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

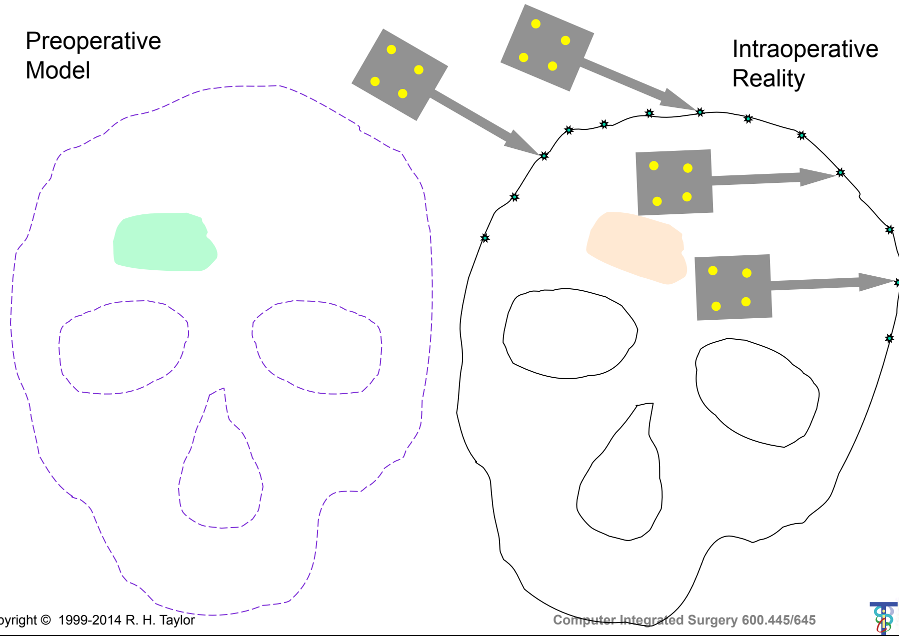
WHITING SCHOOL OF ENGINEERING
THE JOHNS HOPKINS UNIVERSITY

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

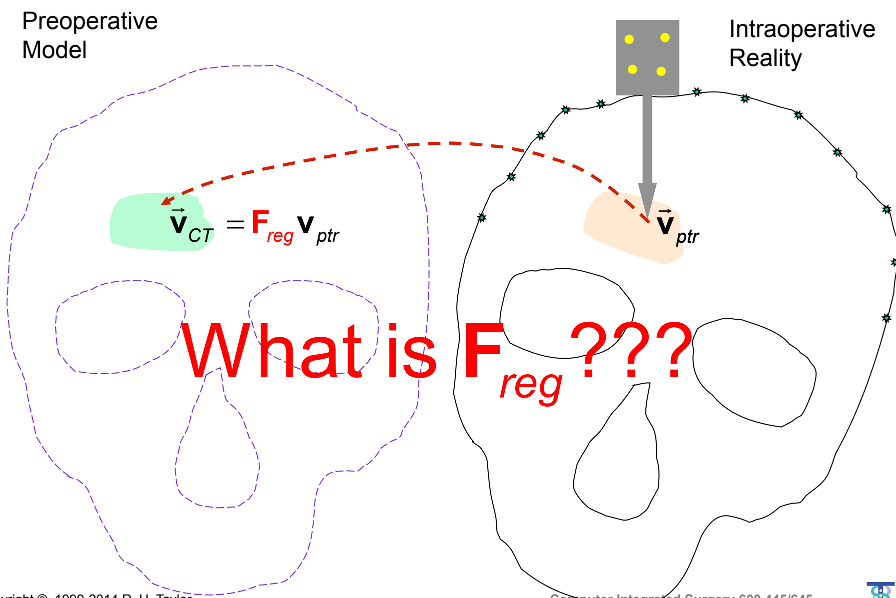
What needs registering?

- **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,

$$\mathbf{Ref_A} \;\&\; \mathbf{Ref_B}$$

and coordinates

$$\mathbf{x_A} \;\&\; \mathbf{x_B}$$

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:

$$\mathbf{x_A} = \mathbf{T}(\mathbf{x_B})$$

Definitions

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

  T(x) = R•x + p

  The key assumption is that deformations may be neglected.

- **Elastic Transformation:** Cases where must take 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.

Disparity 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:

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

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

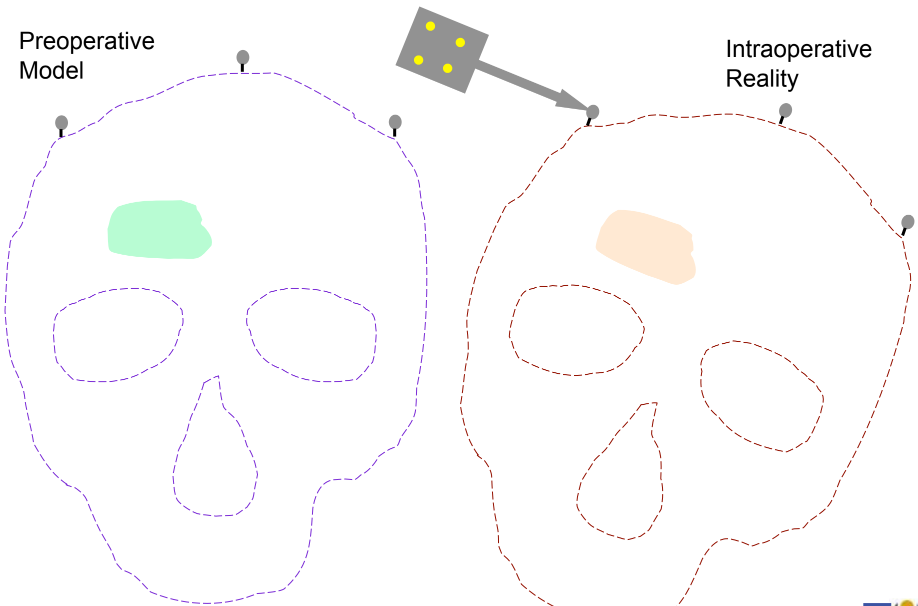
$$D = \underset{i}{\text{median}}\ distance(F_{Ai}, \mathbf{T}(F_{Bi}))$$

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

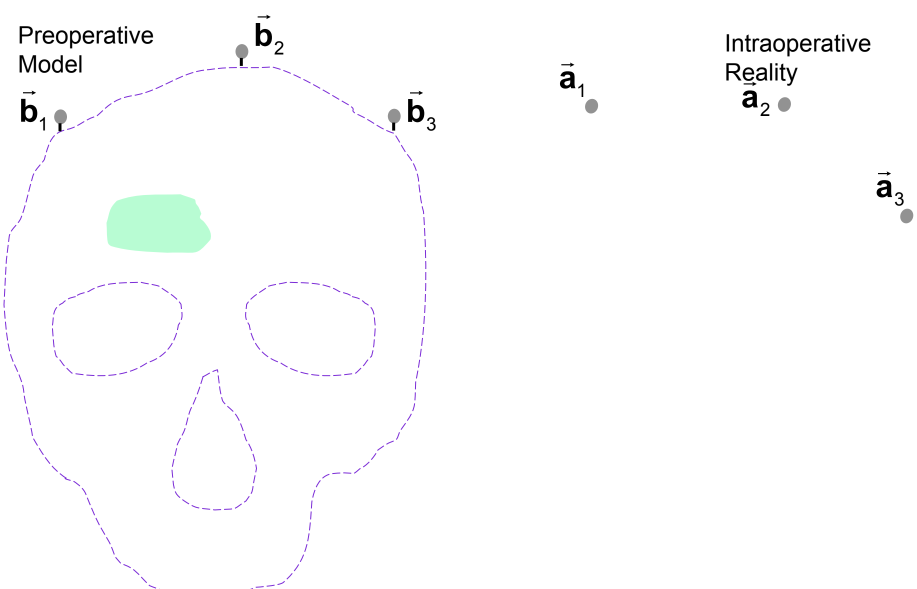
Optimization

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

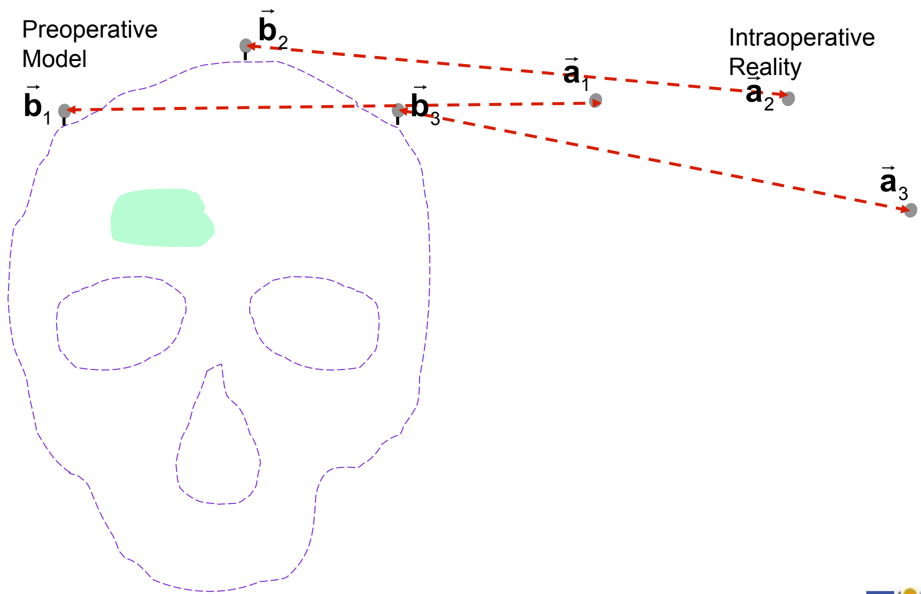
A typical fiducial-based registration problem



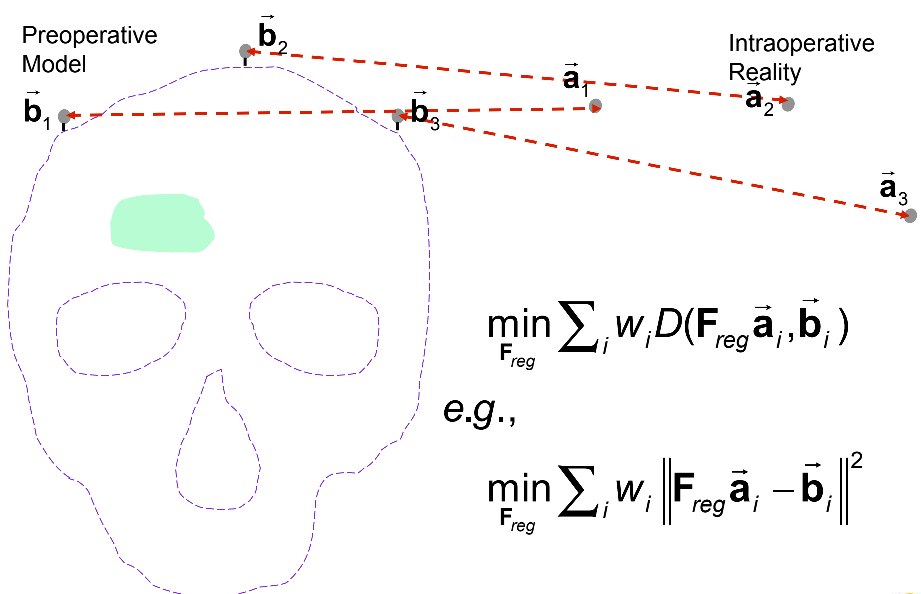
What the computer knows



Identify corresponding points



Find best rigid transformation!

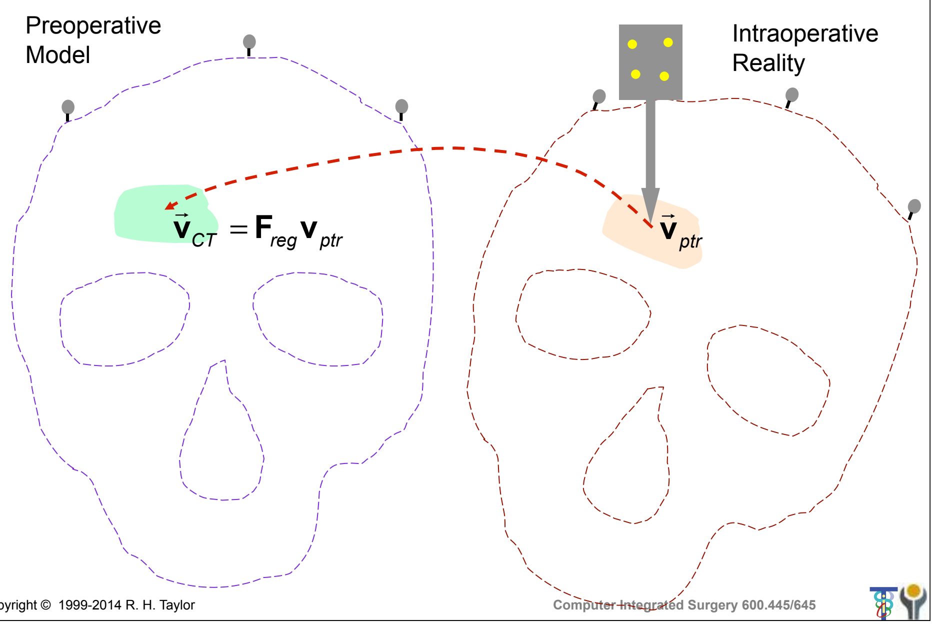


$$\min_{\mathbf{F}_{reg}} \sum_i w_i D(\mathbf{F}_{reg}\vec{\mathbf{a}}_i, \vec{\mathbf{b}}_i)$$

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



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.

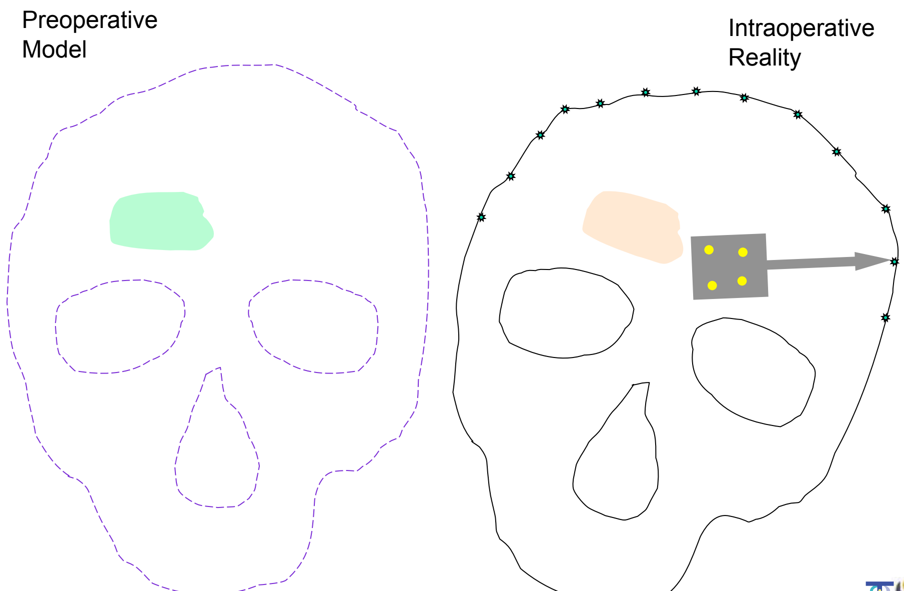
Examples

- 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]

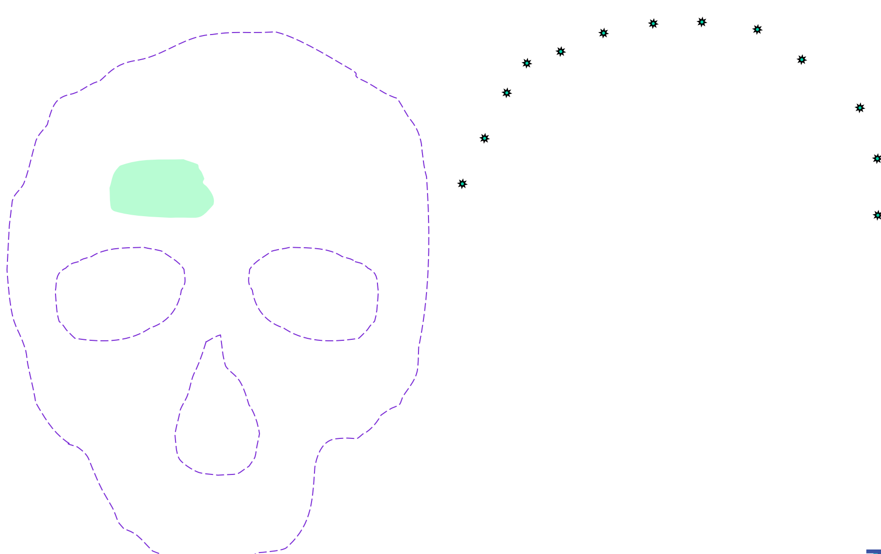
Copyright © 1999-2014 R. H. Taylor

Computer Integrated Surgery 600.445/645

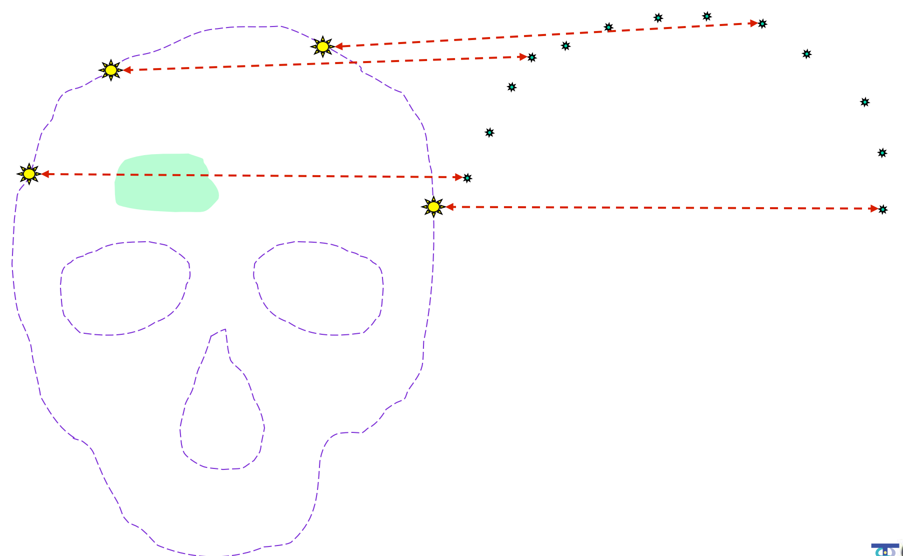
A typical surface registration problem



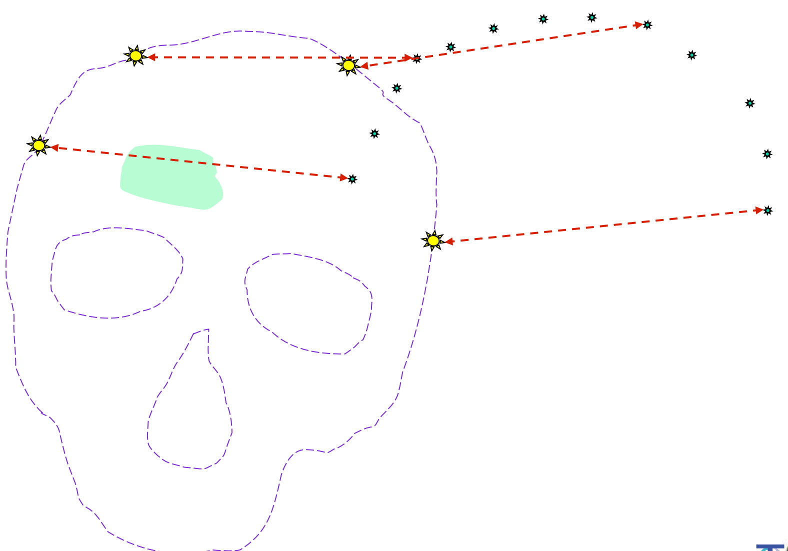
What the computer knows



Find corresponding points & pull!



Find corresponding points & pull!

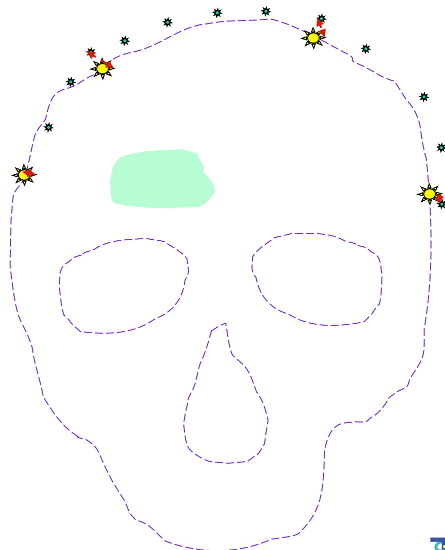


Find corresponding points & pull!

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

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

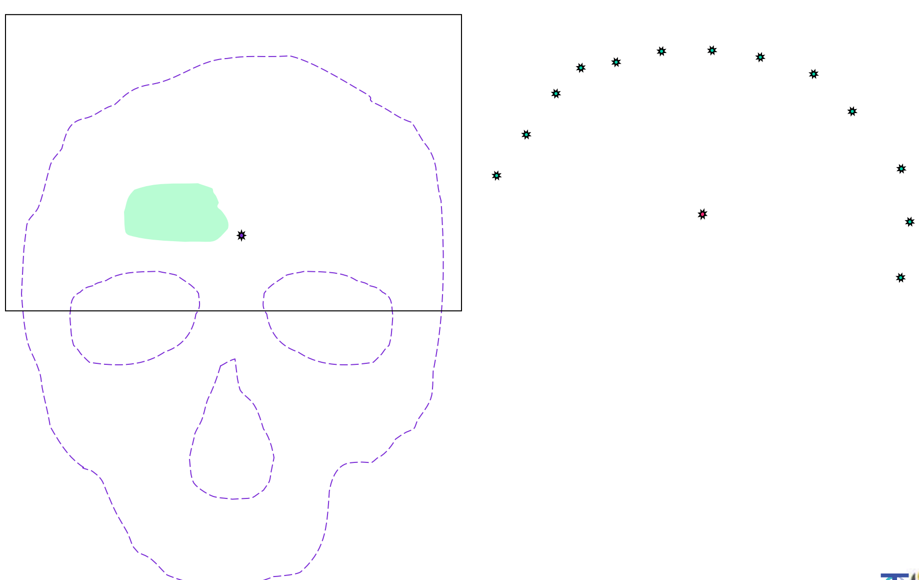
  where d_S is defined below, given a good initial guess for $\mathbf{T}$.
- Optimization uses standard numerical method (steepest gradient descent [Powell]) to find six parameters (3 rotations, 3 translations) defining $\mathbf{T}$.

Head in Hat Algorithm

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

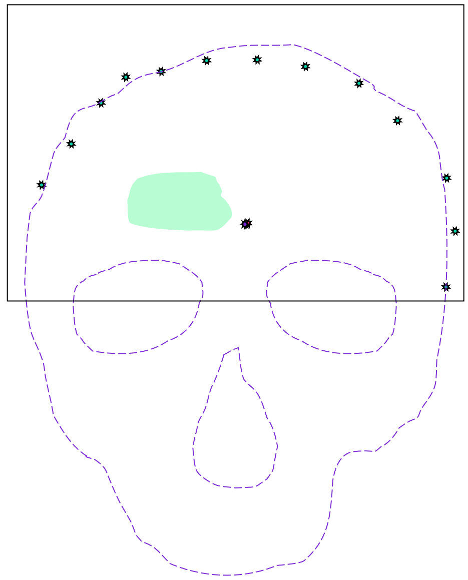
1. Compute centroid $\mathbf{g}_B$ of surface $\mathcal{F}_B$.
2. Determine a point $\mathbf{q}_i$ that lies on the intersection of the line $\mathbf{g}_B - \mathbf{f}_i$ and $\mathcal{F}_B$.
3. Then, $d_S(\mathcal{F}_B, \mathbf{f}_i) = \|\mathbf{q}_i - \mathbf{f}_i\|$

Head-in-hat algorithm: step 0

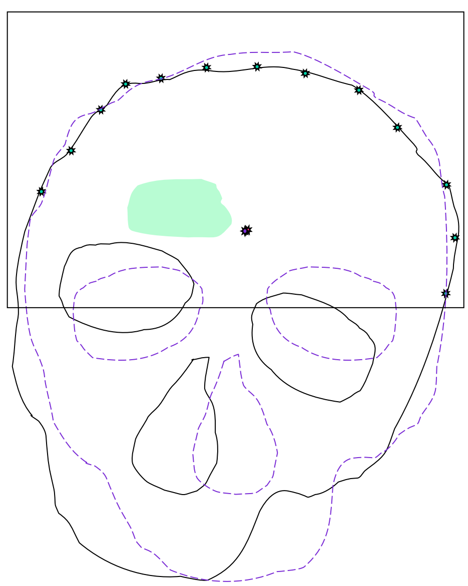


Copyright © 1999-2014 R. H. Taylor
Computer Integrated Surgery 600.445/645

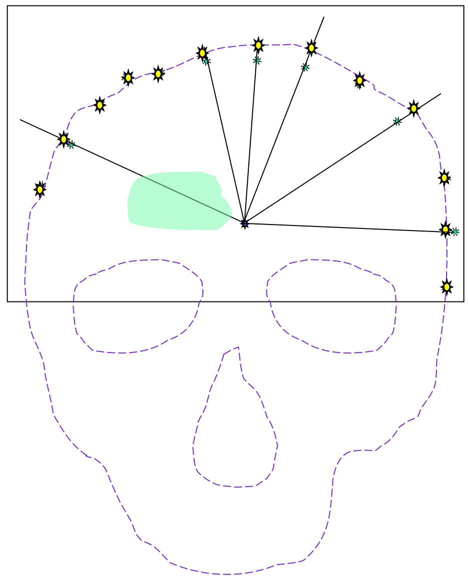
Head-in-hat algorithm: step1



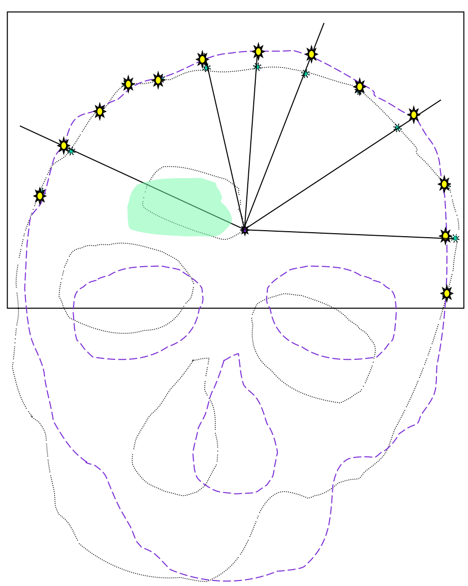
Head-in-hat algorithm: step1



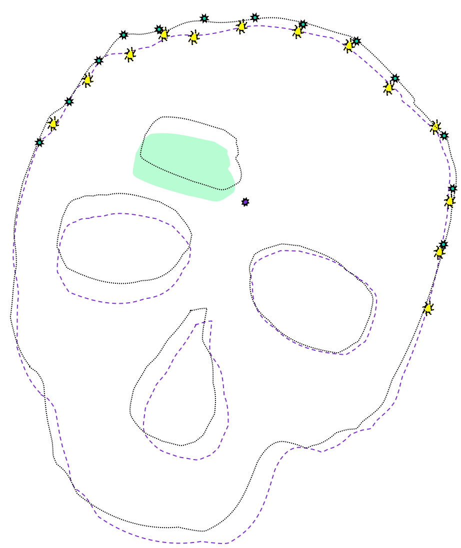
Head-in-hat algorithm: step 2



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

Copyright © 1999-2014 R. H. Taylor

Computer Integrated Surgery 600.445/645

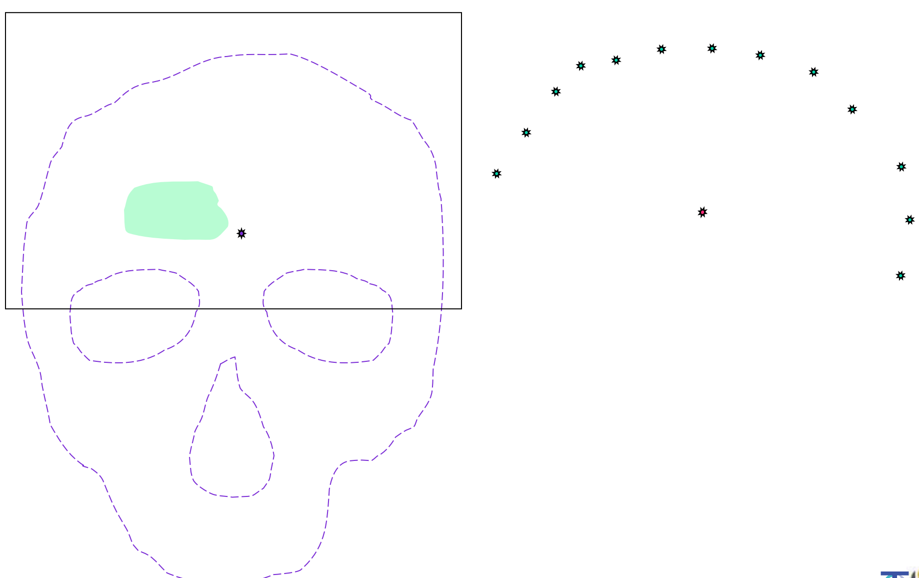
Iterative Closest Point

- 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

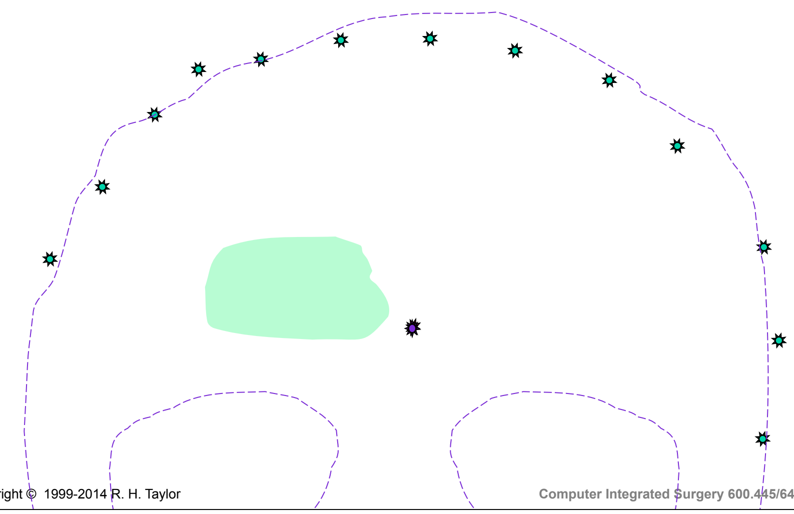
$$D_{k+1} = \sum_i \|\mathbf{v}_i - \mathbf{T}_{k+1} \cdot \mathbf{f}_i)\|^2$$

- Physical Analogy

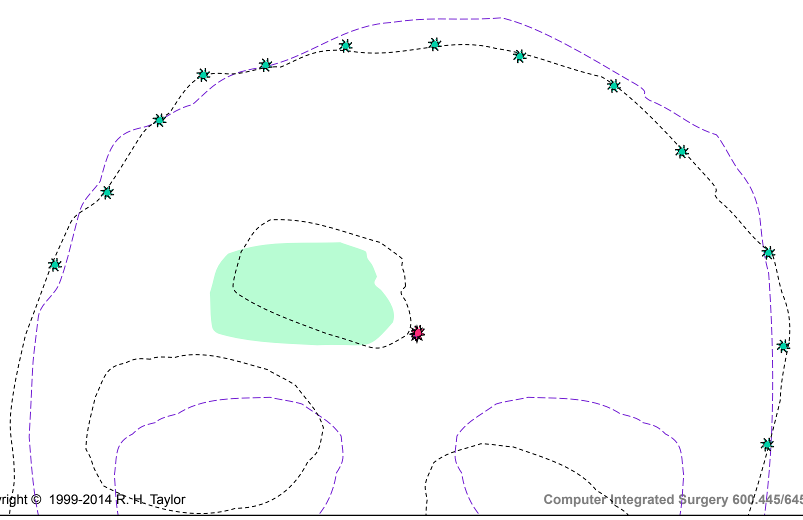
Iterative Closest Point: step 0



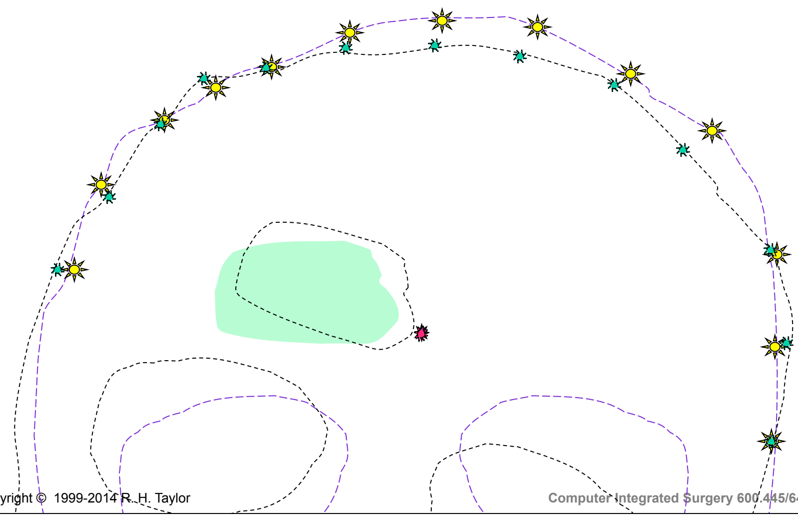
Iterative Closest Point: step1



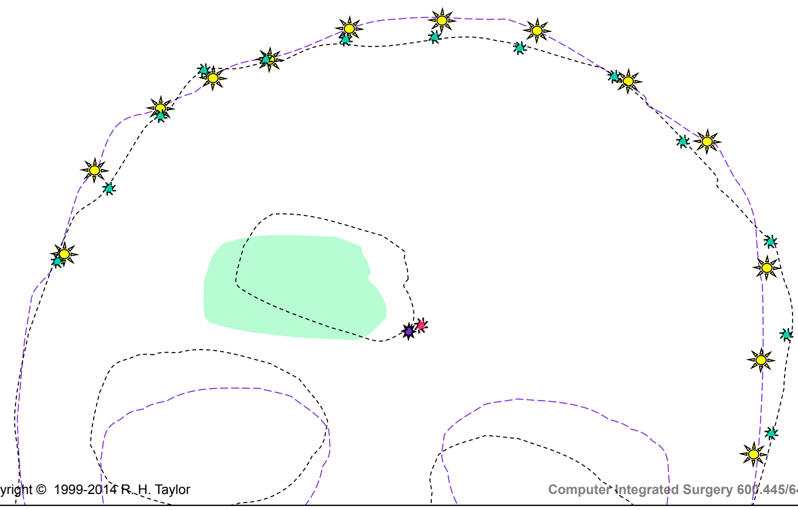
Iterative Closest Point: step1



Iterative Closest Point: step2

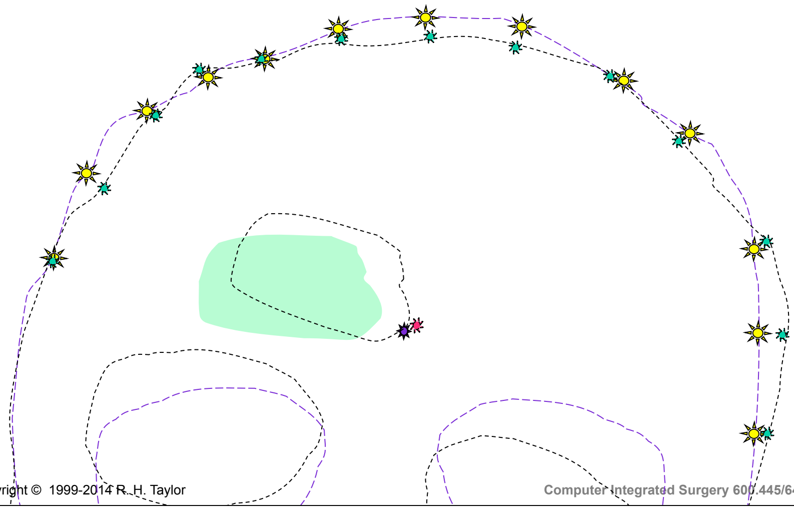


Iterative Closest Point: step 3

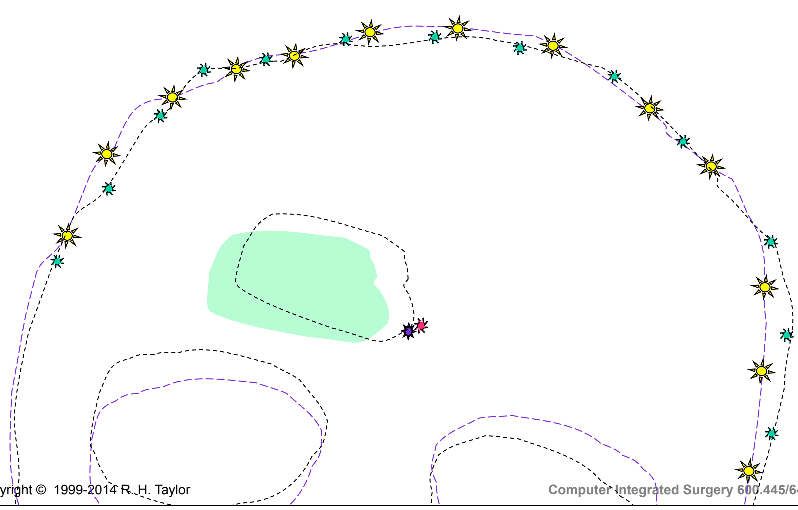


Copyright © 1999-2014 R. H. Taylor
Computer Integrated Surgery 600.445/645

Iterative Closest Point: step 2 interation 2

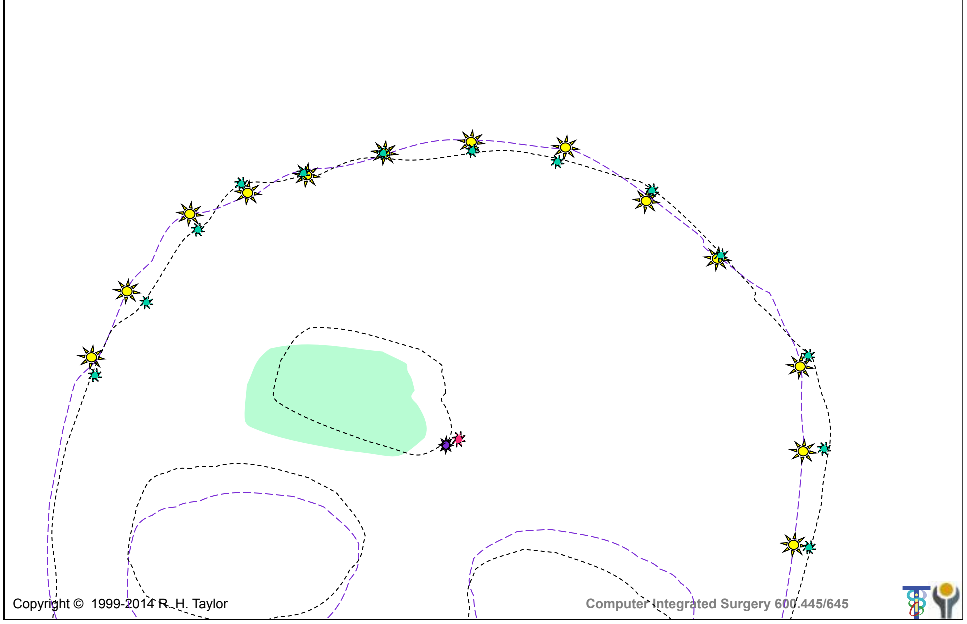


Iterative Closest Point: step 3 interation 2

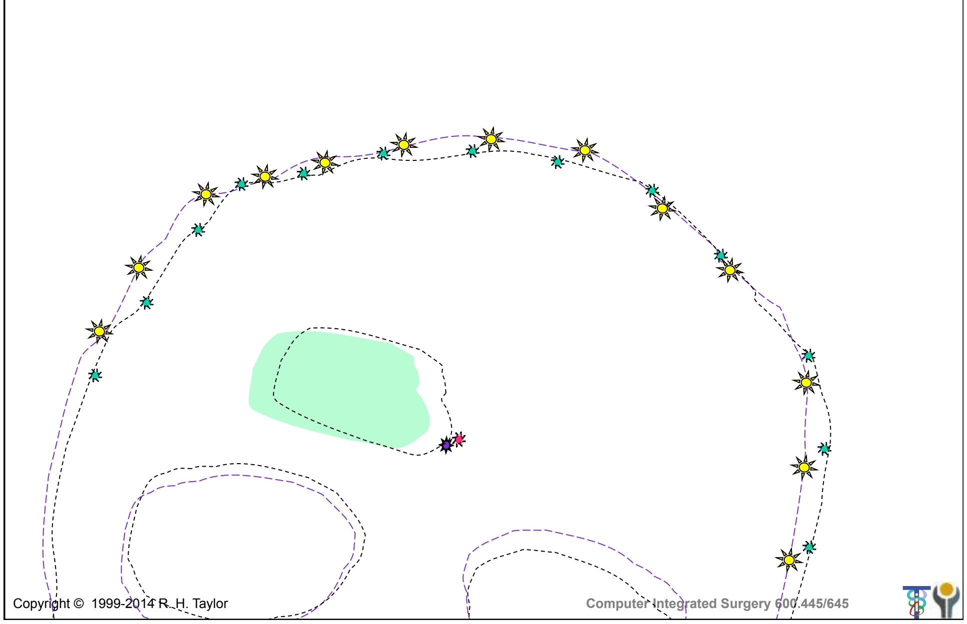


Copyright © 1999-2014 R. H. Taylor
Computer Integrated Surgery 600.445/645

Iterative Closest Point: step 2 interation 3

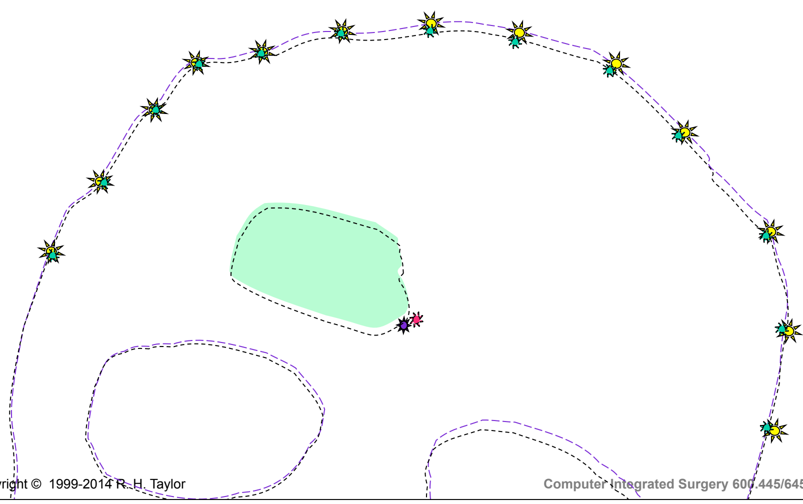


Iterative Closest Point: step 3 interation 3



Copyright © 1999-2014 R. H. Taylor
Computer Integrated Surgery 600.445/645

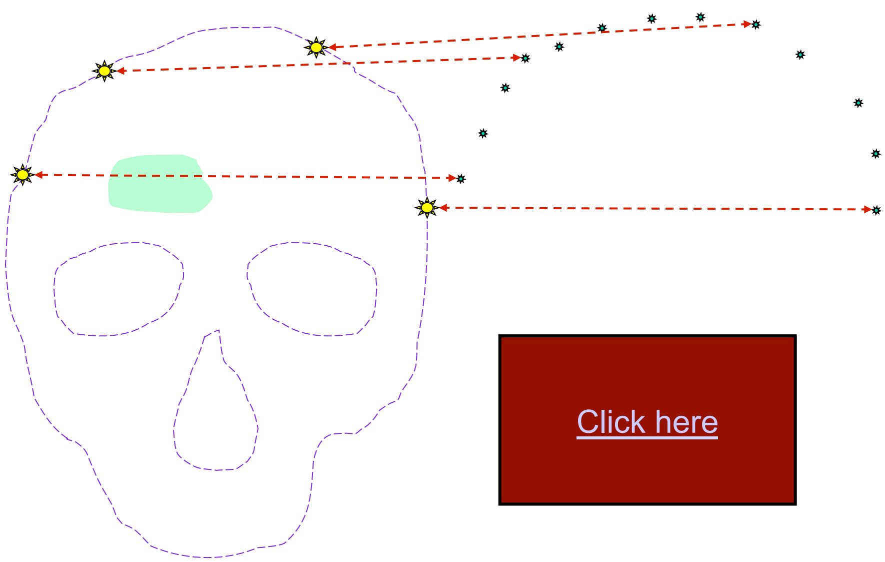
Iterative Closest Point: step 3 interation N



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 $\mathrm{d}_k = \lVert \vec{\mathbf{c}}_k - \mathbf{F}_n \cdot \vec{\mathbf{q}}_k \rVert$
$\mathtt{I} = \{\cdots, i_k, \cdots\}$	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 $\vec{\mathbf{b}}_k - \mathbf{F} \cdot \vec{\mathbf{a}}_k$
$\sigma_n, \ (\varepsilon_{\max})_n, \bar{\varepsilon}_n$	$\dfrac{\sum_k \vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}{NumElts(\mathtt{E})}$; $\max_k \sqrt{\vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}$; $\dfrac{\sum_k \sqrt{\vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}}{NumElts(\mathtt{E})}$

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.

$$n \leftarrow 0$$
$$\mathtt{I} \leftarrow \{\cdots, 1, \cdots\}$$
$$\mathtt{C} \leftarrow \{\cdots, \text{point on } \mathtt{m}_1, \cdots\}$$
$$\mathtt{D} \leftarrow \{\cdots, \lVert \vec{\mathbf{c}}_k - \mathbf{F}_0 \bullet \vec{\mathbf{q}}_k \rVert, \cdots\}$$

Copyright © 1999-2014 R. H. Taylor
Computer Integrated Surgery 600.445/645

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

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

// 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 2: (transformation update)

$n \leftarrow n+1$

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

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

Step 3: (adjustment)

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

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

Step 4: (iteration)

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

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 $3(\bar{\varepsilon})_n$. 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 $\sigma_n, \bar{\varepsilon}_n$ and/or $(\varepsilon_{max})_n$ are less than desired thresholds and $\gamma \le \bar{\varepsilon}_n / \bar{\varepsilon}_{n-1} \le 1$ for some value γ (e.g., $\gamma \cong .95$) for several iterations.

Copyright © 1999-2014 R. H. Taylor
Computer Integrated Surgery 600.445/645

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

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

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)

$n \leftarrow n+1$

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

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

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

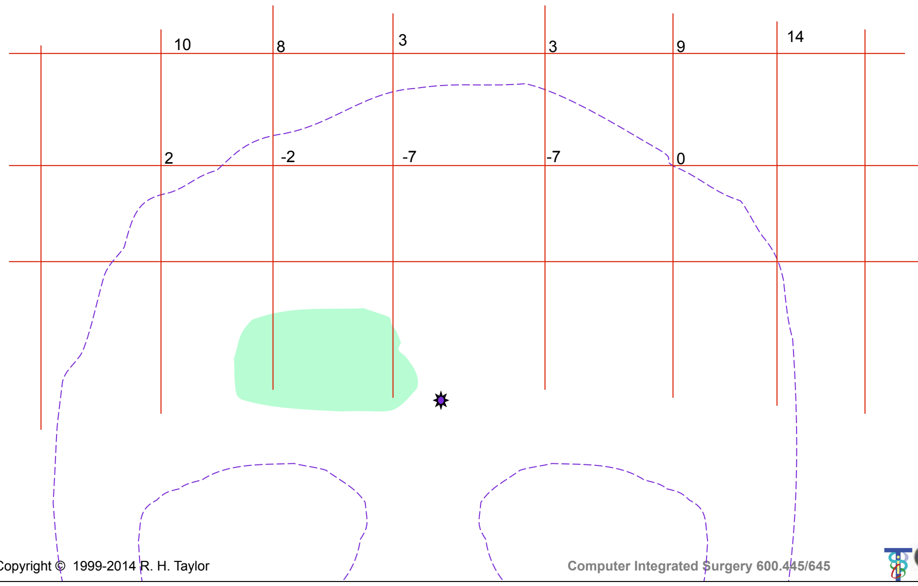
Distance Maps

- Many authors
- Somewhat related to ICP
- 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

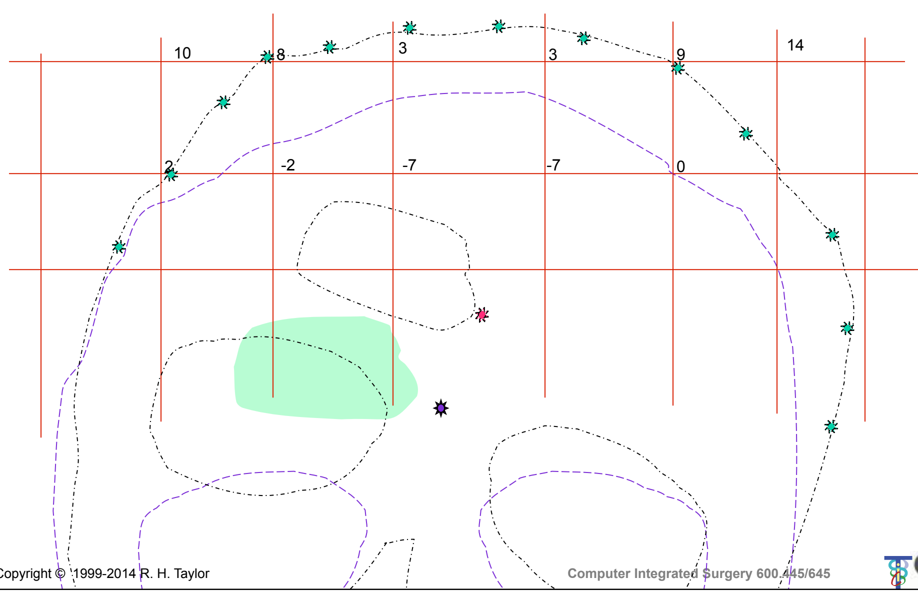
Distance Maps (Continued)

- Approach is to **precompute** $d_S(\mathcal{F}, \mathbf{v}_j)$ for a lattice of points $\mathbf{v}_j$.
- Then, to compute $d_S(\mathcal{F}, \mathbf{f}_i)$:
  1. Determine the set $\mathcal{V}$ of lattice points surrounding $\mathbf{f}_i$.
  2. Look up the distances $\{d_j = d_S(\mathcal{F}, \mathbf{v}_j)\}$ for $\mathbf{v}_j \in \mathcal{V}$.
  3. Estimate d_S from the d_j, e.g., by trilinear interpolation
- Various techniques to do the optimization

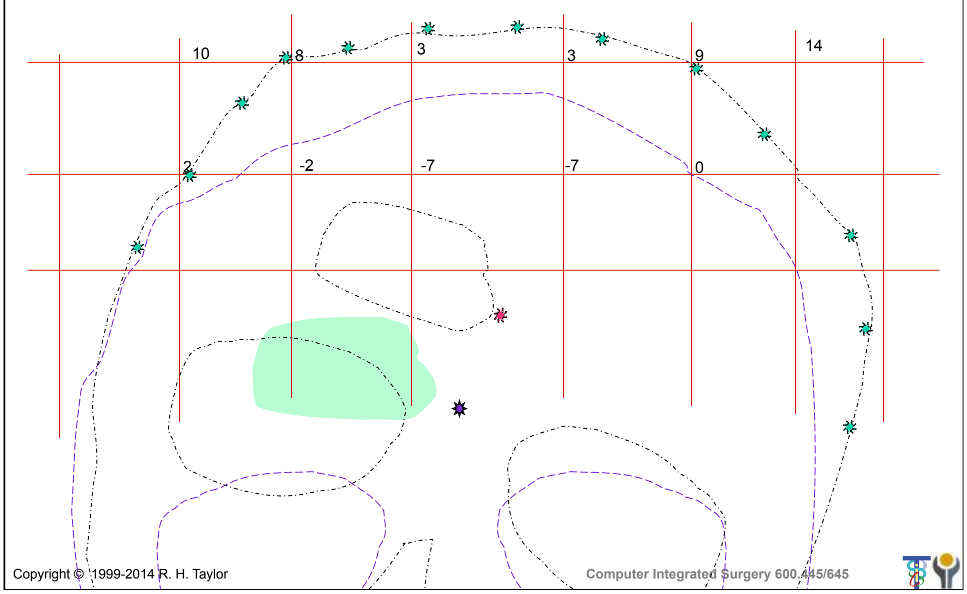
Distance Maps: step 0



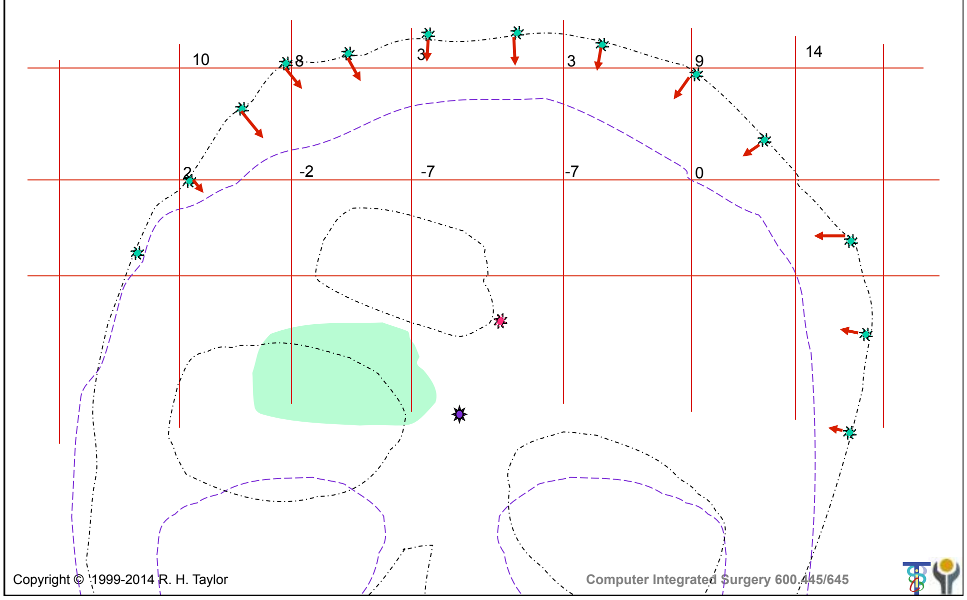
Distance Maps: step 1



Distance Maps: step 1

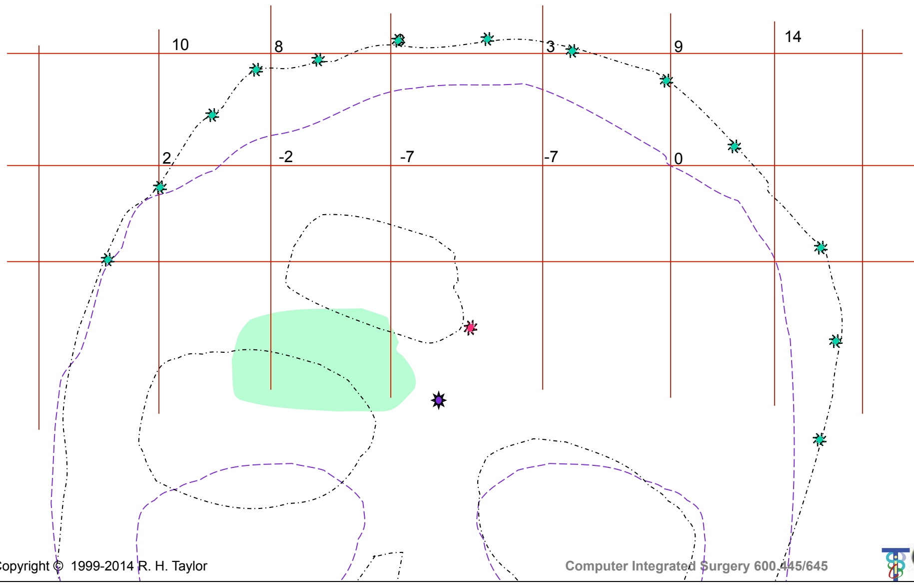


Distance Maps: step 2



Copyright © 1999-2014 R. H. Taylor
Computer Integrated Surgery 600.445/645

Distance Maps: step 3



Distance Maps: Iteration Step

1. Determine cell $\mathcal{V}_i$ for each $\mathbf{p}_i = \mathbf{T} \cdot \mathbf{f}_i$. Let $\overline{\lambda}_i$ be the corresponding interpolation parameters for $\mathbf{p}_i$ within cell.
2. Determine small motion $\Delta\mathbf{T}$ that minimizes

$$\sum_i [(\Delta\mathbf{T}\mathbf{p}_i - \mathbf{p}_i) \cdot \nabla d_S(\overline{\lambda_i}, \mathcal{V}_i)]$$

or

$$\sum_i -[(\Delta\mathbf{T}\mathbf{p}_i - \mathbf{p}_i) \cdot \nabla d_S(\overline{\lambda_i}, \mathcal{V}_i)]$$

3. Update $\mathbf{T} \leftarrow \Delta\mathbf{T} \bullet \mathbf{T}$

Update: Distance Maps

The use of a fairly coarse grid of distances, described in previous slides, is no longer really needed. 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

A contour-based 2D-3D method …

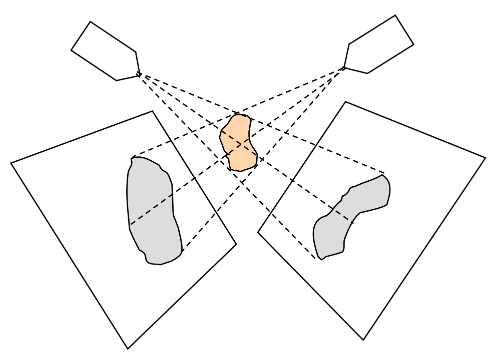
Gueziec *et al., 1998*

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**



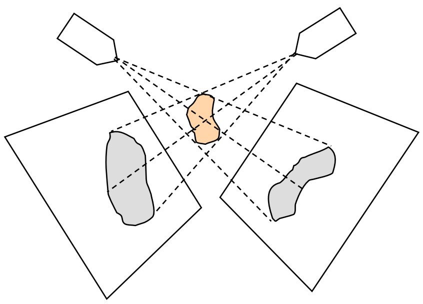
A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

Computer Integrated Surgery 600.445/645

A countour-based 2D-3D method …

Gueziec *et al., 1998*

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



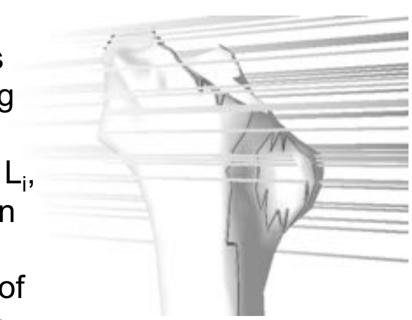
A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

A countour-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.

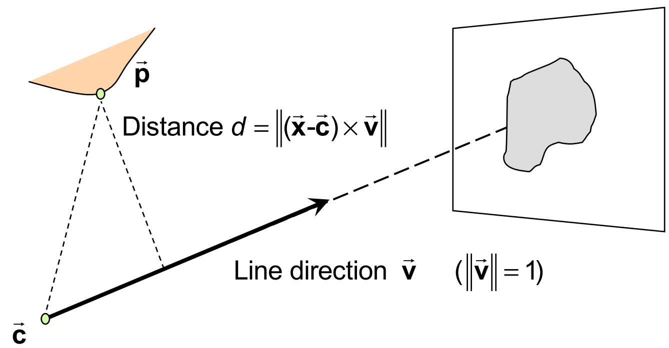
Step 2: For each x-ray path line line L_i, identify the closest point p_i on an apparent projection contour. This will give a set of points on the body surface to be moved toward the corresponding x-ray lines



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 countour-based 2D-3D method …

Gueziec *et al., 1998*

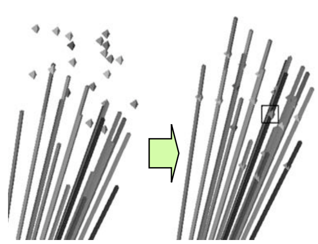


Note: It is convenient to use the x-ray source position (i.e., the center of convergence for a bundle of x-ray projection lines) as the value for $\vec{\mathbf{c}}$.

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 countour-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 $\vec{\mathbf{p}}_i \leftarrow \mathbf{R}\vec{\mathbf{p}}_i + \vec{\mathbf{t}}$

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

Step 3: Solve the least squares problem:

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

where ρ is sufficiently small so that $\mathbf{I}+2\mathbf{U}$ approximates a rotation

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

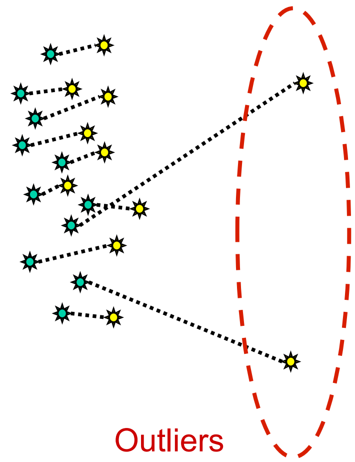
Update $\mathbf{p}_i \leftarrow \Delta\mathbf{R}\mathbf{p}_i + \Delta\vec{\mathbf{t}}$; $\mathbf{R} \leftarrow \Delta\mathbf{R}\mathbf{R}$; $\vec{\mathbf{t}} \leftarrow \Delta\mathbf{R}\vec{\mathbf{t}} + \Delta\vec{\mathbf{t}}$

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 …

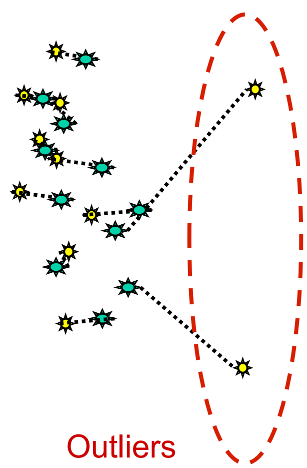
- Basic idea is to identify outliers and give them little or no weight.



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 …

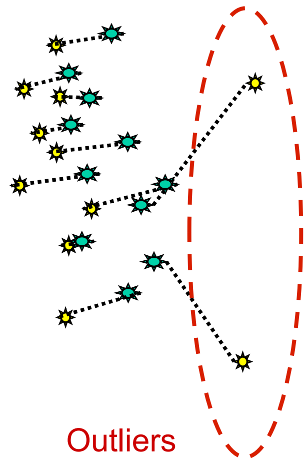
- Basic idea is to identify outliers and give them little or no weight.



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 …

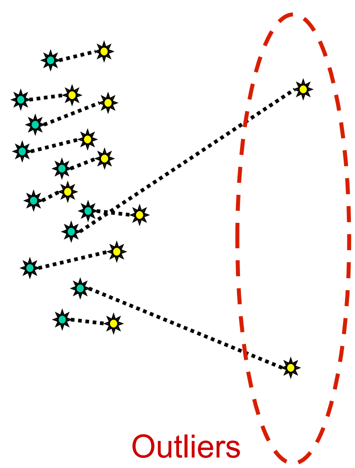
- Basic idea is to identify outliers and give them little or no weight.



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.

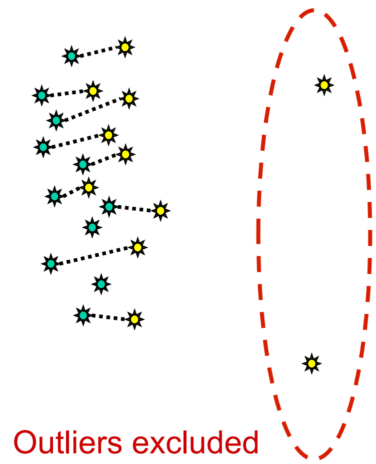


R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," Comput. Vision, Graphics, Image Processing-IU, vol. 60, no. 3, pp. 313–342, 1994.

Copyright © 1999-2014 R. H. Taylor
Computer Integrated Surgery 600.445/645

Robust Pose Estimation …

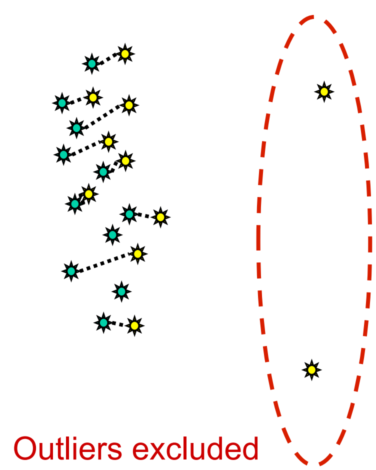
- Basic idea is to identify outliers and give them little or no weight.



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.

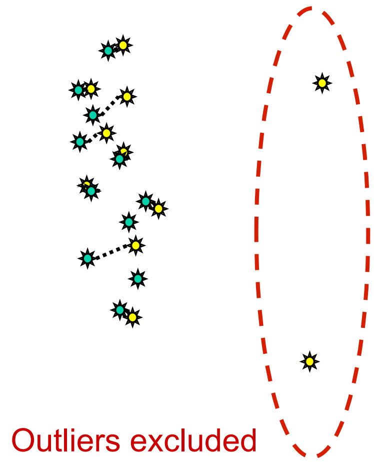


R. Kumar and A. R. Hanson, “Robust methods for estimating pose and a sensitivity analysis,” Comput. Vision, Graphics, Image Processing-IU, vol. 60, no. 3, pp. 313–342, 1994.

Copyright © 1999-2014 R. H. Taylor
Computer Integrated Surgery 600.445/645

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

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

(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 $\vec{\mathbf{u}} = \vec{\mathbf{0}}$, $\Delta\mathbf{t} = \vec{\mathbf{0}}$

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)

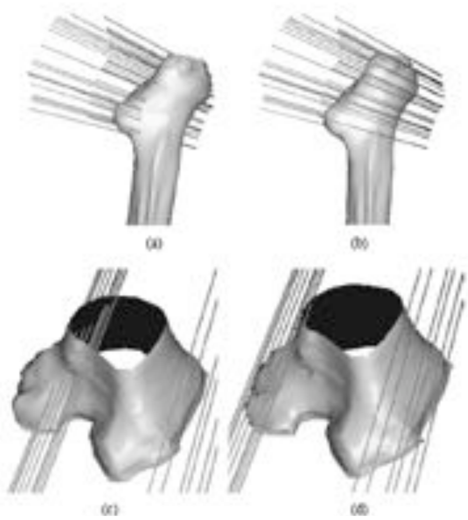
Step 3.3: Iterate steps 3.1 and 3.2 until a suitable termination condition is reached.

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



After

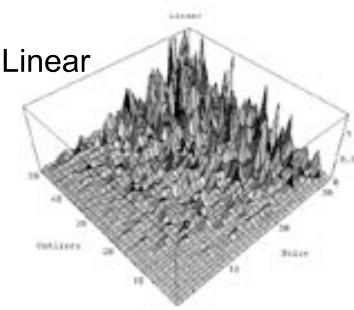
A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

Copyright © 1999-2014 R. H. Taylor
Computer Integrated Surgery 600.445/645

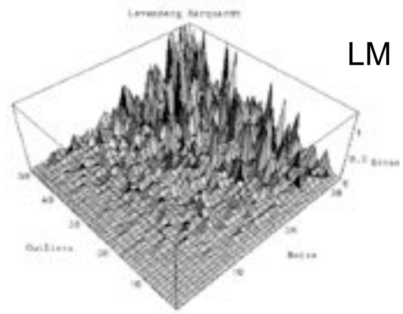
A countour-based 2D-3D method … results

Gueziec *et al., 1998*

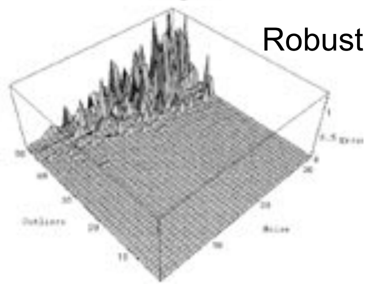
Linear



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 EXECUTION 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 (cont'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

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

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.,

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

Then an appropriate distance metric is the Mahalabonis distance

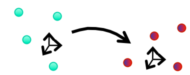
$$\text{D}(\vec{\mathbf{a}}_k, \vec{\mathbf{b}}_k) = (\vec{\mathbf{a}}_k - \vec{\mathbf{b}}_k)^T \mathbf{C}_k^{-1} (\vec{\mathbf{a}}_k - \vec{\mathbf{b}}_k) = \vec{\mathbf{d}}_k^T \Lambda_k^{-1} \vec{\mathbf{d}}_k$$

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.

Iterative Closest Point (ICP) Revisited

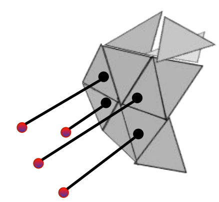


- Widely popular and useful method for point cloud to surface registration introduced by Besl & McKay in 1992
- Many variants proposed since its inception affecting all aspects of the algorithm (robustness, matching criteria, match alignment, etc.)

➢ **Matching Phase**:

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

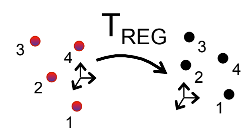
$$\boldsymbol{y_i} = \mathrm{C_{CP}}(T(\boldsymbol{x_i}), \Psi) = \underset{\boldsymbol{y} \in \Psi}{\operatorname{argmin}} \|\boldsymbol{y} - T(\boldsymbol{x_i})\|_2$$



➢ **Registration Phase:**

compute transformation to minimize sum of square distances between matches

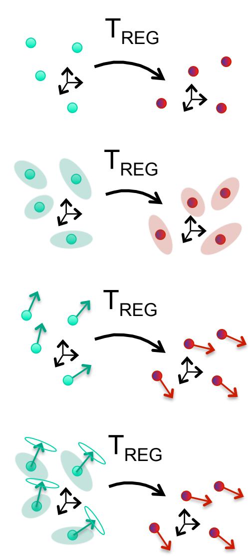
$$T = \underset{T}{\operatorname{argmin}} \sum_{i=1}^{n} \|\boldsymbol{y_i} - T(\boldsymbol{x_i})\|_2^2$$



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

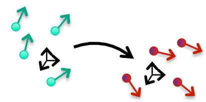
Paired Point Algorithms Outline

- Iterative Closest Point (ICP)
  - the standard algorithm
  - position-only method
  - isotropic (Gaussian) noise model
- Iterative Most Likely Point (IMLP)
  - position-only method
  - generalized Gaussian noise model
- Iterative Most Likely Oriented Point (IMLOP)
  - position & orientation method
  - isotropic orientation (Fisher) and position (Gaussian) noise model
- Generalized IMLOP
  - extension of IMLOP
  - generalize orientation (Kent) and position (generalized Gaussian) noise model



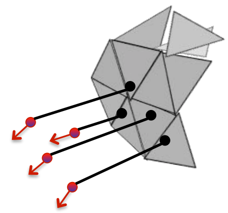
S. Billings and R. H. Taylor, "Iterative Most Likely Oriented Point Registration", *in Medical Image Computing and Computer-Assisted Interventions (MICCAI)*, Boston, October, 2014. (accepted).

Iterative Most Likely Oriented Point (IMLOP)



- **Matching Phase**:

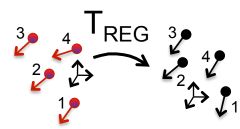
  for each oriented point in the source shape, find the most likely oriented point on the target shape



$$\boldsymbol{y_i} = \mathrm{C_{MLP}}(T(\boldsymbol{x_i}), \Psi) = \operatorname*{argmax}_{\boldsymbol{y}\in\Psi} f_{\mathrm{match}}(T(\boldsymbol{x_i}), \boldsymbol{y})$$

- **Registration Phase:**

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



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

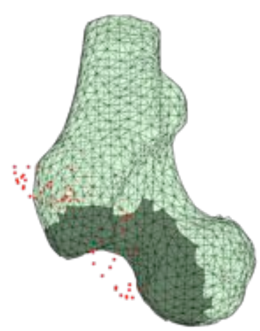
S. Billings and R. H. Taylor, "Iterative Most Likely Oriented Point Registration", *in Medical Image Computing and Computer-Assisted Interventions (MICCAI)*, Boston, October, 2014. (accepted).

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

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

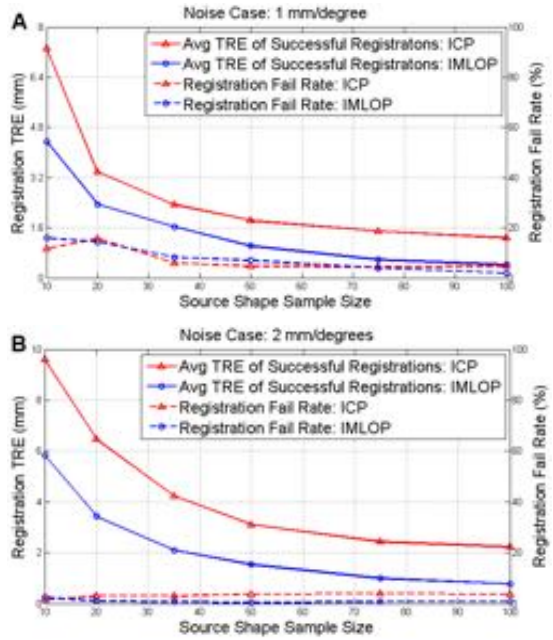


Example source point cloud sampled from dark region of target mesh.

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).

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

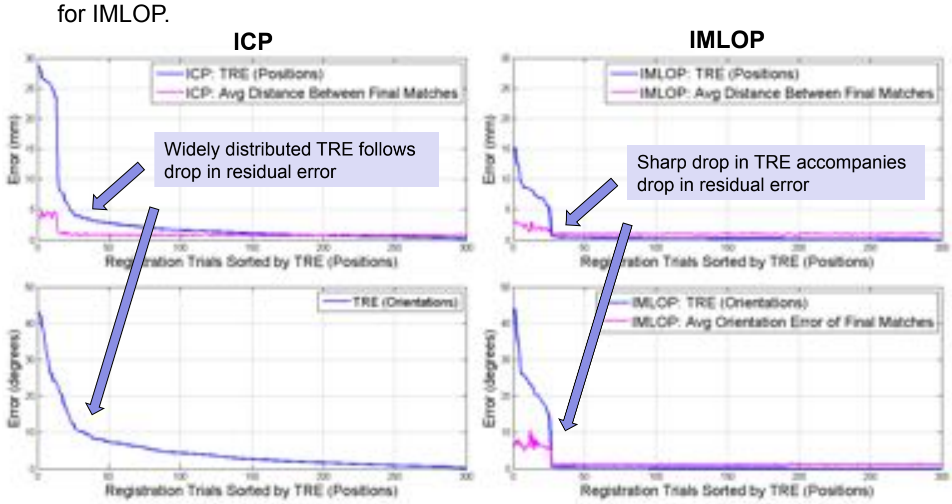
Registration failure threshold set to twice the noise level for both position and orientation.



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

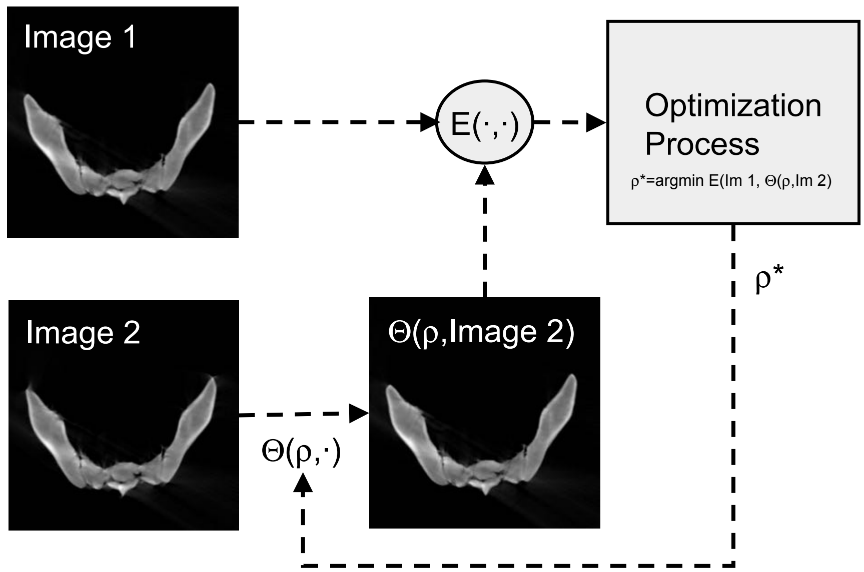
Experiments

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.



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).**

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

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

Normalized cross correlation

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

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)

$$Similarity(A,B) = H(A) + H(B) - H(A,B)$$

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

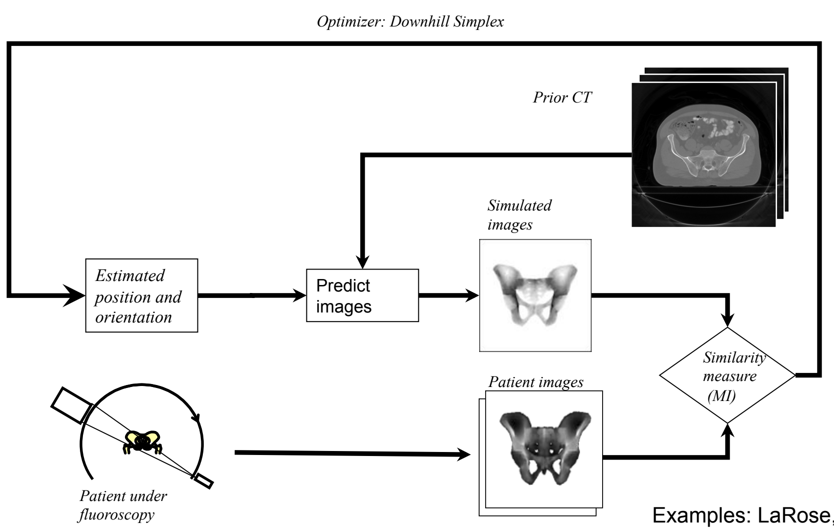
$$Similarity(A,B) = \frac{H(A)+H(B)}{H(A,B)}$$

Objective function

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

Rigid 3D/2D Registration

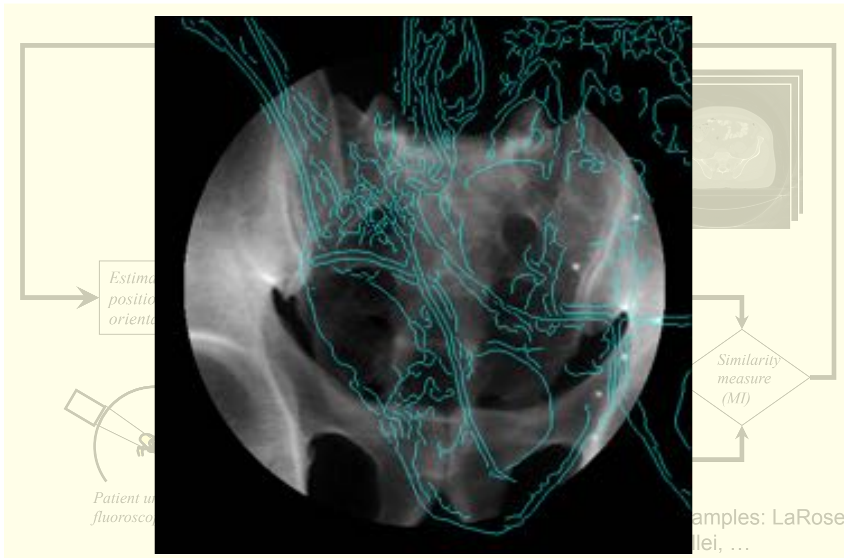
Ofri Sadowsky



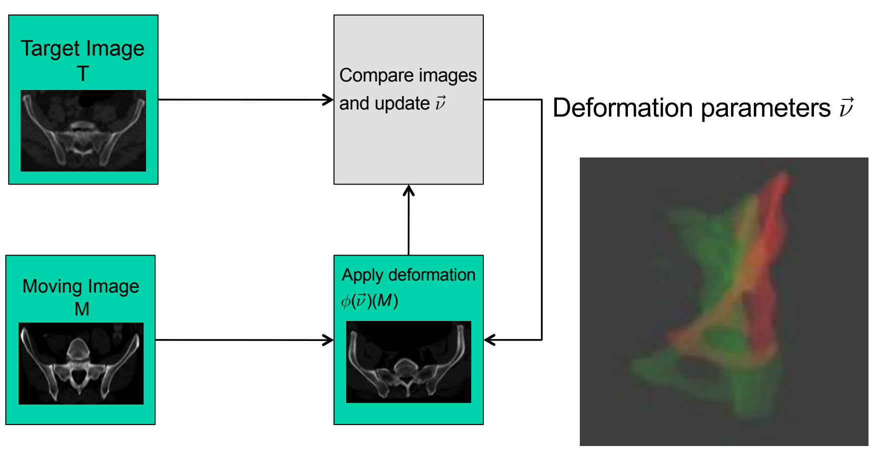
Examples: LaRose, Zollei, …

Rigid 3D/2D Registration

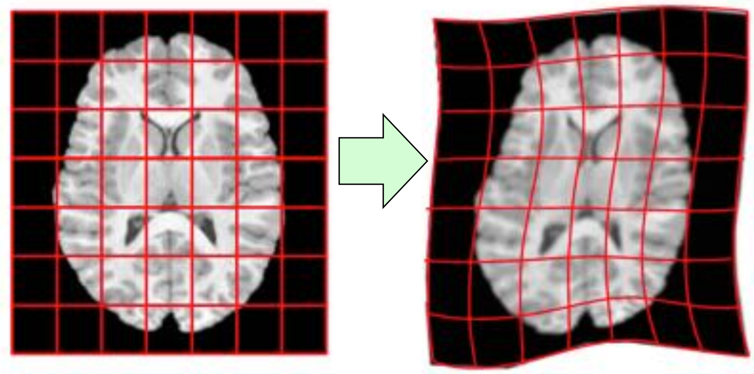
Ofri Sadowsky



Deformable Registration



Deformable Registration



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

- 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 Φ is bijective an both Φ and Φ^{-1} are smooth

Images: Tom Fletcher

Radial Basis Functions

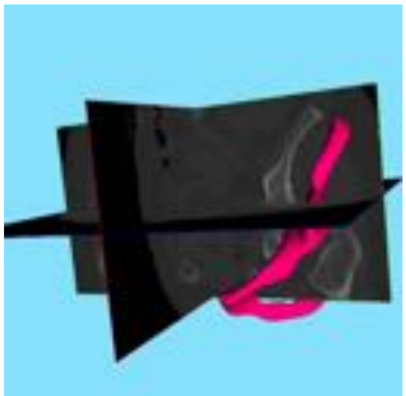
Given a scalar function $\phi(\bullet)$ and a set of sample points $\vec{\mathrm{p}}_k$ with associated deformations $\vec{\mathbf{d}}_k$, one can represent the deformation Φ at a point $\vec{\mathbf{x}}$ by

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

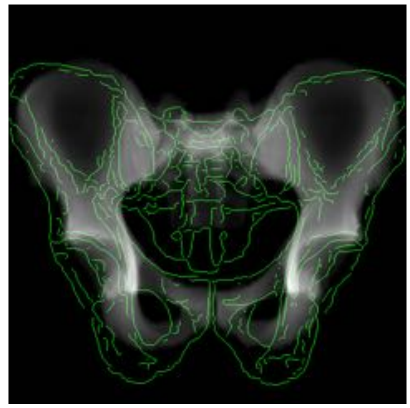
- Many possible functions to use for ϕ
  - Common choices include Gaussians and "thin plate splines", which have non-compact support (i.e., $\Phi(y)>0$ for arbitrarily large y)
  - Others have compact support (i.e., $\Phi(y)=0$ for $|y|>$ some value)*

* 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

Deformable Registration to Statistical "Atlases"



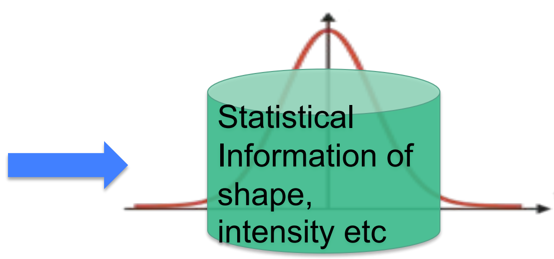
Deformable 3D/3D
Jianhua Yao



Deformable 2D/3D
Ofri Sadowsky

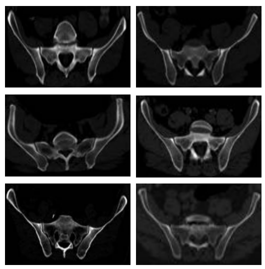
What is a "Statistical Atlas" ?

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

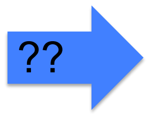


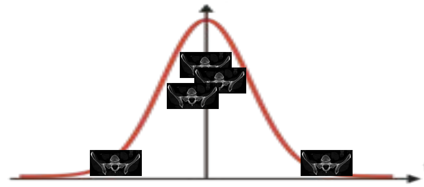
Credit: G. Chintalapani 2010

Statistical Atlases

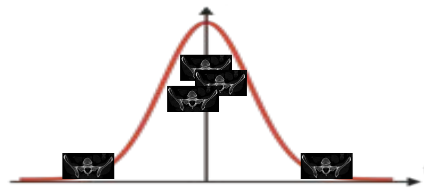


CT scans from a population





Shape distribution



Intensity distribution

Slide Credit: G. Chintalapani 2010

Deformable Altas-based Registration

- Much of the material that follows is derived from the Ph.D. thesis work of J. Yao and Ofri Sadowsky:
  - 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

Digression on “active appearance models”

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

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

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

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

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 rows of **U**. Treating $\vec{\mathbf{D}}^{(j)}$ as a column vector, we can write this as

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

If we define

$\mathbf{M} = \left[\mathbf{U}^{(1)} \quad \cdots \quad \mathbf{U}^{(N)} \right]$ (i.e., the first N columns of **U**)

we get the expression

$\vec{\mathbf{D}}^{(j)} = \mathbf{M}\vec{\lambda}$ where $\vec{\lambda}$ is the j^{th} 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

$$
\begin{aligned}
\mathbf{M}\,diag(\vec{\sigma})\mathbf{V}^T &= \mathbf{D} \\
\mathbf{M}\,diag(\vec{\sigma}) &= \mathbf{DV} \\
\mathbf{M} &= \mathbf{DV}\,diag(\vec{\sigma})^{-1} \\
&= \mathbf{DV}\begin{bmatrix} 1/\sigma_1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & & & \vdots \\ \vdots & & 1/\sigma_k & & \vdots \\ \vdots & & & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1/\sigma_N \end{bmatrix}
\end{aligned}
$$

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

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

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

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

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

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

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

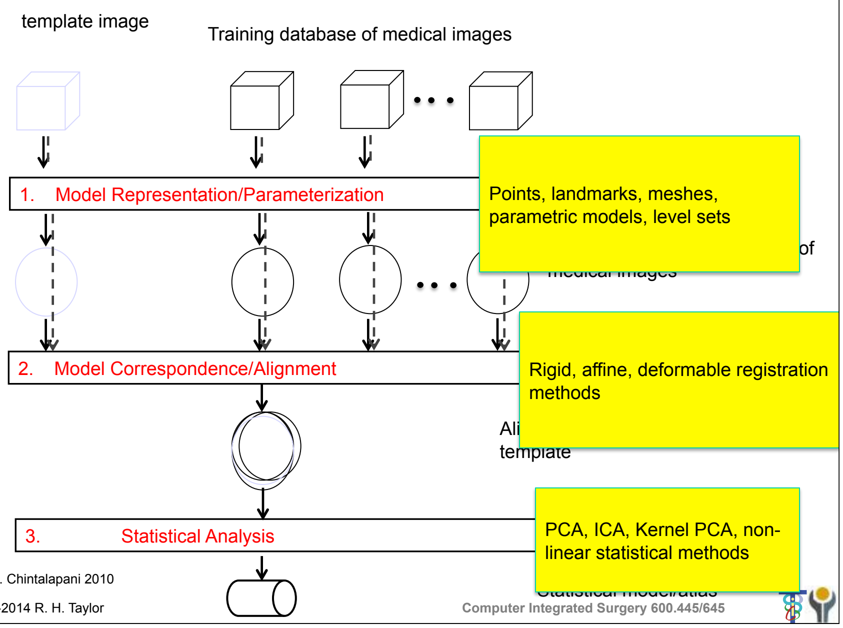
Thus, one representation of a statistical "atlas" of models consists of

- 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]

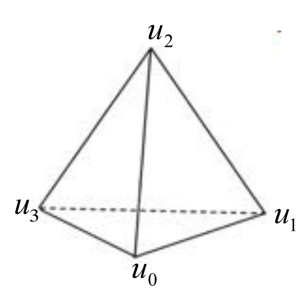
$$P^d(\mathbf{u}) = \sum_{|\mathbf{k}|=d} C_{\mathbf{k}} B^d_{\mathbf{k}}(\mathbf{u})$$

where

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

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

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

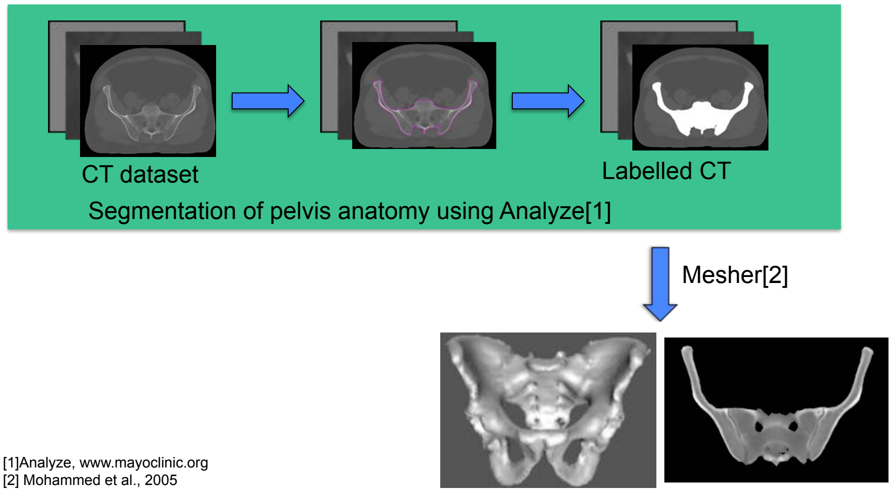


- Alternative might be to use voxels directly after deformation to mean shape

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

Credit: G. Chintalapani 2010

Model Creation



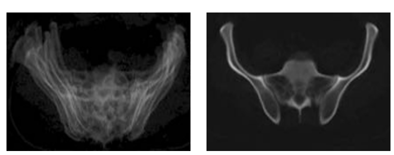
[1]Analyze, www.mayoclinic.org
[2] Mohammed et al., 2005

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

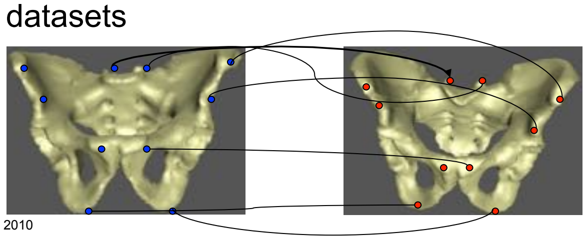
Slide Credit: G. Chintalapani 2010

Model Correspondence

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



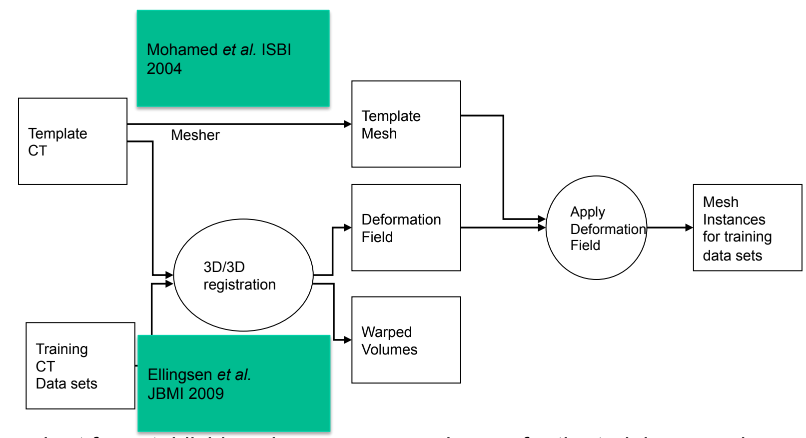
- Need to establish point correspondence between the training datasets



Slide Credit: G. Chintalapani 2010

Model Shape Correspondences

- **Automatic deformable registration based shape correspondences**



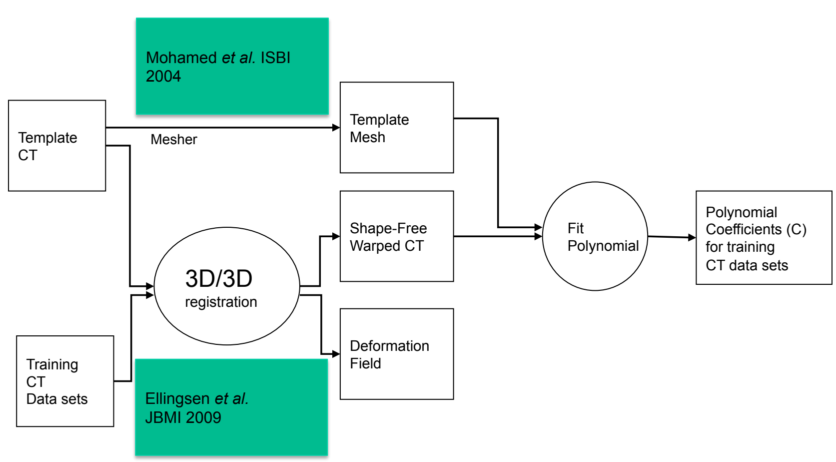
Flowchart for establishing shape correspondences for the training sample

Slide Credit: G. Chintalapani 2010

[1] Rueckert et al., MICCAI 03

Model Intensity Correspondences

- Automatic deformable registration based correspondences

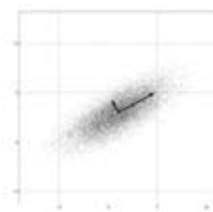


Flowchart for establishing intensity correspondences for the training sample

Principal Component Analysis

- Given the mesh instances of training sample,

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



- Compute mean and subtract the mean from the sample

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

- Compute

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

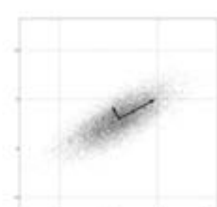
With principal components in U and eigen values $\lambda = \frac{1}{N-1} DD^T$

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

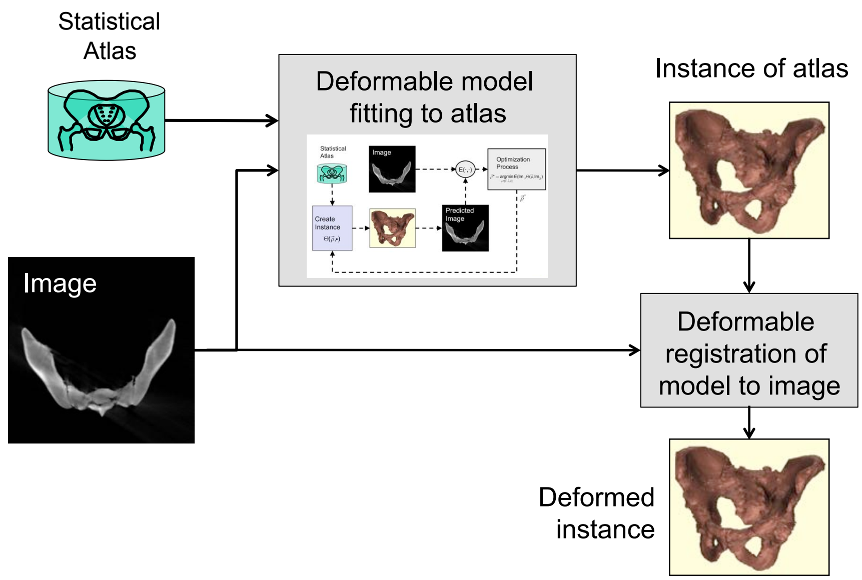
Deformable Registration Between Shape/Density Atlas and Patient CT

- 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

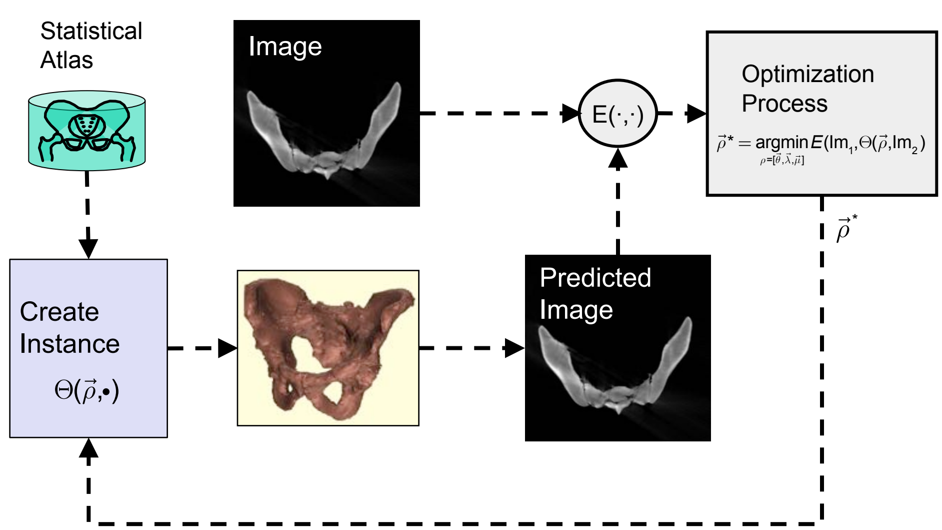
Jianhua Yao

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)
- Global Deformation
  - Statistical deformation mode (M_i)
- Local Deformation
  - Adjustment of every vertex

Jianhua Yao

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)
  - Levenberg-Marquardt (requires computing gradients)

Jianhua Yao

Local Deformation

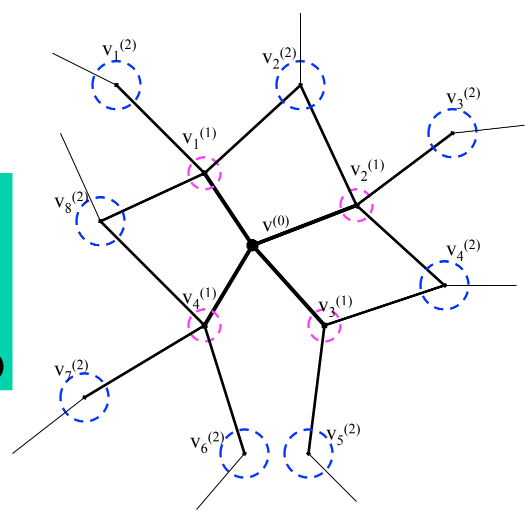
- Motivation: Statistical deformation can't capture all the variability due to the limited number of models in the training set
- Locally adjust the location of vertices to match the boundary of the bone and the interior density property
- Use multiple-layer flexible mesh template matching to find the 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

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



Jianhua Yao

Template matching

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

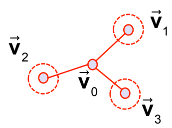
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

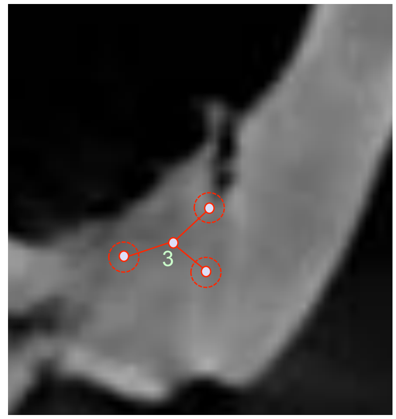
For each neighbor $\vec{\mathbf{v}}_k$

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 $\vec{\mathbf{x}}_0$:

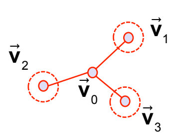
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

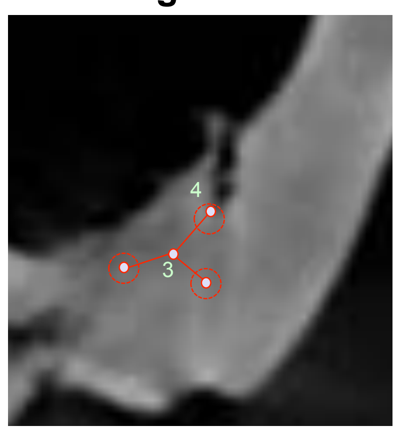
For each neighbor $\vec{\mathbf{v}}_k$

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 $\vec{\mathbf{x}}_0$:

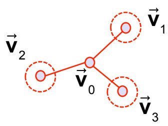
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

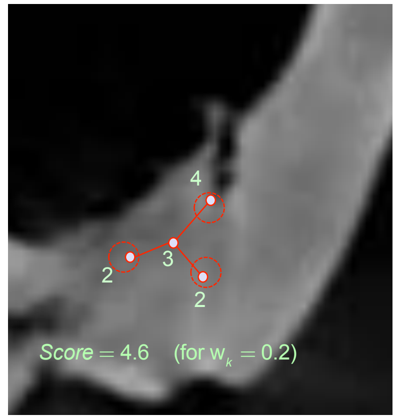
For each neighbor $\vec{\mathbf{v}}_k$

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 $\vec{\mathbf{x}}_0$:

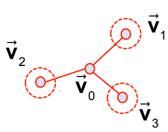
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

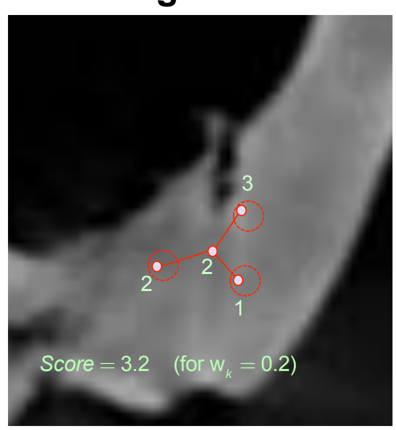
For each neighbor $\vec{\mathbf{v}}_k$

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

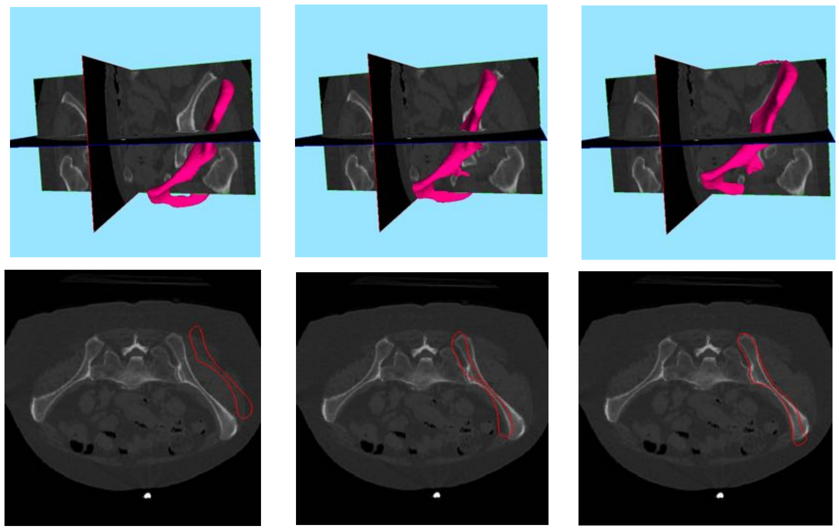
Score $(\vec{\mathbf{x}}_0) = E(\vec{\mathbf{x}}_0, \vec{\mathbf{v}}_0) + \sum_k w_k E(\vec{\mathbf{x}}_k, \vec{\mathbf{v}}_k)$

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





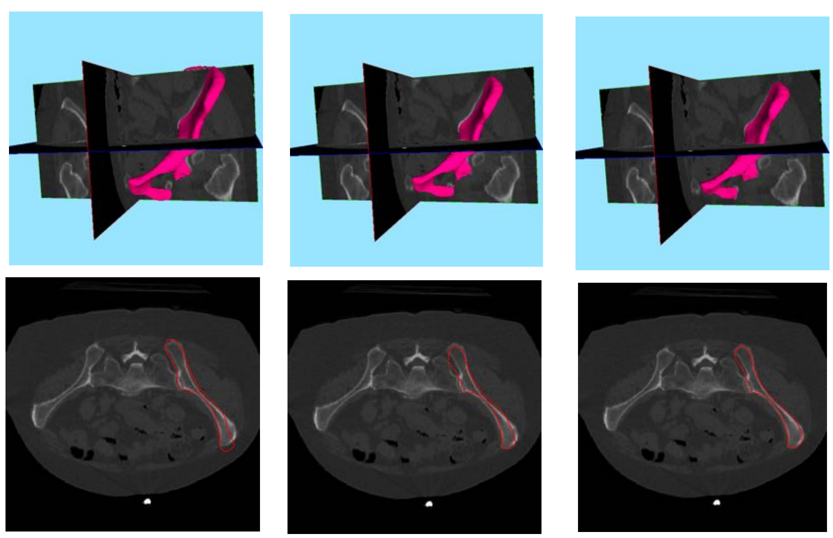
Results (Affine Transformation)



Initial Intermediate Final

Jianhua Yao

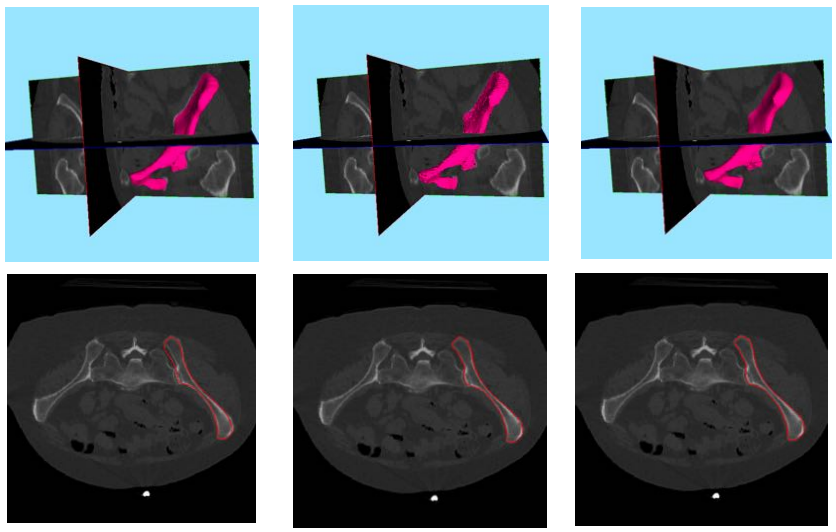
Results (Global Deformation)



Initial Intermediate Final

Jianhua Yao

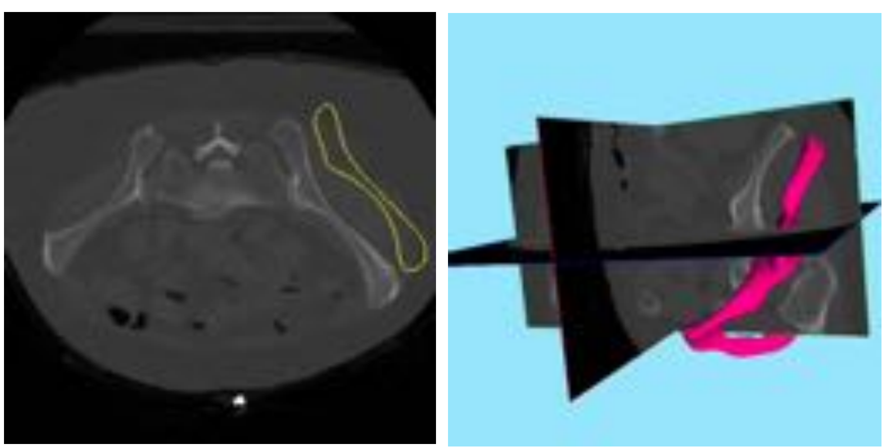
Results (Local Deformation)



Initial Intermediate Final

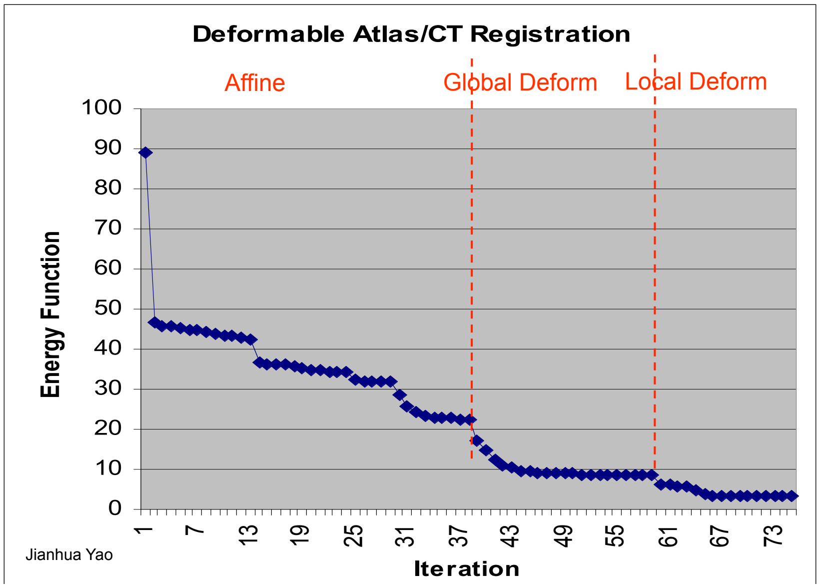
Jianhua Yao

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

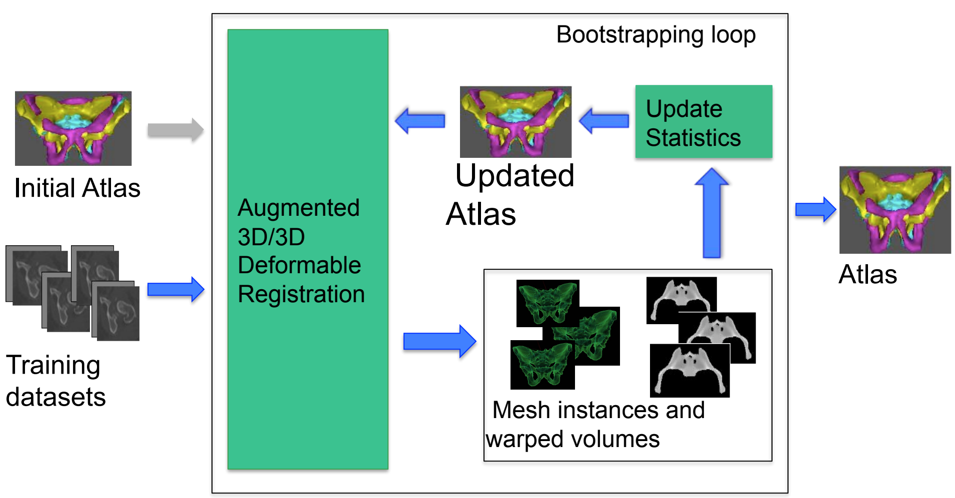


Jianhua Yao

Results (Deformable Registration)



Iterative "bootstrapping" of Atlas



Chintalapani *et al.* MICCAI 2007

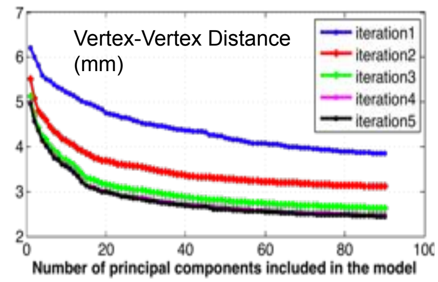
Slide credit: Gouthami Chintalapani

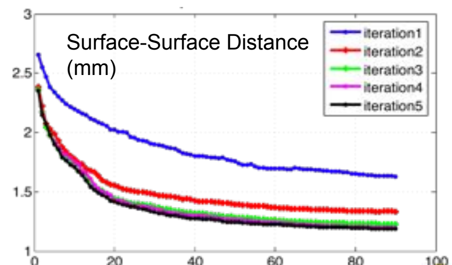
Leave-Out Validation Experiments

- # 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

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

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





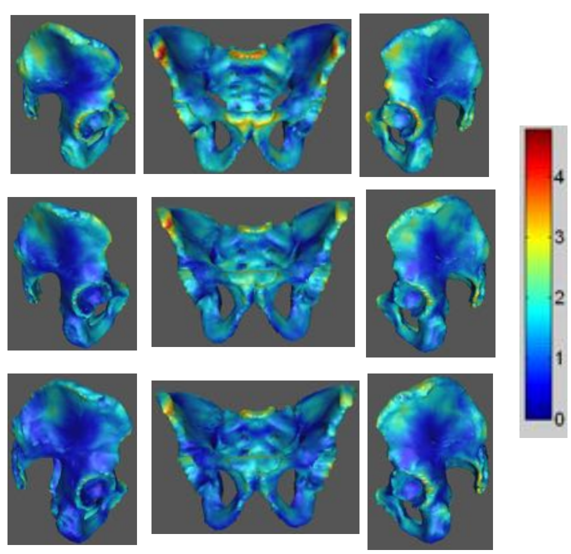
Slide credit: Gouthami Chintalapani

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

Slide credit: Gouthami Chintalapani

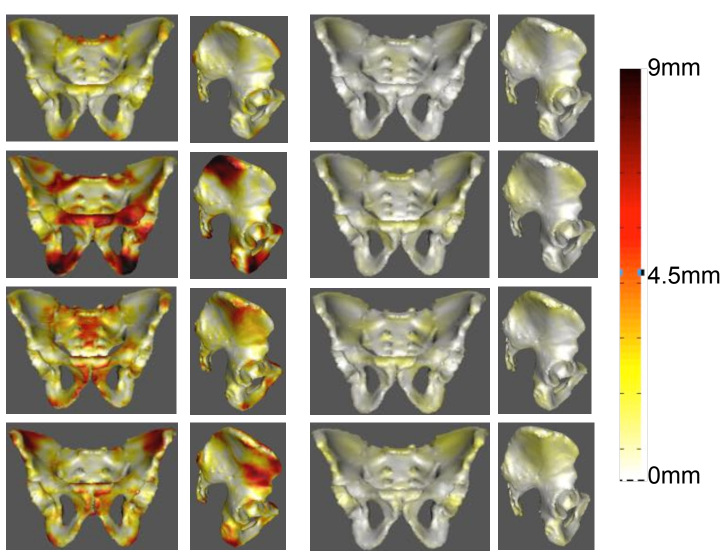
Average Difference between Atlases 1,2 and 3

Mean

Mean + Mode 1

Mean + Mode 2

Mean + Mode 3



Before iteration

After iteration

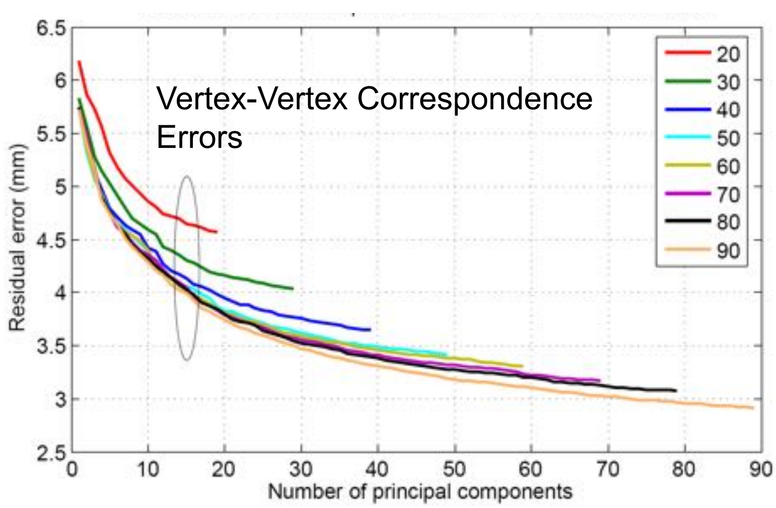
Slide credit: Gouthami Chintalapani

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

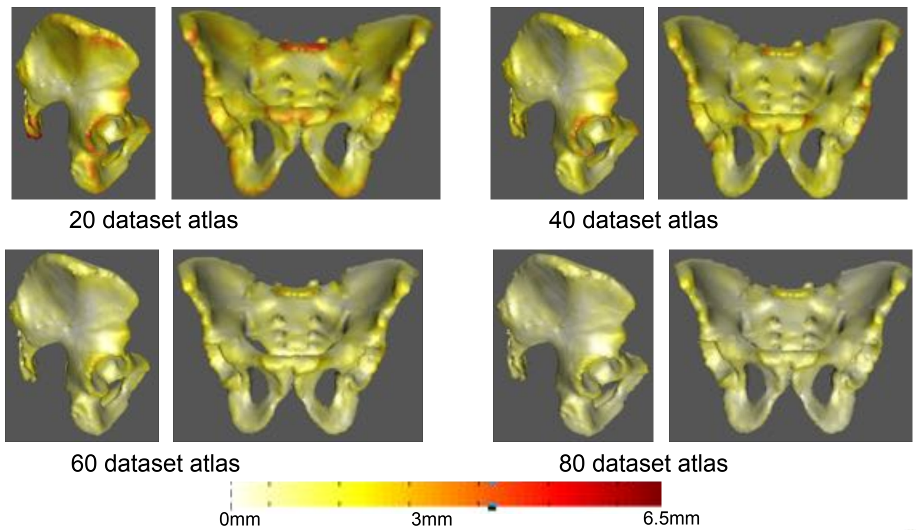
 Slide credit: Gouthami Chintalapani

Training Sample Size



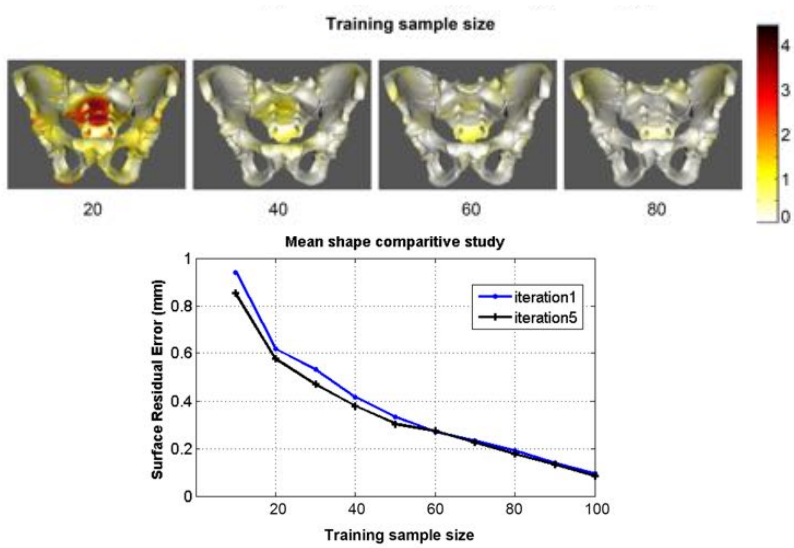
 Slide credit: Gouthami Chintalapani

Surface residual error using 18 modes for different sample set sizes



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.

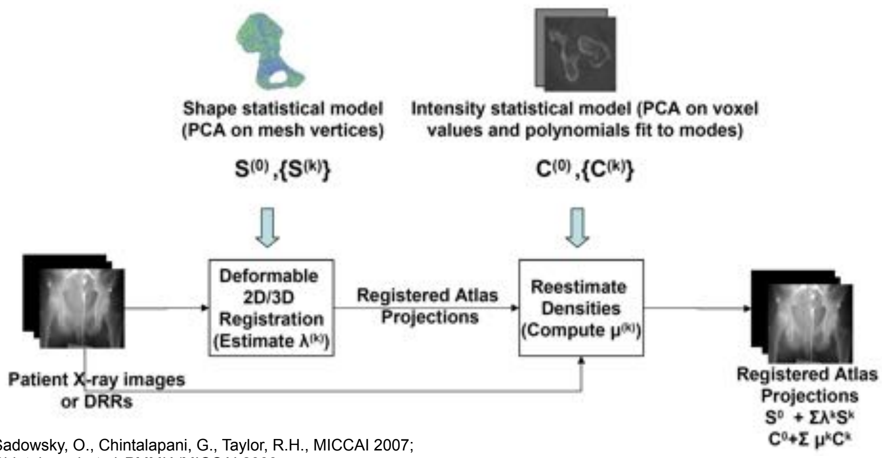
Mesh Vertex Improvement
(click here)

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

Jianhua Yao

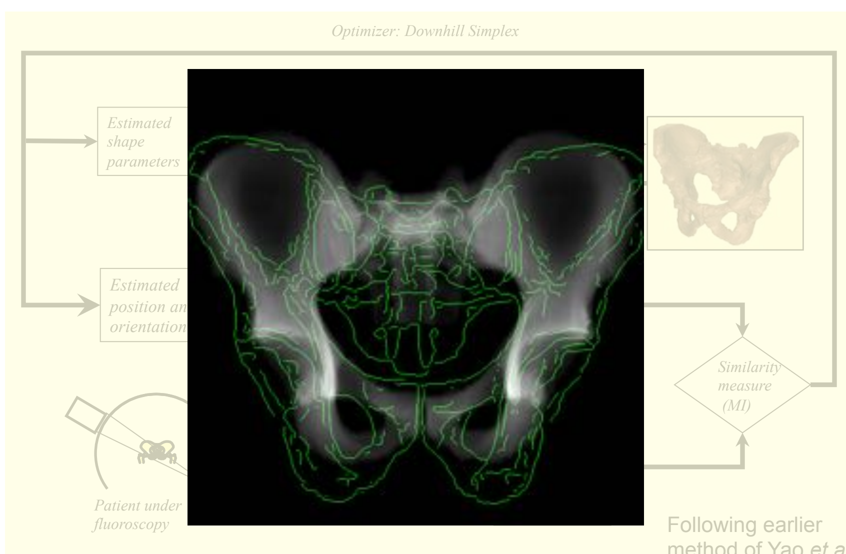
2D/3D Registration – Shape and Intensity Models



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

Deformable 3D/2D Registration

Ofri Sadowsky



2D/3D Registration – Shape and Intensity

(1)	(2)	(3)	(4)	(5)
#	$S^{true} - S^{est}$ (mm)	RMS (V^{true} , V^{est}_{mean}) (HU)	RMS(V^{true} , V^{est}_{modes}) (HU)	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**

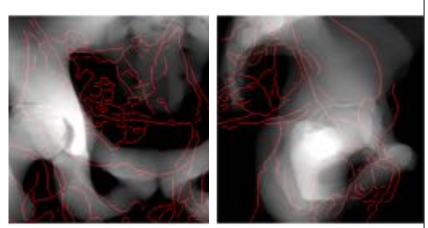
Table 1: Residual errors from leave-out-validation tests of the augmented registration algorithm. Column 2 shows the surface distance after 2D/3D shape registration. Columns 3 shows residual errors when using mean density only and column 4 shows residual errors with mean density and density modes. The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

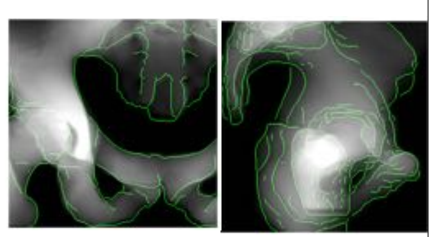
Slide credit: Gouthami Chintalapani

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

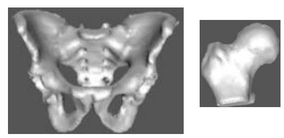


Hip atlas registered to hip projection images

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

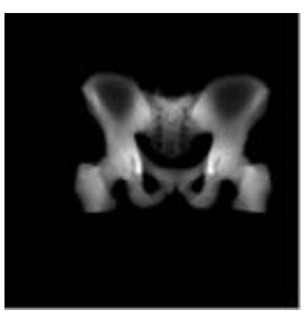


Separate Rigid

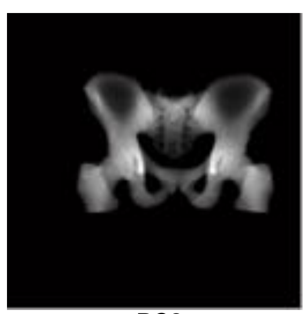
Combined Statistical Analysis

Slide credit: Gouthami Chintalapani

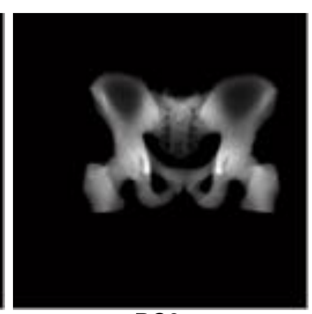
Multi-Component Hip Atlas



PC1



PC2



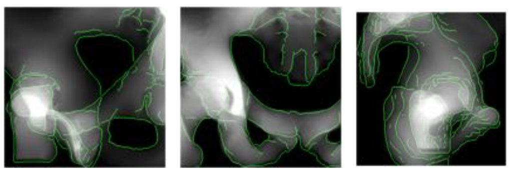
PC3

[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

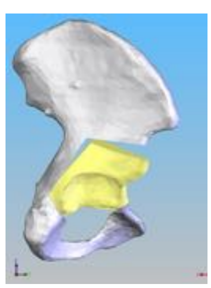


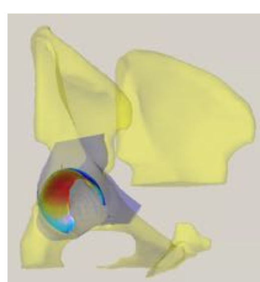
Atlas projections overlaid on DRR images after registration

2D/3D deformable registration

Chintalapani *et al.* CAOS 2009

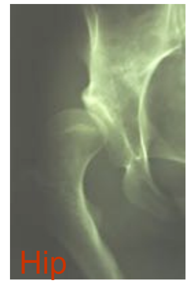
Applications – Hip Osteotomy

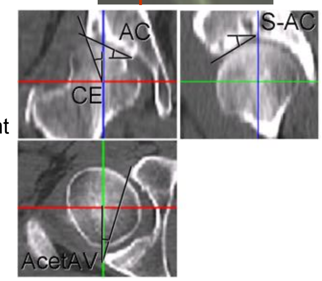




Background

- **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



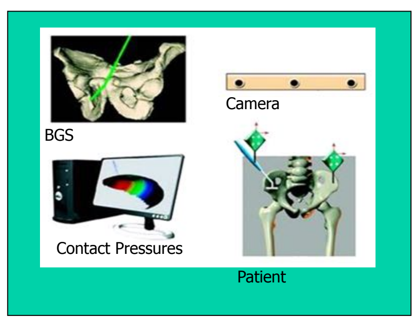


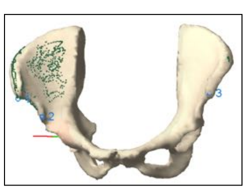
Anatomical measurements used to diagnose hip dysplasia

Slide credit: Gouthami Chintalapani, Mehran Armand

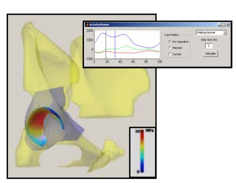
Biomechanical Guidance System (BGS)

- **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)

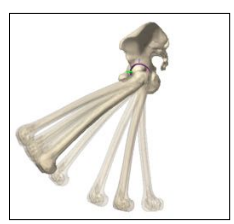




Model to Patient Registration



Joint contact-pressure after PAO

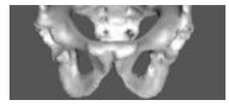


Hip-range-of-motion

Slide credit: Gouthami Chintalapani, Mehran Armand

Atlas Based Extrapolation of CT

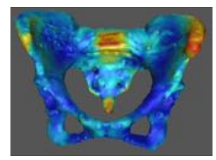
- **Problem**: Partial CT scans of patients
  - Dose minimization for young female patients
  - But the BGS needs full pelvis CT for planning



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

- **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

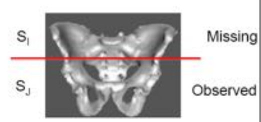


Distribution of surface registration errors of a patient pelvis model estimated from partial CT scan

Chintalapani *et al.* SPIE 2010

Atlas Adaptation to Partial Data

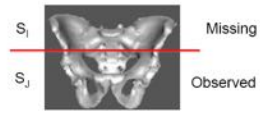
- Surface based registration of the observed data
- Rearrange $\bar{S}$ and U such that
  - Compute th $\bar{S} = \begin{bmatrix} \bar{S}_I \\ \bar{S}_J \end{bmatrix}$ $U = \begin{bmatrix} U_I \\ U_J \end{bmatrix}$ (R, T) between the atlas and th ... ith the mode weight parameters from observed data
- Infer the missing region



$$S^{est} = \begin{bmatrix} S_I^{est} \\ S_J^{est} \end{bmatrix} = \begin{bmatrix} \bar{S}_I \\ \bar{S}_J \end{bmatrix} + \lambda \begin{bmatrix} U_I \\ U_J \end{bmatrix}$$

Chintalapani *et al.* SPIE 2010

Atlas Adaptation to Partial Data with Xray Images



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

$$F, \Phi = \operatorname{argmax} \sum_i MI\left(I_i, DRR\left(F.\left(\bar{S}_I + \Phi U_I\right)\right)\right)$$

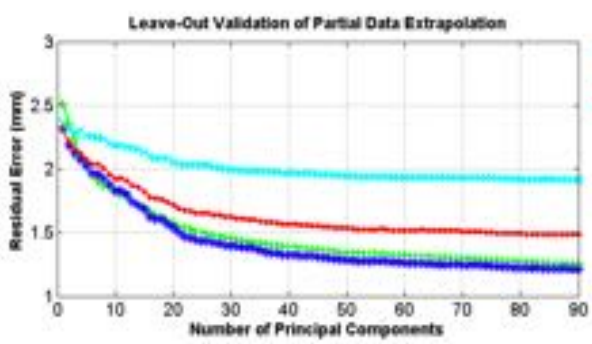
- Final atlas extrapolated model is given as

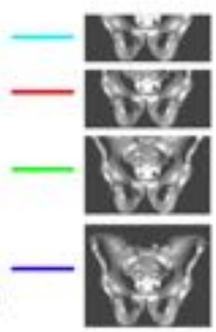
$$^{r}S^{est} = \begin{bmatrix} ^{r}S_I^{est} \\ S_J^{true} \end{bmatrix} = \begin{bmatrix} \left(\omega \circ R \circ S_I^{est} + T + \lambda U_I\right) \\ S_J^{true} \end{bmatrix}$$

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

Slide credit: Gouthami Chintalapani

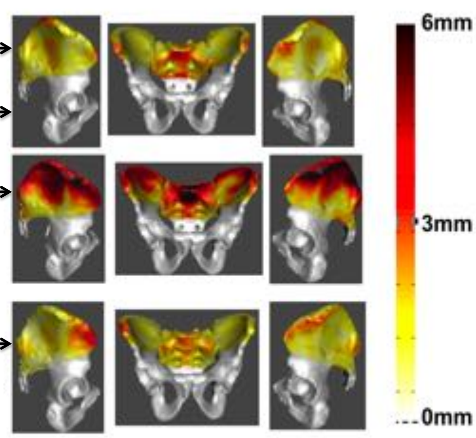
Results – Atlas Experiments

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

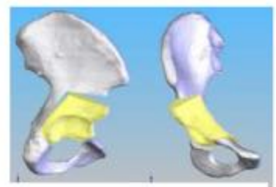
Slide credit: Gouthami Chintalapani, Mehran Armand

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



Results – Osteotomy Simulations

#	Full CT				Partial CT				Partial CT + X-ray			
	rot (○)	trans (mm)	mean (mm)	max (mm)	rot (○)	trans (mm)	mean (mm)	max (mm)	rot (○)	trans (mm)	mean (mm)	max (mm)
1	2.63	1.03	1.68	5.86	4.23	2.17	2.05	7.55	2.56	2.73	1.65	5.86
2	1.29	0.97	1.42	5.56	2.62	3.39	1.77	7.15	2.18	3.90	1.85	8.26
3	1.66	3.58	1.46	5.94	8.37	6.27	1.87	6.41	3.06	3.92	1.50	5.87
4	0.87	0.91	1.21	4.16	2.00	2.32	1.64	5.96	1.42	2.64	1.46	6.35
5	1.27	1.09	0.95	3.68	4.96	5.87	1.61	5.47	2.20	1.87	1.22	4.53
6	1.64	1.97	1.58	6.93	4.32	4.12	1.84	8.75	1.46	2.74	1.44	6.17
avg	1.56	1.59	1.38	5.35	4.41	4.02	1.79	6.88	2.14	2.96	1.52	6.16

Results from ICP registration experiments

#	Full CT				Partial CT				Partial CT + X-ray			
	Left		Right		Left		Right		Left		Right	
	rot (○)	trans (mm)	rot (○)	trans (mm)	rot (○)	trans (mm)	rot (○)	trans (mm)	rot (○)	trans (mm)	rot (○)	trans (mm)
1	0.46	3.66	0.84	4.37	0.33	2.74	2.14	10.75	0.47	3.67	0.91	5.09
2	0.38	2.49	0.32	1.52	0.52	3.83	0.48	2.35	0.20	1.21	0.24	0.61
3	0.57	4.28	0.48	3.10	1.79	10.57	2.09	11.2	0.46	2.93	1.47	8.65
4	0.09	0.58	0.22	1.55	0.47	3.41	0.69	3.69	0.42	3.53	0.37	2.09
5	0.44	2.62	0.37	1.81	1.46	8.37	1.62	8.2	0.53	3.56	0.83	4.23
6	0.36	1.72	0.47	2.25	1.52	11.4	0.71	4.02	0.16	0.93	0.25	0.97
avg	0.38	2.55	0.45	2.43	1.01	6.72	1.28	6.70	0.37	2.63	0.67	3.60

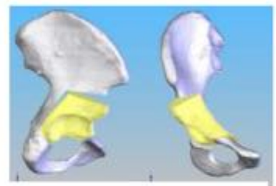
Results from Fragment Tracking Experiments

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



Results – Osteotomy Simulations

#	Full CT				Partial CT				Partial CT + X-ray			
	rot (o)	trans (mm)	mean (mm)	max (mm)	rot (o)	trans (mm)	mean (mm)	max (mm)	rot (o)	trans (mm)	mean (mm)	max (mm)
1	2.63	1.03	1.68	5.86	4.23	2.17	2.05	7.55	2.56	2.73	1.65	5.86
2	1.29	0.97	1.42	5.56	2.62	3.39	1.77	7.15	2.18	3.90	1.85	8.26
3	1.66	3.58	1.46	5.94	8.37	6.27	1.87	6.41	3.06	3.92	1.50	5.87
4	0.87	0.91	1.21	4.16	2.00	2.32	1.64	5.96	1.42	2.64	1.46	6.35
5	1.27	1.09	0.95	3.68	4.96	5.87	1.61	5.47	2.20	1.87	1.22	4.53
6	1.64	1.97	1.58	6.93	4.32	4.12	1.84	8.75	1.46	2.74	1.44	6.17
avg	1.56	1.59	1.38	5.35	4.41	4.02	1.79	6.88	2.14	2.96	1.52	6.16

Results from ICP registration experiments

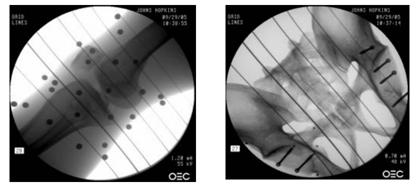
#	Full CT				Partial CT				Partial CT + X-ray			
	Left		Right		Left		Right		Left		Right	
	rot (o)	trans (mm)	rot (o)	trans (mm)	rot (o)	trans (mm)	rot (o)	trans (mm)	rot (o)	trans (mm)	rot (o)	trans (mm)
1	0.46	3.66	0.84	4.37	0.33	2.74	2.14	10.75	0.47	3.67	0.91	5.09
2	0.38	2.49	0.32	1.52	0.52	3.83	0.48	2.35	0.20	1.21	0.24	0.61
3	0.57	4.28	0.48	3.10	1.79	10.57	2.09	11.2	0.46	2.93	1.47	8.65
4	0.09	0.58	0.22	1.55	0.47	3.41	0.69	3.69	0.42	3.53	0.37	2.09
5	0.44	2.62	0.37	1.81	1.46	8.37	1.62	8.2	0.53	3.56	0.83	4.23
6	0.36	1.72	0.47	2.25	1.52	11.4	0.71	4.02	0.16	0.93	0.25	0.97
avg	0.38	2.55	0.45	2.43	1.01	6.72	1.28	6.70	0.37	2.63	0.67	3.60

Results from Fragment Tracking Experiments

Slide credit: Gouthami Chintalapani

C-arm Distortion

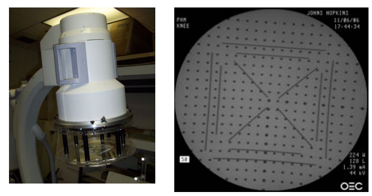
- What is distortion ?
  - –Avg distortion**: 2.14** mm/pixel
  - –max distortion: **4.60** mm/pixel



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

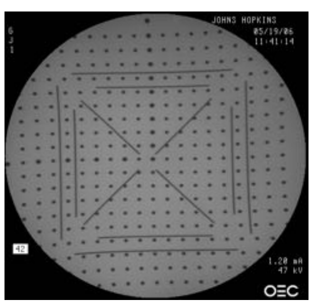
- How to rectify images ?
  - Phantom based correction
  - Polynomial functions to model distortion

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

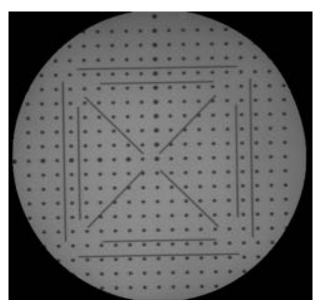


Typical bi-planar phantom used for C-arm calibration

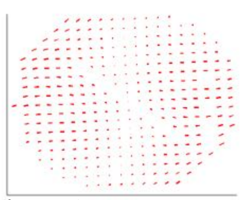
C-Arm Distortion Correction



Warped X-ray image of the phantom



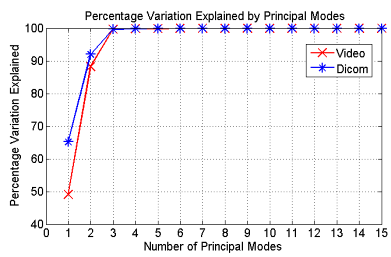
Dewarped X-ray image

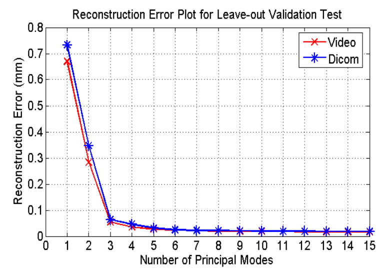


Distortion vector map

Statistical Characterization of C-Arm Distortion correction using PCA

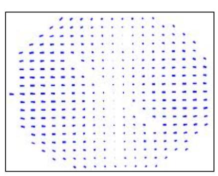
- Principal component analysis on distortion maps
  - 120 images, one every 3 degrees approx., along propeller axis (similar to the full sweep data used for 3D reconstruction)
  - 200 images to span the sphere defined by the “C” of the c-arm



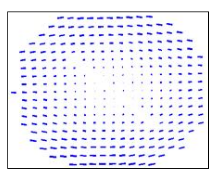


Slide credit: Gouthami Chintalapani

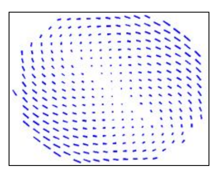
Circular Trajectory



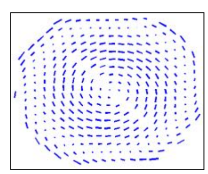
mean



mode1

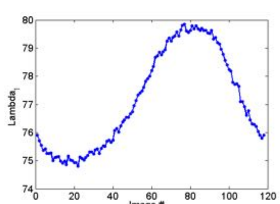


mode2

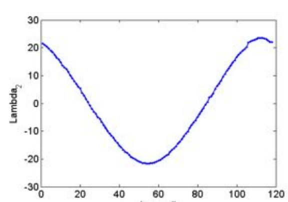


mode3

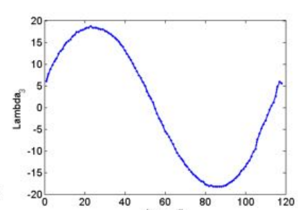
Distortion patterns from PCA modes



Lambda_1



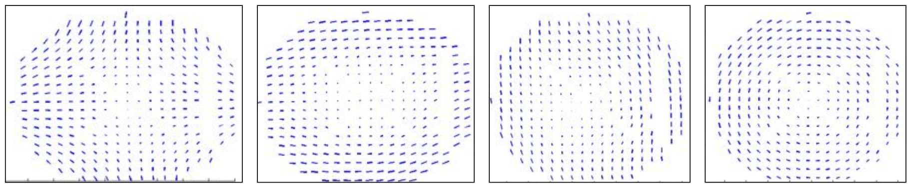
Lambda_2



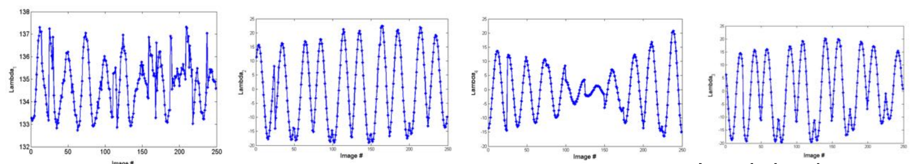
Lambda_3

Slide credit: Gouthami Chintalapani

C-arm Imaging Volume



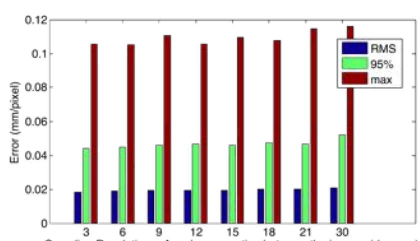
mode1 mode2 mode3 mode4



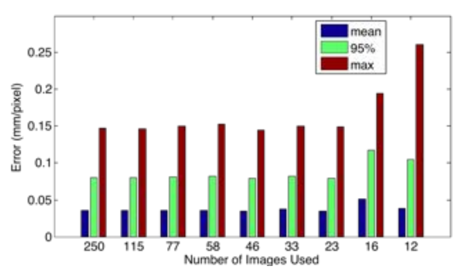
Lambda_1 Lambda_2 Lambda_3 Lambda_4

Sampling Resolution

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



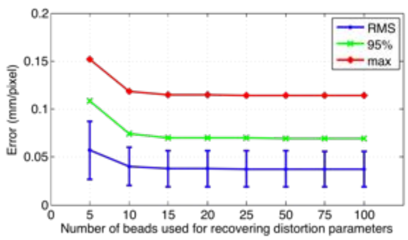
Circular Trajectory



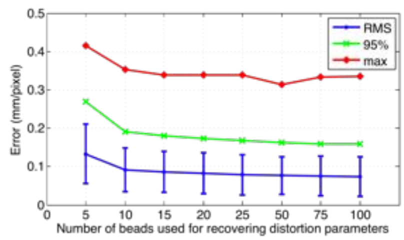
C-arm Imaging Volume

Recovering Distortion Parameters

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

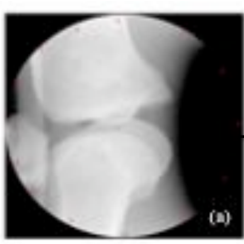


Circular Arc



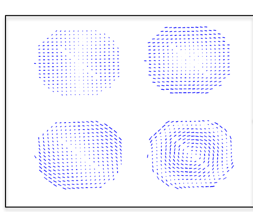
C-arm Imaging Volume

Small Phantom based Distortion Correction



Example patient image peripheral beads

Peripheral bead locations

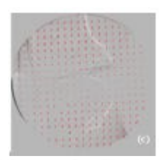


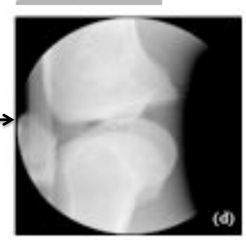
Prior distortion model

Distortion modes

Distortion mode matching

Distortion corrected image

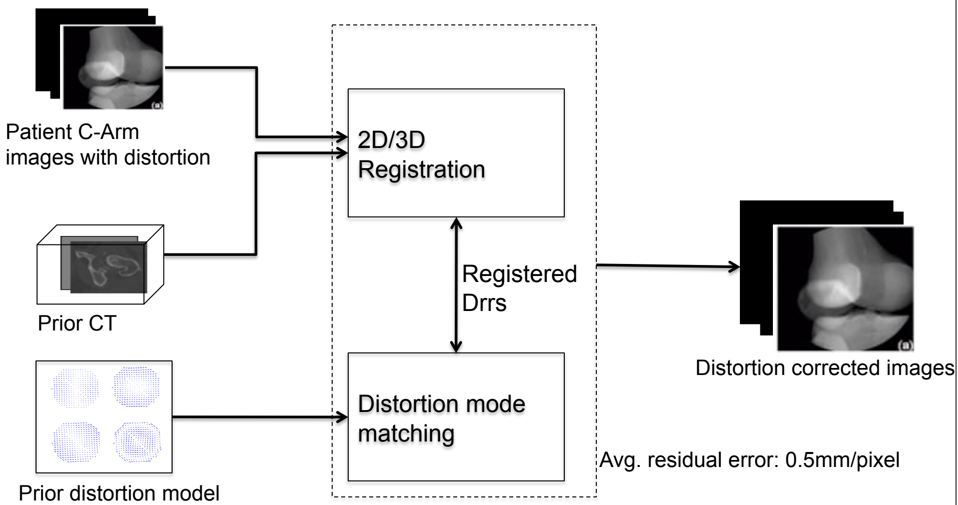




Avg. residual error: 0.2mm/pixel
Max. residual error: 0.8mm/pixel

Chintalapani *et al.* SPIE 2007

Using Patient CT as Fiducial



Chintalapani *et al.* ISBI 2007

Slide credit: Gouthami Chintalapani