



Next: [Texture Model Formulation](#) **Up:** [Statistical Models of Shape](#) **Previous:** [Introduction](#)

Subsections

- [Overview](#)
- [Shapes and Landmarks](#)
- [Obtaining Landmarks](#)
- [Shape Alignment](#)
 - [The Procrustes Shape Distance Metric](#)
 - [Aligning a Set of Shapes](#)
- [Modelling Shape Variation](#)
 - [Reducing Non-linearity](#)
 - [Improving Specificity in the PDM](#)
- [Summary](#)

Shape Model Formulation

Overview

The following chapter provides the fundamental concepts and techniques needed to understand the statistical models of shape used in AAMs. First the concept of a *shape* is defined, next - the basis of the mathematical framework - the concept of *landmarks* is treated. The chapter is concluded by demonstrating how shape variation can be efficiently modeled using *principal component analysis*. Effort has been put into making the treatment rich on examples and references to further treatment of the topics.

Shapes and Landmarks

The first matter to clarify is: What do we actually understand by the term *shape*? A starting point could be the few definitions given below:

"A collection of corresponding border points." [\[62\]](#)

"The characteristic surface configuration of a thing; an outline or a contour." [\[1\]](#)

"Something distinguished from its surroundings by its outline." [\[1\]](#)

Though the above captures the characteristics of the term *shape* fairly well; this thesis will adapt the definition by D.G. Kendall [\[20\]](#) and define shape as:

Definition 2 ‡ **Shape** is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.

The term *shape* is - in other words - invariant to Euclidean transformations. This is reflected in figure [4.1](#). The next question that naturally arises is: How should one describe a shape? In everyday conversation, unknown shapes are often described as references to known shapes - e.g. *"Italy has the shape of a boot"*. Such descriptions can obviously not easily be utilized in an algorithmic framework.

Figure: Four exact copies of the same shape, but under different euclidean transformations.



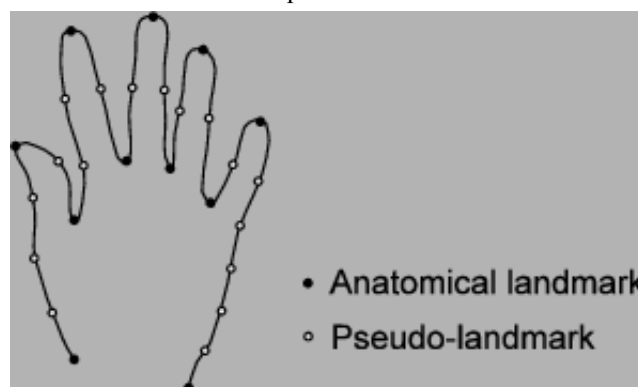
One way to describe a shape is by locating a finite number of points on the outline. Consequently, the concept of a *landmark* is adapted [20]:

Definition 3 \downarrow A **landmark** is a point of correspondence on each object that matches between and within populations.

Dryden & Mardia further more discriminates landmarks into three subgroups [20]:

- **Anatomical landmarks** Points assigned by an expert that corresponds between organisms in some biologically meaningful way.
- **Mathematical landmarks** Points located on an object according to some mathematical or geometrical property, i.e. high curvature or an extremum point.
- **Pseudo-landmarks** Constructed points on an object either around the outline or between landmarks.

Figure: A hand annotated using 11 anatomical landmarks and 17 pseudo-landmarks.



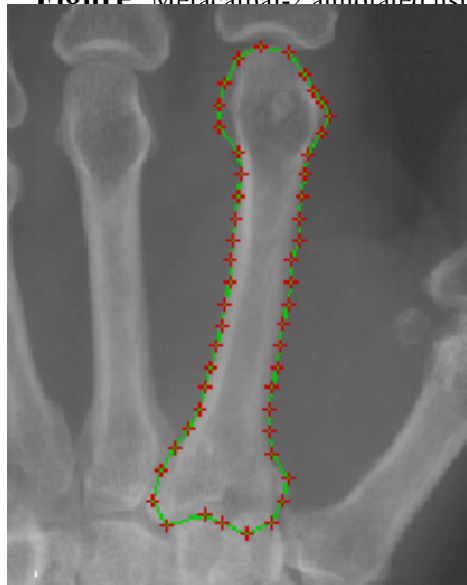
Synonyms for landmarks include homologous points, nodes, vertices, anchor points, fiducial markers, model points, markers, key points etc. A mathematical representation of an n -point shape in k dimensions could be to concatenate each dimension into a kn -vector. In the following only 2D shapes are considered, all though most of the results in the remaining part of the thesis extend directly to 3D - and often even higher dimensionalities. Hence $k=2$. The vector representation for planar shapes would then be:

$$\mathbf{x} = [x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n]^T \quad (4.1)$$

Notice that the above representation does not contain any explicit information about the point connectivity.

Obtaining Landmarks

Although the concept of landmarks conceptually is very useful - the acquisition of such can be very cumbersome. For 2D images the process could involve manually placing of hundreds of points including constantly comparing to other annotations to ensure correspondence. It should be needless to mention that this approach becomes substantially more tedious and cumbersome in the 3D (x,y,z) and 4D $(x,y,z,time)$ case. To ease the burden effort has been put into the development of automatic and semi-automatic placement of landmarks. One could claim that solving the problem of automatic placement of landmarks equals solving *the general correspondence problem* in computer vision. Myriads of attempts have been done regarding that matter. If it successfully could be done one would only need to annotate one "gold" image of the object in question, and the solution to the correspondence problem could solve the object segmentation in this bottom-up fashion. This is - in general - unfortunately not possible. For that reason we need to constrain the solution space somewhat. Defining these constraints - and handling outliers - constitutes the major part of all work in the field of computer vision. One way to constrain the solution space, is to use a manual trained sparse model to initially place the landmark points. If necessary, the points can be corrected manually. Notice however - in the case of basic AAMs - if no adjustments of the points are done, then the training example only adds new texture variation to the model, since the shape itself is a superposition of known shapes. Regarding semi-automatic placement of landmarks several successful attempts have been done. Most of these assume that a dense sampling of the object outline is given beforehand. One example is that of Sclaroff & Pentland [59] where a *finite element model* (FEM) using Galerkin interpolants is built over the set of shape points^{4.1}. The correspondence of a single point to another set of points is determined by comparing the displacement vectors of the point as given by the finite element model. In this way the point set is described in terms of generalized symmetries (i.e. the objects FEM-eigenmodes). One major advantage hereof is that the two point sets can be of unequal sizes. Another example include the work of Duta et al. [21] where *k-means* clustering of the training shapes is performed and followed by a Procrustes analysis of each cluster. Each shape is trimmed into a sparse representation and compared to a dense representation of the remaining shapes. Comparisons are collected into a *pair wise mean alignment matrix* which is used to determine the best point correspondences. Point connectivity is used to increase robustness. Another example of using connectivity information while establishing point correspondences is the work by Andresen & Nielsen [2] where 3D registration solutions is constrained to a surface and an assumption of a non-folding displacement field. This method is called *Geometry-Constrained Diffusion*. Efford [25] identifies landmarks from a dense object contour by estimating the curvature using a gaussian smoothing of the contour representation to obtain robustness from contour noise. Mathematical landmarks are consequently identified as extremums in the curvature function. Semi-landmarks are interpolated as uniformly spaced points between the mathematical landmarks. Quite recently^{4.2} Walker et al. [71] proposed an iterative algorithm for determine point correspondence. This was accomplished using feature vectors for each pixel inside a manually drawn region of interest (ROI) of each training image. Feature vectors were first and second order normalized Gaussian partial derivatives. It was shown that AAMs trained on the automatically generated training set could be of higher quality than AAMs built on hand annotated training sets. However, since AAMs consider both shape and texture as object class descriptors we suggest that the point correspondence determination should not solely rely on changes in curvature or direction of FEM-eigenmode displacement vectors. Solutions should further be constrained by including information of the textural variation around the points. This will lead to better models.

Figure: Metacarpal-2 annotated using 50 landmarks.

Another substantial problem in obtaining landmarks is that some object classes lack points, which can be classified as corresponding across examples. This is especially true for many biological shapes and is treated in depth by Bookstein [6]. Another source for this type of problems is occlusions in the 3D to 2D projections in perspective images. Annihilation of points can also be observed in malformation of organic shapes. All examples in the remains of this part of the thesis are based on annotations of a bone in the human hand. The image modality is radiographs and the precise name of the bone is *metacarpal-2*. An example of such an annotation is given in fig. 4.3. For further information on AAMs on metacarpals, refer to the experimental part of this thesis. As a concluding remark, one should remember that annotations by human experts itself contains errors. This is the core problem in obtaining the so-called *gold standards* to evaluate medical image analysis^{4.3} techniques against. To evaluate this type of noise, annotations are often done several times by several graders to assess the *between* grader and *within* grader variation. This is also known as the *reproducibility* and *repeatability*.

Shape Alignment

To obtain a true shape representation - according to our definition - location, scale and rotational effects need to be filtered out. This is carried out by establishing a *coordinate reference* - w.r.t. position, scale and rotation, commonly known as *pose* - to which all shapes are aligned. Some literature also operates with the concept of *pre-shape* as introduced by Kendall [20]. Pre-shape is the last step toward true shape - rotational effects still need to be filtered out. Below an alignment procedure for obtaining such a coordinate reference is described. This is commonly known as *Procrustes analysis*^{4.4} [6,14,20,35]. To aid the understanding and handling of a set of shapes from the same object class the term *shape space* is introduced. Adapted to our nomenclature from [20] this is defined as:

Definition 4 \downarrow The **Shape Space** is the set of all possible shapes of the object in question. Formally, the shape space Σ_k^n is the orbit shape of the non-coincident n point set configurations in the \mathbb{R}^k under the action of the Euclidean similarity transformations.

If k denotes the Euclidean dimensions and n denotes the number of landmarks, the dimension of the shape space, follows from the above definition:

$$M = kn - k - 1 - \frac{k(k-1)}{2} \quad (4.2)$$

Proof Initially we have kn dimensions. The translation removes k dimensions, the uniform scaling one dimension and the rotation $\frac{1}{2}k(k-1)$ dimensions.

If a relationship between the distance in shape space and Euclidean distance in the original plane can be established, the set of shapes actually forms a Riemannian manifold containing the object class in question. This is also denoted as the *Kendall shape space* [6]. This relationship is called a *shape metric*. Often used shape metrics include the Hausdorff distance [42], the strain energy [59] and the Procrustes distance [21,20,6,14]. Where the two former compare shapes with unequal amount of points, the latter requiring corresponding point sets. In the following, the Procrustes distance is used.

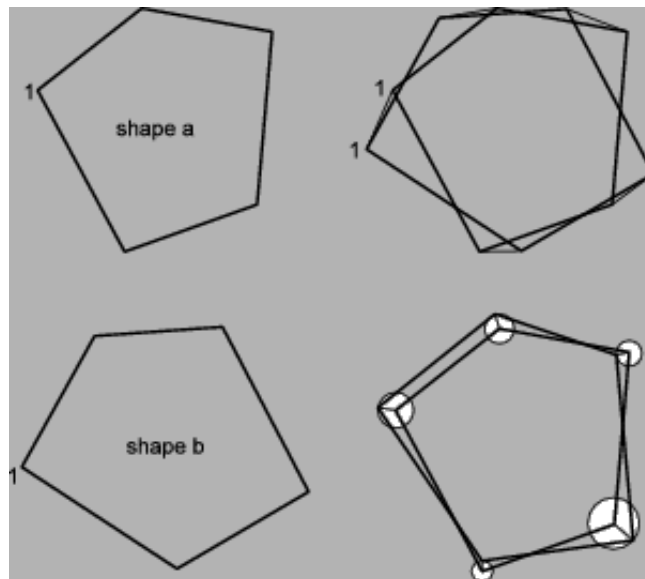
The Procrustes Shape Distance Metric

The Procrustes distance is a least-squares type shape metric that requires shapes with one-to-one point correspondence. To determine the Procrustes distance between two shapes involves four steps:

1. Compute the centroid of each shape.
2. Re-scale each shape to have equal size.
3. Align w.r.t. position the two shapes at their centroids.
4. Align w.r.t. orientation by rotation.

The rotational step and the graphic interpretation of the Procrustes distance can be seen on fig. 4.4.

Figure: The Procrustes distance.



Mathematically the squared Procrustes distance between two shapes, \mathbf{x}_1 and \mathbf{x}_2 , is the sum of the squared point distances after alignment:

$$P_d^2 = \sum_{j=1}^n [(x_{j1} - x_{j2})^2 + (y_{j1} - y_{j2})^2] \quad (4.3)$$

The *centroid* of a shape can be interpreted as center of mass of the physical system consisting of unit masses at each landmark. Thus to compute the centroid:

$$(\bar{x}, \bar{y}) = \left(\frac{1}{n} \sum_{j=1}^n x_j, \frac{1}{n} \sum_{j=1}^n y_j \right) \quad (4.4)$$

To perform step 2 we obviously need to establish a *size metric*:

Definition 5 ‡ A **shape size metric** $S(\mathbf{x})$ is any positive real valued function of the shape vector that fulfils the following property:

$$S(a\mathbf{x}) = aS(\mathbf{x})$$

In the following the *Frobenius norm* is used as a shape size metric:

$$S(\mathbf{x}) = \sqrt{\sum_{j=1}^n [(x_j - \bar{x})^2 + (y_j - \bar{y})^2]} \quad (4.5)$$

Another often used scale metric is the *centroid size*^{4.5}:

$$S(\mathbf{x}) = \sum_{j=1}^n \sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2} \quad (4.6)$$

To filter out the rotational effects the following *singular value decomposition* technique is used as suggested by Bookstein [6]:

1. Arrange the size and position aligned \mathbf{x}_1 and \mathbf{x}_2 as $n \times k$ matrices^{4.6}.
2. Calculate the SVD, \mathbf{UDV}^T , of $\mathbf{x}_1^T \mathbf{x}_2$
3. Then the rotation matrix needed to optimally superimpose \mathbf{x}_1 upon \mathbf{x}_2 is \mathbf{VU}^T . In the planar case:

$$\mathbf{VU}^T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (4.7)$$

As an alternative Cootes et al. suggest a variation on Procrustes distance-based alignment by minimizing the closed form of $|T(\mathbf{x}_1) - \mathbf{x}_2|^2$ where T in the Euclidean case is:

$$T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (4.8)$$

The term $|T(\mathbf{x}_1) - \mathbf{x}_2|^2$ is then simply differentiated w.r.t. (a, b, t_x, t_y) . The solution to alignment using the affine transformation is also given. Notice however that this transformation changes the actual shape. Refer to [14] for the calculations. This concludes the topic of how to provide a consistent metric in shape space and how to align two shapes.

Aligning a Set of Shapes

All though an analytic solution exists [41] to the alignment of a set of shapes the following simple iterative approach suggested by Bookstein et al. [6,14] will suffice.

1. Choose the first shape as an estimate of the mean shape.
2. Align all the remaining shapes to the mean shape.
3. Re-calculate the estimate of the mean from the aligned shapes
4. If the mean estimate has changed return to step 2.

Convergence is thus declared when the mean shape does not change significantly within an iteration. Bookstein notes that two iterations of the above should be sufficient in most cases. The remaining question is how to obtain an estimate of the mean shape? ^{4.7} The most frequently used is the *Procrustes mean shape* or just the *Procrustes mean*: If N denotes the number of shapes:

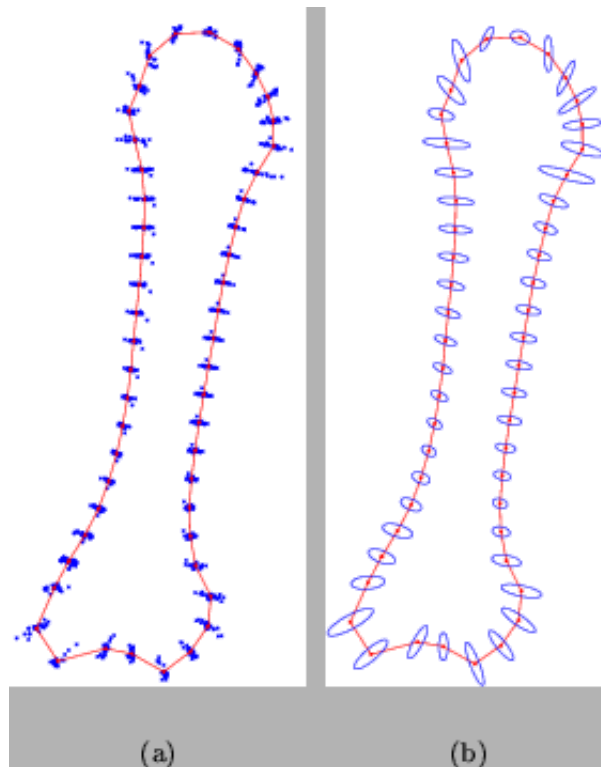
$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad (4.9)$$

This is also referred to as the Frechét mean.

Figure: A set of 24 unaligned shapes. Notice the position-outlier to the right.



Figure: (a) The PDM of 24 aligned shapes. (b) Ellipsis fitted to the single point distribution of figure (a).

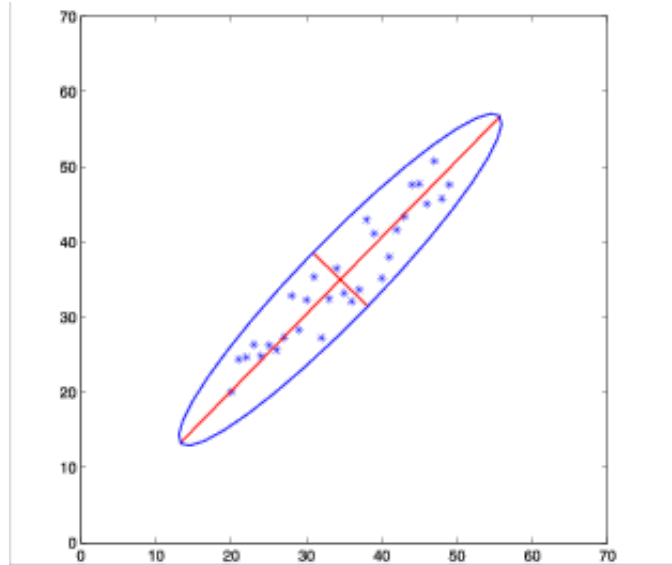


As an example figure [4.5](#) shows the landmarks of a set of 24 unaligned shapes. The result of the shape alignment can be seen as a scatter plot on figure [4.6](#) (a) where the mean shape is superimposed as a fully drawn shape. This is called the *point distribution model* (PDM) of our shapes. How to model the variation within the PDM is the topic of the forthcoming section. To give a more clear impression of the point variation over the set of shapes, an ellipsis has been fitted to each mean model point in figure [4.6](#) (b). [4.8](#)

Modelling Shape Variation

As the previous sections have considered the definition and handling of shapes, this section will demonstrate how intra-class shape variation can be described consistently and efficiently. The fact alone that equivalence classes of shapes can be established - e.g. "*We have a collection of shapes formed as leaves.*" - hint us in the direction that there must be some sort of inter-point correlation present. Naturally, as this actually is the only degrees of freedom left to constitute the perception of a shape, since - according to the definition of shape - all position, scale and rotational effects are filtered out. A classical statistical method of dealing with such redundancy in multivariate data is the linear orthogonal transformation; *principal component analysis* (PCA). Based on work by Karl Pearson the principal component analysis method was introduced by Harold Hotelling in 1933 [54]. The principal component analysis is also known as the *Karhunen-Loeve transform*.

Figure: Principal axis. 2D example.



Conceptually the PCA performs a *variance maximizing rotation* of the original variable space. Furthermore, it delivers the new axes ordered according to their variance. This is most easily understood graphically. In figure 4.7 the two principal axes of a two dimensional data set is plotted and scaled according to the amount of variation that each axis explains. Hence, the PCA can be used as a dimensionality reduction method by producing a projection of a set of multivariate samples into a subspace constrained to explain a certain amount of the variation in the original samples. One application of this is visualization of multidimensional data. 4.9 In connection to the example in figure 4.7 one could choose to discard the second principal axis, and visualize the samples by the orthogonal projection of the point upon the first (and largest) axis. Another application of PCA is to determine any underlying variables or to identify intra-class clustering or outliers. In our application of describing shape variation by using PCA a shape of n points is considered a data point in a $2n^{\text{th}}$ dimensional space. But as stated above it is assumed that this space is populated more sparsely than the original $2n$ dimensions. It has been seen in eq. (4.2) that the reduction should be at least $k - 1 - \frac{1}{2}k(k - 1)$ due to the alignment process. In practice the PCA is performed as an eigenanalysis of the covariance matrix of the aligned shapes. The latter is also denoted the *dispersion matrix*. It is assumed that the set of shapes constitute some ellipsoid structure of which the centroid can be estimated 4.10:

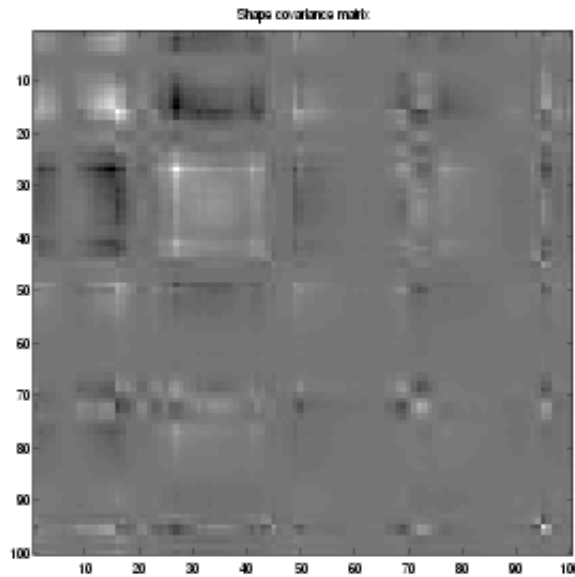
$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad (4.10)$$

The maximum likelihood (ML) estimate of the covariance matrix can thus be given as:

$$\Sigma_g = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \quad (4.11)$$

To prove the assumption of point correlation right, the correlation matrix of the training set of 24 metacarpal-2 bones is shown in figure 4.8. In the case of completely uncorrelated variables, the matrix would be uniformly gray except along its diagonal. Clearly, this is not the case.

Figure: Shape covariance matrix. Black, grey & white maps to negative, none & positive covariance.



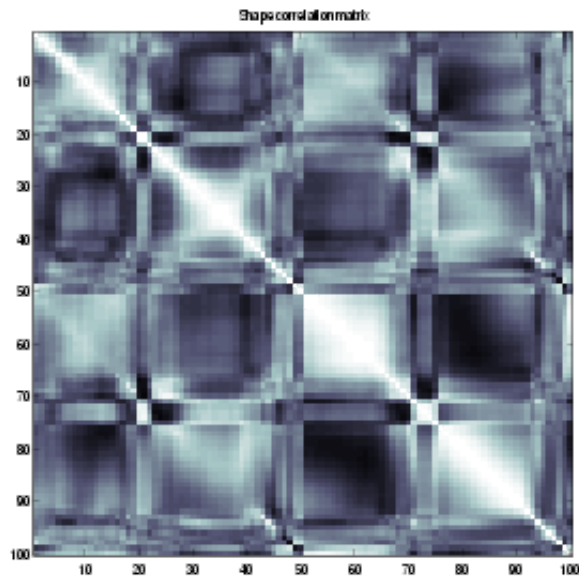
The point correlation effect can be emphasized by normalizing the covariance matrix by the variance. Hence the *correlation matrix*, Γ , is obtained.

$$\mathbf{V} = \text{diag}\left(\frac{1}{\sqrt{\text{diag}(\Sigma)}}\right) = \begin{bmatrix} \frac{1}{\sigma_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\sigma_n} \end{bmatrix} \quad (4.12)$$

$$\Gamma = \mathbf{V}\Sigma\mathbf{V}^T \quad (4.13)$$

Recalling the shape vector structure; $xxyy$; it is from figure 4.9 - not surprisingly - seen that the x - and y -component of each point is somewhat correlated.

Figure: Shape correlation matrix. Black, white maps to low, high correlation.



The principal axes of the $2n^{th}$ dimensional shape ellipsoid are now given as the eigenvectors, Φ_s , of the covariance matrix.

$$\Sigma_s \Phi_s = \Phi_s \Lambda_s \quad (4.14)$$

Where Λ_s denotes a diagonal matrix of eigenvalues

$$\Lambda_s = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{2n} \end{bmatrix} \quad (4.15)$$

corresponding to the eigenvectors in the columns of Φ_s .

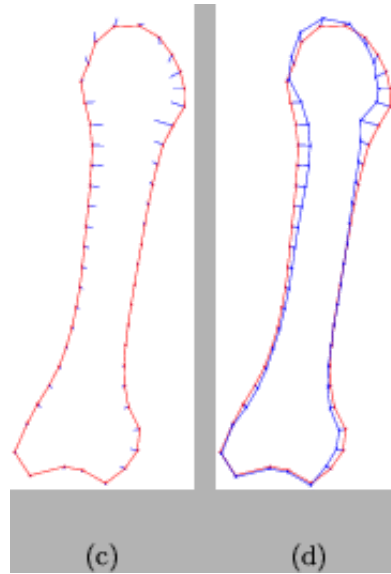
$$\Phi_s = \begin{bmatrix} \phi_1 & \dots & \phi_{2n} \end{bmatrix} \quad (4.16)$$

A shape instance can then be generated by deforming the mean shape by a linear combination of eigenvectors:

$$\mathbf{x} = \bar{\mathbf{x}} + \Phi_s \mathbf{b}_s \quad (4.17)$$

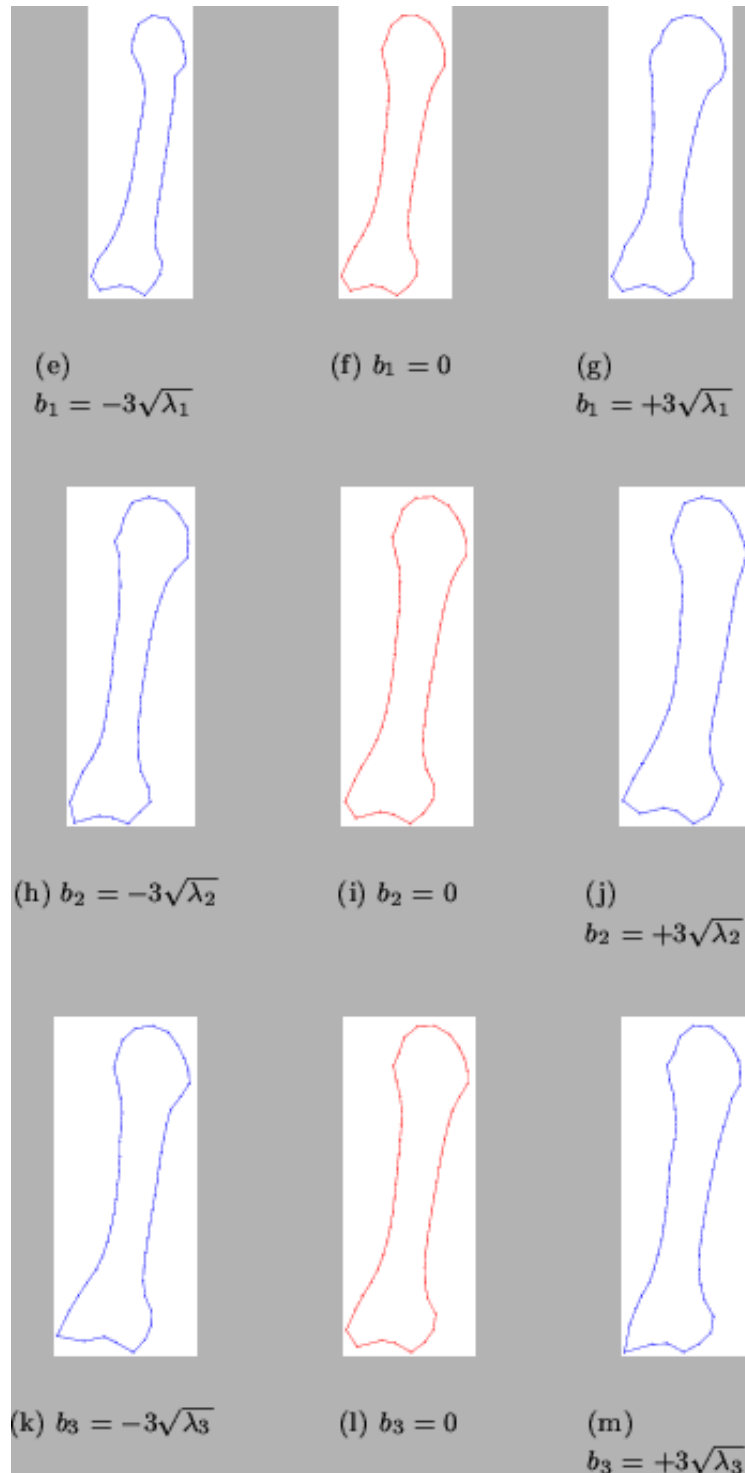
where \mathbf{b}_s is shape model parameters. Essentially the point or *nodal representation* of shape has now been transformed into a *modal representation* where modes are ordered according to their *deformation energy* - i.e. the percentage of variation that they explains. Notice that an eigenvector is a set of displacement vectors, along which the mean shape is deformed. To stress this point, the first eigenvector has been plotted on the mean shape in figure 4.10 (a). The resulting deformation of the mean shape can be seen in figure 4.10 (b).

Figure: (a) Mean shape and deformation vectors of the 1st eigenvector. (b) Mean shape, deformation vectors of the 1st eigenvector and deformed shape.



As a further example of such modal deformations, the first three - most significant - eigenvectors are used to deform the mean metacarpal shape in figure 4.11.

Figure: Mean shape deformation using 1st, 2nd and 3rd principal mode. $b_i = -3\sqrt{\lambda_i}$, $b_i=0$, $b_i = 3\sqrt{\lambda_i}$.



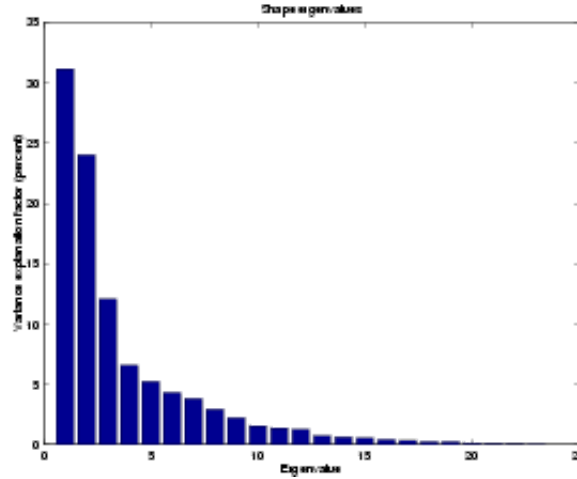
What remains is to determine how many modes to retain. This leads to a trade-off between the accuracy and the compactness of the model. However, it is safe to consider small-scale variation as noise. It can be shown that the variance along the axis corresponding to the i^{th} eigenvalue equals the eigenvalue itself, λ_i . Thus to retain p percent of the variation in the training set, t modes can be chosen satisfying:

$$\sum_{i=1}^t \lambda_i \geq \frac{p}{100} \sum_{i=1}^{2n} \lambda_i \quad (4.18)$$

Notice that this step basically is a regularization of the solution space. In the metacarpal case 95% of the shape variation can be modeled using 12 parameters. A rather substantial reduction since the shape

space originally had a dimensionality of $2n = 2 \times 50 = 100$. To give an idea of the decay rate of the eigenvalues a percentage plot is shown in figure [4.12](#).

Figure: Shape eigenvalues in descending order.

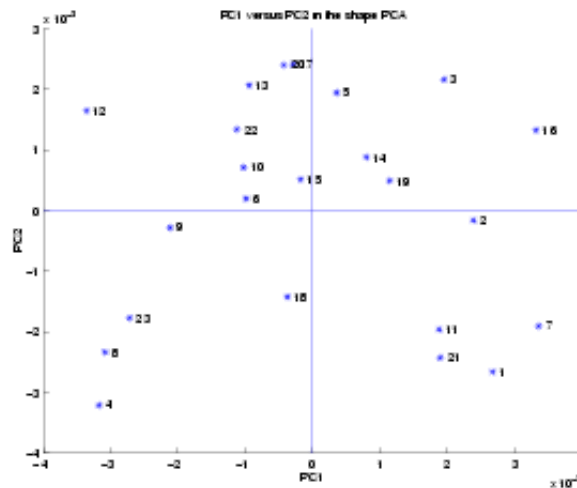


To further investigate the distribution of the \mathbf{b}_s -parameters in the metacarpal training set $b_{s,2}$ is plotted as a function of $b_{s,1}$ in figure [4.13](#). These are easily obtained due to the linear structure of (4.17) and since the columns of Φ_s are inherently orthogonal.

$$\mathbf{b}_s = \Phi_s^{-1}(\mathbf{x} - \bar{\mathbf{x}}) = \Phi_s^T(\mathbf{x} - \bar{\mathbf{x}}) \quad (4.19)$$

No clear structure is observed in figure [4.13](#), thus concluding that the variation of the metacarpal shapes can be meaningfully described by the linear PCA transform. This however is not a general result for organic shapes due to the highly non-linear relationships observed in nature.

Figure: PC1 ($b_{s,1}$) vs. PC2 ($b_{s,2}$) in the shape PCA.



An inherently problem with PCA is that it is linear, and can thus only handle data with linear behavior. An often seen problem with data given to a PCA is the so-called *horse-shoe effect*, where pc1 and pc2 is distributed as a horse-shoe pointing either upwards or downwards^{[4.11](#)}. This simple non-linearity in data - which can be interpreted as a *parabola bending* of the hyper ellipsoid - causes

the PCA to fail in describing the data in a compact and consistent way, since the data structure can not be recovered using linear transformations only. This topic is treated in depth later on.

This section is concluded by remarking that the use of the PCA as a statistical reparametrisation of the shape space provides a compact and convenient way to deform a mean shape in a controlled manner similar to what is observed in a set of training shapes. Hence the shape variation has been modeled by obtaining a compact shape representation. Furthermore the PCA provides a simple way to compare a new shape to the training set by performing the orthogonal transformation into **b**-parameter space and evaluating the probability of such a shape deformation. This topic is treated in depth in section [12.2](#) - Performance Assessment.

Reducing Non-linearity

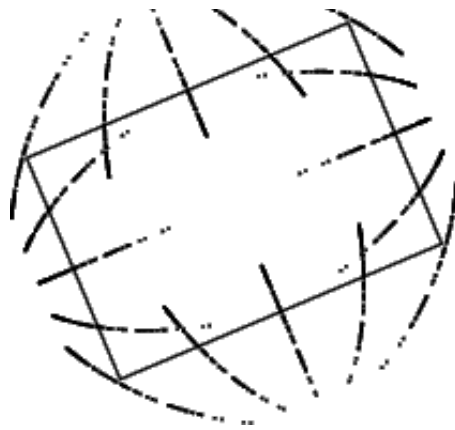
One source of non-linearity in the shape model is the alignment procedure. In the alignment procedure described earlier the shapes were size-normalized by scaling to unit scale using $1/S(\mathbf{x})$. In this way, the corners of a set of aligned rectangles with varying aspect ratio forms a unit circle (see fig. [4.15](#), the unaligned shapes are shown on fig. [4.14](#)). Due to this non-linearity the PCA on the shapes must use two parameters to span the shape space: $\lambda_1 = 99.6\%$, $\lambda_2 = 0.4\%$ even though variation only exists on one parameter (the aspect ratio). A closer look at figure [4.15](#) also shows that the overlaid mean shape does not correspond to an actual shape in the training set. To avoid this non-linearity in the aligned training set the shape can be projected into *tangent space* by scaling by $1/(\mathbf{x} - \bar{\mathbf{x}})$ [[12,14](#)].

Figure: Training set of 100 unaligned artificially generated rectangles containing 16 points each.



Figure: Point cloud from aligned rectangles sized to unit scale, $|\mathbf{x}| = 1$. The mean shape is fully shown.





The projection into tangent space align all rectangles with corners on straight lines (see fig. 4.16) thus enabling modeling of the training set using only linear displacements. Notice how the mean shape is contained in the training set since the PCA now only uses one parameter, $\lambda_1 = 100\%$, to model the change in aspect ratio. In this way, the distribution of PCA-parameters can be kept more compact and non-linearities can be reduced. This leads to better and simpler models.

Figure: Point-cloud from aligned rectangles sized to unit scale, $|\mathbf{x}| = 1$, and transformed into tangent space. The mean shape is fully shown.



Improving Specificity in the PDM

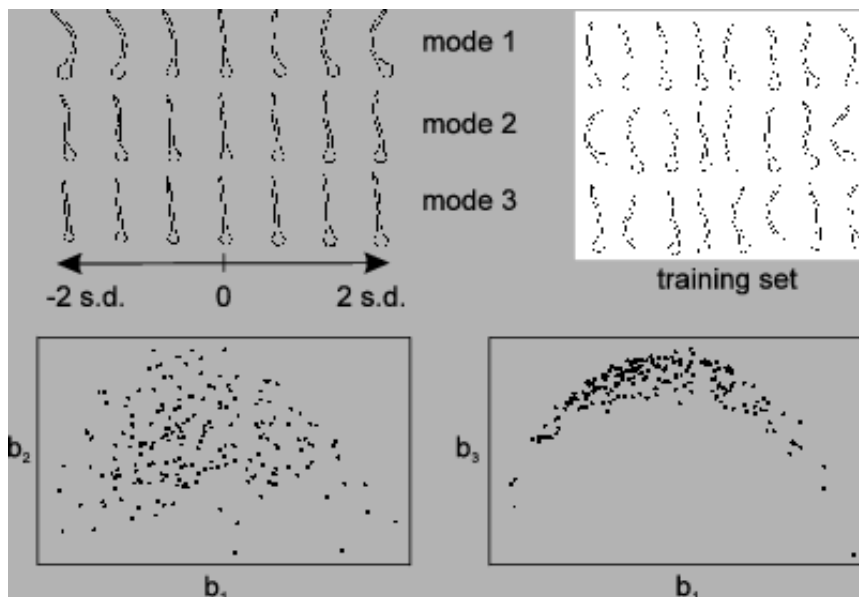
Aside the alignment procedure, several factors can contribute to the breakdown of the PCA, due to non-linearities.

- **Articulated shapes** Shapes with pivotal rotations around one or more points are inherently non-linear.
- **Bad landmarks** Manually placed landmarks can easily cause non-linearities.
- **Bending** Can also be interpreted as a piece-wise rotation.

Examples of the breakdown includes the tadpoles, watch model and chromosomes of Sozou et al. [63,64]. Chromosomes also constituted the original example of a PDM breakdown in [15]. Examples of the tadpole model are given in figure 4.17. Here a clear non-linear dependency between b_1 and b_3 (lower right) is seen, which also is clearly reflected in the deformations given by the principal modes (upper left). This behavior has been coined the *horse-shoe effect*, and serves as an example on structure that can't be decomposed by the linear PCA, namely one of the most simple non-linear

dependencies one can think one; the quadratic.

Figure: Tadpole example of a PCA breakdown. Notice in mode 1, how the head size and length is correlated with the bending. This is easily seen in the scatter plot of PCA parameter 1 vs. 3 (lower right), where b_3 has a simple non-linear dependency of b_1 . Adapted from [64].



Another way to view the problem, is that the PCA-approach is based on the assumption, that all shapes of the class ends on the same manifold. More precisely as a hyper ellipsoid cluster in the new basis spanned by the PCA. However when dealing with non-linearity the ellipsoid changes into a more structured form. Dealing with objects with discretized behavior^{4.12} for example, also changes the ellipsoid, here into a clustered distribution of PCA parameters. To accommodate this, it is proposed to approximate the distribution with a mixture of gaussian blobs in shape parameter space [12] thus avoiding illegal shapes in the PDM. This is accomplished using expectation maximization to fit the blobs to the shape parameters of the training set. The problem of implausible shapes in non-linear shapes has also been addressed by Heap & Hogg [38] using polar coordinates in the PDM without loss of computational performance. The algorithm automatically classifies landmarks into either the Cartesian or polar domain. One of the early attempts include Sozou et al. where polynomial regression (PRPDM) was used to fit higher order polynomials to the non-linear axis of the training set - see figure 4.17. Later Sozou et al. out-performed the PRPDM by using a back propagation neural network^{4.13} to perform non-linear principal component analysis (MLPPDM). The downside to this approach is a substantial increase in the - off-line - computation of the PDM. The bottom line of all this is - if your shape parameters lie in more than one cluster or if dependencies between shape parameters exist - then the standard PDM is not specific enough. This can lead to more or less serious artifacts.

Summary

Throughout this chapter, a complete mathematical framework and the necessary set of definitions and concepts have been introduced to the extent that an efficient treatment of shape and shape modeling can be done. Further more selected topics have been discussed to increase the understanding of the AAM framework. Emphasis has been put on the application in the Active Appearance Models though the methods presented are applicable in a wide range of situations.



Next: [Texture Model Formulation](#) **Up:** [Statistical Models of Shape](#) **Previous:** [Introduction](#)

Active Appearance Models (be)
2000-09-20