NSF Engineering Research Center for Computer Integrated Surgical Systems and Technology

LABORATORY FOR Computational Sensing + Robotics THE JOHNS HOPKINS UNIVERSITY

Registration – Part 3

600.455/655 Computer Integrated Surgery

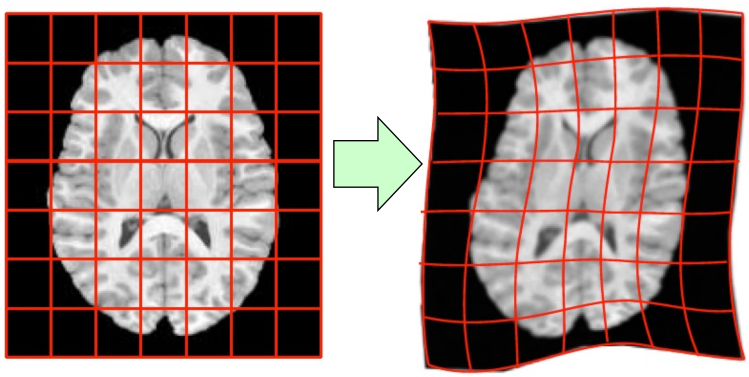
Russell H. Taylor
John C. Malone Professor of Computer Science,
with joint appointments in Mechanical Engineering, Radiology & Surgery
Director, Laboratory for Computational Sensing and Robotics
The Johns Hopkins University
rht@jhu.edu

100 YEARS JOHNS HOPKINS ENGINEERING

WHITING SCHOOL OF ENGINEERING
THE JOHNS HOPKINS UNIVERSITY

1

Deformable Registration



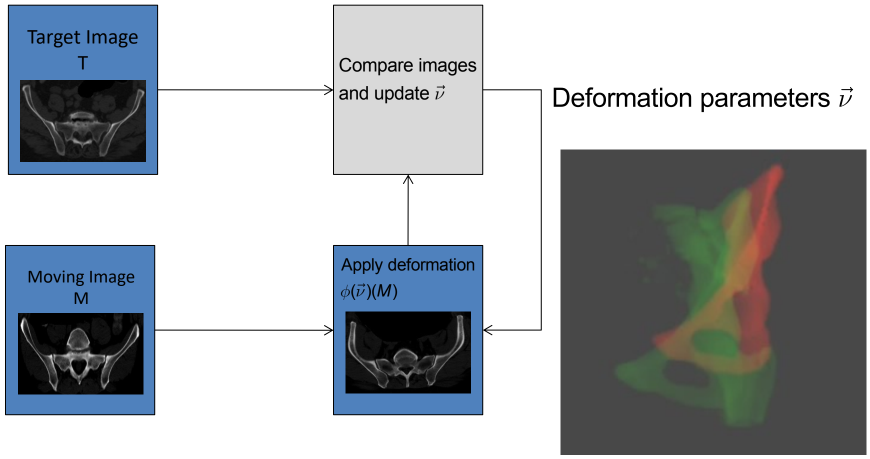
$\mathrm{Im}(\vec{\mathbf{x}})$ $\mathrm{Im}(\Phi(\vec{\rho},\vec{\mathbf{x}}))$

- 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 $\Phi$is bijective and both Φ and Φ^{-1} are smooth

Images: Tom Fletcher

2

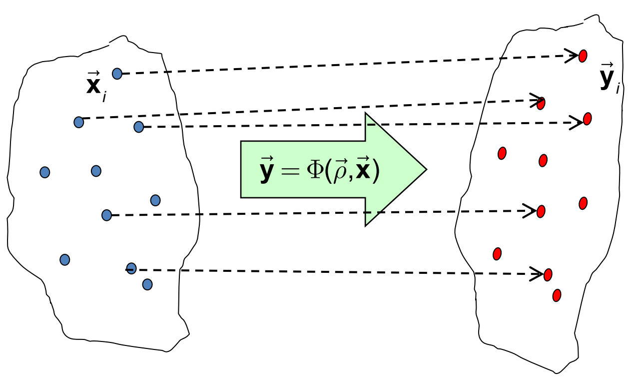
Deformable Registration



Engineering Research Center for Computer Integrated Surgical Systems and Technology

3

Deformable Registration from Point Cloud Matches



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

Images: Tom Fletcher

Engineering Research Center for Computer Integrated Surgical Systems and Technology

4

Deformable warping from point cloud matches

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

$$\vec{\mathbf{u}} = TrimToBox(\vec{\mathbf{x}})$$
$$\vec{\mathbf{y}} = \sum_{i,j,k} \vec{\mathbf{c}}_{i,j,k} B_i(u_x) B_j(u_y) B_k(u_z)$$
or
$$\vec{\mathbf{y}} = \sum_{i,j,k} \vec{\mathbf{c}}_{i,j,k} N_i(u_x) N_j(u_y) N_k(u_z)$$

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

5

Radial Basis Functions

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

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

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

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

6

Thin Plate Splines

- Minimum energy spline deformations

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

where $U(r) = r^2 \log(r)$ for 2D images

- Global support
- Popularized by Fred Bookstein for analysis of anatomic variation
  - F. L. Bookstein, *Morphometric tools for landmark data*, Geometry and biology: Cambridge University Press, 1991.

Thin Plate Splines Digression

- Some citations (from G. Donato and S. Belongie, “Approximation Methods for Thin Plate Spline Mappings and Principal Warps”, 2002; http://www.cs.ucsd.edu/Dienst/UI/2.0/Describe/ncstrl.ucsd_cse/CS2003-0764)

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

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

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

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

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

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

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

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

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

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

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

M-dimensional Thin Plate Spline Summary

Given

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

where

$$U(r) = r^2 \log(r) \quad \text{for 2D}$$
$$= r^2 \log(r^2) \quad \text{for 3D}$$
$$\vec{\mathbf{v}} = [v_1,\cdots,v_M]^T$$
$$\vec{\mathbf{p}}_i = [p_1,\cdots,p_M]_i^T$$
$$\mathbf{P} = [\vec{\mathbf{p}}_1,\cdots,\vec{\mathbf{p}}_N]^T$$
$$\mathbf{C} = [\vec{\mathbf{c}}_1,\cdots,\vec{\mathbf{c}}_N]$$
$$\mathbf{B} = [\vec{\mathbf{b}}_1,\cdots,\vec{\mathbf{b}}_M]$$

Note: Some sources give

$$U(r) = \begin{cases} r^{4-m}\ln(r) & \text{for m=2 or 4} \\ r^{4-m} & \text{otherwise} \end{cases}$$

10

M-dimensional Thin Plate Spline Fitting

Given

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

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

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

To do this, solve the linear system

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

where

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

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

11

TPS 2D case

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

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

To do this, we solve the least squares problem

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

where $\mathrm{U}_{i,j}=\mathrm{U}_{j,i}=\mathrm{U}(\left\|\vec{\mathbf{p}}_i - \vec{\mathbf{p}}_j\right\|)$

12

M-dimensional Thin Plate Spline Fitting

Define

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

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

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

See

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

13

Other Radial Basis Functions

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

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

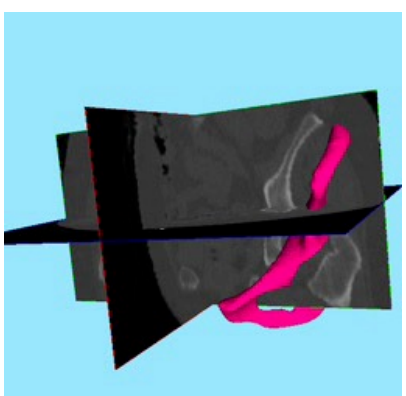
The last one is probably the most popular for global support. There are also radial basis functions with "compact" support. For example*

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

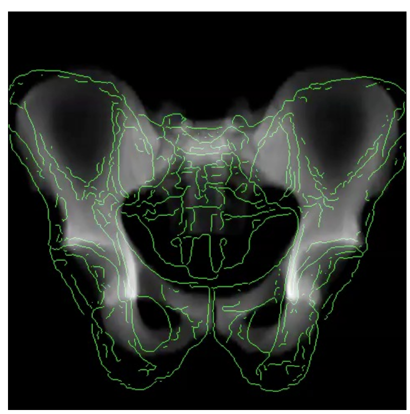
* See: M. Fornefett, K. Rohr, and H. S. Stiehl, "Radial basis functions with compact support for elastic registration of medical images", *Image and Vision Computing,* vol. 19- 1,Äì2, pp. 87-96, 2001. http://www.sciencedirect.com/science/article/pii/S0262885600000573 http://dx.doi.org/10.1016/S0262-8856(00)00057-3

14

Deformable Registration to Statistical "Atlases"



Deformable 3D/3D
Jianhua Yao



Deformable 2D/3D
Ofri Sadowsky

15

Deformable Altas-based Registration

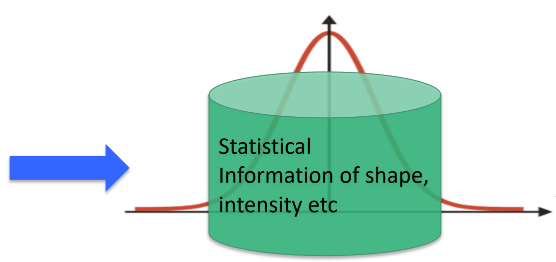
- Much of the material that follows is derived from the Ph.D. thesis work of J. Yao, Ofri Sadowsky, and Gouthami Chintalapani:
  - J. Yao, "Statistical bone density atlases and deformable medical image registrations", Ph. D. Thesis, Computer Science, The Johns Hopkins University, Baltimore, 2001.
  - 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

Engineering Research Center for Computer Integrated Surgical Systems and Technology

16

What is a "Statistical Atlas" ?

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

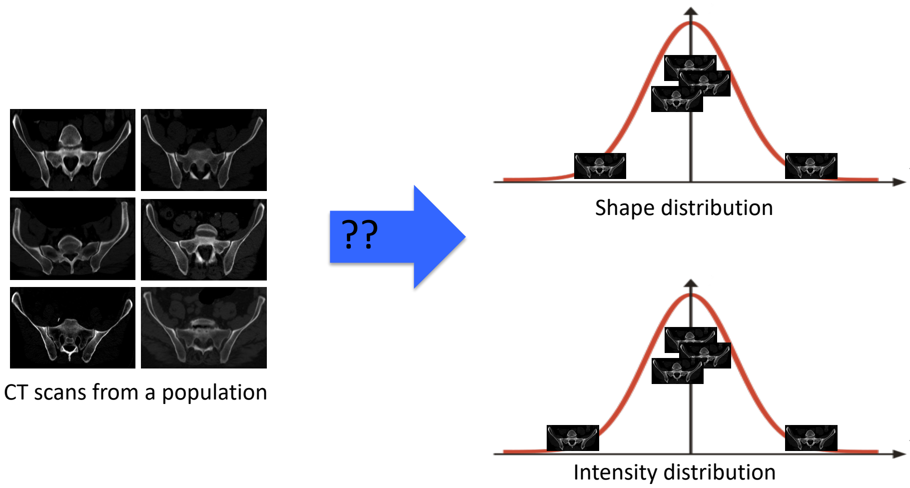


Credit: G. Chintalapani 2010

Engineering Research Center for Computer Integrated Surgical Systems and Technology

17

Statistical Atlases



Slide Credit: G. Chintalapani 2010

Engineering Research Center for Computer Integrated Surgical Systems and Technology

18

Statistical models

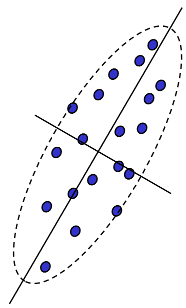
- The next few slides will review the use of the Singular Value Decomposition (SVD) in constructing statistical shape models.
- There is a close relationship between this material and the "principal components analysis" (PCA) methods you may have encountered in a statistics class.

Engineering Research Center for Computer Integrated Surgical Systems and Technology

19

Principal Components Analysis (PCA)

Suppose that you have a set of N vectors $\vec{\mathbf{a}}_i$ in an M dimensional space?
Is there a natural "coordinate system" for these vectors?



Principal Components Analysis (PCA)

We proceed as follows

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

Then form the singular value decomposition

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

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

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

Principal Components Analysis (PCA)

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

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

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

where

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

So

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

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

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

Engineering Research Center for Computer Integrated Surgical Systems and Technology

22

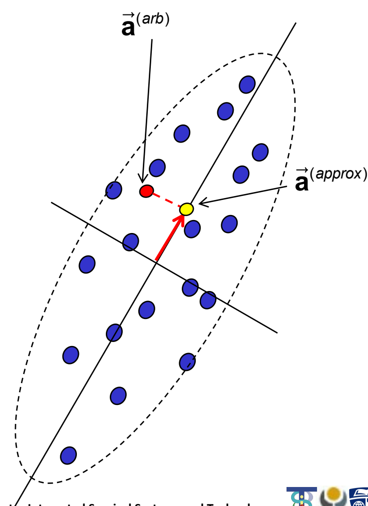
Principal Components Analysis (PCA)

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

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

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

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



Engineering Research Center for Computer Integrated Surgical Systems and Technology

23

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

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

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

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

24

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

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

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

25

Statistical Atlases & PCA

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

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

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

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

Engineering Research Center for Computer Integrated Surgical Systems and Technology

26

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.

Engineering Research Center for Computer Integrated Surgical Systems and Technology

27

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**. Many 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

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

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

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

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

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

32

Statistical Atlas

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

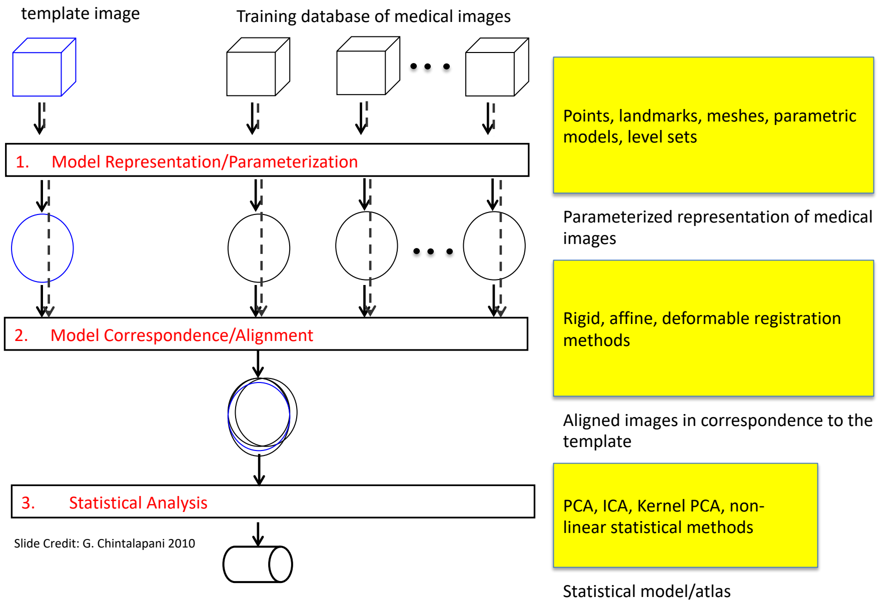
- An average model $\vec{\mathbf{X}}^{(avg)}$
- An eigen matrix **M** of variation modes
- A diagonal covariance matrix Σ^2 for the modes

This information may be used in many ways, including

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

33

Statistical Atlas Construction



Model Representation

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

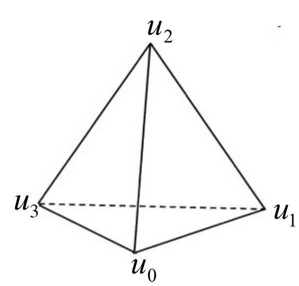
$$P^d(\mathbf{u}) = \sum_{|\mathbf{k}|=d} C_{\mathbf{k}} B^d_{\mathbf{k}}(\mathbf{u})$$

where

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

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

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

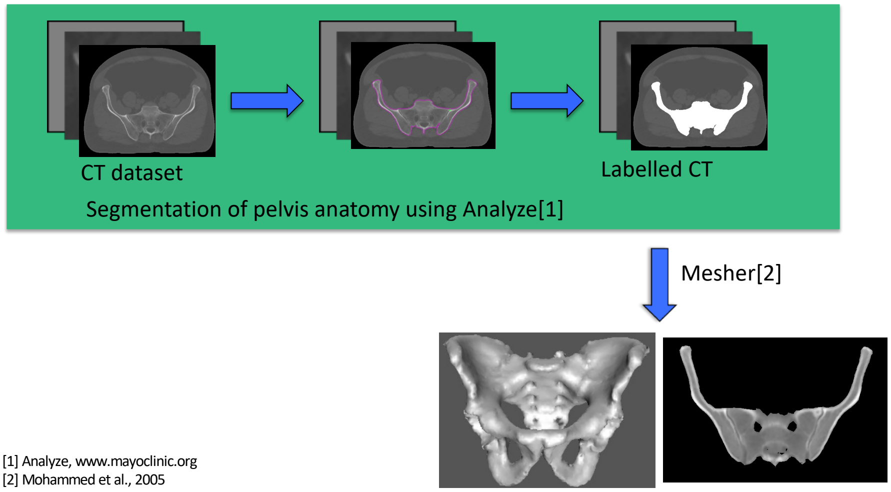


- Alternative might be to use voxels directly after deformation to mean shape

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

Credit: G. Chintalapani 2010

Model Creation



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

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

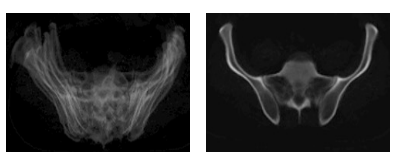
Slide Credit: G. Chintalapani 2010

Engineering Research Center for Computer Integrated Surgical Systems and Technology

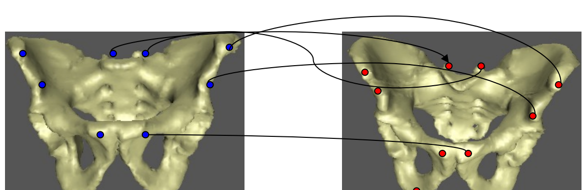
36

Model Correspondence

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



- Need to establish point correspondence between the training datasets



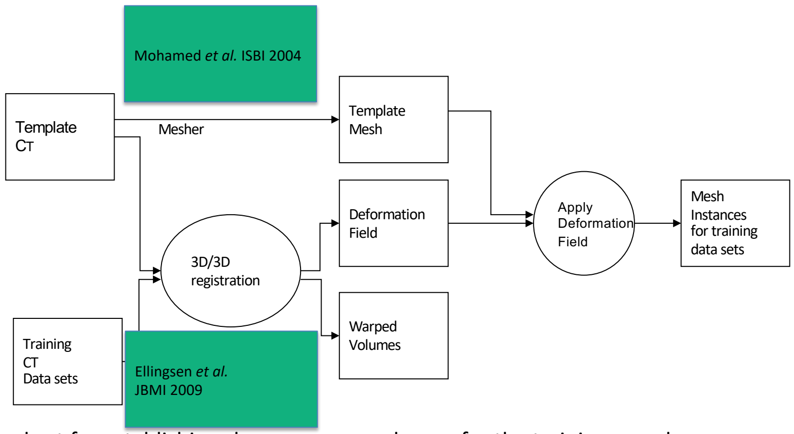
Slide Credit: G. Chintalapani 2010

Engineering Research Center for Computer Integrated Surgical Systems and Technology

37

Model Shape Correspondences

- **Automatic deformable registration based shape correspondences**



Flowchart for establishing shape correspondences for the training sample

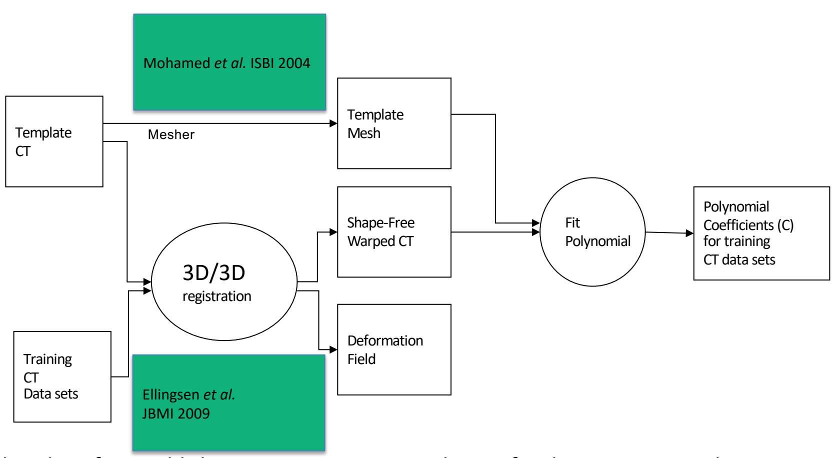
Slide Credit: G. Chintalapani 2010

[1] Rueckert et al, MICCAI

38

Model Intensity Correspondences

- **Automatic deformable registration based correspondences**



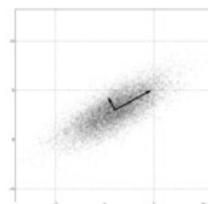
Flowchart for establishing intensity correspondences for the training sample

39

Principal Component Analysis

- Given the mesh instances of training sample,

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



- Compute mean and subtract the mean from the sample

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

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

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

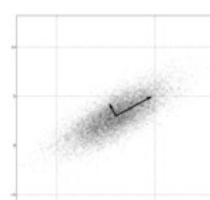
Slide Credit: G. Chintalapani 2010

40

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

41

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

Engineering Research Center for Computer Integrated Surgical Systems and Technology

42

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

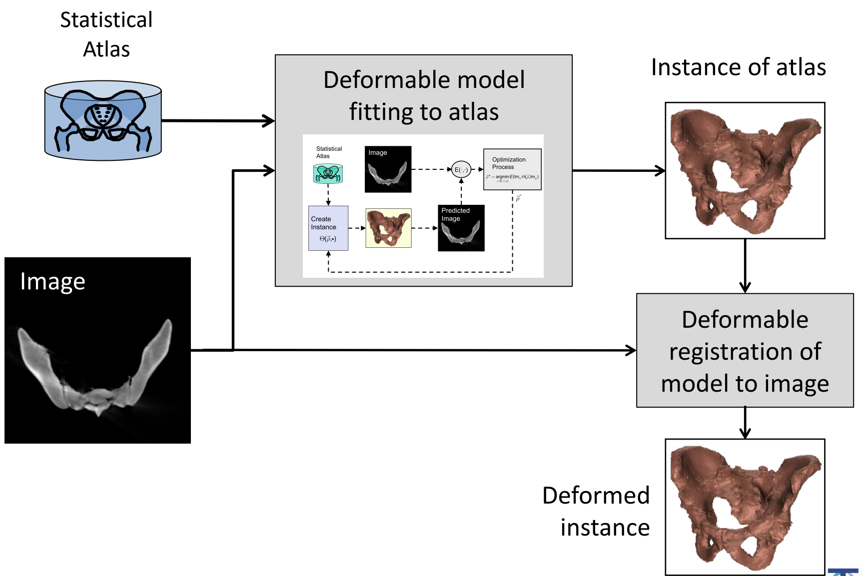
Jianhua Yao

Engineering Research Center for Computer Integrated Surgical Systems and Technology

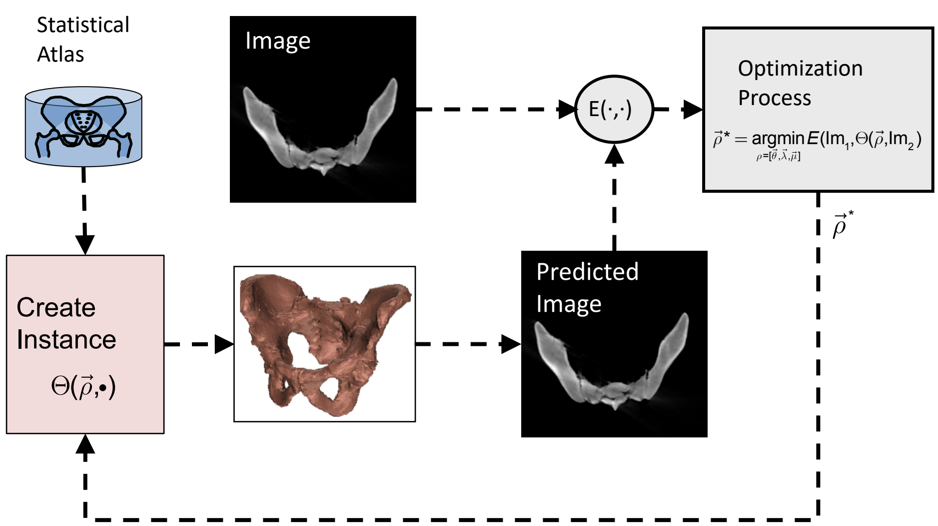
Slide credit: Gouthami Chintalapani

43

Typical pipeline for atlas-assisted registration/registration



Deformable model fitting



Deformable Registration Scheme

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

Jianhua Yao

46

Optimization Algorithm

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

Jianhua Yao

47

Local Deformation

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

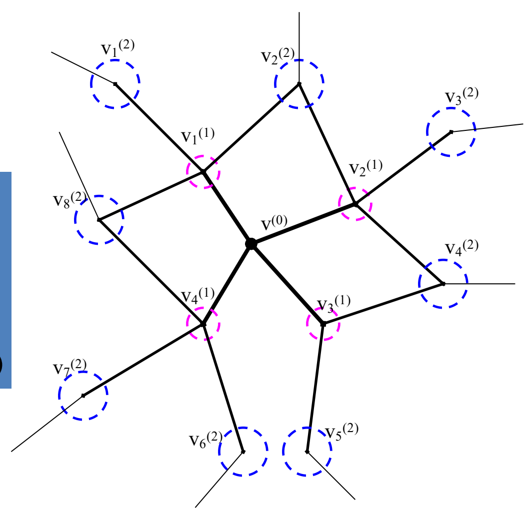
Jianhua Yao

50

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
Copyright 2021 R. H. Taylor
Engineering Research Center for Computer Integrated Surgical Systems and Technology

51

Template matching

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

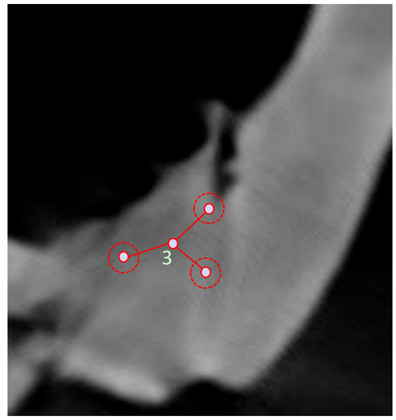
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

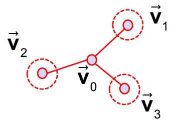
For each neighbor $\vec{\mathbf{v}}_k$

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





Engineering Research Center for Computer Integrated Surgical Systems and Technology

52

Template matching

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

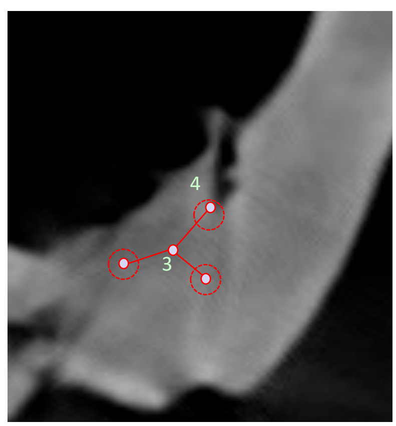
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

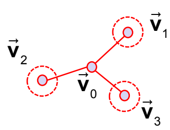
For each neighbor $\vec{\mathbf{v}}_k$

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





Engineering Research Center for Computer Integrated Surgical Systems and Technology

53

Template matching

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

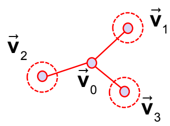
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

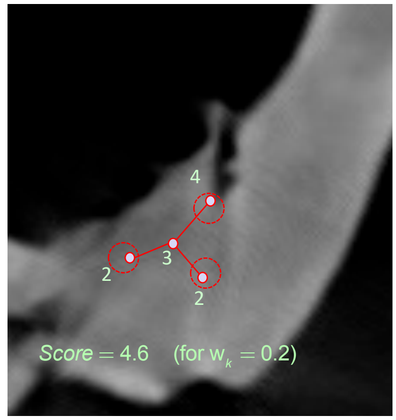
For each neighbor $\vec{\mathbf{v}}_k$

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





Engineering Research Center for Computer Integrated Surgical Systems and Technology

54

Template matching

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

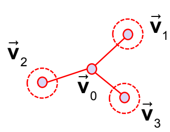
Place $\vec{\mathbf{v}}_0$ at $\vec{\mathbf{x}}_0$

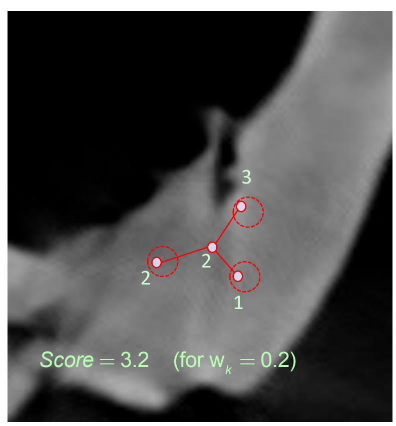
For each neighbor $\vec{\mathbf{v}}_k$

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

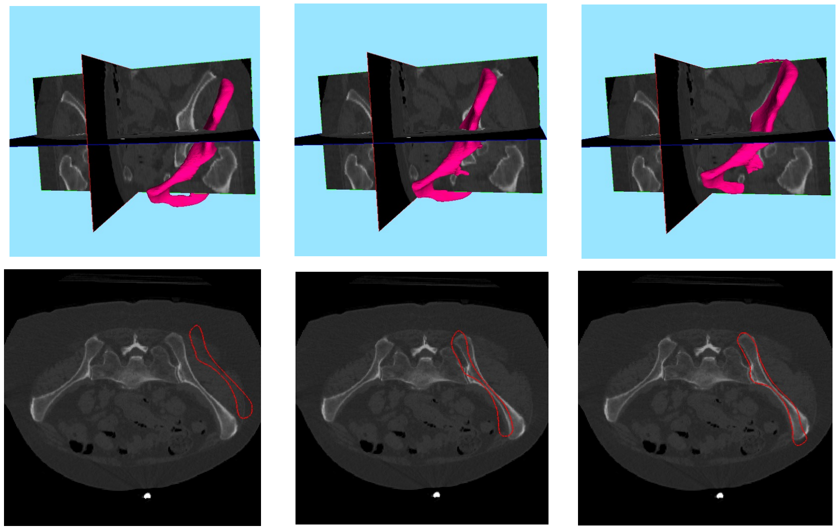




Engineering Research Center for Computer Integrated Surgical Systems and Technology

55

Results (Affine Transformation)



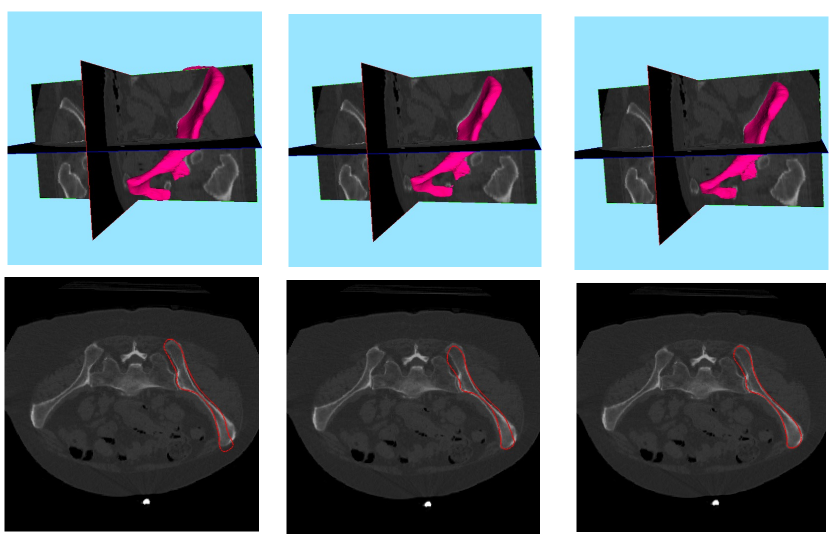
Initial | Intermediate | Final

Jianhua Yao

Engineering Research Center for Computer Integrated Surgical Systems and Technology

56

Results (Global Deformation)



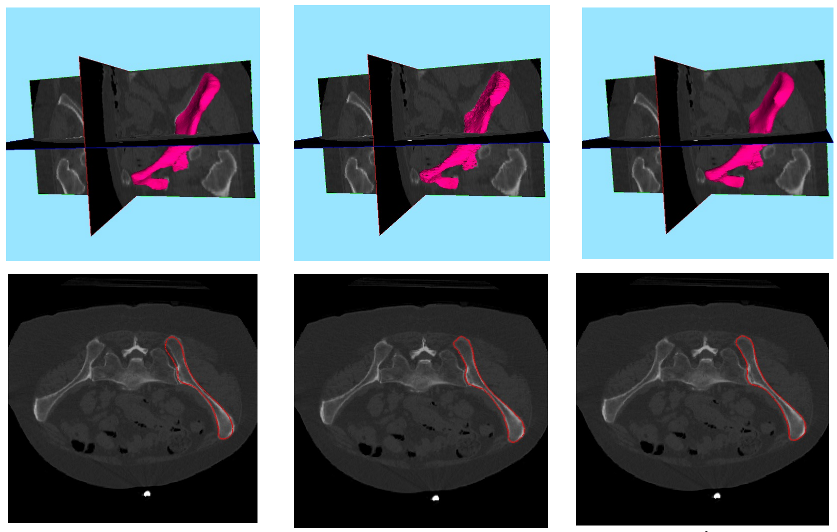
Initial | Intermediate | Final

Jianhua Yao

Engineering Research Center for Computer Integrated Surgical Systems and Technology

57

Results (Local Deformation)



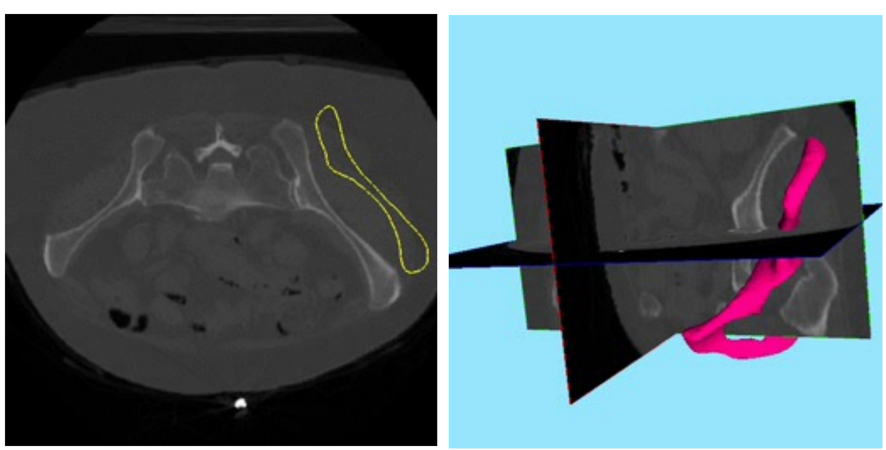
Initial Intermediate Final

Jianhua Yao

Engineering Research Center for Computer Integrated Surgical Systems and Technology

58

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

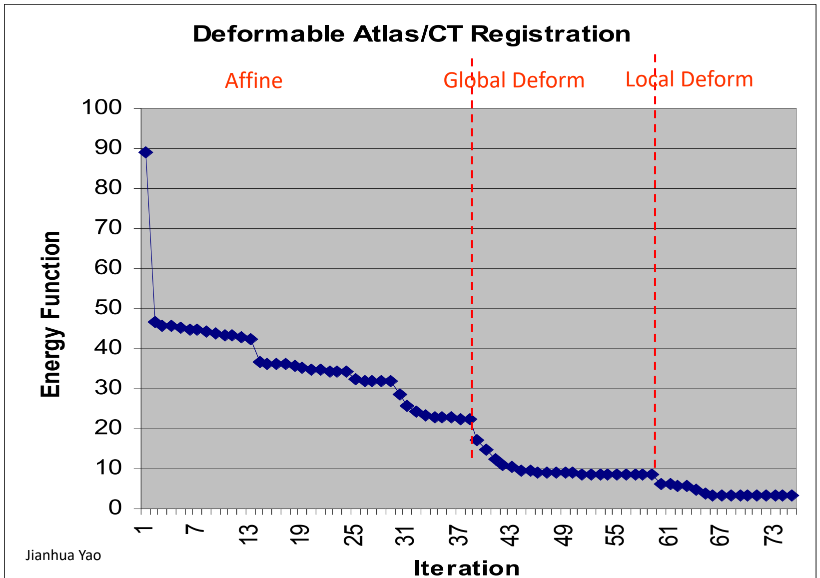


Jianhua Yao

Engineering Research Center for Computer Integrated Surgical Systems and Technology

59

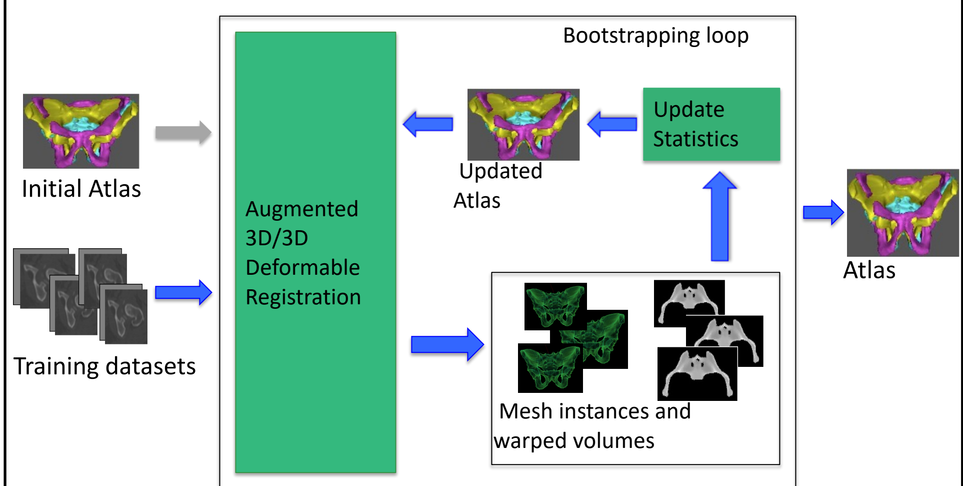
Results (Deformable Registration)



Engineering Research Center for Computer Integrated Surgical Systems and Technology

60

Iterative "bootstrapping" of Atlas



Chintalapani *et al.* MICCAI 2007

Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

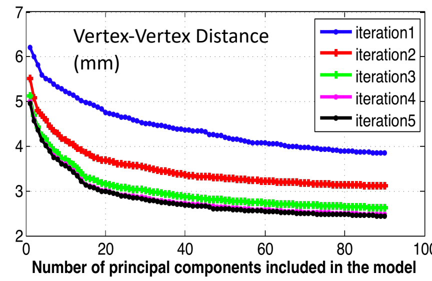
61

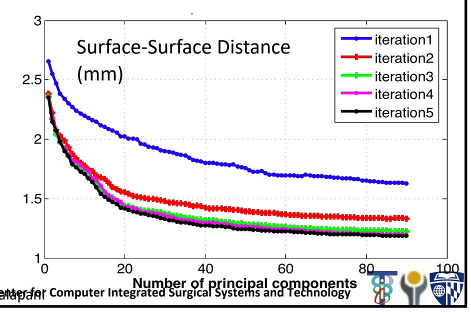
Leave-Out Validation Experiments

- # of iterations: 5
- # of data sets: 110
- # of data sets in atlas: 90
- # of data sets left out: 20
- Given a left-out dataset, s_j compute the estimated shape from atlas using

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

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





Slide credit: Gouthami Chintalapani

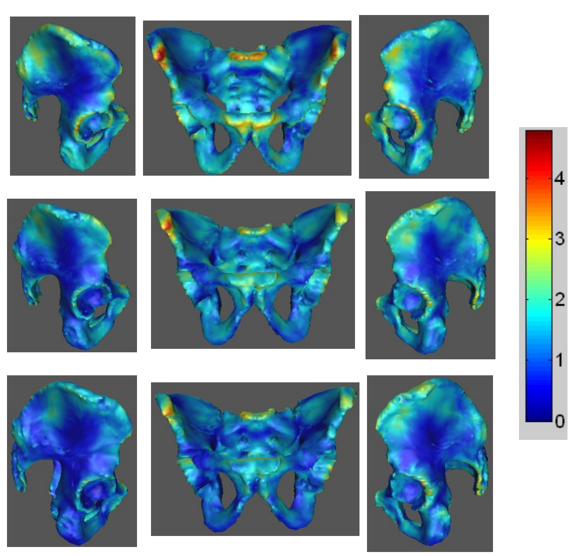
62

Distribution of Surface Registration Errors

Iteration 1

Iteration 3

Iteration 5



Slide credit: Gouthami Chintalapani

63

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

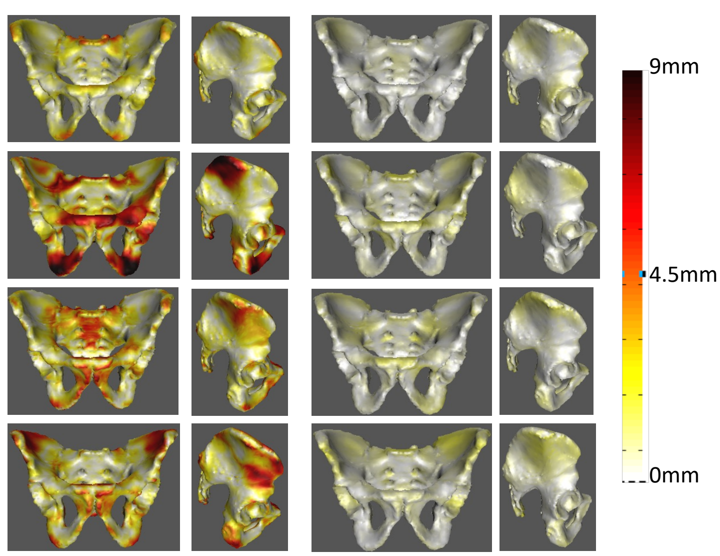
Average Difference between Atlases 1,2 and 3

Mean

Mean + Mode 1

Mean + Mode 2

Mean + Mode 3



Before iteration After iteration

Slide credit: Gouthami Chintalapani

Training Sample Size

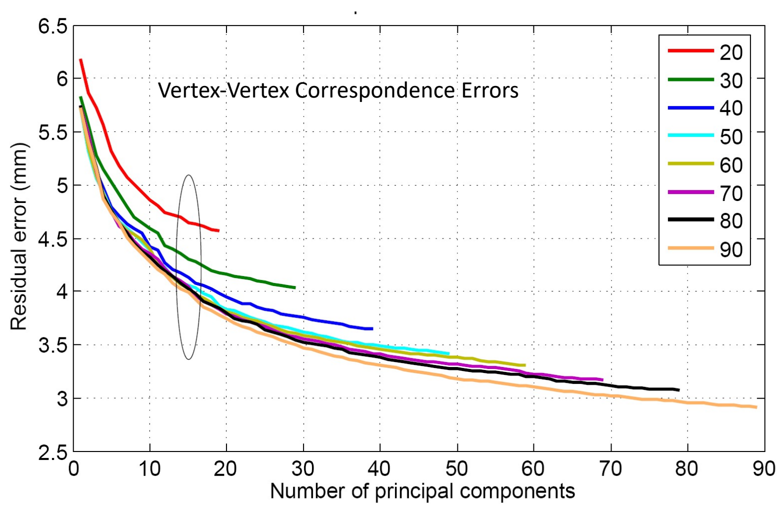
- Goal:
  - To determine the size of the training sample to build a stable statistical atlas
- Criteria:
  - Atlas is stable
  - No significant improvement in residual error
- Experimental setup:
  - Varying sample size 20, 40, 60, 80
  - Leave-20-out validation test
- Result:
  - Minimum of 50 data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

Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

66

Training Sample Size

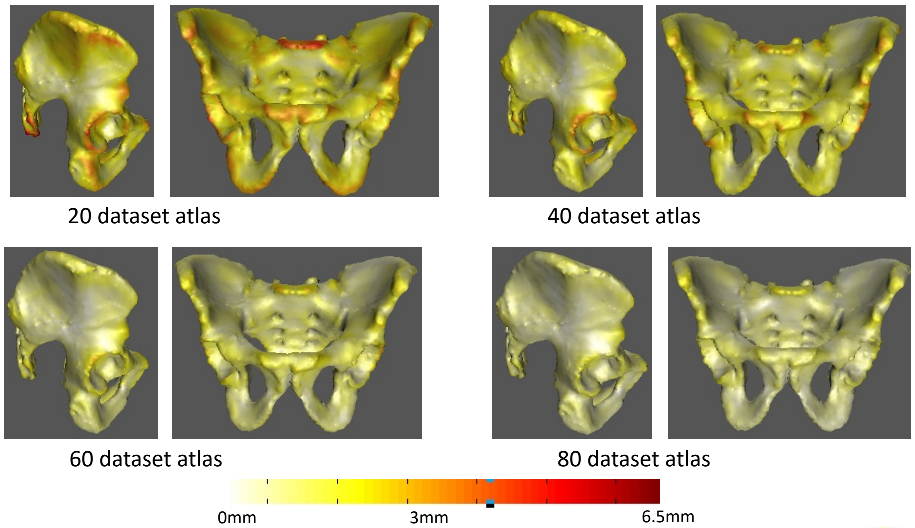


Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

67

Surface residual error using 18 modes for different sample set sizes

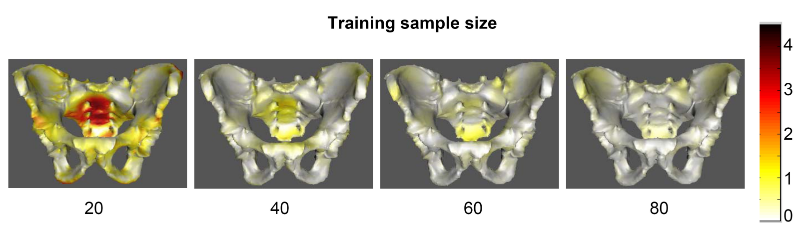


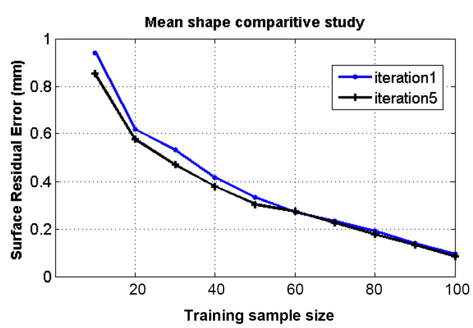
Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

68

Stability Analysis – Mean Shape





Engineering Research Center for Computer Integrated Surgical Systems and Technology

Slide credit: Gouthami Chintalapani

69

Shape Atlas Mesh Refinement

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

Mesh Vertex Improvement
(click here)

Active Appearances

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

Statistical Appearance Models

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

$$\|\mathbf{G}\| = 1$$
$$\sum \mathbf{G}_k = 0$$

Labelled image — Points — Shape-free patch

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

Appearance models, con'd

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

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

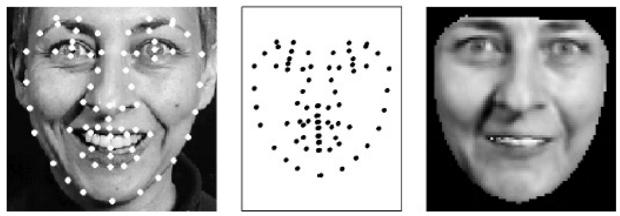
In fact, they created a multi-resolution hierarchy with models similar to the above at different resolutions.

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

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

Training Set for 2001 Cootes & Taylor paper

- 400 faces
- 68 points
- 10000 intensity values



Labelled image Points Shape-free patch

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

Complication

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

Answer : Form combined vector of shape and intensity variation

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

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

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

Further complication

- How do you find the right weights to use?

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

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

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

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

Engineering Research Center for Computer Integrated Surgical Systems and Technology

77

Face modes

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

Shape

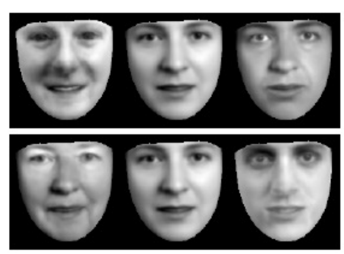


Fig. 3. First two modes of grey-level variation (±3 sd)

Intensity

Source: Cootes *et al. 1998*

Engineering Research Center for Computer Integrated Surgical Systems and Technology

78

Face modes

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

Combined

Source: Cootes *et al. 1998*

Basic Algorithm

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

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

Basic Iteration of the Method

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

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

Source: Cootes *et al. 2001*

Basic Iteration of the Method

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

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

Source: Cootes *et al. 2001*

Results

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

Source: Cootes *et al. 1998*

83

Results

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

84

Results: Knee Example

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

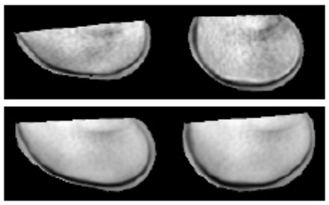


Fig. 12. First two modes of appearance variation of knee model

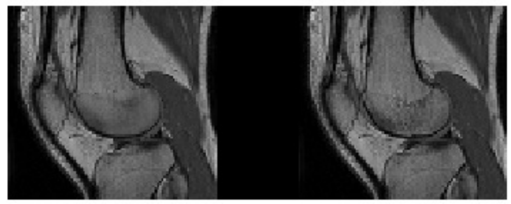


Fig. 13. Best fit of knee model to new image given landmarks

Source: Cootes *et al. 1998*

Results: Knee Example

Initial 2 its Converged (11 its)

Fig. 14. Multi-Resolution search for knee

Source: Cootes *et al. 1998*

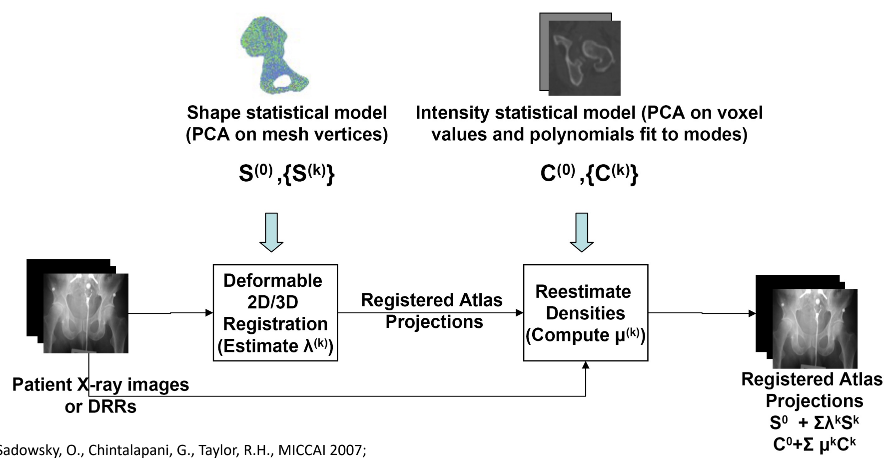
Deformable registration between density atlas and a set of 2D X-Rays

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

Jianhua Yao

87

2D/3D Registration – Shape and Intensity Models



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

Slide credit: Gouthami Chintalapani

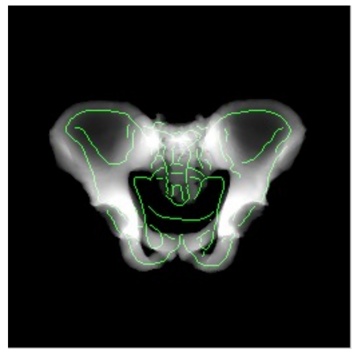
90

2D/3D Registration – Shape and Intensity

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

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

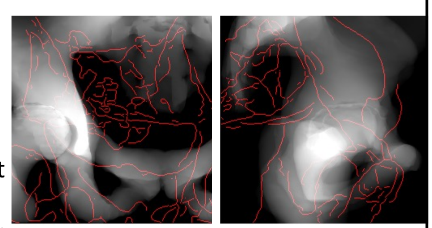
Avg surface registration accuracy: 2.21mm
Avg. reduction in RMS errors intensity: 27%



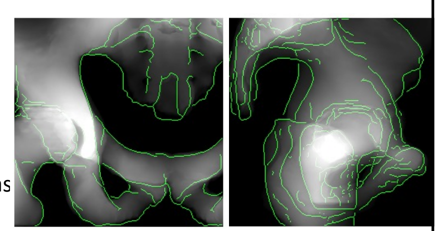
Slide credit: Gouthami Chintalapani

2D/3D Registration – Hip Model

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



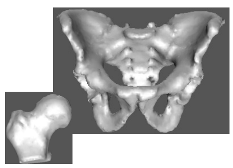
Pelvis atlas registered to hip projection images



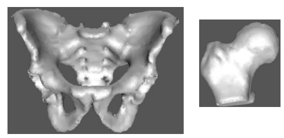
Hip atlas registered to hip projection images

Multi-Component Atlas

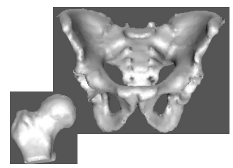
1. Two components – pelvis and femur
2. Create mesh instances of pelvis and femur separately
3. Align pelvis and femur meshes together
4. Align pelvis meshes
5. Align femur meshes
6. Concatenate pelvis and femur meshes
7. PCA on the concatenated mesh



Combined Rigid+Scale



Separate Rigid



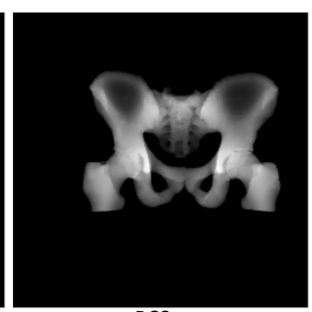
Combined Statistical Analysis

Slide credit: Gouthami Chintalapani

Multi-Component Hip Atlas

PC1

PC2



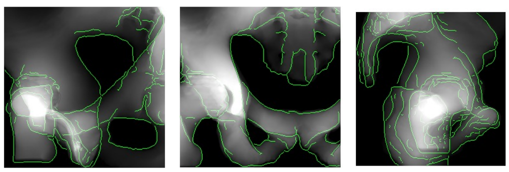
PC3

[1] Chintalapani *et al.* CAOS 2009

Slide credit: Gouthami Chintalapani

2D/3D Registration – Hip Model

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



Atlas projections overlaid on DRR images after registration

2D/3D deformable registration

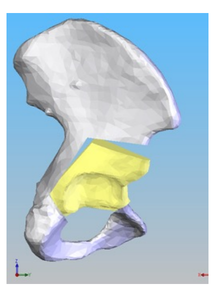
Chintalapani *et al.* CAOS 2009

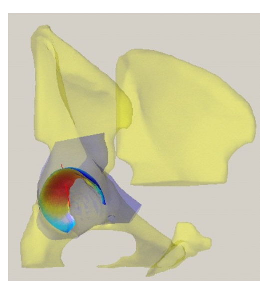
Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

95

Applications – Hip Osteotomy





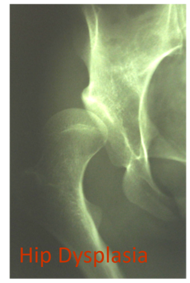
Slide credit: Gouthami Chintalapani

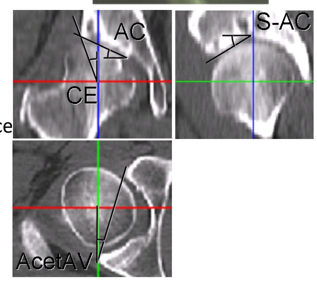
Engineering Research Center for Computer Integrated Surgical Systems and Technology

96

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 proce
- **Limitations of current navigation systems:**
  - Lack the ability to track bone fragment alignment
  - Do not provide anatomical measurements
  - Omit biomechanical-based planning and guidance
  - Ignore the risk of reducing joint range-of-motion



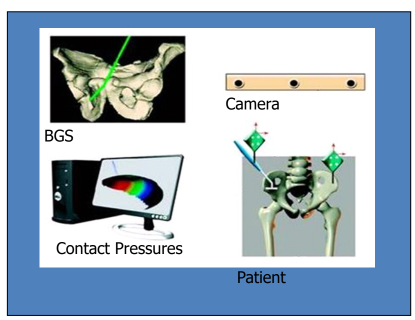


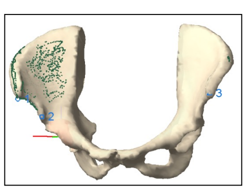
Anatomical measurements used to diagnose hip dysplasia

Slide credit: Gouthami Chintalapani, Mehran Armand

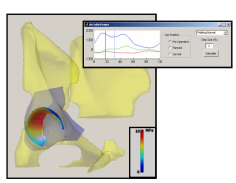
Biomechanical Guidance System (BGS)

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

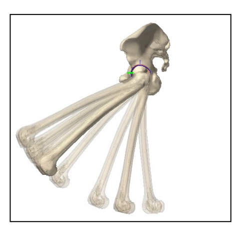




Model to Patient Registration



Joint contact-pressure after PAO

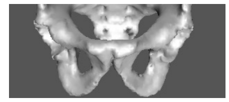


Hip-range-of-motion

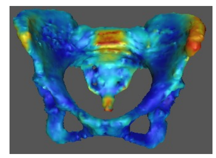
Slide credit: Gouthami Chintalapani, Mehran Armand

Atlas Based Extrapolation of CT

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



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



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

Chintalapani *et al.* SPIE 2010

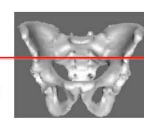
Slide credit: Gouthami Chintalapani

Atlas Adaptation to Partial Data

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

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

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



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

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

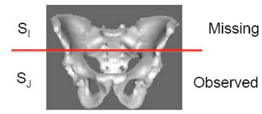
Estimate the total shape as

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

Chintalapani *et al.* SPIE 2010

Slide credit: Gouthami Chintalapani

Atlas Adaptation to Partial Data with Xray Images



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

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

- Final atlas extrapolated model is given as

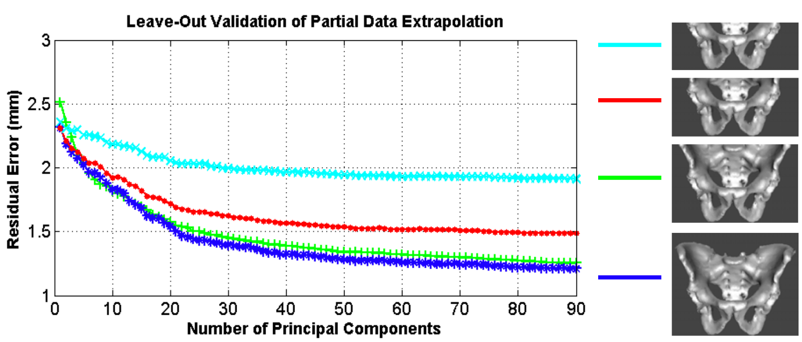
$$S^{(est)} = \begin{bmatrix} \left(s\mathbf{R}\left(\bar{S}_I + \mathbf{U}_I\vec{\lambda}\right) + \vec{\mathbf{p}}\right) \\ S_J^{(obs)} \end{bmatrix}$$

Chintalapani *et al.* SPIE 2010

Slide credit: Gouthami Chintalapani

101

Results



Slide credit: Gouthami Chintalapani

102

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

Engineering Research Center for Computer Integrated Surgical Systems and Technology

103

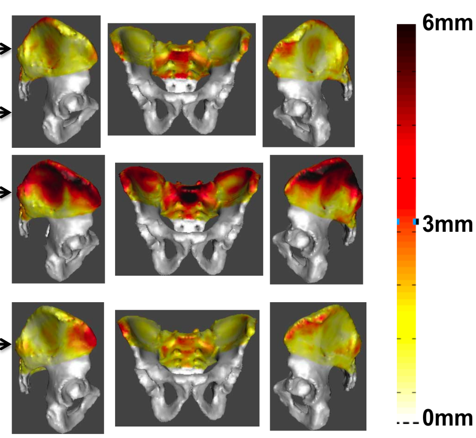
Results – Atlas Experiments

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

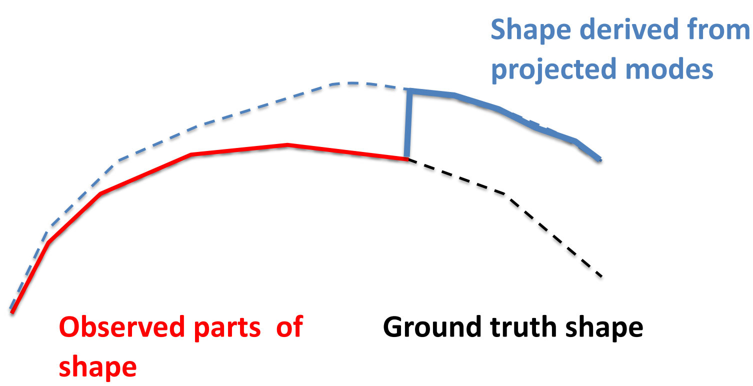
Chintalapani *et al.* SPIE 2010

Slide credit: Gouthami Chintalapani, Mehran Armand

Engineering Research Center for Computer Integrated Surgical Systems and Technology

104

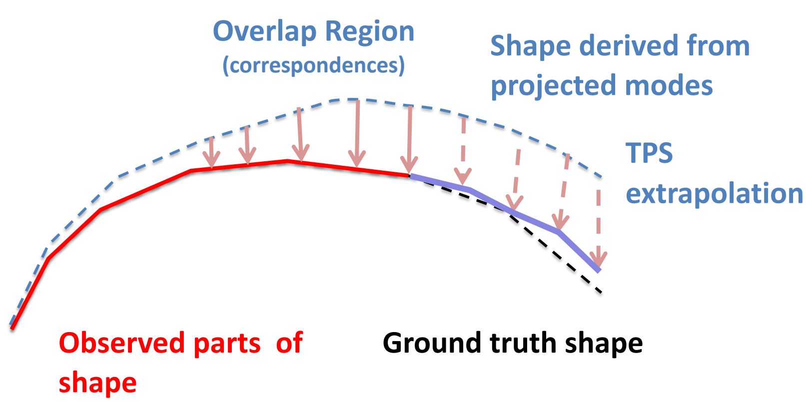
Cut-and-Paste Model Completion



Engineering Research Center for Computer Integrated Surgical Systems and Technology

105

Model Completion with Thin Plate Spline



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

Engineering Research Center for Computer Integrated Surgical Systems and Technology

106

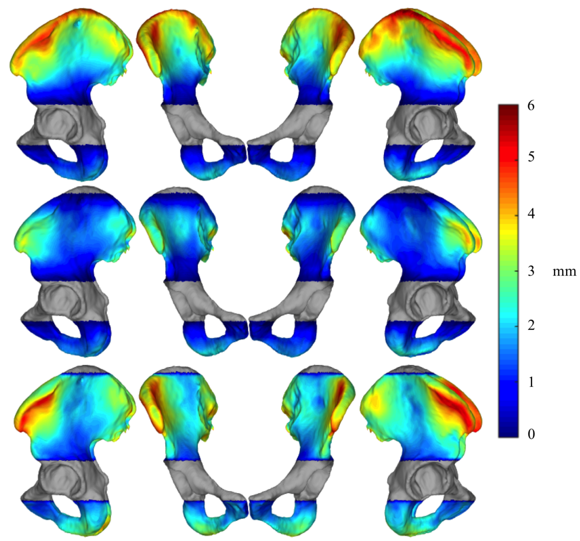
Model Completion of Pelvis from Partial CT Only

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

Smooth extrapolation using only acetabulum scan

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

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



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

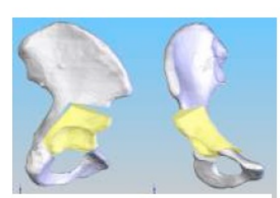
107

Osteotomy Simulations

- Atlas extrapolated model is used primarily for two reasons:

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

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

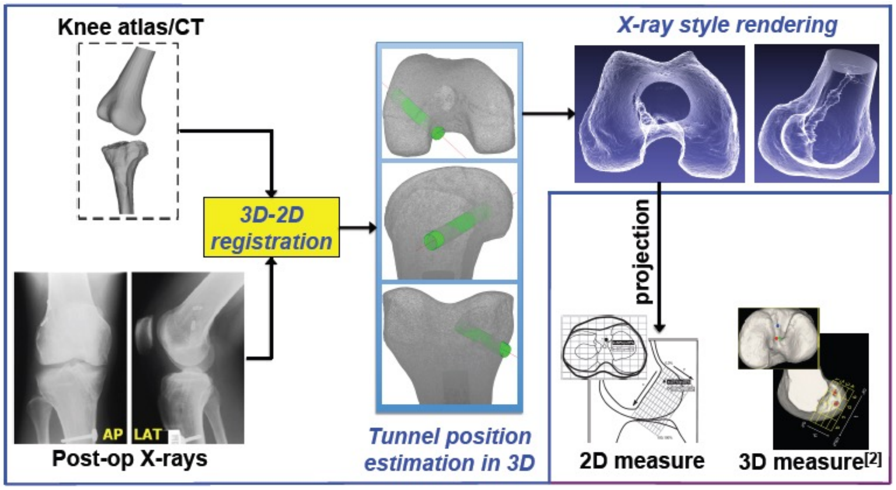


Slide credit: Gouthami Chintalapani, Mehran Armand

108

Statistical Assessment of ACL Tunnel Positions

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



113

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

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

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

M-Step (Pose):

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

M-Step (Shape):

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

where

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

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

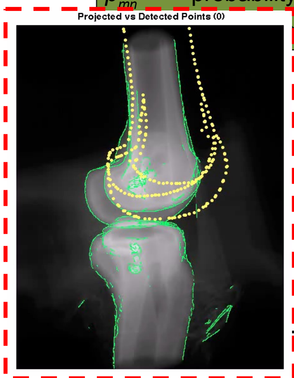
114

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

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

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

matches image contour element n



M-Step (Pose):

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

M-Step (Shape):

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

where

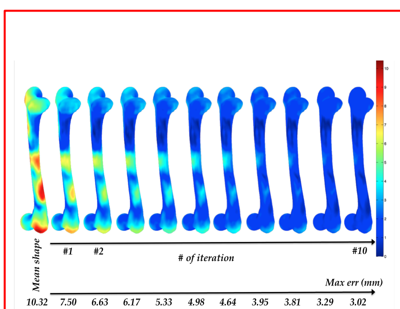
$$d_{mn} = \left\| \sqrt{p_{mn}}\, \vec{V}_n \left[\vec{C}_n - \left(\mathbf{R}\vec{X}_m^{(0)} + \vec{\mathbf{t}} + \sum_k \mathbf{R}\vec{e}_m^{(k)} \right) \right] \right\|$$

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

Engineering Research Center for Computer Integrated Surgical Systems and Technology

115

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



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

ted model contour

mage contour element n

(Pose):

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

M-Step (Shape):

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

where

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

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

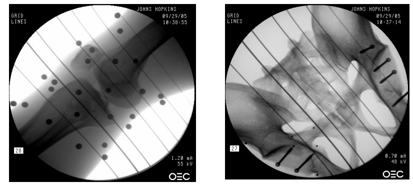
Engineering Research Center for Computer Integrated Surgical Systems and Technology

116

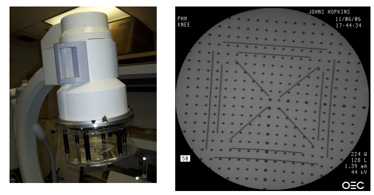
C-arm Distortion

- What is distortion ?
  - –Avg distortion: **2.14** mm/pixel
  - –max distortion: **4.60** mm/pixel
- How to rectify images ?
  - Phantom based correction
  - Polynomial functions to model distortion

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



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



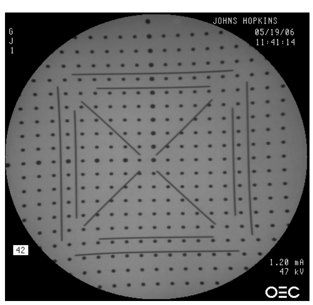
Typical bi-planar phantom used for C-arm calibration

Slide credit: Gouthami Chintalapani

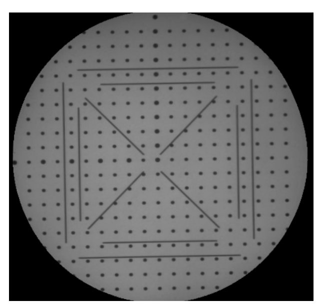
Engineering Research Center for Computer Integrated Surgical Systems and Technology

117

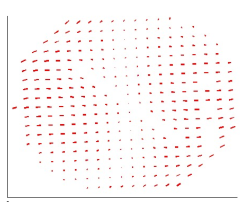
C-Arm Distortion Correction



Warped X-ray image of the phantom



Dewarped X-ray image



Distortion vector map

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

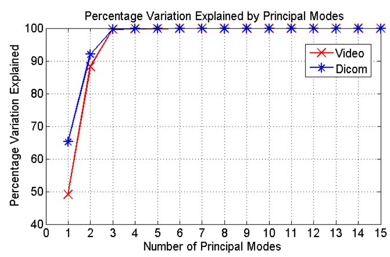
Slide credit: Gouthami Chintalapani

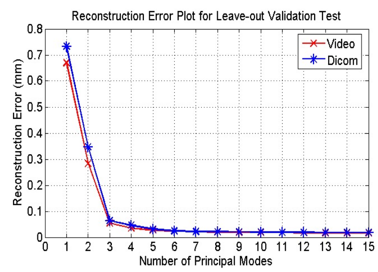
Engineering Research Center for Computer Integrated Surgical Systems and Technology

118

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



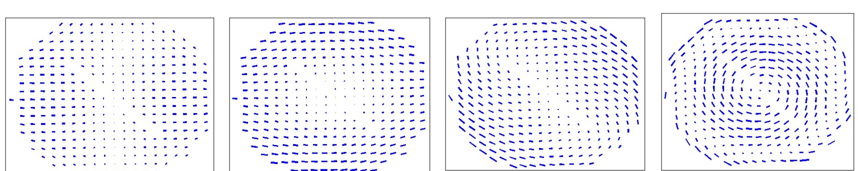


Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

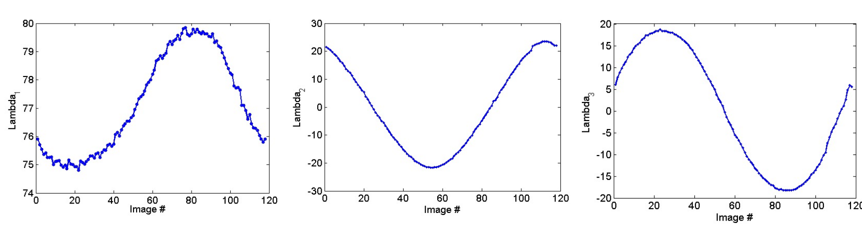
119

Circular Trajectory



mean mode1 mode2 mode3

Distortion patterns from PCA modes



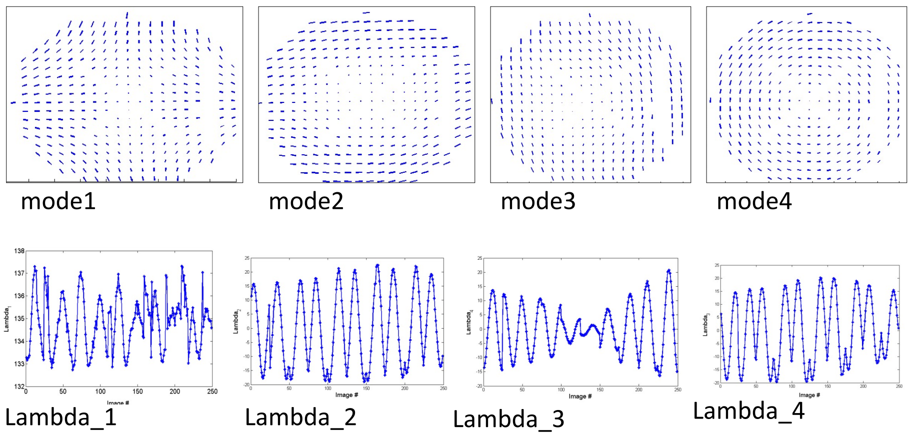
Lambda_1 Lambda_2 Lambda_3

Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

120

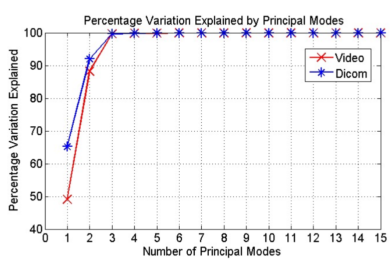
C-arm Imaging Volume

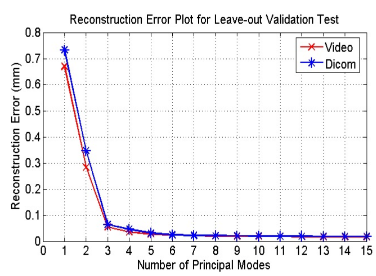


Slide credit: Gouthami Chintalapani

Eigen Analysis of Distortion Maps

- First three modes are significant and explain 99% of the variation
- Leave-out validation tests indicate that the distortion parameters can be recovered with an accuracy of less than 0.1 mm/pixel.



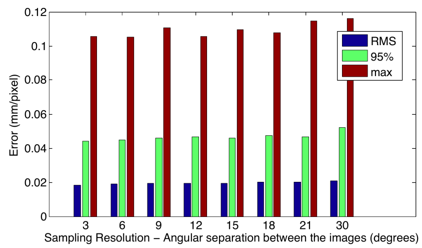


$$\epsilon = \left\| \triangle \vec{d} - \left(M_0 + \sum_{i=1}^{n} \lambda_i D_i \right) \right\|^2$$

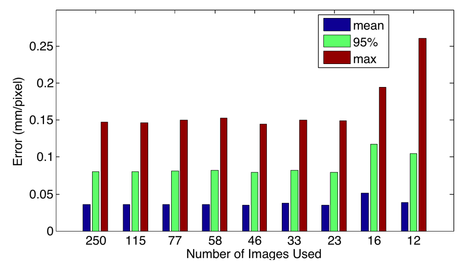
Slide credit: Gouthami Chintalapani

Sampling Resolution

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



Circular Trajectory



C-arm Imaging Volume

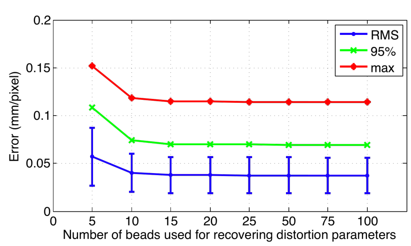
Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

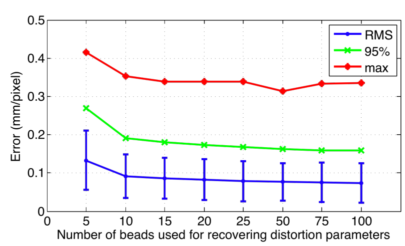
123

Recovering Distortion Parameters

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



Circular Arc



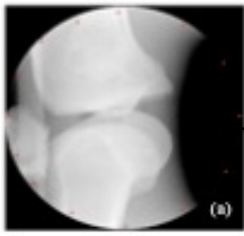
C-arm Imaging Volume

Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

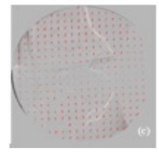
124

Small Phantom based Distortion Correction

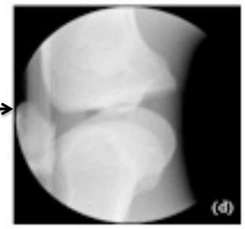


Example patient image peripheral beads

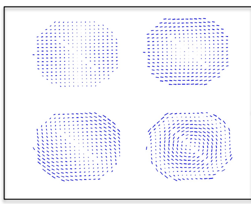
Peripheral bead locations



Distortion corrected image



Distortion mode matching



Prior distortion model

Distortion modes

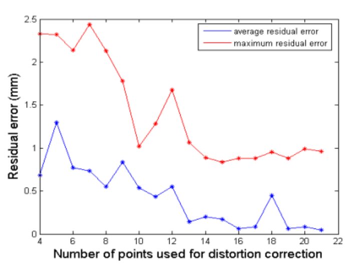
Avg. residual error: 0.2mm/pixel
Max. residual error: 0.8mm/pixel

Slide credit: Gouthami Chintalapani

 Chintalapani *et al.* SPIE 2007 **Engineering Research Center for Computer Integrated Surgical Systems and Technology**

125

Small Phantom based Distortion Correction



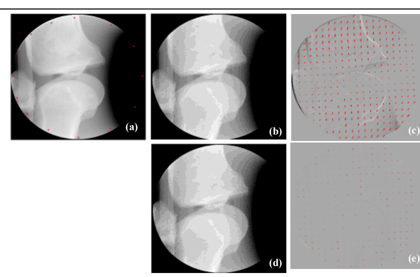


Fig. 2 (Left) Residual Error in distortion vs number of points used for distortion correction.

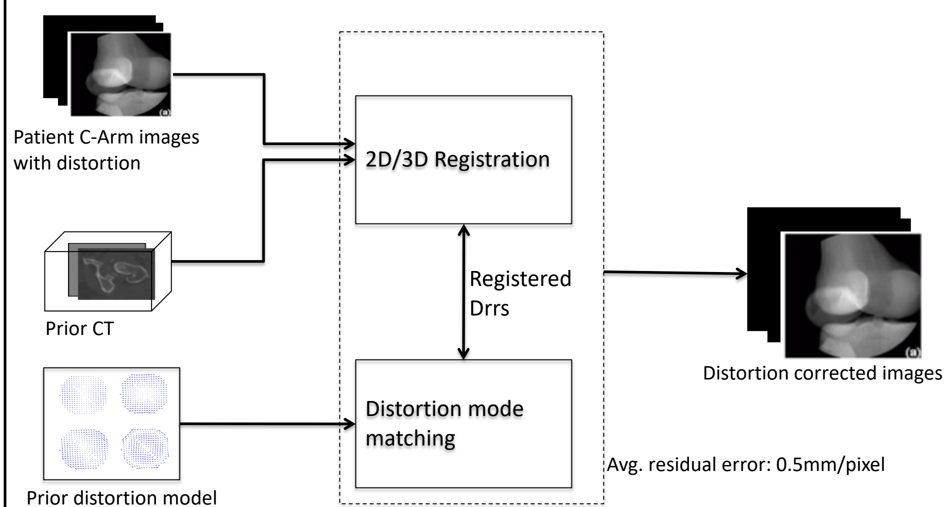
Fig.2. (Right) Results from simulation experiments using simpler phantom. (a) Knee X-ray image with phantom BBs overlaid in red color (b) distortion corrected image with dense grid pattern phantom (c) (b) – (a) with distortion vectors overlaid in red (d) distortion corrected image with using BBs in (a) and PCA (e) (b) – (d) with the residual distortion vectors overlaid in red

Statistical Characterization of C-arm Distortion with Intra-operative Application

 Engineering Research Center for Computer Integrated Surgical Systems and Technology

126

Using Patient CT as Fiducial

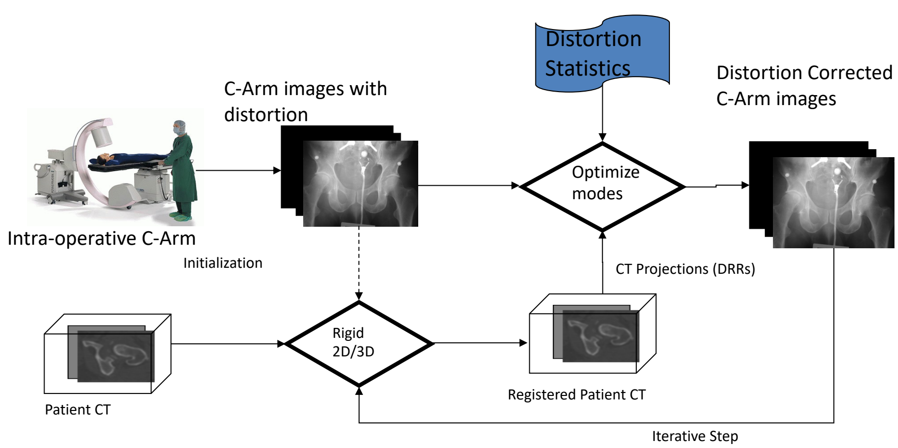


Chintalapani *et al.* ISBI 2007 Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

127

C-Arm Distortion Correction Using Patient CT as Fiducial



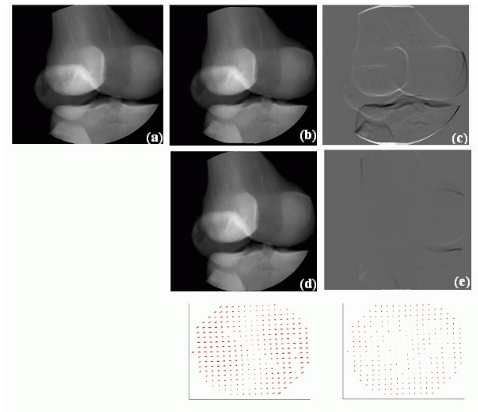
Thanks to Ofri Sadowsky for assistance with 2D/3D registration

Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

128

C-Arm Distortion Correction Using Patient CT as Fiducial



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

Slide credit: Gouthami Chintalapani

Engineering Research Center for Computer Integrated Surgical Systems and Technology

129