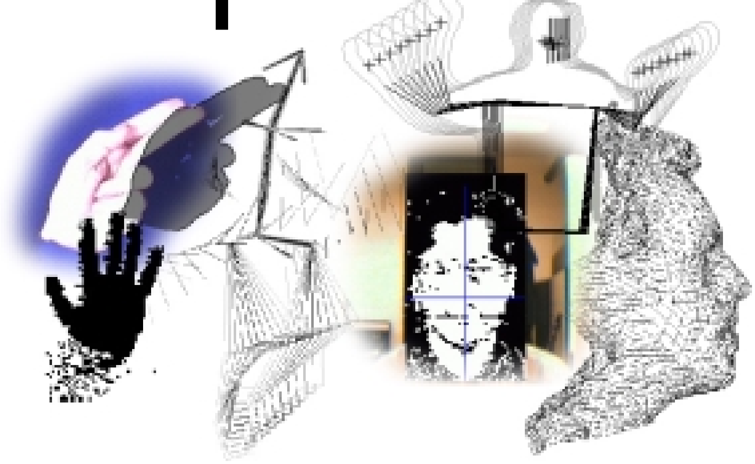
Chapter 6



6 Cluster Constraints on Shape Space

6.1 Introduction

Thus far techniques have been discussed to project a non-linear data set into a lower dimensional space where further analysis is feasible. Once the shape space and its non-linearity have been estimated through cluster analysis, this segregation is modelled through multiple linear PDMs. The position and bounds of each linear patch is obtained by performing PCA on each extracted cluster and its members. The dimensional reduction allows the non-linear analysis (clustering) to be performed on high dimensional problems, but provides no added benefit to the final model. Each sub PCA cluster has the original dimensionality of the training set.

The inherent dimensional reduction of the linear PDM often provides a useful representation during classification. However, by breaking the original space up into linear patches this benefit of the model is lost. To provide static classification as demonstrated in [Bowden 96] a linear PDM formulation would still need to be maintained in addition to the composite model. This would not be the case if each patch of the composite model segregated the space in such a way as to naturally aid classification. By retaining the dimensional reduction of the linear model throughout, and applying the constraints to the reduced data set, several advantages are achieved:

1. The dimensional reduction is retained throughout the model, providing a simplified model for classification.
2. For complex models where the number of clusters is high, the computational complexity of applying constraints is decreased.
3. Any noise within the model is filtered out by the linear PDM before constraints are applied.

The remainder of this chapter is concerned with the application of constraints to the dimensionally reduced data. Section 6.2 will discuss the application of these constraints. Section 6.3 will evaluate the approach and make comparisons with the previous chapter. Section 6.4 will demonstrate how this new model can be used in classification using sign language as an exemplar application. Section 6.5 will evaluate the performance of the proposed appraoch and lastly, conclusions will be drawn.

6.2 Constraining Shape Space

The basic procedure proposed in the previous chapter is outlined in Figure 6.2.1 where $\Re^N$ is the original dimensionality of the training set and $\Re^M$ is the reduced dimensionality of the training set after it has been projected down into the PCA space ($\underline{x} : \Re^M \rightarrow \Re^N$).

Previous work by the author and other researchers (see section 2.4) has shown how the reduced dimensionality of PCA space is invaluable in the classification of static poses of the model. Indeed, this is often used as an important tool in classification. It is therefore beneficial to combine these techniques in the modelling of a non-linear data set.

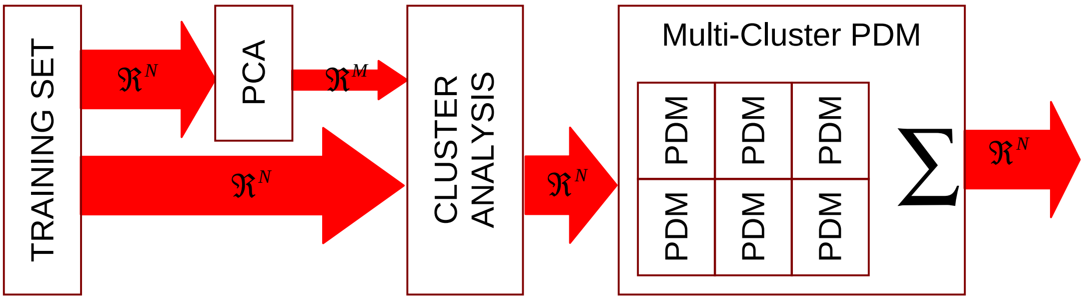


Figure 6.2.1 - Cluster Based non-linear PDM

Figure 6.2.2 gives an overview of this new approach, which will be referred to as '*Constraining Shape Space*' or **CSSPDM**. In this procedure the dimensional reduction of the PCA is retained throughout the entire model. In addition to the cluster analysis, PCA is performed on each cluster in the dimensionally reduced space, constraining the model in PCA space. The model must then be projected back up into the original dimensionality to extract the final shape.

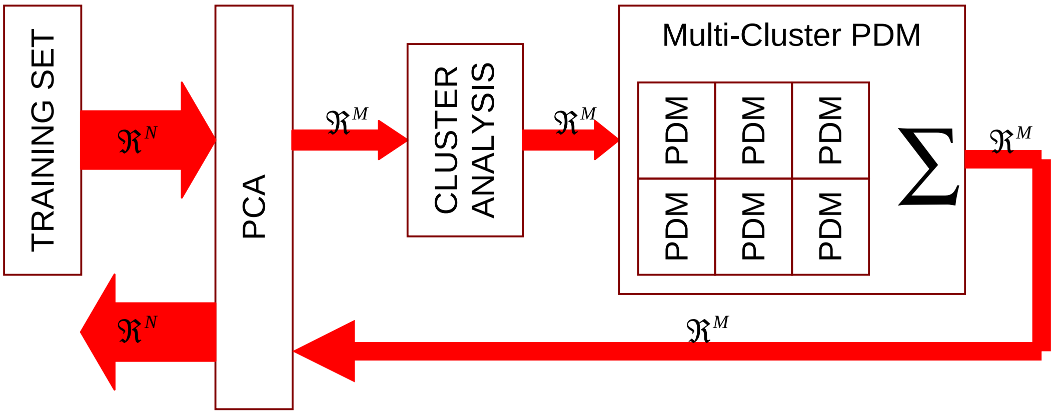


Figure 6.2.2 - Cluster Based non-linear Constraints on Shape Space

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 present

4. Use this natural number to segregate the data set into multiple clusters using fuzzy k-means

4. Perform PCA on each cluster of training set

> *PCA is performed on the reduced dimensionality cluster. Here models must be transformed to the reduced space at runtime, the closest allowable shape from the model reconstructed and transformed back to the original dimensionality.*

6.3 Evaluation

In principle, this procedure should produce identical results to that produced by applying the constraints to the original training set, with the added advantage of the computational saving of performing the constraints within the reduced space. However, in practice this approach performs better due to the data smoothing effect of the initial linear projection, which reduces the dimensionality. Each linear patch has a far lower dimensionality, hence the linear patch can be modelled to encompass all the deformation. The initial linear projection is where the data smoothing (lossy compression) occurs and as such the model's accuracy is limited by this single factor.

In order to assess the performance of the technique the experiments detailed in chapter 5.7 can be repeated and the error graphs produced. Returning to the robot arm example (chapter 5.5.1), after the initial dimensional reduction from $\Re^{32}$ to $\Re^{4}$ the reduced dimension training set is fuzzy-clustered in the same manner. From this data clustering PCA is performed on each linear patch in $\Re^{4}$ space. As the maximum number of eigenvectors for each cluster cannot exceed the dimensionality of the space, each cluster is constructed so as to encompass 100% of the deformation (i.e. all four modes are used). This means that no decisions need be made for the dimensionality of individual clusters and therefore simplifes the procedure of model construction.

The error metrics previously defined (section 5.7) are now used to assess the new model's ability at both reproducing valid shapes and constraining non-valid shapes.

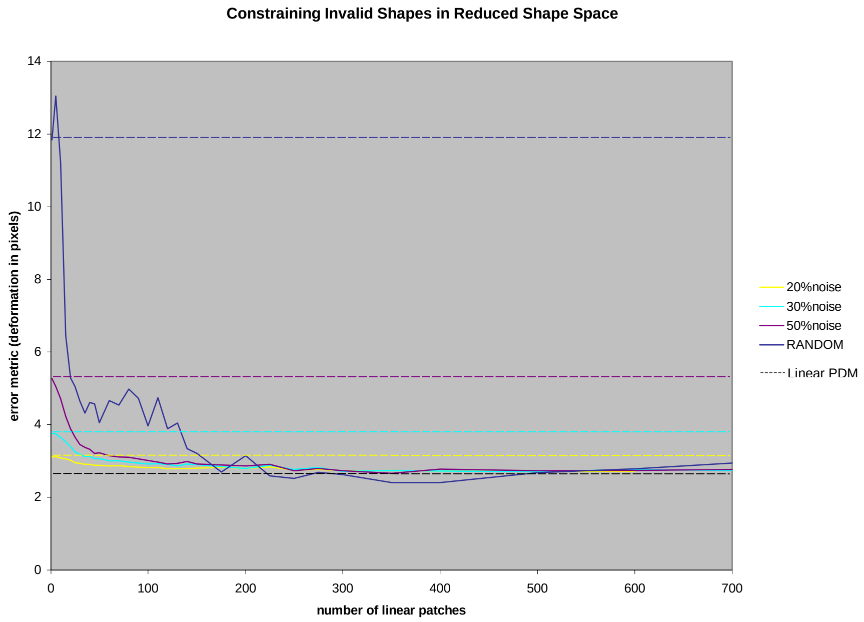


Figure 6.3.1 - Error graph showing ability to constrain non-valid shapes

Figure 6.3.1 demonstrates the result of measuring the performance of this new technique on the non-valid test set described in the previous chapter for increasing numbers of linear patches. This procedure was repeated for the original training set perturbed by noise and for a completely random training set. As would be expected, increasing the number of linear patches decreases the error rate and hence results in less invalid deformation being produced.

The dotted lines show the error produced by the single linear PDM in comparison to each of the data sets. The linear PDM produces identical results to that of using a single linear patch in the CSSPDM. The single cluster contains 100% of the deformation of the reduced data and therefore has no effect. Due to the dimensional reduction of the linear PDM being the same as that used in reducing the dimensionality of the data, the single cluster has no effect as they are essentially the same. However, because the linear PDM remains present throughout the technique the resulting loss of data ensures that no level of constraints can perform better than the 2.7 error rate produced by the information loss of the data projection. Surprisingly however, for models with between 200 and 500 linear patches, the technique does produce higher accuracy rates but only in the order of fractions of a pixel. By altering the dimensional reduction to utilise more eigenvectors in the initial projection and hence retaining more information from the model, this baseline error can be reduced further. However, this poses the same question as the linear PDM and the trade-off between accuracy and compactness/robustness (see section 3.2.6).

The important features of Figure 6.3.1 are that the error rate is significantly reduced by increasing the number of linear patches initially, and the most benefit can be deemed to be at around 20 linear patches which correlates with the initial analysis of the data set.

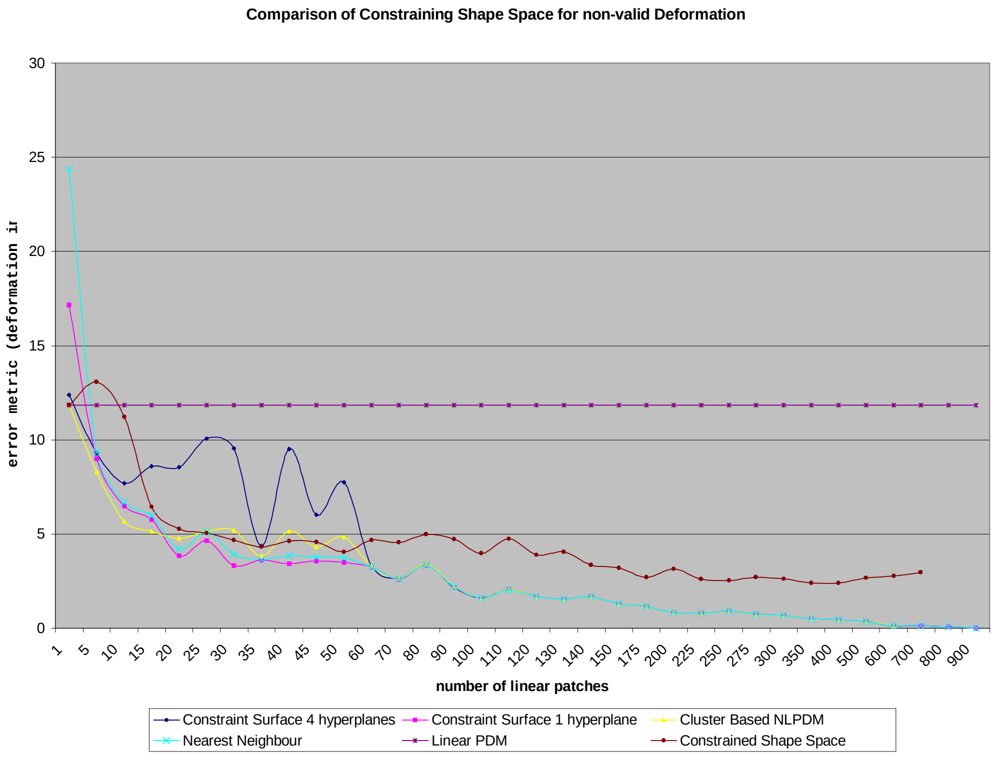


Figure 6.3.2 - Error graph showing comparison of Constraining Shape space against previously discussed Techniques

Figure 6.3.2 shows the error line produced from the random data set in Figure 6.3.1 superimposed upon the results of the previously discussed approaches from section 5.7. It can clearly be seen that although *Constraining Shape Space* does not produce the lowest error rates, it does perform comparably with the lower of the error plots generated by other techniques. Since it has already been established in the previous chapter that the CBNLPDM produces the most desirable results, the comparative performance of this solution is of primary concern. The data smoothing of the dimensional reduction can be attributed to the smoothed error graph produced by this technique. Although the error rate does not reach zero, like many of the other approaches, it follows the same trend until more than 60 linear patches are used. Since the model only utilises 20, this artefact of the approach can be disregarded, as model complexity would never reach this level. It is also important to bear in mind that the minimum error of the *Constrained Shape Space* approach can be reduced further by reducing the information loss of the dimensional reduction (the initial linear PDM projection) and including more information in the model i.e. using more eigenvectors.

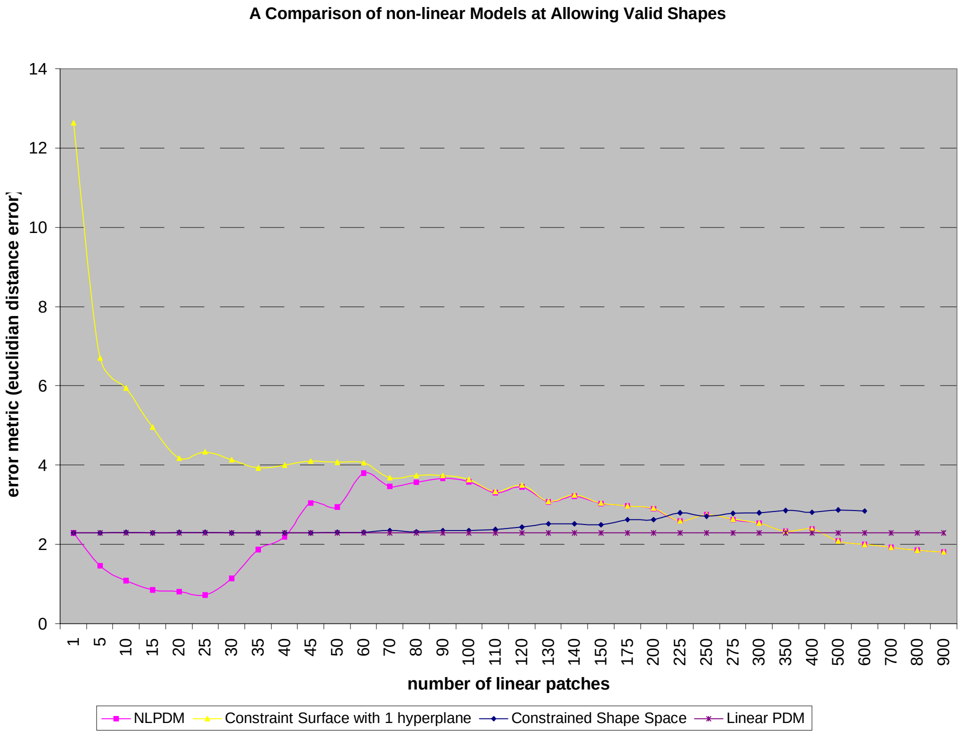


Figure 6.3.3 - Error graph showing ability to model valid shapes

Figure 6.3.3 shows the result of measuring the performance of this new technique upon the valid unseen test set (described in the previous chapter) for increasing numbers of linear patches. The performance would be expected to be comparable with the linear PDM model (as the initial projection is a linear PDM). Although the performance is not as high as the CBNLPDM, it performs significantly better than the constraint surface or nearest neighbour approach which are not shown on this graph due to the extremely high error rates they produce (around 45). In a similar manor to the linear PDM, the CSSPDM error rate can be further reduced by reducing the dimensional reduction of the initial projection to include more deformation. However, this in turn will increase the dimensionality of the model and hence computational complexity in analysis and runtime application. When the huge dimensional reductions that can be achieved for analysis are considered, this slight degradation in performance can be justified. In this example the reduction from 32 to 4 may not be considered advantageous but when larger dimensional examples are considered (examples in next section and later chapters) the benefits of this approach can be seen.

To summaries these techniques,

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
4. Perform PCA on each cluster of training set
```

When performing PCA on individual clusters two approaches can be taken.

(1) PCA can be performed on the reduced training set cluster. Here models must be transformed to the reduced space at runtime, the shape reconstructed and transformed back. This is slightly more computationally expensive, but has the advantage that the original encoding remains and therefore aids simple pose analysis/recognition.

(2) PCA can be performed on the original training set clusters after the clusters are transformed back into the original space. This technique is slower in analysis but faster at runtime and ensures that little high frequency information is lost.

6.4 Classification

6.4.1 Introduction

Due to the nature of constraining shape space, much of the segregation of the data set which is important to classification is contained within the model. In addition to this, the improved modelling capability of the non-linear estimation allows more complex problems to be tackled. If the assumption is made that similar poses of a model produce similar training vectors and each pose of the model corresponds to a point in shape space, it is therefore a fair assumption that similar poses of the model will produce tight clusters within this shape space. These clusters should automatically be modelled by the non-linear constraints that are placed on the model and facilitate more complex static pose recognition. The application of gesture recognition provides an ideal application for the proof of this assumption.

6.4.2 Sign Language & Gesture Recognition

American Sign Language or ASL has a finger spelt alphabet similar to other national sign languages. These simple gesture alphabets are used to spell names or words (letter by letter), for which there is no signing either known or present in the vocabulary. ASL provides a more suitable problem domain over British Sign Language as the BSL finger spelt alphabet is a two-handed system. Although this two handed system in reality provides a method of signing which is far easier to understand, it presents added difficulty for computer vision tasks due to the problems associated with occlusion.

Watson presented a review of work related to hand gesture interface techniques which consisted of glove sensor-based techniques, vision-based techniques and the analysis of drawing gestures [Watson93]. These were later summarised and techniques evaluated in by Handouyahia, Ziou and Wang [Handouyahia 99] and are discussed later in this chapter.

Figure 6.4.1 shows the ASL[2] alphabet with images taken from the training set.

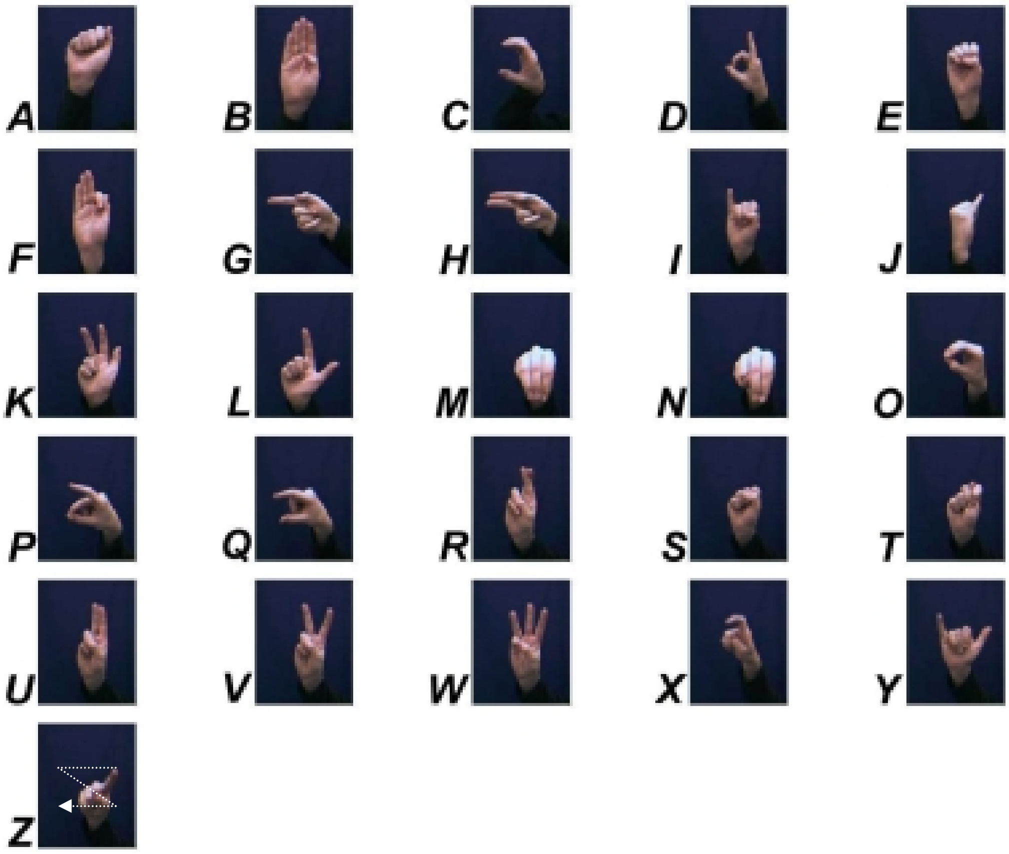


Figure 6.4.1 - The American Sign Language Finger Spelling Alphabet

It can clearly be seen from Figure 6.4.1 that each letter of the alphabet corresponds to a specific pose of the hand, with the exception of the letter 'z' which is a dynamic gesture and requires movement. This being the case, each gesture should occupy a distinct area in shape space.

6.4.3 Constructing the Non linear Hand Model

Several image sequences were recorded which encapsulated numerous occurrences of each of the letters of the alphabet. These sequences included three 'runs' through the alphabet, along with a small selection of simple sentences and words. These image sequences were recorded using a blue backdrop and sleeve to allow simple extraction using chroma key techniques.

[2] American Sign Language alphabet is almost identical to the alphabet of International Sign Language (ISL).

Once these sequences had been extracted, the hand was segmented to produce a binary image, and a contour-tracing algorithm initiated to extract the external contour of the hand for each image frame. Figure 6.4.2 shows: (a) a sample image frame of the hand; (b) the binary image produced from chroma keying; (c) and (d) the resulting extracted boundary. The procedure was then repeated for every image frame, providing training examples of the hand as it moves throughout the alphabet and the possible shapes it can take as it makes transitions between the letters.

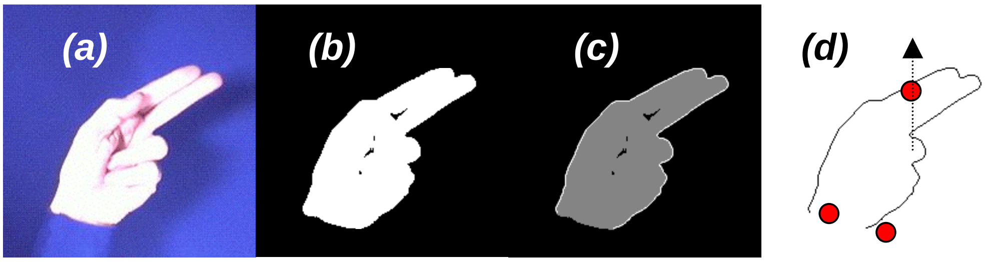


Figure 6.4.2 - Extracting Training Examples for ASL Data Set

(a) Hand image, (b) Segmented hand, (c) Extracted Contour (d) Resampled Contour

Before any statistical analysis can be performed, the training examples must first be resampled and aligned. The contour was automatically allocated 3 landmark points around the contour as shown in Figure 6.4.2(d). These landmark points were allocated at the start and finish of the contour and one at the vertical extremity within a 10° arc of the centeroid of the boundary. Once done, these landmarks were used to resample the boundary using linear interpolation to produce a contour consisting of 200 connected points. The low number of landmark points and the simple landmark identification used guarantees that non-linearity through non-optimum landmark point assignment will be present within the training set. However, this non-linearity will be modelled through the use of the *Constrained Shape Space* non-linear model discussed earlier. No rotational alignment was performed to preserve as much information about the pose of the model within the shape space. This again would introduce non-linearity into the model. The rotation non-linearity is necessary in the recognition of gestures.

Poses produced by the dynamic gesture ('z' for example) are similar to other gestures ('g') except for the rotation of the hand pose around the camera’s z-axis. If this rotation were to be removed, then the distinction between these two poses would be lost. Again the non-linear constraints will model this non-linearity and allow simple distinctions to be made.

Finally any translation of the hand model in the xy image plane was removed by translating the origin of the contour to that of the wrist, located by taking the mean of the start and finish points of the contour. This approach removes any translation of the hand in the image plane, but assumes that the hand is kept at a consistent distance from the camera throughout the training set and hence has no need to be scaled.

Once the training set had been prepared, a total of 7441 example contours were produced and labelled with the actual letter the pose corresponded to. Poses that were deemed transitory poses between real gestures were labelled as ***null*** gestures.

Under the normal procedure for the construction of a PDM, the last phase before PCA is performed would be to normalise all contour boundaries, ensuring a consistent training set. However, for reasons that have already been mentioned with regard to rotation, it is important that this information is preserved. Theoretically the length of vectors on which PCA is performed should not affect the resulting model except for its overall size. However, due to the nature of floating-point arithmetic and the problems associated with overflow errors, it is still necessary to reduce the size of the computations. This is facilitated by dividing each training vector not by its own length (as in normalisation), but by the length of the mean vector of the training set. This effectively normalises the training set but retains any subtle size deviations between examples.

6.4.4 The Linear ASL Model

The Linear ASL model is now generated by performing linear PCA upon the training set. Figure 6.4.3 shows the primary modes of the linear ASL PDM and how these modes deform the model from the mean.

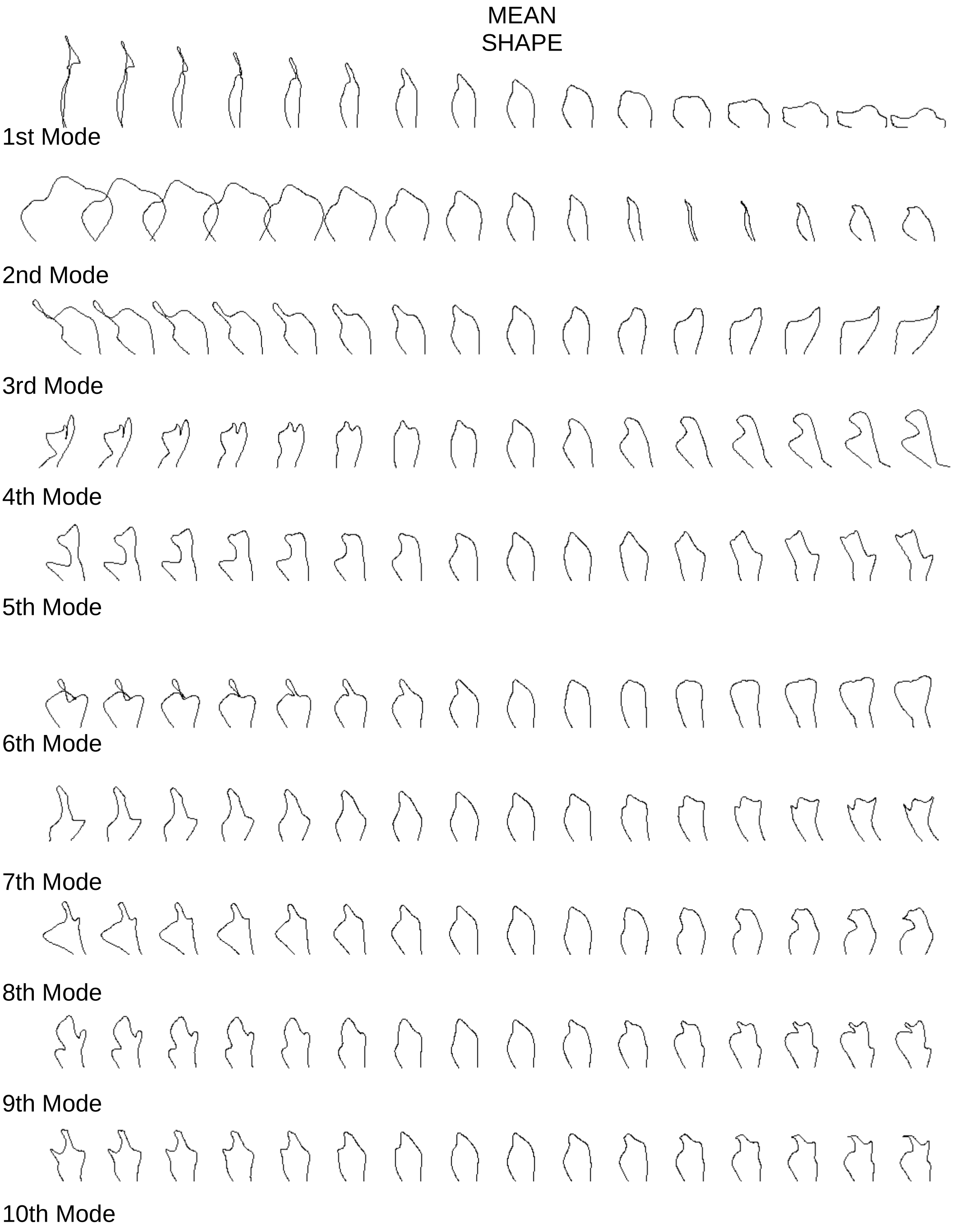


Figure 6.4.3 - The linear ASL PDM Model

It can clearly be seen that the major modes of variation include large amounts of deformation which, when put together, will produce an unreliable model capable of producing far too much deformation (see examples in Figure 6.4.4)

By analysing the eigenvalues of the covariance matrix it can be determined that the first 30 eigenvectors corresponding to the 30 largest eigenvalues encompass

99.6% of the deformation within the model. Unfortunately, due to the natural rotational non-linearity and high order non-linearity which has been introduced into the model during re-sampling (as discussed in the previous section), this linear model is unsuitable for tracking and classification. Figure 6.4.4 shows a selection of invalid shapes that can be constructed from the linear ASL PDM. These examples were produced by generating random vectors that were within the bounds of the linear model. It is the linear PDM's ability to allow invalid shapes which make the model unreliable for tracking and classification. These invalid deformations are due to the linear approximation of the non-linear data set.

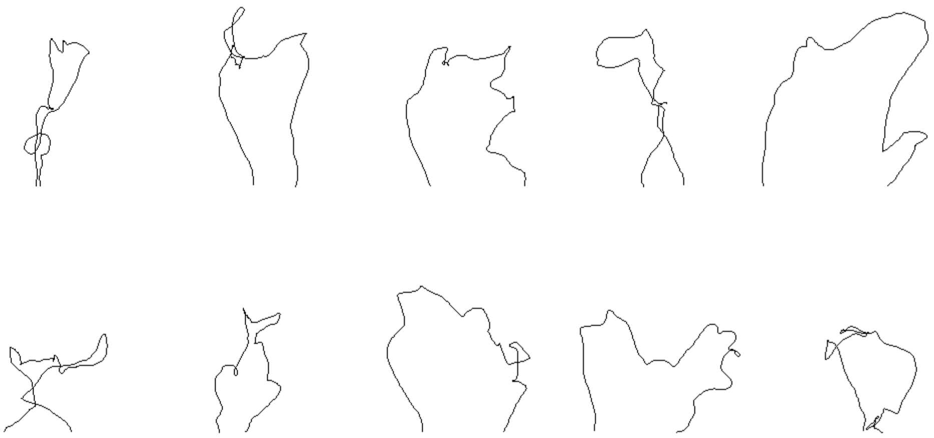


Figure 6.4.4 - Example Invalid Shapes produced by the linear ASL PDM

6.4.5 Adding non-linear Constraints

Using the procedure previously outlined, non-linear constraints to the model are added by performing cluster analysis on the dimensionally reduced data set after it has been projected down into PCA space. From the linear model it has been determined that the 30 primary modes encompass 99.6% of the deformation, by projecting each of the training vectors down into this space (as previously described), a dimensional reduction of 400 to 30 is achieved. Cluster analysis is now performed upon the reduced data set.

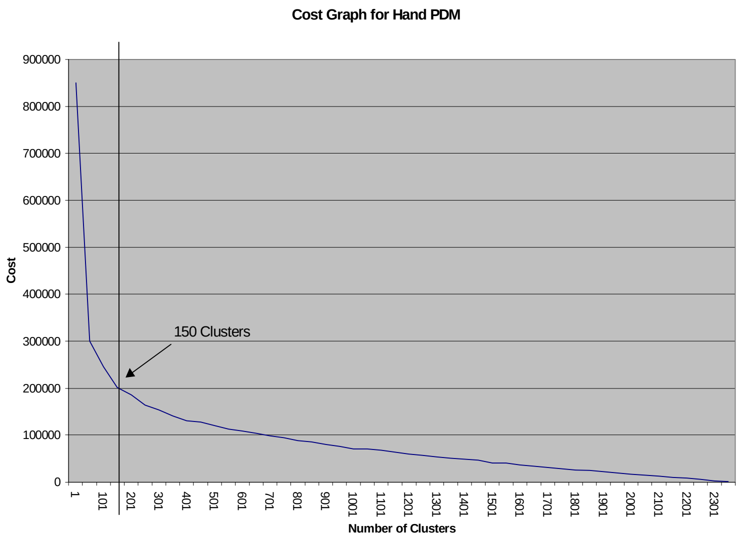


Figure 6.4.5 - Cluster Analysis on Dimensional Reduced ASL Training Set

Figure 6.4.5 shows the resulting cost graph from the cluster analysis of the reduced data set and the natural number of clusters estimated to be 150. The fuzzy k-means algorithm is then used to segregate the space into 150 clusters. These clusters are then learnt by performing PCA on their members.

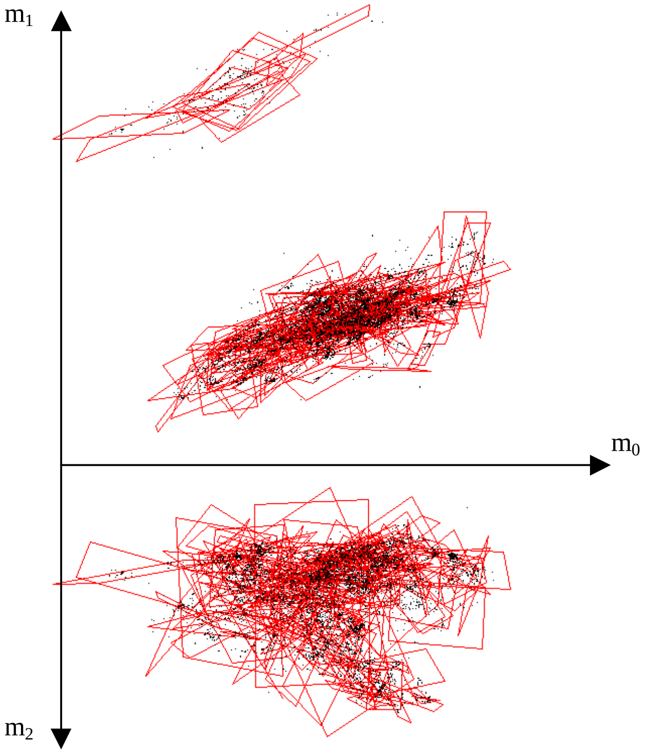


Figure 6.4.6 - Constrains on PCA space for the ASL Model

Figure 6.4.6 shows the PCA space for the model projected into 3 dimensions for visualisation purposes, with the constraints shown as the bounding boxes (first two primary modes) of the linear patches (clusters) extracted via PCA. Notice the two distinct clusters produced in the direction $m_{1,}$ meaning that the shape space is discontinuous and there is no smooth path between the two distinct areas of shape space. This is due to the simple landmark identification and the problems associated with it. Further discontinuities may exist in the model which are not apparent in the dimensions that are shown in Figure 6.4.6. These types of spaces and solutions to the problems they introduce will be discussed in the chapter on temporal dynamics (specifically sections 7.3-7.4 for the ASL shape space)

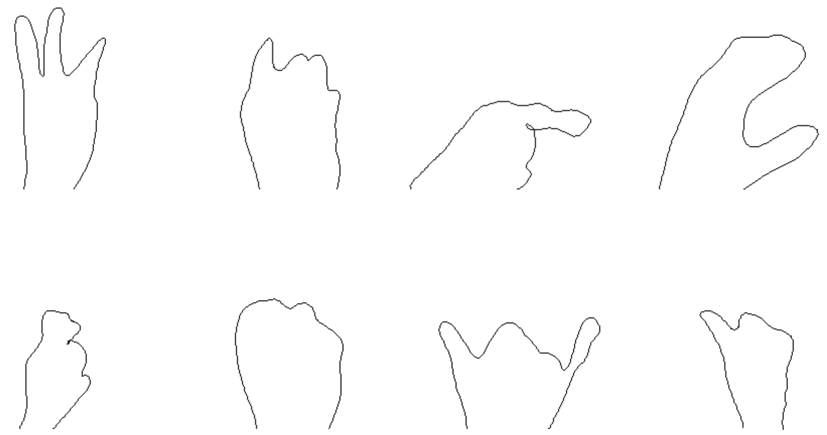


Figure 6.4.7 - Example Shapes Produced by the constrained non-linear ASL PDM

Figure 6.4.7 shows random shapes generated within the constrained model, If these are compared with those produced in Figure 6.4.4, it can be seen that the constrained model contains far less invalid deformation and therefore results in a more reliable model for tracking. Each random shape is also very close to a natural gesture in ASL and it is this correlation between cluster and gesture that can be used to perform gesture recognition.

6.4.6 Recognising Gestures

Ideally for an alphabet with 26 characters, the model would contain 26 clusters, where each cluster directly corresponds to a specific letter. However the non-linearity of the model requires far more clusters to encompass the deformation reliably. As a result, multiple clusters may correspond to a single letter. This is due to:

1. The presence of null (transitional) poses of the hand within the training set should not correspond directly to any specific letter. As these null poses will be distributed throughout the space it is incorrect to assume that it is possible to model them with a single cluster.
2. The landmark point assignment used may result in two very similar poses of the model occupying completely different areas of the PCA space (i.e. discontinuous shape space). Therefore, again, it is incorrect to assume that any single letter will produce a single tight cluster.
3. The presence of dynamic gestures like 'z' requires movement of the hand to complete the gesture. This movement results in a trajectory in PCA space that corresponds to a letter rather than a cluster. This trajectory may require multiple clusters in order to model the deformation.

Once these issues are considered it is apparent that in order to classify any specific gesture, multiple clusters must be assigned to each letter rather than single clusters as previously used in previous work by the author [Bowden 96; Bowden 97]. This can be achieved by analysing the training set and probabilistically assigning each cluster to a specific letter. This provides a conditional probability that the model represents a letter given that model is in any specific cluster. These conditional probabilities are constructed in a probability matrix as shown in Figure 6.4.8.

$$
\begin{array}{c|ccccc}
 & letterA & letterB & \cdots & letterZ & null \\
\hline
Cluster_1 & p_{C_1A} & p_{C_1B} & \cdots & p_{C_1Z} & p_{C_1null} \\
Cluster_2 & p_{C_2A} & p_{C_2B} & \cdots & p_{C_2Z} & p_{C_2null} \\
Cluster_3 & p_{C_3A} & p_{C_3B} & \cdots & p_{C_3Z} & p_{C_3null} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
Cluster_{149} & p_{C_{149}A} & p_{C_{149}B} & \cdots & p_{C_{149}Z} & p_{C_{149}null} \\
Cluster_{150} & p_{C_{150}A} & p_{C_{150}B} & \cdots & p_{C_{150}Z} & p_{C_{150}null}
\end{array}
$$

Figure 6.4.8 - Probability Matrix for ASL Classification

As each of the vectors from the training set has been pre-assigned a letter which provides a label for each shape of the training set, the matrix can be constructed by calculating which cluster a specific training example belongs to, and assigning that cluster to the labelled letter. Each training example is projected down into the PCA space and the closest cluster, α, located. The value along the row α, $P_{\alpha\beta}$ which corresponds to the letter β is then incremented. This procedure is carried out for the entire training set and each row normalised to calculate the conditional probability that any cluster belongs to a letter i.e. $\sum_i P(letter | Cluster_i) = 1$. Now by locating which cluster the model exists in there is a conditional probability that the model is representing a letter, with the highest probability for a cluster representing the most likely letter. By analysing this matrix information about how this correlation is achieved can be extracted. Table 6.4-1 shows how many clusters each letter uses in this mapping.

Letter	N' Clusters	Letter	N' Clusters
a	23	O	16
b	9	P	9
c	8	Q	10
d	9	R	15
e	26	S	15
f	13	T	20
g	14	U	15
h	13	V	5
i	12	W	2
j	11	X	7
k	7	Y	9
l	4	Z	26
m	11	NULL	130
n	8		

Table 6.4-1 - Correlation between ASL Gestures and Clusters in non-linear Model

Cluster	N'	Cluster	N'	Cluster	N'	Cluster	N'	Cluster	N'	Cluster	N'
N'	Letters	N'	Letters	N'	Letters	N'	Letters	N'	Letters	N'	Letters
1	4	27	2	53	4	79	3	105	4	131	2
2	2	28	3	54	3	80	3	106	2	132	5
3	2	29	3	55	1	81	6	107	2	133	4
4	3	30	4	56	2	82	3	108	2	134	2
5	4	31	1	57	2	83	2	109	4	135	1
6	2	32	2	58	2	84	2	110	5	136	7
7	5	33	2	59	7	85	3	111	3	137	2
8	4	34	3	60	3	86	2	112	2	138	2
9	5	35	1	61	1	87	2	113	2	139	3
10	5	36	3	62	2	88	5	114	3	140	3
11	2	37	2	63	2	89	2	115	4	141	2
12	4	38	3	64	1	90	5	116	3	142	2
13	2	39	4	65	3	91	3	117	4	143	2
14	3	40	3	66	4	92	5	118	5	144	0
15	1	41	4	67	6	93	1	119	4	145	5
16	4	42	3	68	2	94	1	120	3	146	3
17	3	43	2	69	4	95	2	121	2	147	3
18	6	44	4	70	2	96	2	122	2	148	4
19	6	45	2	71	4	97	4	123	2	149	5
20	4	46	4	72	6	98	4	124	3	150	4
21	3	47	2	73	3	99	4	125	2	**Average**	**2.98**
22	1	48	2	74	3	100	4	126	3		
23	1	49	7	75	1	101	1	127	3		
24	1	50	2	76	4	102	2	128	4		
25	2	51	5	77	3	103	3	129	1		
26	1	52	3	78	3	104	3	130	1		

Table 6.4-2 - Correlation between Clusters of non-linear model and ASL Gesture

Table 6.4-2 shows the number of ASL gestures that correspond to each cluster. It would be expected that each cluster would correspond to only one letter, however due to inconsistencies in labelling and the complexity of the model this is not the case. The average cluster corresponds to 2.98 letters, but the matrix gives us a probability that the cluster corresponds to a specific letter; The highest probability entry in the matrix gives the best estimate to the recognised letter.

	Highest Probabilistic Match	Second Highest Probabilistic Match
Minimum	0.285714	0
Maximum	1	0.454545
Mean	0.706031	0.210881

Table 6.4-3 - Analysing the Resulting Probabilities

Table 6.4-3 shows the range of probabilities that result for this procedure. Using an unseen test set of segmented hand shapes with (hand labelled) letter ground truth for comparison, the average probability for the best match of the matrix is around 0.7. The maximum value of 1 demonstrates that some clusters exclusively belong to specific gestures and this can be confirmed by the presence of clusters assigned to only one cluster in Table 6.4-2. The next highest probability from the matrix is also shown with the mean value being much lower than that of the best match, demonstrating that although there is some ambiguity between gestures there is significant distinction probabilistically as to the function of each cluster.

By comparing the resulting highest probability match with the original labelled letter for each of the training examples and converting this to a percentage, a measure of the classifications accuracy can be determined.

Out of a total of 4741 examples the highest probability match was correct in 3348 cases, with the second highest probability match being correct in 1000 cases. This gives a 70.62% accuracy for the most likely match, with 20.09% accuracy for the next most likely match. From this it can be said that there is a

91.71% chance that the correct letter for each pose will be recognised as one of the two highest probability matches from the matrix.

6.5 Evaluation

Initially these results may not seem overwhelming, however the complexity of performing such a task using computer vision is considerable due to the variability of the hand and the problems associated with accurately segmenting or extracting features which represent its shape. If other approaches are considered this becomes apparent. Table 6.5-1 [Handouyahia 99] summarises other authors approaches the problem.

Authors/ Properties	Size of Vocab	Type of Vocab	Capture	Representation	Recognition	Success Rate %
Gourley[3]	26	ASL[4]	Elect[5]	Templates	Perceptron Neural Network	95
Harling[3]	5	ASL[4]	Elect[5]	Templates	Perceptron Neural Network	96
Murkami[3]	42	JSL[6]	Elect[5]	Templates	Perceptron Neural Network	98
Takahashi[3]	46	JSL[6]	Elect[5]	Joint and orientation coding	Template Matching	65
Gao[3]	13	D.Set[7]	Camera	Convex/Concave coding	Backpropogation Network	80
Uras[3]	25	ISL[8]	Camera	First size functions family	K-Nearest Neighbour	85
Uras[3]	25	ISL[8]	Camera	Second size functions family	K-Nearest Neighbour	86
Freeman[3]	15	D.Set[7]	Camera	Orientation Histograms	K-Nearest Neighbour	75
Handouyahia[3]	25	ISL[8]	Camera	Moment Based Size Functions	Perceptron Neural Network	90
Our Method	26	ASL[4]	Camera	NL Point Distribution Model	Fuzzy Nearest Neighbour	71(92)

Table 6.5-1 - Table Showing a Summary of Gesture Recognition Methods

The highest accuracy rates are achieved using an electrical sensor based data glove as an input device. Those techniques that rely upon computer vision perform less well. The higher accuracy's are also generated for systems which use neural networks to provide the mapping between feature space and gesture space. If the simplicity of the CSSPDM augmented with the conditional probabilities which provide the gesture recognition is considered then the attraction of this approach becomes apparent.

[3] Details of the authors work are contained in and Handouyahia 99 and Watson 93

[4] American Sign Language

[5] Electronic sensor based glove

[6] Japanese Sign Language

[7] The type of the vocabulary is pre-defined

[8] International Sign Language

It is also important to note that the CSSPDM is assessing the model at every frame and attempting to recognise the gesture contained there in. This assessment of each frame is static. No temporal or contextual information is used. Further constraints could be applied from the English Language to increase accuracy (see Chapter 7). Since humans tend to pause slightly at each gesture, the accuracy could be further increased by accumulating probabilities over time, i.e. consecutive frames would 'vote' towards the current gesture, further reducing the effect of noise.

Selected Feature/Criteria	Scale Invariant	Translation Invariant	Rotation Invariant	Lighting Invariant	Robust to N' of Fingers	Computational Complexity
Basic Chain Code[3]	No	Yes	No	No	No	Low
Convex-Concave Coding[3]	Yes	Yes	Yes	No	Yes	Low
Fourier Desc.[3]	No	No	No	No	No	Low
Hu Invariant Moments[3]	Yes	Yes	Yes	No	No	High
Alt Invariant Moments[3]	Yes	Yes	No	No	No	High
Principal axes[3]	Yes	Yes	No	No	No	Low
Grey Level Histogram[3]	No	Yes	Yes	No	No	Low
Hist. Of Local Orientation[3]	Yes	Yes	No	No	No	Low
Size Functions[3]	Yes	Yes	No	Yes	Yes	High
Moment Based Size Funct[3]	Yes	Yes	No	Yes	Yes	Low
Authors Method[3]	Yes	Yes	Yes	Yes	Yes	Low

Table 6.5-2 - Table Showing the Evaluation of Features used in Various Gesture Recognition Methods

The CSSPDM naturaly lends itself to the probabilistic classification of pose, however if the CSSPDM is compared to other features used in Gesture Recognition, its benefits can clearly be seen. Table 6.5-2 [Handouyahia 99] summarises features used by other methods.

Unlike other approaches the CSSPDM is:

1. **Scale Invariant**: Gestures can be executed by different people with different hand sizes.
2. **Translation Invariant**: The location of the hand in the image plane can change.

3. **Rotation Invariant**: The hand can rotate around the cameras z-axis, other rotations of the hand can be incorporated into the deformation of the model.
4. **Lighting Invariant**: The illumination and background of the scene can change.
5. **Robustness to number of fingers**: Additional training data can be incorporated into the model to allow for individual changes in hand shape and gesture.
6. **Computation Complexity**: The simplicity of the linear mathematics and single layer of conditional probability means the method is fast to compute.

6.6 Conclusions

This chapter has demonstrated that by projecting the dataset through a linear PDM and hence reducing the overall dimensionality of the problem before further non-linear constraints are applied, several benefits are gained:

1. The data is smoothed before constraints are applied, producing better results in the final model.
2. The data reduction of the CSSPDM produces a significant computational saving over the CBNLPDM at the cost of accuracy. However this accuracy can easily be controlled to ensure model precision is maintained.
3. Construction is simplified as only one decision need be made as to the information loss of the model. In CBNLPDMs each cluster requires a different number of eigenvectors to achieve the required accuracy while compressing the data. However, CSSPDMs need not be concern with the local dimensionality of clusters as the initial projection allows each linear patch to model 100% of the deformation of that cluster.

Furthermore, it has been shown that, although the nature of the space is complex, simple classification techniques can be applied to perform static recognition of object shape and pose. These models allow deformable models to be constructed which, under the linear constraints of a simple PDM, would fail to be robust enough for “Real World” applications.

One important consideration is that as models become more complex, the simple gradient descent approach used on linear models begins to fail. These issues will be addressed in the next chapter.