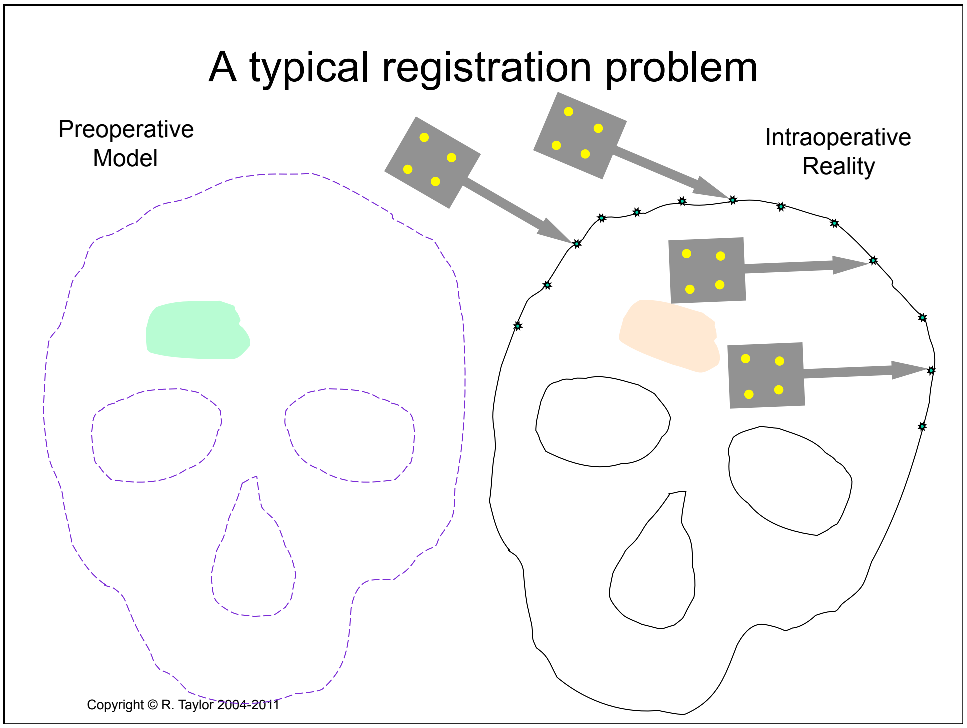
Registration

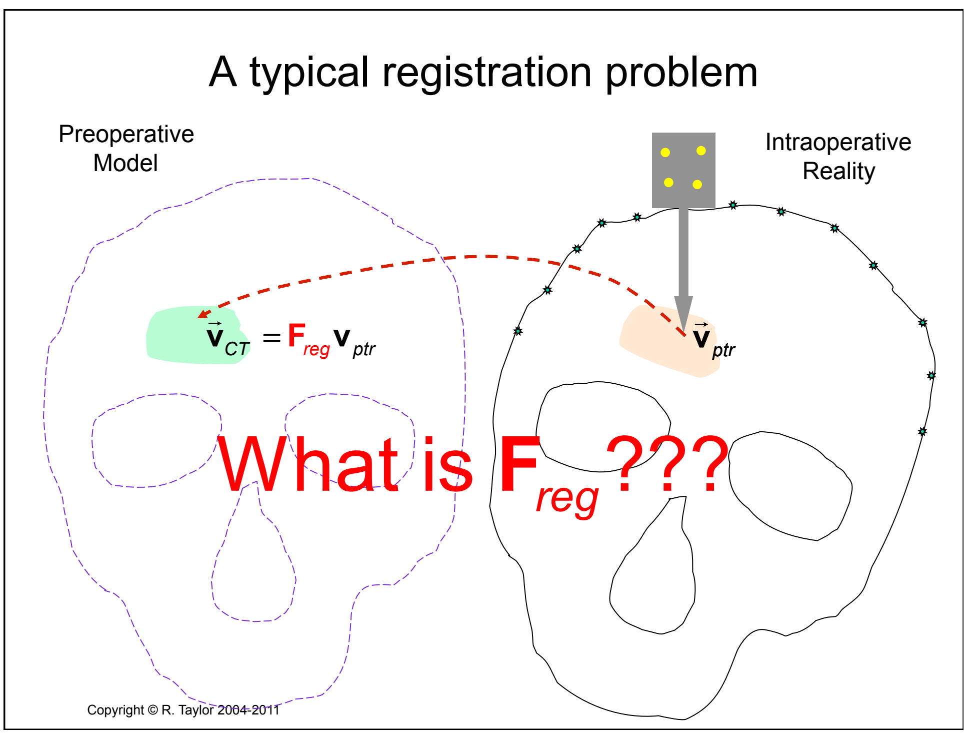
600.445 Computer-Integrated Surgery
Russell H. Taylor

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
Preoperative Model
Intraoperative Reality
Copyright © R. Taylor 2004-2011



A typical registration problem
Preoperative Model
Intraoperative Reality
$\vec{\mathbf{v}}_{CT} = \mathbf{F}_{reg}\mathbf{v}_{ptr}$
$\vec{\mathbf{v}}_{ptr}$
What is $\mathbf{F}_{reg}$???
Copyright © R. Taylor 2004-2011

Framework for feature-based methods

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

Taxonomy of methods

- Feature-based
- Intensity-based

Definitions

Overall Goal: Given two coordinate systems,

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

and coordinates

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

associated with homologous 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:

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

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

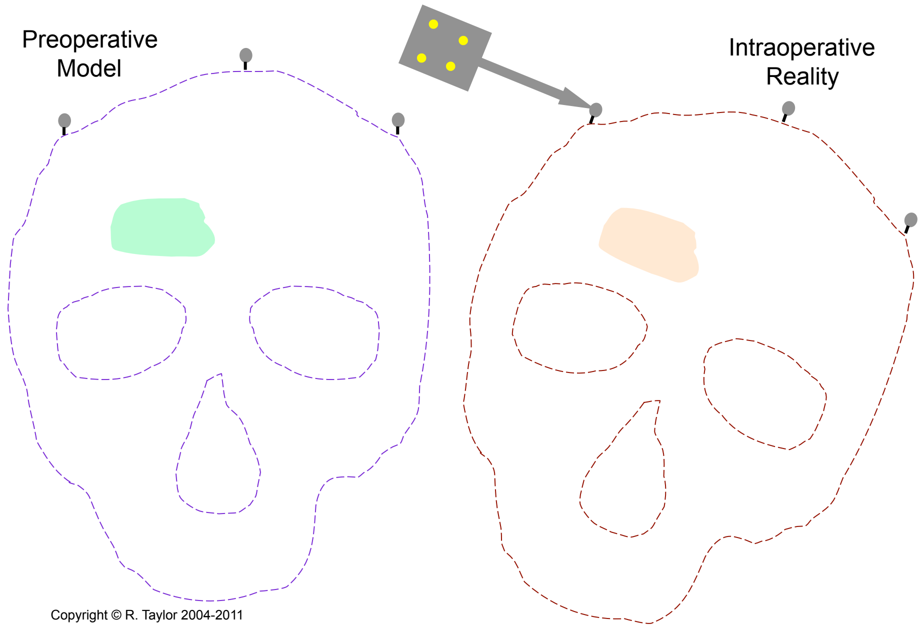
$$D = \operatorname*{median}_i distance(F_{Ai}, \mathbf{T}(F_{Bi}))$$

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

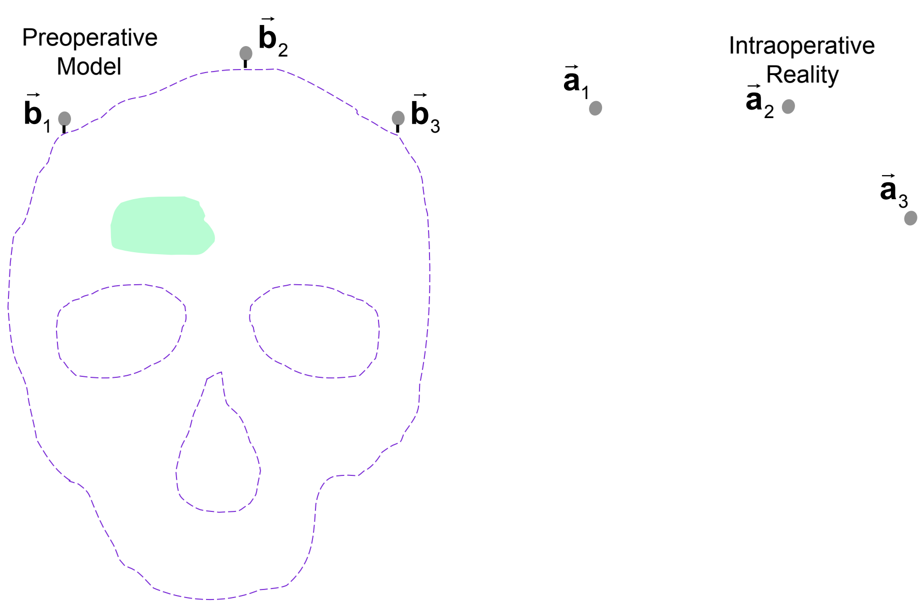
Optimization

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

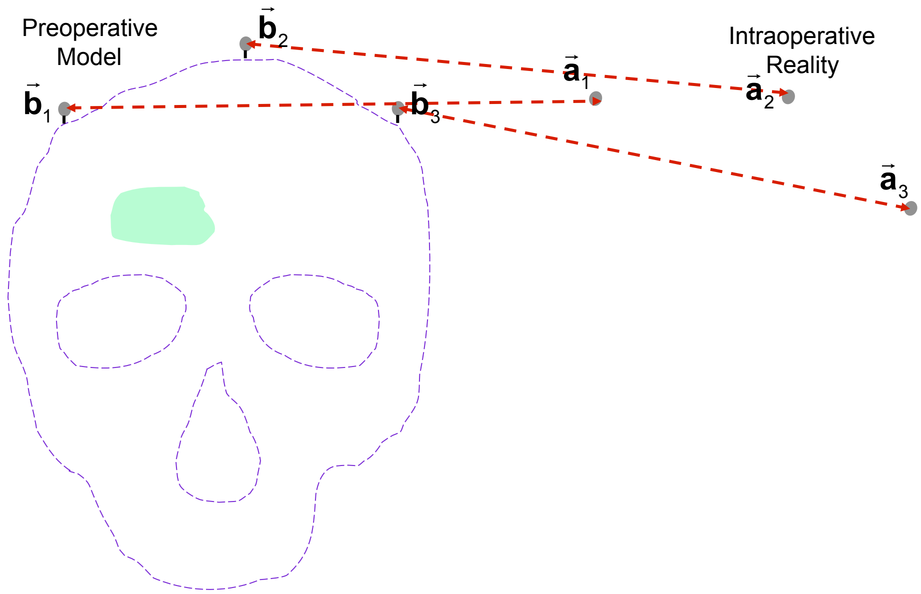
A typical fiducial-based registration problem



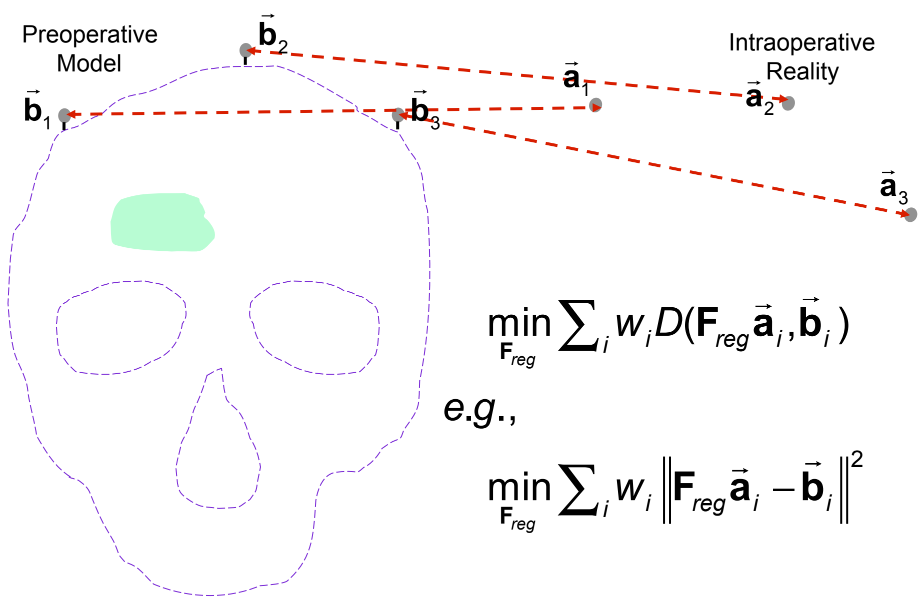
What the computer knows



Identify corresponding points



Find best rigid transformation!

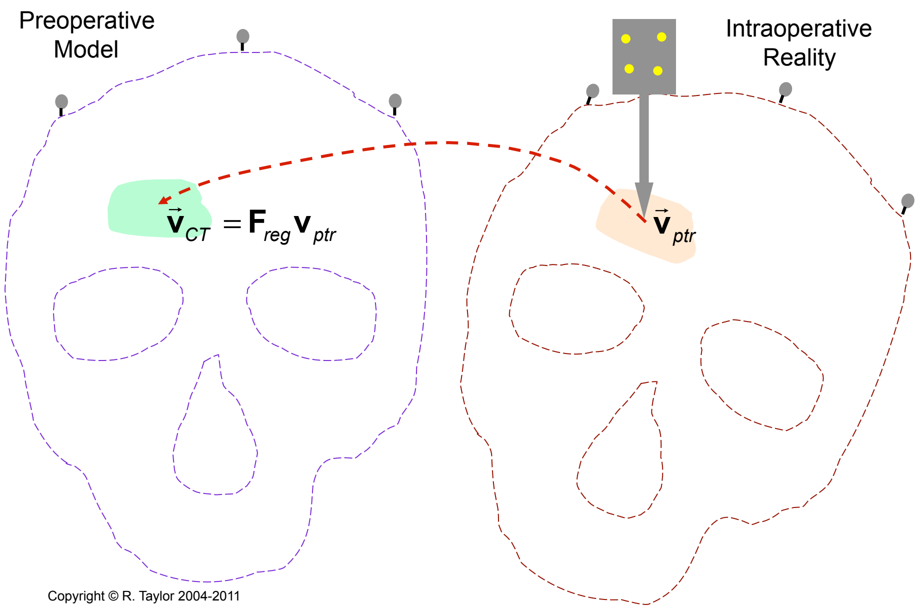


$$\min_{\mathbf{F}_{reg}} \sum_i w_i D(\mathbf{F}_{reg}\vec{\mathbf{a}}_i, \vec{\mathbf{b}}_i)$$

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

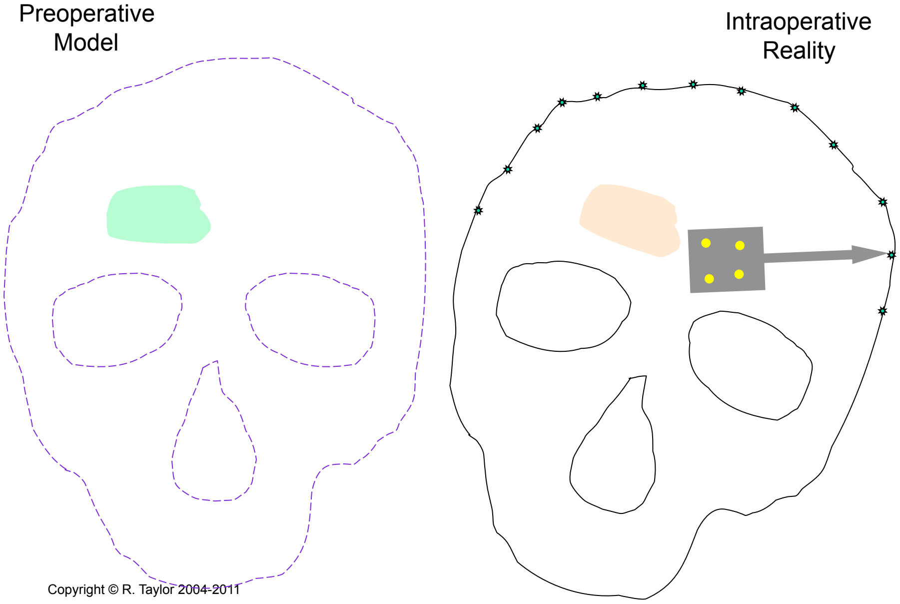
Outline:

- 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.

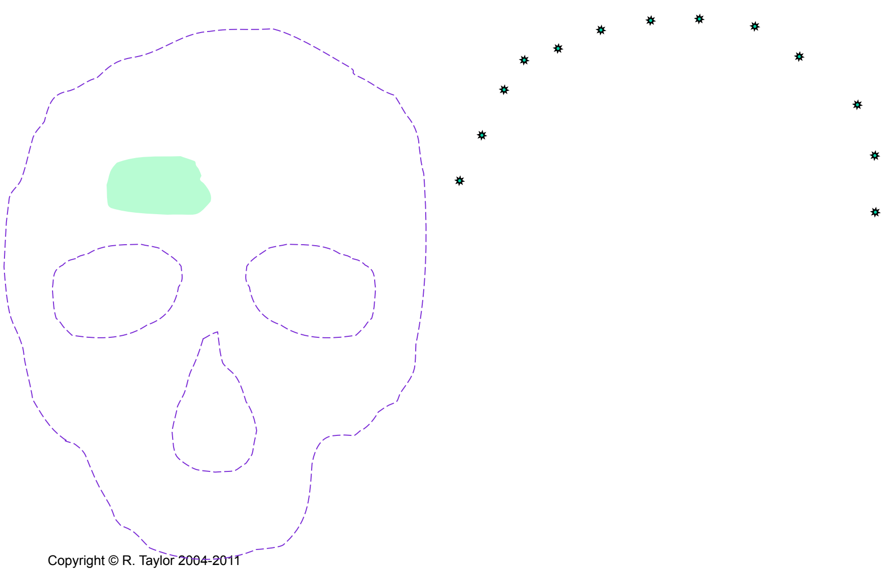
Examples

- 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]

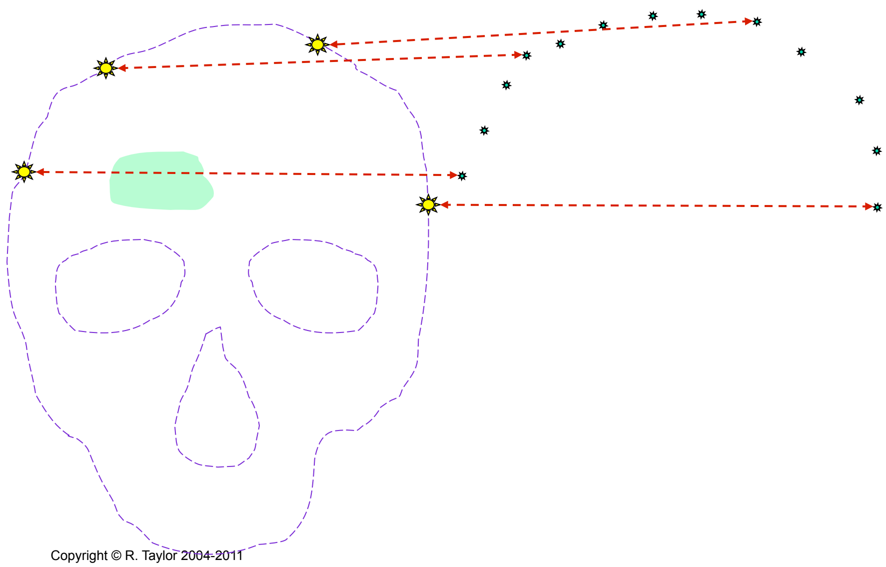
A typical surface registration problem



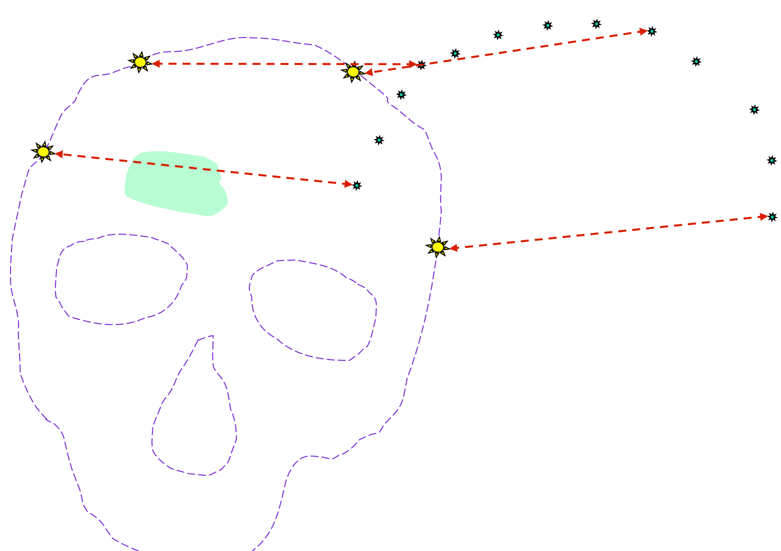
What the computer knows



Find homologous points & pull!



Find homologous points & pull!

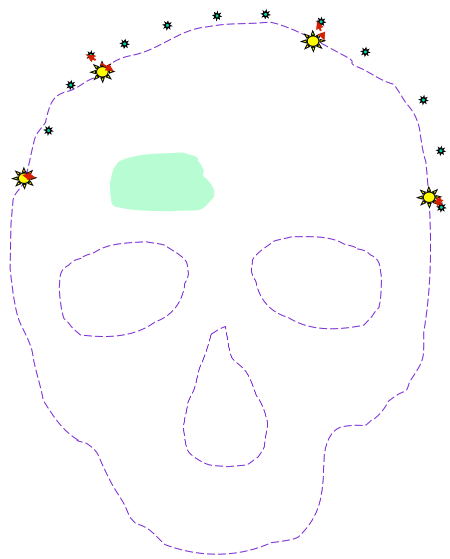


Find homologous points & pull!

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 $\mathbf{f}_i \in \mathcal{F}_H$ and a surface model $\mathcal{F}_B$, computes a rigid transformation $\mathbf{T}$ that minimizes

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

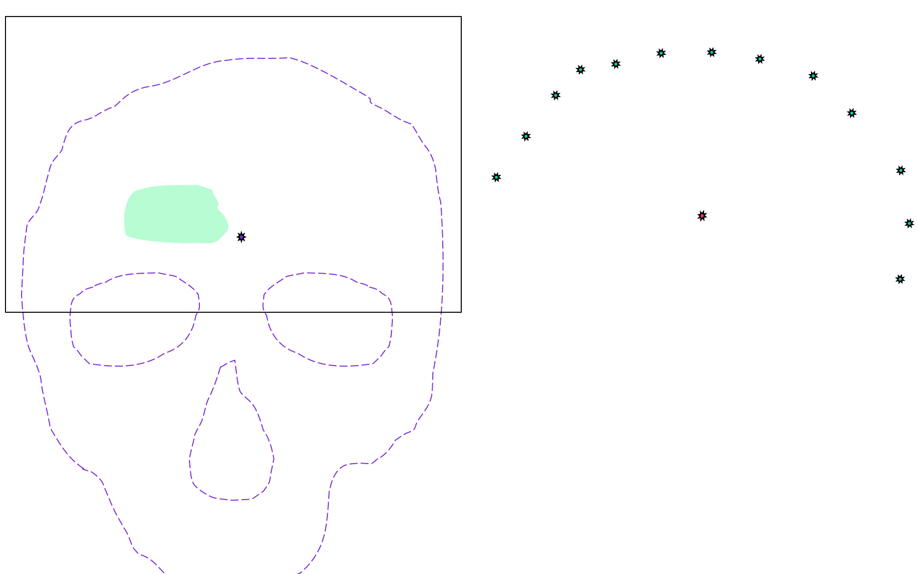
  where d_S is defined below, given a good initial guess for $\mathbf{T}$.
- Optimization uses standard numerical method (steepest gradient descent [Powell]) to find six parameters (3 rotations, 3 translations) defining $\mathbf{T}$

Head in Hat Algorithm

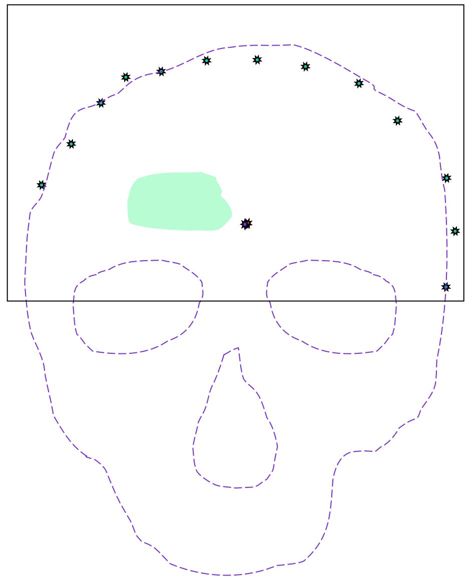
Definition of $d_S(\mathcal{F}_B, \mathbf{f}_i)$

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

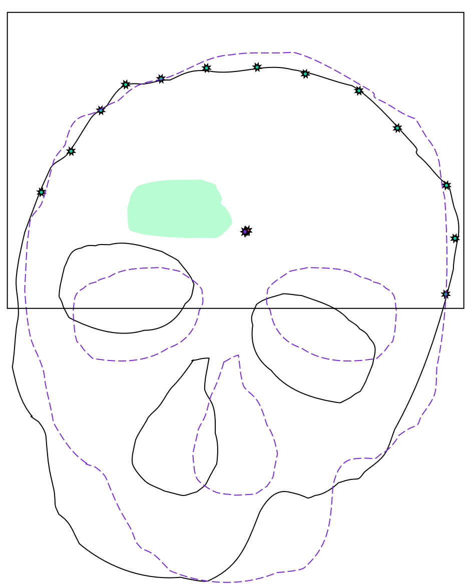
Head-in-hat algorithm: step 0



Head-in-hat algorithm: step1

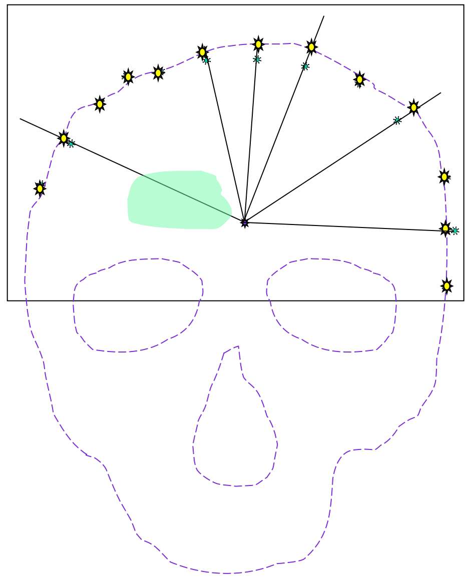


Head-in-hat algorithm: step1

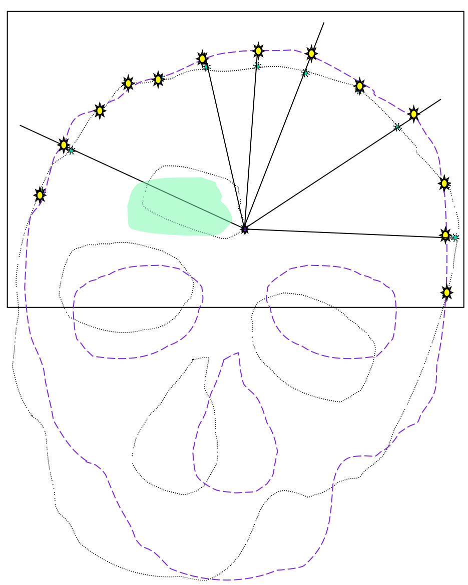


Copyright © R. Taylor 2004-2011

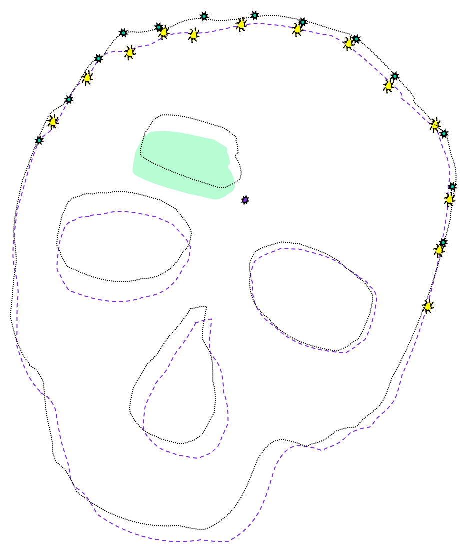
Head-in-hat algorithm: step 2



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

Minimizing Rigid Registration Errors

Typically, given a set of points $\{\mathbf{a}_i\}$ in one coordinate system and another set of points $\{\mathbf{b}_i\}$ in a second coordinate system

Goal is to find $[\mathbf{R},\mathbf{p}]$ that minimizes

$$\eta = \sum_i \mathbf{e}_i \bullet \mathbf{e}_i$$

where

$$\mathbf{e}_i = (\mathbf{R} \bullet \mathbf{a}_i + \mathbf{p}) - \mathbf{b}_i$$

This is tricky, because of $\mathbf{R}$.

Minimizing Rigid Registration Errors

Step 1: Compute

$$\overline{\mathbf{a}} = \frac{1}{N}\sum_{i=1}^{N} \vec{\mathbf{a}}_i \qquad \overline{\mathbf{b}} = \frac{1}{N}\sum_{i=1}^{N} \vec{\mathbf{b}}_i$$

$$\tilde{\mathbf{a}}_i = \vec{\mathbf{a}}_i - \overline{\mathbf{a}} \qquad \tilde{\mathbf{b}}_i = \vec{\mathbf{b}}_i - \overline{\mathbf{b}}$$

Step 2: Find $\mathbf{R}$ that minimizes

$$\sum_i (\mathbf{R} \cdot \tilde{\mathbf{a}}_i - \tilde{\mathbf{b}}_i)^2$$

Step 3: Find $\vec{\mathbf{p}}$

$$\vec{\mathbf{p}} = \overline{\mathbf{b}} - \mathbf{R} \cdot \overline{\mathbf{a}}$$

Step 4: Desired transformation is

$$\mathbf{F} = Frame(\mathbf{R}, \vec{\mathbf{p}})$$

Solving for R: iteration method

Given $\{\cdots,(\tilde{\mathbf{a}}_i,\tilde{\mathbf{b}}_i),\cdots\}$, want to find $\mathbf{R} = \arg\min \sum_i \left(\mathbf{R}\tilde{\mathbf{a}}_i - \tilde{\mathbf{b}}_i\right)$

Step 0: Make an initial guess $\mathbf{R}_0$

Step 1: Given $\mathbf{R}_k$, compute $\breve{\mathbf{b}}_i = \mathbf{R}_k^{-1}\tilde{\mathbf{b}}_i$

Step 2: Compute $\Delta\mathbf{R}$ that minimizes

$$\sum_i (\Delta\mathbf{R}\,\tilde{\mathbf{a}}_i - \breve{\mathbf{b}}_i)^2$$

Step 3: Set $\mathbf{R}_{k+1} = \mathbf{R}_k\Delta\mathbf{R}$

Step 4: Iterate Steps 1-3 until residual error is sufficiently small (or other termination condition)

Iterative method: Solving for ΔR

Approximate $\Delta\mathbf{R}$ as $(\mathbf{I} + skew(\vec{\alpha}))$. I.e.,

$$\Delta\mathbf{R} \bullet \mathbf{v} \approx \mathbf{v} + \vec{\alpha}\times\mathbf{v}$$

for any vector $\mathbf{v}$. Then, our least squares problem becomes

$$\min_{\Delta\mathbf{R}} \sum_i (\Delta\mathbf{R}\bullet\tilde{\mathbf{a}}_i - \breve{\mathbf{b}}_i)^2 \approx \min_{\vec{\alpha}} \sum_i (\tilde{\mathbf{a}}_i - \breve{\mathbf{b}}_i + \vec{\alpha}\times\tilde{\mathbf{a}}_i)^2$$

This is linear least squares problem in $\vec{\alpha}$.

Then compute $\Delta\mathbf{R}(\vec{\alpha})$.

Direct Techniques to solve for R

- Method due to K. Arun, et. al., IEEE PAMI, Vol 9, no 5, pp 698-700, Sept 1987

Step 1: Compute

$$\mathbf{H} = \sum_i \begin{bmatrix} \overline{a}_{i,x}\overline{b}_{i,x} & \overline{a}_{i,x}\overline{b}_{i,y} & \overline{a}_{i,x}\overline{b}_{i,z} \\ \overline{a}_{i,y}\overline{b}_{i,x} & \overline{a}_{i,y}\overline{b}_{i,y} & \overline{a}_{i,y}\overline{b}_{i,z} \\ \overline{a}_{i,z}\overline{b}_{i,x} & \overline{a}_{i,z}\overline{b}_{i,y} & \overline{a}_{i,z}\overline{b}_{i,z} \end{bmatrix}$$

Step 2: Compute the SVD of $\mathbf{H} = \mathbf{USV^t}$

Step 3: $\mathbf{R} = \mathbf{VU^t}$

Step 4: Verify $Det(\mathbf{R}) = 1$. If not, then algorithm may fail.

- Failure is rare, and mostly fixable. The paper has details.

Quarternion Technique to solve for R

- B.K.P. Horn, "Closed form solution of absolute orientation using unit quaternions", *J. Opt. Soc. America*, A vol. 4, no. 4, pp 629-642, Apr. 1987.
- Method described as reported in Besl and McKay, "A method for registration of 3D shapes", *IEEE Trans. on Pattern Analysis and Machine* Intelligence, vol. 14, no. 2, February 1992.
- Solves a 4x4 eigenvalue problem to find a unit quaternion corresponding to the rotation
- This quaternion may be converted in closed form to get a more conventional rotation matrix

Digression: quaternions

Invented by Hamilton as a way to express the ratio of vectors. Can be thought of as

4 elements: $\mathbf{q} = [q_0, q_1, q_2, q_3]$

scalar & vector: $\mathbf{q} = s + \vec{\mathbf{v}} = [s, \vec{\mathbf{v}}]$

$$\mathbf{q} = q_0 + q_1\vec{\mathbf{i}} + q_2\vec{\mathbf{j}} + q_3\vec{\mathbf{k}}$$

Properties:

Linearity: $\lambda\mathbf{q}_1 + \mu\vec{\mathbf{q}}_2 = [\lambda s_1 + \mu s_2, \lambda\vec{\mathbf{v}}_1 + \mu\vec{\mathbf{v}}_2]$

Conjugate: $\mathbf{q}^* = s - \vec{\mathbf{v}} = [s, -\vec{\mathbf{v}}]$

Product: $\mathbf{q}_1 \circ \mathbf{q}_2 = [s_1 s_2 - \vec{\mathbf{v}}_1 \bullet \vec{\mathbf{v}}_2, s_1\vec{\mathbf{v}}_2 + s_2\vec{\mathbf{v}}_1 + \vec{\mathbf{v}}_1 \times \vec{\mathbf{v}}_2]$

Transform vector: $\mathbf{q} \circ \vec{\mathbf{p}} \quad = \mathbf{q} \circ [0, \vec{\mathbf{p}}] \circ \mathbf{q}^*$

Norm: $\|\mathbf{q}\| = \sqrt{s^2 + \vec{\mathbf{v}} \bullet \vec{\mathbf{v}}} = \sqrt{q_0{}^2 + q_1{}^2 + q_2{}^2 + q_3{}^2}$

Digression continued: unit quaternions

We can associate a rotation by angle θ about an axis $\vec{\mathbf{n}}$ with the unit quaternion:

$$Rot(\vec{\mathbf{n}}, \theta) \Leftrightarrow \left[\cos\frac{\theta}{2},\ \sin\frac{\theta}{2}\vec{\mathbf{n}}\right]$$

Exercise: Demonstrate this relationship. I.e., show

$$Rot((\vec{\mathbf{n}}, \theta) \bullet \vec{\mathbf{p}} = \left[\cos\frac{\theta}{2},\ \sin\frac{\theta}{2}\vec{\mathbf{n}}\right] \circ [0, \vec{\mathbf{p}}] \circ \left[\cos\frac{\theta}{2},\ -\sin\frac{\theta}{2}\vec{\mathbf{n}}\right]$$

Rotation matrix from unit quaternion

$$\mathbf{q} = [q_0, q_1, q_2, q_3]; \quad \|\mathbf{q}\| = 1$$

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} q_0^{\ 2} + q_1^{\ 2} - q_2^{\ 3} - q_3^{\ 3} & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^{\ 2} - q_1^{\ 2} + q_2^{\ 3} - q_3^{\ 3} & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^{\ 2} - q_1^{\ 2} - q_2^{\ 3} + q_3^{\ 3} \end{bmatrix}$$

Unit quaternion from rotation matrix

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{bmatrix}; \quad \begin{array}{l} a_0 = 1 + r_{xx} + r_{yy} + r_{zz};\ a_1 = 1 + r_{xx} - r_{yy} - r_{zz} \\ a_2 = 1 - r_{xx} + r_{yy} - r_{zz};\ a_3 = 1 - r_{xx} - r_{yy} + r_{zz} \end{array}$$

$a_0 = \max\{a_k\}$	$a_1 = \max\{a_k\}$	$a_2 = \max\{a_k\}$	$a_3 = \max\{a_k\}$
$q_0 = \frac{\sqrt{a_0}}{2}$	$q_0 = \frac{r_{yz} - r_{zy}}{4q_1}$	$q_0 = \frac{r_{zx} - r_{xz}}{4q_2}$	$q_0 = \frac{r_{xy} - r_{yx}}{4q_3}$
$q_1 = \frac{r_{xy} - r_{yx}}{4q_0}$	$q_1 = \frac{\sqrt{a_1}}{2}$	$q_1 = \frac{r_{xy} + r_{yx}}{4q_2}$	$q_1 = \frac{r_{xz} + r_{zx}}{4q_3}$
$q_2 = \frac{r_{zx} - r_{xz}}{4q_0}$	$q_2 = \frac{r_{xy} + r_{yx}}{4q_1}$	$q_2 = \frac{\sqrt{a_2}}{2}$	$q_2 = \frac{r_{yz} + r_{zy}}{4q_3}$
$q_3 = \frac{r_{yz} - r_{zy}}{4q_0}$	$q_3 = \frac{r_{xz} + r_{zx}}{4q_1}$	$q_3 = \frac{r_{yz} + r_{zy}}{4q_2}$	$q_3 = \frac{\sqrt{a_3}}{2}$

Quaternion method for R

Step 1: Compute

$$\mathbf{H} = \sum_i \begin{bmatrix} \overline{a}_{i,x}\overline{b}_{i,x} & \overline{a}_{i,x}\overline{b}_{i,y} & \overline{a}_{i,x}\overline{b}_{i,z} \\ \overline{a}_{i,y}\overline{b}_{i,x} & \overline{a}_{i,y}\overline{b}_{i,y} & \overline{a}_{i,z}\overline{b}_{i,z} \\ \overline{a}_{i,z}\overline{b}_{i,x} & \overline{a}_{i,z}\overline{b}_{i,y} & \overline{a}_{i,z}\overline{b}_{i,z} \end{bmatrix}$$

Step 2: Compute

$$\mathbf{G} = \begin{bmatrix} trace(\mathbf{H}) & \Delta^T \\ \Delta & \mathbf{H} + \mathbf{H}^T - trace(\mathbf{H})\mathbf{I} \end{bmatrix}$$

where $\Delta^T = \begin{bmatrix} \mathbf{H}_{2,3} - \mathbf{H}_{3,2} & \mathbf{H}_{3,1} - \mathbf{H}_{1,3} & \mathbf{H}_{1,2} - \mathbf{H}_{2,1} \end{bmatrix}$

Step 3: Compute eigen value decomposition of $\mathbf{G}$

$\text{diag}(\overline{\lambda}) = \mathbf{Q}^T\mathbf{G}\mathbf{Q}$

Step 4: The eigenvector $\mathbf{Q}_k = [q_0, q_1, q_2, q_3]$ corresponding to the largest eigenvalue λ_k is a unit quaternion corresponding to the rotation.

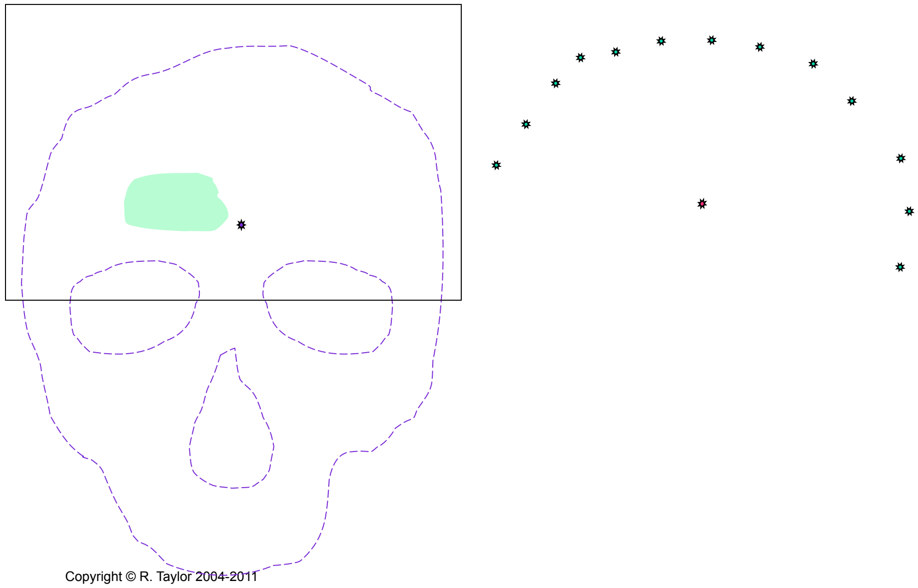
Iterative Closest Point

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

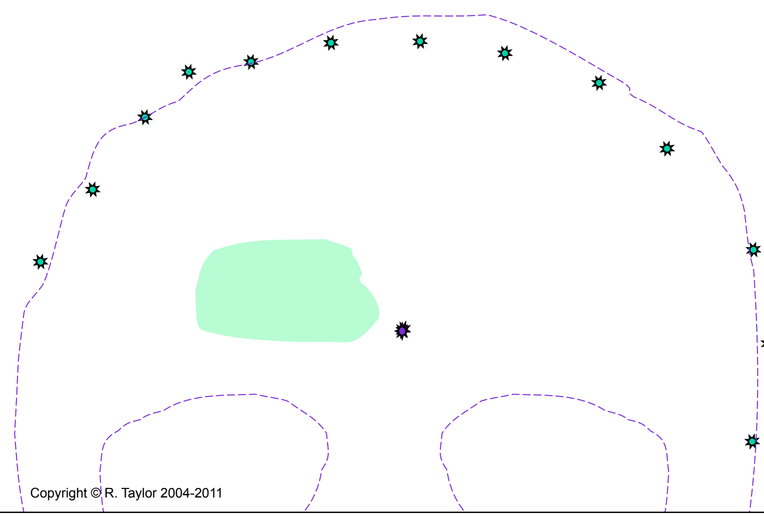
$$D_{k+1} = \sum_i \|\mathbf{v}_i - \mathbf{T}_{k+1} \cdot \mathbf{f}_i\|^2$$

- Physical Analogy

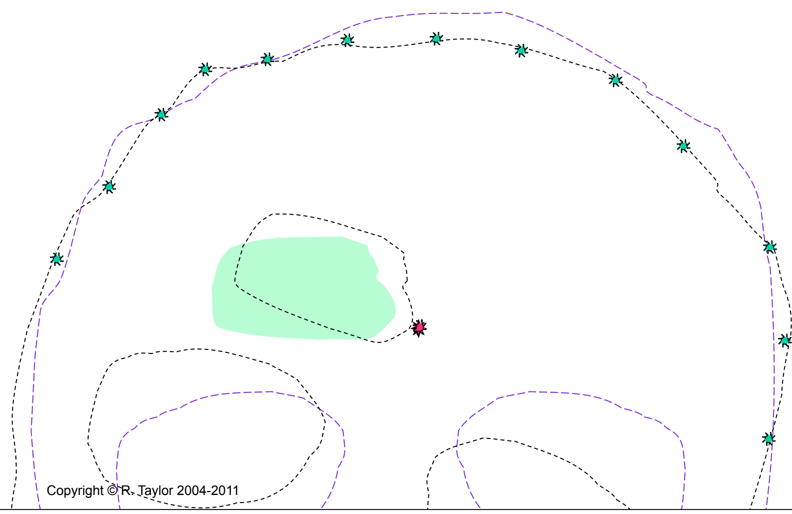
Iterative Closest Point: step 0



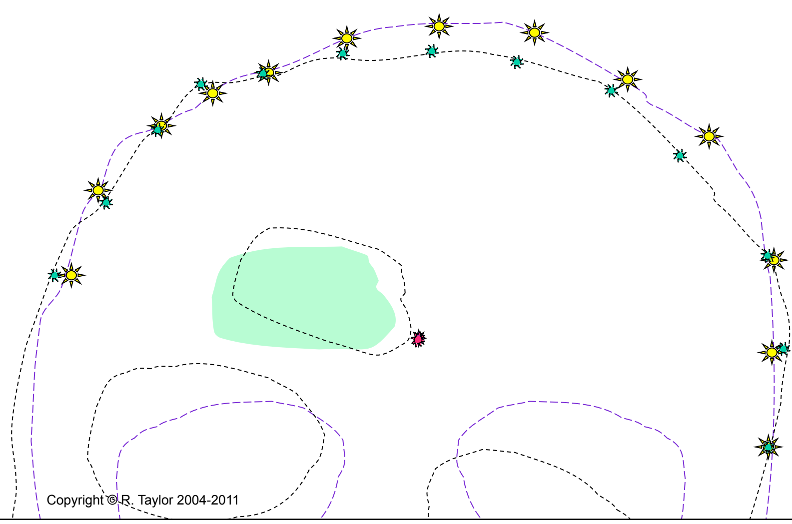
Iterative Closest Point: step1



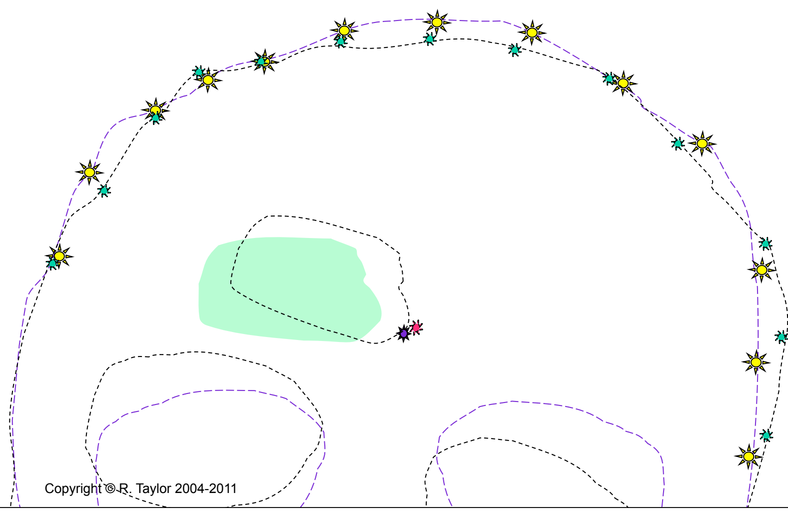
Iterative Closest Point: step1



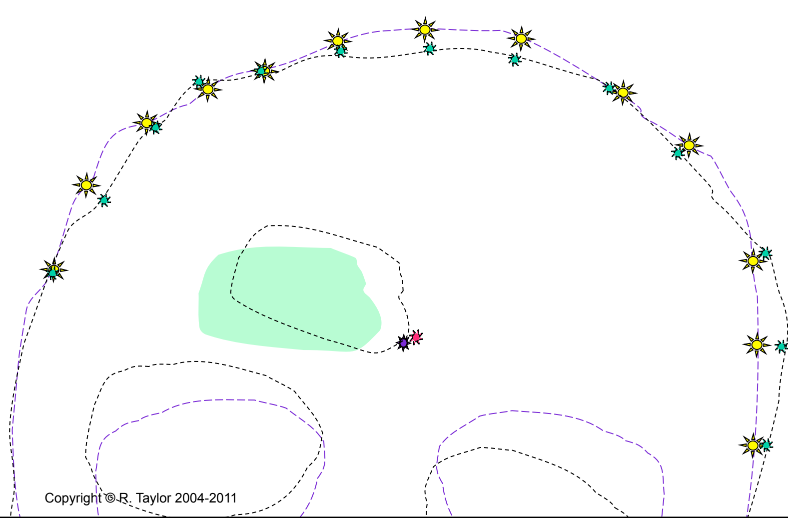
Iterative Closest Point: step2



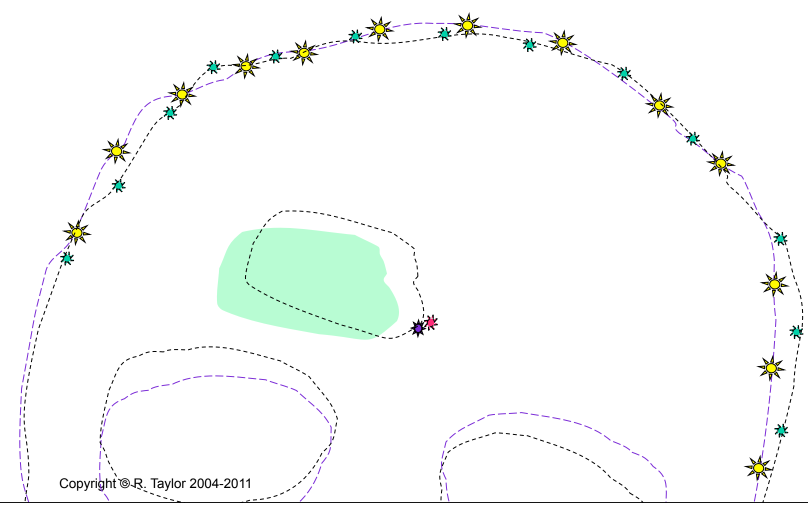
Iterative Closest Point: step 3



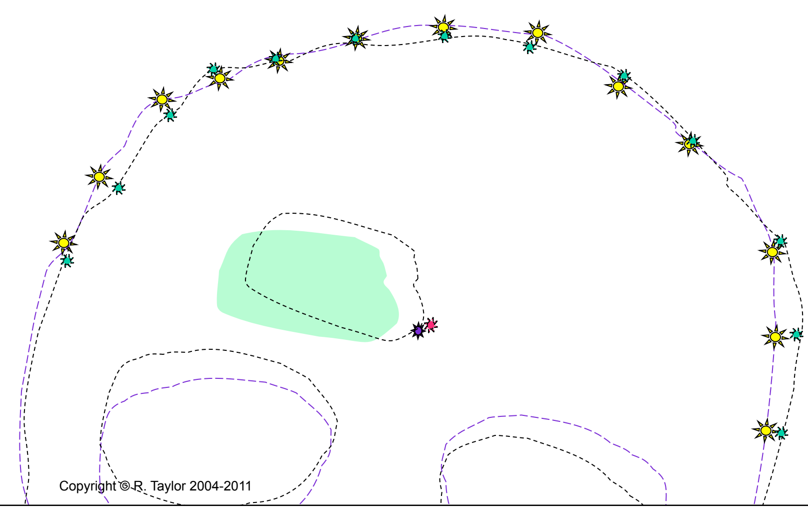
Iterative Closest Point: step 2 interation 2



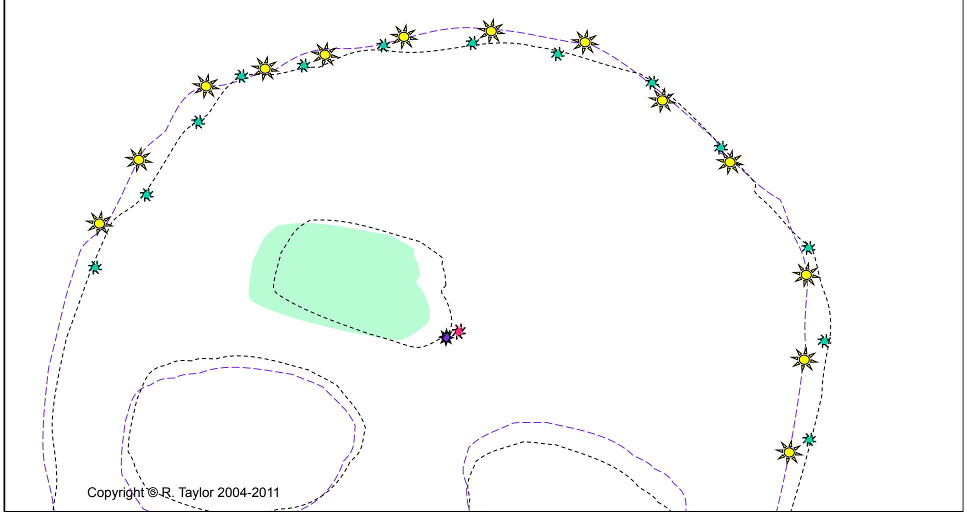
Iterative Closest Point: step 3 interation 2



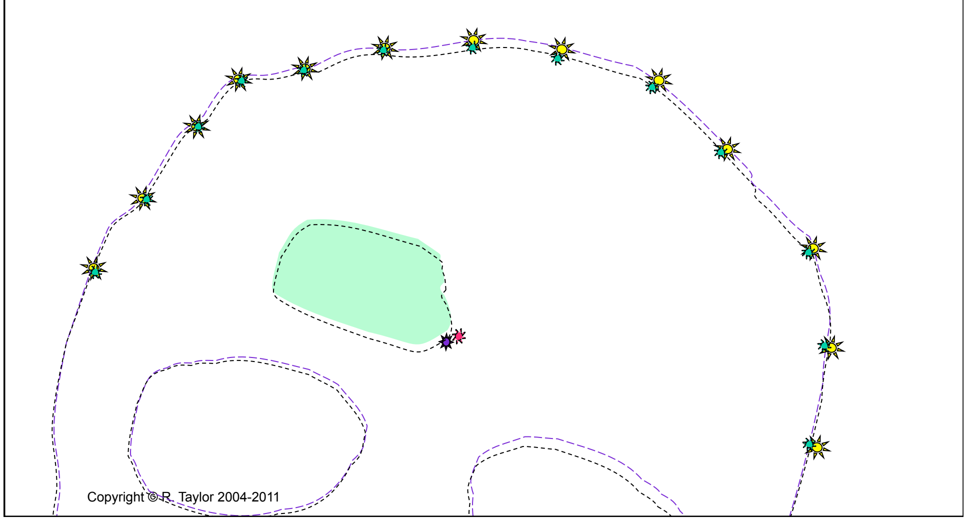
Iterative Closest Point: step 2 interation 3



Iterative Closest Point: step 3 interation 3



Iterative Closest Point: step 3 interation N



Iterative Closest Point: Discussion

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

Outline of a practical ICP code

Given

1. Surface model 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 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 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 `M` to `Q`
$\mathtt{D} = \{\cdots, d_k, \cdots\}$	Distances $d_k = \lVert \vec{\mathbf{c}}_k - \mathbf{F}_n \cdot \vec{\mathbf{q}}_k \rVert$
$\mathtt{I} = \{\cdots, i_k, \cdots\}$	Indices of triangles $\mathtt{m}_{i_k}$ corresp. to $\vec{\mathbf{c}}_k$
$\mathtt{A} = \{\cdots, \vec{\mathbf{a}}_k, \cdots\}$	Subset of `Q` with valid matches
$\mathtt{B} = \{\cdots, \vec{\mathbf{b}}_k, \cdots\}$	Points on `M`corresponding to `A`
$\mathtt{E} = \{\cdots, \vec{\mathbf{e}}_k, \cdots\}$	Residual errors $\vec{\mathbf{b}}_k - \mathbf{F} \cdot \vec{\mathbf{a}}_k$
$\sigma_n, \ (\varepsilon_{\max})_n, \bar{\varepsilon}_n$	$\dfrac{\sum_k \vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}{NumElts(\mathtt{E})}; \quad \max_k \sqrt{\vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}; \ \dfrac{\sum_k \sqrt{\vec{\mathbf{e}}_k \cdot \vec{\mathbf{e}}_k}}{NumElts(\mathtt{E})}$

Outline of a practical ICP code

Step 0 : (initialization)

Input surface model `M` and points `Q`.

Build an appropriate data structure (e.g., octree, kD tree) `T` to facilitate finding the closest point matching search.

$n \leftarrow 0$

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

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

$\mathtt{D} \leftarrow \{\cdots, \lVert \vec{\mathbf{c}}_k - \mathbf{F}_0 \bullet \vec{\mathbf{q}}_k \rVert, \cdots\}$

Outline of a practical ICP code

Step 1: (matching)

A $\leftarrow \varnothing$; B $\leftarrow \varnothing$

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

begin

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

// Note: develop first with simple

// search. Later make more

// sophisticated, using T

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

// See also subsequent notes

end

Outline of a practical ICP code

Step 2 : (transformation update)

$n \leftarrow n+1$

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

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

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

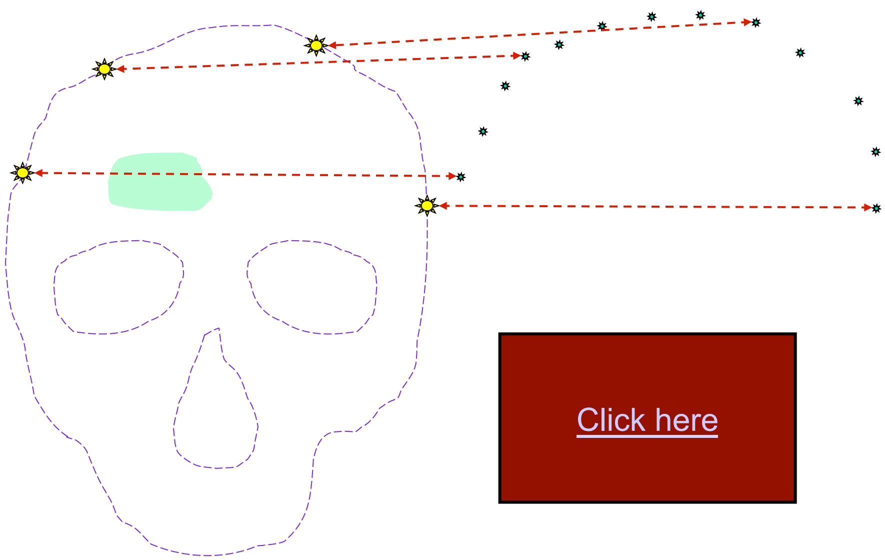
Also, if the mesh is incomplete, it may be advantageous to exclude any matches with triangles at the edge of the mesh.

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.

Digression: Finding Point Pairs



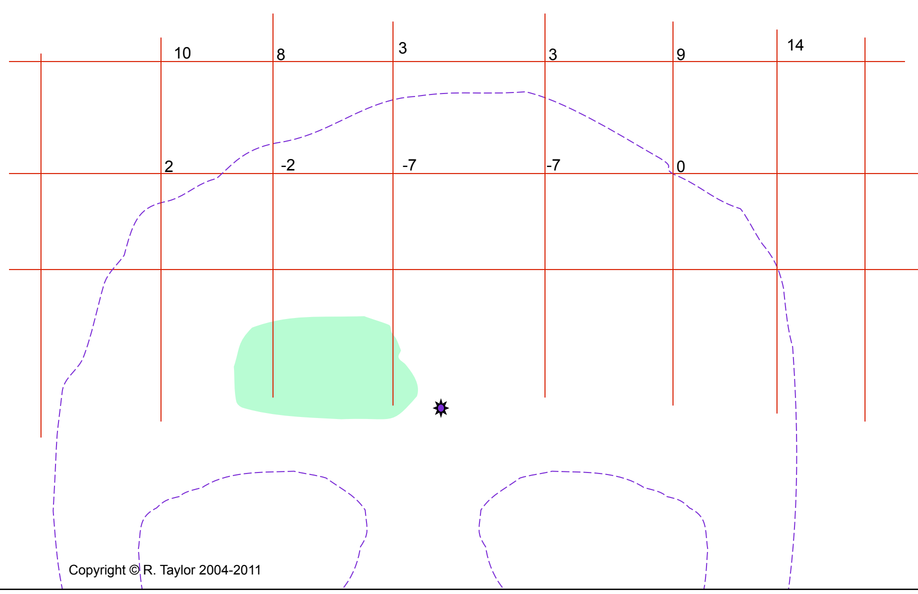
Distance Maps

- 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

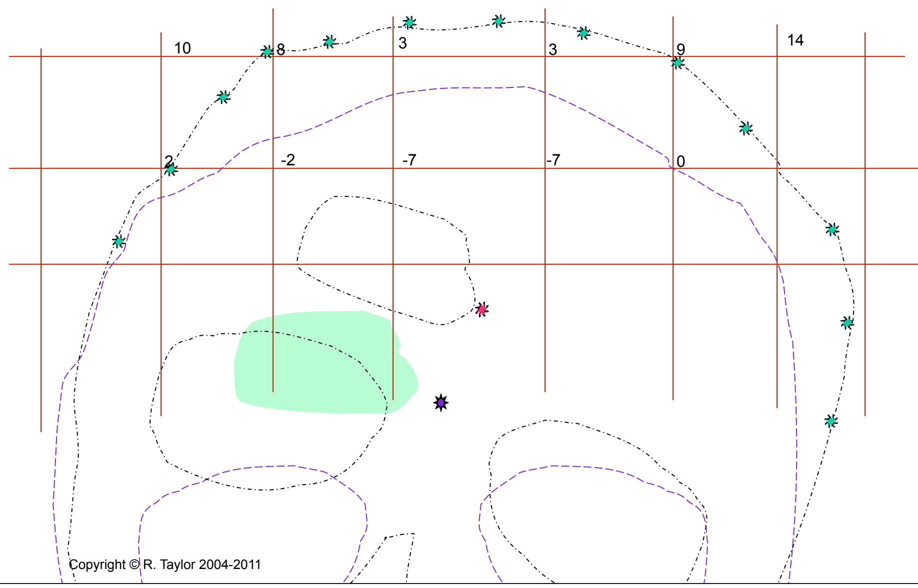
Distance Maps (Continued)

- 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

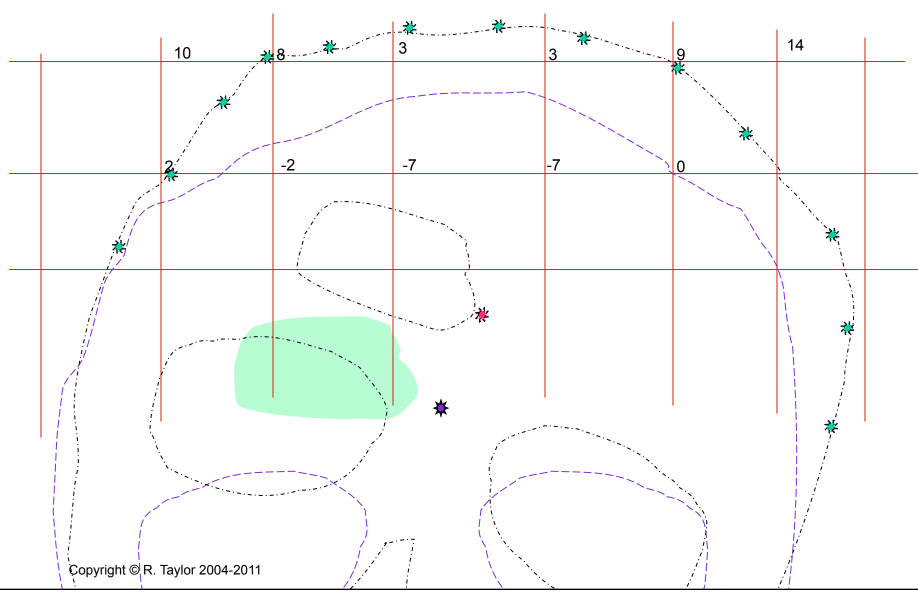
Distance Maps: step 0



Distance Maps: step 1

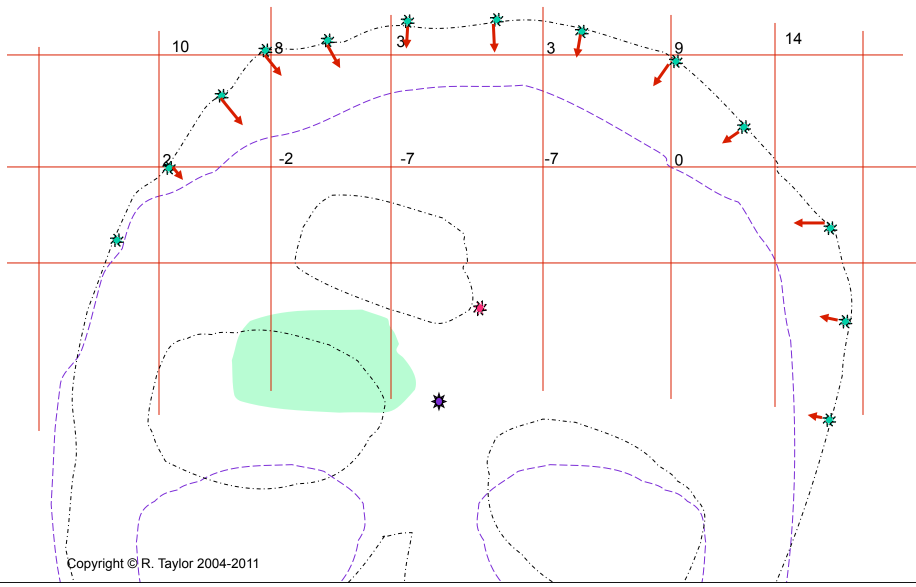


Distance Maps: step 1

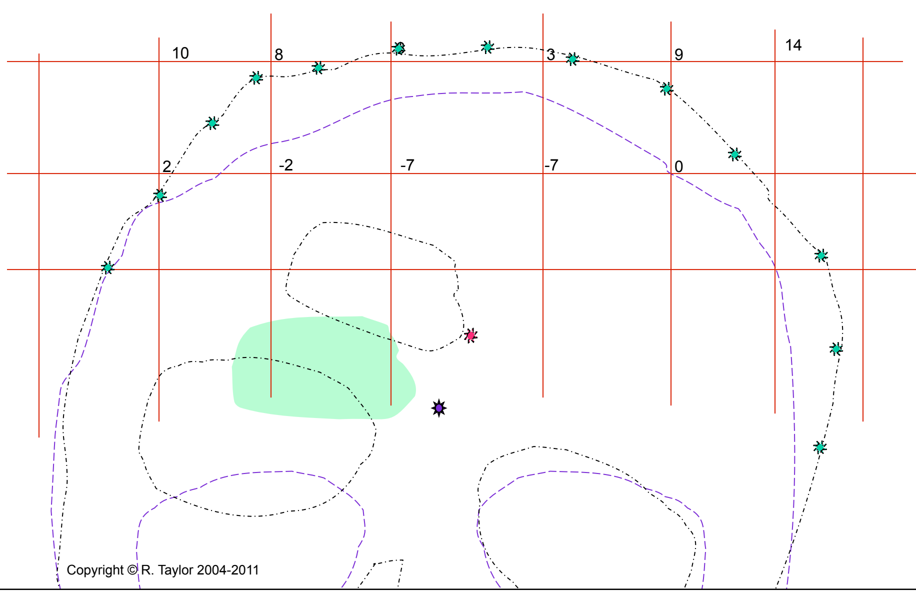


Copyright © R. Taylor 2004-2011

Distance Maps: step 2



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

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

or

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

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

A contour-based 2D-3D method …

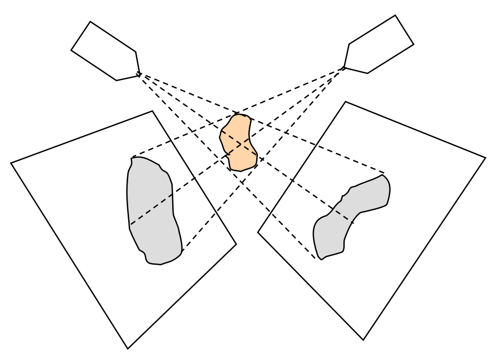
Gueziec *et al., 1998*

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**

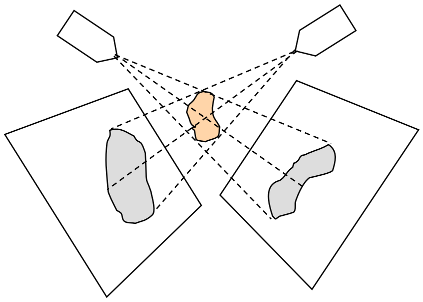


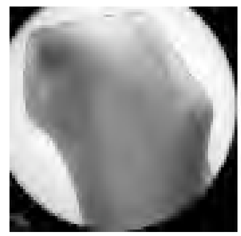
A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

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





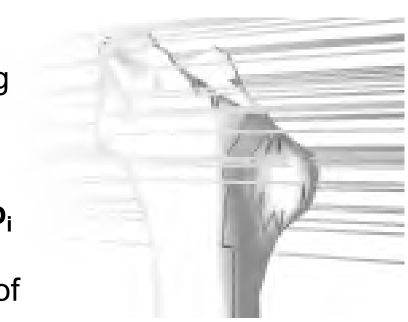
A. Guéziec, P. Kazanzides, B. Williamson, and R. Taylor, "Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot," IEEE Transactions on Medical Imaging, vol. 17, pp. 715-728, 1998.

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.

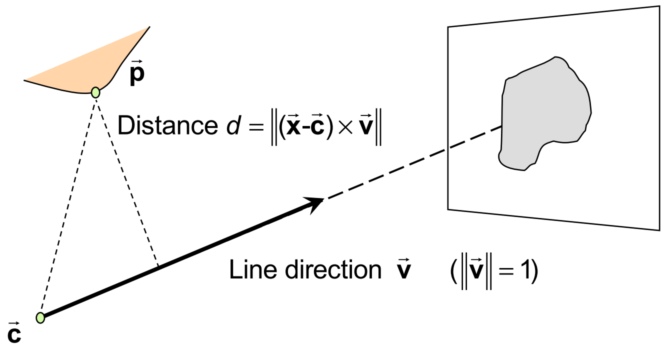
Step 2: For each x-ray path line line $\mathbf{L_i}$, identify the closest point $\mathbf{p_i}$ on an apparent projection contour. This will give a set of points on the body surface to be moved toward the corresponding x-ray lines



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 $\mathbf{p_i}$ and the $\mathbf{L_i}$.

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

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

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.

Copyright © R. Taylor 2004-2011

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:

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

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

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

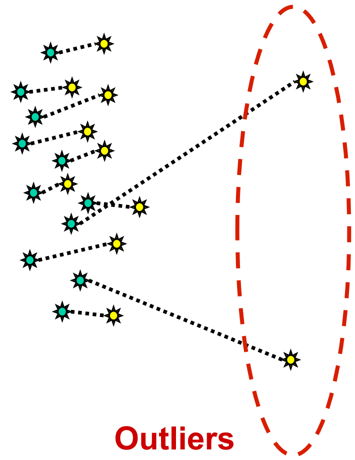
Update $\mathbf{p}_i \leftarrow \Delta\mathbf{R}\mathbf{p}_i + \Delta\vec{\mathbf{t}}$; $\mathbf{R} \leftarrow \Delta\mathbf{R}\mathbf{R}$; $\vec{\mathbf{t}} \leftarrow \Delta\mathbf{R}\vec{\mathbf{t}} + \Delta\vec{\mathbf{t}}$

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 …

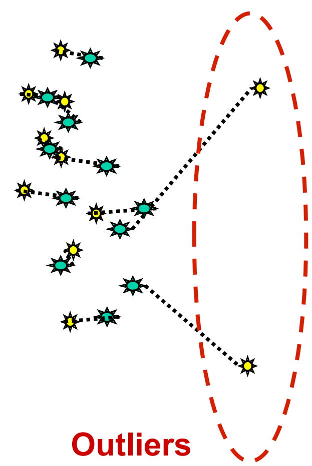
- Basic idea is to identify outliers and give them little or no weight.



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 …

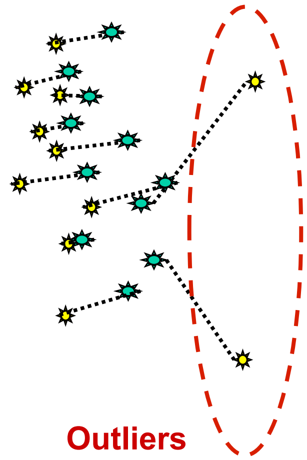
- Basic idea is to identify outliers and give them little or no weight.



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 …

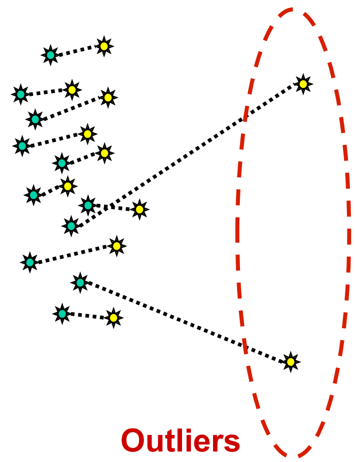
- Basic idea is to identify outliers and give them little or no weight.



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 …

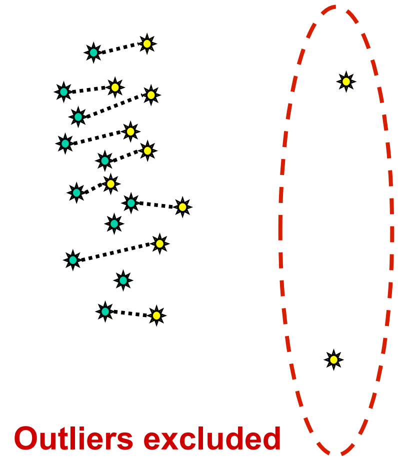
- Basic idea is to identify outliers and give them little or no weight.



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.

Robust Pose Estimation …

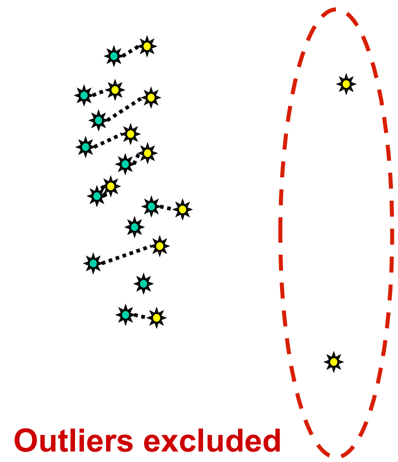
- Basic idea is to identify outliers and give them little or no weight.



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

Robust Pose Estimation …

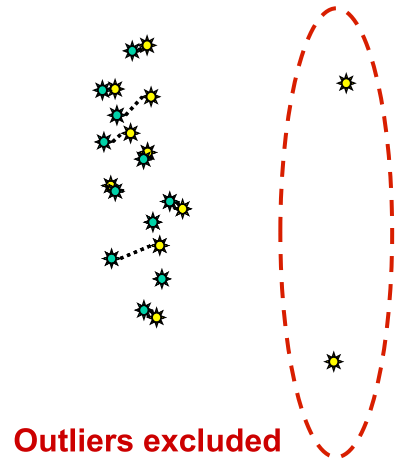
- Basic idea is to identify outliers and give them little or no weight.



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

Robust Pose Estimation …

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



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

Robust M-Estimator …

(Following development in Gueziec et al., 1998)

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

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

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

Step 3: Solve a robust linearized problem

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

(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 $\mathbf{C}\vec{x} = \vec{\mathbf{d}}$, where $\vec{x}^t = [\vec{\mathbf{u}}^t, \vec{\mathbf{t}}^t]$

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

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

(**Note**: We use α=2)

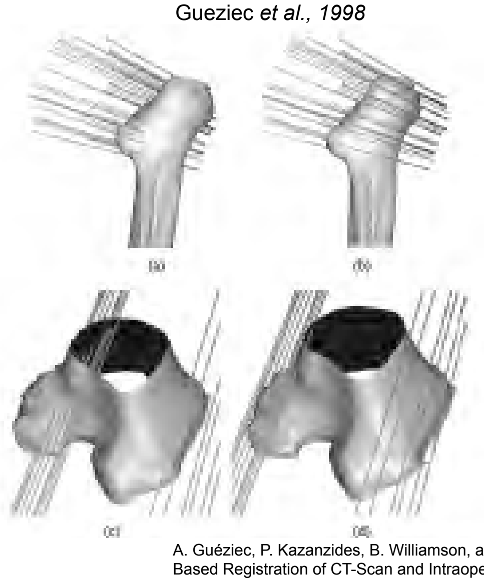
Step 3.3: Iterate steps 3.1 and 3.2 until a suitable termination condition is reached.

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



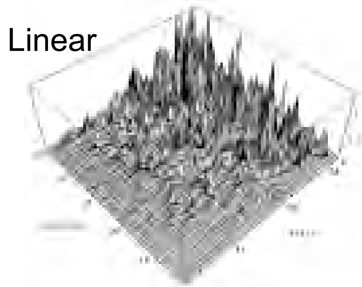
After

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.

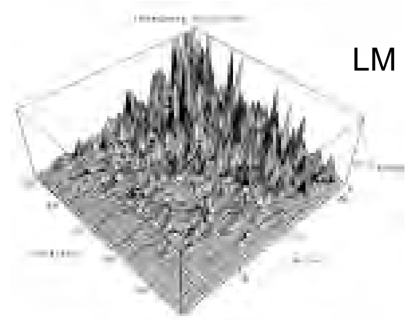
A countour-based 2D-3D method … results

Gueziec *et al., 1998*

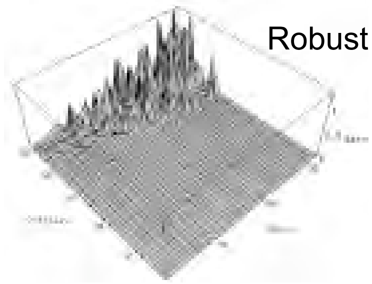
Linear



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 x_s be a point on the surface and $T(\eta)$ represent a small perturbation with parameters η with respect to the surface at point x_s.

$$x'_s = T(\eta)x_s$$

Then we define $V(x_s)$ to be

$$V(x_s) = \frac{\partial D(T(\eta)x_s)}{\partial \eta} = \begin{bmatrix} n_s \\ x_s \times n_s \end{bmatrix}$$

where n_s is the unit normal to the surface at x_s. So,

$$D(T(\eta)x_s) \approx V^T(x_s)\eta$$

Squaring this gives

$$\begin{aligned} D^2(T(\eta)x_s) &\approx \eta^T V(x_s) V^T(x_s)\eta \\ &= \eta^T M(x_s)\eta \end{aligned}$$

Note that M is a 6×6 positive semi-definite, symmetric matrix.

Sample set analysis: sensitivity

For a region R we define

$$\begin{aligned} E(\eta) &= \eta^T \left[\sum_{x_s \in R} M(x_s)\right] \eta \\ &= \eta^T \Psi_R \eta \\ &= \eta^T (Q \Lambda Q^T) \eta \\ &= \sum_{1 \le i \le 6} \lambda_i (\eta^T \cdot q_i)^2 \end{aligned}$$

- Note that the eigenvectors q_i correspond to small differential transformations $T(q_i)$, and can sort eigenvalues so that

$$\lambda_1 \ge \lambda_2 \ge \dots \ge \lambda_6$$

- Note that eigenvector q_1 corresponds to direction of greatest constraint.
- Similarly, can also think of 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 x_k
- Prune for reachability
- For each point, compute constraint direction $\nabla_k = \nabla f(x_k)$. To a first approximation, a measurement at x_k with accuracy ϵ_k constrains η by

$$|\nabla_k \eta| \le \epsilon_k$$

- Now select subset of the x_k that minimizes, e.g.,

$$\min_{\lambda_k} \max \eta^T S \eta$$

subject to

$$\lambda_k \in \{0, 1\}$$
$$\lambda_k \nabla_k \eta \le \epsilon_k$$
$$\sum_k \lambda_k \le \text{maxsize}$$

There are various ways to do this

Sample Set Selection: Another Alternative (con'd)

- [illegible]

[illegible]

[illegible]

- [illegible]

[illegible]

[illegible]

[illegible]

where

[illegible]

[illegible]

[illegible]

Related concept: Estimation with Uncertainty

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

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

Then an appropriate distance metric is the Mahalabonis distance

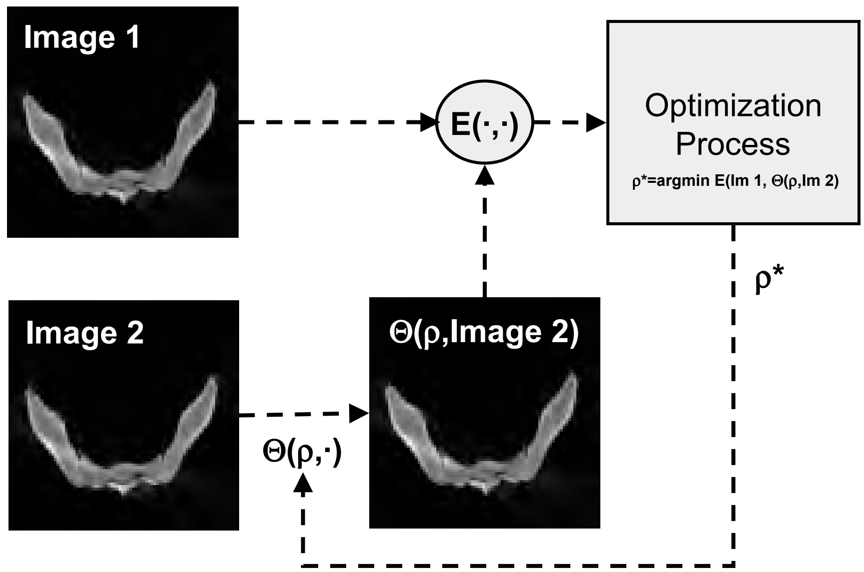
$$\mathrm{D}(\vec{\mathbf{a}}_k, \vec{\mathbf{b}}_k) = (\vec{\mathbf{a}}_k - \vec{\mathbf{b}}_k)^T \mathbf{C}_k^{-1} (\vec{\mathbf{a}}_k - \vec{\mathbf{b}}_k) = \vec{\mathbf{d}}_k^T \Lambda_k^{-1} \vec{\mathbf{d}}_k$$

where

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

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

Intensity-based methods



Intensity-based methods

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

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

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

Typical energy functions
(not an exhaustive list)

Normalized image subtraction

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

Normalized cross correlation

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

Mutual information

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

Mutual Information

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

Mutual Information

Entropy

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

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

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

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

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

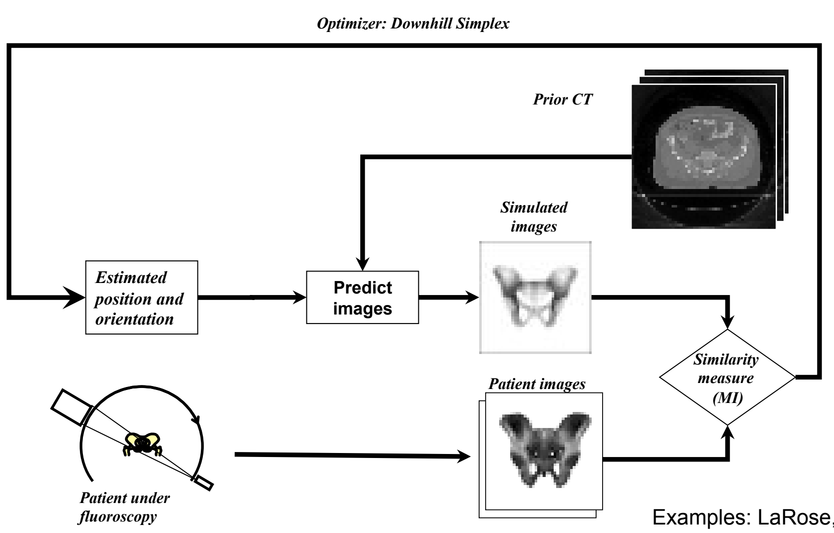
$$\mathit{Similarity}(A,B) = \frac{H(A)+H(B)}{H(A,B)}$$

Objective function

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

Rigid 3D/2D Registration

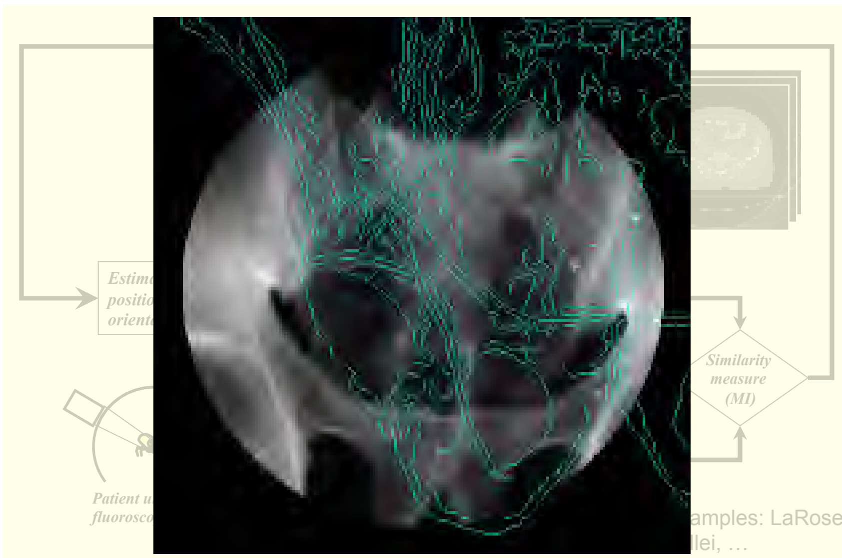
Ofri Sadowsky



Examples: LaRose, Zollei, …

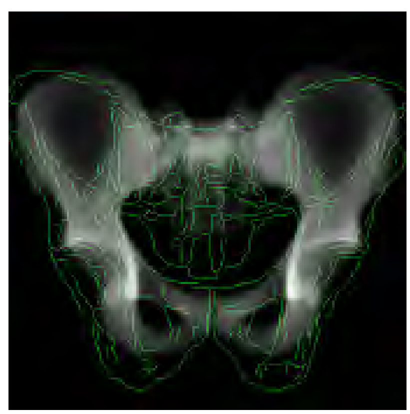
Rigid 3D/2D Registration

Ofri Sadowsky

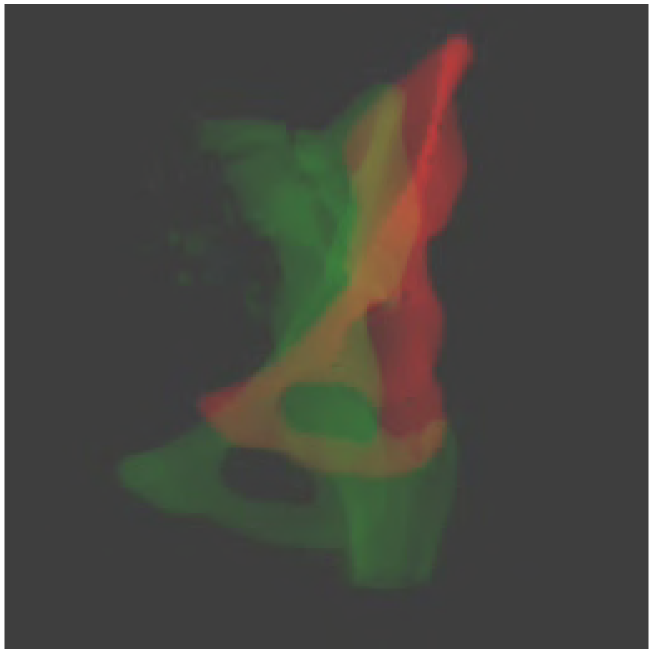


Deformable Registration to Statistical "Atlases"

Deformable 3D/3D
Jianhua Yao



Deformable 2D/3D
Ofri Sadowsky



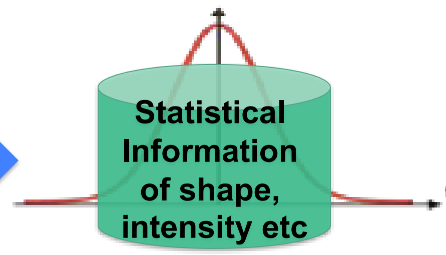
Jianhua Yao

Deformable Altas-based Registration

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

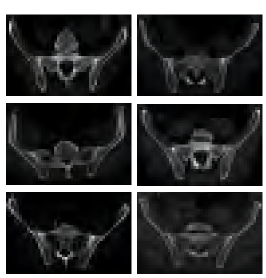
What is a "Statistical Atlas" ?

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

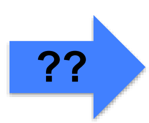


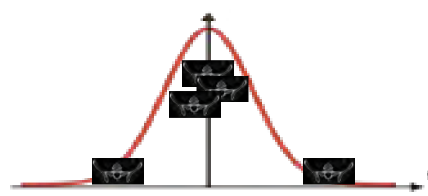
Credit: G. Chintalapani 2010

Statistical Atlases

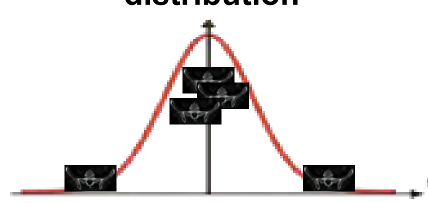


CT scans from a population





Shape distribution



Intensity distribution

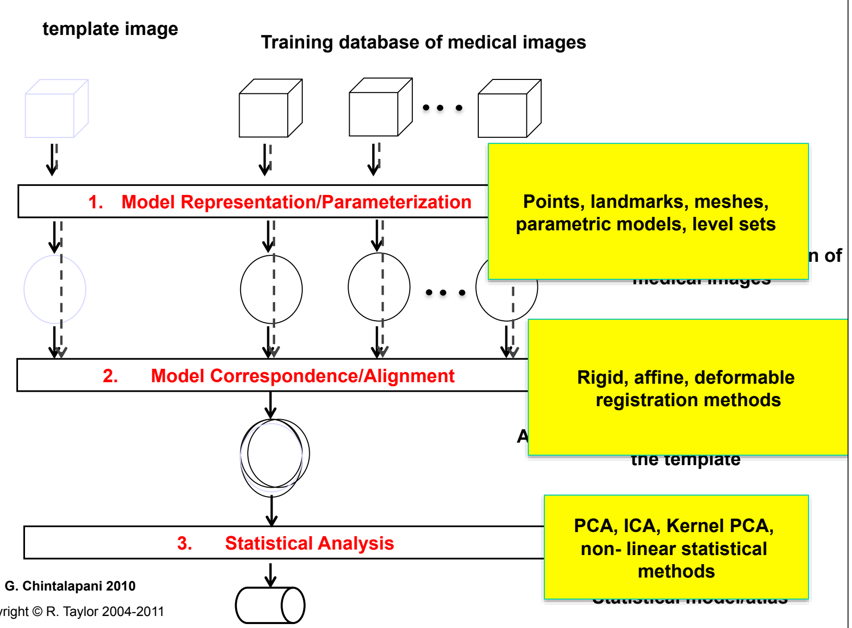
Slide Credit: G. Chintalapani 2010

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”

Atlas Construction



Slide Credit: G. Chintalapani 2010

Model Representation

- Tetrahedral mesh represents shape
- Bernstein polynomials approximate CT density within each tetrahedron[1,2]

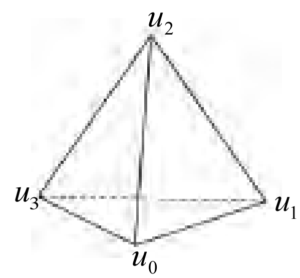
$$P^d(\mathbf{u}) = \sum_{|\mathbf{k}|=d} C_{\mathbf{k}} B^d_{\mathbf{k}}(\mathbf{u})$$

where

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

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

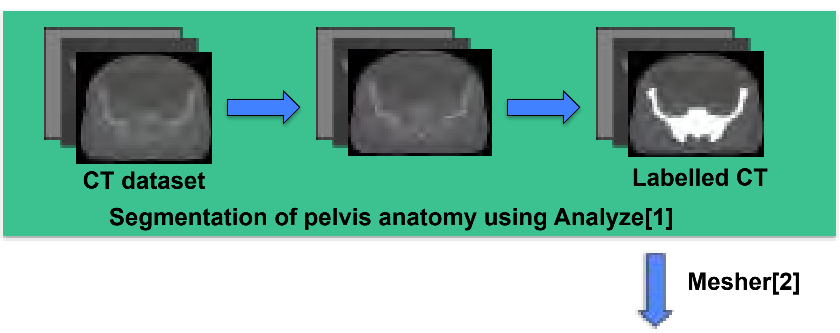
$$B^d_{\mathbf{k}}(\mathbf{u}) = \frac{d!}{k_0!k_1!k_2!k_3!} u_0^{k_0} u_1^{k_1} u_2^{k_2} u_3^{k_3}$$



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

Credit: G. Chintalapani 2010

Model Creation



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

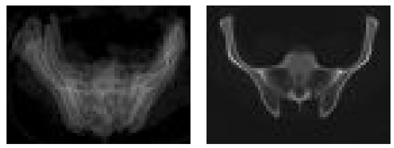
[1]Analyze, www.mayoclinic.org

[2] Mohammed et al., 2005

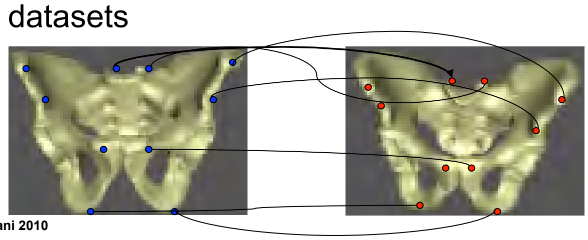
Slide Credit: G. Chintalapani 2010

Model Correspondence

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



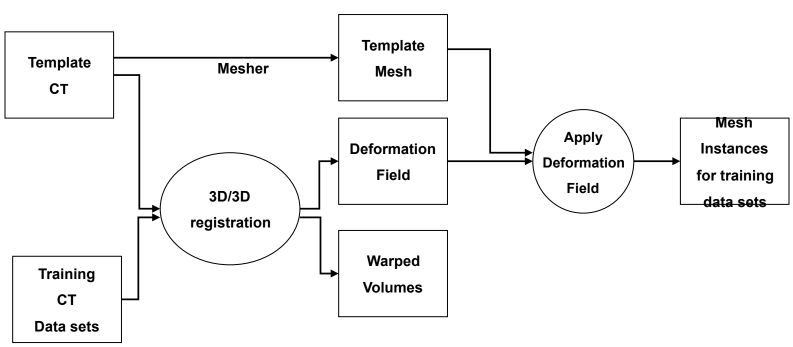
- Need to establish point correspondence between the training datasets



Slide Credit: G. Chintalapani 2010

Model Shape Correspondences

- **Automatic deformable registration based shape correspondences**



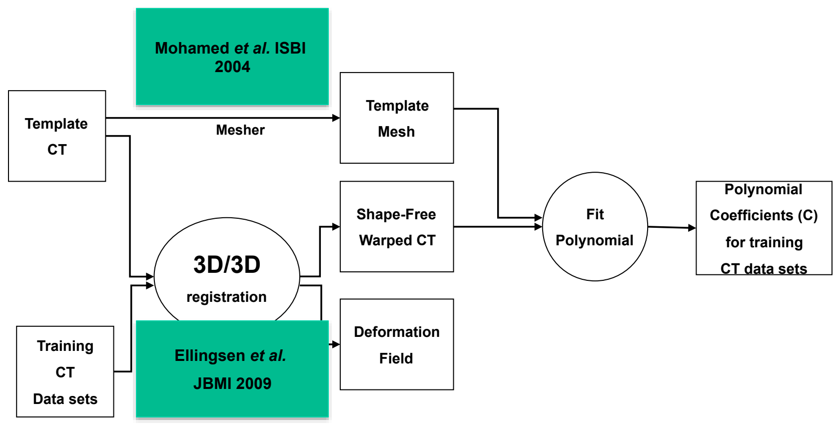
Flowchart for establishing shape correspondences for the training sample

Slide Credit: G. Chintalapani 2010

[1] Rueckert et al., MICCAI 03

Model Intensity Correspondences

- **Automatic deformable registration based correspondences**

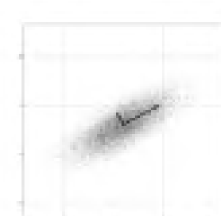


Flowchart for establishing intensity correspondences for the training sample

Principal Component Analysis

- Given the mesh instances of training sample,

$$\not{S} = \begin{bmatrix} \hat{s}_1 & \hat{s}_2 & . & . & \hat{s}_N \end{bmatrix}_{3nXN} = \begin{bmatrix} x_{11} & x_{12} & . & . & x_{1N} \\ y_{11} & y_{12} & . & . & y_{1N} \\ z_{11} & z_{12} & . & . & z_{1N} \\ . & . & . & . & . \\ y_{n1} & y_{n2} & . & . & z_{nN} \\ z_{n1} & z_{n2} & . & . & z_{nN} \end{bmatrix}$$



- Compute mean and subtract the mean from the sample

- Compute

$$\not{S} = S - \bar{s} = S - \frac{1}{N}\sum_{i=1}^{N} \hat{s}_i$$

$$SVD(\not{S}) = UDV^T$$

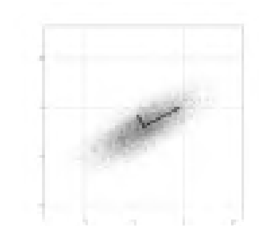
With principal components in U and eigen values $\lambda = \frac{1}{N-1} DD^T$

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

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



- 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

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

- Intensity statistical model: Polynomial coefficients become data matrix

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

Slide Credit: G. Chintalapani 2010

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

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

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

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

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 $\mathbf{U}$. 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

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

we get the expression

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

Statistical Atlases & PCA

Note that while $\mathbf{U}$ is $3N_{vertices} \times 3N_{vertices}$ (i.e., huge), $\mathbf{M}$ 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 $\mathbf{M}$ are also columns of $\mathbf{U}$, they are orthogonal. Hence $\mathbf{M}^T\mathbf{M} = \mathbf{I}_{N \times N}$. But $\mathbf{M}\mathbf{M}^T = \mathbf{C}$ will be an $3N_{vertices} \times 3N_{vertices}$ matrix that will not in general be diagonal.

Statistical Atlases & PCA

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

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

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

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

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

Statistical Atlases & PCA

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

$$\vec{\mathbf{D}}^{(\text{inst})} = \mathbf{M}\vec{\lambda}^{(\text{inst})}$$

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

$$= \vec{\lambda}^{(inst)}$$

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

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

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

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

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 **V** 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.

Deformable Registration Between 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

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) methods 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

Jianhua Yao

Energy Function

- To measure the density and shape difference between model and image

$$E(mdl,img) = w_s E^{(s)}(mdl,img) + w_d E^{(d)}(mdl,img)$$

$$E^{(s)}(mdl,img) = \sum_{i=1}^{N(v)} \left(\vec{g}^{(mdl)}(v_i) \cdot \vec{g}^{(img)}(v_i) \right)$$

$$E^{(d)}(mdl,img) = \sum_{i=1}^{N(t)} \left(\oint_{\mu} \left(\frac{d^{(mdl)}(t_i,\mu) - d^{(img)}(t_i,\mu)}{d^{(mdl)}(t_i,\mu)} \right)^2 \right)$$

Jianhua Yao

Local Deformation

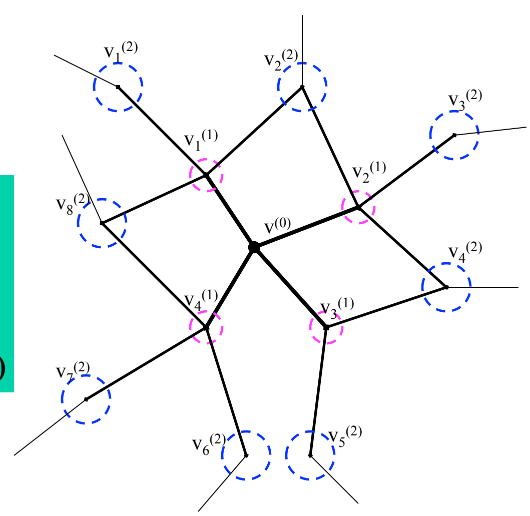
- Motivation: Statistical deformation can't capture all the variability due to the limited number of models in the training set
- Locally adjust the location of vertices to match the boundary of the bone and the interior density property
- Use multiple-layer flexible mesh template matching to find the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

Jianhua Yao

Multiple-layer Flexible Mesh Template

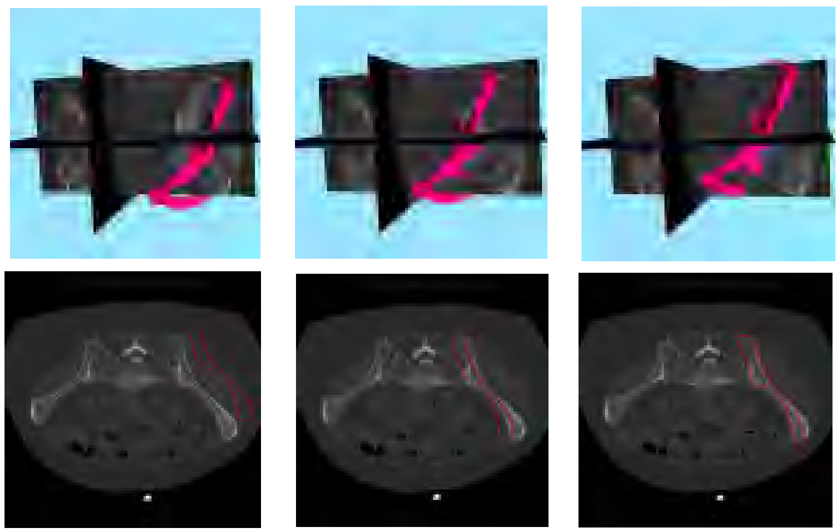
- Each vertex on the model defines a mesh template
- Template is in the form

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



Jianhua Yao

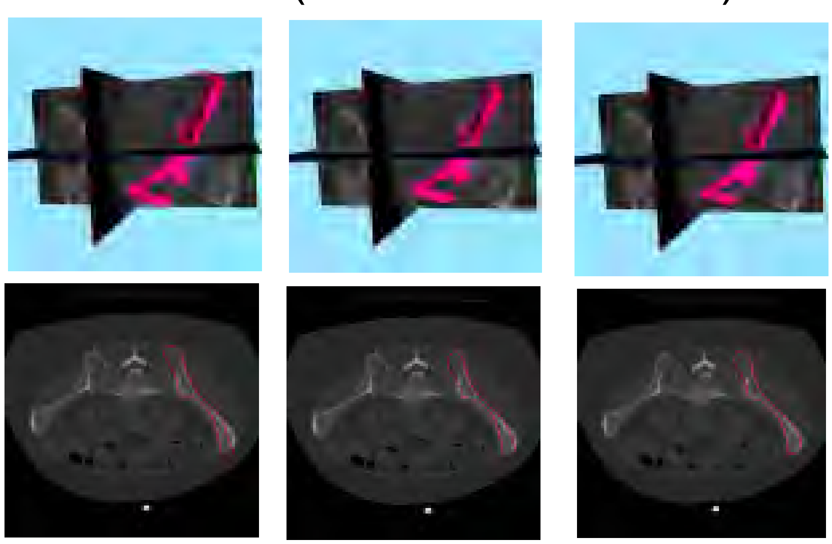
Results (Affine Transformation)



Initial Intermediate Final

Jianhua Yao

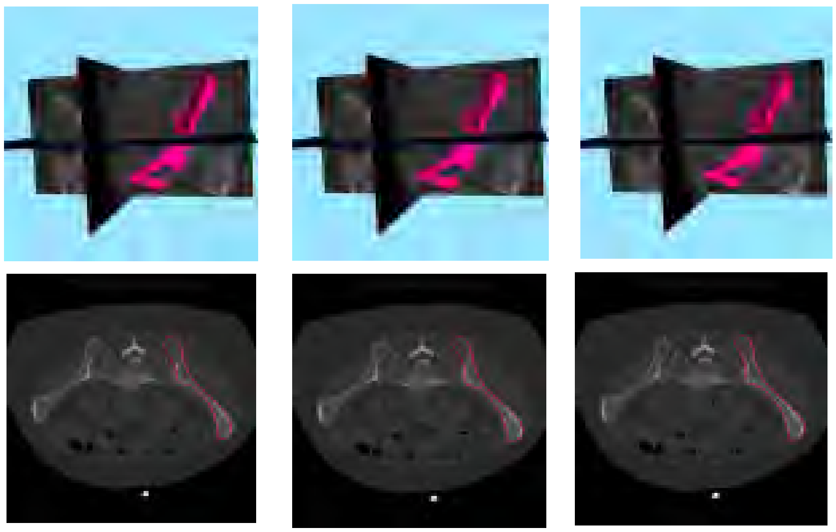
Results (Global Deformation)



Initial Intermediate Final

Jianhua Yao

Results (Local Deformation)



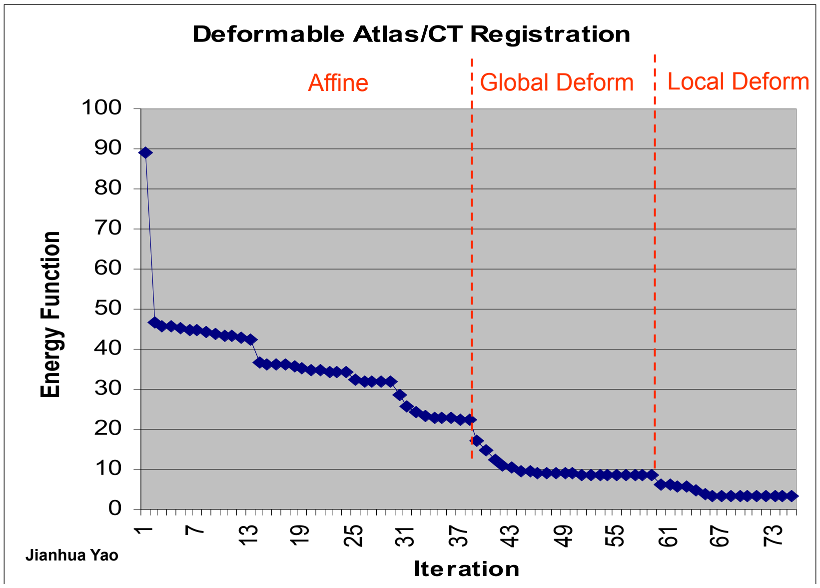
Initial Intermediate Final

Jianhua Yao

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

Jianhua Yao

Results (Deformable Registration)



Jianhua Yao

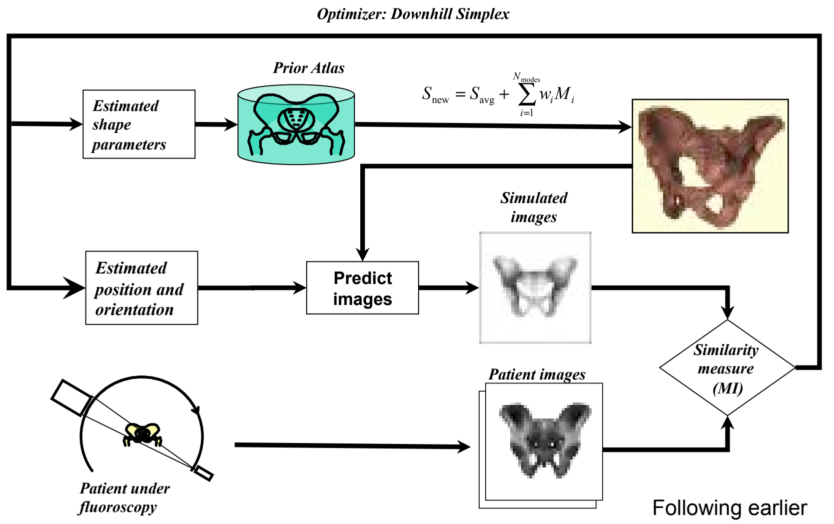
Deformable registration between density atlas and a set of 2D X-Rays

- 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

Deformable 3D/2D Registration

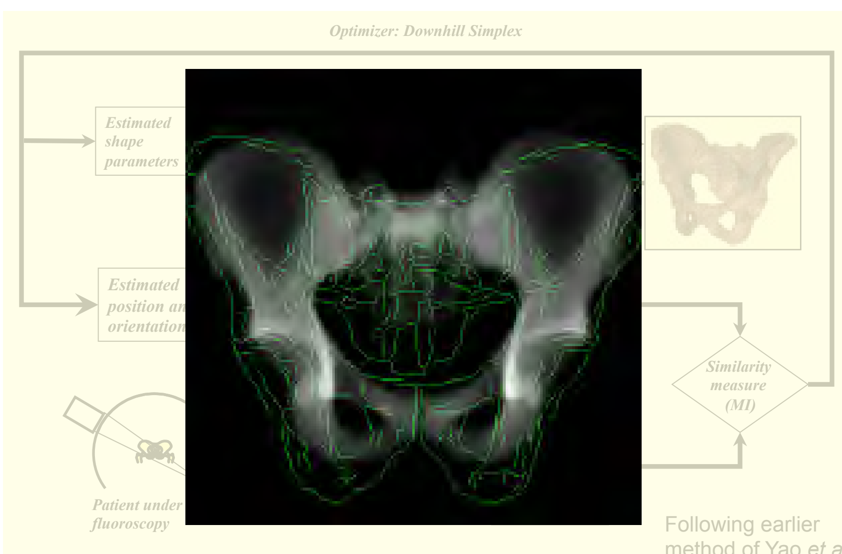
Ofri Sadowsky



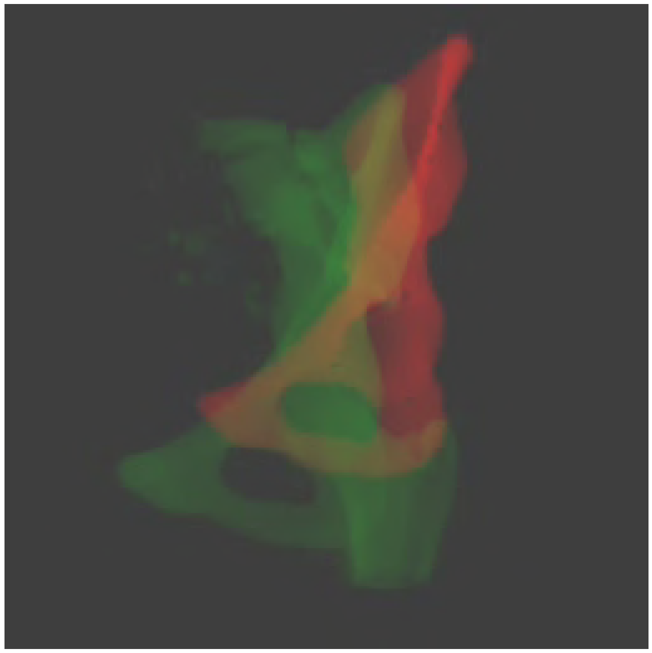
Following earlier method of Yao *et al.*

Deformable 3D/2D Registration

Ofri Sadowsky



Following earlier method of Yao *et al.*



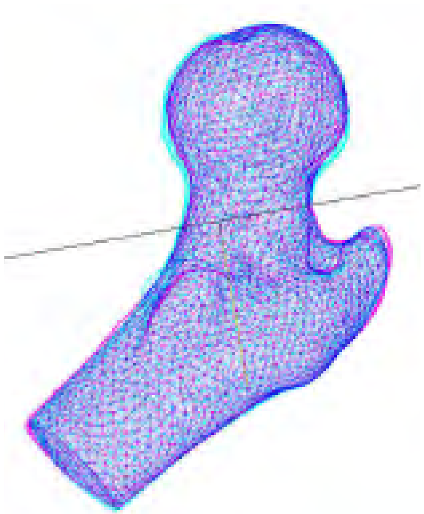
Jianhua Yao

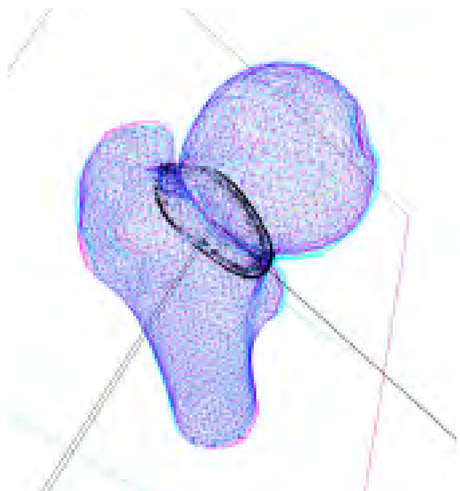
2D-3D Reconstruction from 3 DEXA Images

Omar Ahmad, *et al.*

Hologic, Inc.

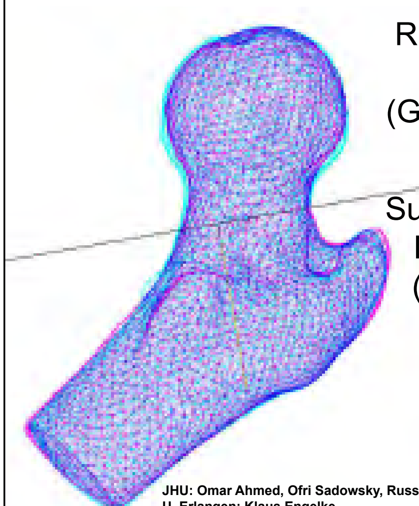
2D-3D Reconstruction from 3 DEXA Images+Atlas





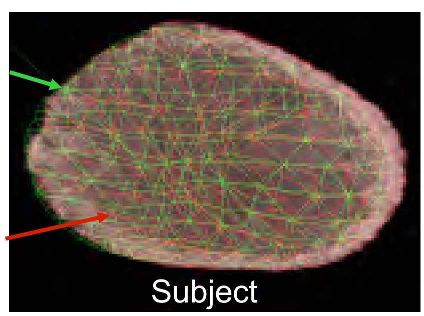
JHU: Omar Ahmed, Ofri Sadowsky, Russell Taylor
U. Erlangen: Klaus Engelke
BIDMC, Boston: Mary Bouxsein
Inst. of Aging Research, Boston: David Karasik
Hologic, Inc.: Krishna Ramamurthi, Kevin Wilson

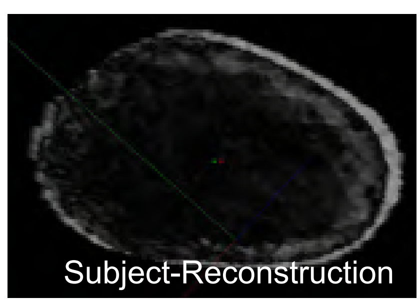
2D-3D Reconstruction from 3 DEXA Images+Atlas



Recon. Mesh (Green)

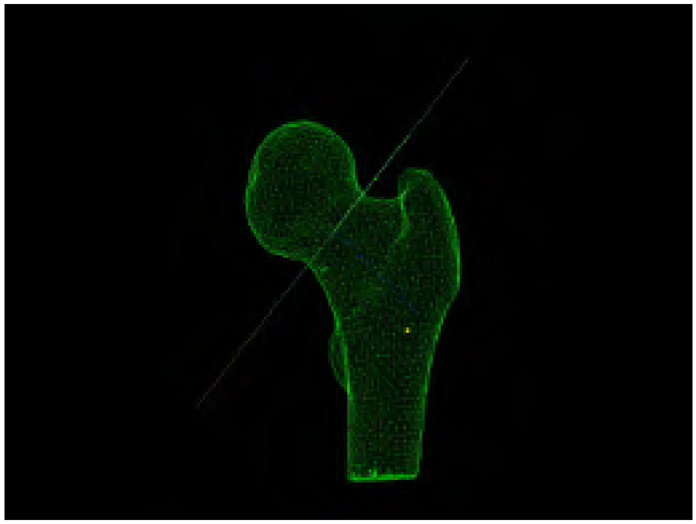
Subject Mesh (Red)





JHU: Omar Ahmed, Ofri Sadowsky, Russell Taylor
U. Erlangen: Klaus Engelke
BIDMC, Boston: Mary Bouxsein
Inst. of Aging Research, Boston: David Karasik
Hologic, Inc.: Krishna Ramamurthi, Kevin Wilson

Femur model from three 2D DXA images



JHU: Omar Ahmed, Ofri Sadowsky, Russell Taylor
U. Erlangen: Klaus Engelke
BIDMC, Boston: Mary Bouxsein
Inst. of Aging Research, Boston: David Karasik
Hologic, Inc.: Krishna Ramamurthi, Kevin Wilson