Alternative Statistical Methods for Bone Atlas Modelling

Sharmishta Seshamani, Gouthami Chintalapani, Russell Taylor
Department of Computer Science, Johns Hopkins University, Baltimore, MD

Traditional bone atlas modelling is carried out using linear methods such as PCA. Such linear models use a mean shape and principal modes to represent the atlas. A new shape, which is a high dimensional data vector, is then described using this mean and a weighted combination of the principal modes. The use of alternate methods for modelling statistical atlases have not been explored very much. Recently, there has been a lot of new work in the areas of multilinear modelling and nonlinear modelling. They present new ways of modelling high dimensional data. In this work, we compare and contrast several linear, multilinear and nonlinear methods for bone atlas modelling.

1. INTRODUCTION

One of the prominent applications or statistical atlases is anatomical modelling of bones. Bone atlases are typically built by collecting a set of sample bone shapes and then developing a model for representing any bone shape using the variation in the sample set. A new patient’s bone shape can then be compared with the database using shape models derived from the examples in the atlas to carry out a diagnostic analysis. For such a procedure, the first step is to be able to model the shape of the new patient with respect to the shapes in the atlas. However, shape data is often very high dimensional in nature. Thus, dimensional reduction techniques are often used to represent these shapes using a small set of coefficients for comparison. Dimensional reduction techniques however need to ensure that the loss of dimensionality does not contribute to loss of relevant information for comparison of shapes.

Traditional bone atlas modelling is carried out using linear methods such as Principal Components Analysis (PCA). Here, the atlas model consists of a a mean shape and principal modes representing the variation of shapes within the atlas. A new shape, which is a high dimensional data vector, is then described using this mean and a weighted combination of the principal modes of variation.

The use of alternate methods for modelling statistical atlases of anatomy has not been explored very much. The area of multilinear modelling and manifold learning present new ways of modelling high dimensional data. Although more computationally expensive than most linear methods, these methods can be useful for capturing variations in the data that are typically missed out by simple linear models. In this paper, we compare and contrast several linear, multilinear and nonlinear methods for bone atlas modelling.

2. METHODS

In this section, we describe various methods used to model a bone atlas. In all our work, we assume the data consists of a set of shapes whose surface meshes have been been extracted. We then represent each shape as a set of vertices on the mesh. Next, we select one shape in the dataset and compute point correspondences of vertices in all other shapes to this shape. This then allows for a representation of a set of shapes with a set of vertex points. The objective now is to model the variability in the shapes using a small number of dimensions. In the first subsection, we describe PCA which is currently used. In the following subsections, we propose alternative statistical methods for modelling shape variation.

2.0.1 Definitions

We first define a shape made up of $P$ 3D vertex points as a matrix: $S \in \mathbb{R}^{P \times 3}$. An atlas consisting of a set of $N$ shapes is denoted $\{S_i | i = 1 \ldots N, S_i \in \mathbb{R}^{P \times 3}\}$. A new shape not in the dataset used for modelling is denoted $S_{\text{new}}$ and the same shape reconstructed from its low dimensional representation is denoted $S_{\text{rec}}$. The reconstruction can then be compared with the original point to determine the error of the model.
2.1 Linear Methods

2.1.1 Principal Component Analysis

Principal component analysis (PCA), which is also known as Karhunen-Loeve (KL) transform, is a classical linear statistical technique that has been applied to many fields, such as knowledge representation, face recognition and image compression. The main objective of PCA is to supply a lower-dimensional picture of a multivariate dataset when viewed from an informative viewpoint. More specifically, the objective is to determine an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

Let us first define a vector \( Y_i \in \mathbb{R}^{3 \times P} \) to be the vectorized form of the shape matrix \( S_i \in \mathbb{R}^{P \times 3} \). We can then define a data matrix \( Y \in \mathbb{R}^{3 \times PXN} \) as \( Y = [Y_1, \ldots, Y_N] \), where each column represents a shape data point. The empirical mean of all \( Y_i \)s is denoted \( \bar{Y} \). Subtracting this from each column in \( Y \) generates a mean-subtracted data matrix \( X \in \mathbb{R}^{3 \times PXN} \). One can think of the columns of this \( X \) as the bases for describing the space of possible variation. However, the number of columns is the same as the number of datapoints \( N \). What we are interested in determining is if there is a smaller number of bases \( k\ll N \) such that the same space can be represented with only \( k \) bases. This would allow for a more compact representation of a shape.

The problem definition therefore is as follows: Determine the set of orthonormal bases that best describe the column space of \( X \). We investigate 3 methods for performing this task.

2.1.2 Singular Value Decomposition Approach

The Singular Value Decomposition (SVD) of \( X \) generates the set of orthonormal bases of the column and row spaces of the matrix:

\[
X = UDV^T
\]

where columns of \( U \) span the column space of \( X \) and columns of \( V \) span the row space of \( X \). \( D \) is a diagonal matrix containing ordered singular values (highest to lowest) which determine the strength of the bases vectors.

Hence we can select the vectors in \( U \) corresponding to the top \( k \) singular values to determine a set \( B \) consisting of \( k \) basis vectors that best describe the data. The algorithm is outlined in algorithm 1.

2.1.3 Covariance Matrix Approach

Another method for computing these basis vectors is with the use of the covariance matrix \( C = XX^T \). Using the SVD of \( X \), we note that \( XX^T \) can be written as:

\[
XX^T = UDV^T V D^T U^T = UD^2 U^T
\]

which is the eigen decomposition of the covariance matrix. We can therefore note that computing the top \( k \) eigen vectors of the covariance matrix also gives us the set of bases \( B \). The algorithm is outlined in algorithm 2.

2.1.4 Alternative Spectral Method

The third method we consider is based on the spectral theorem. Here, we consider the affinity matrix \( A = X^TX \). The spectral theorem gives:

\[
V^TX^TXV = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}
\]

where \( Q \) is diagonal and positive definite and \( V \) are unitary orthonormal vectors. Partitioning \( V \) appropriately, we can write:

\[
\begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} X^TX \begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} V_1^TX^TXV_1 & V_1^TX^TXV_2 \\ V_2^TX^TXV_1 & V_2^TX^TXV_2 \end{bmatrix} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}
\]

Therefore, \( V_1^TX^TXV_1 = Q \). Define

\[
U_1 = XV_1Q^{-1/2}
\]

Then

\[
U_1Q^{1/2}V_1^T = MV_1Q^{-1/2}Q^{1/2}V_1^T = X
\]
We can then generate \( U \) as:

\[
U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}
\]

where \( U_2 \) can be selected to ensure that \( U \) is unitary and orthonormal. We can now write \( X \) as:

\[
\begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} Q^{1/2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} Q^{1/2}V^T \\ 0 \end{bmatrix} = U_1Q^{1/2}V^T = X
\]

This shows that we can obtain the \( U \) corresponding to column space generated by the SVD of \( X \). Hence, this gives us the same basis \( U \) as in the PCA case. The algorithm is outlined in 3. This method has two advantages. The first is that in the case where the dimension of the data is much larger than the number of data points, the covariance matrix \( XX^T \) would be very large. Instead, in such cases \( X^TX \) would be much smaller and computing its eigen decomposition is therefore more computationally efficient. Secondly, \( X^TX \) can be interpreted as a matrix containing the dot products of all pairs of datapoints. This is an example of an affinity matrix. This allows us to then extend PCA to other methods that use different definitions for the affinity matrix.

### 2.1.5 Compact Representation

Thus, the compact representation model for a point using linear methods is a data mean \( \bar{Y} \) and a set of bases \( B \) that represent principal modes of variation in the mean centered data.

**Algorithm 1** PCA Algorithm with SVD

**Require:** A training set \( Y = [Y_1; Y_2; \ldots; Y_N]^T \), the number of bases \( d \)

1. Compute the empirical mean \( \bar{Y} = \frac{1}{N} \sum_i Y_i \)
2. Construct the mean centered data matrix where \( X_i = Y_i - \bar{Y} \):
   \[
   X = [X_1; X_2; \ldots; X_N]^T
   \]
3. Compute the SVD of \( X = U\Sigma V^T \).
4. Select the first \( d \) vectors of \( U \) to form the set of bases \( B = [u_1 \ldots u_k] \)

**Output:** \( (\bar{Y}, B) \)

**Algorithm 2** PCA Algorithm with Eigen Decomposition of Covariance Matrix

**Require:** A training set \( Y = [Y_1; Y_2; \ldots; Y_N]^T \), the number of bases \( d \)

1. Compute the empirical mean \( \bar{Y} = \frac{1}{N} \sum_i Y_i \)
2. Construct the mean centered data matrix where \( X_i = Y_i - \bar{Y} \):
   \[
   X = [X_1; X_2; \ldots; X_N]^T
   \]
3. Compute the Covariance matrix \( C = XX^T \).
4. Perform an eigen decomposition of \( C = UD^2U^T \) to obtain eigen vectors (columns of \( U \)) and their eigen values in \( D^2 \).
5. Select the first \( d \) vectors of \( U \) to form the set of bases \( B = [u_1 \ldots u_k] \)

**Output:** \( (\bar{Y}, B) \)

### 2.1.6 Projection and Reconstruction

Given a new sample \( S_{\text{new}} \), we can first vectorize it to generate \( Y_{\text{new}} \). We are now interested in obtaining a lower dimensional representation. This can be done by first mean centering the point and then projecting it using each basis vectors in \( B \):

\[
Y_{\text{proj}} = B^T(Y_{\text{new}} - \bar{Y})
\]
Algorithm 3: PCA Algorithm with Spectral Decomposition

Require: A training set $Y = [Y_1; Y_2 \ldots Y_N]^T$, the number of bases $d$

Compute the empirical mean $\bar{Y} = \frac{1}{N} \sum_i Y_i$

Construct the mean centered data matrix where $X_i = Y_i - \bar{Y}$:

$$X = [X_1; X_2; \ldots; X_N]^T$$

Compute the affinity matrix $A = X^T X$

Perform an eigen decomposition $X^T X = V D^2 V^T$ to obtain eigen vectors (columns of $V$) and their eigen values in $D^2$.

Compute $U$ using equations 5 and 7

Select the first $d$ vectors of $U$ to form the set of bases $B = [u_1 \ldots u_k]$

Output: $(\bar{Y}, B)$

We can now estimate a reconstruction of a data point $Y_{rec}$ given its low dimensional representation $Y_{proj}$ and the shape model $(\bar{Y}, B)$ as follows:

$$Y_{rec} = \bar{Y} + B Y_{proj}$$ (10)

In matrix form, the entire operation of projection and reconstruction can be written as:

$$Y_{rec} = \bar{Y} + B (B^T (Y_{new} - \bar{Y}))$$

In the next subsections, we describe a few alternate methods for statistical modelling of data. Each method is an extension of one of the approaches used for PCA.

2.2 Multidimensional Modelling: Tensor PCA

In all the linear methods described in the previous section, the input is required to be a 2D data matrix $Y$. When dealing with a set consisting of 3D vertices, one has to vectorize each matrix in order to populate this data matrix $Y$. Reducing the data into this format may cause a loss of relevant information.

Multilinear methods allow for the extension of PCA onto higher dimensional representations of the data. Let us first define an operator "||" which concatenates two 2D matrices in the third dimension. Intuitively, we can think of this as appending slices to form a cube (Figure 1). We can now consider a 3 dimensional matrix generated by concatenating $N$ PX3 shape matrices in the third dimension as:

$$S = [S_1 || S_2 || \ldots || S_N]$$

This forms a $P \times 3 \times N$ array or tensor. Tensor algebra defines methods for computing SVDs for high dimensional matrices. By inspecting Algorithm 1, we note that we compute the bases by taking the SVD of the data matrix. If our data matrix is now a 3 dimensional matrix, we can apply tensor algebra to compute the higher order SVD (HO-SVD). Appendix 1 shows the method for computing this HO-SVD. The algorithm for generating the model with Tensor PCA is shown in algorithm 4.

2.2.1 Compact Representation

Using this, we obtain $d_1$ basis vectors of length $P$, $d_2$ basis vectors of length 3 and $d_3$ basis vectors of length $N$. This is different from the $d$ basis vectors of length $3P$ that are obtained from PCA. Moreover, each set of bases represent variation by maintaining the 3D structure of the points. In practice, since we want to capture variation in all 3 coordinates, we do not perform dimensional reduction in the second dimension ($d_2 = 3$). Note that in the PCA case, we were essentially computing $d_3$ and the basis vectors of the third dimension of this tensor.
Figure 1: The figure on the left shows the matrix structure used for PCA. The figure on the right shows the structure of the Tensor generated by concatenating each shape $S_i$ in the third dimension. This allows for computing bases that allow for different modes of variation.

**Algorithm 4** Algorithm for computing bases with Tensor PCA

**Require:** A training set $S \in \mathbb{R}^{P \times 3 \times N}$, the number of bases for each dimension $d_1, d_2, d_3$

Compute the empirical mean $\bar{S} = \frac{1}{N} \sum_i S_i$

Construct the mean centered data matrix where $X_i = S_i - \bar{S}$:

$$X \in \mathbb{R}^{P \times 3 \times N}$$

Compute the HO-SVD of $\tilde{X} = \tilde{Z} \times U_1 \times U_2 \times U_3$.

Select the top $d_1$ vectors of $U_1$ to populate the set $B_1$, top $d_2$ vectors of $U_2$ to populate $B_2$ and top $d_3$ vectors of $U_3$ to form $B_3$

**Output:** $(\bar{S}, B_1, B_2, B_3)$
2.2.2 Projection and Reconstruction

Given a new sample $S_{new}$, we are now interested in obtaining a lower dimensional representation. This can be done by first mean centering the point and then projecting it using the set of basis vectors in $B_1, B_2$ and $B_3$. The following steps explain the details of the procedure. First, mean center $S_{new}$:

$$S_{center} = S_{new} - \bar{S}$$

Next, perform an n-mode product (Appendix, section 4.2) of all three basis matrices. (This is carried out by flattening the matrices along the dimension of projection.)

$$S_{proj1} = B_1^T S_{center}$$
$$S_{proj2} = B_2^T S_{proj1}$$
$$S_{proj3} = B_3^T S_{proj2}$$

$$S_{proj} = S_{proj3}$$

In summary, projection is carried out with:

$$Y_{proj} = (S_{new} - \bar{S}) \times B_1^T \times B_2^T \times B_3^T$$

Note that this is simply the higher order extension of equation 9. Next, in order to compute the reconstruction, we perform the following:

$$S_{rec1} = B_1 S_{proj1}$$
$$S_{rec2} = B_2 S_{rec1}$$
$$S_{rec3} = B_3 S_{rec2}$$

$$S_{rec} = S_{rec3} + \bar{S}$$

In summary, reconstruction is carried out with:

$$S_{rec} = \bar{S} + (S_{proj} \times B_1 \times B_2 \times B_3)$$

Note that this is simply the higher order extension of equation 10.

2.3 Nonlinear Methods

High-dimensional data, meaning data that requires more than two or three dimensions to represent, can be difficult to interpret. One approach to simplification is to assume that the data of interest lies on an embedded non-linear manifold within the higher-dimensional space. If the manifold is of low enough dimension then the data can be visualised in the low dimensional space. In such cases, linear modelling methods may not be able to capture the essential properties of the data. With high dimensional data such as the shape models in question, it is impossible to visualize the structure of the datapoints. Therefore, an exploration of nonlinear techniques can give us insight into the structure of the data. In this section we discuss two nonlinear methods which extend the PCA algorithm 3.
2.3.1 KPCA

In the PCA Algorithm 3, we note that the matrix $A = X^T X$ is made up of entries where each entry in the matrix is $A(i,j) = x_i^T x_j$. We can essentially think of each entry as a function which represents the affinity between two points. KPCA exploits the idea of generating affinities by mapping the points $x_i$'s to a different dimension and computing the dot product of the vectors in that space. Since we are only interested in the dot product, the explicit mapping function need not be known.

Instead, any symmetric, positive definite function that takes in two vectors and returns a scalar can be used as a kernel. We can generate a kernel matrix $K$ by populating it as follows:

$$K(i,j) = k(x_i, x_j)$$

where $k$ is the kernel function. Examples of common kernel function used include the polynomial kernel:

$$k(x_i, x_j) = (x_i^T x_j + 1)^n$$

the rbf (radial basis function) kernel:

$$k(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}$$

and the sigmoid kernel:

$$k(x_i, x_j) = \tanh(\kappa(x_i, x_j) + \Theta)$$

Once the Kernel $K$ is computed, the algorithm follows the rest of the steps as the PCA Algorithm 3. It is summarized in Algorithm 5.

Algorithm 5 KPCA Algorithm

Require: A training set $Y = [Y_1; Y_2 \ldots Y_N]^T$, the number of bases $d$, a kernel function $k$

- Compute the kernel matrix $K$, where $K(i,j) = k(Y_i, Y_j)$ and mean center it to obtain $\bar{K}$
- Perform an eigen decomposition of $\bar{K}$ to obtain eigen vectors (columns of $V$) and their eigen values in $D^2$.
- Compute $U$ using equations 5 and 7
- Select the first $d$ vectors of $U$ to form the set of bases $B = [u_1 \ldots u_k]$

Output: $(B,Y)$

2.3.2 Compact Representation

The model for compact representation comprises of the dataset $Y$ and bases $B$.

2.3.3 Projection and Reconstruction

Given a new sample $S_{new}$, we can first vectorize it to generate $Y_{new}$. We apply the kernel function for the test point with each of the $N$ training points:

$$K_{new}(i,1) = k(Y_{new}, Y_i)$$

Next, we apply the bases $B$ to this vector to obtain the projection:

$$Y_{proj} = B^T K_{new}$$

We can now estimate a reconstruction of a data point $Y_{rec}$ given its low dimensional representation $Y_{proj}$ and the shape model as follows. First compute the estimate of the kernel functions

$$K_{rec} = BY_{proj}$$

We now have the reprojected estimate of $k(Y_{test}, Y_i)$ for all $N$ points and we can apply any nonlinear optimization technique to estimate $Y_{rec}$, depending on the selection of the kernel. In this paper, we use the rbf kernel and use Schoelkopf’s fixed point algorithm for this.
2.3.4 Isomap

The Isomap algorithm\(^1\) is based on the method called Multidimensional Scaling (MDS).\(^2,4\) This method is a linear method that can be used for computing modes of a dataset where the given data is in the form of a distance matrix (ie: every entry in the matrix is a distance between two data points in the set). In the simplest case of euclidean distances, it can be shown that the use of this distance matrix is equivalent to the use of the affinity matrix \(X^T X\) after mean centering. We refer the reader to\(^2\) for more details of this derivation.

The important idea to extract is the fact that this can be extended to the use of any distance matrix. This allows us to model nonlinear distances (as opposed to non-linear affinities as KPCA) and use these to compute a lower dimensional representation.

The idea behind the Isomap algorithm is to construct a new kind of distance matrix that computes distances between points on a manifold (rather than simply the euclidean distance) and then apply the eigen analysis just as MDS does. These distances are referred to as geodesic distances and are constructed by assuming local linearity which allows us to approximate the distance between points in a local neighbourhood with the euclidean distance. This is followed by constructing a graph to connect all points and the use of dijstra’s algorithm to compute the distance between any two points in the dataset.

Once the distance matrix is constructed, it’s eigen vectors are computed. The points can then be projected by selection of the top eigen vectors as bases. Due to the nature of this formulation, the model does not have a mean point. Secondly, in order to compute the distance to a new point, the set of training points and the geodesic structure need to be used. Thus, projection of an arbitrary new point cannot be performed independent of the training data. The trained model therefore does not consist of a mean and modes but simply a set of original points, and a set of bases as in the case of KPCA. The algorithm is outlined in Algorithm 6.

**Algorithm 6 Isomap Algorithm**

**Require:** A training set \(Y = [Y_1; Y_2; \ldots; Y_N]^T\), the number of bases \(d\), the number of local neighbors \(n\)

- Compute geodesic distances between all points to populate the distance matrix \(D\).
- Perform an eigen decomposition of \(D\) to obtain eigen vectors (columns of \(V\)).
- Compute \(U\) using equations 5 and 7.
- Select the first \(d\) vectors of \(U\) to form the set of bases \(B = [u_1 \ldots u_k]\).
- Project all input data points using \(B\) to obtain a set of corresponding projected points: \(P = P_1 \ldots P_N\).

**Output:** \((Y, P)\)

2.3.5 Compact Representation

The model contains the original dataset \(Y\) and the projected points \(P\).

2.3.6 Projection and Reconstruction

Given a new sample \(S_{\text{new}}\), we can first vectorize it to generate \(Y_{\text{new}}\). We first compute its \(n\) nearest neighbors in the training set \(G = \{G_1 \ldots G_n | G \in Y\}\) and their corresponding projections \(Q\{Q_1 \ldots Q_n | Q \in P\}\). Compute the weights \(\lambda_i\) such that

\[
Y_{\text{new}} = \sum_i \lambda_i G_i
\]

Project \(Y_{\text{new}}\) to the lower dimensional subspace using \(\lambda\) and \(Q\):

\[
Y_{\text{proj}} = \sum_i \lambda_i Q_i
\]

In order to reproject it to the original dimension, repeat this process backwards starting from \(Y_{\text{proj}}\). First compute its \(n\) nearest neighbors of the projected point \(Y_{\text{proj}} Q' = \{Q'_1 \ldots Q'_n | Q' \in P\}\). Note that this may differ from the set \(Q\) computed above. Compute their corresponding points in the training set \(\{G'_1 \ldots G'_n | G' \in Y\}\). Compute the weights \(\mu_i\) such that

\[
Y_{\text{proj}} = \sum_i \mu_i Q_i
\]
Reconstruct the higher dimensional representation using $\mu$ and $G'$:

$$Y_{rec} = \sum \mu_i G'_i$$

3. EXPERIMENTS AND RESULTS

We experiment with the atlas of a pelvis. The pelvis atlas consists of 90 male patient samples. For each sample, a CT image of the bone was first acquired. This was followed by 3D point extraction. Each bone sample in the pelvis atlas consists of 5496 3D points. Thus, each pelvis sample can be modelled by 164988 values. Next, point-to-point correspondences were established within each atlas such that the $n$th point in all samples correspond. This provides a reason to model bone samples as a set of points.

We then perform a leave-one-out modelling of the atlas using the different models using different parameters and report the average vertex error in mm. Given $M$ samples, the model parameter that was varied is the number of modes in the model (used for projection). In the case of the linear model (PCA), these modes are simply the top eigenvectors of the data matrix $Y$ generated by the training data. The model was then tested on the left out sample by projecting that sample using these modes. The projected sample was then re-projected back to the original dimension and the test error is computed as the mean euclidean distance between original and re-projected points.

We then applied PCA, Kernel PCA (with RBF kernel), Isomap and Tensor PCA to the pelvis atlas and compared the average vertex error of all methods to the baseline (PCA) in mm. Figure 2 shows the average errors for the leave one out validation for all methods compared to the baseline PCA, starting with one mode up to the maximum number of modes (maximum number of samples) for these pelvis experiments. Figure 3 shows the mean shape and the reconstruction error of all points in a color coded surface plot. Blue areas represent low errors and red areas represent high error. The colors are normalized with respect to the highest error. Therefore, we note that tensor PCA and PCA both perform considerably better than the nonlinear methods. We next take a closer look at the errors of PCA versus Tensor PCA. Figure 4 shows the comparison where the colors are now normalized only with respect to the highest error using these two methods. We note that Tensor PCA has much lower errors than PCA.

4. DISCUSSION

In summary we have presented various methods for dimensionality reduction and statistical modelling for a pelvic bone atlas. These models are all based on different derivations of PCA which is the linear method that is currently in use for modelling bone atlases. In particular, we have extended the linear case to a multilinear case (tensor PCA) and two nonlinear cases (Kernel PCA with an RBF kernel and Isomap). The baseline method (PCA) outperforms Kernel PCA with the RBF kernel and Isomap. However, we note that Tensor PCA outperforms the baseline after the first few modes are added. What this means is that maintaining the inherent structure of the bone model is useful for dimensionality reduction. However, given that the nonlinear methods did not perform as well as PCA, we can infer that the structure of data does not lie on a highly nonlinear manifold and thus does not need this type of modelling.

5. ACKNOWLEDGMENTS

This work was supported by NIH grants 1-R01-EB006839-01A1 and 5R21EB008227-02 and JHU internal funds.

REFERENCES


---

Figure 2: PelvisResults: Comparison of different methods with the baseline PCA (blue) method for modelling bone atlases. The graphs show the average point to point error in mm as the number of modes are increased.
Figure 3: Comparison of vertex error with 20 modes for four different methods. From left to right: PCA, Tensor PCA, Kernel PCA and Isomap. The plots are normalized with respect to the highest error in all four reconstructions. The scale on the right shows error in mm.

Figure 4: Comparison of vertex error with 20 modes for PCA (left) and Tensor PCA (Right). The plots are normalized with respect to the highest error in these two reconstructions. The scale on the right shows error in mm.