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

Deformable Registration

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

Images: Tom Fletcher

Copyright 2021 R. H. Taylor
Engineering Research Center for Computer Integrated Surgical Systems and Technology
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?
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.
    \[
    \begin{align*}
    \tilde{u} &= \text{TrimToBox}(\tilde{x}) \\
    \tilde{y} &= \sum_{i,j,k} \tilde{c}_{i,j,k} B_i(\tilde{u}_x) B_j(\tilde{u}_y) B_k(\tilde{u}_z) \\
    \text{or} \\
    \tilde{y} &= \sum_{i,j,k} \tilde{c}_{i,j,k} N_i(\tilde{u}_x) N_j(\tilde{u}_y) N_k(\tilde{u}_z)
    \end{align*}
    \]
  - Note: In this case, the coefficients will also parameterize the “shape”

Radial Basis Functions

Given a scalar function \(\phi(\cdot)\) and a set of sample points \(\tilde{p}_h\) with associated deformations \(\tilde{d}_h\), one can represent the deformation \(\Phi\) at a point \(\tilde{x}\) by

\[
\Phi(\tilde{x}) = \sum_h \tilde{d}_h \phi_h \left( \|\tilde{x} - \tilde{p}_h\| \right)
\]

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

  http://dx.doi.org/10.1016/S0262-8856(00)00057-3
Thin Plate Splines

- Minimum energy spline deformations

\[ TPS(\mathbf{v}; \mathbf{a}, \mathbf{B}, \mathbf{C}, \mathbf{P}) = \mathbf{a} + \mathbf{B} \cdot \mathbf{v} + \sum \mathbf{C} \cdot \mathbf{U}(\mathbf{v} - \mathbf{p}) \]

where \( \mathbf{U}(r) = r^2 \log(r) \) for 2D images

- Global support
- Popularized by Fred Bookstein for analysis of anatomic variation
  

Thin Plate Splines Digression


M-dimensional Thin Plate Spline Summary

Given

\[ TPS(\vec{v}; \vec{a}, \vec{B}, \vec{C}, P) = \vec{a} + \vec{B} \cdot \vec{v} + \sum c_i U(||\vec{v} - \vec{p}_i||) \]

where

\[ U(r) = \begin{cases} r^2 \log(r) & \text{for 2D} \\ r^2 \log(r^2) & \text{for 3D} \end{cases} \]

\[ \vec{v} = [v_1, \ldots, v_M]^T \]

\[ \vec{p}_i = [p_{i1}, \ldots, p_{iM}]^T \]

\[ P = [\vec{p}_1, \ldots, \vec{p}_N]^T \]

\[ C = [\vec{c}_1, \ldots, \vec{c}_N]^T \]

\[ B = [\vec{b}_1, \ldots, \vec{b}_M]^T \]

Note: Some sources give

\[ U(r) = \begin{cases} r^{m+1} \ln(r) & \text{for } m=2 \text{ or 4} \\ r^{m+1} & \text{otherwise} \end{cases} \]

M-dimensional Thin Plate Spline Fitting

Given

\[ \vec{V} = [\vec{v}_1, \ldots, \vec{v}_N], \quad \vec{F} = [\vec{f}_1, \ldots, \vec{f}_N] \]

find \( \vec{a}, \vec{B}, \vec{C} \) such that

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

To do this, solve the linear system

\[ \begin{bmatrix} K_{(N\times N)} & \vec{I}_{(N\times 1)} & \vec{V} \\
\vec{I}_{(1\times N)} & 0 & 0 \\
\vec{V}^T & 0 & 0_{(M\times M)} \end{bmatrix} \begin{bmatrix} \vec{C}^T \\
\vec{a}^T \\
\vec{B}^T \end{bmatrix} = \begin{bmatrix} \vec{F}^T \\
0 \\
0_{(M\times 1)} \end{bmatrix} \]

where

\[ K_{ij} = K_{ji} = \frac{1}{2} \frac{U(||\vec{v}_i - \vec{v}_j||)}{U(||\vec{v}_i - \vec{v}_j||)} \quad \text{with } U(r) = r^2 \log(r) \text{ or } U(r) = r^2 \log(r^2) \]

\[ K_{ij} = \left( \vec{v}_i - \vec{v}_j \right) \cdot \left( \vec{v}_i - \vec{v}_j \right) \log \left( \sqrt{||\vec{v}_i - \vec{v}_j||} \right) \cdot \left( \vec{v}_i - \vec{v}_j \right) \]
TPS 2D case

Given a set of points $\mathbf{p}_i = [x_i, y_i]$ and corresponding points $\mathbf{p}_i^* = [x_i^*, y_i^*]$, we want to find TPS parameters such that $\mathbf{p}_i^* = TPS(\mathbf{p}_i, a, b, c, d)$.

To do this, we solve the least squares problem

$$
\begin{bmatrix}
0 & \cdots & U_{1,k} & \cdots & U_{1,N} & 1 & x_1 & y_1 \\
\vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots \\
U_{k,1} & \cdots & 0 & \cdots & U_{k,N} & 1 & x_k & y_k \\
\vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots \\
U_{N,1} & \cdots & U_{N,k} & \cdots & 0 & 1 & x_N & y_N \\
1 & \cdots & 1 & \cdots & 1 & 0 & 0 & 0 \\
x_1 & \cdots & x_k & \cdots & x_N & 0 & 0 & 0 \\
y_1 & \cdots & y_k & \cdots & y_N & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
c_1 \\
\vdots \\
c_N \\
\vdots \\
\end{bmatrix}
= 
\begin{bmatrix}
\mathbf{p}_1^* \\
\vdots \\
\mathbf{p}_N^* \\
\vdots \\
\end{bmatrix}
$$

where $U_{ij} = U_{ji} = U(\|\mathbf{p}_i - \mathbf{p}_j\|)$

M-dimensional Thin Plate Spline Fitting

Define

$$
L_{[M+N+1;M+N+1]} =
\begin{bmatrix}
K_{[NN]} & \mathbf{1}_{[N-1]} & \mathbf{V} \\
\mathbf{1}_{[N-1]} & 0 & 0 \\
\mathbf{V}^T & 0 & 0_{[M+M]} \\
\end{bmatrix}
$$

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 $\mathbf{V}_i$ 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

Other Radial Basis Functions

Note that the function $U(r)$ in the previous discussion is 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

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

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

$$
\Psi(r, \sigma) = \begin{cases} 
1 - \left(1 - \frac{r}{\sigma}\right)^{2k+1} & \text{if } 0 \leq r \leq \sigma \\
0 & \text{otherwise}
\end{cases}
$$

http://dx.doi.org/10.1016/S0262-8856(00)00057-3

Deformable Registration to Statistical “Atlases”

Deformable 3D/3D
Jianhua Yao

Deformable 2D/3D
Ofri Sadowsky
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:

- 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

Statistical models

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

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

We proceed as follows:

$$\tilde{a}^{(avg)} = \frac{1}{N} \sum_j \tilde{a}_j; \quad \tilde{b}_j = \tilde{a}_j - \tilde{a}^{(avg)}; \quad B = [\tilde{b}_1, \ldots, \tilde{b}_N].$$

Then form the singular value decomposition

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

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

$$B = [\tilde{u}_1, \ldots, \tilde{u}_N, \tilde{u}_{N+1}, \ldots, \tilde{u}_M] \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ \vdots & \vdots \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} V^T = [\tilde{u}_1, \ldots, \tilde{u}_N] \sum_{(N)} V^T = U^{(N)} \Sigma^{(N)} V^T.$$
Principal Components Analysis (PCA)

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

$$B = [\overrightarrow{u}_1, \ldots, \overrightarrow{u}_N] \sum^{(N)} V^T = U^{(N)} \sum^{(N)} V^T$$

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

where

$$\Lambda^{(k)} = \text{transpose}(U^{(N)}) \overrightarrow{b}_k$$

So

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

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

$$\hat{a}_k \approx \bar{a}^{(avg)} + \lambda_1^{(k)} \overrightarrow{u}_1 + \cdots + \lambda_D^{(k)} \overrightarrow{u}_D$$

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

$$\overrightarrow{b}^{(arb)} = \overrightarrow{a}^{(arb)} - \bar{a}^{(avg)}$$

$$\Lambda^{(arb)} = \text{transpose}(U^{(D)}) \overrightarrow{b}^{(arb)}$$

$$\overrightarrow{a}^{(arb)} \approx \bar{a}^{(avg)} + \lambda_1^{(arb)} \overrightarrow{u}_1 + \cdots + \lambda_D^{(arb)} \overrightarrow{u}_D$$
Statistical Atlases & PCA

Given a set of N models \( \mathbf{X}^{(i)} = [\mathbf{x}_k^{(i)}]^T = [\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \mathbf{x}_k^{(3)}, \ldots] \), compute

\[
\bar{\mathbf{X}}^{(\text{avg})} = \begin{bmatrix}
\vdots \\
\bar{x}_k^{(\text{avg})} \\
\vdots
\end{bmatrix}
\]

where \( \bar{x}_k^{(\text{avg})} = \frac{1}{N} \sum_j \mathbf{x}_k^{(j)} \) and the differences

\[
\bar{\mathbf{D}}^{(j)} = \bar{\mathbf{X}}^{(j)} - \bar{\mathbf{X}}^{(\text{avg})} = \begin{bmatrix}
\vdots \\
\bar{d}_k^{(j)} \\
\vdots
\end{bmatrix}
\]

where \( \bar{d}_k^{(j)} = \mathbf{x}_k^{(j)} - \bar{x}_k^{(\text{avg})} \). Create the matrix

\[
\mathbf{D} = \begin{bmatrix}
\bar{d}_1^{(1)} & \ldots & \bar{d}_k^{(1)} & \ldots & \bar{d}_N^{(1)} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\bar{d}_1^{(N)} & \ldots & \bar{d}_k^{(N)} & \ldots & \bar{d}_N^{(N)}
\end{bmatrix}_{3 \text{Nvertices} \times N}
\]

Statistical Atlases & PCA

Compute the singular value decomposition of \( \mathbf{D} \)

\[
\mathbf{D} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T
\]

where \( \mathbf{\Sigma} = \begin{bmatrix} \text{diag}(\bar{\sigma}) \\ \mathbf{0} \end{bmatrix} \).

\[
\mathbf{D} = \mathbf{U} \begin{bmatrix} \text{diag}(\bar{\sigma}) \mathbf{V}^T \\ \mathbf{0} \end{bmatrix}
\]

Note that

\[
\frac{1}{N-1} \mathbf{D}^T \mathbf{D} = \frac{1}{N-1} \mathbf{V} \mathbf{\Sigma} \mathbf{\Sigma}^T \mathbf{V}^T = \frac{1}{N-1} \mathbf{V} \mathbf{\Sigma}^2 \mathbf{V}^T
\]

\[
\frac{1}{N-1} \mathbf{D} \mathbf{D}^T = \frac{1}{N-1} \mathbf{U} \mathbf{\Sigma} \mathbf{\Sigma} \mathbf{V}^T = \frac{1}{N-1} \mathbf{U} \mathbf{\Sigma}^2 \mathbf{V}^T
\]
Statistical Atlases & PCA

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

\[
\mathbf{D}^{(j)} = \mathbf{U} \cdot \begin{bmatrix}
\lambda^{(j)}_1 \\
\vdots \\
\lambda^{(j)}_M \\
0\\
\end{bmatrix}
\]

where \( \begin{bmatrix}
\lambda^{(j)}_1 \\
\vdots \\
\lambda^{(j)}_M \\
0\\
\end{bmatrix} \) is the \( j \)th column of \( \mathbf{D}^{(j)} \) is the \( j \)th column of \( \text{diag}(\mathbf{\hat{\sigma}})\mathbf{V}^T \).

If we define

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

we get the expression

\[
\mathbf{D}^{(j)} = \mathbf{M} \mathbf{\hat{\lambda}} \quad \text{ where } \mathbf{\hat{\lambda}} \text{ is the } j \text{th column of } \text{diag}(\mathbf{\hat{\sigma}})\mathbf{V}^T.
\]

Statistical Atlases & PCA

Note that while \( \mathbf{U} \) is 3\( N \)\(_{\text{vertices}} \times 3\( N \)\(_{\text{vertices}} \) (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 \( \sigma_i \). Since the standard algorithms for SVD produce positive singular values \( \sigma \), 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 3\( N \)\(_{\text{vertices}} \times 3\( N \)\(_{\text{vertices}} \) 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 $3N_{\text{verts}} \times N$ matrix $D$. Many SVD packages give you the option to compute only the singular values $\sigma$ and the right hand side matrix $V$ or its transpose. Then, $M$ can be computed from

$$M \text{diag}(\sigma)V^T = D$$
$$M \text{diag}(\sigma) = DV$$

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

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

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

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

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

$$= \bar{\lambda}^{(\text{inst})}$$
Statistical Atlases & PCA

Suppose that we select \( \tilde{\lambda} = [\lambda_1, \ldots, \lambda_n]^T \) as a random variable with some distribution having expected value \( E(\tilde{\lambda}) = \tilde{\lambda} \) and covariance
\[
\text{cov}(\tilde{\lambda}) = E(\tilde{\lambda} \cdot \tilde{\lambda}^T) = \begin{bmatrix} E(\lambda_1^2) & \cdots & E(\lambda_1 \lambda_n) \\ \vdots & \ddots & \vdots \\ E(\lambda_n \lambda_1) & \cdots & E(\lambda_n^2) \end{bmatrix} = \Sigma^2
\]
and compute a corresponding random model \( \tilde{\mathbf{X}}(\tilde{\lambda}) \)
\[
\tilde{\mathbf{X}}(\tilde{\lambda}) = \tilde{\mathbf{X}}(\text{avg}) + \mathbf{M} \cdot \tilde{\lambda}
\]
What can we say about the expected value and covariance of \( \tilde{\mathbf{X}}(\tilde{\lambda}) \)?

Statistical Atlases & PCA

For the expected value, we have
\[
E(\tilde{\mathbf{X}}(\tilde{\lambda})) = E(\tilde{\mathbf{X}}(\text{avg}) + \mathbf{M} \cdot \tilde{\lambda}) \\
= \tilde{\mathbf{X}}(\text{avg}) + \mathbf{M} \cdot E(\tilde{\lambda}) = \tilde{\mathbf{X}}(\text{avg}) + \mathbf{M} \cdot \tilde{\lambda}
\]
Then
\[
\text{cov}(\tilde{\mathbf{X}}(\tilde{\lambda})) = E(\tilde{\mathbf{D}}(\tilde{\lambda}) \cdot \tilde{\mathbf{D}}(\tilde{\lambda})^T) \quad \text{where} \quad \tilde{\mathbf{D}}(\tilde{\lambda}) = \tilde{\mathbf{X}}(\tilde{\lambda}) - \tilde{\mathbf{X}}(\text{avg})
\]
\[
= E(\mathbf{M} \cdot \tilde{\lambda} \cdot \tilde{\lambda}^T \cdot \mathbf{M}) \\
= \mathbf{M} \cdot E(\tilde{\lambda} \cdot \tilde{\lambda}^T) \cdot \mathbf{M}^T \\
= \mathbf{M} \cdot \Sigma^2 \cdot \mathbf{M}^T
\]
Statistical Atlases & PCA

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

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

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

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

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

Statistical Atlas

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

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

This information may be used in many ways, including

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

1. Model Representation/Parameterization
   - Points, landmarks, meshes, parametric models, level sets
   - Parameterized representation of medical images

2. Model Correspondence/Alignment
   - Rigid, affine, deformable registration methods
   - Aligned images in correspondence to the template

3. Statistical Analysis
   - PCA, ICA, Kernel PCA, non-linear statistical methods
   - Statistical model/atlas

Model Representation

- Tetrahedral mesh represents shape

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

\[
P^d(u) = \sum_{\mathcal{H}} c_i B^d_i(u)
\]

where

\[
k = (k_0, k_1, k_2, k_3) \quad u = (u_0, u_1, u_2, u_3)
\]

\[
|k| = k_0 + k_1 + k_2 + k_3 \quad |u| = 1
\]

\[
B^d_i(u) = \frac{d!}{k_0! k_1! k_2! k_3!} u_0^{k_0} u_1^{k_1} u_2^{k_2} u_3^{k_3}
\]

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

Model Creation

- CT dataset
  - Segmentation of pelvis anatomy using Analyze[1]
- Labelled CT
- Mesher[2]

[2] Mohammed et al., 2005

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

Model Correspondence

- Need to establish a common coordinate frame for the training database
- Need to establish point correspondence between the training datasets
Model Shape Correspondences

- Automatic deformable registration based shape correspondences

Model Intensity Correspondences

- Automatic deformable registration based correspondences
Principal Component Analysis

• Given the mesh instances of training sample,

\[
S = \begin{bmatrix}
\hat{x}_1 & \hat{x}_2 & \ldots & \hat{x}_N \\
\hat{y}_1 & \hat{y}_2 & \ldots & \hat{y}_N \\
\hat{z}_1 & \hat{z}_2 & \ldots & \hat{z}_N \\
\end{bmatrix}
\]

• Compute mean and subtract the mean from the sample

\[
\bar{S} = S - \bar{S} = S - \frac{1}{N} \sum_{i=1}^{N} \hat{S}_i
\]

\[
SVD(S) = UDV^T
\]

With principal components in U and eigen values

\[
\lambda = \frac{1}{N-1} DD^T
\]

Slide Credit: G. Chintalapani 2010

Principal Component Analysis

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

\[
\bar{S} + \sum_{k=1}^{N-1} U_k \lambda_k
\]

• Compact model \(\rightarrow\) fewer components

• Select first ‘d’ components represented by the ‘d’ eigen values
Statistical Shape and Intensity Models

- Shape statistical model: Mesh vertices become data matrix
  \[ \bar{s} + \sum_{k=1}^{d} U_k \lambda_k = \bar{s} + \bar{U}^T \lambda \]
- Intensity statistical model: Polynomial coefficients become data matrix
  \[ \bar{c} + \sum_{k=1}^{p} Y_k \mu_k = \bar{c} + Y^T \mu \]

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
Typical pipeline for atlas-assisted registration/registration

Image → Statistical Atlas → Deformable model fitting to atlas → Instance of atlas → Deformable registration of model to image → Deformed instance

Deformable model fitting

Statistical Atlas → Create Instance $\Theta(\hat{\rho}, \cdot) →$ Image $\cdot E(\cdot, \cdot)$ $\rightarrow$ Optimization Process $\hat{\rho}^* = \arg\min E(I_{m_1}, \Theta(\hat{\rho}, I_{m_2})) \rightarrow$ Predicted Image

Copyright 2021 R. H. Taylor
Deformable Registration Scheme

- Affine Transformation
  - Translation $T=(t_x, t_y, t_z)$
  - Rotation $R=(r_x, r_y, r_z)$
  - Scale $S=(s_x, s_y, s_z)$ [Similarity if $s_x=s_y=s_z$]

- Global Deformation
  - Statistical deformation mode ($M_i$)

- Local Deformation
  - Adjustment of every vertex

Optimization Algorithm

- Direction Set (Powell’s) method in multi-dimensions
  - Search the parameter space to minimize the cost functions
  - Advantage
    - Don’t need to compute derivative of cost functions
    - Much fewer evaluations than downhill simplex methods

- Alternatives
  - Downhill Simplex (similar advantages)
  - Covariance Matrix Adaptation Evolution Strategy (CMA-ES) method (similar advantages)
  - Levenberg-Marquardt (requires computing gradients)
  - Many others
Local Deformation

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

Multiple-layer Flexible Mesh Template

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

$$(0, \text{Sphere}(v_1^{(1)} - v_0^{(0)}, r_1), \text{Sphere}(v_2^{(1)} - v_0^{(0)}, r_1), \cdots, \text{Sphere}(v_1^{(2)} - v_0^{(0)}, r_2), \text{Sphere}(v_1^{(2)} - v_0^{(0)}, r_2), \cdots)$$
Template matching

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

- Place $\mathbf{v}_k$ at $\mathbf{x}_0$.
- For each neighbor $\mathbf{v}_s$:
  - Find the $\mathbf{x}_j$ near $\mathbf{v}_s$ that minimizes $E(\mathbf{x}_j, \mathbf{v}_s)$.
  - Score $\mathbf{x}_j = E(\mathbf{x}_0, \mathbf{v}_s) + \sum w_k E(\mathbf{x}_j, \mathbf{v}_s)$.
- Pick the $\mathbf{x}_j$ with the best score.

\[ \mathbf{x}_0 \]

Template matching

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

- Place $\mathbf{v}_k$ at $\mathbf{x}_0$.
- For each neighbor $\mathbf{v}_s$:
  - Find the $\mathbf{x}_j$ near $\mathbf{v}_s$ that minimizes $E(\mathbf{x}_j, \mathbf{v}_s)$.
  - Score $\mathbf{x}_j = E(\mathbf{x}_0, \mathbf{v}_s) + \sum w_k E(\mathbf{x}_j, \mathbf{v}_s)$.
- Pick the $\mathbf{x}_j$ with the best score.

\[ \mathbf{x}_0 \]
Template matching

For each pixel location $\mathbf{x}_0$:
Place $\mathbf{v}_0$ at $\mathbf{x}_0$.
For each neighbor $\mathbf{v}_k$:
Find the $\mathbf{x}$ near $\mathbf{v}_k$ that minimizes $E(\mathbf{x}, \mathbf{v}_k)$.
Score $(\mathbf{x}) = E(\mathbf{x}, \mathbf{v}_0) + \sum_k w_k E(\mathbf{x}, \mathbf{v}_k)$.
Pick the $\mathbf{x}$ with the best score.

Score = 4.6 (for $w_k = 0.2$)
Results (Affine Transformation)

Initial  Intermediate  Final

Results (Global Deformation)

Initial  Intermediate  Final
Results (Local Deformation)

Initial | Intermediate | Final

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

Deformable Atlas/CT Registration

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Affine</th>
<th>Global Deform</th>
<th>Local Deform</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21-30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31-40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41-50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51-60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61-70</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71-80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81-90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91-100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Energy Function

Jianhua Yao

Iterative “bootstrapping” of Atlas

Initial Atlas

Augmented 3D/3D Deformable Registration

Bootstrapping loop

Updated Atlas

Update Statistics

Atlas

Mesh instances and warped volumes

Chintalapani et al. MICCAI 2007

Copyright 2021 R. H. Taylor
Leave-Out Validation Experiments

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

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

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

Distribution of Surface Registration Errors
Choice of Initial Template

• Claim:
  – iterative method does not depend on the choice of template

• Criteria:
  – Mean shape converges
  – Modes exhibit similar deformation patterns

• Experimental setup:
  – Three random templates
  – Atlases with and without bootstrapping compared

• Result
  – All three atlases exhibit similar deformation patterns after bootstrapping

Average Difference between Atlases 1, 2, and 3
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
Surface residual error using 18 modes for different sample set sizes

20 dataset atlas
40 dataset atlas
60 dataset atlas
80 dataset atlas

0mm 3mm 6.5mm

Stability Analysis – Mean Shape

Training sample size

Mean shape comparative study

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

Active Appearances

- The material following is based on
- 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 $G$

\[
\|G\| = 1 \\
\sum G_k = 0
\]

Appearance models, con’d

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

\[
G^{(j)} = G^{(avg)} + M_G \cdot \lambda^{(j)} = G^{(avg)} + \sum_{i=1}^{N_M} M_G^{(i)} \cdot \lambda^{(i)} \\
X^{(j)} = X^{(avg)} + M_X \cdot \lambda^{(j)} = X^{(avg)} + \sum_{k=1}^{N_M} M_X^{(k)} \cdot \lambda^{(k)}
\]

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.

Training Set for 2001 Cootes & Taylor paper

- 400 faces
- 68 points
- 10000 intensity values

Complication

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

**Answer:** Form combined vector of shape and intensity variation

\[
Y = W_X \left( X - X^{(\text{avg})} \right) G - G^{(\text{avg})}
\]

where \( W_X \) is a diagonal matrix of weights. Then do PCA on \( Y \).
Further complication

• How do you find the right weights to use?

Answer (from Cootes et al. 1998):

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

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

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

**Face modes**

Fig. 2. First two modes of shape variation ($\pm 3$ std)

Fig. 3. First two modes of grey-level variation ($\pm 3$ std)

Source: Cootes et al. 1998
Face modes

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

Combined

Source: Cootes et al. 1998

Basic Algorithm

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

Basic Iteration of the Method

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

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

Source: Cootes et al. 2001

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

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

Source: Cootes et al. 1998

Results

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

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

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

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

Source: Cootes et al. 1998

Results: Knee Example

Initial  2 it  Converged (11 it)

Fig. 14. Multi-Resolution search for knee

Source: Cootes et al. 1998
Deformable registration between density atlas and a set of 2D X-Rays

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

2D/3D Registration – Shape and Intensity Models

- Shape statistical model (PCA on mesh vertices): \( S^{(0)}, \{S^{(k)}\} \)
- Intensity statistical model (PCA on voxel values and polynomials fit to modes): \( C^{(0)}, \{C^{(k)}\} \)
- Deformable 2D/3D Registration (Estimate \( \lambda^{(s)} \))
- Registered Atlas Projections
- Reestimate Densities (Compute \( \mu^{(b)} \))
- Registered Atlas Projections

2D/3D Registration – Shape and Intensity

<table>
<thead>
<tr>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>$S_{\text{true}}$ (mm)</td>
<td>RMS ($V_{\text{true}} - V_{\text{model}}$) (HU)</td>
<td>RMS($V_{\text{true}} - V_{\text{model}}$) (HU)</td>
<td>$\Delta$ ((3)-(4))/(3)</td>
</tr>
<tr>
<td>1</td>
<td>1.94</td>
<td>100.92</td>
<td>68.88</td>
<td>46.43</td>
</tr>
<tr>
<td>2</td>
<td>1.62</td>
<td>128.32</td>
<td>96.0</td>
<td>25.19</td>
</tr>
<tr>
<td>3</td>
<td>1.90</td>
<td>98.4</td>
<td>77.12</td>
<td>21.63</td>
</tr>
<tr>
<td>4</td>
<td>2.60</td>
<td>51.68</td>
<td>41.6</td>
<td>19.50</td>
</tr>
<tr>
<td>5</td>
<td>2.48</td>
<td>109.14</td>
<td>84.8</td>
<td>22.51</td>
</tr>
<tr>
<td>6</td>
<td>1.95</td>
<td>73.44</td>
<td>50.56</td>
<td>31.15</td>
</tr>
<tr>
<td>7</td>
<td>2.30</td>
<td>72.96</td>
<td>47.52</td>
<td>34.84</td>
</tr>
<tr>
<td>8</td>
<td>2.93</td>
<td>101.28</td>
<td>85.76</td>
<td>15.32</td>
</tr>
<tr>
<td>avg</td>
<td>2.21</td>
<td>93.18</td>
<td>67.78</td>
<td>27.07</td>
</tr>
</tbody>
</table>

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%

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
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

Multi-Component Hip Atlas

PC1
PC2
PC3

2D/3D Registration – Hip Model

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

Atlas projections overlaid on DRR images after registration

2D/3D deformable registration

Chintalapani et al. CAOS 2009

Applications – Hip Osteotomy
Background

• Hip dysplasia:
  – Malformation of the hip (normally a ball and socket joint)
  – Significant cause of osteoarthritis, especially in young adults

• Surgery goals:
  – Reduce pain symptoms
  – Realign joint to contain the femoral head
  – Diminish risk for degenerative joint changes
  – Improve contact pressure distribution

• Periacetabular Osteotomy (PAO):
  – Maintains pelvic structural stability
  – Preserves viable vascular supply
  – Technically challenging tool placement and realignment procedure

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

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)
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

Chintalapani et al. SPIE 2010

---

Atlas Adaptation to Partial Data

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

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

\[
\begin{bmatrix}
S_1 \\
S_2 \\
\end{bmatrix} = \begin{bmatrix}
S_j \\
S_j \\
\end{bmatrix} \quad U = \begin{bmatrix}
U_i \\
U_i \\
\end{bmatrix}
\]

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

\[
(s, R, \tilde{p}, \tilde{\lambda}) = \arg \min_{(s, R, \tilde{p}, \tilde{\lambda})} \| S_j^{(\text{obs})} - (sR(\bar{S} + U_j\tilde{\lambda}) + \tilde{p} \|
\]

Estimate the total shape as

\[
S^{(\text{est})} = \begin{bmatrix}
(sR(\bar{S} + U_j\tilde{\lambda}) + \tilde{p}) \\
S_j^{(\text{obs})}
\end{bmatrix}
\]

Chintalapani et al. SPIE 2010
Atlas Adaptation to Partial Data with Xray Images

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

\[
(s, \mathbf{R}, \mathbf{p}, \lambda) = \arg\max_s \sum_k \text{MI}(I_k, \text{DRR}(\text{DensityAtlas}, s\mathbf{R}(\mathbf{S}_j + \mathbf{U} \lambda) + \mathbf{p}))
\]

- Final atlas extrapolated model is given as

\[
S^{(\text{est})} = \left[ \begin{array}{c} 
\text{sR} \left( \mathbf{S}_j + \mathbf{U} \lambda \right) + \mathbf{p} \\
\mathbf{S}_j^{(\text{obs})}
\end{array} \right]
\]

Chintalapani et al. SPIE 2010

Results

Leave-Out Validation of Partial Data Extrapolation

- Residual Error (mm)
- Number of Principal Components

Copyright 2021 R. H. Taylor
## Results – Atlas Experiments

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

<table>
<thead>
<tr>
<th>#</th>
<th>Full CT</th>
<th>Partial CT</th>
<th>Partial CT + X-ray</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>max</td>
<td>std</td>
</tr>
<tr>
<td>1</td>
<td>1.41</td>
<td>8.20</td>
<td>1.06</td>
</tr>
<tr>
<td>2</td>
<td>1.88</td>
<td>7.25</td>
<td>1.42</td>
</tr>
<tr>
<td>3</td>
<td>1.55</td>
<td>7.72</td>
<td>1.20</td>
</tr>
<tr>
<td>4</td>
<td>1.32</td>
<td>5.77</td>
<td>1.01</td>
</tr>
<tr>
<td>5</td>
<td>1.72</td>
<td>8.29</td>
<td>1.17</td>
</tr>
<tr>
<td>6</td>
<td>1.69</td>
<td>10.58</td>
<td>1.55</td>
</tr>
<tr>
<td>avg</td>
<td>1.59</td>
<td>7.96</td>
<td>1.23</td>
</tr>
</tbody>
</table>

Chintalapani et al. SPIE 2010

Copyright 2021 R. H. Taylor
Cut-and-Paste Model Completion

Model Completion with Thin Plate Spline

Model Completion of Pelvis from Partial CT Only

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

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
### Statistical Assessment of ACL Tunnel Positions

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

![Diagram of knee atlas/CT, 3D-2D registration, x-ray style rendering, tunnel position estimation in 3D, 2D measure, 3D measure]

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

**E-Step**: Estimate $\rho_{mn}$ from current $R, \mathbf{t}, \alpha$, where $\rho_{mn} = p_{mn}$, probability that projected model contour element $m$ matches image contour element $n$.

**M-Step (Pose)**: 
$$\{R, \mathbf{t}\} = \arg \min_{R, \mathbf{t}} \sum_{m,n} \rho_{mn} \| \bar{x}_n - T(\bar{X}_m;R,\mathbf{t}) \|$$

**M-Step (Shape)**: 
$$\bar{\alpha} = \arg \min \rho \sum_{m,n} d_{mn}^2 + (1 - \rho) \sum_{x} \frac{\alpha_x^2}{\lambda_x}$$

where 
$$d_{mn} = \sqrt{\rho_{mn} \bar{V}_n} \left( C_n - \left( R X_m^{(\delta)} + \mathbf{t} + \sum_{x} R \bar{e}_{mn}^{(x)} \right) \right)$$
Basic Approach: Contour-based deformable 2D-3D registration

E-Step: Estimate $p_{mn}$ from current $R, \mathbf{t}, \mathbf{c}$, where $p_{mn}$ is the probability that projected model contour matches image contour element $n$.

M-Step (Pose):
$[R, \mathbf{t}] = \arg \min \sum_{mn} p_{mn} \| \tilde{x}_n - T(\tilde{X}_m; R, \mathbf{t}) \|$

M-Step (Shape):
$\mathbf{c} = \arg \min \rho \sum_{mn} d_{mn}^2 + (1 - \rho) \sum_{k} \frac{c_{nk}^2}{\lambda_k}$
where $d_{mn} = \sqrt{p_{mn} \sum_{i} \mathbf{C}_n - R \hat{X}_m^{(i)} + \mathbf{t} + \sum_{k} R \mathbf{E}_m^{(k)}}$.
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

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

Typical bi-planar phantom used for C-arm calibration

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

C-Arm Distortion Correction

Warped X-ray image of the phantom
Dewarped X-ray image

Distortion vector map

\[\Delta d = (\Delta u, \Delta v) = (u_d, v_d) - (u_0, v_0)\]
Statistical Characterization of C-Arm
Distortion correction using PCA

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

Circular Trajectory

Distortion patterns from PCA modes
C-arm Imaging Volume

mode1  mode2  mode3  mode4

Lambda_1  Lambda_2  Lambda_3  Lambda_4

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.
Sampling Resolution

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

Recovering Distortion Parameters

- Use as few beads as possible to recover the distortion mode parameters
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
Using Patient CT as Fiducial

- Patient C-Arm images with distortion
- Prior CT
- Prior distortion model

2D/3D Registration

- Registered Drrs

Distortion mode matching

- Distortion corrected images

Avg. residual error: 0.5mm/pixel

Chintalapani et al. ISBI 2007

C-Arm Distortion Correction Using Patient CT as Fiducial

- C-Arm images with distortion
- Intra-operative C-Arm
- Patient CT

Initialization

Optimize modes

- CT Projections (DRRs)

Rigid 2D/3D

- Registered Patient CT

Iterative Step

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

Slide credit: Gouthami Chintalapani
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.