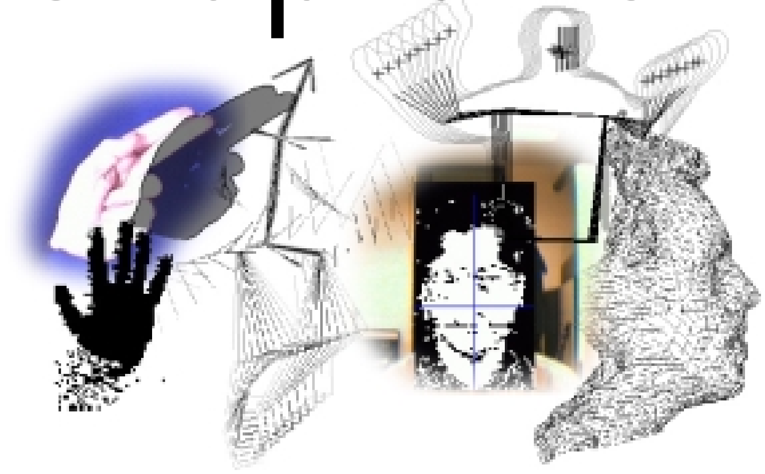
Chapter 5



5 Cluster Based Non Linear Point Distribution Models

5.1 Introduction

As was already mentioned in chapter 2, the major drawback with models which rely upon principal component analysis to model deformation is the non-linearity which is introduced either as natural curvature, inherent to the model, or introduced during the alignment and construction process of the PDM. This non-linearity within shape space (or PCA space) results in poor performance due to the linear nature of the underlying mathematics.

Bregler and Omohundro proposed estimating non-linearity by breaking PCA space down into piecewise linear clusters which could then be modelled with multiple hyperplanes [Bregler 94]. More details on this technique are discussed section 5.4. However, these *Constraint Surfaces* do not place any limits upon the local linear patches within the model and hence the surface extends to infinity producing un-specific models. The work of Bregler also concentrates on extremely low dimensional shape spaces with minimum non-linearity, where little concern is given to the application of computationally expensive techniques. In practice, the technique does not perform well in high dimensional spaces (as will be shown) due to both the computational complexity of cluster analysis and PCA, in addition to the problems associated with discontinuous shape spaces[1].

The remainder of this chapter will propose an alternative approach, which, although similar in nature, produces a more specific model. The construction of such models along with the parameter selection will also be discussed. Section 5.3 will present the use of dimensional reduction techniques to disregard redundancy in high dimensional data, allowing analysis to be performed in lower dimensional spaces. Section 5.4 will discuss the method behind piecewise linear approximations. Section 5.5 will then demonstrate the use of the technique with example data sets. Section 5.6 will discuss the application of the model. Finally the technique will be evaluated and compared to other approaches in section 5.7 and conclusions drawn.

5.2 An Example of non-linearity

One of the classic examples within the field of neural networks is that of a helical data set. Helical datasets are often used to assess a neural network's ability at creating a non-linear mapping. Figure 5.2.1 shows a helix in three dimensions from a front and plan view. Although the helix exists in 3D, it is actually a one-dimensional data set, and can be smoothly paramertised by a single value if the primary non-linear axis, which follows the path of the helix, can be extracted.

[1] see Figure 5.4.5 and associated text for details

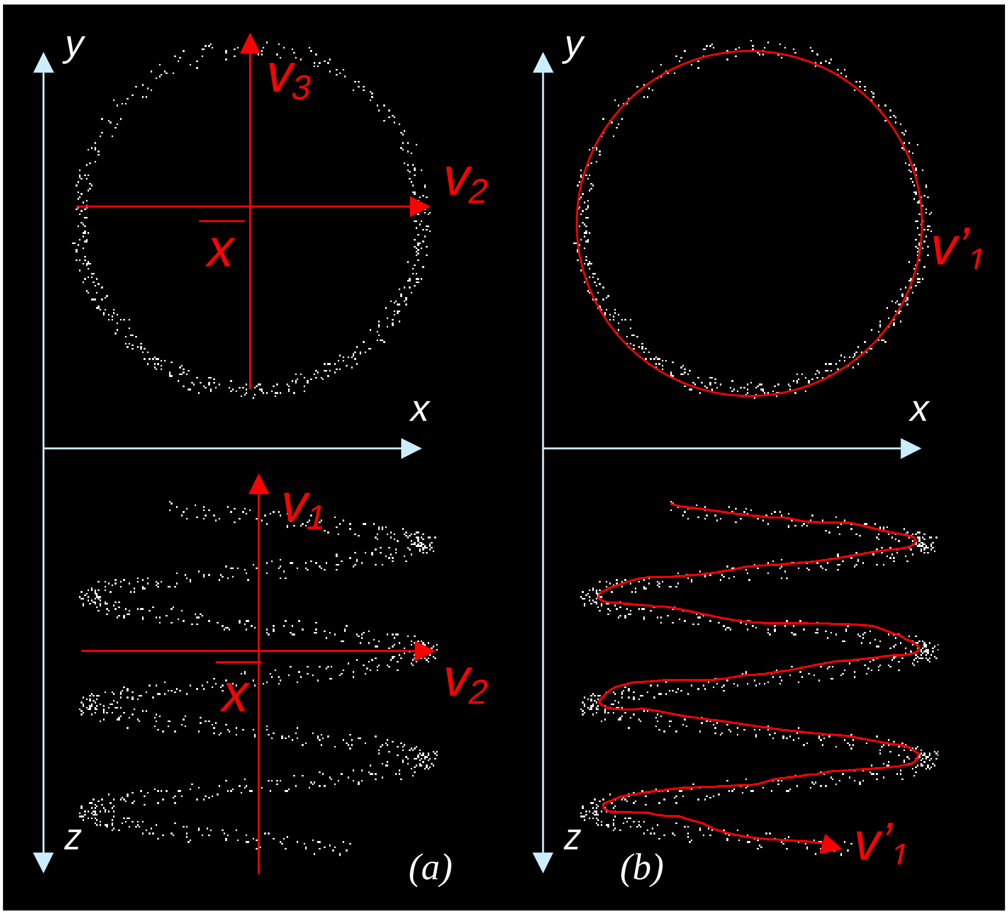


Figure 5.2.1 - Linear PCA, three-dimensional helical data set

(a) a helix using three orthogonal axis, (b) a single non-linear axis

Figure 5.2.1(a) shows the helix represented using three primary axes as determined by linear principal component analysis: $\bar{x}$ is the mean value and exists outside the bounds of the helix, the vectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ are the three orthogonal axes as extracted through PCA. The helix does not lie on any single axis and all three must be used in order to reproduce the path of the helix.

In terms of shape space, where the primary concern is to encompass the bounds of a training set in the most compact and constrained way possible, this is an extremely inaccurate representation as both the mean shape and primary modes are not indicative of the training set shape (ie the helix). Using this linear approach would not only allow paths to be produced which are indicative of the helix but many other non-representative paths within the volume bounded by the vectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$.

Figure 5.2.1(b) shows the helix parameterised by a single non-linear axis which closely follows the path of the helix. Any point on the helix can be represented by a single parameter which indicates the distance along this primary axis from some origin. In order to accurately represent the non-linear data set, a means of extracting the non-linenear axis is required. Unfortunately the data set is seldom parameterised by a single axis and the problem of extraction is compounded by the high dimensional nature of computer vision applications.

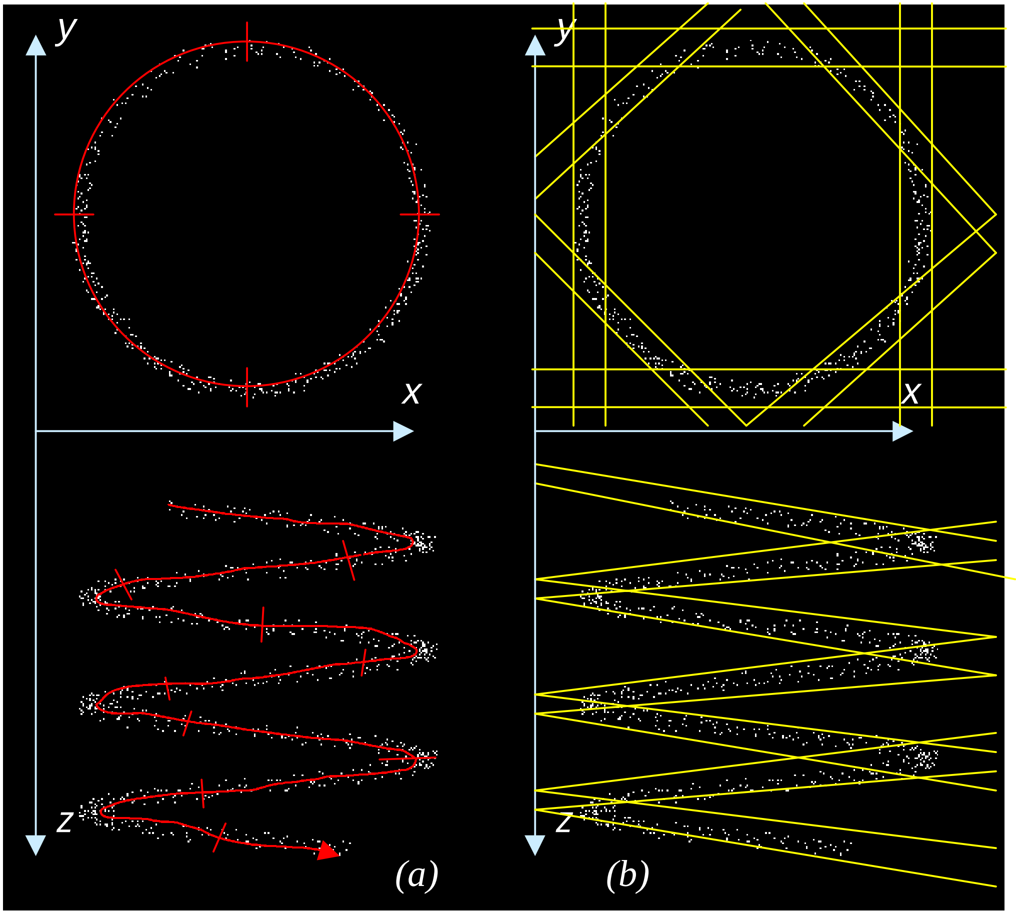


Figure 5.2.2– Non-linear PCA, three dimensional helical dataset

(a) non-linear modes of variation (b) segmenting shape space with multiple planes

Figure 5.2.2(a) shows a secondary axis fitted to the data set. Here, the secondary mode changes dependent upon the position along the primary axis. The fitting, therefore, becomes a computationally expensive process in even the lowest of dimensional spaces. Figure 5.2.2(b) shows how the space can be segregated through the use of multiple hyper-planes. This is akin to the procedure used by a neural network when fitting to a data set. Although faster than attempting to fit true curved axis to the data, it is essentially **estimating** the curvature to a specified degree and hence has a loss in accuracy. This procedure also becomes an infeasible approach as the dimensionality of the space increases. In order to find a suitable technique for performing non-linear PCA, two considerations must be addressed: the dimensionality of the data set must be reduced to a managable level; a means of estimating the non-linearity (while retaining a low computational complexity in both analysis and run time implementation) of the final model is required.

5.3 Reducing Dimensionality

It is often important to decide what is the actual dimensionality of a data set, as the true dimensionality is often lower than the dimensionality of the space in which the data lies. This statement is more accurate when large dimensional spaces are considered. For example a data set may exist in two dimensions, but if it lies along a straight line then the true dimensionality is 1D. If, in general, the position $\underline{x} \in \Re^N$ of a point in N-dimensional space were representable by a relationship of the form $\underline{x} = \underline{x}(\underline{u})$, where $\underline{u}$ is a point in $\Re^M$, then the data is said to be M-dimensional. The transformation $\underline{x} : \Re^M \to \Re^N$ provides the mapping between the two spaces and allows any point $\underline{x} \in \Re^N$ to be dimensionally reduced to $\Re^M$ [Waite, 1992]

Using PCA, the value of *M* can be determined and the information loss estimated. This procedure also provides the transformation matrix that facilitates the projection $\Re^M \to \Re^N$.

The process of principal component analysis realigns the axis to fit the major deviation of the data set. These *extracted* axes can be used to describe the data in a new co-ordinate frame, which is the principle behind the PDM. As is typically the case, training data can be represented using fewer eigenvectors than the original dimensionality (see Chapter 2). This is itself a *lossy* dimensional reduction technique and relies on transforming the shape space into a lower dimensional space. In this reduced dimensional space the original data and its deformation from the mean can be expressed using the fewest number of parameters possible as determined from the eigenvectors of the covariance matrix.

By transforming the eigenvectors into percentiles it can be quickly seen how the dimensionality of the reduced space relates to the information loss of the reduction technique. By using the same analysis of this information as is used in the construction of the PDM (see section 3.2) a suitable mapping can be determined which provides minimal loss of information, typically less than 1%.

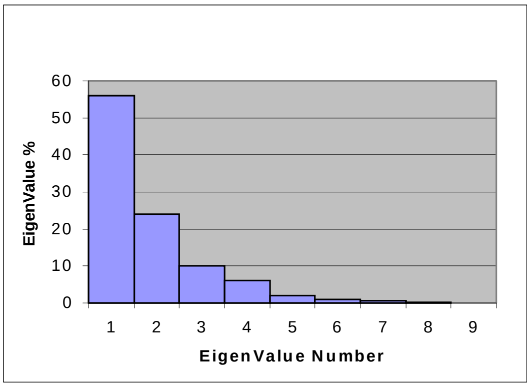


Figure 5.3.1- Table showing eigenvalues of co-variance matrix extracted via PCA

Figure 5.3.1 shows an example bar chart of eigenvalues extracted from a co-variance matrix, converted into percentiles and sorted into order. It can be seen that the 1st mode contains the majority of the deformation within the data set with the subsequent eigenvectors contributing in diminishing amounts. By summing the percentage contribution of each of the eigenvectors, a suitable dimensionality for the reduction can be determined (see section 3.2). For this example 99% of the deformation is encompassed within the first 6 eigenvalues with the last three contributing little to the information. These smaller 3 modes can therefore safely be discarded without adversely affecting the information content of the data set. It is also useful to note that these smaller modes are often largely attributable to noise within the data set and hence discarding this information can have benefits in smoothing the data.

Once the dimensionality, *M,* of the reduced space $\Re^M$ has been determined, the *M* primary eigenvectors can be used to project the original data set into this lower dimensionality. This is achieved by projecting the training examples onto each of the eigenvectors in turn, and recording the distance from the mean. The resulting transformed training set will therefore be represented in the lower dimensional space (using the co-ordinate frame of the eigenvectors), while the important information about the shape and size of the data remains preserved.

The dimensionally reduced vector is calculated as $\mathbf{x_r} \in \Re^M = (d_1, d_2, \ldots\ldots, d_M)$, where the j^{th} component,

$$d_j = v_j \bullet (x - \bar{x})$$ **Equation 5-1**

or alternatively in matrix form where $\mathbf{P} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t)^T$ is a matrix of the first *t* eigenvectors

$$\mathbf{x}_r = \mathbf{P}^T (\mathbf{x} - \bar{\mathbf{x}})$$ **Equation 5-2**

To reconstruct the original vector x, from the d_j component of the reduced vector x_r,

$$x = \bar{x} + \sum_{j=1}^{nr} d_j v_j$$ **Equation 5-3**

Note that equation 5-2 is the formulation for the linear PDM, where each component of the reduced vector is effectively the weighting parameter of the final shape.

This does not provide a true dimensional reduction, as *M* eigenvectors $v_{1 \rightarrow M}$ must be stored for use in the transformation between the reduced and original dimensional spaces. However, the primary concern, which is perfectly satisfied by this technique, is to reduce the dimensionality of the training set for non-linear analysis.

5.4 Estimating Non-linearity

It has already been shown how non-linearity can be estimated by breaking the shape space down through the use of multiple planes (Figure 5.2.2). A similar procedure can be performed by breaking the curvature of the space up into piecewise linear patches which estimate any curvature present. This is similar to the polygonal representation of a parameterised surface. As the number of polygons increase, so the visual accuracy of the resultant surface increases. However, as in most graphical (polygonal) representations there is a trade-off between the number of polygons (and hence render speed) and the accuracy of the representation. This optimum number of polygons is easily selected for graphical representation dependent upon simple visual criteria. For high dimensional data sets this number is more difficult to determine.

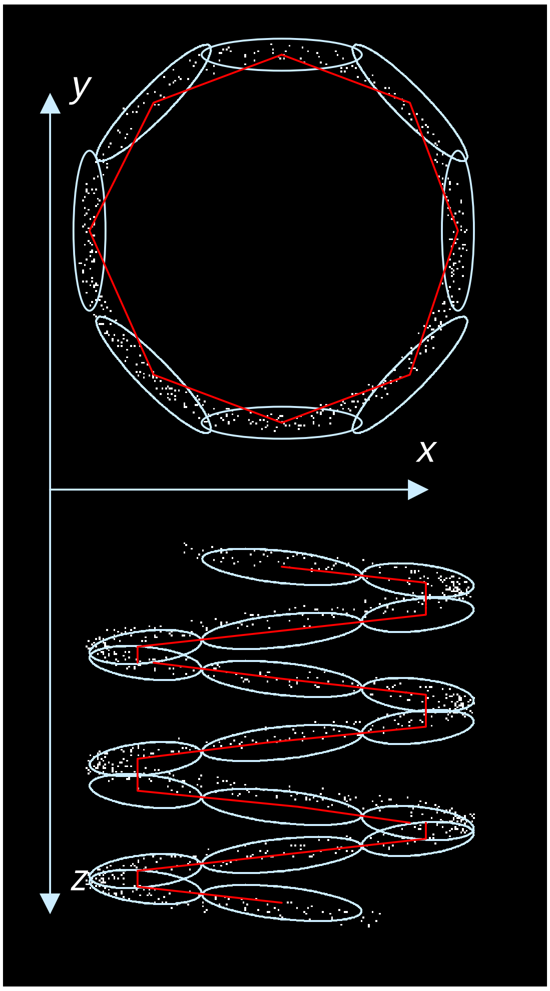


Figure 5.4.1 - Cluster Based Approximation

Figure 5.4.1 shows the helical data set broken down into smaller clusters which themselves can be treated as linear patches. The centres of each of these clusters when connected allow the estimation of the primary mode of the helix. Each cluster contains local information on how the data set varies, and must be analysed further in order to provide an accurate representation of the space. However, providing the space is segregated into a sufficient number of clusters, each can be treated as piecewise linear patches which encompass the major curvature of the space. The assumption that each cluster is approximately linear allows a local linear mathematical model to be used, such as principal component analysis. To provide a smooth transition between these linear patches it is important that there is a good overlap between them. This is important where a gradient descent approach is to be used in tracking, as a single iteration of the model may not be sufficient to allow the model to make the transition between two adjacent, non-connecting clusters.

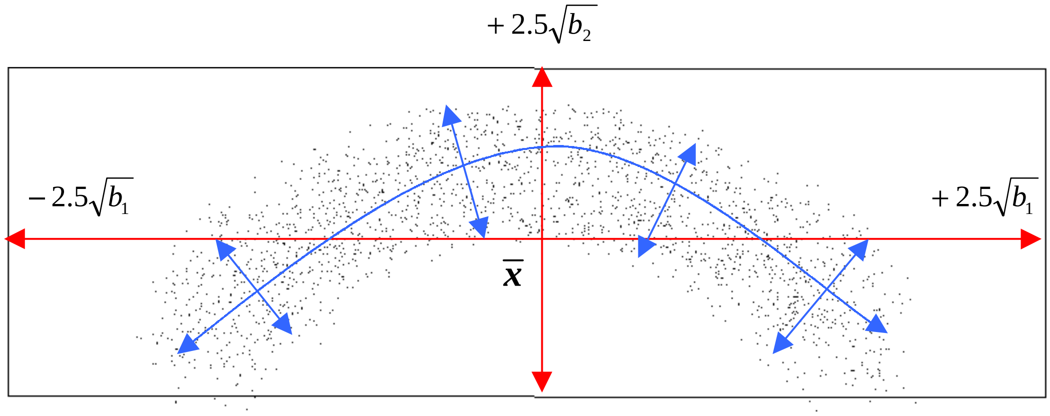
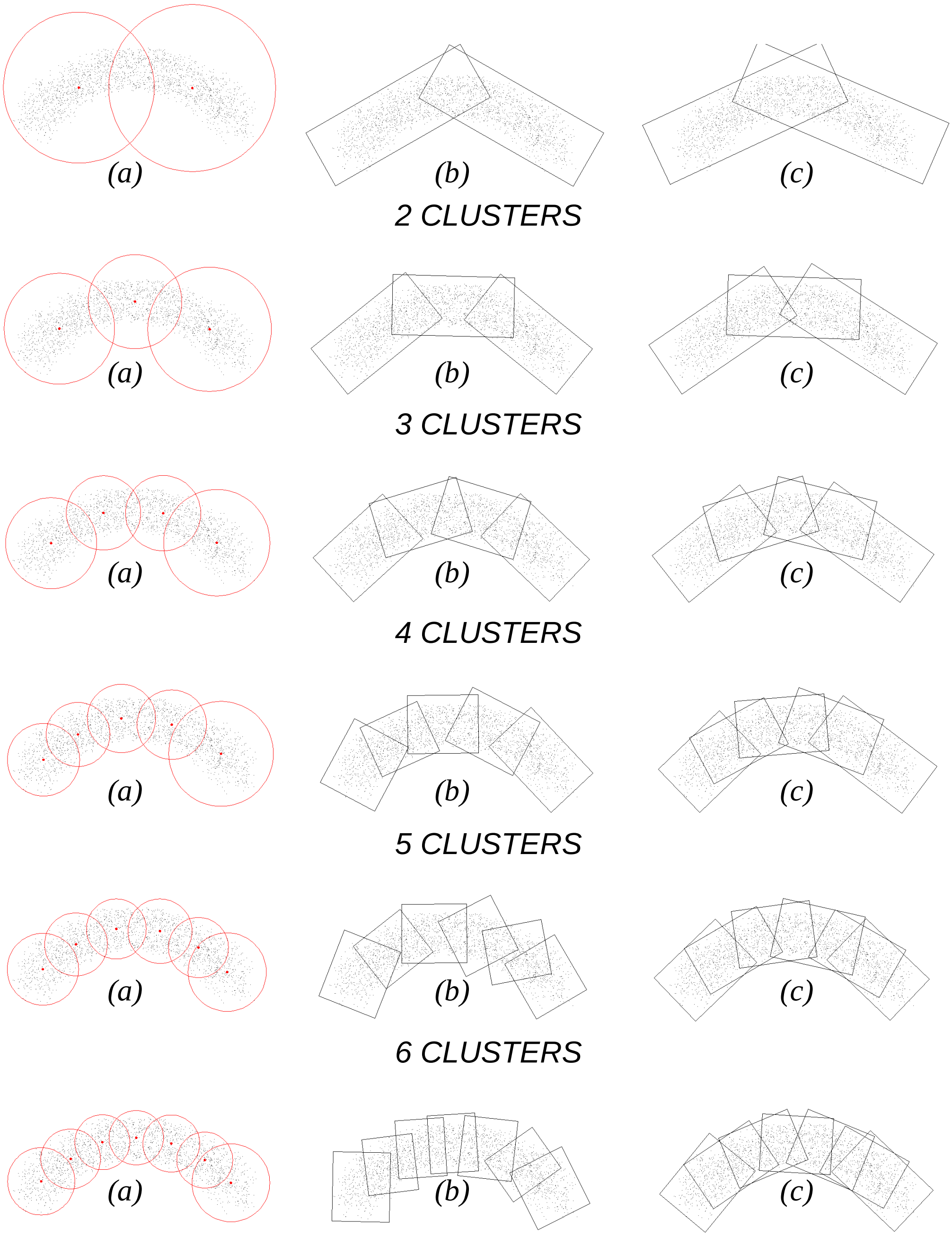


Figure 5.4.2 - Linear principal components of a curved data set

Figure 5.4.2 shows a synthetic data set with 2000 members in a two-dimensional curve. Performing standard linear PCA on this data set gives two primary modes, which are represented by the red arrows. Using suitable limits to bound these modes (2.5 times the square root of the corresponding eigenvalue from the mean shape) gives the bounding box shown in the diagram. It can be clearly seen that the mean shape is only just within the training set and the boundaries encompass far more of the space than is inhabited by the data points. The blue lines show the ***ideal*** primary and secondary non-linear axis of the data set.

Using this piecewise linear approximation to model the non-linear data set results in a more constrained model which better represents the original shape space. Figure 5.4.3 demonstrates the use of (a) cluster analysis to break down the original space into linear patches, and (b) the resulting bounds of these patches after linear PCA have been performed upon them for increasing number of clusters. (c) shows the results of the fuzzy k-means algorithm.



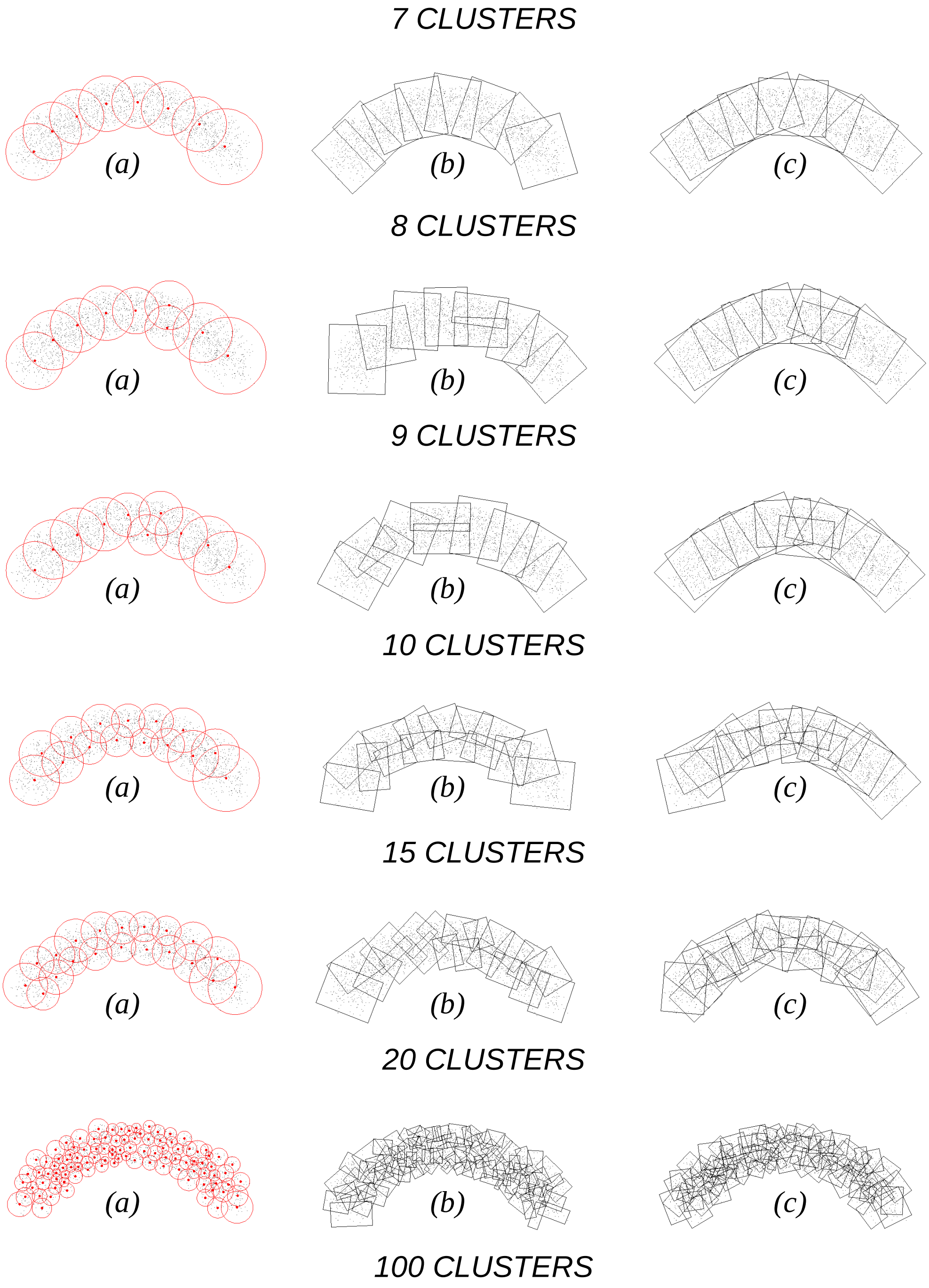


Figure 5.4.3 - Cluster analysis on shape space

(a) Cluster centres and bounds, (b) k-means (c) Fuzzy k-means

Figure 5.4.3(a) shows the results of running a k-means clustering (see Appendix 1) algorithm on the synthetic data set with curvature. The red points depict the centres of the final extracted clusters and the circles show the approximate bounds of these clusters. Using cluster analysis to segregate the space, PCA is then performed upon each cluster and the results are shown in Figure 5.4.3(b).

Each bounding box shows the extent of each linear patch, modelled as $\pm 2.5\sqrt{\lambda_i}$ (as described earlier). It should be noted that as the number of clusters is increased the resulting model better encompasses the curvature, although the rate of increase in accuracy diminishes as more patches are used.

It is clear from the 2-cluster example that it performs significantly better than the single linear PCA model and greatly reduces the redundant space, which is incorporated into the final model. When the number is increased to 3 or 4 clusters there remains a visible benefit in the accuracy of the model. However, as the number of clusters is increased further it becomes increasingly hard to determine if the benefits in model specificity can be justified against the increase in computational complexity. In the analysis of *true* data, where it becomes impossible to visualise the high dimensionality of the space, such visual assessment is not possible. An alternative method of assessment for choosing the number of clusters can be provided through normal cluster analysis as described in Appendix 1. From Figure 5.4.4 the natural number of clusters can be estimated to be 5 which ties in with the visual observations discussed earlier.

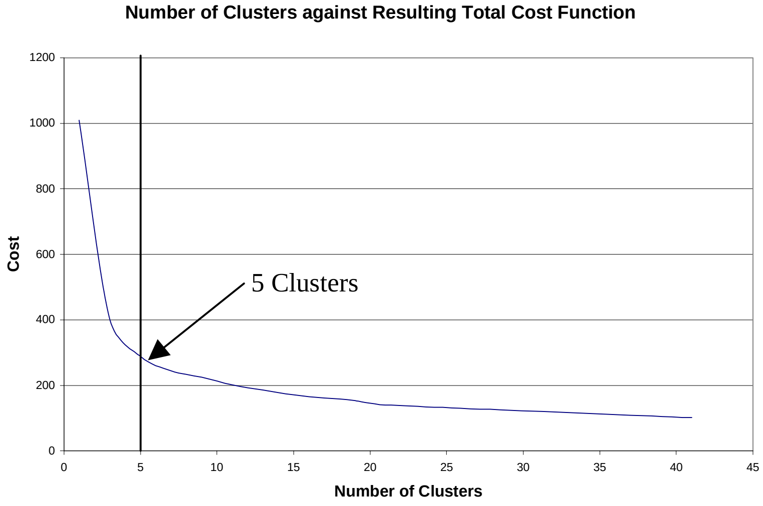


Figure 5.4.4 - Cost graph for synthetic curved data set

Figure 5.4.3(c) shows the results of using a fuzzy k-means clustering algorithm (see Appendix 1) on the same data set. It can be clearly seen that using the fuzzy algorithm significantly increases the overlap between adjacent clusters and provides a smoother composite model for estimating non-linearity. This is important during tracking, especially when using a gradient descent approach (iterative refinement approach). This ensures there exists a smooth path between the composite elements of the model.

Bregler and Omohundro [Bregler 94] made no provision for this problem when separating the shape space into sub-clusters and hence this adds to the observed model error which will be shown during comparison in section 5.7.

This technique also allows discontinuous surfaces to be modelled accurately, which is an important consideration when attempting to model non-linearities for computer vision applications. If a test example were to be considered in which a break exists in the training set (see Figure 5.4.5), then existing techniques would attempt to model this discontinuity by a single model. The resulting linear PDM would be similar in nature to that shown in Figure 5.4.2(a).

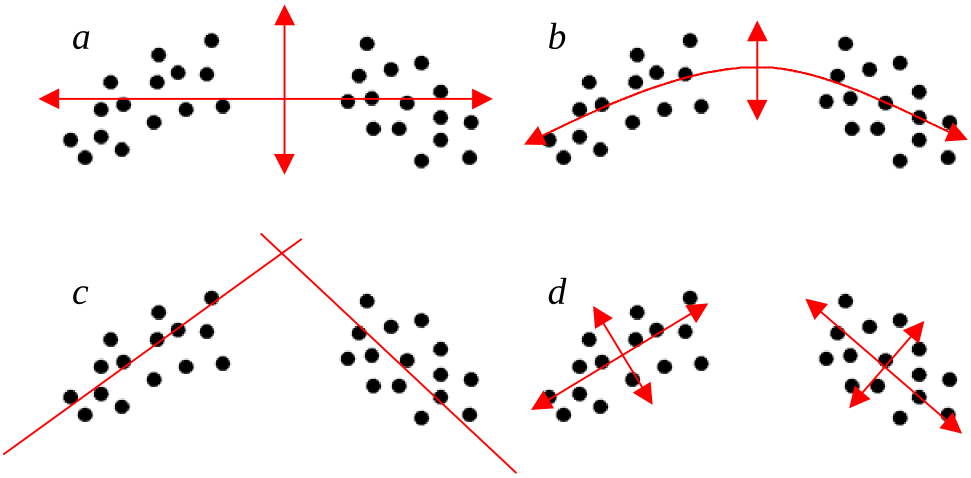


Figure 5.4.5 - Modelling Discontinues Data Sets - Types of Model

(a) Linear PDM, (b) Polynomial Regression PDM,
(c) Constraint Surface (d) Composite NLPDM

Figure 5.4.5 shows an example discontinuous data set with various forms of PDM model fitted: *(a)* shows the linear PDM which models the entire space as a single rectangle, the mean within the central ***null*** space; *(b)* shows the non-linear axis of a polynomial model smoothly parameterising the curvature, still with a mean shape within the ***null*** space; *(c)* shows the constraint surface approach of

Bregler which models the space as two finite thickness infinite hyperplanes; and (d) shows the composite NLPDM technique proposed here.

If new points are considered and the closest valid shape found within the model, the performance of each approach can be assessed.

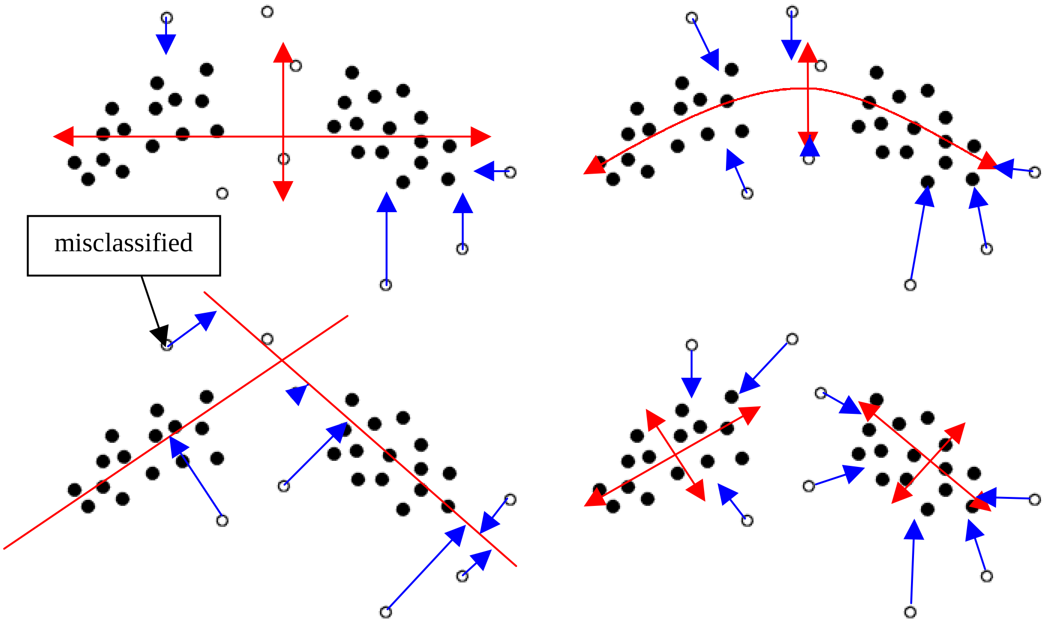


Figure 5.4.6 - Modelling Discontinues Data Sets - Nearest Valid Shape

(a) Linear PDM, (b) Polynomial Regression PDM,
(c) Constraint Surface (d) Composite NLPDM

It can be seen from Figure 5.4.6 that the linear PDM performs poorly for both the modelling of curvature and the discontinuity of the data set: many points remain unconstrained within the central *null* area. The polynomial model works well at modelling curvature; however, it performs poorly at modelling discontinuity. Although points on the extremities are drawn closer to the original training set shape, points within the *null* area remain unchanged. The constraint surface models curvature to an extent, but draws all model points to lie along the hyperplanes and does not work well for the discontinuity. In addition, the unlimited extent of the hyperplanes introduces further errors at boundaries, allowing points to be misclassified to the wrong hyperplane. The composite NLPDM seems to be able to model both types of non-linearity correctly, and only introduces boundary errors due to the rectangular assumption of linear patches.

An example of complex discontinuous surfaces can be found in Section 7.3.

5.5 Composite NLPDM

This section presents two test cases to demonstrate the validity of the approach at modelling non-linear data sets. The examples were chosen to represent both high non-linearity and high dimensionality. The construction of the composite non-linear PDM is outlined below.

An algorithmic overview is given below.

```
1. Perform PCA on training set
2. For each training example do
        Project training example onto eigenvectors, recording distance from mean.
        Concatenate these distances into a reduced dimensional vector.
3. Perform cluster analysis on dimensionally reduced data set to determine natural number of clusters
4. Use this natural number to segregate the data set into multiple clusters using fuzzy k-means
5. Perform PCA on each cluster of training set
```

5.5.1 Robot Arm

The first example that will be considered is of a relatively low dimensionality, but with high non-linearity present. The robot arm example meets these criteria as the nature of its hierarchical, pivotal construction guarantees a non-linear data set. The training data for the robot arm example was constructed automatically from a synthetic model used to generate examples that encompassed the total possible movement of the arm. Figure 5.5.1 shows the construction of the arm model. The 2D representation of a robot arm consists of four rectangles, each rectangle described by four key points at its corners. This gives a total of 16 2D key points which, when concatenated together, provide a 32 dimensional vector that describes the shape of the arm at any time. The model also incorporates 3 pivotal joints, which allow the constituent sections of the arm to rotate about each other. Examples were generated for the arm in all its various positions by taking examples of the model as the joints were rotated from ±45° in 10° intervals. This resulted in a 32 dimensional training set containing 918 examples.

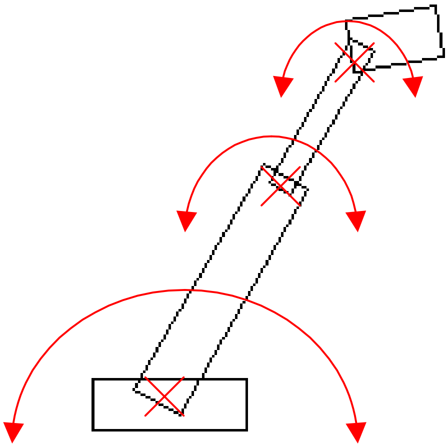


Figure 5.5.1 - The construction of a non linear robot arm data set

Figure 5.5.2 shows examples taken from the synthetic training set.

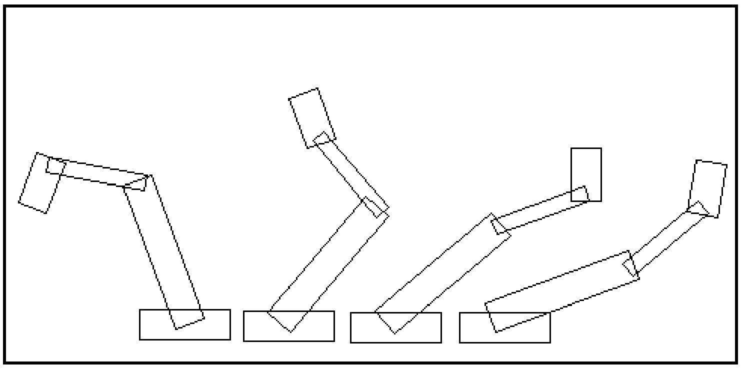


Figure 5.5.2 - A selection of training examples from the robot arm data set

As the dimensionality of the model is already low (i.e. 32D) it is not necessary to perform dimensional reduction on the model and therefore k-means analysis can be carried out on the raw data set. Performing standard cluster analysis (see Appendix 1) the graph in Figure 5.5.3 is produced and indicates the natural number of clusters to be approximately 20. Using this number of fixed clusters the fuzzy k-means algorithm is applied in order to segregate the data set into its constituent linear patches.

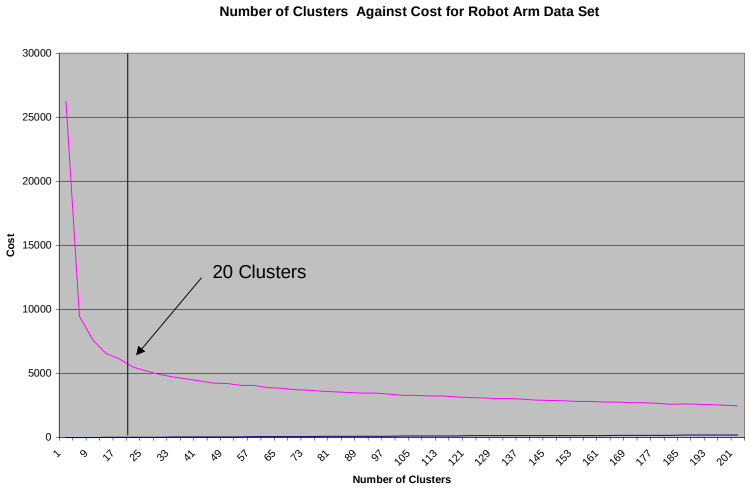


Figure 5.5.3 - Cluster analysis on raw robot arm data set

Figure 5.5.4 shows the resulting boundaries on the data set after PCA has been performed on the extracted clusters projected into 2-dimensions. Note that rectangles are skewed due to the projection of each model ($m_{0\text{-}31}$) down from 32 to 2 dimensions. This figure clearly shows the non-linearity of the model and how the linear patches estimate this curvature.

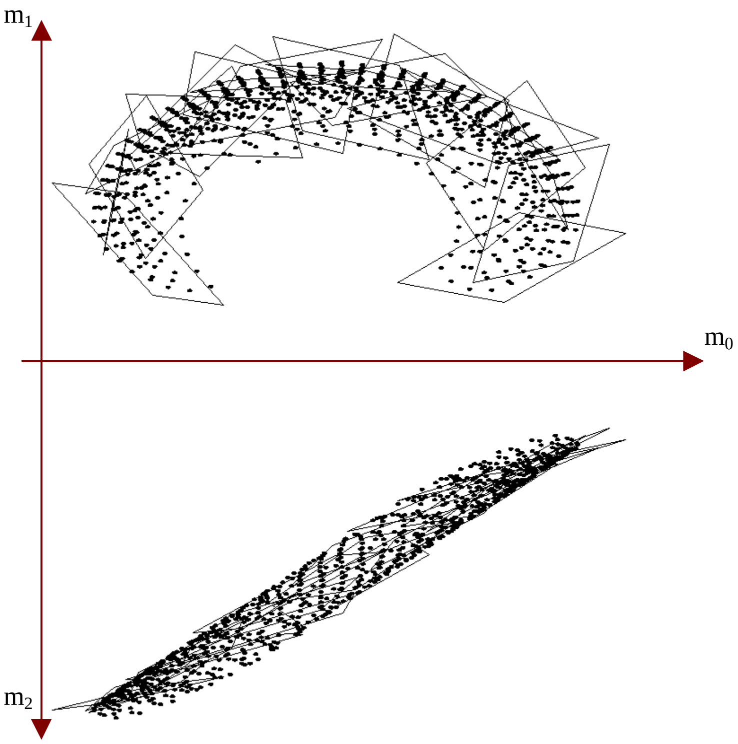


Figure 5.5.4 - Linear patches of the robot arm data set

In order to validate the hypothesis that reducing the dimensionality of the data set before analysis does not affect the information content of the resulting model, the procedure was repeated upon the data set after dimensional reduction.

PCA was first performed upon the raw data set and from the eigenvalues a suitable reduction was determined. 99% of the deformation is contained within the first 4 eigenvectors, corresponding to the four largest eigenvalues. The data set was then projected down into this 4 dimensional space using *equation 5-1* (page 65). Cluster analysis was then performed to extract the natural number of clusters and the fuzzy k-means algorithm performed to extract the membership of each cluster. The results of cluster analysis can be seen in Figure 5.5.5. Apart from the difference in the scale of cost, the graph is almost identical to that previously produced and, as in Figure 5.5.4, provides a natural number of clusters equal to approximately 20.

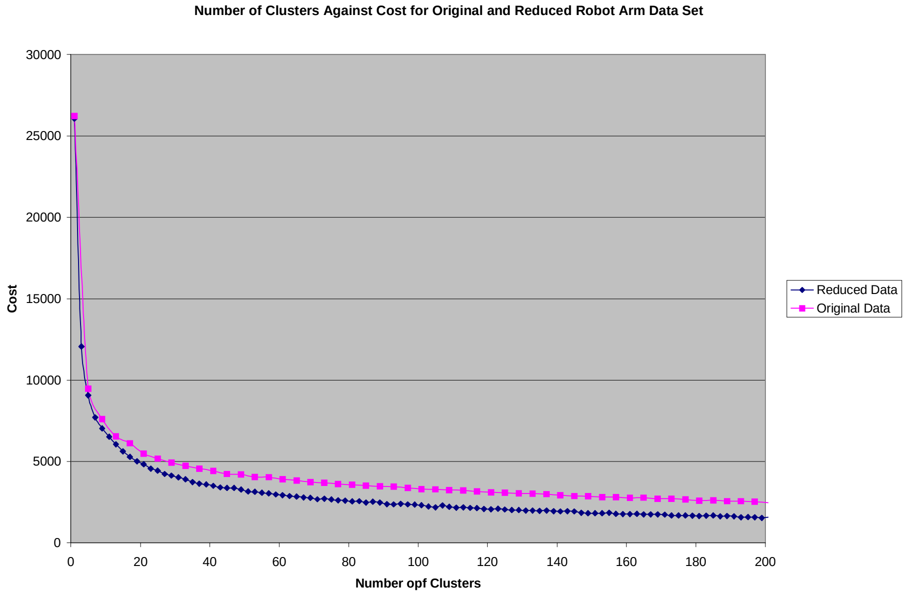


Figure 5.5.5 - Cluster analysis on the reduced robot arm data set

Once the cluster membership has been extracted, each element of the clusters is transformed back into the original space using the *equation 5-2* (page 65) before PCA is performed. This procedure leads to the loss of up to 1% information due to the lossy compression technique used. As an alternative, the reduced vectors can be used merely as pointers to the original data set, since the 1st element of the reduced data corresponds to the 1st element of the original data. Once this reverse mapping has been completed, PCA is performed on each of the fuzzy clusters to produce the composite model as done previously.

The lower cost solutions for the reduced dimensional data results from the disregarded data no longer contributing to the overall cost of the k-means function. However, although this makes little difference to the selection of the *natural number*, it provides a huge computational saving as the analysis is performed in a 4 dimensional space rather than one of 32. In fact, if the assumption is made that the primary modes contain the largest contribution to the separation of shape space (which is known), then this cluster analysis could feasibly be performed with even higher dimensional reductions. However, it is not obvious how this number would be selected.

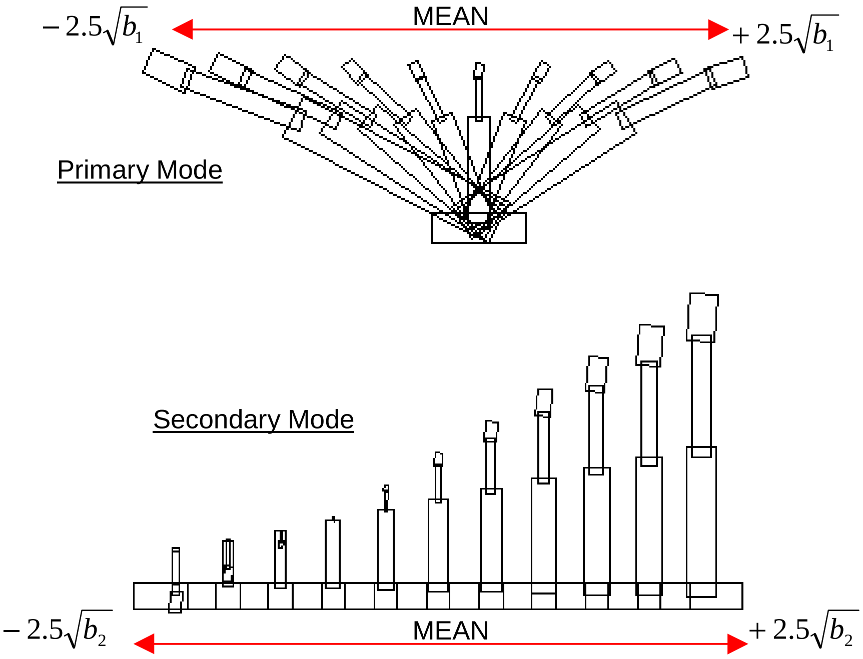


Figure 5.5.6 - Primary modes of the linear robot arm PDM

Figure 5.5.6 shows the primary and secondary modes of variation of the linear PDM. The non-linearity of the model is clear in the distortion of the dimensions of the robot arm. The primary mode encompasses movement along the horizontal, but also has distortion in the size of the arm, which must be rectified by other higher modes of variation. The second mode encompasses movement in the vertical, with more extreme size distortions, especially at the head of the model. Below the mean on the second axis, the model takes on shapes which were not present within the training set by inverting the arm back upon its self.

Figure 5.5.7 shows examples from the final composite non-linear model. It demonstrates that much of the non-linearity has been removed except in the end of the model where small abnormal deformations can still be seen. By increasing the number of clusters this can be reduced further, but at a computational cost at run-time.

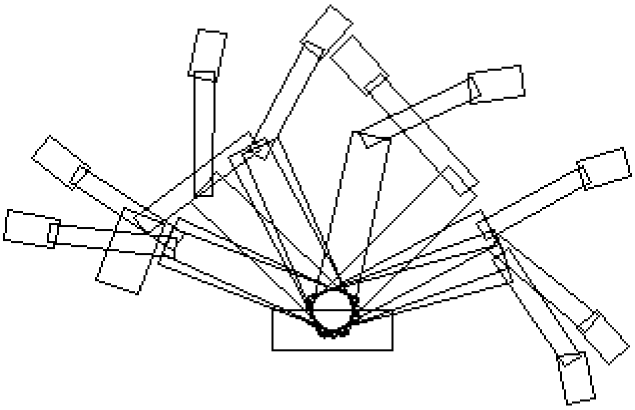


Figure 5.5.7 - Examples from the non-linear robot arm PDM

5.5.2 Image Space

An image training set was constructed from a sequence of 200 images of a head turning in the image frame. No alignment was performed so as to produce as non-linear a problem as possible. Each frame is 80 by 60 pixels in size, producing a 4800 dimensional training vector. PCA is first performed and the 33 eigenvectors corresponding to the 33 largest eigenvalues extracted. These vectors account for 99.9% of the deformation in the training set. Figure 5.5.8 shows the first and second modes of variation after linear PCA

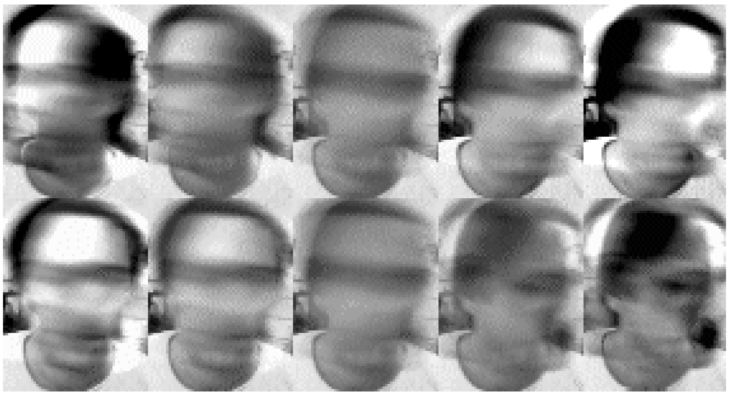


Figure 5.5.8 - Primary modes of the image PDM

Each vector is then projected into this PCA space (using *equation 5-1 page 65)* giving a new dimensionally reduced training set on which cluster analysis can be performed. This generates a dimensional reduction of 4800 to 33.

Cluster analysis results in an estimate for the natural number of clusters, *k*=15. PCA is performed on each of the 15 clusters in turn to generate the composite non-linear model. Selected shapes reconstructed from the composite model are shown in Figure 5.5.9. Notice that each model has reduced blurring, due to the original data set being subdivided into smaller clusters. Each cluster now has less information to encode and hence linear PCA can better estimate the deformation.

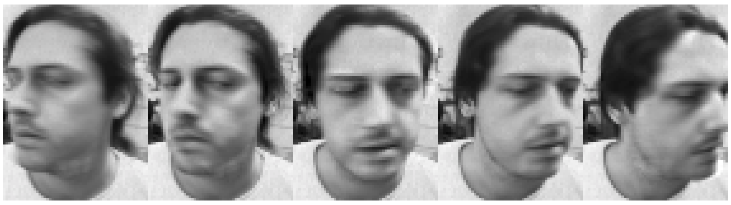


Figure 5.5.9 - Examples from the composite non-linear image PDM

As mentioned earlier the technique also has the advantage that the hyper surface, or volume, on which the data lies need not be contiguous. For example, given an image sequence of two people, one with glasses and one with a beard, both linear PCA and the high order non-linear approaches will model the data set with a principal mode which interpolates between the two. However, there is no example in the training set where both glasses and a beard are present. The cluster-based technique will separate these two distinct clusters, allowing the model to ‘jump’ between the two, better representing the training set. This issue and its implications will be discussed in the following chapters.

5.6 Application of the Model

To apply the model to an image, a similar procedure to the linear PDM (see section 2.3) can be used. After making an iterative refinement to the model within the image frame, the closest possible shape within the *learnt* bounds of the model is calculated. This constrained shape is then used as the model pose for the next iteration.

In the case of the linear PDM, this *constrained shape* is found by projecting the model into the PCA space and reconstructing the closest allowable model (point in shape space) that is within the bounds of the linear model. The same procedure can be used in the composite model. However, the closest allowable point may exist in any of the clusters which constitute the non-linear model. The centre of each cluster can be used to check for closest cluster in Euclidean distance from the model point. However, using a Euclidean distance metric makes the assumption that all clusters are of the same size. Figure 5.6.1 illustrates this problem. Assuming a point *p* in shape space, it should be apparent that the point belongs to the cluster *C1*. Using a Euclidean distance metric will result in the point being assigned to the cluster *C2* due to the size difference in the clusters. However, the point *p* is actually closer to the cluster *C1* even though in Euclidean space the point is further from the centre C1 due to the standard deviation of the clusters.

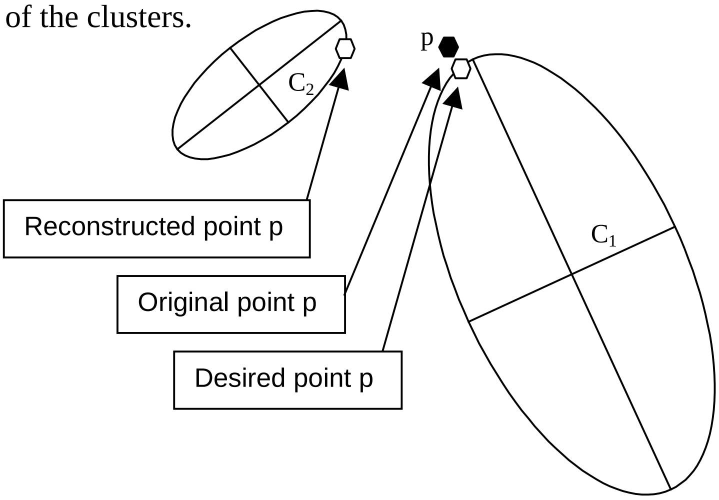


Figure 5.6.1 - Distance Metrics in Shape Space

To overcome this problem a Mahalnobis distance metric can be used. However, due to the simplicity of the k-means-clustering algorithm, it is a fair assumption that if the selected natural number is correct then clusters will be similar in size. It is important to bear this consideration in mind, especially when discontinuous surfaces are considered. In these situations, many clusters may be of different sizes and therefore the Mahalnobis approach should be used.

An algorithmic overview for model application is:

For a new shape **S**,

1. Transform **S** from image frame to PDM model basis eg. Normalise and align (as in alignment of training set)
2. Locate closest cluster centre and hence linear patch $\mathbf{P_i}$ using either Euclidean or Mahalnobis distance metric
3. Project **S** down onto linear patch $\mathbf{P_i}$
4. Project back up to reconstruct closest allowable shape **S'**
5. Transform **S'** back into image frame co-ordinates

5.7 Evaluation and Performance

To asses the performance of the approach to the modelling of non-linear data sets an error metric must be defined which provides a measure of the accuracy of an approach. As has already been demonstrated, a common problem with the linear representation of non-linear data is the tendency to over-generalise shape and to incorporate non-valid deformations into the model. These non-valid deformations often manifest themselves as the distortion in scaling of the model as observed in the robot arm example (section 5.5.1). In this example, the robot arm should remain constant in size and area as it rotates around its pivotal joints. Since this size is the major artefact of the linear representation, it provides a suitable error metric with which to assess non-linear performance.

Random points chosen from within the linear PCA space are selected and then projected into the composite model. The constraints of the model are applied and the resulting (supposedly valid shape) assessed by calculating the length of the model perimeter (projected onto the image plane). Since the ideal length of a valid shape should remain constant (in this case 66 pixels), any deviation from this constant can be used as a measure of the model's inability to reproduce valid shapes.

A number of random shapes were generated and passed through the model, the absolute difference from the ideal length recorded and the mean calculated over the test set. This procedure was then repeated for the constraint surface, a nearest neighbour approach and the cluster based NLPDM proposed here for varying numbers of clusters between *1* to *n* (where n equals the number of training examples). The procedure is outlined thus,

1. Take *n* random shapes X_i^{rand}
2. Project each X_i^{rand} into non-linear model and find closest reconstructed point X_i^{recon}
3. Calculate length in image plane of projected model X_i^{recon}, L_i^{recon}
4. Calculate length in image plane of any valid model X, L^{valid}

Calculate deformation error metric $e = \frac{1}{n}\sum_{i=1}^{n}\left|L_i^{recon} - L^{valid}\right|$ **Equation 5-4**

This error metric provides a zero error if the resulting reconstructed model is valid in shape. Therefore, the higher the error, the worse the performance of the constraints and hence the worse the performance of the model. By repeating this procedure for varying number of clusters between 1 (which is effectively a linear PDM) and 912 clusters (the number of training examples and therefore nearest neighbour), we can assess the advantage on model specificity as the number of clusters increases. Figure 5.7.1 shows the resulting graph from this analysis.

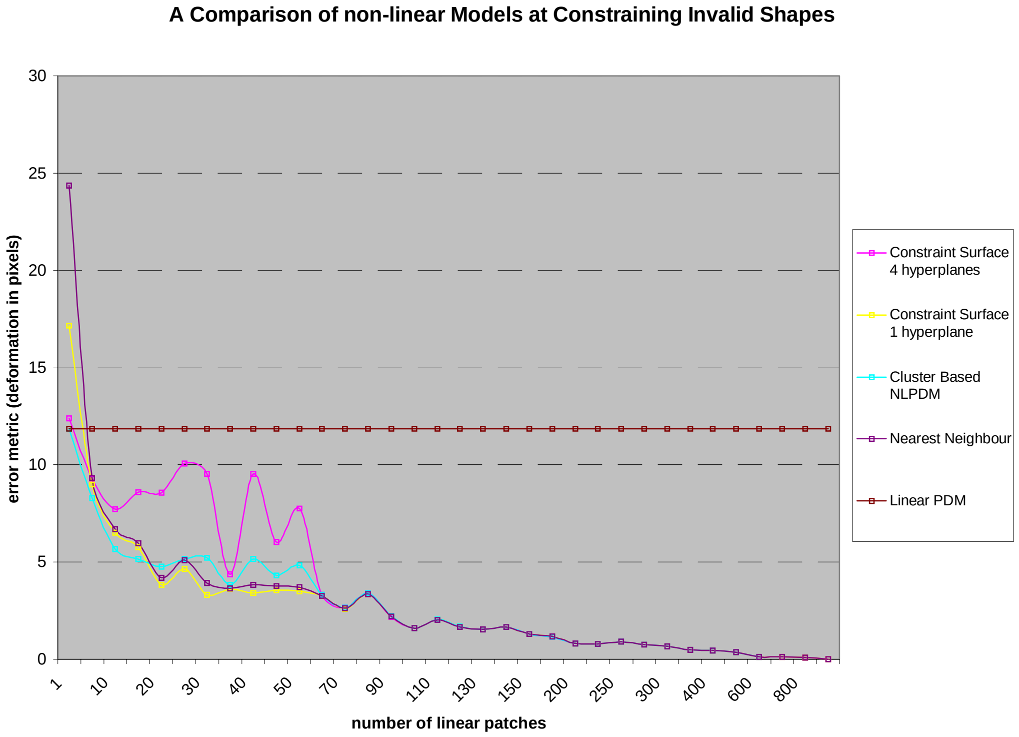


Figure 5.7.1 - Graph showing error rates of non-linear approximation techniques

The single hyper plane constraint surface, the nearest neighbour approach and the cluster based NLPDM all perform comparably and provide far lower error rates than either the multi-plane constraint surface or the linear PDM. However, the cluster based NLPDM (CB-NLPDM) provides lower errors until 5 patches are reached. With only a single linear patch the CB-NLPDM is effectively a linear PDM and as such does not produce errors that exceed the linear PDM. However, the other approaches produce significantly higher errors than even the linear model until sufficient patches have been introduced. As the number of clusters increases, so the error rate decreases, showing that the procedure does indeed increase the model's ability at representing non-linearity. The yellow trace on the graph shows the error results of the unconstrained surface approach of Bregler [Bregler 94] which, although performing slightly better between 25 and 70 patches, produces higher error rates at the pre-chosen patch number of 20 which was determined earlier from cluster analysis. It is important to note that this error graph confirms the results of the cluster analysis for the natural patch number, as further increases beyond 20 result in less significant results in the final model.

This confirms the conclusion that the approach for the selection of the natural number of clusters is valid, and hence the number of patches needed by the model is correct.

As the number of clusters increases to 912 (which is the number of examples within the training set) the error reaches zero. This is to be expected: when the number of clusters is equal to the number of training examples, each cluster contains only one member. The procedure then becomes a nearest neighbour approach. Since each nearest neighbour is in fact a valid training example, the validity of the shape is ensured, hence the zero error. This fact also explains the error results of the nearest neighbour approach which performs comparably to the other techniques. The question could be posed, why not use a nearest neighbour approach to perform the procedure simply and accurately? However, there are two issues, which have not as yet been considered.

1. The speed of the procedure increases as the number of linear patches (clusters) increases, as each patch is itself a linear PDM.
2. A nearest neighbour approach is only valid if every possible model pose is represented within the training set. This is often not the case and the power of the linear PDM is the ability to model shapes not present within the training set by linearly interpolating between examples.

It is therefore apparent that in order to consider the validity of any technique, two questions must be posed.

Does the model stop non-valid shapes from being produced?
(which has already been addressed in Figure 5.7.1)

Does the model allow valid shapes which were not present within the training set to be reproduced?

In order to answer this latter question a new set of experiments must be devised.

By constructing a new set of *n* examples that are all valid in shape and deformation not present within the training set (possible due to the synthetic nature of the test case), the ability of the CBNLPDM at reproducing unseen, valid shapes can be assessed. Using the same equation 5.4 (page 82) along with a Euclidean distance measure between the 'original valid but unseen data' and the 'reconstructed shape' this feature of the model can be assessed.

1. Take *n* valid shapes not present in the training set X_i^{new}
2. Project each X_i^{new} into non-linear model and find closest reconstructed point X_i^{recon}
3. Calculate the length in image plane of projected model X_i^{recon}, L_i^{recon}
4. Calculate the length in image plane of any valid model X, L^{valid}

Calculate deformation error metric $= \frac{1}{n}\sum_{i=1}^{n}\left|L_i^{recon} - L^{valid}\right|$

Euclidean distance error $= \frac{1}{n}\sum_{i=1}^{n} D\left(X_i^{recon} - X_i^{new}\right)$

Using these error metrics it would be expected that if the model were performing perfectly any valid shape projected into the model would have zero deformation error and zero Euclidean error. However, using the nearest neighbour approach would result in a zero deformation error but produce a high distance error. The result of performing this analysis on the data for both approaches is shown in Figure 5.7.2, Figure 5.7.3 and Figure 5.7.4. The test set consisted of examples generated from $\pm 38°$ angles and $17°$ intervals producing 135 valid, but unseen, examples with which to test the various models.

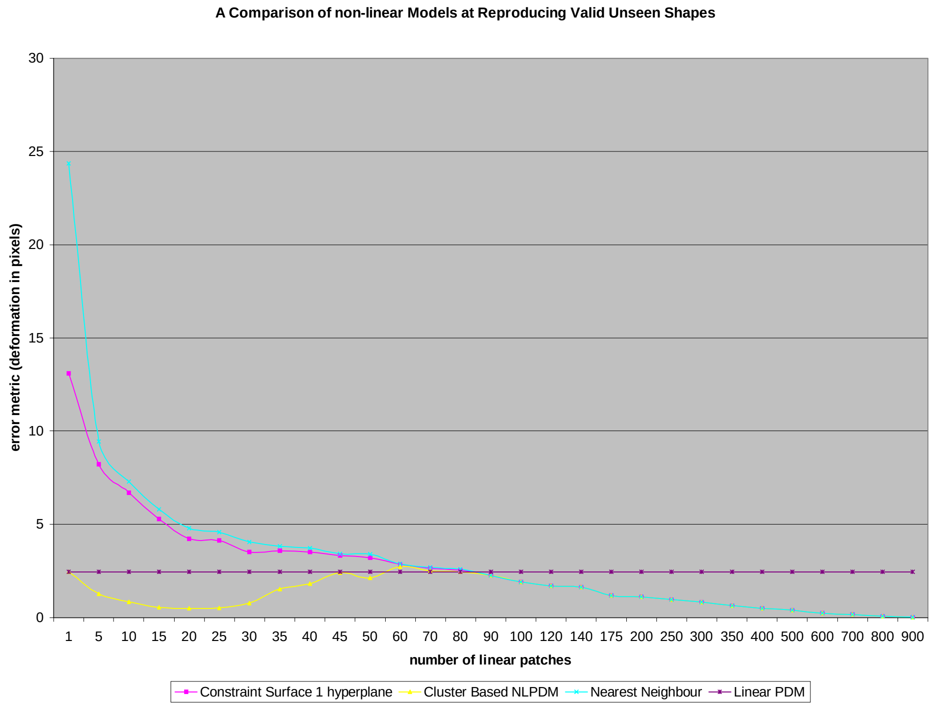


Figure 5.7.2 - Graph showing error rates of non-linear approximation techniques for Constraining Valid Unseen Data

Figure 5.7.2 shows the results generated via the deformation error metric for valid, but unseen, shapes applied to the various models. From this graph it can clearly be seen that the linear PDM produces a low baseline error of around 2.5 pixels deformation. This demonstrates the ability of the linear PDM to encapsulate the deformation of the training set, allowing valid shapes to be reproduced which were not present within the original data. It is not until in excess of 85 linear patches are used that either the nearest neighbour or constraint surface performs comparably to the linear PDM. The nearest neighbour approach generates the highest error rates as was suspected. The constraint surface with 4-hyperplanes produces the same results as the proposed NLPDM technique, both of which produce by far the lowest errors. Using 20 linear patches, both techniques produce their lowest error rates of approximately 0.5 pixels deformation, which again confirms the selection of the natural number of clusters for the data set.

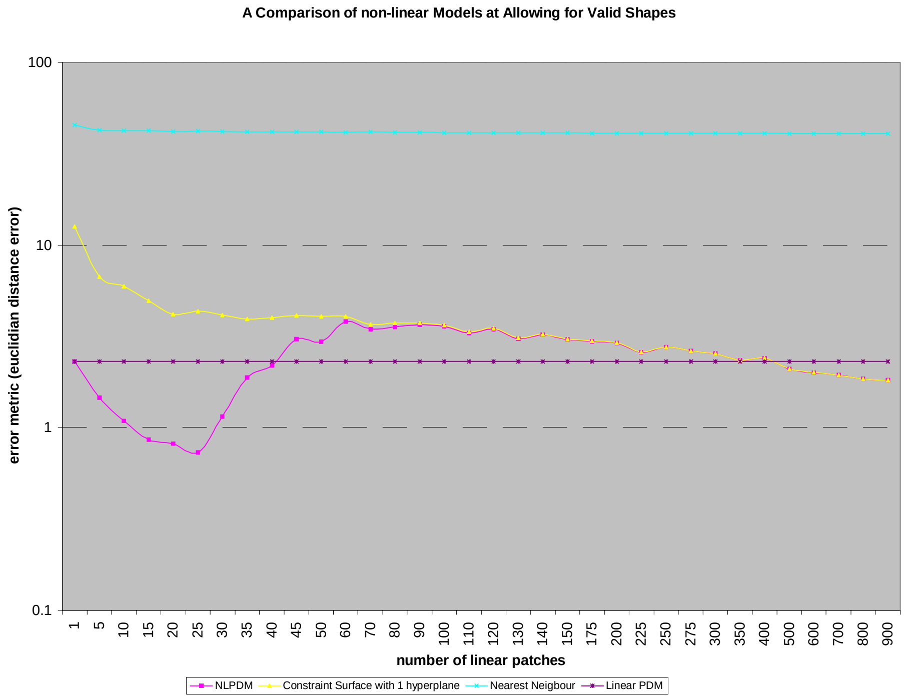


Figure 5.7.3 - Graph showing error rates of non-linear approximation techniques for Allowing Valid Unseen Data

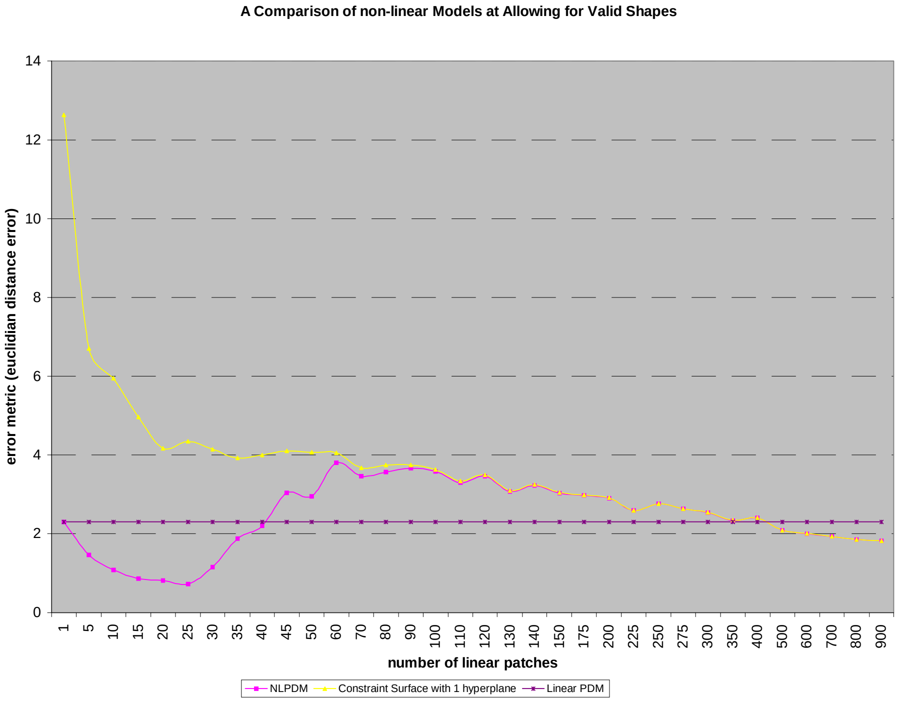


Figure 5.7.4 - Graph showing error rates of non-linear approximation techniques for Allowing Valid Unseen Data

Figure 5.7.3 shows the results generated via the Euclidean distance error metric for valid, but unseen, shapes applied to the various models. The figure uses a logarithmic scale due to the extremely high error rates produced by the nearest neighbour approach. Figure 5.7.4 shows the same data (without the nearest neighbour approach present) on a linear scale. It can clearly be seen that the nearest neighbour approach produces error rates far in excess of any other approach. The linear PDM produces a low baseline error, which could be reduced further by increasing the number of modes of variation. The constraint surface with 1-hyperplane produces much higher error rates than the linear PDM and does not perform comparably with the linear PDM until around 450 linear patches, where each patch effectively has only two members. If a patch has only two members then it can have only one hyperplane, which means that more planes are required to model the data. This is confirmed by the 4-hyperplane approach which produces error rates identical to the NLPDM model, both of which produce errors of around 0.7-0.8 at the chosen number of clusters. If all these graphs are considered, the lowest errors are produced at 20-30 linear patches which suggests that the natural number may be slightly higher than was chosen. However, changing this number would result in little gain in accuracy.

Model Approach	**Ability to Constrain Unseen Data**	**Ability to Constrain Valid Data**	**Ability to Allow Valid Data**
Linear PDM	*BAD*	*POOR/GOOD*	*GOOD*
Nearest Neighbour	*GOOD*	*BAD*	*BAD*
Constraint Surface 1 hyperplane	*GOOD*	*BAD*	*POOR*
Constraint Surface 4 hyperPlanes	*POOR*	*GOOD*	*GOOD*
Cluster Based NLPDM	*GOOD*	*GOOD*	*GOOD*

Figure 5.7.5 - Table Showing Comparison of Techniques

If the performance of each technique is considered for each of the comparative studies performed, the conclusions can be summarised in a table, as shown in Figure 5.7.5. From this table it can be demonstrated that the proposed NLPDM approach produces superior performance in all aspects of modelling.

5.8 Conclusions

In conclusion, a NLPCA technique has been presented which models non-linearity by breaking the problem down into a set of linear models, which estimate high dimensional curvature. This has the advantages of the speed and simplicity of linear PCA, whilst providing a robust solution to object modelling. It has been shown how this technique performs in comparison to similar techniques and how the simple selection of model parameters can produce optimum solutions in the final model. These models have been shown to work on both low dimensional, high non-linear, and high dimensional, high non-linear problems where other procedures would fail.