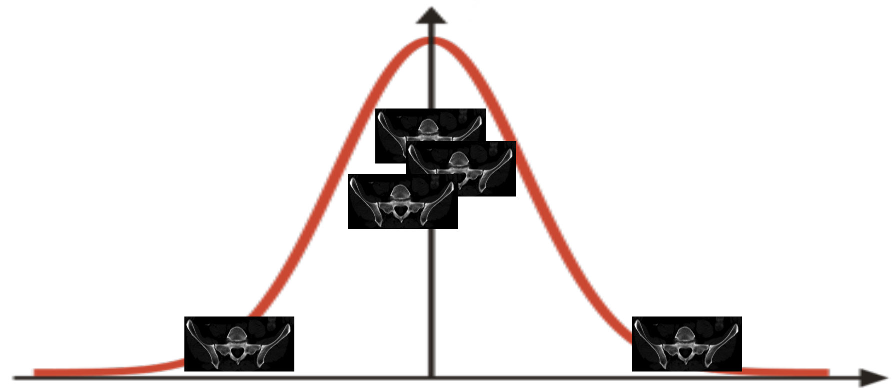
Credit: G. Chintalapani 2010

Statistical Atlases

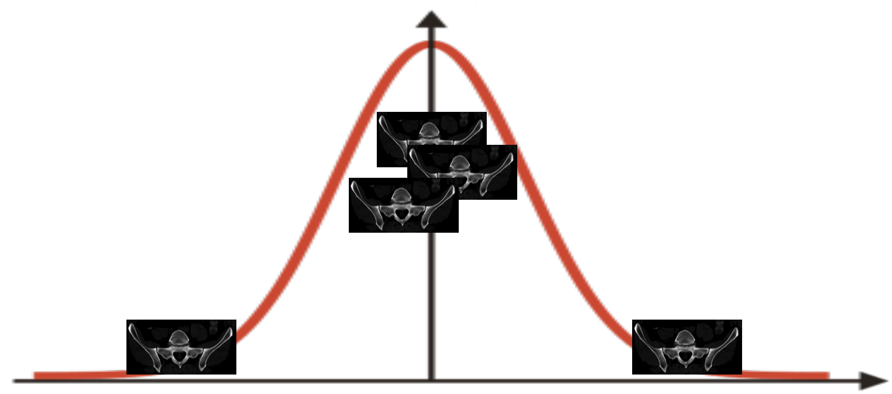


CT scans from a population

??



Shape distribution



Intensity distribution

Slide Credit: G. Chintalapani 2010

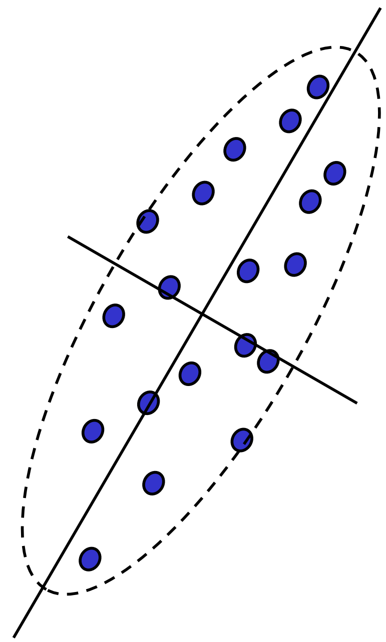
Statistical models

- The next few slides will review the use of the Singular Value Decomposition (SVD) in constructing statistical shape models.
- There is a close relationship between this material and the "principal components analysis" (PCA) methods you may have encountered in a statistics class.

Principal Components Analysis (PCA)

Suppose that you have a set of N vectors $\vec{\mathbf{a}}_i$ in an M dimensional space?

Is there a natural "coordinate system" for these vectors?



Principal Components Analysis (PCA)

We proceed as follows

$$\vec{\mathbf{a}}^{(avg)} = \frac{\sum_i \vec{\mathbf{a}}_i}{N}; \quad \vec{\mathbf{b}}_i = \vec{\mathbf{a}}_i - \vec{\mathbf{a}}^{(avg)}; \quad \mathbf{B} = \left[\vec{\mathbf{b}}_1, \cdots \vec{\mathbf{b}}_N\right];$$

Then form the singular value decomposition

$$\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^T = \mathbf{U}\begin{bmatrix} \Sigma^{(N)} \\ \mathbf{0} \end{bmatrix}\mathbf{V}^T \quad \text{where } \Sigma^{(N)} = diag(\sigma_1, \cdots, \sigma_N)$$

Then we note that $\mathbf{M} = \mathbf{U}\Sigma^2\mathbf{U}^T$. Of course $\mathbf{U}$ is huge, but we have the following useful fact. We note that

$$\mathbf{B} = \left[\vec{\mathbf{u}}_1, \cdots, \vec{\mathbf{u}}_N, \vec{\mathbf{u}}_{N+1}, \cdots, \vec{\mathbf{u}}_M\right]\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_N \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \end{bmatrix}\mathbf{V}^T = \left[\vec{\mathbf{u}}_1, \cdots, \vec{\mathbf{u}}_N\right]\Sigma^{(N)}\mathbf{V}^T = \mathbf{U}^{(N)}\Sigma^{(N)}\mathbf{V}^T$$

Principal Components Analysis (PCA)

This means that any column $\vec{\mathbf{b}}_k$ of **B** may be expressed as a linear combination of the first N columns of **U**

$$\mathbf{B} = \left[\vec{\mathbf{u}}_1, \cdots, \vec{\mathbf{u}}_N\right] \Sigma^{(N)} \mathbf{V}^T = \mathbf{U}^{(N)} \Sigma^{(N)} \mathbf{V}^T$$

$$\vec{\mathbf{b}}_k = \lambda_1^{(k)} \vec{\mathbf{u}}_1 + \cdots + \lambda_N^{(k)} \vec{\mathbf{u}}_N = \mathbf{U}^{(N)} \Lambda^{(k)}$$

where

$$\Lambda^{(k)} = \mathit{transpose}(\mathbf{U}^{(N)}) \vec{\mathbf{b}}_k$$

So

$$\vec{\mathbf{a}}_k = \vec{\mathbf{a}}^{(avg)} + \vec{\mathbf{b}}_k = \vec{\mathbf{a}}^{(avg)} + \lambda_1^{(k)} \vec{\mathbf{u}}_1 + \cdots + \lambda_N^{(k)} \vec{\mathbf{u}}_N$$

But often the last few values of the λ_k are small. If we ignore all but the first D values, we have

$$\vec{\mathbf{a}}_k \approx \vec{\mathbf{a}}^{(avg)} + \lambda_1^{(k)} \vec{\mathbf{u}}_1 + \cdots + \lambda_D^{(k)} \vec{\mathbf{u}}_D$$

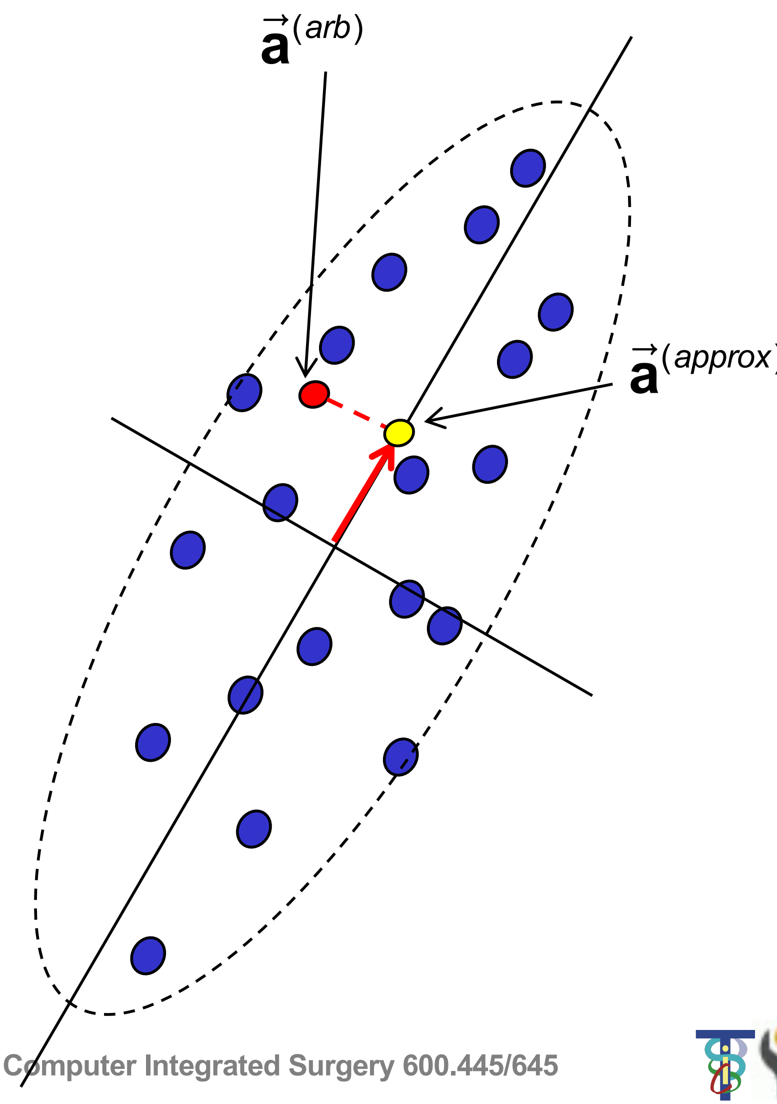
Principal Components Analysis (PCA)

Suppose now that we have an arbitrary $\vec{\mathbf{a}}^{(arb)}$. We can approximate $\vec{\mathbf{a}}^{(arb)}$ as follows:

$$\vec{\mathbf{b}}^{(arb)} = \vec{\mathbf{a}}^{(arb)} - \vec{\mathbf{a}}^{(avg)}$$

$$\Lambda^{(arb)} = transpose(\mathbf{U}^{(D)})\vec{\mathbf{b}}^{(arb)}$$

$$\vec{\mathbf{a}}^{(arb)} \approx \vec{\mathbf{a}}^{(avg)} + \lambda_1^{(arb)}\vec{\mathbf{u}}_1 + \cdots + \lambda_D^{(arb)}\vec{\mathbf{u}}_D$$



Statistical Atlases & PCA

Given a set of N models $\vec{\mathbf{X}}^{(j)} = \left[\vec{\mathbf{x}}_k^{(j)}\right]^T = \left[\cdots x_k^{(j)}, y_k^{(j)}, z_k^{(j)}, \cdots\right]$, compute

$$\vec{\mathbf{X}}^{(avg)} = \begin{bmatrix} \vdots \\ \vec{\mathbf{x}}_k^{(avg)} \\ \vdots \end{bmatrix}$$ where $\vec{\mathbf{x}}_k^{(avg)} = \frac{1}{N}\sum_j \vec{\mathbf{x}}_k^{(j)}$ and the differences

$$\vec{\mathbf{D}}^{(j)} = \vec{\mathbf{X}}^{(j)} - \vec{\mathbf{X}}^{(avg)} = \begin{bmatrix} \vdots \\ \vec{\mathbf{d}}_k^{(j)} \\ \vdots \end{bmatrix}$$ where $\vec{\mathbf{d}}_k^{(j)} = \vec{\mathbf{x}}_k^{(j)} - \vec{\mathbf{x}}_k^{(avg)}$. Create the matrix

$$\mathbf{D} = \left[\cdots \quad \vec{\mathbf{D}}^{(j)} \quad \cdots\right]_{[3Nvertices \times N]} = \begin{bmatrix} \vec{\mathbf{d}}_1^{(1)} & \cdots & \vec{\mathbf{d}}_k^{(1)} & \cdots & \vec{\mathbf{d}}_1^{(1)} \\ \vdots & & \vdots & & \vdots \\ \vec{\mathbf{d}}_k^{(1)} & \cdots & \vec{\mathbf{d}}_k^{(j)} & \cdots & \vec{\mathbf{d}}_k^{(N)} \\ \vdots & & \vdots & & \vdots \\ \vec{\mathbf{d}}_{Nvertices}^{(1)} & \cdots & \vec{\mathbf{d}}_{Nvertices}^{(j)} & \cdots & \vec{\mathbf{d}}_{Nvertices}^{(N)} \end{bmatrix}$$

Statistical Atlases & PCA

Compute the singular value decomposition of **D**

$$\mathbf{D} = \mathbf{U}\Sigma\mathbf{V}^T \qquad \text{where } \Sigma = \begin{bmatrix} diag(\vec{\sigma}) \\ \mathbf{0} \end{bmatrix}.$$

$$\mathbf{D} = \mathbf{U}\begin{bmatrix} diag(\vec{\sigma})\mathbf{V}^T \\ \mathbf{0} \end{bmatrix}$$

Note that

$$\frac{1}{N}\mathbf{D}^T\mathbf{D} = \frac{1}{N}\mathbf{V}\Sigma\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T = \frac{1}{N}\mathbf{V}\Sigma^2\mathbf{V}^T$$

$$\frac{1}{N}\mathbf{D}\mathbf{D}^T = \frac{1}{N}\mathbf{U}\Sigma\mathbf{V}^T\mathbf{V}\Sigma\mathbf{U}^T = \frac{1}{N}\mathbf{U}\Sigma^2\mathbf{U}^T$$

Statistical Atlases & PCA

Any individual model $\mathbf{D}^{(j)}$ can be written as a linear combination of the columns of **U**. Treating $\vec{\mathbf{D}}^{(j)}$ as a column vector, we can write this as

$$\overrightarrow{\mathbf{D}}^{(j)} = \mathbf{U} \bullet \begin{bmatrix} \lambda_1^{(j)} \\ \vdots \\ \lambda_N^{(j)} \\ \vec{\mathbf{0}} \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} \lambda_1^{(j)} \\ \vdots \\ \lambda_N^{(j)} \\ \vec{\mathbf{0}} \end{bmatrix} \text{ is the } j^{th} \text{ column of } \begin{bmatrix} diag(\vec{\sigma})\mathbf{V}^T \\ \mathbf{0} \end{bmatrix}$$

If we define

$$\mathbf{M} = \begin{bmatrix} \mathbf{U}^{(1)} & \cdots & \mathbf{U}^{(N)} \end{bmatrix} \text{ (i.e., the first } N \text{ columns of } \mathbf{U}\text{)}$$

we get the expression

$$\overrightarrow{\mathbf{D}}^{(j)} = \mathbf{M}\vec{\lambda} \text{ where } \vec{\lambda} \text{ is the } j^{th} \text{ column of } \left(diag(\vec{\sigma})\mathbf{V}^T \right).$$

Statistical Atlases & PCA

Note that while **U** is $3N_{vertices} \times 3N_{vertices}$ (i.e., huge), **M** has only the first N columns, since there are at most N non-zero singular values

In fact, we usually also truncate even more, only saving columns corresponding to relatively large singular values σ_{i}. Since the standard algorithms for SVD produce positive singular values σ_{i} sorted in descending order, this is easy to do.

Note also, that since the columns of **M** are also columns of **U**, they are orthogonal. Hence $\mathbf{M}^T\mathbf{M} = \mathbf{I}_{N\times N}$. But $\mathbf{M}\mathbf{M}^T = \mathbf{C}$ will be an $3N_{vertices} \times 3N_{vertices}$ matrix that will not in general be diagonal.

Statistical Atlases & PCA

As a practical matter, it is not a good idea to ask your SVD program to produce the full matrix **U** for an $3N_{vertices} \times N$ matrix **D**. Most SVD packages give you the option to compute only the singular values $\vec{\sigma}$ and the right hand side matrix **V** or its transpose. Then, **M** can be computed from

$$\mathbf{M}\,diag(\vec{\sigma})\mathbf{V}^T = \mathbf{D}$$

$$\mathbf{M}\,diag(\vec{\sigma}) = \mathbf{DV}$$

$$\mathbf{M} = \mathbf{DV}\,diag(\vec{\sigma})^{-1}$$

$$= \mathbf{DV}\begin{bmatrix} 1/\sigma_1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & & & \vdots \\ \vdots & & 1/\sigma_k & & \vdots \\ \vdots & & & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1/\sigma_N \end{bmatrix}$$

Statistical Atlases & PCA

Similarly, given a vector $\vec{\mathbf{D}}^{(\text{inst})}$ we can find a corresponding vector $\vec{\lambda}^{(\text{inst})}$ from the following

$$\begin{aligned}\vec{\mathbf{D}}^{(\text{inst})} &= \mathbf{M}\vec{\lambda}^{(\text{inst})} \\ \mathbf{M}^T\vec{\mathbf{D}}^{(\text{inst})} &= \mathbf{M}^T\mathbf{M}\vec{\lambda}^{(\text{inst})} \\ &= \vec{\lambda}^{(inst)}\end{aligned}$$

Statistical Atlases & PCA

Suppose that we select $\vec{\lambda} = [\lambda_1, \cdots, \lambda_N]^T$ as a random variable with some distribution having expected value $E(\vec{\lambda}) = \vec{\mathbf{0}}$ and covariance

$$\operatorname{cov}(\vec{\lambda}) = E(\vec{\lambda} \bullet \vec{\lambda}^T) = \begin{bmatrix} E(\lambda_1^{\;2}) & \cdots & E(\lambda_1 \lambda_N) \\ \vdots & \ddots & \vdots \\ E(\lambda_N \lambda_1) & \cdots & E(\lambda_N^{\;2}) \end{bmatrix} = \Sigma^2$$

and compute a corresponding random model $\vec{\mathbf{X}}(\vec{\lambda})$

$$\vec{\mathbf{X}}(\vec{\lambda}) = \vec{\mathbf{X}}^{(avg)} + \mathbf{M} \bullet \vec{\lambda}$$

What can we say about the expected value and covariance of $\vec{\mathbf{X}}(\vec{\lambda})$?

Statistical Atlases & PCA

For the expected value, we have

$$\begin{aligned} E(\vec{\mathbf{X}}(\vec{\lambda})) &= E(\vec{\mathbf{X}}^{(avg)} + \mathbf{M} \bullet \vec{\lambda}) \\ &= \vec{\mathbf{X}}^{(avg)} + \mathbf{M} \bullet E(\vec{\lambda}) = \vec{\mathbf{X}}^{(avg)} + \mathbf{M} \bullet \vec{\mathbf{0}} \\ &= \vec{\mathbf{X}}^{(avg)} \end{aligned}$$

Then

$$\begin{aligned} \mathrm{cov}(\vec{\mathbf{X}}(\vec{\lambda})) &= E(\vec{\mathbf{D}}(\vec{\lambda}) \bullet \vec{\mathbf{D}}(\vec{\lambda})^T) \quad \text{where} \quad \vec{\mathbf{D}}(\vec{\lambda}) = \vec{\mathbf{X}}(\vec{\lambda}) - \vec{\mathbf{X}}^{(avg)} \\ &= E(\mathbf{M} \bullet \vec{\lambda} \bullet \vec{\lambda}^T \bullet \mathbf{M}) \\ &= \mathbf{M} \bullet E(\vec{\lambda} \bullet \vec{\lambda}^T) \bullet \mathbf{M}^{\mathbf{T}} \\ &= \mathbf{M} \bullet \Sigma^2 \bullet \mathbf{M}^{\mathbf{T}} \end{aligned}$$

Statistical Atlases & PCA

Thus, if we assemble a representative sample set of models $\vec{\mathbf{X}}^{(j)}$, and compute the average model $\vec{\mathbf{X}}^{(avg)}$ and the SVD of the corresponding matrix $\mathbf{D} = \left[\cdots \left(\vec{\mathbf{X}}^{(j)} - \vec{\mathbf{X}}^{(avg)} \right) \right]$, then we have a way of generating an arbitrary number of models

$$\vec{\mathbf{X}}^{(\text{inst})} = \vec{\mathbf{X}}^{(avg)} + \mathbf{M}\vec{\lambda}^{(inst)} = \vec{\mathbf{X}}^{(avg)} + \sum_{\text{k}} \vec{\mathbf{M}}^{(k)} \lambda_k^{(inst)}$$

with the same mean and covariance. I.e., we know how the individual features $\vec{\mathbf{x}}_k^{(\text{inst})}$ co-vary.

Further, given a representative model instance $\vec{\mathbf{X}}^{(inst)}$ we can compute a corresponding set of mode weights $\vec{\lambda}^{(\text{inst})}$ from

$$\vec{\lambda}^{(\text{inst})} = \mathbf{M}^T \left(\vec{\mathbf{X}}^{(inst)} - \vec{\mathbf{X}}^{(avg)} \right)$$

Statistical Atlas

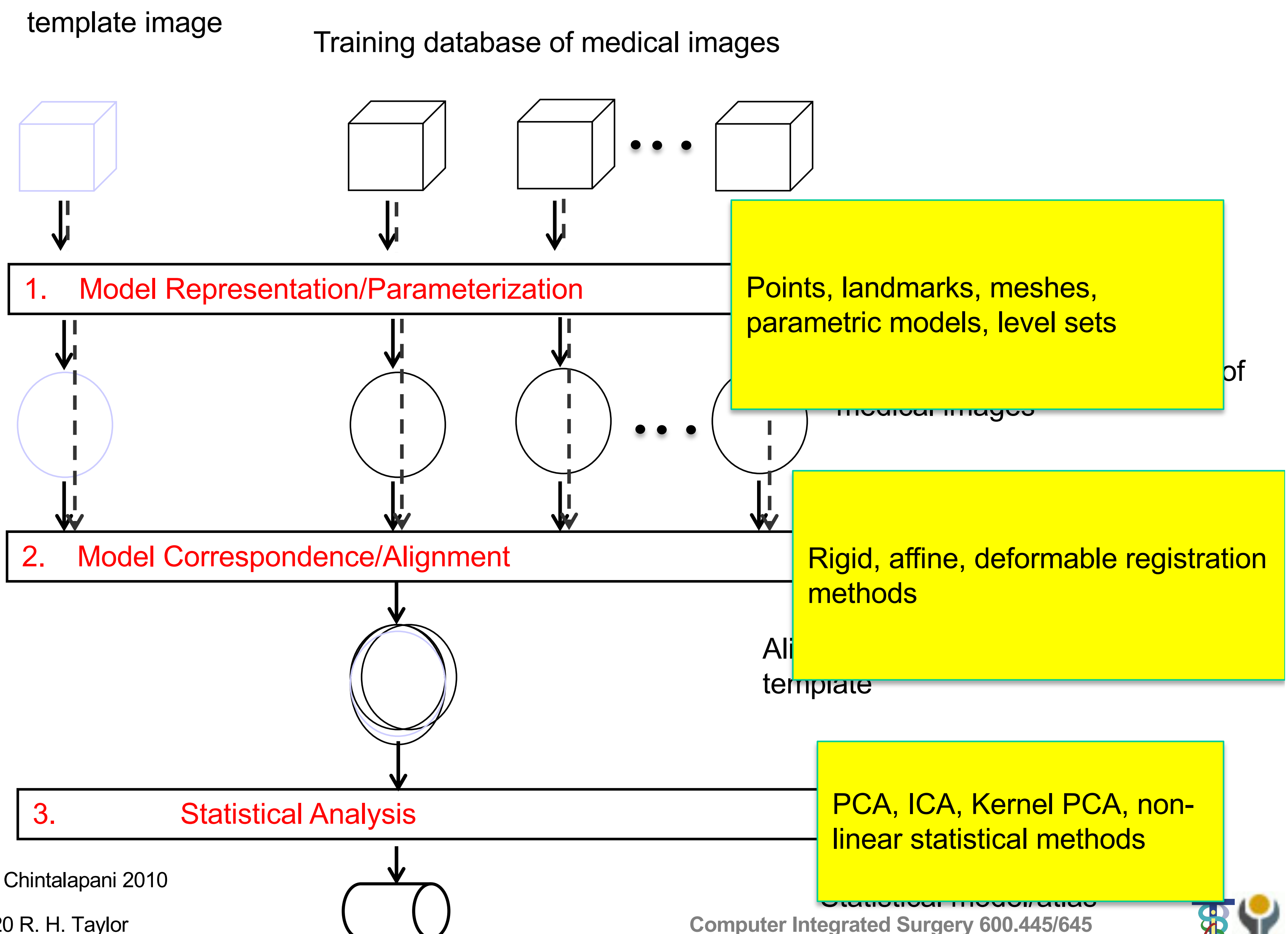
Thus, one representation of a statistical "atlas" of models consists of

- An average model $\vec{\mathbf{X}}^{(avg)}$
- An eigen matrix **M** of variation modes
- A diagonal covariance matrix Σ^2 for the modes

This information may be used in many ways, including

- Atlas-based deformable segmentation/registration
- Statistical analysis of anatomic variation
- etc.

Statistical Atlas Construction



Slide Credit: G. Chintalapani 2010

Model Representation

- Tetrahedral mesh represents shape

- Bernstein polynomials approximate CT density within each tetrahedron[1,2]

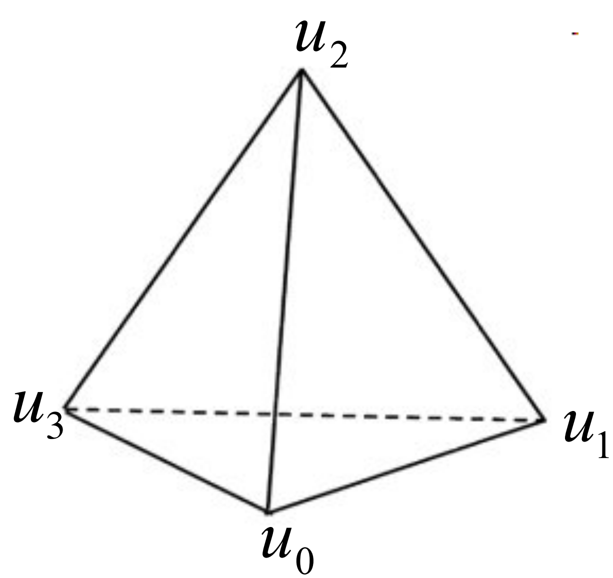
$$P^d(\mathbf{u}) = \sum_{|\mathbf{k}|=d} C_{\mathbf{k}} B_{\mathbf{k}}^d(\mathbf{u})$$

where

$$\mathbf{k} = (k_0, k_1, k_2, k_3) \quad \mathbf{u} = (u_0, u_1, u_2, u_3)$$

$$|\mathbf{k}| = k_0 + k_1 + k_2 + k_3 \qquad |\mathbf{u}| = 1$$

$$B_{\mathbf{k}}^d(\mathbf{u}) = \frac{d!}{k_0! k_1! k_2! k_3!} u_0^{k_0} u_1^{k_1} u_2^{k_2} u_3^{k_3}$$

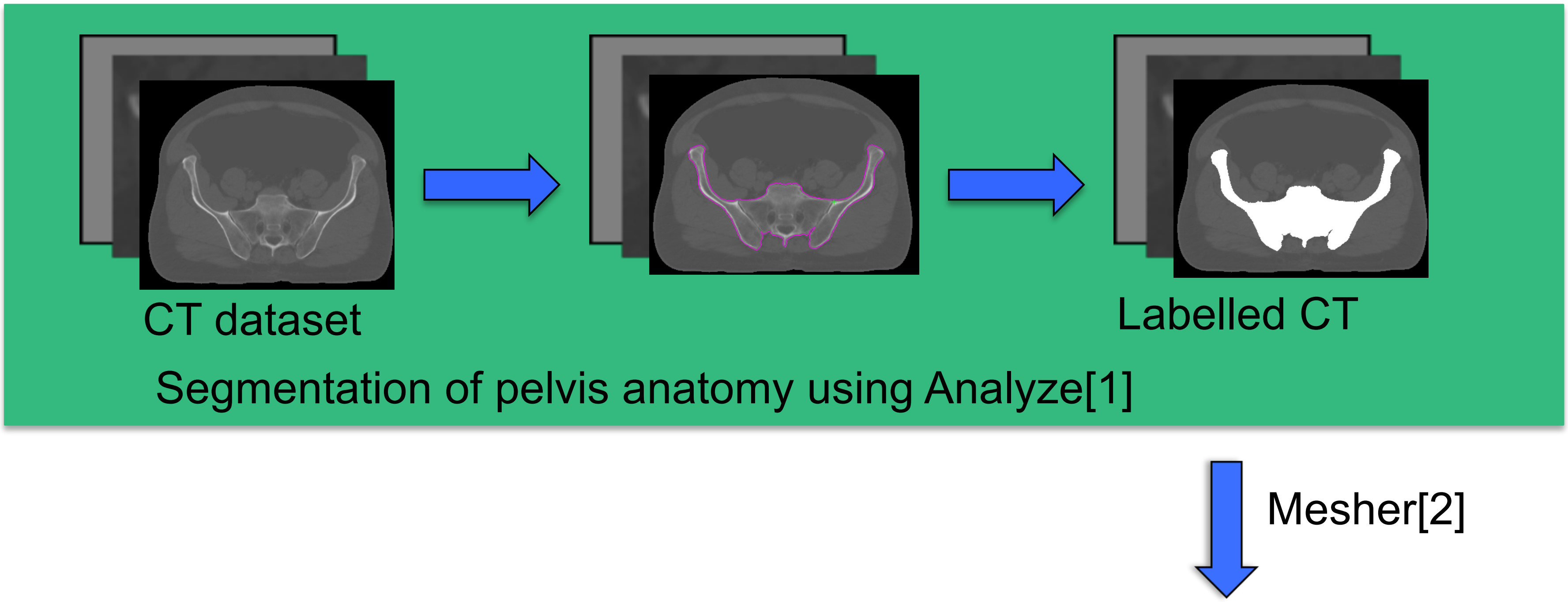


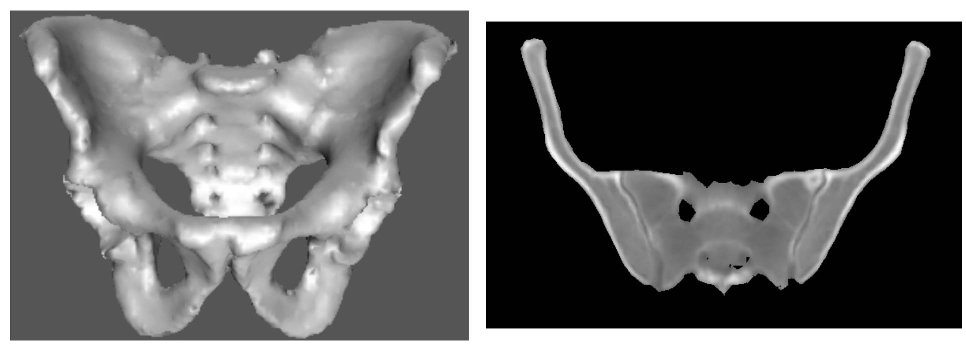
- Alternative might be to use voxels directly after deformation to mean shape

[1] Yao, PhD Thesis, 2002; [2] Sadowsky, PhD Thesis, 2008

Credit: G. Chintalapani 2010

Model Creation





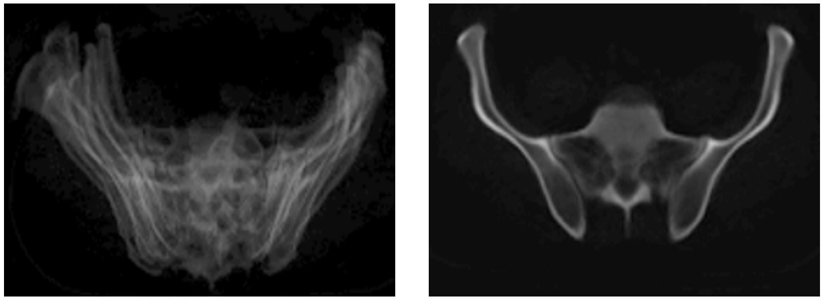
Surface rendering of pelvis tetrahedral model; Cross-section of tetrahedral model showing CT densities

[1]Analyze, www.mayoclinic.org
[2] Mohammed et al., 2005

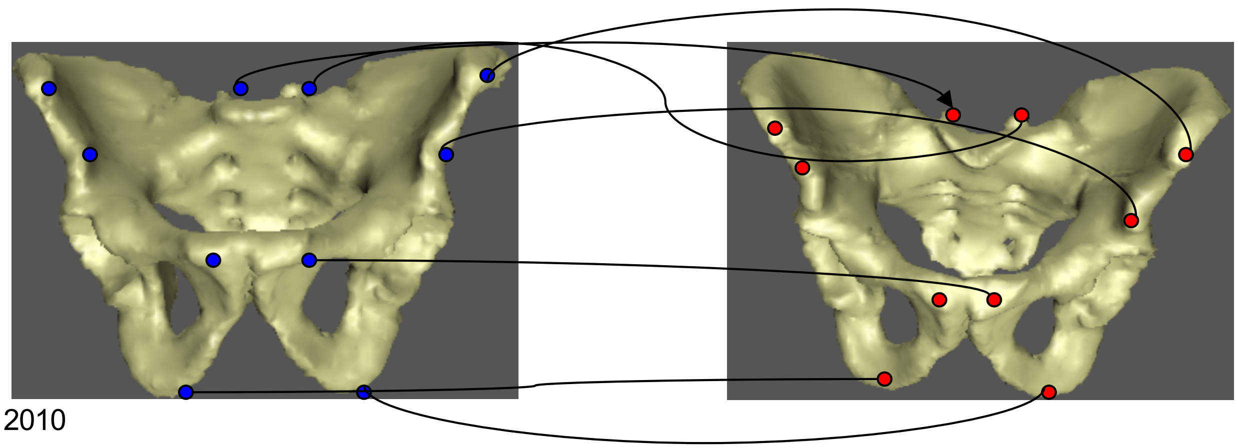
Slide Credit: G. Chintalapani 2010

Model Correspondence

- Need to establish a common coordinate frame for the training database



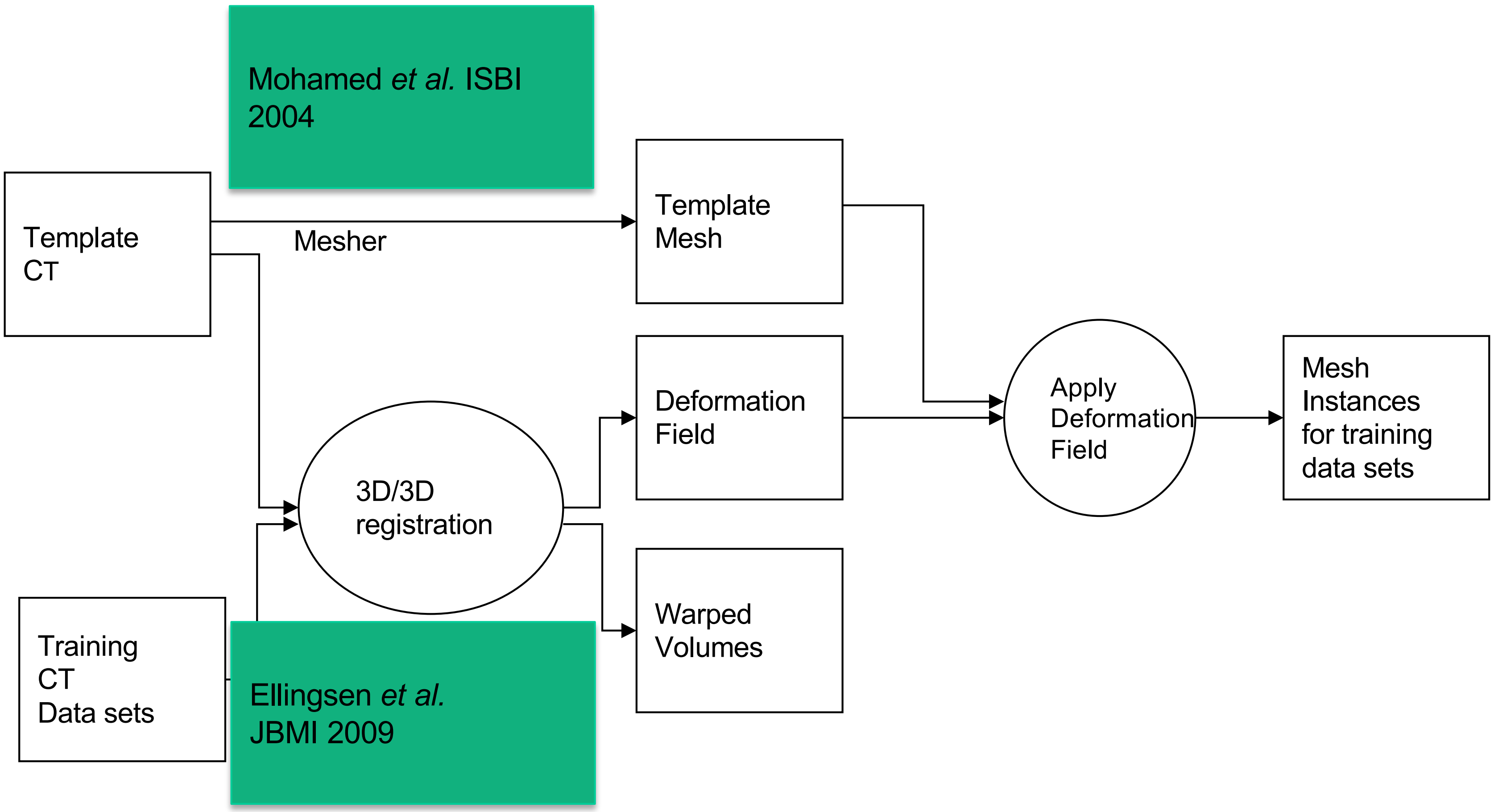
- Need to establish point correspondence between the training datasets



Slide Credit: G. Chintalapani 2010

Model Shape Correspondences

- **Automatic deformable registration based shape correspondences**



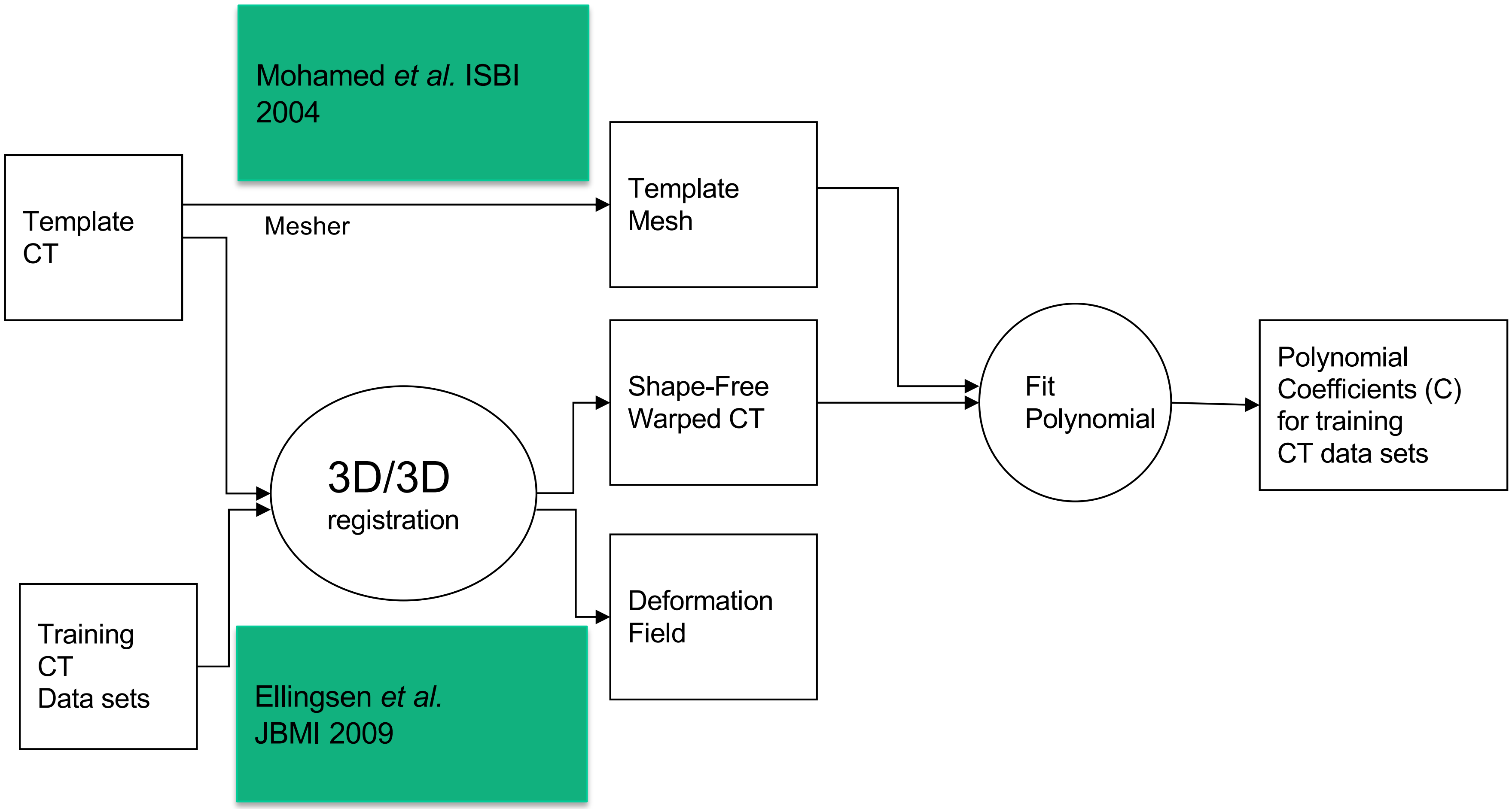
Flowchart for establishing shape correspondences for the training sample

Slide Credit: G. Chintalapani 2010

[1] Rueckert et al., MICCAI 03

Model Intensity Correspondences

- **Automatic deformable registration based correspondences**

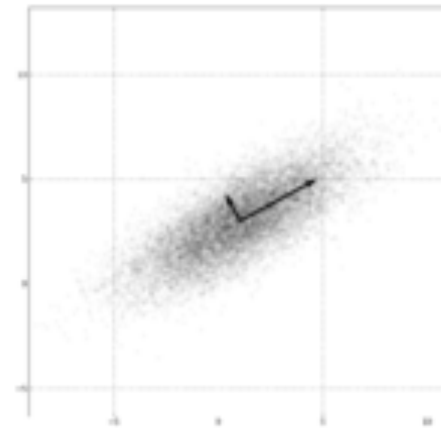


Flowchart for establishing intensity correspondences for the training sample

Principal Component Analysis

- Given the mesh instances of training sample,

$$S = \begin{bmatrix} \hat{s}_1 & \hat{s}_2 & . & . & \hat{s}_N \end{bmatrix}_{3nXN} = \begin{bmatrix} x_{11} & x_{12} & . & . & x_{1N} \\ y_{11} & y_{12} & . & . & y_{1N} \\ z_{11} & z_{12} & . & . & z_{1N} \\ . & . & . & . & . \\ y_{n1} & y_{n2} & . & . & z_{nN} \\ z_{n1} & z_{n2} & . & . & z_{nN} \end{bmatrix}$$



- Compute mean and subtract the mean from the sample

- Compute $$\tilde{S} = S - \bar{s} = S - \frac{1}{N}\sum_{i=1}^{N} \hat{s}_i$$

$$SVD(\tilde{S}) = UDV^T$$

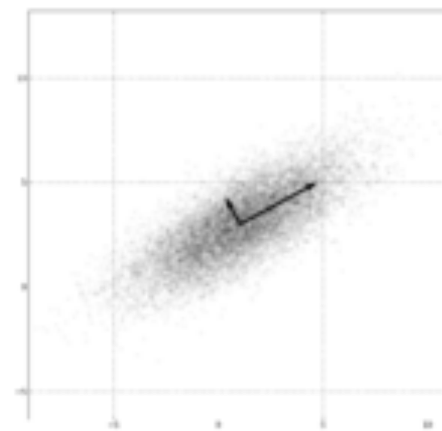
With principal components in U and eigen values $$\lambda = \frac{1}{N-1} DD^T$$

Slide Credit: G. Chintalapani 2010

Principal Component Analysis

- Given the PCA model, any data instance can be expressed as a linear combination of the principal components

$$\bar{s} + \sum_{k=1}^{N-1} U_k \lambda_k$$



- Compact model → fewer components

- Select first 'd' components represented by the 'd' eigen values

Slide Credit: G. Chintalapani 2010

Statistical Shape and Intensity Models

- Shape statistical model: Mesh vertices become data matrix

$$\bar{s} + \sum_{k=1}^{d} U_k \lambda_k = \bar{s} + U^T \lambda$$

- Intensity statistical model: Polynomial coefficients become data matrix

$$\bar{c} + \sum_{k=1}^{p} Y_k \mu_k = \bar{c} + \mathrm{Y}^T \mu$$

Slide Credit: G. Chintalapani 2010

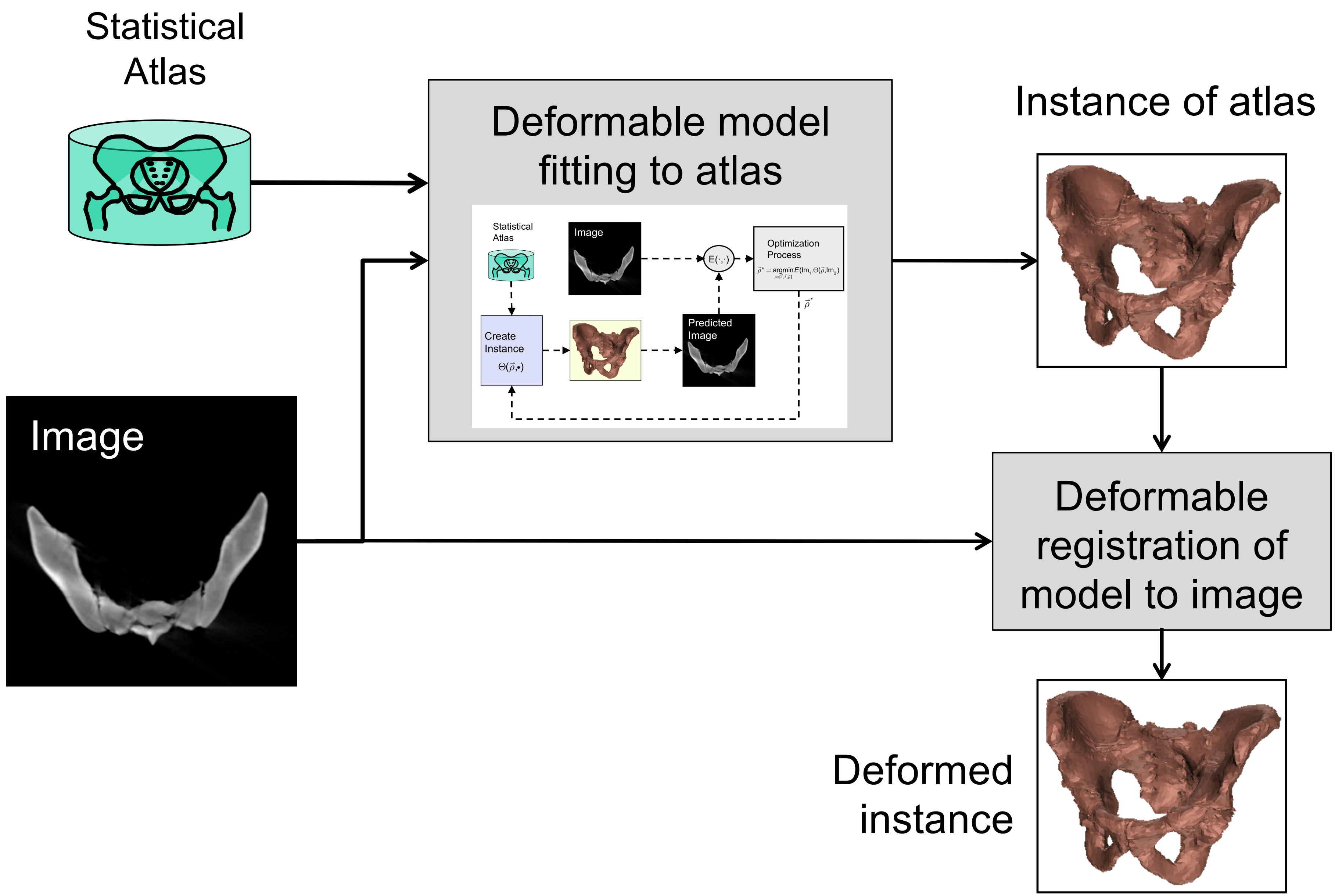
Deformable Registration Between Shape/Density Atlas and Patient CT

- Goal: Register and Deform the statistical density atlas to match patient anatomy
- Significance:
  - Building patient specific model with same topology (mesh structure) as the atlas
  - Automatic segmentation
  - Accumulatively building models for training set
  - Pathological diagnosis

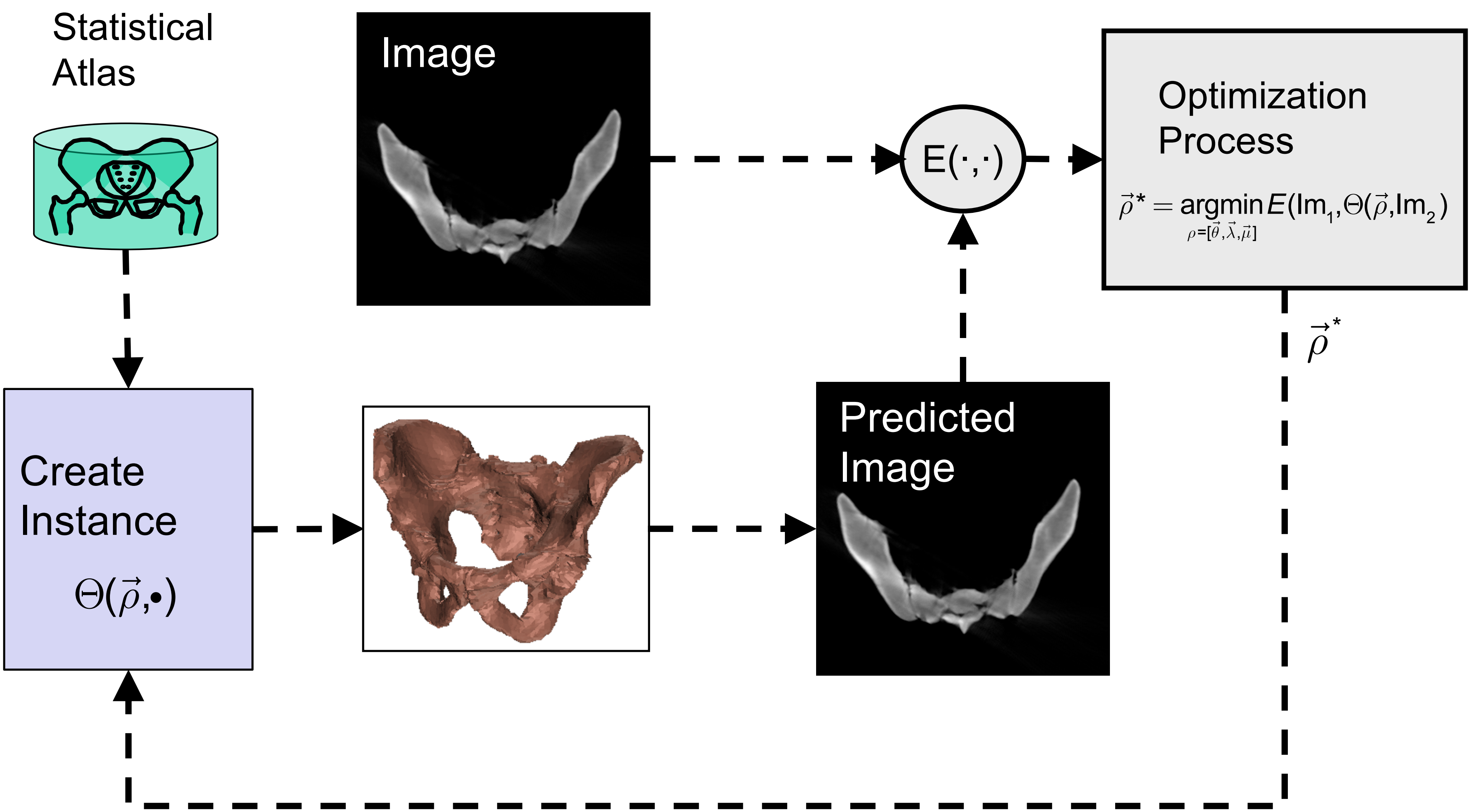
Jianhua Yao

Slide credit: Gouthami Chintalapani

Typical pipeline for atlas-assisted registration/registration



Deformable model fitting



Deformable Registration Scheme

- Affine Transformation
  - Translation T=(t_x, t_y, t_z)
  - Rotation R=(r_x, r_y, r_z)
  - Scale S=(s_x, s_y, s_z) [Similarity if $s_x = s_y = s_z$]
- Global Deformation
  - Statistical deformation mode (M_i)
- Local Deformation
  - Adjustment of every vertex

Optimization Algorithm

- Direction Set (Powell's) method in multi-dimensions
  - Search the parameter space to minimize the cost functions
  - Advantage
    - Don't need to compute derivative of cost functions
    - Much fewer evaluations than downhill simplex methods
- Alternatives
  - Downhill Simplex (similar advantages)
  - Covariance Matrix Adaptation Evolution Strategy (CMA-ES) method (similar advantages)
  - Levenberg-Marquardt (requires computing gradients)
  - Many others

Local Deformation

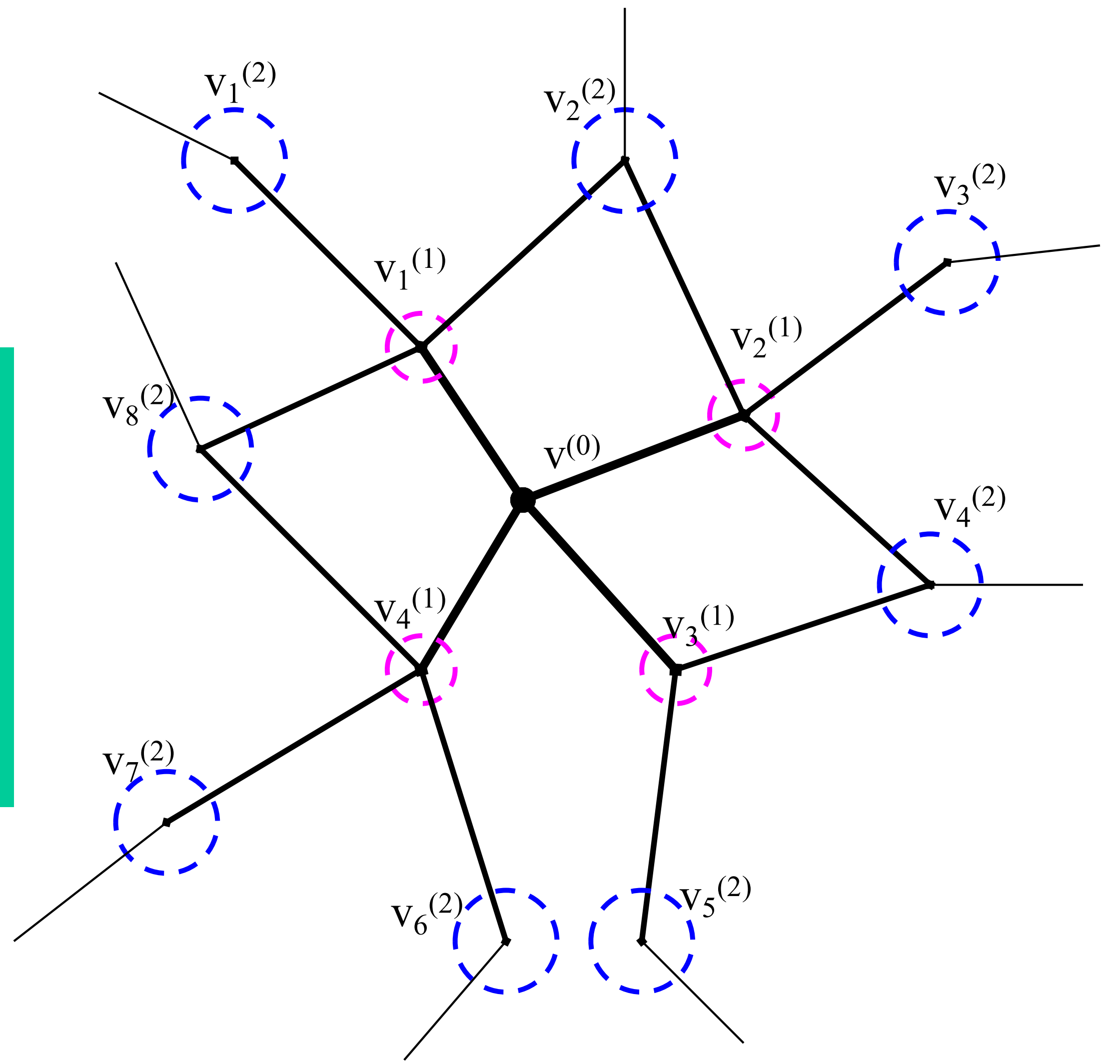
- Motivation: Statistical deformation can’t capture all the variability due to the limited number of models in the training set
- Locally adjust the location of vertices to match the boundary of the bone and the interior density properties
- Use multiple-layer flexible mesh template matching to find the correspondence between model vertices and image voxels
- Apply radial basis function (or other scheme) based on vertex-to-voxel location matches

Jianhua Yao

Multiple-layer Flexible Mesh Template

- Each vertex on the model defines a mesh template
- Template is in the form

$$\begin{aligned}(0, &Sphere(v_1^{(1)} - v^{(0)}, r_1),\\ &Sphere(v_2^{(1)} - v^{(0)}, r_1), \cdots,\\ &Sphere(v_1^{(2)} - v^{(0)}, r_2),\\ &Sphere(v_1^{(2)} - v^{(0)}, r_2), \cdots)\end{aligned}$$



Template matching

For each pixel location $\vec{\mathbf{x}}_0$:

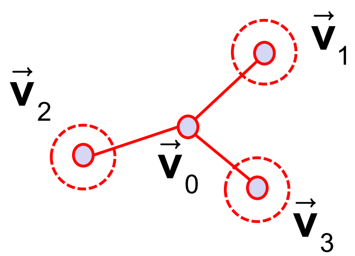
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

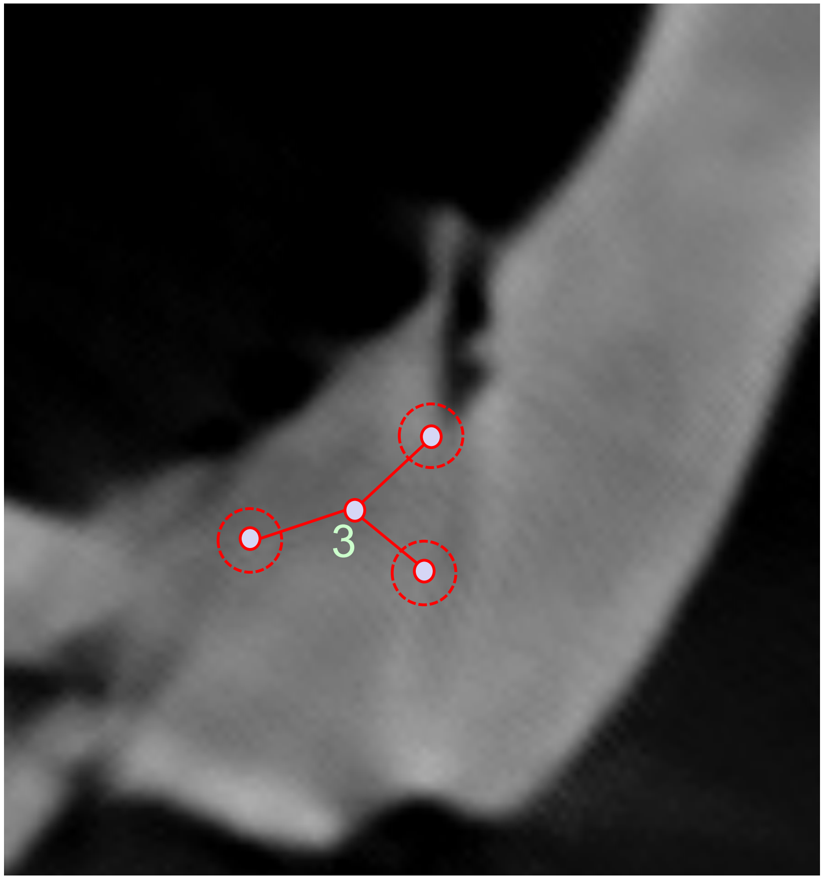
For each neighbor $\vec{\mathbf{v}}_k$

Find the $\vec{\mathbf{x}}_k$ near $\vec{\mathbf{v}}_k$ that minimizes $E(\vec{\mathbf{x}}_k, \vec{\mathbf{v}}_k)$

Score $(\vec{\mathbf{x}}_0) = E(\vec{\mathbf{x}}_0, \vec{\mathbf{v}}_0) + \sum_k w_k E(\vec{\mathbf{x}}_k, \vec{\mathbf{v}}_k)$

Pick the $\vec{\mathbf{x}}_0$ with the best score





Template matching

For each pixel location $\vec{\mathbf{x}}_0$:

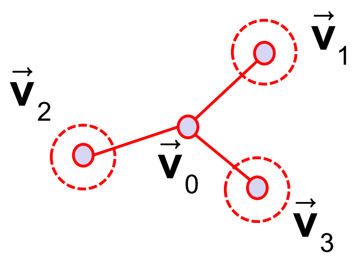
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

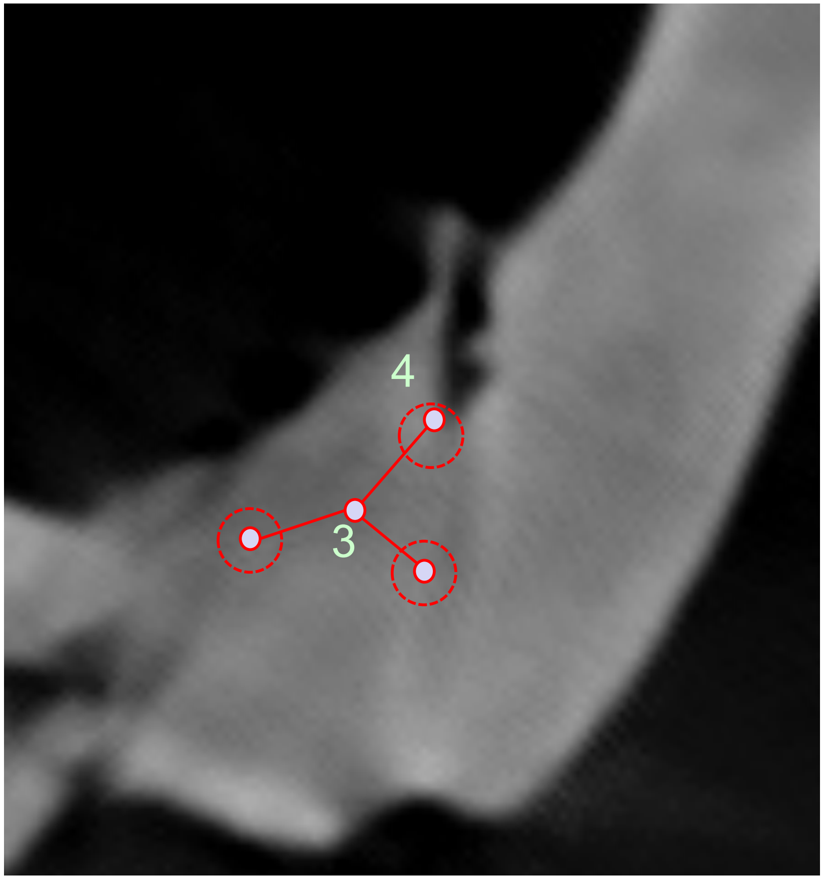
For each neighbor $\vec{\mathbf{v}}_k$

Find the $\vec{\mathbf{x}}_k$ near $\vec{\mathbf{v}}_k$ that minimizes $E(\vec{\mathbf{x}}_k, \vec{\mathbf{v}}_k)$

Score $(\vec{\mathbf{x}}_0) = E(\vec{\mathbf{x}}_0, \vec{\mathbf{v}}_0) + \sum_k w_k E(\vec{\mathbf{x}}_k, \vec{\mathbf{v}}_k)$

Pick the $\vec{\mathbf{x}}_0$ with the best score





Template matching

For each pixel location $\vec{\mathbf{x}}_0$:

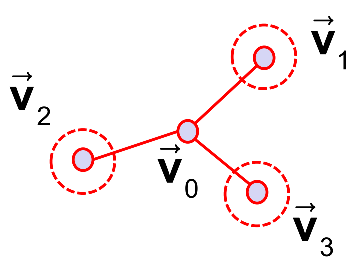
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

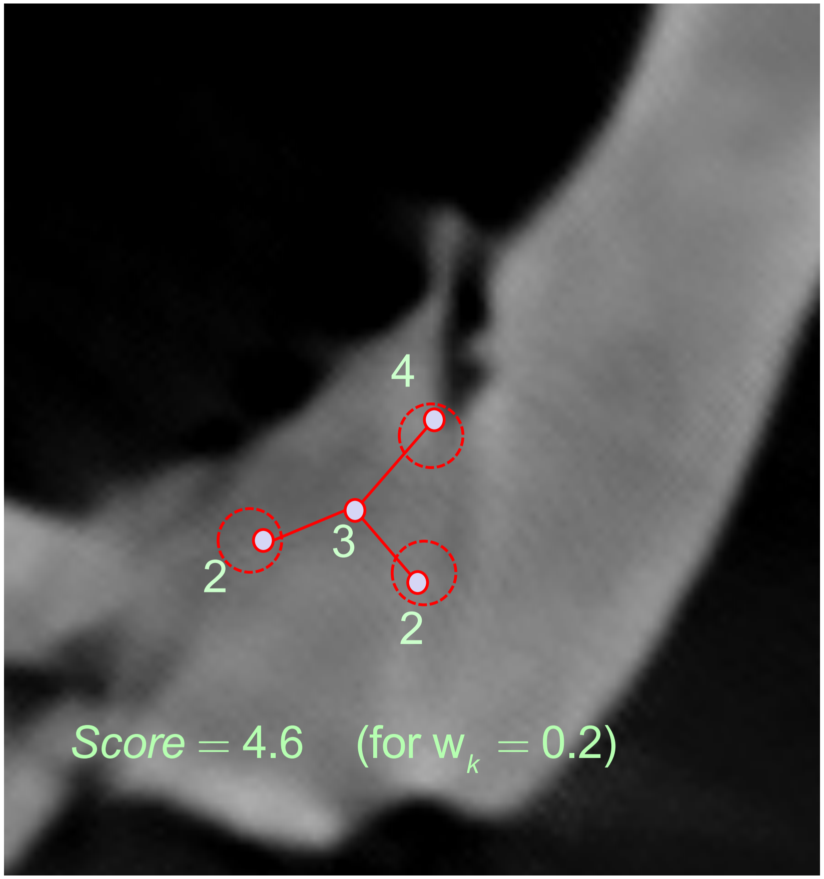
For each neighbor $\vec{\mathbf{v}}_k$

Find the $\vec{\mathbf{x}}_k$ near $\vec{\mathbf{v}}_k$ that minimizes $E(\vec{\mathbf{x}}_k, \vec{\mathbf{v}}_k)$

Score $(\vec{\mathbf{x}}_0) = E(\vec{\mathbf{x}}_0, \vec{\mathbf{v}}_0) + \sum_k w_k E(\vec{\mathbf{x}}_k, \vec{\mathbf{v}}_k)$

Pick the $\vec{\mathbf{x}}_0$ with the best score



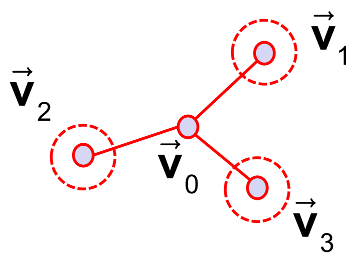


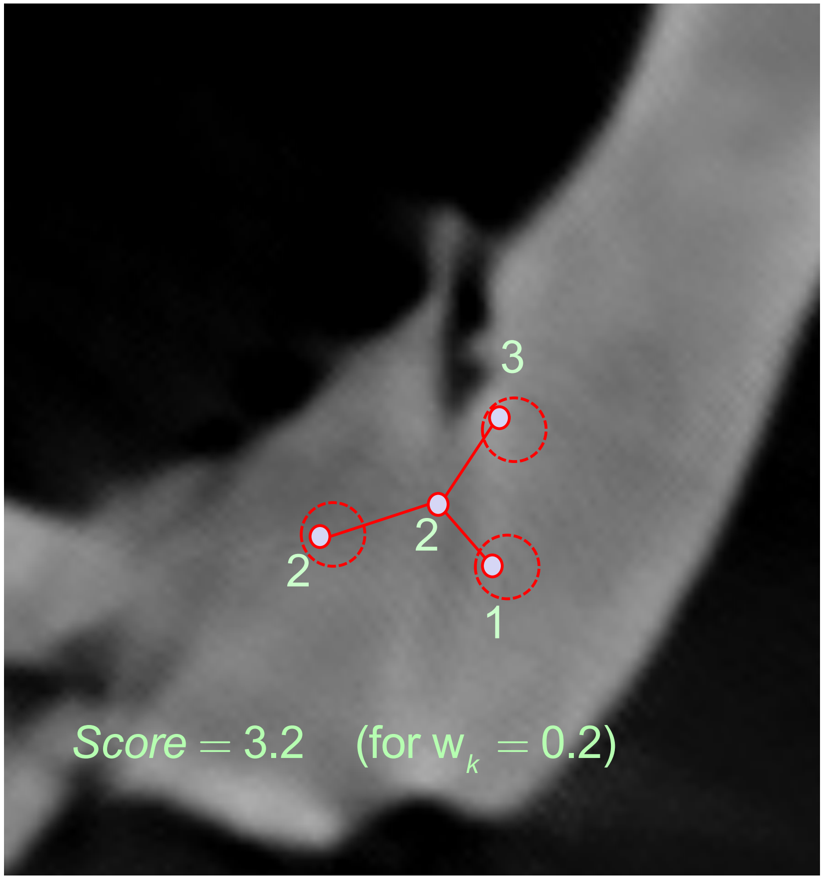
Template matching

For each pixel location $\vec{\mathbf{x}}_0$:

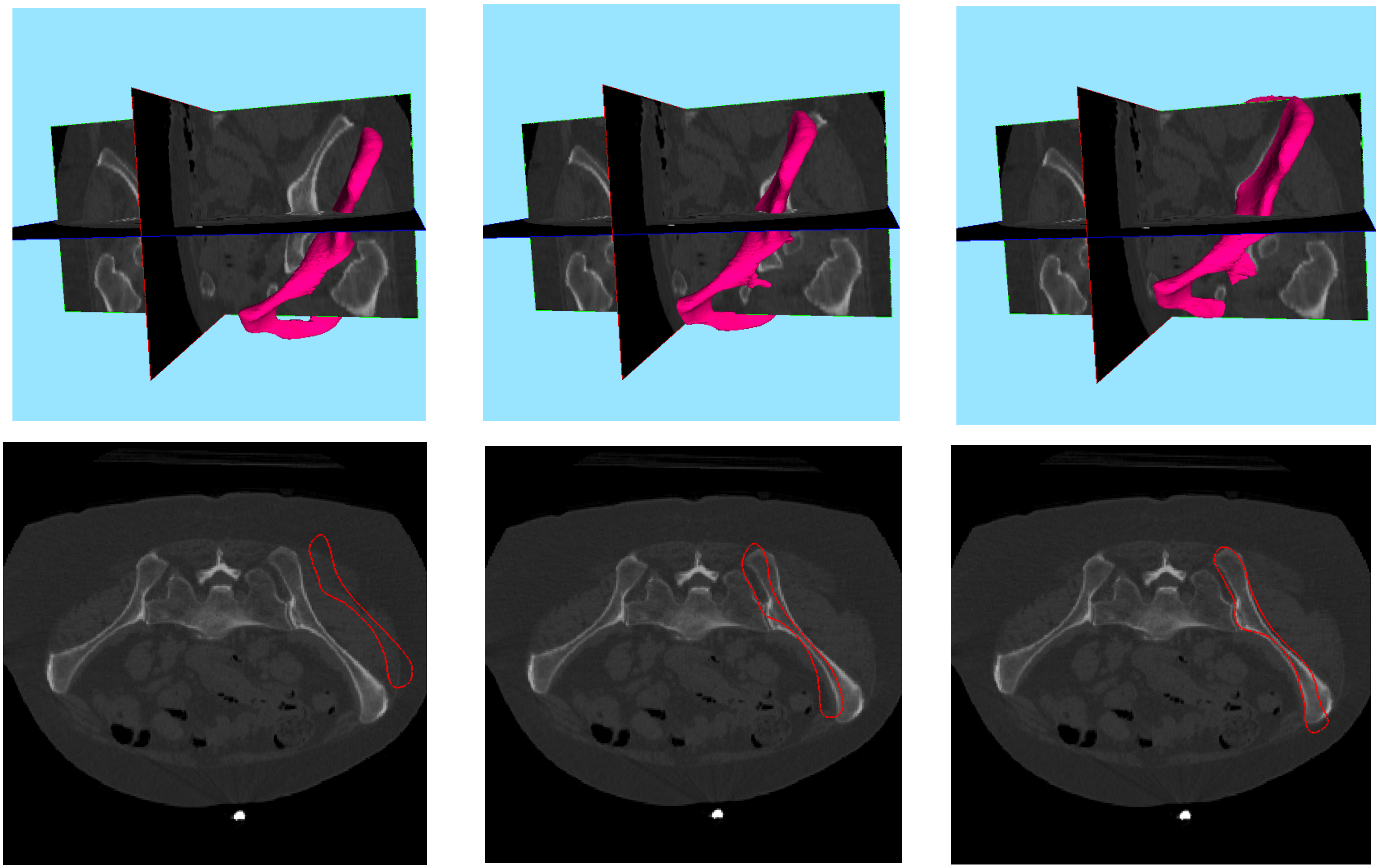
- Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$
- For each neighbor $\vec{\mathbf{v}}_k$
  - Find the $\vec{\mathbf{x}}_k$ near $\vec{\mathbf{v}}_k$ that minimizes $E(\vec{\mathbf{x}}_k,\vec{\mathbf{v}}_k)$
- Score $(\vec{\mathbf{x}}_0) = E(\vec{\mathbf{x}}_0,\vec{\mathbf{v}}_0) + \sum_k w_k E(\vec{\mathbf{x}}_k,\vec{\mathbf{v}}_k)$

Pick the $\vec{\mathbf{x}}_0$ with the best score





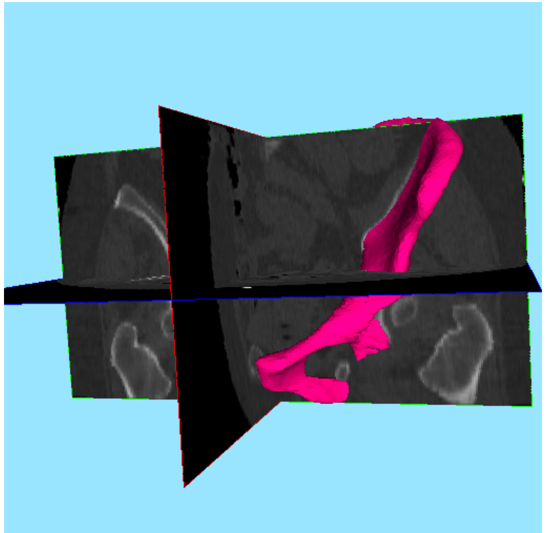
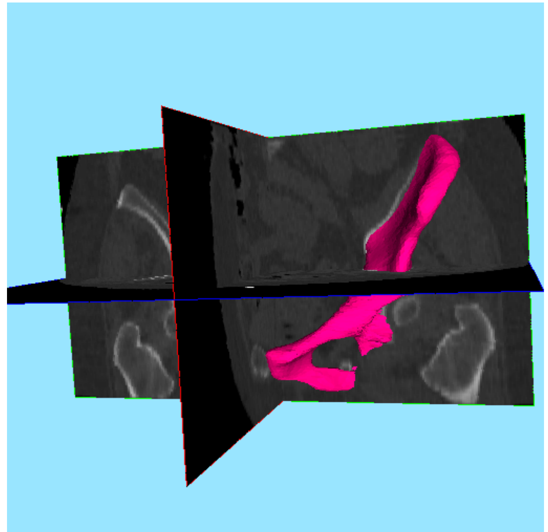
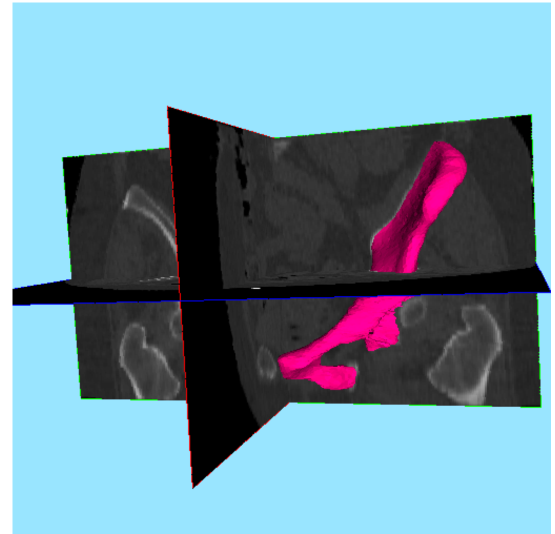
Results (Affine Transformation)

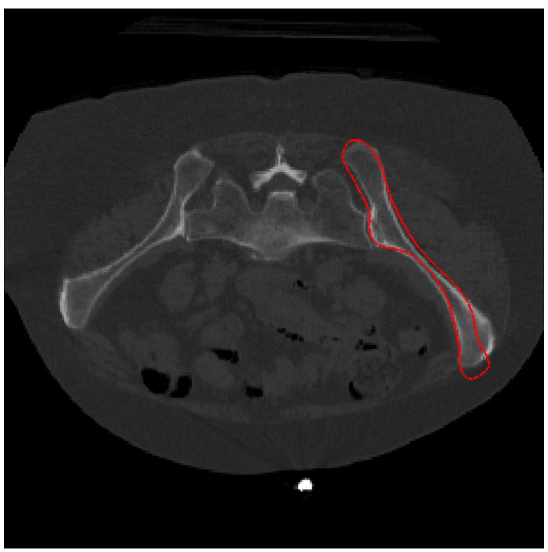


Initial Intermediate Final

Jianhua Yao

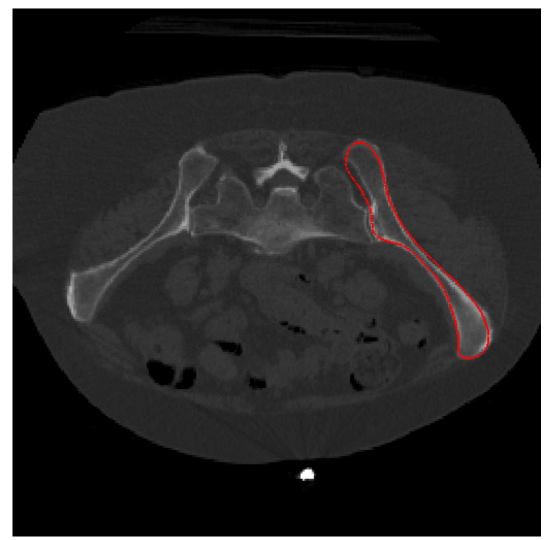
Results (Global Deformation)



Initial

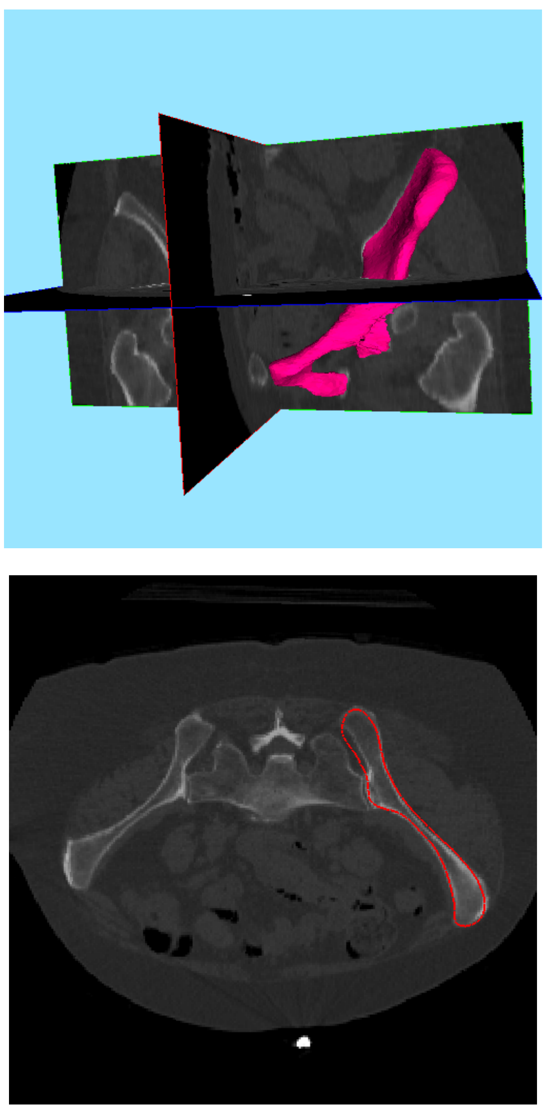
Intermediate



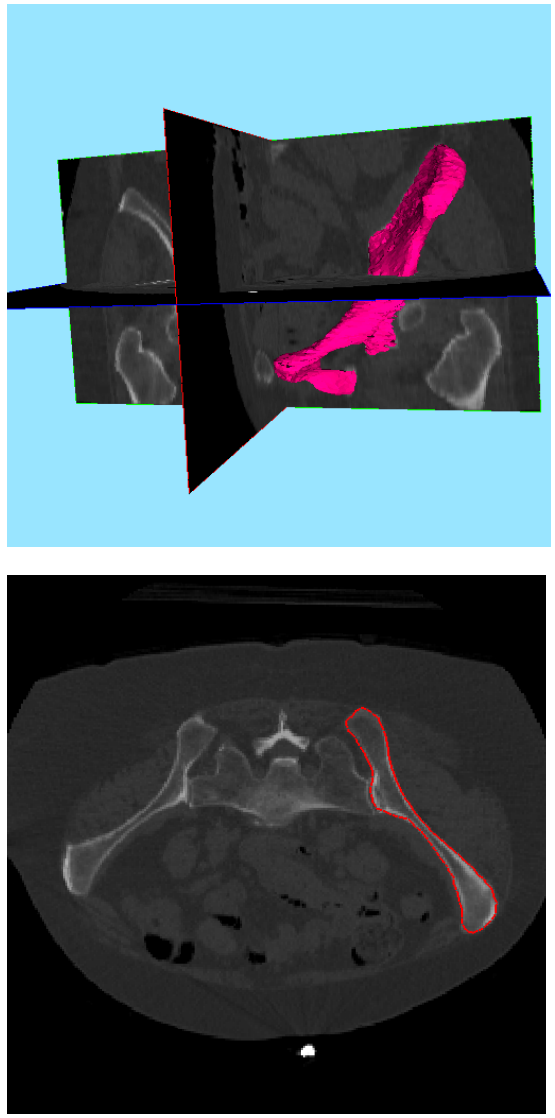
Final

Jianhua Yao

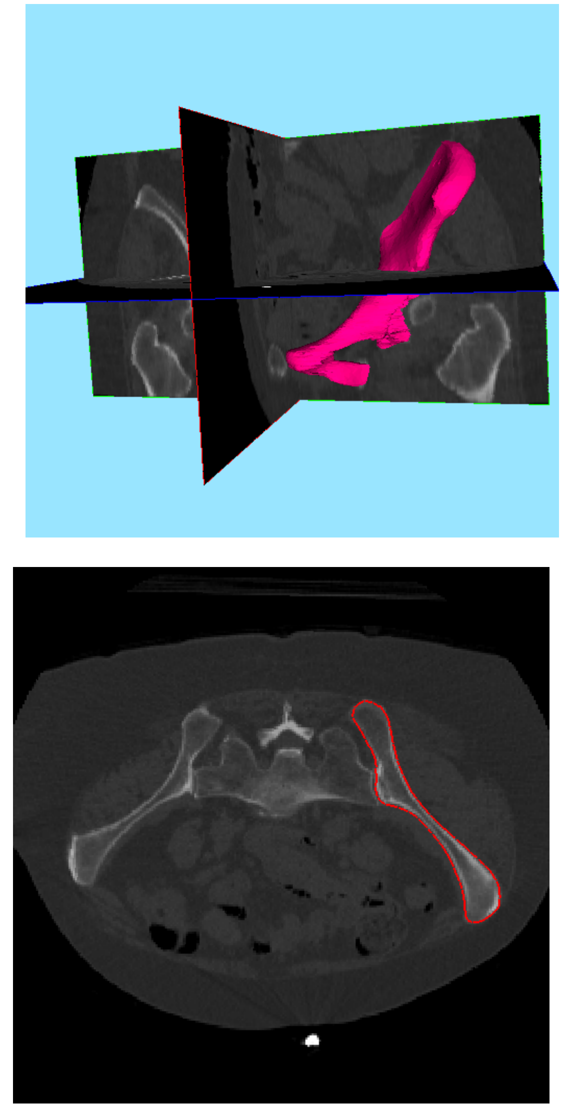
Results (Local Deformation)



Initial



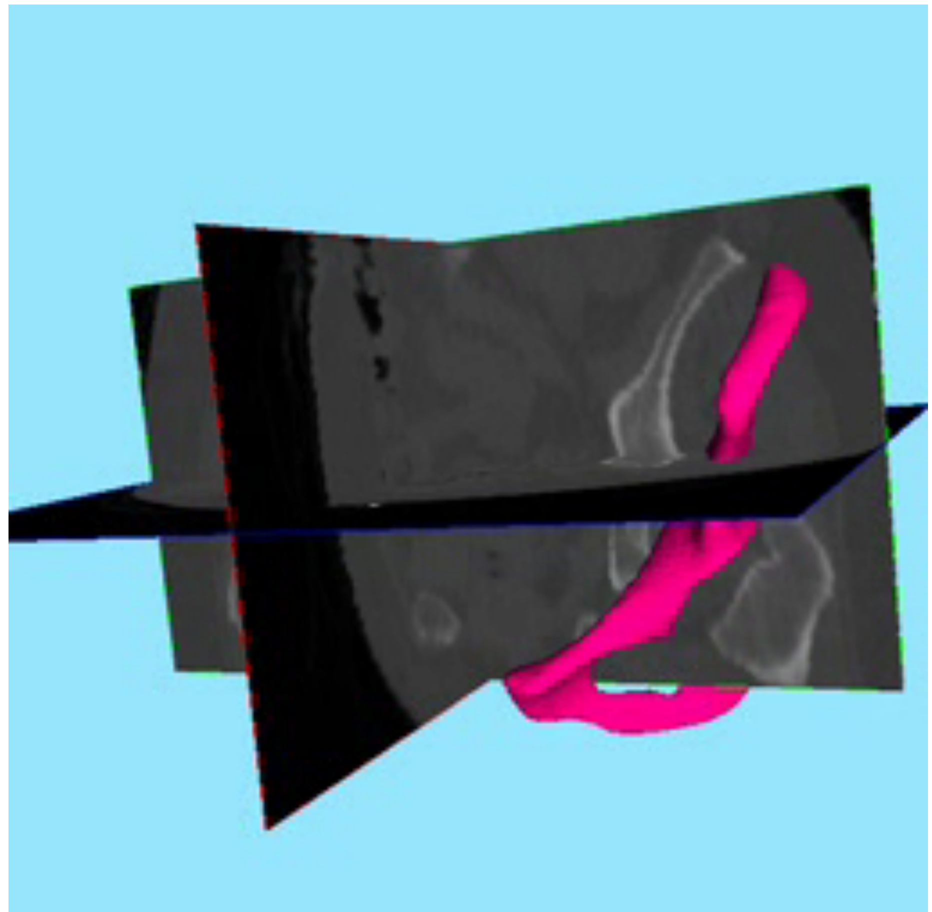
Intermediate



Final

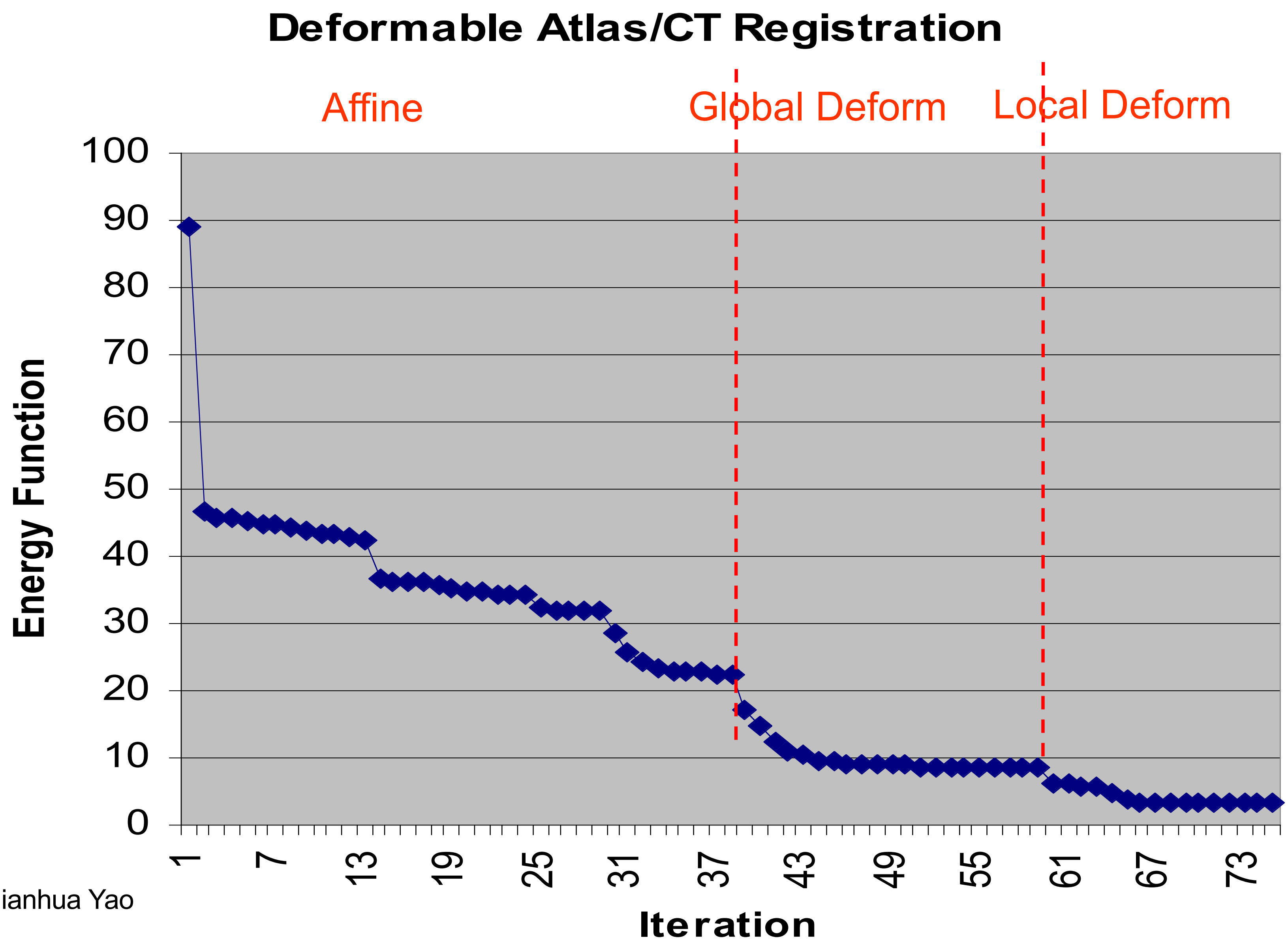
Jianhua Yao

Deformable Atlas-to-CT Registration (3D-3D)



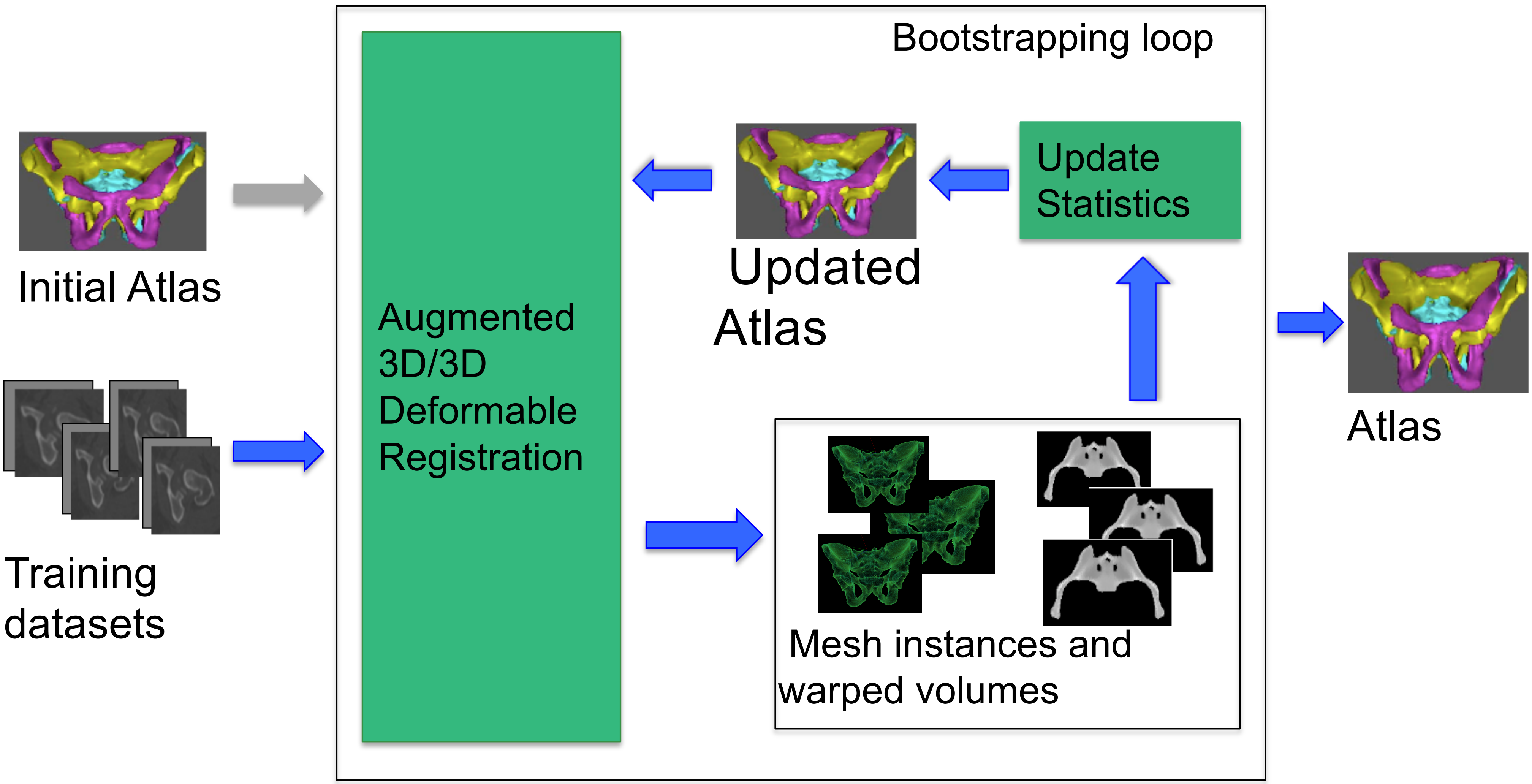
Jianhua Yao

Results (Deformable Registration)



Jianhua Yao

Iterative “bootstrapping” of Atlas



Chintalapani *et al.* MICCAI 2007

Slide credit: Gouthami Chintalapani

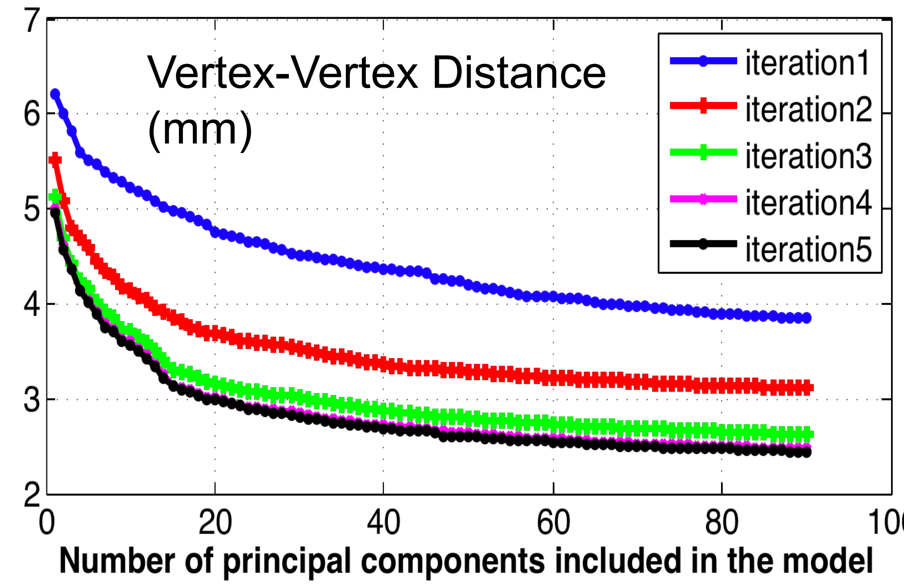
Leave-Out Validation Experiments

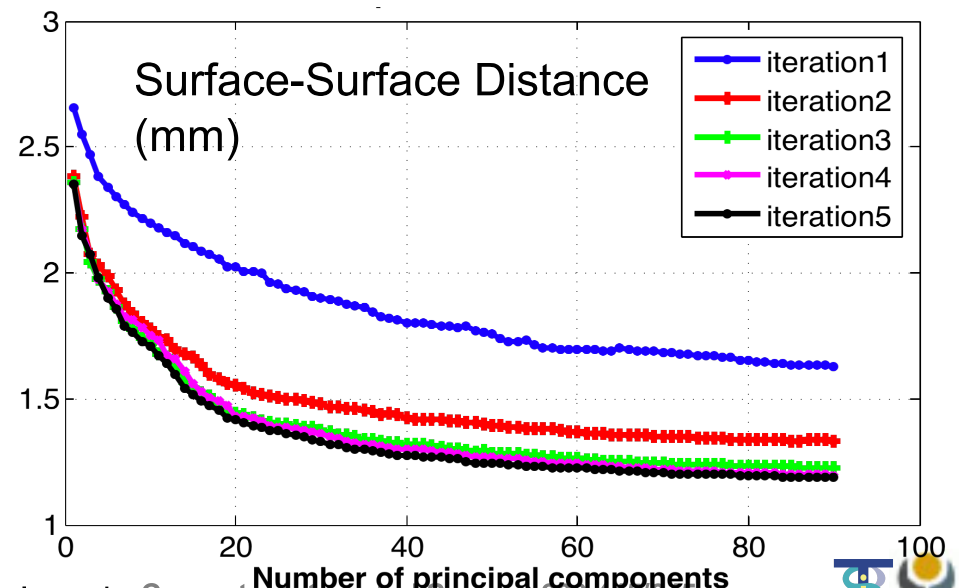
- # of iterations: 5
- # of data sets: 110
- # of data sets in atlas: 90
- # of data sets left out: 20
- Given a left-out dataset, s_j

compute the estimated shape from atlas using

$$\lambda = U' * (s_j - \bar{S})$$

$$s_j^{est} = \bar{S} + U\lambda$$



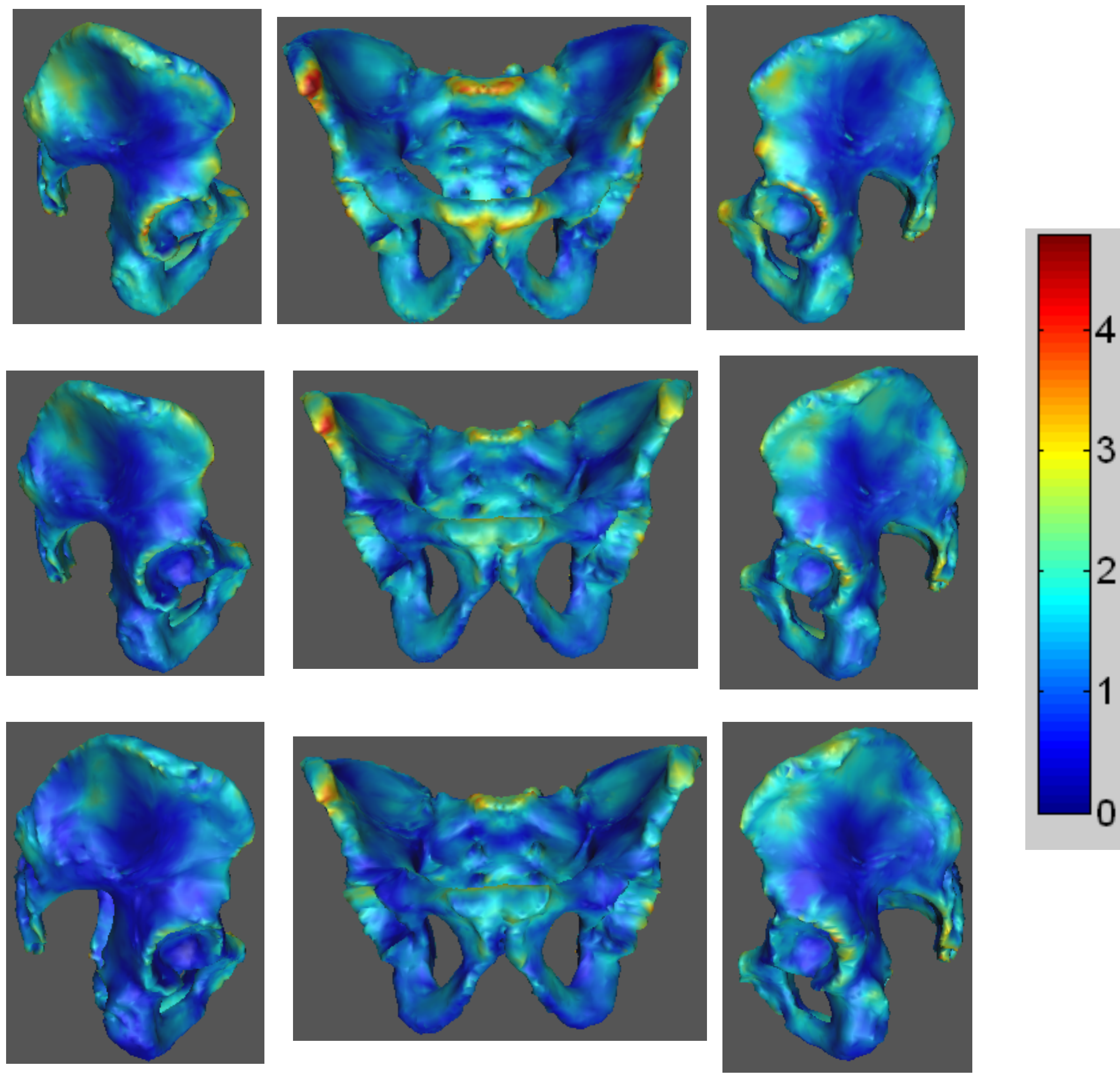


Distribution of Surface Registration Errors

Iteration 1

Iteration 3

Iteration 5

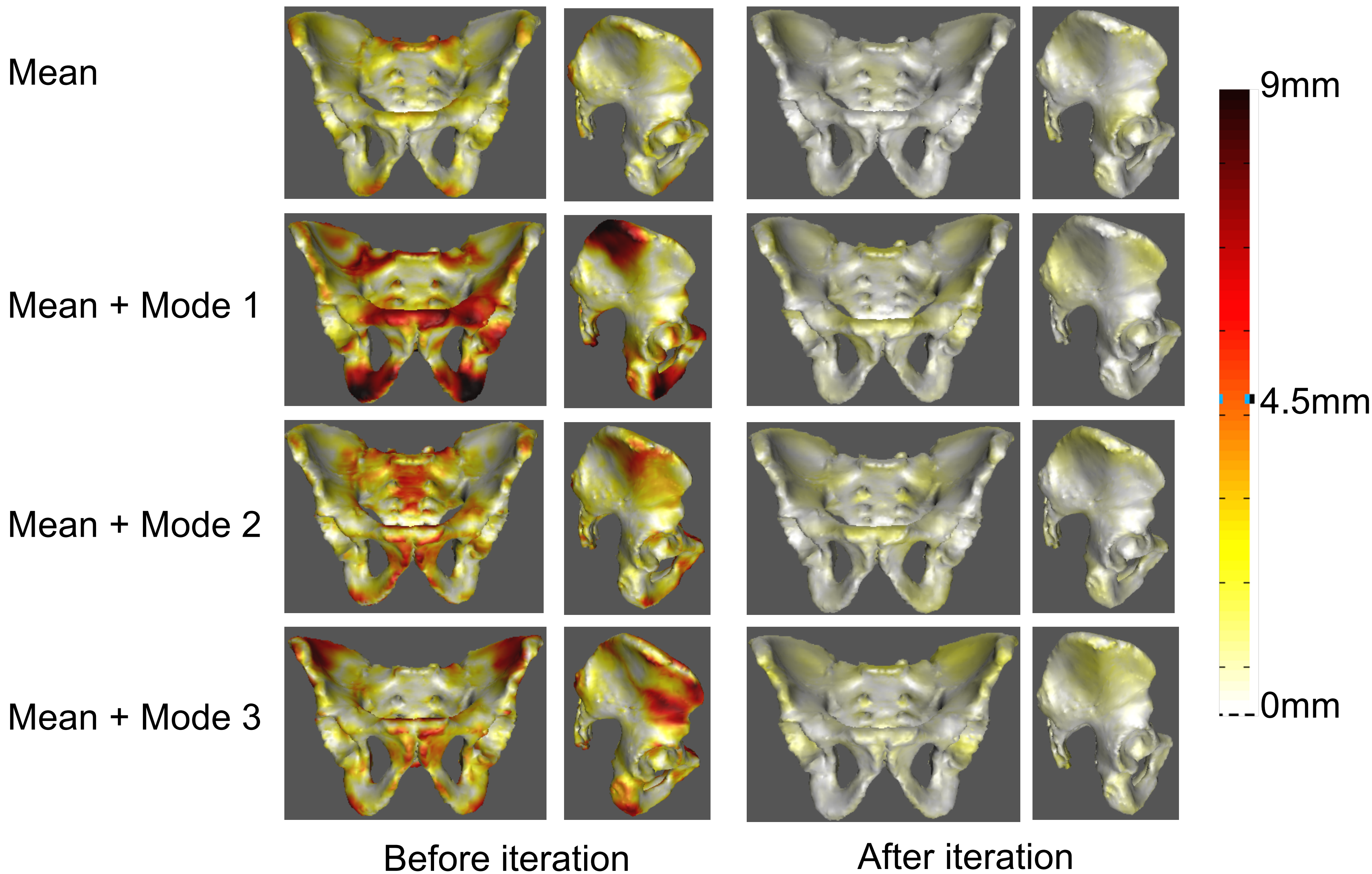


Slide credit: Gouthami Chintalapani

Choice of Initial Template

- Claim:
  - iterative method does not depend on the choice of template
- Criteria:
  - Mean shape converges
  - Modes exhibit similar deformation patterns
- Experimental setup:
  - Three random templates
  - Atlases with and without bootstrapping compared
- Result
  - All three atlases exhibit similar deformation patterns after bootstrapping

Average Difference between Atlases 1,2 and 3

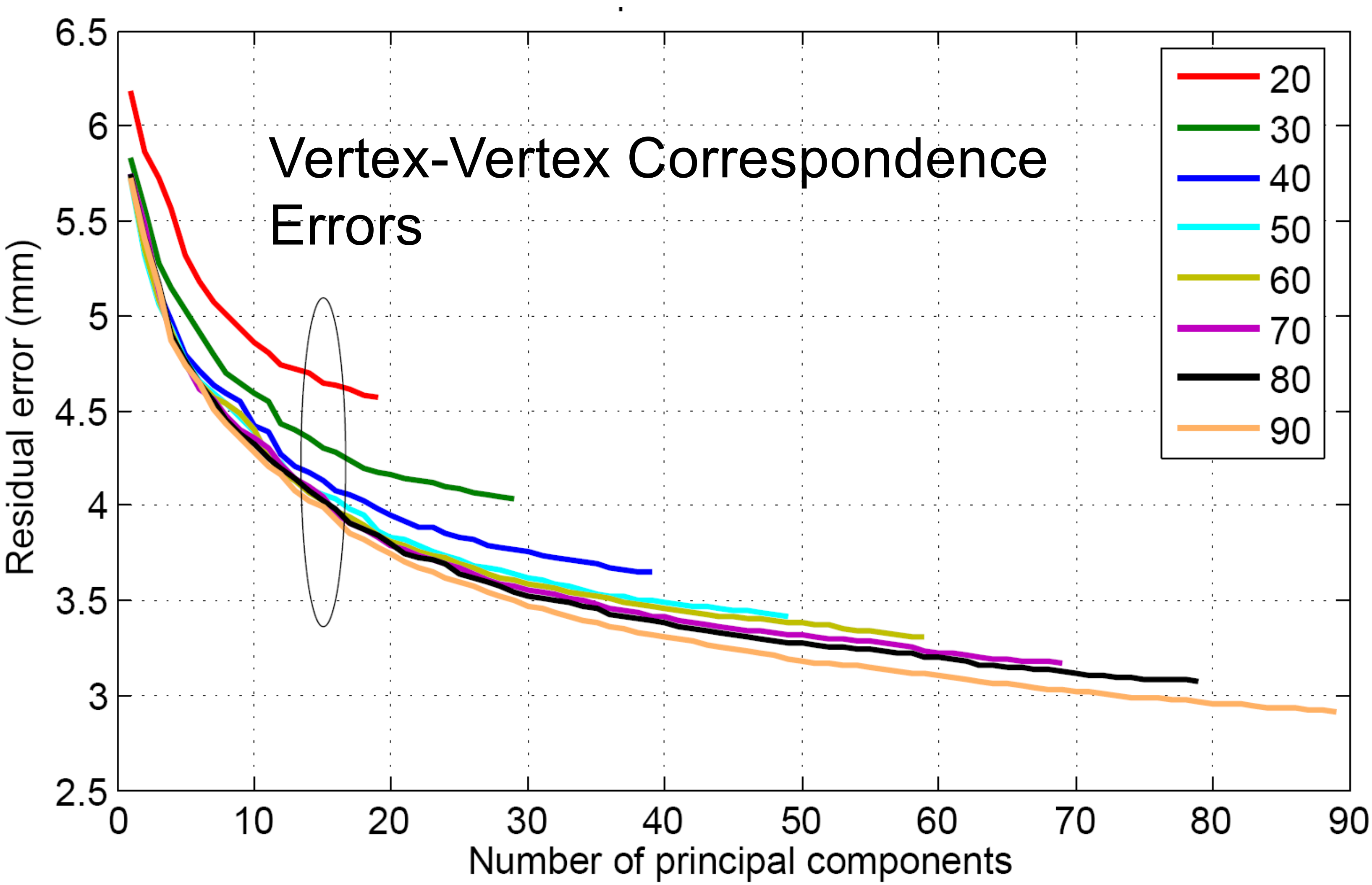


Slide credit: Gouthami Chintalapani

Training Sample Size

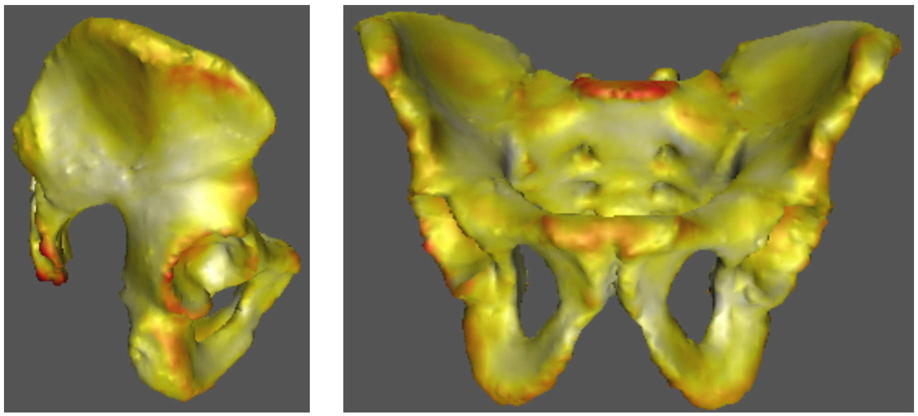
- Goal:
  - To determine the size of the training sample to build a stable statistical atlas
- Criteria:
  - Atlas is stable
  - No significant improvement in residual error
- Experimental setup:
  - Varying sample size 20, 40, 60, 80
  - Leave-20-out validation test
- Result:
  - Minimum of 50 data sets are required for pelvis atlas

Training Sample Size

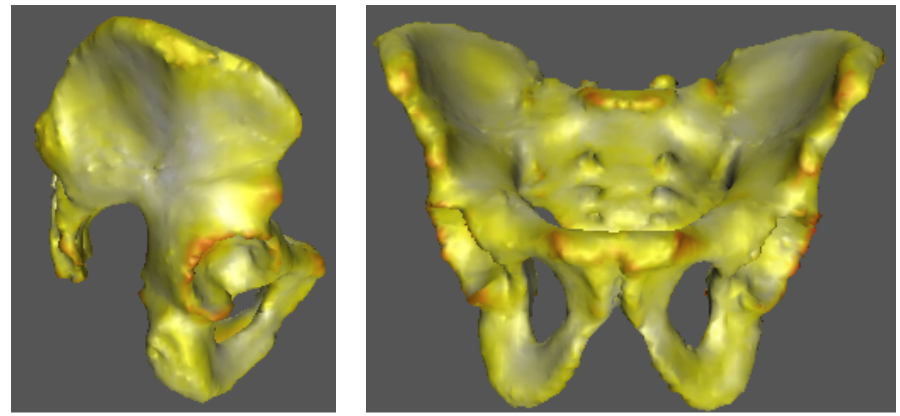


Slide credit: Gouthami Chintalapani

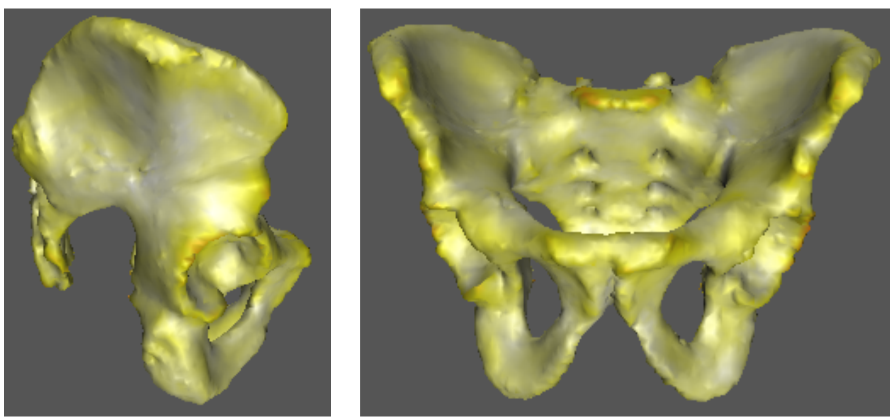
Surface residual error using 18 modes for different sample set sizes



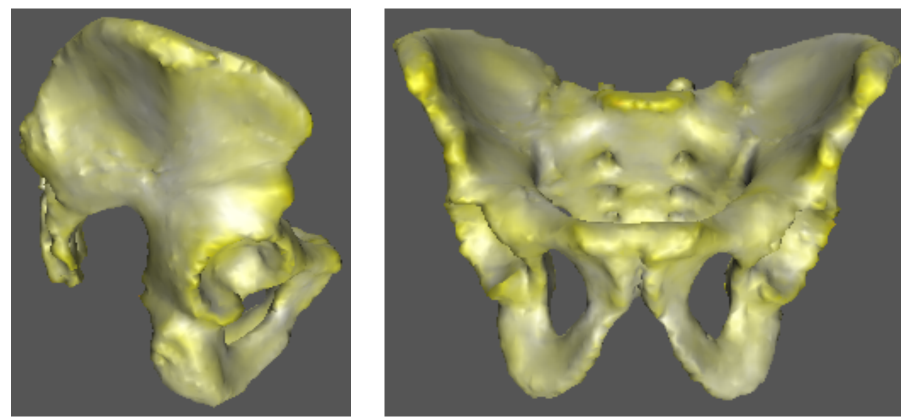
20 dataset atlas



40 dataset atlas



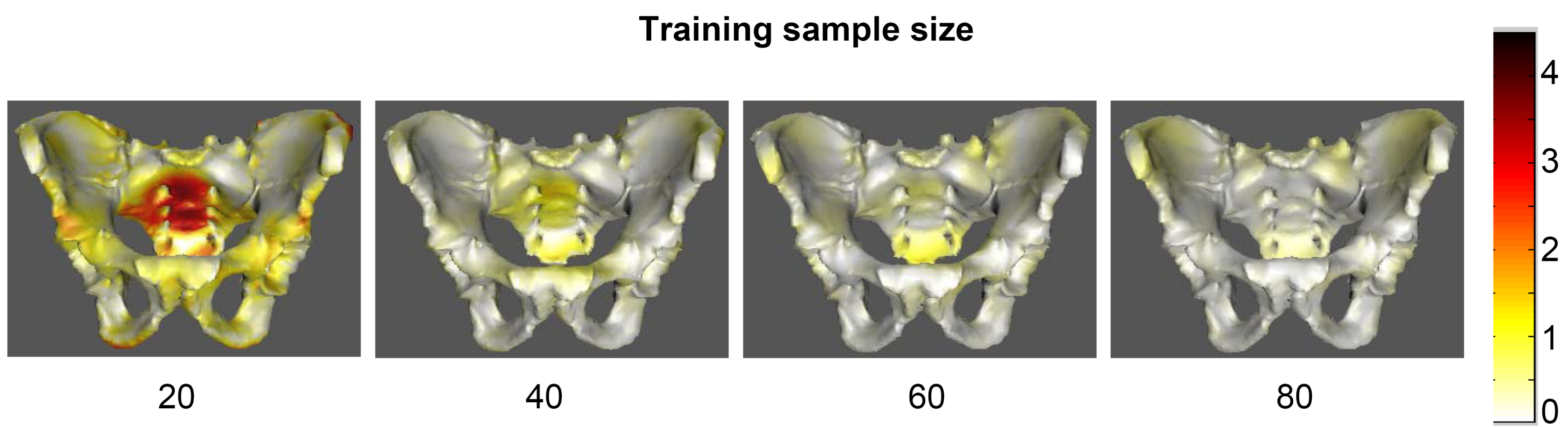
60 dataset atlas

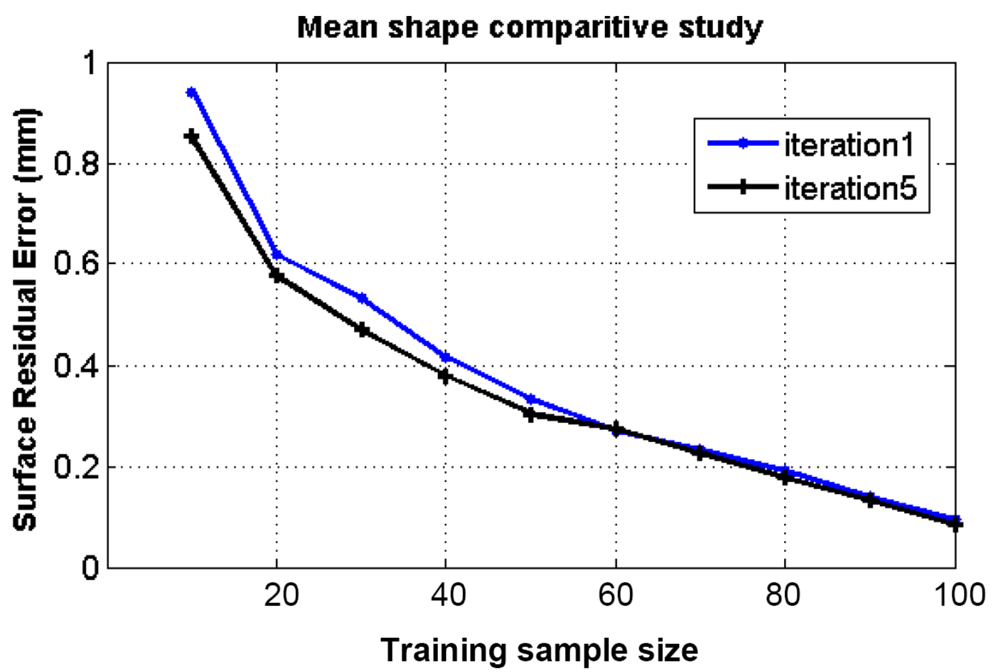


80 dataset atlas

Slide credit: Gouthami Chintalapani

Stability Analysis – Mean Shape





Slide credit: Gouthami Chintalapani

Shape Atlas Mesh Refinement

- Note that the methods described so far all assume that the vertices of the mesh after deformable registration all correspond to each other
- This is often not the case
- Also, some image segmentation methods we would like to use do not always produce the same surface mesh
- Is there anything we can do???
  - Yes: The basic idea is to do deformable registration of statistical model vertices to the surface(s) to find corresponding points, and then iterate.

Mesh Vertex Improvement (click here)

Active Appearances

- The material following is based on
  - T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active Appearance Models", Proc. Fifth European Conf. Computer Vision, H. Burkhardt and B. Neumann, eds., vol. 2, pp. 484-498, 1998.
  - T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 681-- 685, June 2001.
- Authors' focus was development of method for matching statistical models of appearance to [2D] images
- Applied to faces, 2D medical images
- Basic idea has since been extended to many applications in 2D & 3D medical imaging

Statistical Appearance Models

- Shape
  - In this case, 2D locations of key feature points
- "Texture"
  - I.e., patterns of intensities or colors across image patches
- Method to build: Identify key points; do deformable warp of points to common coordinate system; normalize intensities; read intensities into an intensity vector **G**

Labelled image Points Shape-free patch

$$\left\| \mathbf{G} \right\| = 1$$

$$\sum \mathbf{G}_k = 0$$

T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 681-- 685, June 2001

Appearance models, con'd

Appearance model is defined by an instance parameter vector $\vec{\lambda}$, mean shape and texture $\mathbf{X}^{(avg)}$ and $\mathbf{G}^{(avg)}$, and variation mode matrices $\mathbf{M}_{\mathbf{X}}$ and $\mathbf{M}_{\mathbf{G}}$. Thus, an instance (j) would be

$$\mathbf{G}^{(j)} = \mathbf{G}^{(avg)} + \mathbf{M}_{\mathbf{G}} \bullet \vec{\lambda}^{(j)} = \mathbf{G}^{(avg)} + \sum_{k=1}^{N_{\mathbf{G}}} \vec{\mathbf{M}}_{\mathbf{G}}^{(k)} \bullet \vec{\lambda}_k^{(j)}$$

$$\mathbf{X}^{(j)} = \mathbf{X}^{(avg)} + \mathbf{M}_{\mathbf{X}} \bullet \vec{\lambda}^{(j)} = \mathbf{X}^{(avg)} + \sum_{k=1}^{N_{\mathbf{X}}} \vec{\mathbf{M}}_{\mathbf{X}}^{(k)} \bullet \vec{\lambda}_k^{(j)}$$

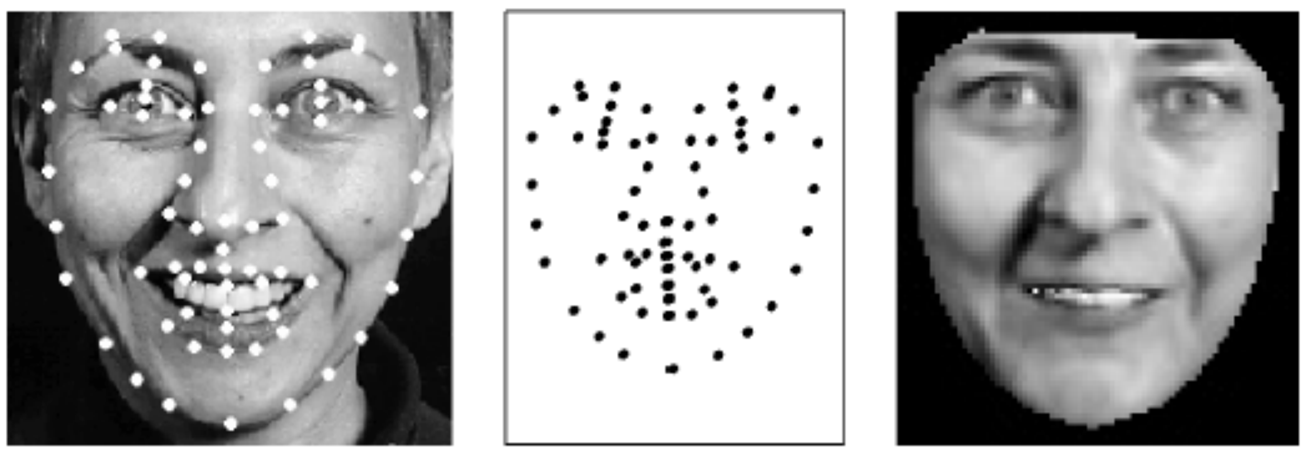
In fact, they created a multi-resolution hierarchy with models similar to the above at different resolutions.

Used linear principal components analysis (PCA) to determine the statistical parameters.

T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 681-- 685, June 2001

Training Set for 2001 Cootes & Taylor paper

- 400 faces
- 68 points
- 10000 intensity values



Labelled image Points Shape-free patch

T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 681-- 685, June 2001

Complication

- How do you do PCA if shape and intensity may co-vary?

Answer : Form combined vector of shape and intensity variation

$$\mathbf{Y} = \begin{bmatrix} \mathbf{W_X}\left(\mathbf{X} - \mathbf{X}^{(avg)}\right) \\ \mathbf{G} - \mathbf{G}^{(avg)} \end{bmatrix}$$

where $\mathbf{W_X}$ is a diagonal matrix of weights. Then do PCA on **Y**.

T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 681-- 685, June 2001

Further complication

- How do you find the right weights to use?

Answer (from Cootes *et al. 1998)*:

The elements of $\mathbf{b}_s$ have units of distance, those of $\mathbf{b}_g$ have units of intensity, so they cannot be compared directly. Because $\mathbf{P}_g$ has orthogonal columns, varying $\mathbf{b}_g$ by one unit moves $\mathbf{g}$ by one unit. To make $\mathbf{b}_s$ and $\mathbf{b}_g$ commensurate, we must estimate the effect of varying $\mathbf{b}_s$ on the sample $\mathbf{g}$. To do this we systematically displace each element of $\mathbf{b}_s$ from its optimum value on each training example, and sample the image given the displaced shape. The RMS change in $\mathbf{g}$ per unit change in shape parameter b_s gives the weight w_s to be applied to that parameter in equation (5).

I.e., do PCA first on shape only and determine an appropriate $\mathbf{V}_\mathbf{X}$. Then find an optimal $\vec{\lambda}^{(j)}$ for each training sample (j). Then vary the values of $\vec{\lambda}^{(j,k)} = \vec{\lambda}^{(j)} + \alpha\vec{\mathbf{e}}_k$ to create new shape models $\mathbf{X}^{(j,k)}$ and determine the corresponding texture vectors $\mathbf{G}^{(j,k)}$. Then the weight

$$w_k = \sqrt{\frac{1}{N}\sum_j \left\| \mathbf{G}^{(j,k)} - \mathbf{G}^{(j)} \right\|^2} \,/\, \alpha.$$

Face modes

Fig. 2. First two modes of shape variation (±3 sd)

Shape

Fig. 3. First two modes of grey-level variation (±3 sd)

Intensity

Source: Cootes *et al. 1998*

Face modes

Fig. 4. First four modes of appearance variation (±3 sd)

Combined

Source: Cootes *et al. 1998*

Basic Algorithm

- Make an initial guess at model weights
- Create a model from weights
- Evaluate error
- Iteratively improve

T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 681-- 685, June 2001

Basic Iteration of the Method

1. Project the texture sample into the texture model frame using $\mathbf{g}_s = T_{\mathbf{u}}^{-1}(\mathbf{g}_{im})$.
2. Evaluate the error vector, $\mathbf{r} = \mathbf{g}_s - \mathbf{g}_m$, and the current error, $E = |\mathbf{r}|^2$.
3. Compute the predicted displacements, $\delta\mathbf{p} = -\mathbf{R}\mathbf{r}(\mathbf{p})$.
4. Update the model parameters $\mathbf{p} \to \mathbf{p} + k\delta\mathbf{p}$, where initially $k = 1$.
5. Calculate the new points, $\mathbf{X}'$ and model frame texture $\mathbf{g}'_m$.
6. Sample the image at the new points to obtain $\mathbf{g}'_{im}$.
7. Calculate a new error vector, $\mathbf{r}' = T_{\mathbf{u}'}^{-1}(\mathbf{g}'_{im}) - \mathbf{g}'_m$.
8. If $|\mathbf{r}'|^2 < E$, then accept the new estimate; otherwise, try at $k = 0.5$, $k = 0.25$, etc.

$$\mathbf{R} = \left(\frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T \frac{\partial \mathbf{r}}{\partial \mathbf{p}}\right)^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T .$$

Source: Cootes *et al. 2001*

Basic Iteration of the Method

1. Project the texture sample into the texture model frame using $\mathbf{g}_s = T_{\mathbf{u}}^{-1}(\mathbf{g}_{im})$.
2. Evaluate the error vector, $\mathbf{r} = \mathbf{g}_s - \mathbf{g}_m$, and the current error, $E = |\mathbf{r}|^2$.
3. Compute the predicted displacements, $\delta\mathbf{p} = -\mathbf{R}\mathbf{r}(\mathbf{p})$.
4. Update the model parameters $\mathbf{p} \rightarrow \mathbf{p} + k\delta\mathbf{p}$, where initially $k = 1$.
5. Calculate the new points, $\mathbf{X}'$ and model frame texture $\mathbf{g}'_m$.
6. Sample the image at the new points to obtain $\mathbf{g}'_{im}$.
7. Calculate a new error vector, $\mathbf{r}' = T_{\mathbf{u}'}^{-1}(\mathbf{g}'_{im}) - \mathbf{g}'_m$.
8. If $|\mathbf{r}'|^2 < E$, then accept the new estimate; otherwise, try at $k = 0.5$, $k = 0.25$, etc.

Note: simple sum of differences. What are some alternatives?

Source: Cootes *et al. 2001*

Results

Fig. 10. Reconstruction (left) and original (right) given original landmark points

Source: Cootes *et al. 1998*

Results

Source: Cootes *et al. 1998* **Fig. 11.** Multi-Resolution search from displaced position

Results: Knee Example

- Trained on 30 knee MRI images
- With 42 landmark points

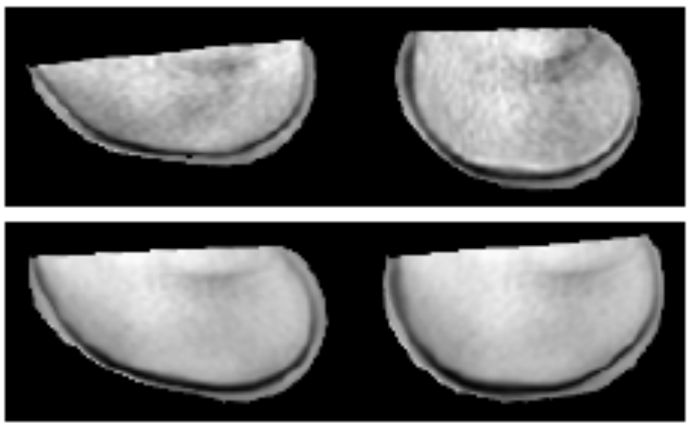


Fig. 12. First two modes of appearance variation of knee model

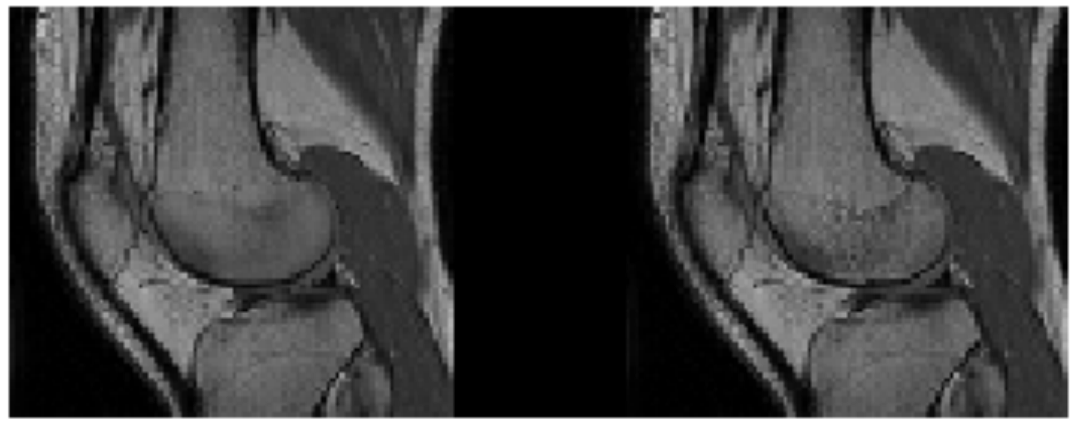


Fig. 13. Best fit of knee model to new image given landmarks

Source: Cootes *et al. 1998*

Results: Knee Example

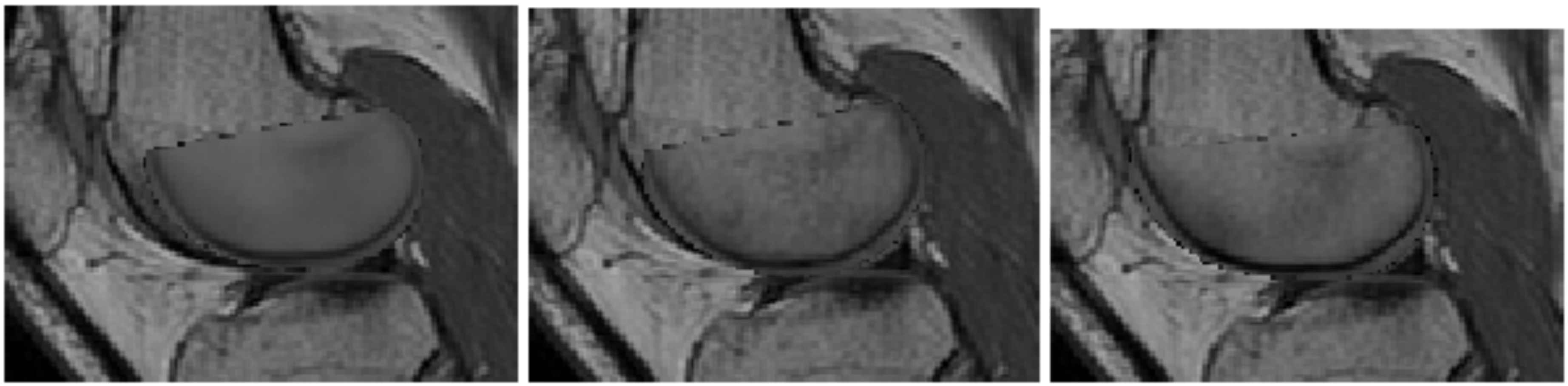


Fig. 14. Multi-Resolution search for knee

Source: Cootes *et al. 1998*

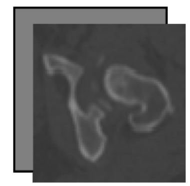
Deformable registration between density atlas and a set of 2D X-Rays

- Goal: Register and Deform the statistical density atlas to match intraoperative x-rays
- Significance:
  - Build virtual patient specific CT without real patient CT
  - Register pre-operative models and intra-operative images
  - Map predefined surgical procedure and anatomical landmarks into intra-operative images

2D/3D Registration – Shape and Intensity Models

Shape statistical model (PCA on mesh vertices)

$S^{(0)}, \{S^{(k)}\}$



Intensity statistical model (PCA on voxel values and polynomials fit to modes)

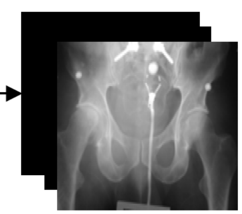
$C^{(0)}, \{C^{(k)}\}$

Patient X-ray images or DRRs

Deformable 2D/3D Registration (Estimate $\lambda^{(k)}$)

Registered Atlas Projections

Reestimate Densities (Compute $\mu^{(k)}$)



Registered Atlas Projections

$S^0 + \Sigma\lambda^k S^k$

$C^0 + \Sigma\, \mu^k C^k$

[1] Sadowsky, O., Chintalapani, G., Taylor, R.H., MICCAI 2007;
[2] Chintalapani *et al.* PMMIA/MICCAI 2009

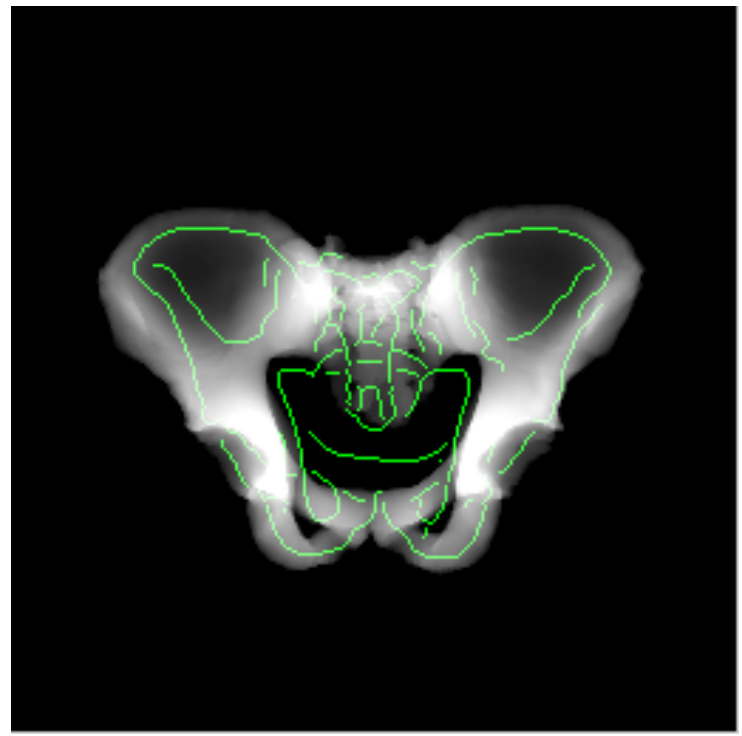
Slide credit: Gouthami Chintalapani

2D/3D Registration – Shape and Intensity

(1) #	(2) $S^{true} - S^{est}$ (mm)	(3) RMS (V^{true} , V^{est}_{mean}) (HU)	(4) RMS(V^{true} , V^{est}_{modes}) (HU)	(5) Δ ((3)-(4))/(3) %
1	1.94	109.92	58.88	**46.43**
2	1.62	128.32	96.0	**25.19**
3	1.90	98.4	77.12	**21.63**
4	2.60	51.68	41.6	**19.50**
5	2.48	109.44	84.8	**22.51**
6	1.95	73.44	50.56	**31.15**
7	2.30	72.96	47.52	**34.84**
8	2.93	101.28	85.76	**15.32**
avg	**2.21**	**93.18**	**67.78**	**27.07**

Table 1: Residual errors from leave-out-validation tests of the augmented registration algorithm. Column 2 shows the surface distance after 2D/3D shape registration. Columns 3 shows residual errors when using mean density only and column 4 shows residual errors with mean density and density modes. The % reduction in RMS error between columns 3 and 4 is given in Column 5

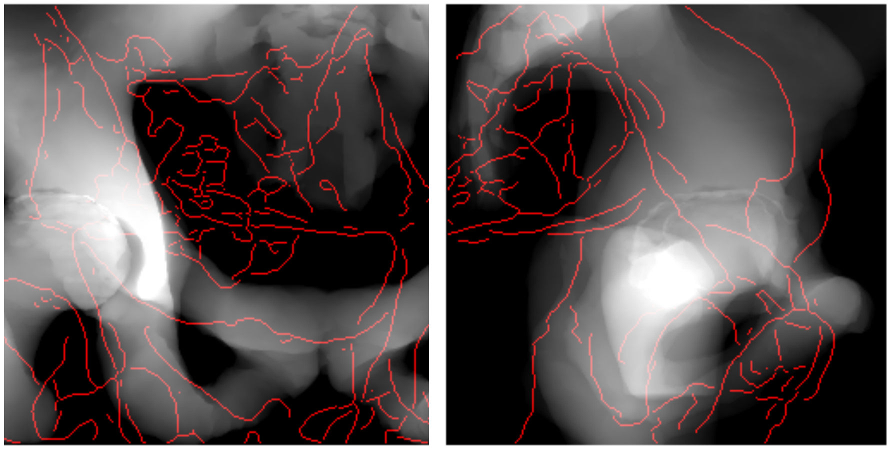
Avg surface registration accuracy: 2.21mm
Avg. reduction in RMS errors intensity: 27%



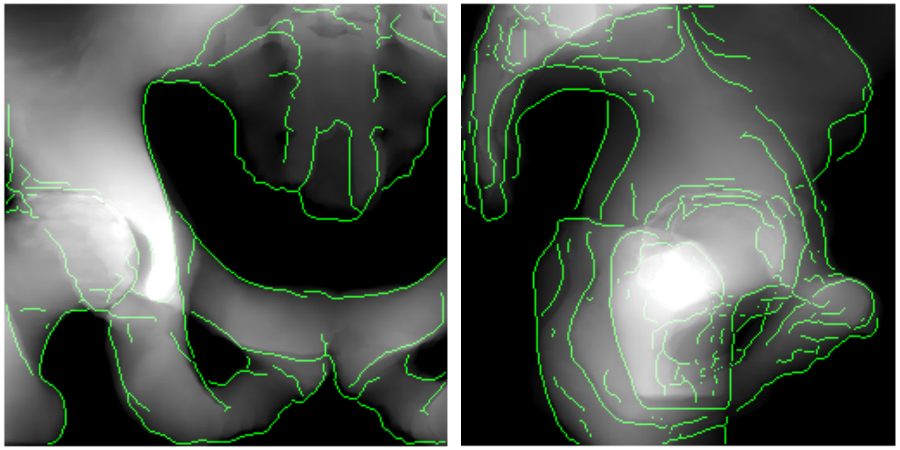
Slide credit: Gouthami Chintalapani

2D/3D Registration – Hip Model

- **Problem**: To create patient specific models using atlas
  - single organ atlases are insufficient
- **Our approach**: Develop a multi-component atlas
  - Use hip atlas instead of a pelvis or femur atlas
  - Extend atlas building framework to incorporate hip joint
  - Extend the registration framework to incorporate articulated hip joint
- **Results**
  - Multi-component atlas registration is accurate compared to individual organ atlas



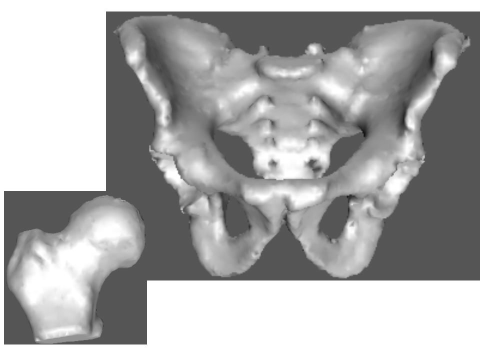
Pelvis atlas registered to hip projection images



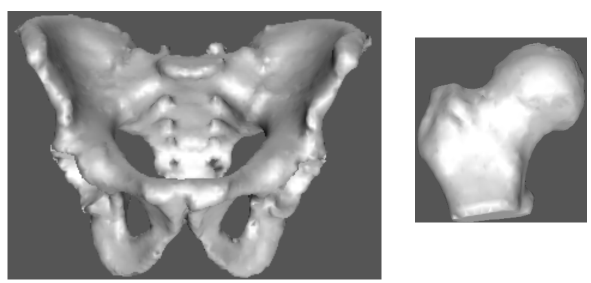
Hip atlas registered to hip projection images

Multi-Component Atlas

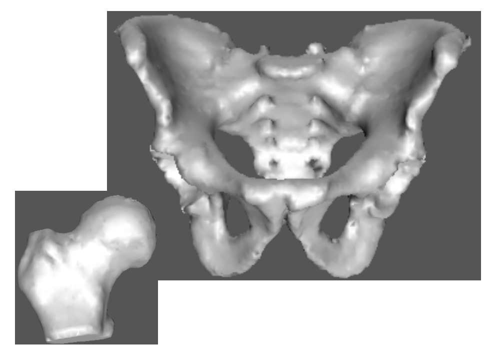
1. Two components – pelvis and femur
2. Create mesh instances of pelvis and femur separately
3. Align pelvis and femur meshes together
4. Align pelvis meshes
5. Align femur meshes
6. Concatenate pelvis and femur meshes
7. PCA on the concatenated mesh



Combined Rigid+Scale



Separate Rigid



Combined Statistical Analysis

Slide credit: Gouthami Chintalapani

Multi-Component Hip Atlas

PC1

PC2

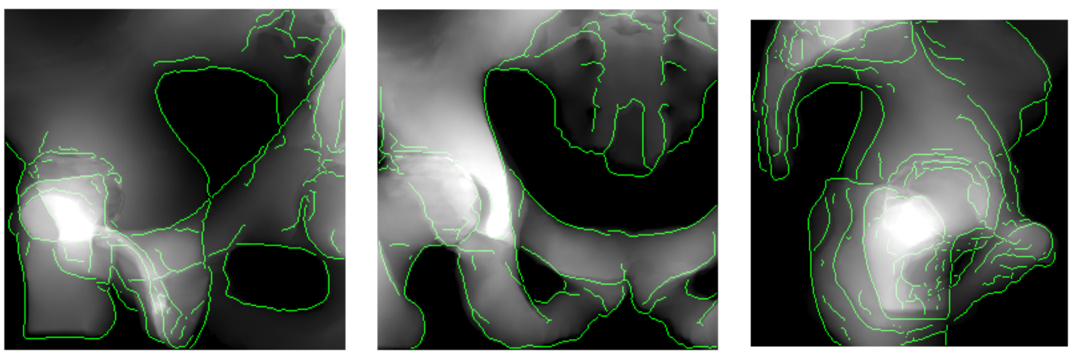
PC3

[1] Chintalapani *et al.* CAOS 2009

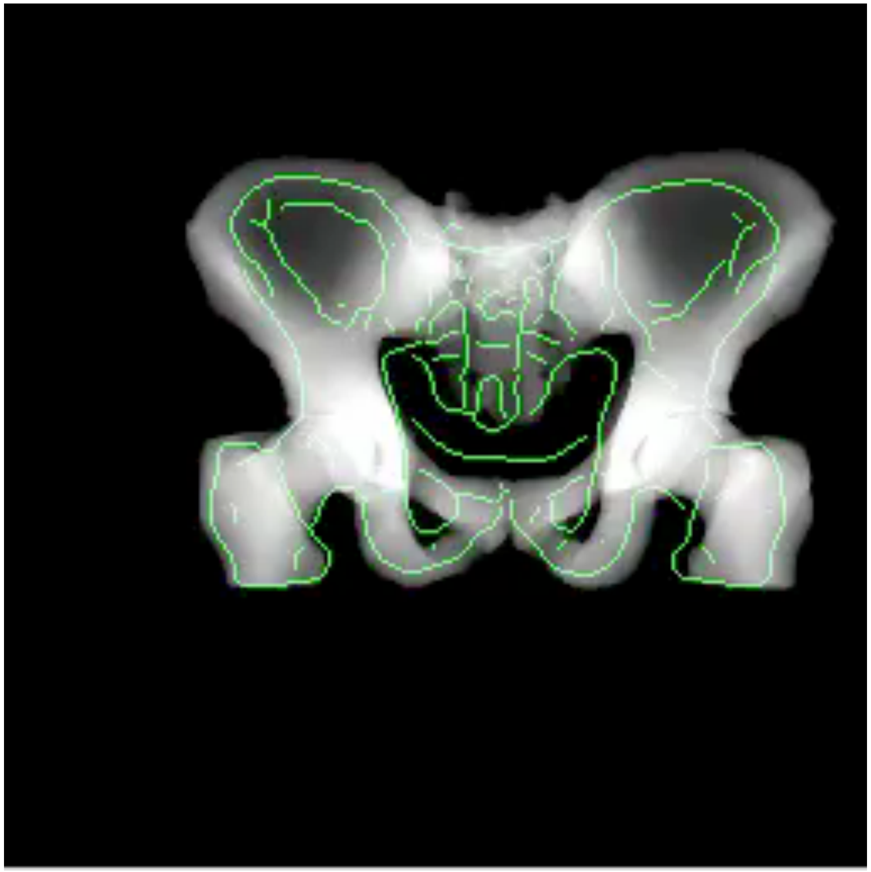
Slide credit: Gouthami Chintalapani

2D/3D Registration – Hip Model

- Registration with truncated images
  - FOV: 160mm
  - Three views
- Avg surface registration accuracy: 2.15 mm



Atlas projections overlaid on DRR images after registration

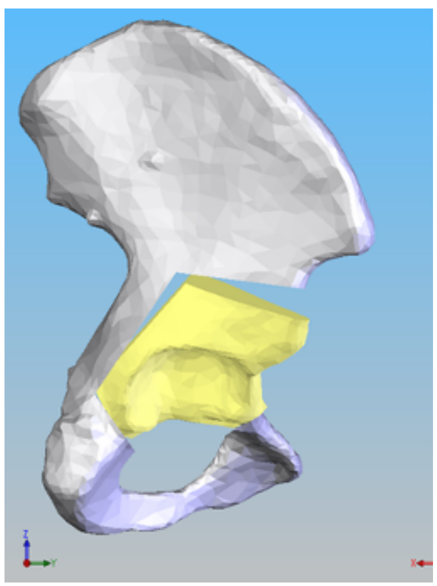


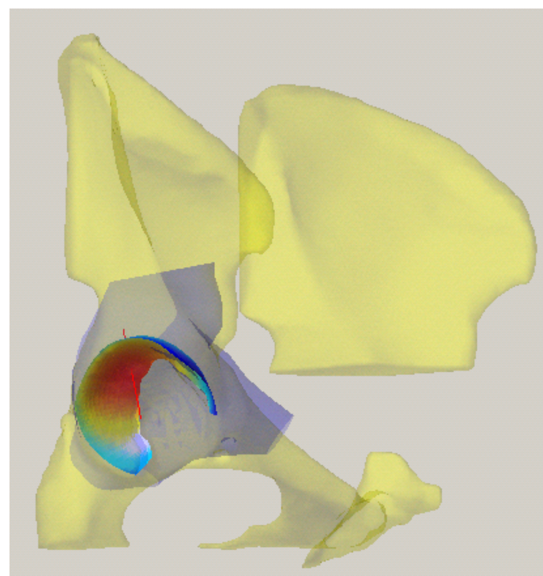
2D/3D deformable registration

Chintalapani *et al.* CAOS 2009

Slide credit: Gouthami Chintalapani

Applications – Hip Osteotomy

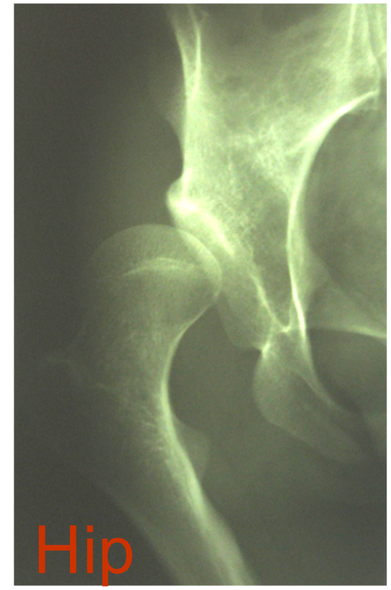


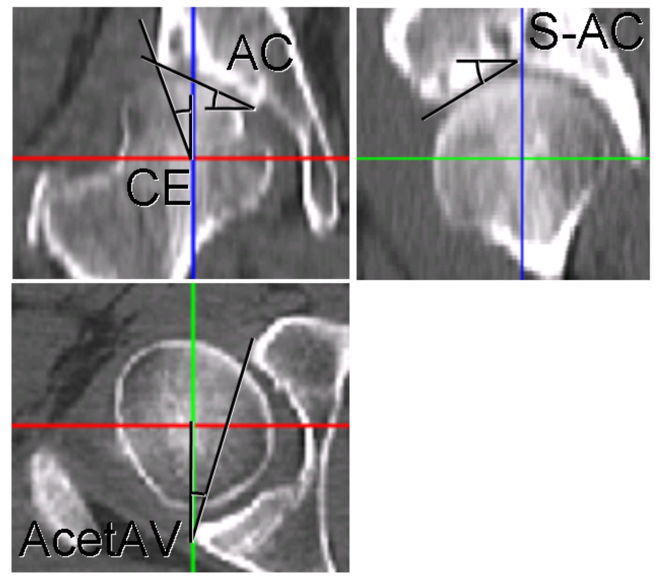


Slide credit: Gouthami Chintalapani

Background

- **Hip dysplasia:**
  - Malformation of the hip (normally a ball and socket joint)
  - Significant cause of osteoarthritis, especially in young adults
- **Surgery goals:**
  - Reduce pain symptoms
  - Realign joint to contain the femoral head
  - Diminish risk for degenerative joint changes
  - Improve contact pressure distribution
- **Periacetabular Osteotomy (PAO):**
  - Maintains pelvic structural stability
  - Preserves viable vascular supply
  - Technically challenging tool placement and realignment procedure
- **Limitations of current navigation systems:**
  - Lack the ability to track bone fragment alignment
  - Do not provide anatomical measurements
  - Omit biomechanical-based planning and guidance
  - Ignore the risk of reducing joint range-of-motion



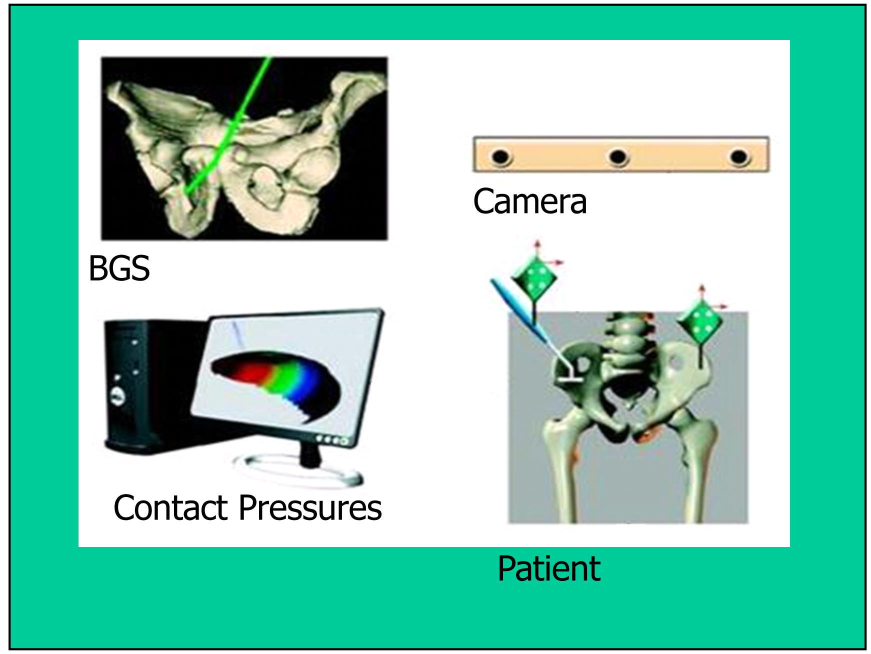


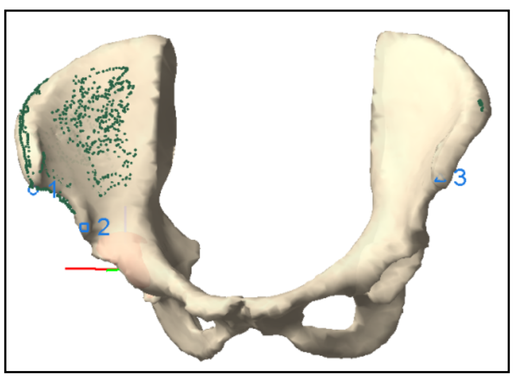
Anatomical measurements used to diagnose hip dysplasia

Slide credit: Gouthami Chintalapani, Mehran Armand

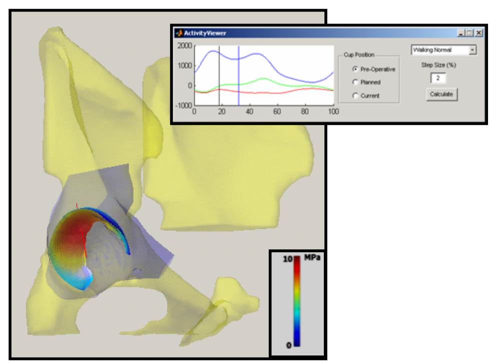
Biomechanical Guidance System (BGS)

- **BGS Preoperatively:**
  - Plans surgical cuts
  - Optimizes contact pressures and joint realignment
  - Calculates anatomical-based angles that are meaningful to the surgical team
- **BGS Intraoperatively:**
  - Tracks surgical tools and bone fragment alignment
  - Computes resulting contact pressures
  - Calculates hip range-of-motion
  - Visualizes the surgical cuts
  - Displays radiation-free Digitally Reconstructed Radiographs (DRR)

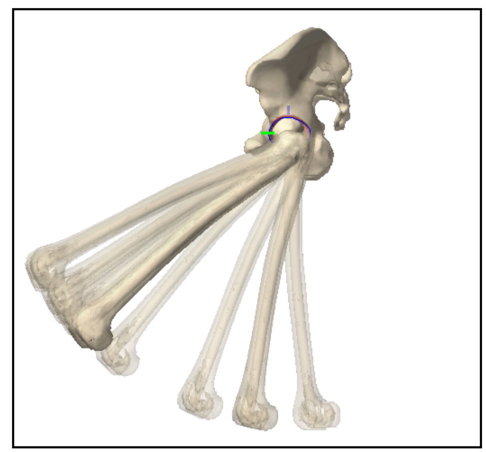




Model to Patient Registration



Joint contact-pressure after PAO

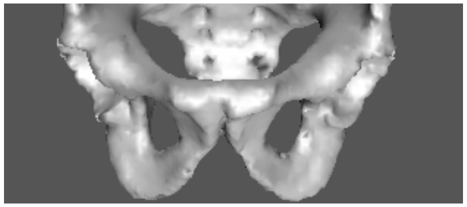


Hip-range-of-motion

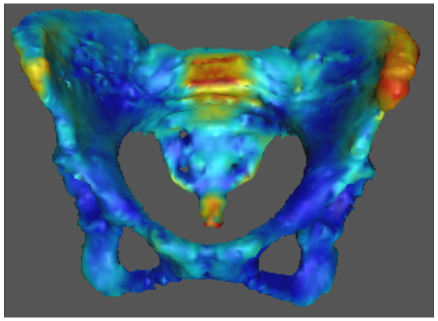
Slide credit: Gouthami Chintalapani, Mehran Armand

Atlas Based Extrapolation of CT

- **Problem**: Partial CT scans of patients
  - Dose minimization for young female patients
  - But the BGS needs full pelvis CT for planning
- **My approach**: Use atlas to predict the missing data
  - Robust probabilistic atlases
  - Improve prediction using pre-op and intra-op x-ray images
- **Preliminary Results**
  - Comparable to the registration errors from full CT scans



Typical pre-operative CT scan of a dysplastic patient undergoing osteotomy



Distribution of surface registration errors of a patient pelvis model estimated from partial CT scan

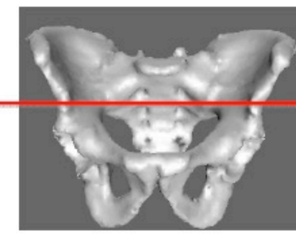
Chintalapani *et al.* SPIE 2010

Atlas Adaptation to Partial Data

Given a statistical shape model with mean $\bar{S}$ and modes $\mathbf{U} = \{U^{(1)} \ldots U^{(M)}\}$

Rearrange vertex indices and partition model into components corresponding to known and unknown parts

$$\bar{S} = \begin{bmatrix} \bar{S}_I \\ \bar{S}_J \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} \mathbf{U}_I \\ \mathbf{U}_j \end{bmatrix}$$



Find a set of registration parameters $(s, \mathbf{R}, \vec{\mathbf{p}}, \vec{\lambda})$

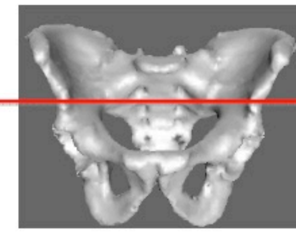
$$(s, \mathbf{R}, \vec{\mathbf{p}}, \vec{\lambda}) = \arg\min \left\| S_J^{(obs)} - \left(s\mathbf{R}\left(\bar{S}_J + \mathbf{U}_J \vec{\lambda} \right) + \vec{\mathbf{p}} \right) \right\|$$

Estimate the total shape as

$$S^{(est)} = \begin{bmatrix} \left(s\mathbf{R}\left(\bar{S}_I + \mathbf{U}_I \vec{\lambda} \right) + \vec{\mathbf{p}} \right) \\ S_J^{(obs)} \end{bmatrix}$$

Chintalapani *et al.* SPIE 2010

Atlas Adaptation to Partial Data with Xray Images



- 2D/3D registration[2] of inferred data with X-ray images

$$\left(s,\mathbf{R},\vec{\mathbf{p}},\vec{\lambda}\right) = \arg\max \sum_k MI(I_k, DRR(DensityAtlas, s\mathbf{R}\left(\bar{S}_J + \mathbf{U}_J\vec{\lambda}\right) + \vec{\mathbf{p}}))$$

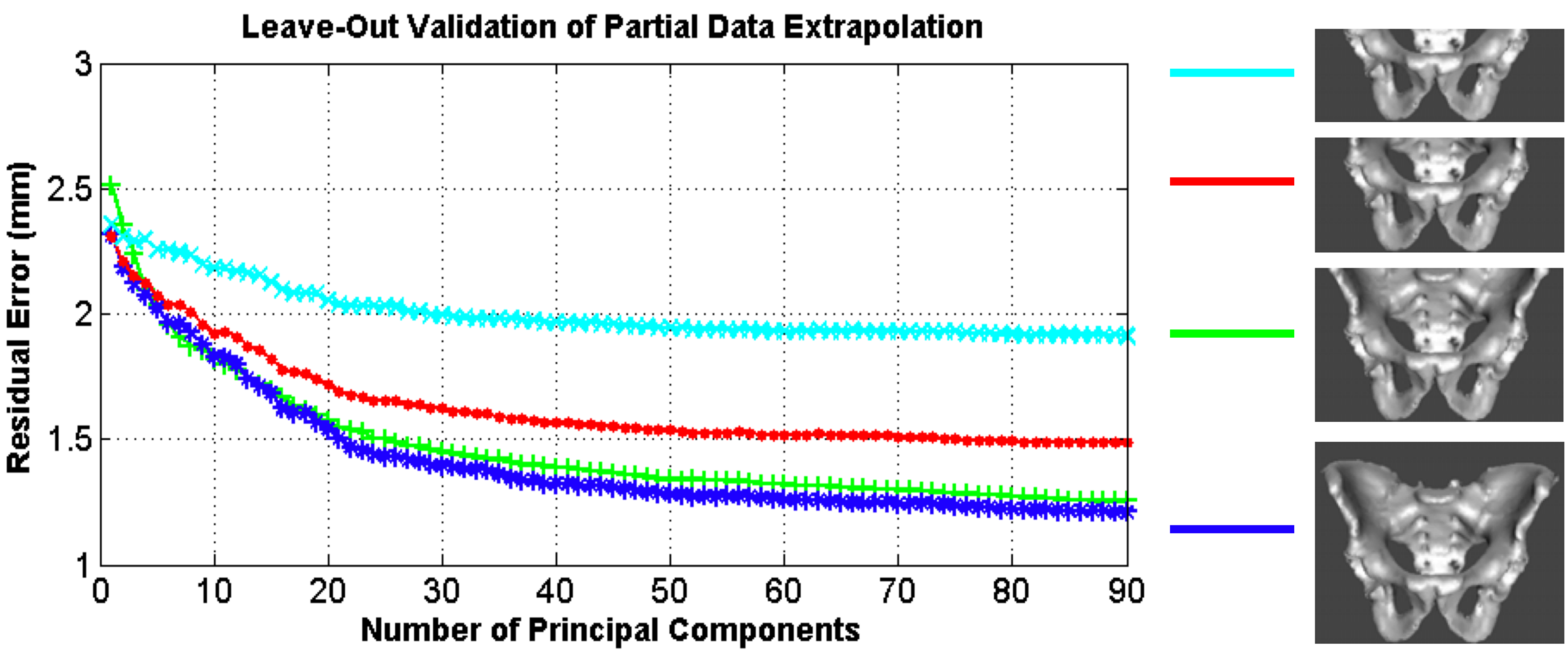
- Final atlas extrapolated model is given as

$$S^{(est)} = \begin{bmatrix} \left(s\mathbf{R}\left(\bar{S}_I + \mathbf{U}_I\vec{\lambda}\right) + \vec{\mathbf{p}}\right) \\ S_J^{(obs)} \end{bmatrix}$$

Chintalapani *et al.* SPIE 2010

Slide credit: Gouthami Chintalapani

Results



Slide credit: Gouthami Chintalapani

Results – Atlas Experiments

#	Full CT				Partial CT				Partial CT + X-ray			
	mean	max	std	95%	mean	max	std	95%	mean	max	std	95%
1	1.41	8.20	1.06	3.45	1.97	14.06	1.69	5.17	1.37	10.94	1.13	3.54
2	1.88	7.25	1.42	4.71	2.15	12.25	1.73	5.28	1.73	14.78	1.71	4.51
3	1.55	7.72	1.20	3.77	2.45	11.33	2.08	6.89	1.41	6.81	1.10	3.54
4	1.32	5.77	1.01	3.27	1.69	9.06	1.43	4.58	1.21	6.80	1.03	3.27
5	1.72	8.29	1.17	3.79	1.62	6.87	1.24	3.93	1.36	8.17	1.13	3.61
6	1.69	10.58	1.55	4.78	2.64	14.87	2.27	7.18	1.71	11.33	1.54	5.06
avg	1.59	7.96	1.23	3.96	2.08	11.40	1.74	5.50	1.46	9.80	1.27	3.92

Atlas inferred CT using full CT scan

Atlas extrapolated CT using partial CT scan

Atlas extrapolated CT using partial CT scan and X-ray images

Chintalapani *et al.* SPIE 2010

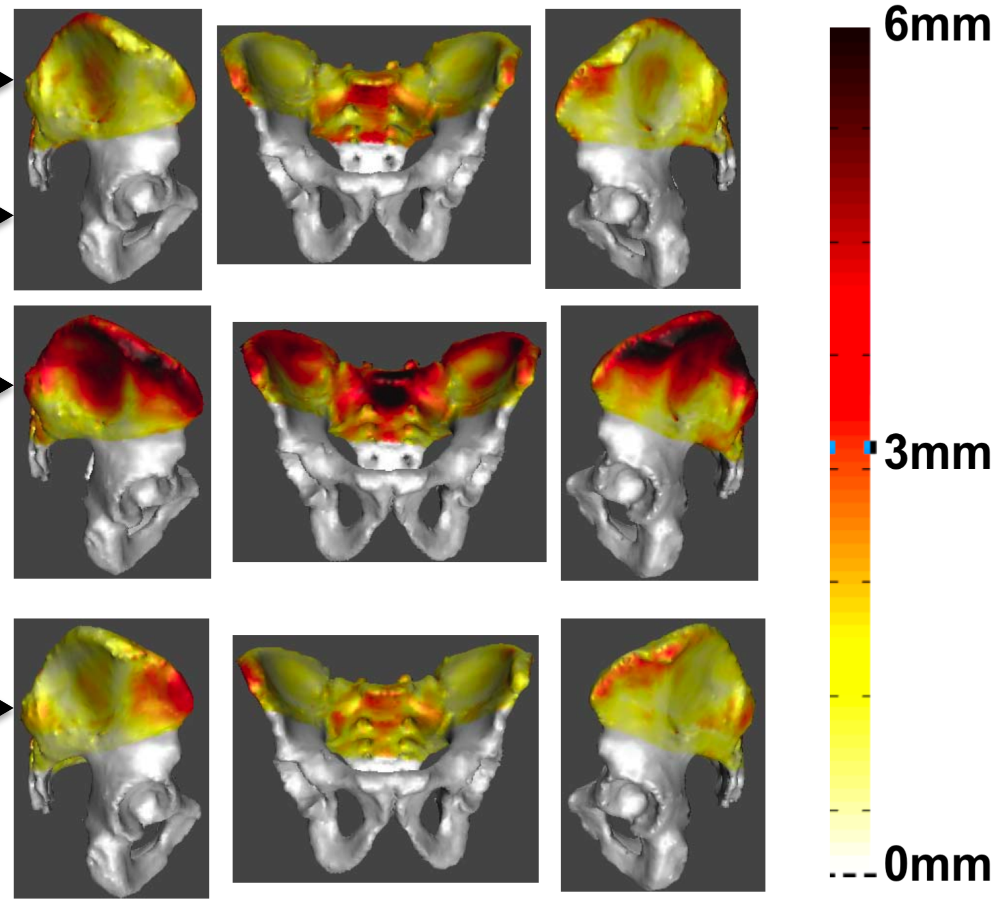
Results – Atlas Experiments

Inferred CT from full CT scan

True CT scan

Extrapolated CT from partial CT scan

Extrapolated CT from partial CT scan and X-ray images

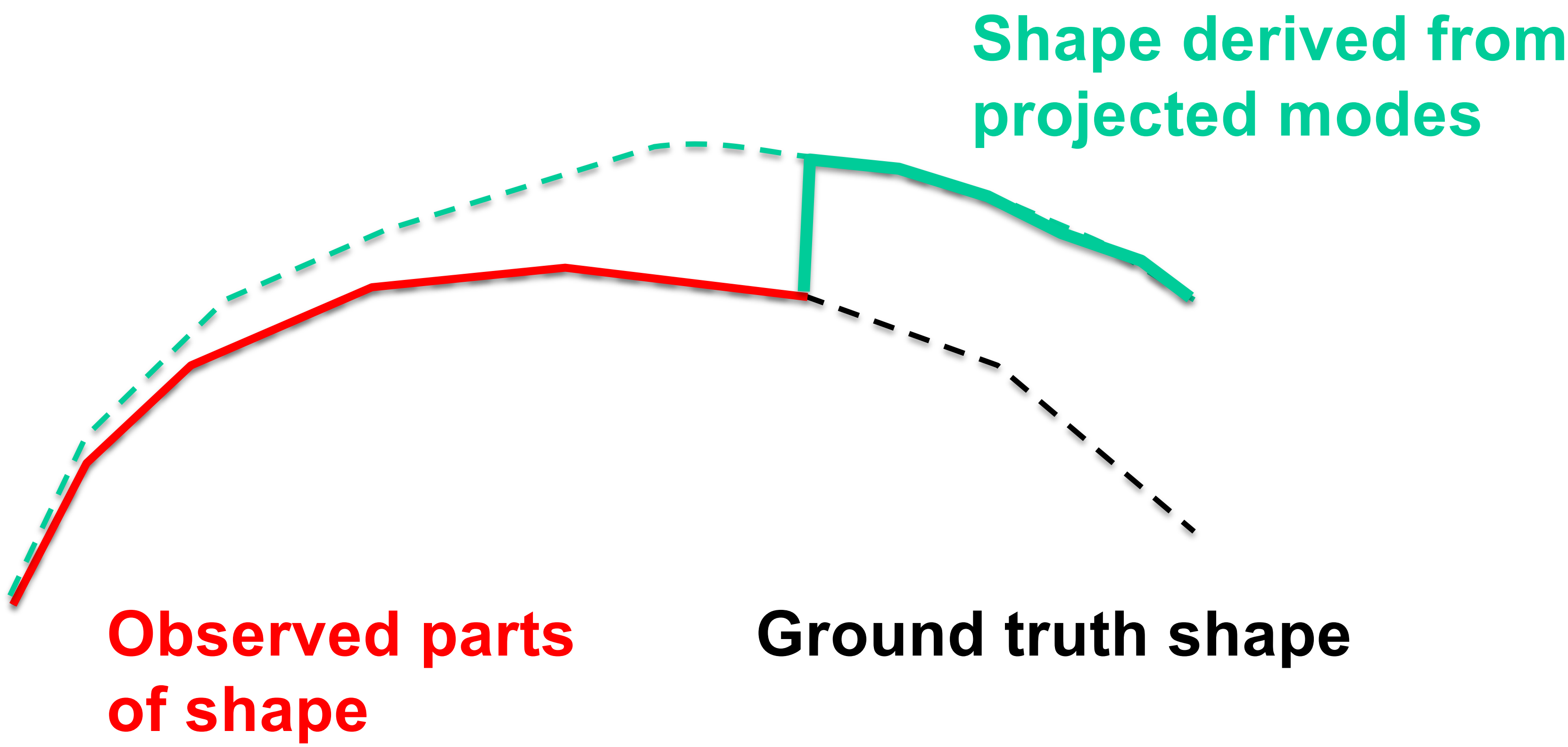


Distribution of surface errors between atlas extrapolated models and the true CT model

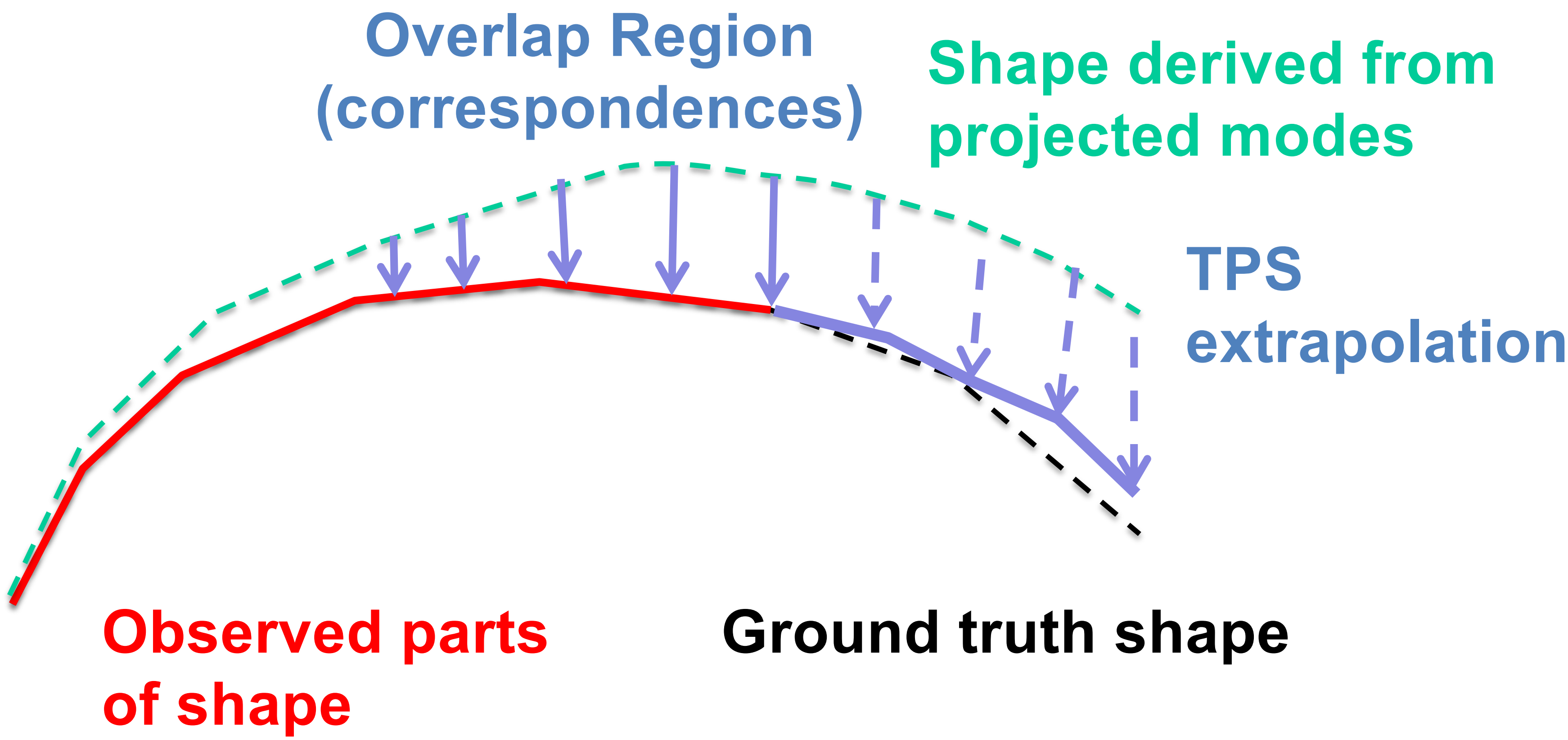
Chintalapani *et al.* SPIE 2010

Slide credit: Gouthami Chintalapani, Mehran Armand

Cut-and-Paste Model Completion



Model Completion with Thin Plate Spline



R. B. Grupp, H. Chiang, Y. Otake, R. J. Murphy, C. R. Gordon, M. Armand, and R. H. Taylor, "Smooth extrapolation of unknown anatomy via statistical shape models", in Proc. SPIE 9415, Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling, San Francisco, 8-10 Feb., 2015. p. 941524. 10.1117/12.2081310

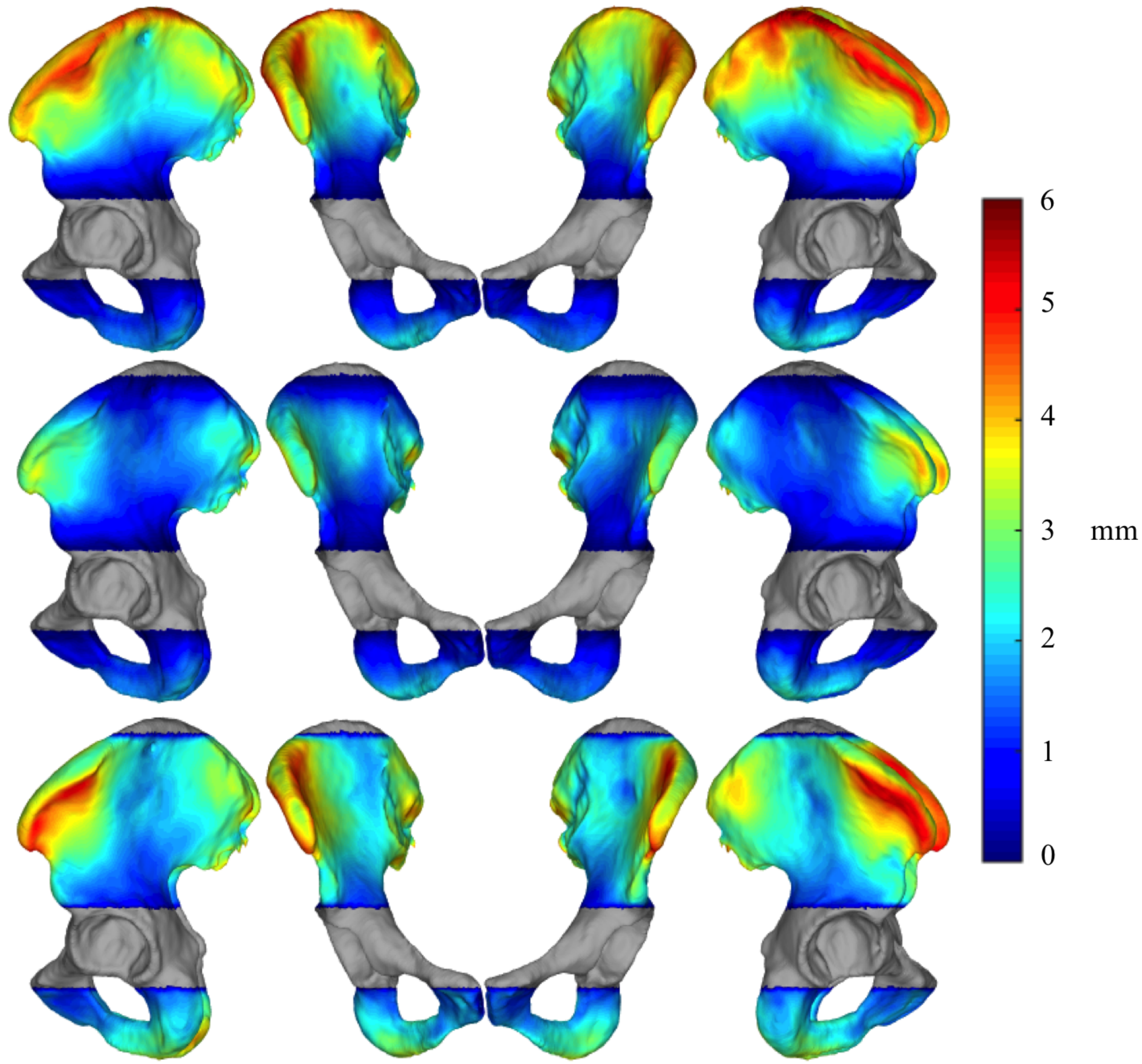
Model Completion of Pelvis from Partial CT Only

R. Grupp, R. Taylor, *et al.,* CAOS 2015

Smooth extrapolation using only acetabulum scan

Smooth extrapolation using only acetabulum scan + 5% of iliac crest

Naïve cut-and-paste extrapolation using only acetabulum scan + 5% of iliac crest



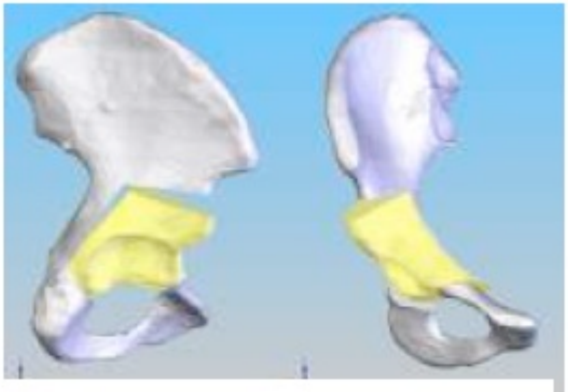
R. Grupp, Y. Otake, R. Murphy, J. Parvizi, M. Armand, and R. Taylor, "Pelvis surface estimation from partial CT for computer-aided pelvic osteotomies," in Computer Assisted Orthopaedic Surgery, Vancouver, June 17-19, 2015..

Osteotomy Simulations

- Atlas extrapolated model is used primarily for two reasons:

1. Model to patient registration
   - simulation experiments
   - six leave out experiments
   - FRE error metric

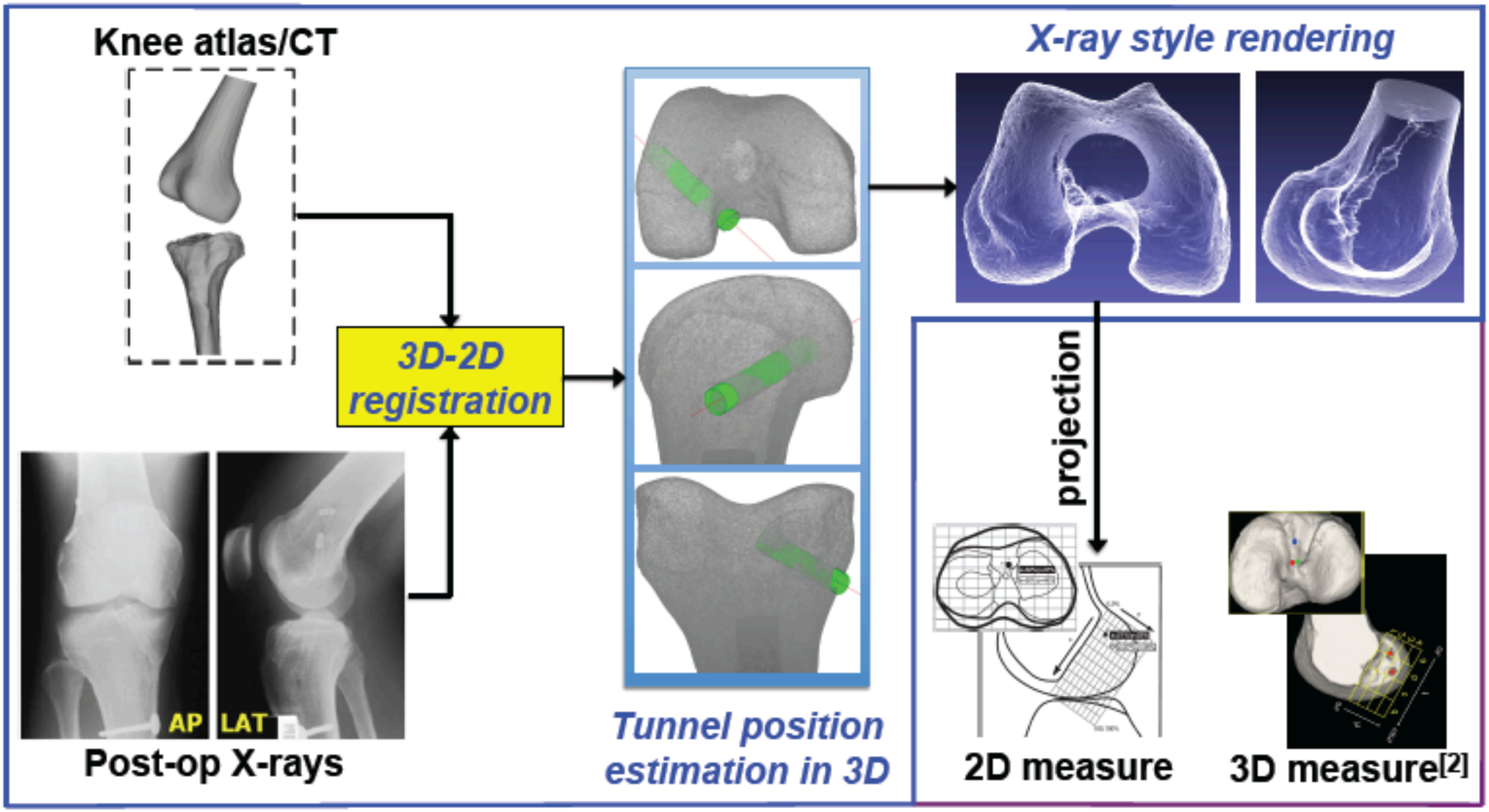
2. Fragment tracking
   - Simulated osteotomy cuts
   - Applied known transformation to the
   - Fragment
   - Computed the fragment transformation
   - Compared it to the known transformation



Slide credit: Gouthami Chintalapani, Mehran Armand

Statistical Assessment of ACL Tunnel Positions

Xin Kang, Russell Taylor, Yoshito Otake, Wai-Pan Yau



Basic Approach: Contour-based deformable 2D-3D registration

E-Step : Estimate p_{mn} from current $\mathbf{R}, \vec{\mathbf{t}}, \vec{\alpha}$, where

$p_{mn} =$ probability that projected model contour element m matches image contour element n

M-Step (Pose) :

$$\left[\mathbf{R}, \vec{\mathbf{t}}\right] = \arg\min \sum_{m,n} p_{mn} \left\| \vec{x}_n - T(\vec{X}_m; \mathbf{R}, \vec{\mathbf{t}}) \right\|$$

M-Step (Shape) :

$$\vec{\alpha} = \arg\min \rho \sum_{m,n} d_{mn}^2 + (1-\rho) \sum_k \frac{\alpha_k^2}{\lambda_k}$$

where

$$d_{mn} = \left\| \sqrt{p_{mn}}\, \vec{V}_n \left[\vec{C}_n - \left(\mathbf{R}\vec{X}_m^{(0)} + \vec{\mathbf{t}} + \sum_k \mathbf{R}\vec{\mathbf{e}}_m^{(k)} \right) \right] \right\|$$

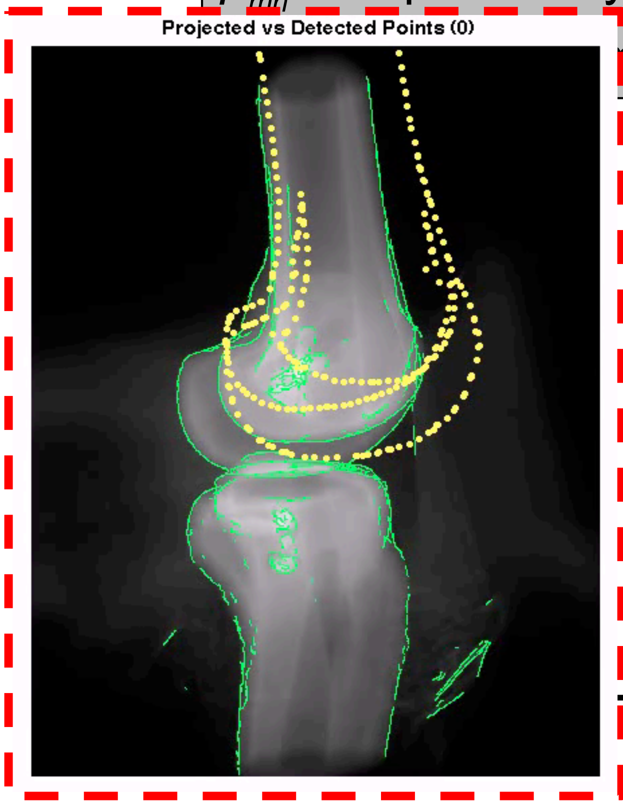
Xin Kang, Russell Taylor, Yoshito Otake, Wai-Pan Yau

Basic Approach: Contour-based deformable 2D-3D registration

E-Step: Estimate p_{mn} from current $\mathbf{R}, \vec{\mathbf{t}}, \vec{\alpha}$, where

$p_{mn} =$ probability that projected model contour

matches image contour element n



M-Step (Pose):

$$\left[\mathbf{R}, \vec{\mathbf{t}}\right] = \arg\min \sum_{m,n} p_{mn} \left\| \vec{x}_n - T(\vec{X}_m; \mathbf{R}, \vec{\mathbf{t}}) \right\|$$

M-Step (Shape):

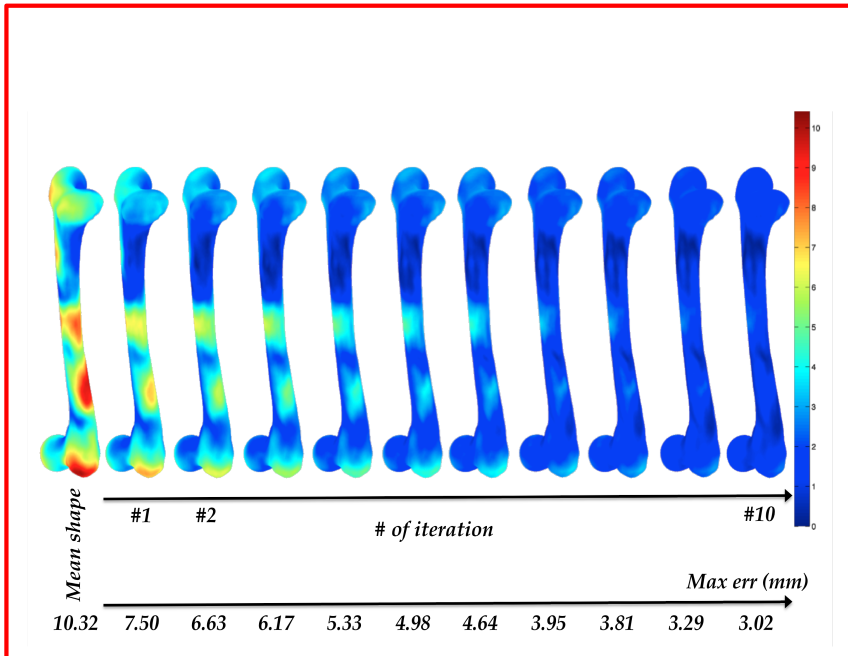
$$\vec{\alpha} = \arg\min \rho \sum_{m,n} d_{mn}^2 + (1-\rho) \sum_k \frac{\alpha_k^2}{\lambda_k}$$

where

$$d_{mn} = \left\| \sqrt{p_{mn}}\, \vec{V}_n \left[\vec{C}_n - \left(\mathbf{R}\vec{X}_m^{(0)} + \vec{\mathbf{t}} + \sum_k \mathbf{R}\vec{\mathbf{e}}_m^{(k)} \right) \right] \right\|$$

Xin Kang, Russell Taylor,
Yoshito Otake, Wai-Pan Yau

Basic Approach: Contour-based deformable 2D-3D registration



rrent $\mathbf{R}, \vec{\mathbf{t}}, \vec{\alpha}$, where

ted model contour

mage contour element n

(Pose):

$$\arg\min \sum_{m,n} p_{mn} \left\| \vec{x}_n - T(\vec{X}_m; \mathbf{R}, \vec{\mathbf{t}}) \right\|$$

M-Step (Shape):

$$\vec{\alpha} = \arg\min \rho \sum_{m,n} d_{mn}^2 + (1-\rho) \sum_k \frac{\alpha_k^2}{\lambda_k}$$

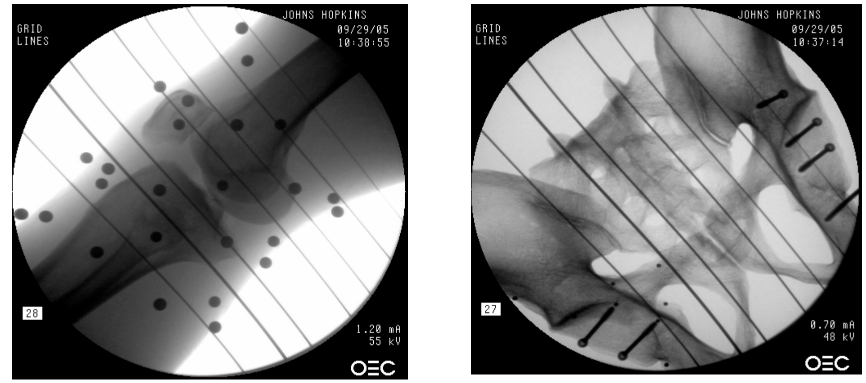
where

$$d_{mn} = \left\| \sqrt{p_{mn}}\, \vec{V}_n \left[\vec{C}_n - \left(\mathbf{R}\vec{X}_m^{(0)} + \vec{\mathbf{t}} + \sum_k \mathbf{R}\vec{\mathbf{e}}_m^{(k)} \right) \right] \right\|$$

Xin Kang, Russell Taylor,
Yoshito Otake, Wai-Pan Yau

C-arm Distortion

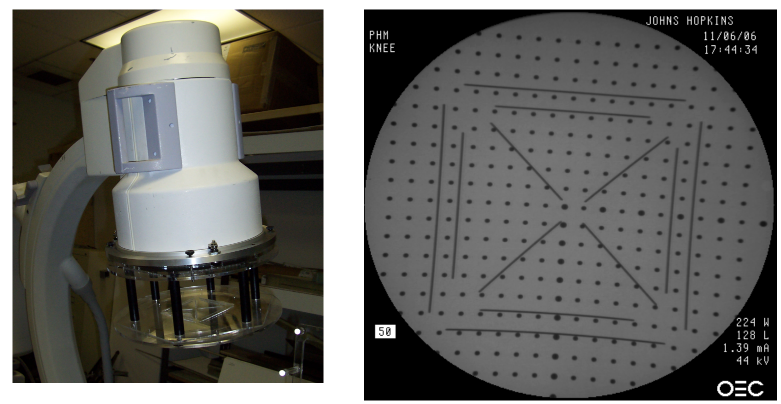
- What is distortion ?
  - –Avg distortion: **2.14** mm/pixel
  - –max distortion: **4.60** mm/pixel



Example C-arm images showing distortion, straight metal wires appear curved due to distortion

- How to rectify images ?
  - Phantom based correction
  - Polynomial functions to model distortion

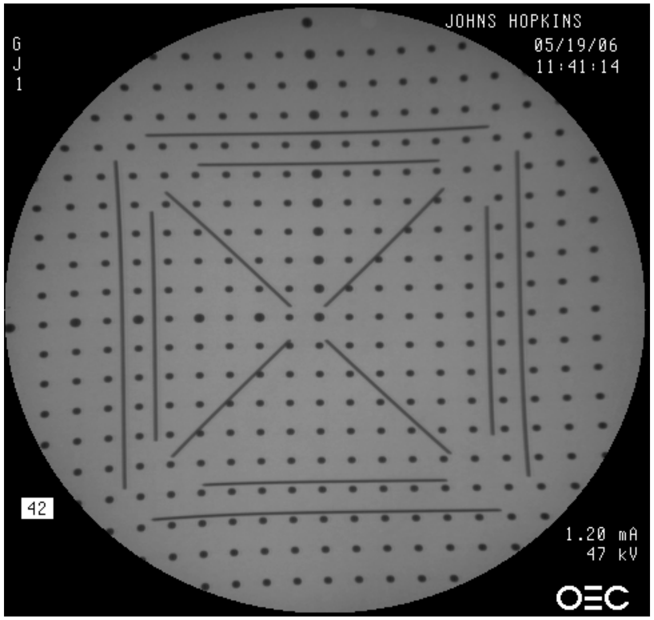
$$(u_d, v_d) = \sum_{i=0}^{n} \sum_{j=0}^{n} C_{ij} B_{ij}(u_0, v_0)$$



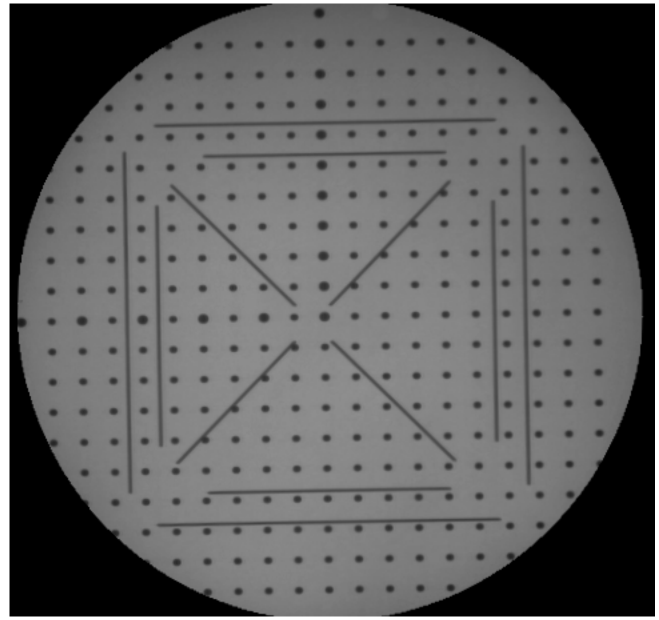
Typical bi-planar phantom used for C-arm calibration

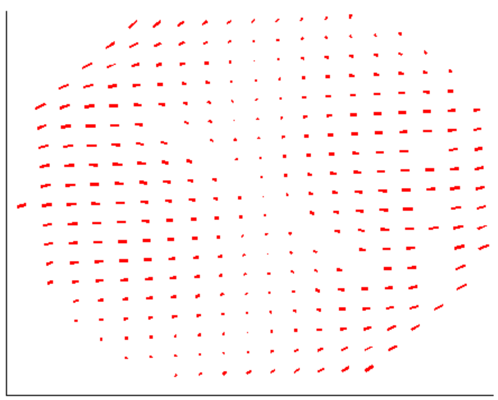
Slide credit: Gouthami Chintalapani

C-Arm Distortion Correction



Warped X-ray image of the phantom

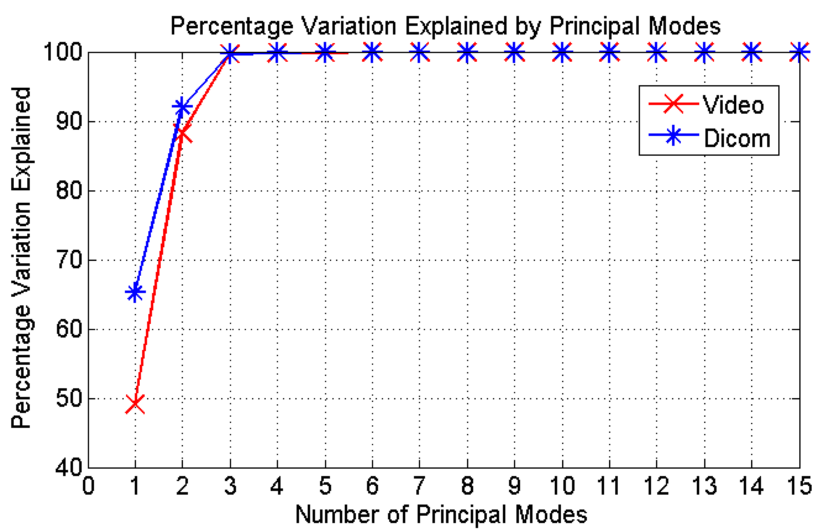

Dewarped X-ray image

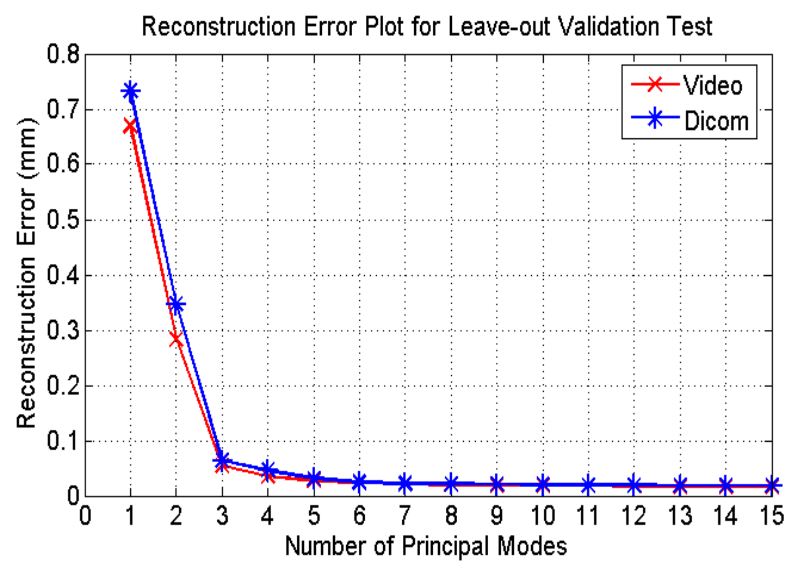

Distortion vector map

$$\triangle \overrightarrow{d} = (\triangle u, \triangle v) = (u_d, v_d) - (u_0, v_0)$$

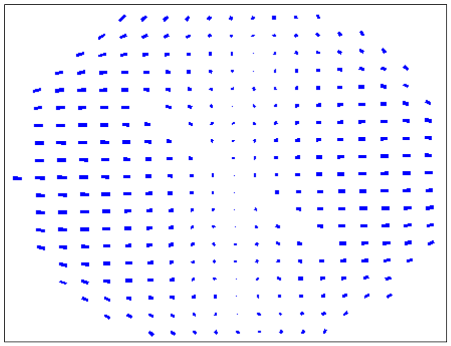
Statistical Characterization of C-Arm Distortion correction using PCA

- Principal component analysis on distortion maps
  - 120 images, one every 3 degrees approx., along propeller axis (similar to the full sweep data used for 3D reconstruction)
  - 200 images to span the sphere defined by the “C” of the c-arm

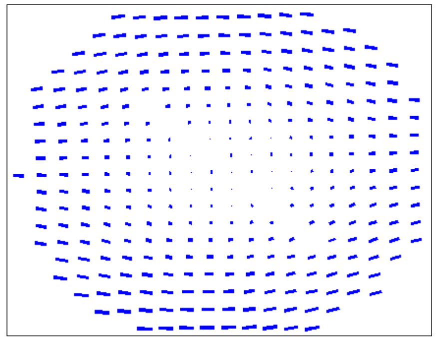




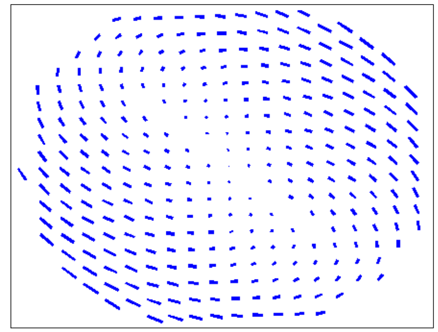
Circular Trajectory



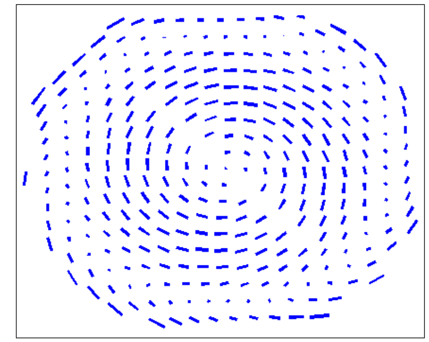
mean



mode1

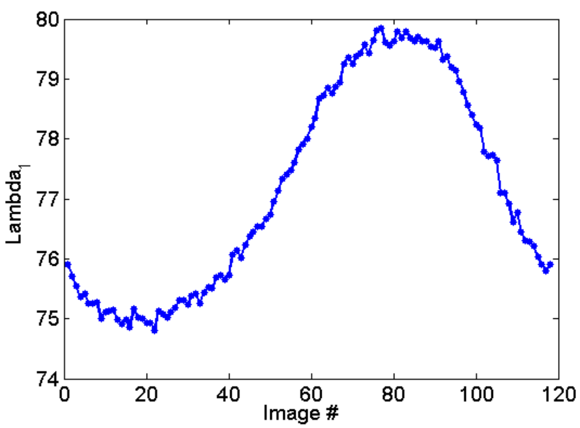


mode2

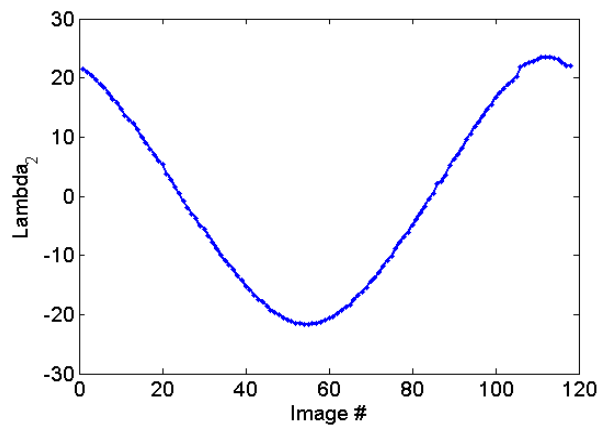


mode3

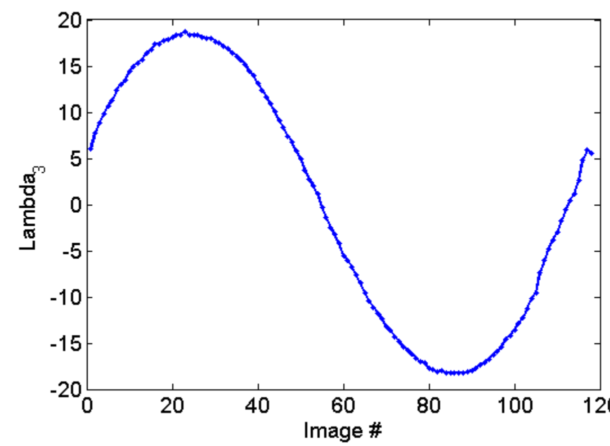
Distortion patterns from PCA modes



Lambda_1



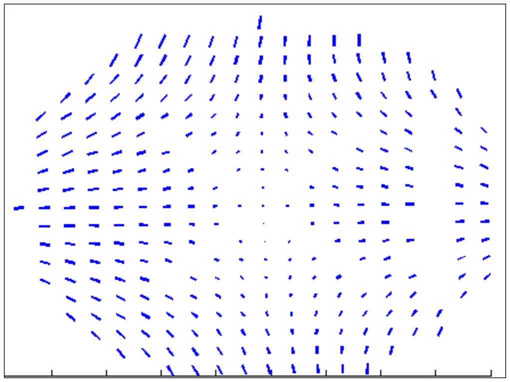
Lambda_2

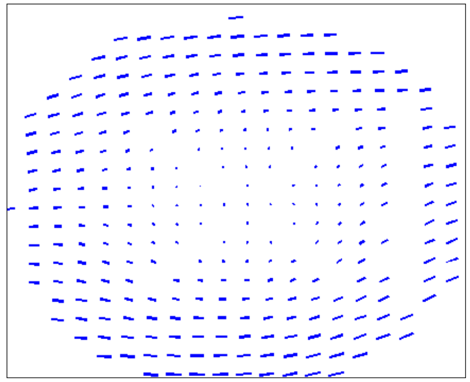


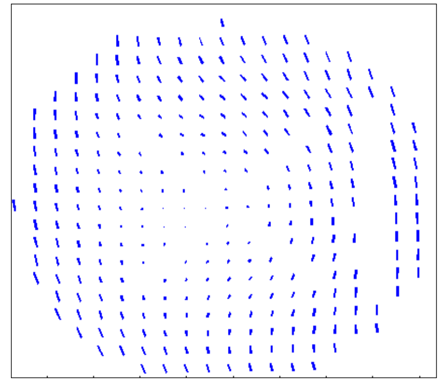
Lambda_3

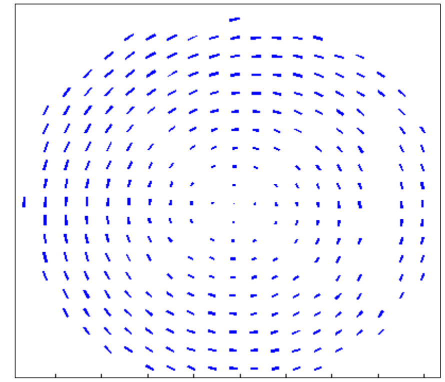
Slide credit: Gouthami Chintalapani

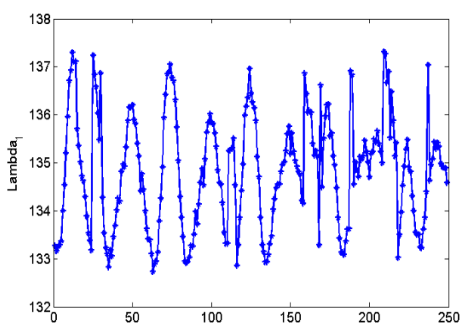
C-arm Imaging Volume


mode1

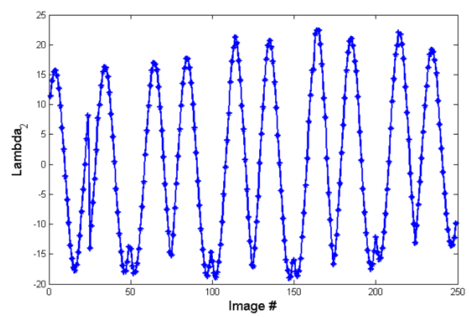

mode2


mode3

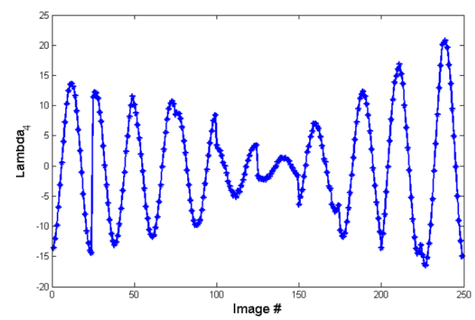

mode4



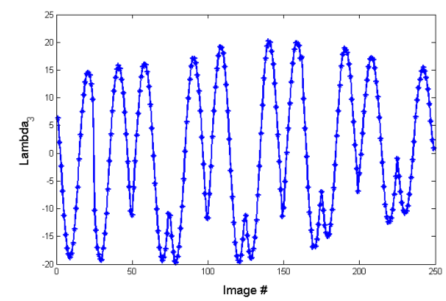
Lambda_1



Lambda_2



Lambda_3

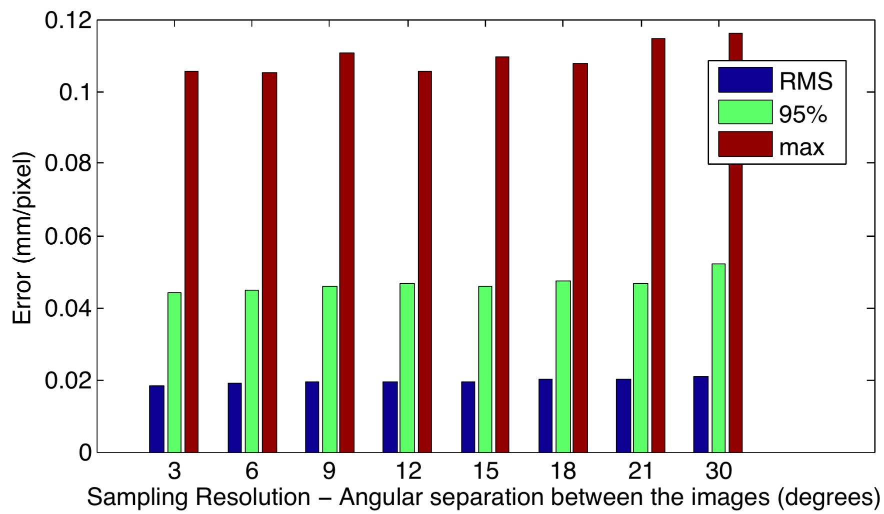


Lambda_4

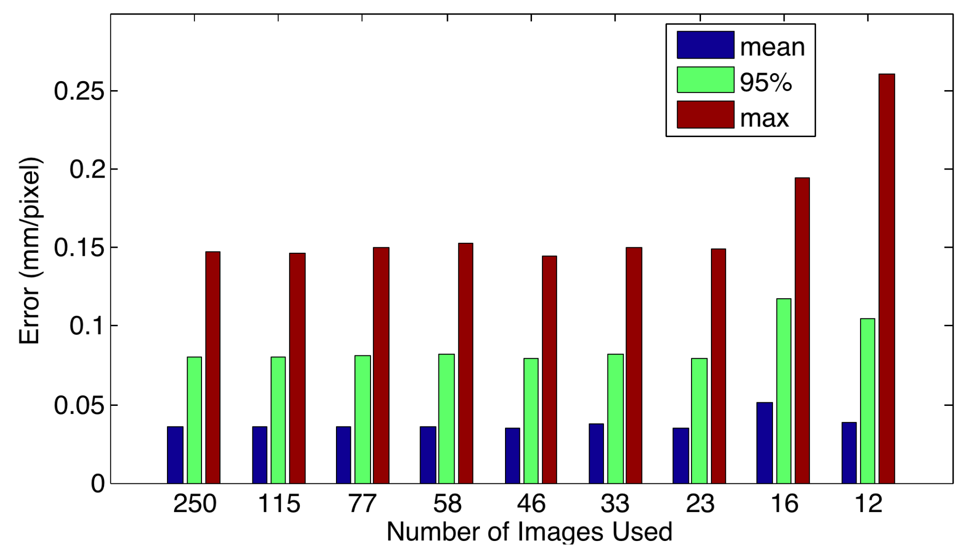
Slide credit: Gouthami Chintalapani

Sampling Resolution

- How many images are required to statistically characterize the distortion patterns ?



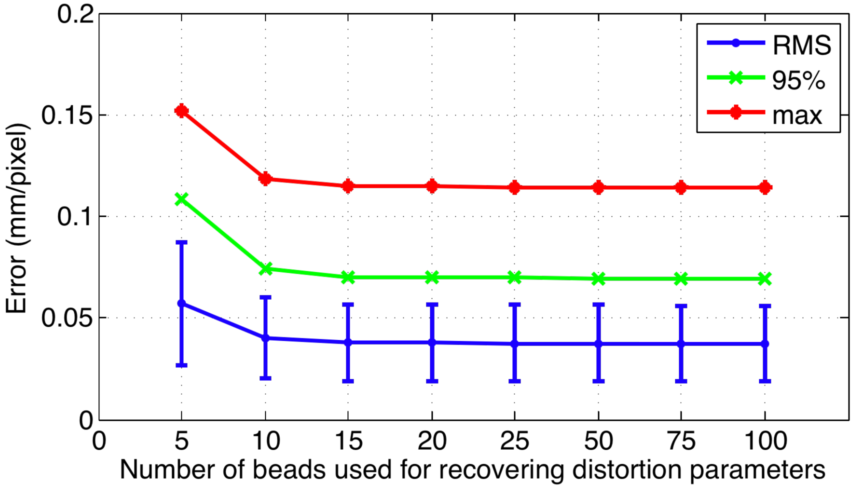
Circular Trajectory



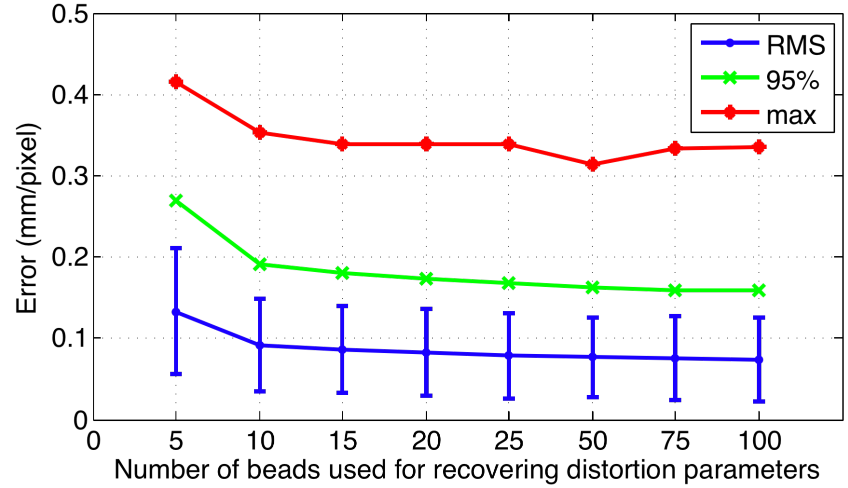
C-arm Imaging Volume

Recovering Distortion Parameters

- Use as few beads as possible to recover the distortion mode parameters

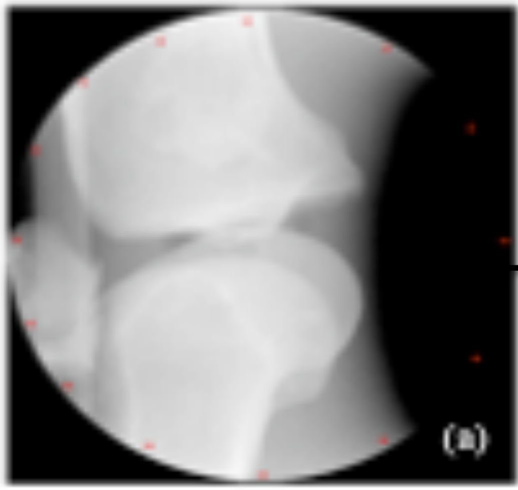


Circular Arc



C-arm Imaging Volume

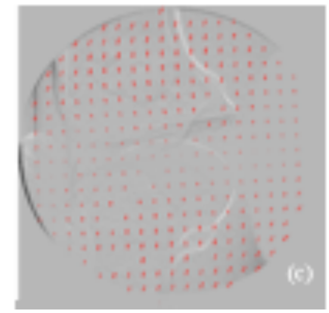
Small Phantom based Distortion Correction



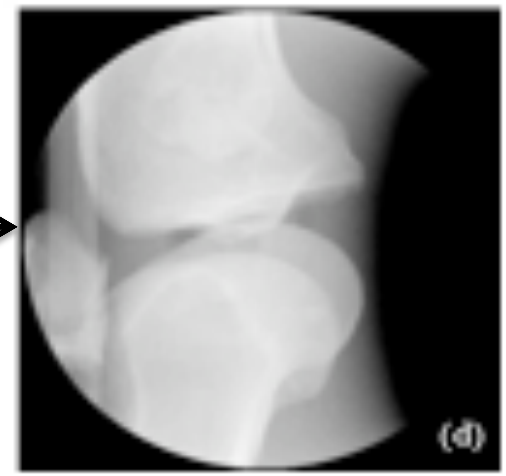
Example patient image peripheral beads

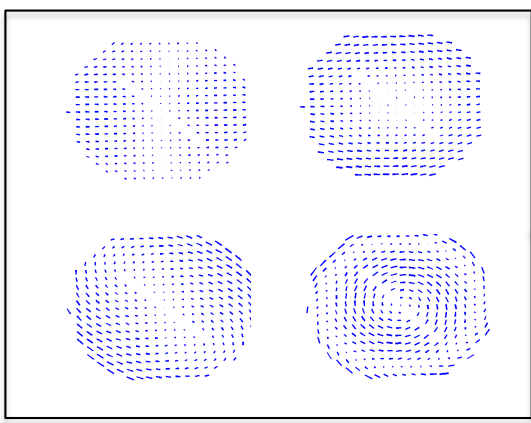
Peripheral bead locations

Distortion corrected image



Distortion mode matching





Prior distortion model

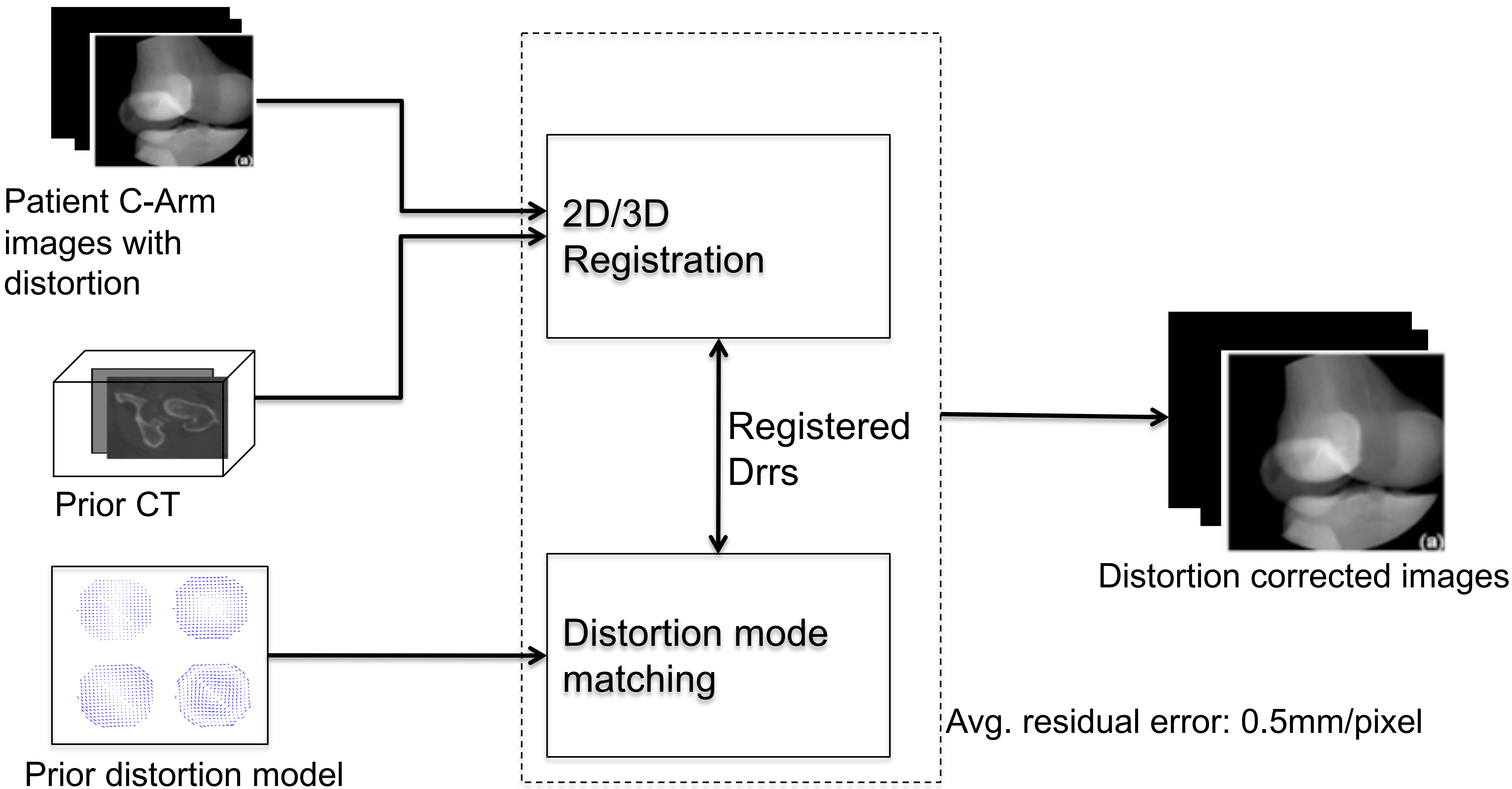
Distortion modes

Avg. residual error: 0.2mm/pixel
Max. residual error: 0.8mm/pixel

Chintalapani *et al.* SPIE 2007

Slide credit: Gouthami Chintalapani

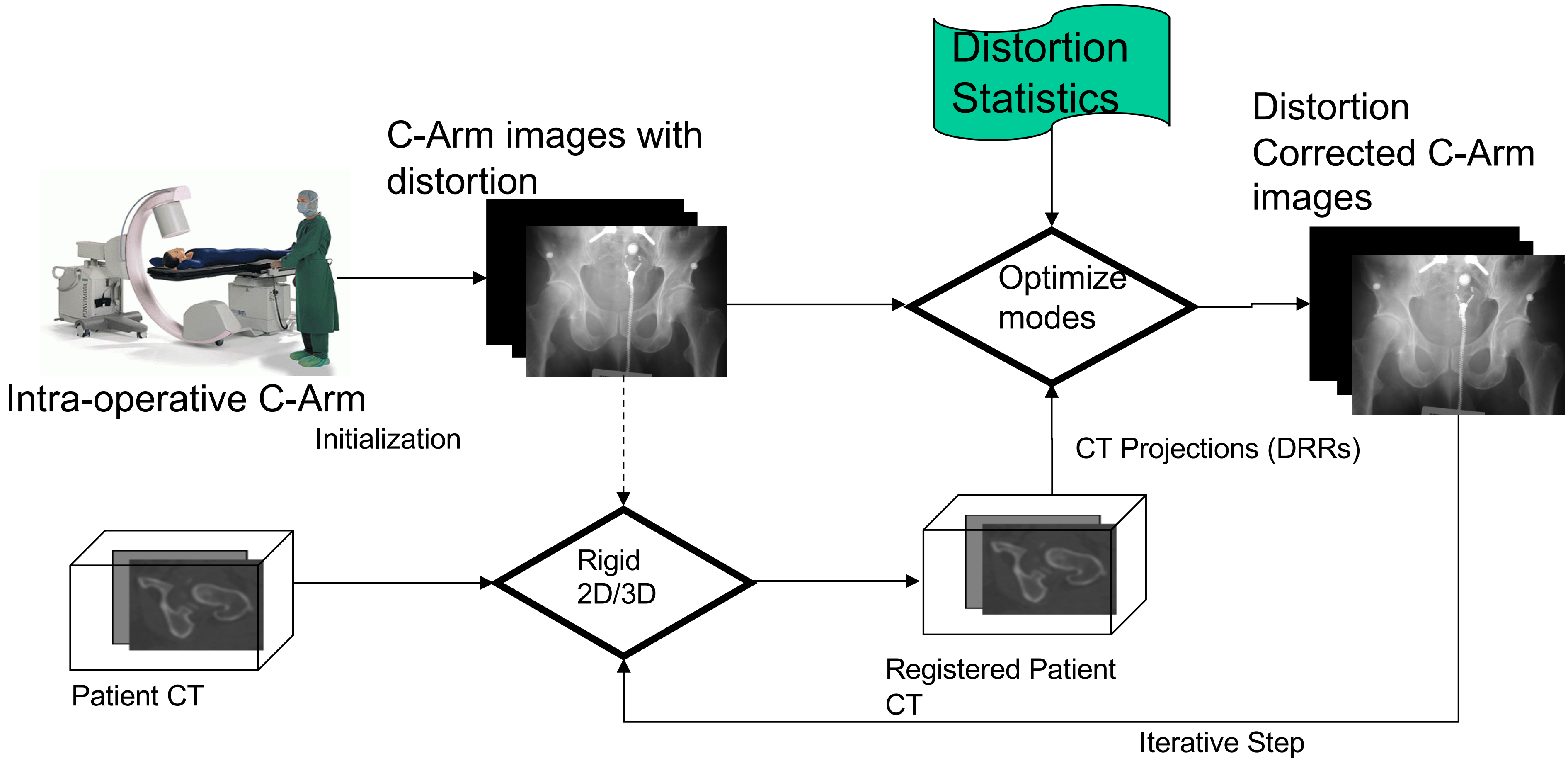
Using Patient CT as Fiducial



Chintalapani *et al.* ISBI 2007

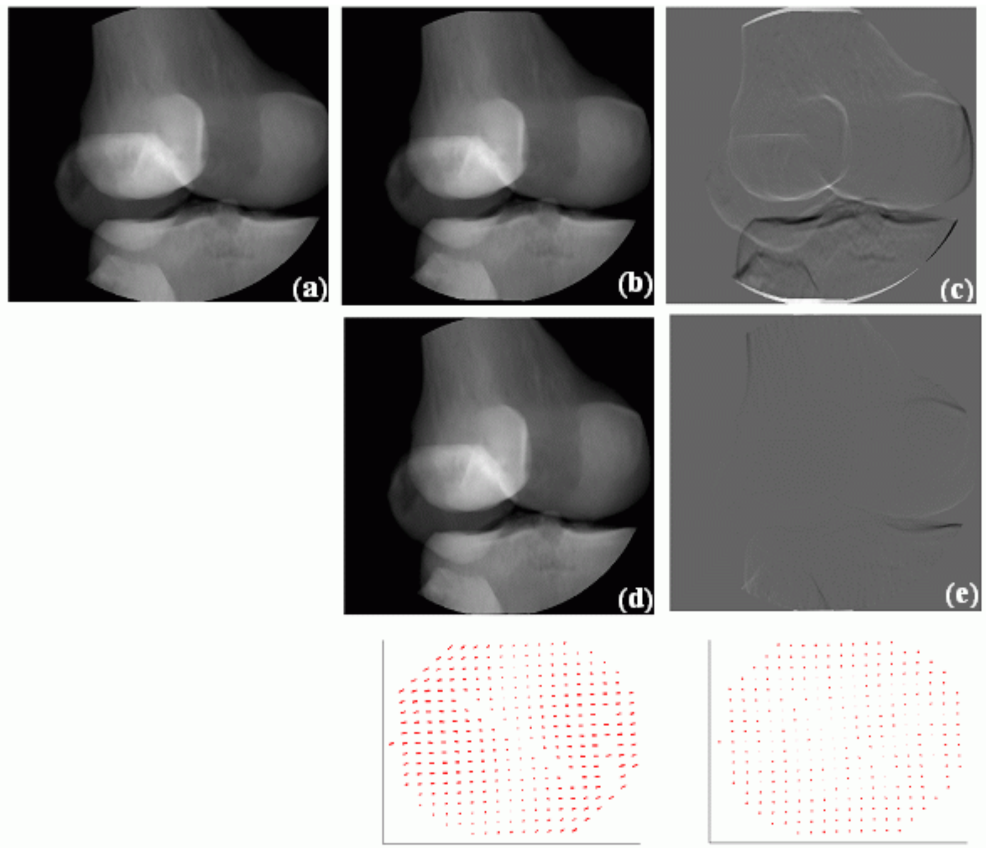
Slide credit: Gouthami Chintalapani

C-Arm Distortion Correction Using Patient CT as Fiducial



Thanks to Ofri Sadowsky for assistance with 2D/3D registration

C-Arm Distortion Correction Using Patient CT as Fiducial



Results from simulation experiments. (a) true projection; (b) warped projection (simulated x-ray); (c) difference between true and warped projection ((a) - (b)); (d) registered and distortion corrected projection; (f) (a) - (d); The bottom row shows the distortion map before and after correction.