Learning Statistical Models of Human Motion

Richard Bowden
Dept Systems Engineering
Brunel University
Uxbridge
Middlesex, UK
UB8 3PH
richard.bowden@brunel.ac.uk

Abstract: Non-linear statistical models of deformation provide methods to learn a priori shape and deformation for an object or class of objects by example. This paper extends these models of deformation to that of motion by augmenting the discrete representation of piecewise non-linear principle component analysis of shape with a markov chain which represents the temporal dynamics of the model. In this manner, mean trajectories can be learnt and reproduced for either the simulation of movement or for object tracking. This paper demonstrates the use of these techniques in learning human motion from capture data.

1: Introduction

Mathematical models of how an object or class of object deform and move with time are important in both computer vision and in graphical simulation. By maintaining an internal representation of shape and motion, object location and tracking can be simplified, or realistic motion can be generated. The key to this approach is the ability to learn what is shape and motion from a training set of examples. In doing so the process must generalize the content of the training set. Recent years have seen extensive work into deformable models based upon the principal component analysis of shape. This paper will demonstrate the use of such techniques in learning models of both deformation and temporal dynamics and their applications to human simulation.

2: Point Distribution Models and Eigenspaces

Point Distribution Models (PDMs) or Eigen Models have proven themselves an invaluable tool in image processing. The ***classic formulation*** combines local edge feature detection and a model-based approach to provide a fast, simple method of representing an object and how its structure can deform [8]. For a 2D contour, each pose of the object is described by a vector $\mathbf{x}_i \in \Re^{2n} = (x_1, y_1, \ldots, x_n, y_n)^T$, representing a set of points specifying the object shape. A training set **E** of *N* vectors is then assembled for a particular model class. The training set is aligned (using translation, rotation and scaling) and the mean shape calculated. To represent the deviation within the shape of the training set, Principal Component Analysis (PCA) is performed on the deviation of the example vectors from the mean using eigenvector decomposition on the covariance matrix of **E [**7]

PCA projects the data into a linear subspace with a minimum loss of information by multiplying the data by the eigenvectors of the covariance matrix (**S**). By analysing the magnitude of the corresponding eigenvalues the minimum dimensionality of the space on which the data lies can be calculated and the information loss estimated [4].

The *t* unit eigenvectors of **S** corresponding to the *t* largest eigenvalues supply the variation modes; *t* will generally be much smaller than *N*, thus giving a very compact model and it is this dimensional reduction that facilitates non-linear analysis. A deformed shape **x** is generated by adding weighted combinations of v_j to the mean shape,

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_{j=1}^{t} b_j \mathbf{v}_j$$

where b_j is the weighting for the j^{th} variation vector. Suitable limits for b_j are $\pm 3\sqrt{\lambda_j}$, where λ_j is the j^{th} largest eigenvalue of **S** [3]. This provides a compact mathematical model of how the shape deforms.

The formulation of the PDM can also be expressed in matrix form [8]

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Pb}$$

where $\mathbf{P} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t)^T$ is a matrix of the first *t* eigenvectors and $\mathbf{b} = (b_1, b_2, \ldots, b_t)^T$ is a vector of weights.

This mathematical model is used to constrain the shape of the PDM when used in object tracking by constraining the shape of the model to lie within the bounds of the model. Given a new shape $\mathbf{x}'$, the closest allowable shape from the model is constructed by finding **b** such that

$$\mathbf{b} = \mathbf{P}^{-1}(\mathbf{x}' - \bar{\mathbf{x}}) \text{ where } -3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i}$$

The closest allowable shape $\mathbf{x}''$ can then be reconstructed as

$$\mathbf{x}'' = \bar{\mathbf{x}} + \mathbf{Pb}$$

The linear formulation of the PDM relies on the assumption that similar shapes produce similar vectors. This being the case, it is a fair assumption that the training set will generate a cluster in some shape space. However, it is unfair to assume that this cluster will be uniform in shape and size. As more complex models are considered the training set may even generate multiple, separate clusters in the shape space.

Under these circumstances the linear PDM will begin to fail as non-linear training sets produce complex high dimensional shapes which, when modeled through the linear mathematics of PCA, produce unreliable models.

2.1: Non-linear Models of Deformation

A number of authors have addressed the problems associated with the construction of non-linear PDMs. Where rotational non-linearity is known to be present within a model this can be removed/reduced by mapping the model into an alternative linear space. Heap and Hogg suggested using a log polar mapping to remove non-linearity from the training set [9]. This allows a non-linear training set to be projected into a linear space where PCA can be used to represent deformation. The model is then projected back into the original space. Although a useful suggestion for applications where the only non-linearity is pivotal and represented in the paths of the landmark points in the original model, it does not provide a solution for the high non-linearity generated from other sources.

Higher order non-linearity is often the result of incorrect labelling of training examples. By carefully selecting landmark points by hand, a near optimum labelling can be achieved which will minimise the non-linearity of a training set. However, for all but the most simple of cases this is not a feasible solution. Often semi-automated procedures are used where a user can speed up the process of labelling example shapes for analysis. Fully automated procedures are rarely used due to the problems of correctly assigning landmarks and the highly non-linear models that this produces.

Work done by Baumberg and Hogg goes some of the way to solving non-linearity in deformable models by using a B-Spline representation. Landmark points for the Spline are represented as a PDM [1]. The curvature of the B-Spline takes on some of the non-linearity of the model and therefore reduces the problems associated with using a linear PDM to represent a non-linear model.

It has been proposed by Kotcheff and Taylor that non-linearity introduced during assembly of a training set could be eliminated by automatically assigning landmark points in order to minimise the non-linearity of the corresponding training cluster [14]. This can be estimated by analysing the size of the linear PDM that represents the training set. The more non-linear a proposed formulation of a training set, the larger the PDM needed to encompass the deformation. The procedure was demonstrated using a small test shape and scoring a particular assignment of landmark points according to the size of the training set (gained from analysis of the principal modes and the extent to which the model deforms along these modes, i.e. the eigenvalues of the covariance matrix). This was formulated as a minimisation problem, using a genetic algorithm. The approach performed well but at a heavy computation cost [14]. As the move to larger, more complex models or 3D models is considered, where dimensionality of the training set is high, this approach becomes unfeasible. A more generic solution is to use accurate non-linear representations. As linear PCA is used for linear PDMs, so, non-linear PCA can be used to model non-linear PDMs and many researchers have proposed approaches to this end.

Sozou et al first proposed using polynomial regression to fit high order polynomials to the non-linear axis of the training set [12]. Although this compensates for some of the curvature represented within the training set, it does not adequately compensate for higher order non-linearity, which manifests itself in the smaller modes of variation as high frequency oscillations. In addition, the order of the polynomial to be used must be selected and the fitting process is time consuming.

Sozou et al further proposed modeling the non-linearity of the training set using a backpropagation neural network to perform non-linear principal component analysis [13]. This performs well, however the architecture of the network is application specific; also, training times and the optimisation of network structure are time consuming. What is required is a means of modeling the non-linearity accurately, but with the simplicity and speed of the linear model.

Several researchers have proposed alternatives, which utilise non-linear approximations, estimating non-linearity through the combination of multiple smaller linear models [2][4][6][7][10]. These approaches have been shown to be powerful at modeling complex non-linearity in extremely high dimensional feature spaces [2][3].

The basic principal behind all these approaches is to break up any curvature into piecewise linear patches, which estimate the non-linearity rather than modeling it explicitly. This is akin to the polygonal representation of a surface. A smooth curved surface can be estimated by breaking it down into small linear patches. In the field of computer graphics this technique is performed to reduce render time. There exists, of course, a trade off between visual accuracy and computation speed (where the minimum numbers of polygons are used to achieve the desired appearance). The same problem is present in non-linear PDM estimation, where the minimum number of linear patches that accurately represent the model must be determined.

Bregler and Omohundro suggested modeling non-linear data sets of human lips using a *Shape Space Constraint Surface* [6]. The surface constraints are introduced to the model by separating the shape surface into linear patches using cluster analysis. However the dimensionality of these '*lip*' shape spaces is low, as is the non-linearity due to the simplified application of the work.

Cootes and Taylor suggested modeling non-linear data sets using a gaussian mixture model, which is fitted to the data using Expectation Maximisation [7]. Multiple gaussian clusters are fitted to the training set. This provides a more reliable model as constraints are placed upon the bounds of each piecewise patch of the shape space, which is modeled by the position, and size of each gaussian.

Both of these estimation techniques become unfeasible as dimensionality and training set size increase. However by projecting the training set down into the linear subspace as derived from PCA the dimensionality and therefore computation complexity of the non-linear analysis can be reduced significantly to facilitate statistical and probabilistic analysis of the training set. This projection relies upon the dimensional reduction of PCA while retaining the preservation of the important information, the shape of the training set [2][3][4].

In a previous publication by the authors this approach to the piecewise linear approximation of non-linear deformation has been used to model the complex deformation of the human body for object tracking. The model constraints are shown in Figure 1 and demonstrates how the model has learnt about the movement of the skeletal structure and how this relates to object tracking [3].

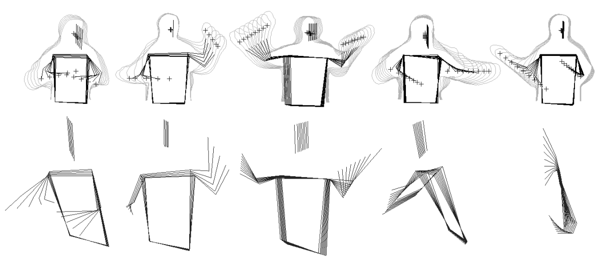


Figure 1 – A nonlinear deformable model of the human body

The remainder of this paper will investigate the extension of this approach with temporal constraints which can be used for supporting multiple hypotheses during tracking or reproducing models of human motion.

3: Learning a Motion Model

The use of computer vision techniques in motion capture is common place in acquiring trajectories for key points of objects that are used to produce life-like 3D animations.

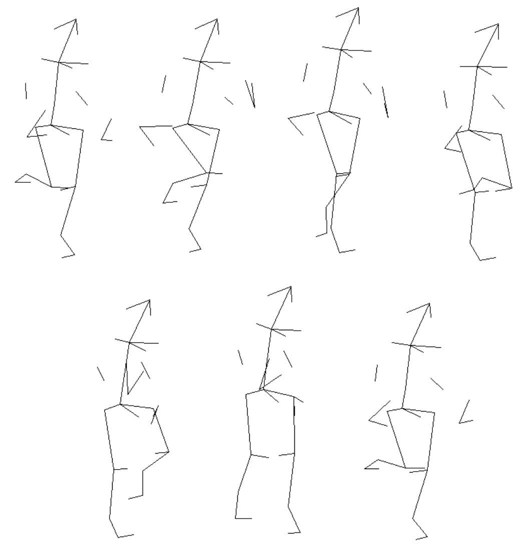


Figure 2 - Examples from a Key-frame animation of a Running Woman

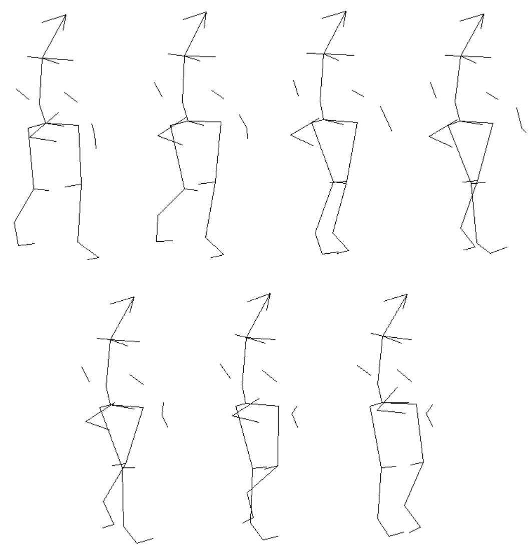


Figure 3 - Examples from a Key-frame animation of a Walking Woman

Figure 2 and figure 3 show motion trajectory files for a running and walking human female. These were captured using retro-reflective IR markers on a real world human subject.

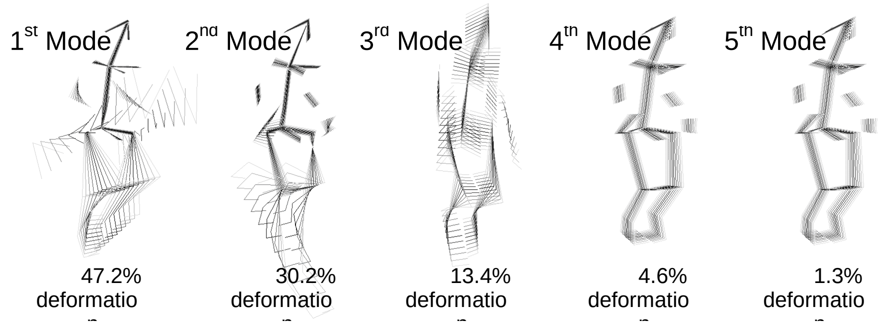


Figure 4 - The Running Linear 3D PDM

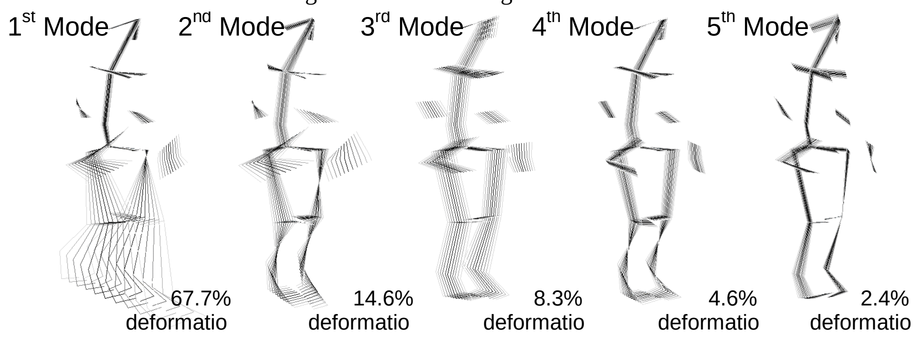


Figure 5 - The walking Linear 3D PDM

By connecting these points with a simple stick model the human motion can be visualized. In computer animation, these key points would be used to animate the articulated sections of a 3D virtual character for computer games or virtual environments.

It is this notion of *key points* in the motion capture process that provides the link between statistical models and animation, where animation key points are akin to the landmark points used in statistical models. If statistical models of shape and deformation can be ***learnt*** from a training set, producing realistic constraints on the shape (or motion of landmark points), then similar ***learnt*** models of animation trajectories can also be achieved.

3.1: The Linear Motion Model

The human motion capture[1] data for both the running and walking woman consists of 32 key points for each frame of the animation, these points can be concatenated into a single 96 dimensional vector $V=(x_1,y_1,z_1, ..., x_{32}, y_{32}, z_{32})$. The running animation consists of 474 key frames recorded at 30Hz which produces a training set of 474, 96 dimensional vectors. The walking animation consists of 270 key frames, again captured at 30Hz using 32 key points producing a training set of 270, 96 dimensional vectors. Now the training sets are in a form that enables further statistical analysis, linear PCA can be performed upon them to produce a linear 3D PDM.

[1] Motion Capture Data Provided by Televirtual

From the eigenvalue analysis, 98.8% of the deformation of the running model is contained within the first 10 eigenvectors, with 99.4% of the walking model being encompassed by the 10 eigenvectors.

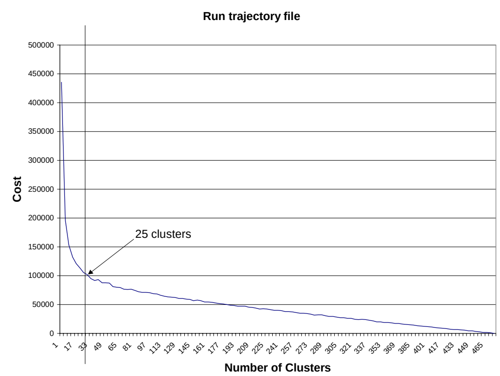
It can be seen from Figures 4 and 5 that the linear 3D PDM does not model the trajectories of key points (and associated body parts) well. The motion files contain perfect landmark point identification between examples. However, the data sets are still non-linear due to the circular motion of body parts. This non-linearity can be seen in figures 7,8 and will be discussed shortly.

3.2: Adding Non-linear Constraints

Although this linear model does not represent the motion represented within the training set it is invaluable in reducing the dimensionality of the data. This is performed by projecting each of the training examples down onto the eigenvectors of the linear PDM. Using the 10 primary modes of the linear model as determined in the previous section, both the running woman data and the walking model are projected down from 96 to 10 dimensions, where a new reduced shape vector $\mathbf{x}_R \in \Re^{10} = \mathbf{P}^{-1}\left(\mathbf{x}-\bar{\mathbf{x}}\right)$ and $\mathbf{P}=(\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_{10})^T$ is a matrix of the first 10 eigenvectors. These lower dimensional data sets are shown in figure 7 as points drawn in 3D from two 2D views.

It has been shown that this initial dimensional reduction produces minimal model degradation and often increases the ability of the model to generalize deformation[2].

A standard k-means clustering algorithm was then performed on the reduced data sets and the cost recorded. The resulting cost files are shown in figure 6. These graphs allow the estimation of the natural number of clusters within the dataset. As the number of clusters increases so the overall cost decreases. However, there becomes a point where increasing the number of clusters further no longer provides a significant reduction in cost, this is said to be the natural number of clusters of a dataset. By analysing these cost graphs the natural number of clusters for the run and walk trajectory files can be estimated to be 25 and 30 respectively. The larger number for the walking model is due to the model translation introduced as the subject establishes a consistent gait, and will be seen shortly.



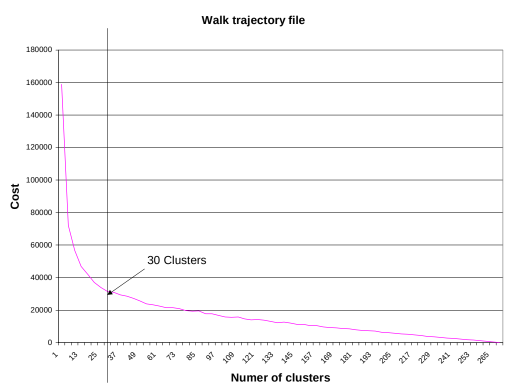
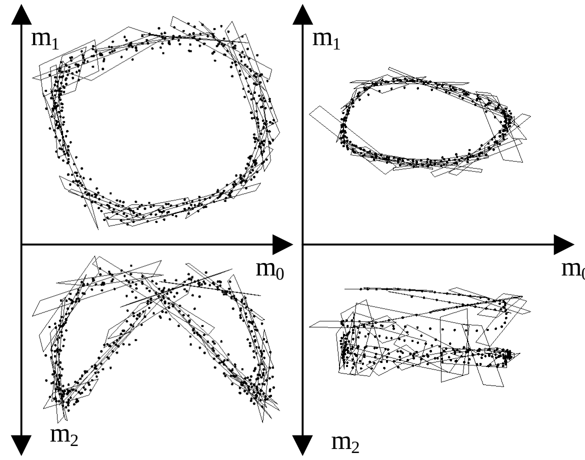


Figure 6 - Cost files for Trajectory Data

Using the natural number of clusters for each data set, the fuzzy k-means algorithm was used to segregate each data set into its composite clusters. Each cluster was then modeled by performing further PCA upon its members. The final non-linear constraints can be seen in figure 7 with the bounds of each cluster drawn as a rectangle over the reduced data set.

From this diagram it can be seen that the clustering algorithm has smoothly estimated the natural curvature of the data set through piecewise linear patches. Each cluster better estimates the model locally, as each linear patch must encode less information.

The Constrained Shape Space PDM (CSSPDM[2] has **learnt** the Motion Capture Space and can be used to reproduce viable shapes from within the model. However, in computer animation this is insufficient. For animation purposes, the ability to model the trajectory through shape space is also required, allowing the motion to be reproduced.



(a) The Running Woman Data Set, (b) The Walking Woman Data Set

Figure 7 - Dimensionally Reduced Data sets with the Cluster Based Constraints

3.3: Learning Temporal Constraints

Thus far the techniques have been used to learn the shape and size of the trajectory space, temporal analysis must now be performed to estimate how the model moves through space with respect to time.

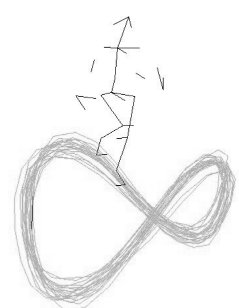


Figure 8 - Trajectory through Reduced Shape Space

Figure 8 shows the 3D trajectory of the reduced dimensional running data set projected down into 3 dimensions. Using simple animation techniques it is possible to watch the model move throughout the space as the animation sequence iterates. It is apparent that the motion is cyclic and consistent in nature and repeats in accordance with the period of the stride of the actor. Therefore, given any point within the space it is possible to predict where the model will move to next, based upon this observed motion.

The model has been estimated in a lower dimensional space; if the trajectory can also be modeled in this lower space then it is likely that paths of motion throughout the space could be determined and reconstructed. The deformation constraints have already broken the shape space down into linear patches with the center of the clusters being the mean shape of the transition at that point in time. It is also known that due to the cyclic nature of the data set, the pattern of movement repeats at regular intervals for fixed speeds of motion. Although this is not a necessary condition, it can effectively be modeled as a self-starting, finite state machine.

The reduced training set can therefore be used to analyse the model and probabilistically learn the transition of the model between clusters. This can be done with a state transition matrix of conditional probabilities otherwise known as a Markov chain.

3.4: Modeling Temporal Constrains as a Markov Chain

A Markovian assumption presumes that the present state of a system (S_t) can always be predicted given the previous n states (S_{t-1}, S_{t-2}, ..., S_{t-n}). A Markov process is a process which moves from state to state dependent only on the previous n states. The process is called an order n model where n is the number of states affecting the choice of the next state. The simplest Markov process is a first order process, where the choice of state is made purely upon the basis of the previous state. This likelihood of one state following another can be expressed as a conditional probability $P(S_t|S_{t-1})$.

A Markov analysis looks at a sequence of events, and analyses the tendency of one event to follow another. Using this analysis, a new sequence of random but related events can be produced which looks similar to the original.

The probability mass function $P\left(C_j^{t_n}\right)$ denotes the unconditional probability of being in cluster j at time t_n, or being in state j after n transitions (time steps). A special situation exists for $n=0$ where $P\left(C_j^{0}\right)$ denotes the probability of starting in state j. However, due to the assumption that the motion is cyclic and the trajectory file starts and ends mid cycle, no information is available for these initial probabilities.

The conditional probability mass function is therefore defined as

$$P\left(C_j^{t_n} \middle| C_k^{t_m}\right)$$

$P\left(C_j^{t_n} \middle| C_k^{t_m}\right)$ gives the probability of being in cluster j at time t_n conditional on being in cluster k at time t_m. In the trajectory file example it is fair to make the assumption that the next state of the model can be determined from the previous state. This can be confirmed by observing the trajectory taken through shape space by the training set (see Figure 8). Provided stationary elements of the chain are ignored, i.e. where $P\left(C_j^{t} \middle| C_j^{t-1}\right) \geq \max_k \left(P\left(C_j^{t} \middle| C_k^{t-1}\right)\right)$ and therefore choosing the 2nd highest probability move at each time step, the continuous transition through shape space can be achieved. If this assumption is made, then the process becomes a first order *Markov process* or *Markov Chain* and $p_{j,k}$ a one step transition probability

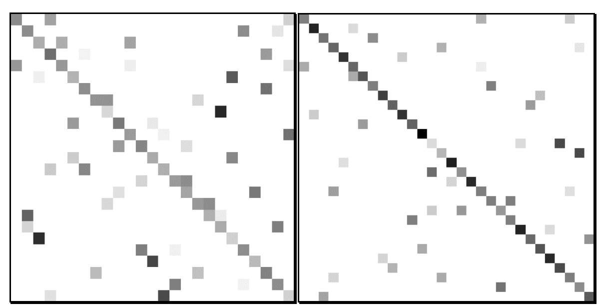
$$p_{j,k} = P\left(C_j^{t} \middle| C_k^{t-1}\right)$$

If there are n clusters in the model, then there are n states in the chain, hence a state transition matrix is an $n_x n$ matrix of one step transition probabilities. This is a discrete probability density function (*p.d.f*).

$$P\left(C_j^{t} \middle| C_k^{t-1}\right) = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,k} \\ p_{2,1} & p_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ p_{j,1} & \cdots & \cdots & p_{j,k} \end{pmatrix},$$

where $p_{j,k} \geq 0$ for all j,k, and $\sum_k p_{j,k} = 1$ for all j.

After construction of the PDF its content can be visualised by converting the matrix to a grey-scale image. Figure 9 shows the resulting images for both the running and walking data sets. It can clearly be seen that high probabilities exist along the diagonal of the image. This diagonal, when $i=j$ or $S_t=S_{t-1}$, demonstrates that the model always has a high probability that it will stay within the same local patch. This can be attributed to the discrete nature of the model, and the fact that each patch is constructed to model local deformation. The darker diagonal in the walking model shows that this model has a higher probability of remaining within a local patch and is a result of the speed of movement. As both sequences were captured at the same rate, the slower movement of the walking model generates more frames in each local patch and hence a lower probability that the model will make a transition to another patch. However, as the numerical identity of each local patch within the matrix is randomly generated by the k-means algorithm, no further conclusions can be drawn from the patterns within the image, hence the random distribution.



(a) The Running Woman Data Set, (b) The Walking Woman Data Set

Figure 9 - Discrete Probability Density Functions

The PDF's shown in Figure 9 provide a conditional probability, that given a cluster at time t, the system will move to another cluster at the next time step. By taking the highest probability move at each time step the highest probability path can be modeled throughout the space.

Using this information and the mean shape of each cluster as key frames, the motion of the training set can be reconstructed. If any cluster of the model is chosen at random and the next highest probabilistic transition made at each time step $\mathbf{argmax}_i(p_{i,j})$ where $i \neq j$, the model should settle within a natural path through the space. This is similar to a finite state machine that has a circular path and is self starting. If the natural number of clusters selected is correct then the cyclic period of the model should be equal to that of the training set. If the cluster number is too high then non-equidistant cluster centers result and the model appears to 'jurk'. If the cluster number is greater than twice the natural number then the model risks having a cyclic period of multiples of that of the true motion.

Figure 10 shows the highest probability path for the running model that consists of 15 clusters. Each pose of the model is the mean shape (exemplar) of a cluster. This model is reconstructed from the information that has been learnt from the motion file and accurately reproduces the original motion. The animation can be further refined by linearly interpolating between these key frames (exemplars), as the linear interpolant along a line between exemplars is equivalent to linearly interpolating all points on the model between key frames. This does however introduce slight non-linear deformities. These deformities can be reduced by projecting the interpolated model into the constrained space to extract the closest allowable model for rendering.

Figure 11 shows the highest probability path through the walking model, consisting of 19 key frames that produce a cyclic path of high probability through the Markov chain. The original model contained 30 clusters and the redundant 11 clusters partly model the introductory gait acceleration, which can be seen in Figure 12. The red line shows this high probability path extracted from the Markov chain. Acting like a self-starting finite state machine, if the model is initiated within the low probability startup area of the space, the chain quickly moves the model to the circular region, where constant cyclic movement occurs.

As these key frames exist in a far lower dimensional space than the original model, polynomials can easily be fitted to smoothly parameterise the nonlinear trajectory through space. The advantages of this reduction in complexity become apparent when more complex models such as polygonal surface approximations are considered. However, in addition to this mean trajectory through shape space, the model also represents statistically how the models can deform from this mean trajectory. By constructing a larger training set which incorporates more variation in subject movement a generic statistical model of specific human motions can be created.

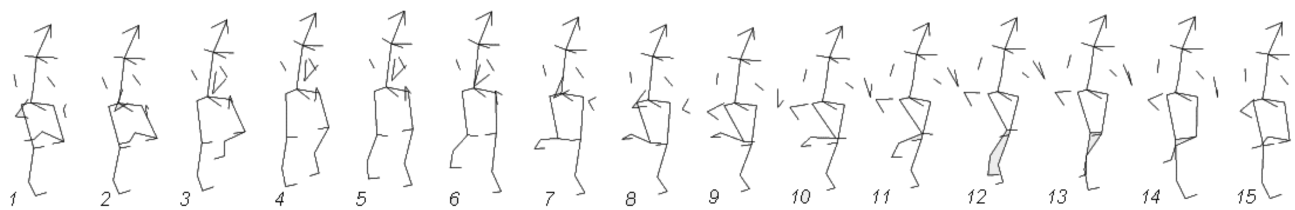


Figure 10 - Extracted Trajectory for Running Model

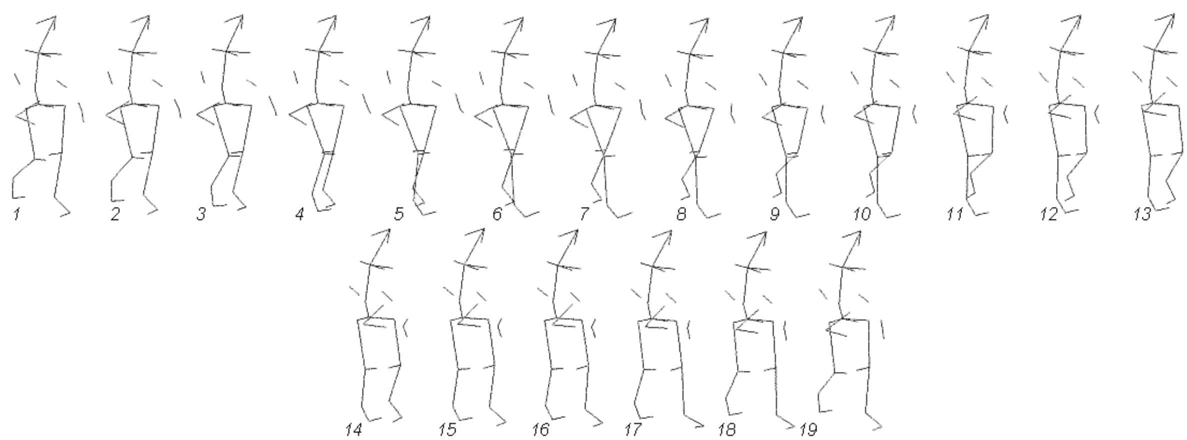


Figure 11 - Extracted Trajectory for Walking Model

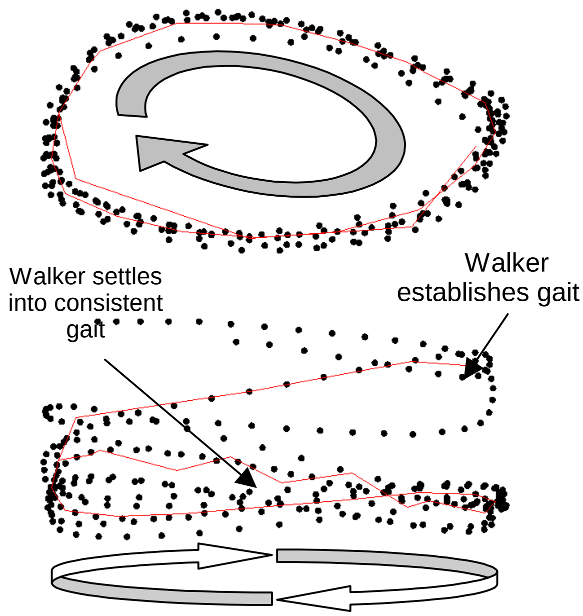


Figure 12 - High Probability Path through Walking Model Shape Space

4: Conclusions

This paper has been shown how the reduced dimensionality and discrete representation of the Constrained Shape Space approach to modeling non-linear data sets can be used to provide simple analysis and reconstruction of motion. This is done by analysing the training set and constructing a Markov Chain, which is a discrete, probabilistic representation of the movement of the model through shape space. It has also been shown how, using this learnt temporal information, animated models can be produced which encapsulate the temporal information learnt from a training set.

It has been shown how by reducing the dimensionality of a training set through linear PCA the non-linear trajectory of a human motion sequence can be represented as a number of key frames. However, these key frames are exemplars within a statistical model which depict the extent to which a model can deviate from this mean trajectory. As the dimensionality of this trajectory is low these key frames can be used to reproduce the motion learn from a training set. Furthermore, these key points can be used to calculate polynomial trajectories which smoothly estimate movement of all elements of a model in a lower dimensional space. As this is the case these techniques can be applied to more complex graphical models such as 3D surface meshes.

Some authors may argue that this model of dynamics constitutes a Hidden Markow Model (HMM) where clusters are hidden states and the transition between clusters are the HMM state transitions. However, ignoring the implied gaussian distribution of each local patch/cluster the transformation between shape space (state space) and model pose is a simple linear mapping and hence not necessarily hidden. For a true extension of this technique to that of a HMM the interested reader is directed to [5] where the use of an additional layer of conditional probability allows the mapping for successful gesture recognition within a CONDENSATION [11] like tracker. Here the shape space truly becomes the hidden layer and the gesture space the observation layer of the model.

5: References

[1] Baumberg, A. and Hogg, D., An Adaptive Eigenshape Model. In: Pycock D, ed. British Machine Vision Conference 1995, BMVC'95. University of Birmingham, Birmingham, UK: BMVA, 1995:87-96.

[2] Bowden, R. Learning non-linear Models of Shape and Motion , PhD Thesis, Dept Systems Engineering, Brunel University, Uxbridge, Middlesex, UB8 3PH, UK. December 99.

[3] Bowden, R., Mitchell, T. A., Sahardi, M., Non-linear Statistical Models for the 3D Reconstruction of Human Pose and Motion from Monocular Image Sequences. Image and Vision Computing, 18(9), pp729-737, 2000.

[4] Bowden R, Mitchell T A and Sahardi M, Cluster Based non-linear Principal Component Analysis, IEE Electronics Letters, 33(22) (1997) 1858-1859.

[5] Bowden R, Sarhadi M, Building Temoral Models for Gesture Recognition, Submitted to BMVC2000.

[6] Bregler C and Omohundro S, Surface Learning with Applications to Lip Reading, Advances in Neural Information Processing Systems 6, San Francisco (1994).

[7] Cootes T F and Taylor C J, A Mixture Model for Representing Shape Variation, in: Clark A F, ed., British Machine Vision Conference 1997, (BMVA, Essex, UK, 1997) 110-119.

[8] Cootes T F, Taylor C J, Cooper D H and Graham J. Active Shape Models - Their Training and Application. Computer Vision and Image Understanding, 61(1) (1995) 38-59.

[9] Heap T and Hogg D C, Automated Pivot Location for the Cartesian-Polar Hybrid Point Distribution Model, in: Pycock D, ed., British Machine Vision Conference 1995, (BMVA, Birmingham, UK, 1997) 97-106.

[10] Heap T and Hogg D C, Improving Specificity in PDMs using a Hierarchical Approach, in: Clark A F, ed. British Machine Vision Conference 1997, (BMVA, Essex, UK, 1997) 80-89.

[11] Isard M, Blake A, CONDENSATION - Conditional Density Propagation for Visual Tracking. International Journal of Computer Vision, 29(1), pp5-28, 1998.

[12] Sozou P D, Cootes T F, Taylor C J and Di-Mauro E C. A Non-linear Generalisation of PDMs using Polynomial Regression, in: Hancock E, ed. British Machine Vision Conference 1994, (BMVA, York, 1994) 397-406.

[13] Sozou P D, Cootes T F, Taylor C J and Di Mauro E C. Non-Linear Point Distribution Modelling using a Multi-Layer Perceptron, in: Pycock D, ed. British Machine Vision Conference 1995, (BMVA, Birmingham, UK, 1995) 107-116.

[14] Kotcheff A C W and Taylor C J, Automatic Construction of Eigenshape Models by Gentic Algorithms, Lecturenotes in Computer Science. 1230 (Springer Verlag 1997) 1-14.