Estimation of dense, non-rigid motion fields from a multi-camera array using a hierarchical mixture model.

Adam Bowen, Andrew Mullins, Roland Wilson, and Nasir Rajpoot*

Department of Computer Science
University of Warwick, UK
rgw@dcs.warwick.ac.uk

Abstract. The problem of modelling objects of arbitrary complecity for video based rendering has been much studied in recent years, with the growing interest in 'free viewpoint' video and similar applications. Typical approaches fall into two categories: those which approximate surfaces from dense depth maps obtained by generalisations of stereopsis methods and those which employ an explicit geometric representation such as a mesh. While the former has generality with respect to geometry, it is inevitably limited in terms of viewpoint; the latter, on the other hand, sacrifices generality of object geometry for freedom to pick an arbitary viewpoint. The purpose of the work reported here is to bridge this gap in object representation, by employing a surface element model, but one which is freed from the restrictions of a mesh. Estimation of the model and tracking it through time from multiple cameras is achieved by novel multiresolution stochastic simulation methods. After a brief outline of the method, its use in modelling human motions using data from the Warwick multicamera studio is presented to illustrate its effectiveness compared to the current state of the art.

1 Introduction

The problem of modelling and tracking 3D objects from one or more video sequences has received considerable attention over recent years, as interest in 'free viewpoint' video has grown and computational and capture costs have fallen. Techniques range from straightforward adaptation of stereo vision algorithms, eg. [2] to meshes and visual hulls [1]. Although the former, requiring no explicit model of the object to be tracked, is general in that respect, it suffers from the restriction on viewing point and the positioning of cameras. On the other hand, techniques based on meshes inevitably make assumptions about the shapes of the objects and may constrain the motions. The problem is that, without such constraints, a mesh or voxel representation simply has too many degrees of freedom to represent any plausible motion. What is needed is a way of constraining the motion to conform to the sorts of movements that, for example, dancers or similarly complex figures might undergo, while retaining the flexibility to represent arbitrary objects and their movements. This naturally suggests a hierarchical model and indeed there have been attempts to use such structures in motion estimation, notably [1] and [3], in which the motion of the trunk is used to constrain movements of limbs.

In this paper, we present a generalisation of these ideas, which is capable of representing objects of arbitrarily complex shapes undergoing smooth motions. It is capable of dealing with articulated objects, but shows the potential to cope with more complex scenes, involving multiple objects. In order to achieve this goal, it is necessary to construct a model which is flexible, yet stable enough to track coherent motions over time. The method we present is based on a general approach to the statistical modelling of data, Gaussian mixtures, which can be combined with a clustering process to group elements on the basis of their visual properties, such as colour, texture and motion, to form a hierarchy of mixture models representing the objects in the scene. This in turn serves to

* This work was supported by the UK EPSRC.

reduce the complexity of motion estimation and to constrain the degrees of freedom, without being tied to a particular model, such as a mesh or voxel representation. It represents an extension of the work reported in [7] to deal with moving 3D data. The approach has been tested on a number of sequences, both publicly available ones, such as the Microsoft Research data [9], and some captured using the Warwick camera array, a studio with 48 fire-wire digital video cameras. Further details of the set-up are available at [11].

After a brief introduction to the theory behind the mixture model and an outline of the algorithms for constructing it and tracking it, we present a series of results to show its efficacy in a number of sequences. Although the results presented here are based on a locally planar surface representation, using patch estimates derived using a particle filter from sets of neighbouring cameras, which we have used throughout our work on free viewpoint video (eg. [5],[4]), the mixture model is capable of being used with arbitrary input data - from point clouds to voxels, with appropriate attributes. Moreover, the resulting motion estimates define a smooth motion field, which may be used to animate arbitrary objects [5]. These are features shared by no other representation of which we are aware. The paper is concluded with a discussion of the advantages and limitations of the model, as well as some possible extensions. Full details of the methods can be found in [5], [6].

2 Construction and Tracking of Gaussian Mixtures

In previous work, it was shown that a Gaussian mixture can be constructed in a top-down fashion, starting with a single component and splitting components when their fit to the data was poor, as judged by either a Bayesian or other criterion, such as AIC [7]. As noted in that paper, the Gaussians have the considerable advantage of forming a closed set under affine transformations, making the motion computations simpler. In the work reported here, a bottom-up process was used, to allow greater control over the number of levels in the tree and number of components/level, which were important in the formulation of the motion estimator. We regard the feature vectors as a set, $\{\boldsymbol{f}_i \in R^n, 1 \leq i \leq N\}$, of points in R^n. Normally, $\boldsymbol{f}_i$ would represent the position, orientation, colour and possibly motion of a 3D surface or volume element, giving up to 9 dimensions, but this is a matter of utility in a particular application: surface texture might also be included. The aim is to represent the data density by a normal mixture with, say, M_l components at level l, where $0 \leq l \leq L$ and $M_l < M_{l-1}$. To ensure a tree structure, components at level l, rather than data, are used as input to the next level, $l+1$, of the tree. Correspondingly, the covariance of the data assigned to any component is carried forward in the clustering at the next level of the tree.

A Bayesian approach to clustering is used, based on the conjugate priors for the Gaussian mixture problem, namely a Dirichlet prior on the mixture weights, a normal on the means and a Wishart density on the covariances [8]. This has the considerable advantage of allowing a Gibbs sampler to be used and, in this application, is not overly restrictive. The complete algorithm is as follows [8]:

1. Simulate from the full conditionals:

$$\pi \sim D(\alpha^*) \tag{1}$$

$$\Sigma_i^{-1} \sim W(W_i^*, r_i^*) \tag{2}$$

$$\boldsymbol{\mu}_i \sim N(\mathbf{y}_i^*, \tau_i^* \Sigma_i) \tag{3}$$

where

$$\alpha^* = \alpha + n \tag{4}$$

$$r_i^* = r + n_i \tag{5}$$

$$\tau_i^* = \frac{\tau}{\tau n_i + 1} \tag{6}$$

$$\mathbf{y}_i^* = \frac{n_i \bar{\mathbf{x}}_i + 1/\tau \mathbf{y}_i}{n_i + 1/\tau} \tag{7}$$

$$W_i^* = \left(W_i^{-1} + n_i \boldsymbol{S}_i + \frac{n_i}{\tau n_i + 1} \left(\bar{\mathbf{x}}_i - \mathbf{y}_i \right) \left(\bar{\mathbf{x}}_i - \mathbf{y}_i \right)^T \right)^{-1} \tag{8}$$

$\bar{\mathbf{x}}_i$ is the sample mean of the i^{th} component, $\boldsymbol{S}_i$ the sample covariance and n_i is the number of data points assigned to it. W_i, α, r, τ_i and $\mathbf{y}_i$ are parameters on the prior distributions for each component. $D(\alpha)$ is a Dirichlet distribution with parameter α and $W(V, n)$ is a Wishart distribution with parameters V and n.

2. Derive the cluster assignments:
   Assign the class label of the i^{th} data point, $\boldsymbol{x}_j$, randomly to one of the components $k \in 1..K$ proportionally to $N(\mathbf{x}_j - \boldsymbol{\mu}_k, \Sigma_k)$ and update the sample means $\bar{\boldsymbol{x}}_k$ and covariances $\boldsymbol{S}_k$ according to

$$\bar{\boldsymbol{x}}_k = \frac{1}{n_k} \sum_{i \in \{j | z_j = k\}} \boldsymbol{x}_i \tag{9}$$

$$\boldsymbol{S}_k = \frac{1}{n_k} \sum_{i \in \{j | z_j = k\}} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)^T \tag{10}$$

Note that, for levels $l > 0$, the covariances of the components must be included in both the covariance estimate (10) and the classifications. This process is iterated until convergence is achieved at a given level, l. Data are then assigned to cluster i on level l using a MAP rule and the clusters on level l, characterised by their estimated means and covariances, are used as data in the process at level $l + 1$.

Once the MGM model has been constructed, it can be tracked through time in various ways, including extended Kalman filtering or by particle filtering, the latter being the method adopted in the experiments [5]. In both cases, the key idea is to use a sequential, Bayesian approach, in which the estimate for level l can be used as a prior for level $l - 1$. Because of the hierarchical nature of the representation, it is sufficient to employ a rigid motion model for each component, requiring estimation of a 6-parameter vector $\boldsymbol{\theta}$ per component and maximising the posterior

$$P(\boldsymbol{\theta}_{l,i} | \boldsymbol{Y}_j, j \in \Lambda_{l,i}) \propto \int_{\boldsymbol{\theta}_{l+1,p(i)}} d\boldsymbol{\theta} \prod_{j \in \Lambda_{l,i}} P(\boldsymbol{Y}_j | \boldsymbol{\theta}_{l,i} \boldsymbol{\theta}) P(\boldsymbol{\theta}_{l,i} | \boldsymbol{\theta}) \tag{11}$$

where $p(i)$ is the parent of component l, i at level $l+1$ and $\boldsymbol{Y}_k$ are the data, which in the experiments consisted of the average colour in each of a set of S patches selected at random from the set associated with component l, i. Elimination of the 'nuisance' variable $\boldsymbol{\theta}_{l+1,p(i)}$ is impractical, but can be avoided by using the MAP estimate $\hat{\boldsymbol{\theta}}_{l+1,p(i)}$, with no obvious degradation in the quality of the estimates. At the top level of the MGM tree, the prior is derived instead from the estimate at the previous time. The particle filter employs a Gaussian approximation to the posterior and used the prior as the importance density, greatly simplifying the computations [10]. A summary of the main steps in the algorithm is:

For levels $L > l \geq 0$,

1. Sample from importance density, $P(\boldsymbol{\theta}_{l,i} | \hat{\boldsymbol{\theta}}_{l-1,p(i)})$, to get weights $w_{l,i}, 1 \leq i \leq M_l$.

2. Compute likelihoods at current estimate $\theta_{l,i}$, $P(Y_j|\boldsymbol{\theta}_{l,i})$.
3. Use weighted likelihoods to estimate posterior mean and covariance.

To understand how the motion field is constructed from the Gaussian components, note that at the finest level, we have a motion for each component $0, i$, consisting of two elements: a translation vector $\boldsymbol{t}_{0,i}$ and a rotation matrix $\boldsymbol{R}_{0,i}$. Now, given an arbitary point in 3D, $\boldsymbol{x}$, say, there are corresponding probabilities $P(i|\boldsymbol{x})$ that it belongs to the ith component,

$$P(i|\boldsymbol{x}) = w_i N(\boldsymbol{x} - \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) / \sum_j w_j N(\boldsymbol{x} - \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{12}$$

where $N(.,.)$ is the 3D normal density with parameters $\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$ for the ith component and w_i is a weight representing the population of the ith component. Where the MGM model uses additional data, such as colour or velocity, the projection onto the three spatial dimensions remains Gaussian. Now the translation at $\boldsymbol{x}$ is simply the conditional mean

$$\boldsymbol{t}(\boldsymbol{x}) = \sum_i P(i|\boldsymbol{x}) \boldsymbol{t}_i \tag{13}$$

and the rotation may be similarly interpolated, in common axes, using the exponential form

$$\boldsymbol{R}(\boldsymbol{x}) = \prod_i \boldsymbol{R}_i^{P(i|\boldsymbol{x})} \tag{14}$$

These are used to update the positions of the individual data elements from the estimated motion at the highest resolution in the MGM tree.

3 Experiments

Experiments were carried out on a number of sequences captured in the Warwick studio, as well as two from the Microsoft Research site. All were captured at 1024×768 pixel resolution, the Warwick ones at 30 fps. Unlike the Microsoft camera arrangement, which has eight cameras arrayed in a line, with a maximum disparity of 150 pixels, the Warwick cameras are arranged to provide full coverage of the room in a non-uniform way: 32 cameras are arranged in a recatngular array on one wall and the remainder in the corners and mid-points of each of the remaining three walls, with one in the ceiling space. The maximum overlap between cameras is only 50%, resulting in disparities of up to 400 pixels. Consequently, accurate modelling and rendering is a far more challenging problem for the Warwick data than for the Microsoft array.

The MGM models were built with either two or three levels, with respectively $M_0 = 50,\ M_1 = 5$ and $M_0 = 250,\ M_1 = 50,\ M_2 = 5$ components, an arrangement that was chosen empirically. It was found that the sampler converged after 200 iterations and this was chosen for the estimation. The prior weights were set to $\alpha = M_{l-1}/M_l, r = 8, \tau = 1$ and W_i was diagonal, with variances based on the spatial and colour variances, after extensive experimentation [6]. The MGM approach was compared to publicly available K-means and hierarchical clustering algorithms, both in terms of model fit, measured by squared error and computation times on different data sets. The results are summarised in Tables 1 and 2, which show that MGM significantly outperforms both the other methods in terms of error and is faster, surprisingly, than this implementation of K-means. The appearance of the induced clusters is illustrated in figure 3, which shows the cluster centres in colour for two sequences.

Motion estimation was only practicable for the Warwick sequences: the Microsoft data do not provide sufficient coverage to obtain reliable estimates, The multiresolution Gaussian particle filter used 10000 particles and the likelihoods were calculated using a set of at most 2000 patches for each

component. The MGM model and motion estimator were reinitialised every 25-30 frames, to prevent error accumulation. These choices were arrived at after experimentation and were found to give good results across the whole set of data. To validate the algorithms, a short 'ground truth' sequence was obtained from the sequence 'Kate', by manually tracking her index finger. The average error over 30 frames was of the order of 15mm, comparable to the best results reported to date, as can be seen from figure 1. To illustrate the estimator's ability to produce a dense motion field, we used it to animate a manually initialised skeleton, as shown in figure 3. Despite having no constraints on joint angles, the results were quite compelling. Finally, views synthesised for the 'Kate' and 'Breakdance' sequences are shown in figures 4 and 5. The reconstructions contain a few rendering artefacts, which result in a typical signal-noise ratio of the order of 25-30dB, worse than the figures obtained by warping [2]; this is the price of the much less restrictive algorithms when applied to relatively low disparity data. Figure 2 shows the reconstruction peak-rms SNR for both sequences, based on a 'leave-one-out' test, in which a known view is reconstructed from the remaining data. Full details of the rendering algorithm are given in [6]. Overall, the results demonstrate the usefulness of a fully Bayesian approach within a hierarchical model framework.

4 Conclusions

This paper has provided an introduction to a general model of motion, using a hierarchical model of the data and the motion estimators derived from that model, for use in multi-camera video. It has the attraction of being flexible, while providing a set of constraints on the motion via the use of a hierarchical Bayesian motion estimator. This not only prevents the problem from becoming ill-posed, it also significantly speeds up computation. The resulting motion estimates can be interpolated in an obvious way using the mixture model to animate arbitrary objects with motions derived from a given sequence. Although far from real time in terms of computation, it has been shown that both the model initialisation phase and the particle filtering converge relatively quickly because of the strong priors, so that typically a few hundred iterations are sufficient to obtain satisfactory results. Moreover, the methods are designed to allow real time rendering of novel views from the representation and this goal has been achieved. In summary, while particular applications will always benefit from imposing tighter constraints than the model presented here, our techniques seem sufficiently promising to warrant further investigation.

	K-Means	Hierarchical Clustering	MGMM
'Kate' sequence	327	394	131
'Hamutal' sequence	179	200	119
'Breakdancers' sequence	132	124	73
'Ballet' sequence	117	113	66

Table 1. Energy of the final clustering in thousands; lower values indicate a bet ter fit.

	K-Means	Hierarchical Clustering	MGMM
'Kate' (34934 patches)	13m 09s	130m 39s	1m 46s
'Hamutal' (17022 patches)	2m 57s	13m 22s	56s
'Breakdancers' (9617 patches)	2m 01s	5m 40s	24s
'Ballet' (3210 patches)	30s	14s	20s

Table 2. Clustering time using the three algorithms on several data sets. The tim es were computed on an Intel Xeon 3.2GHz, the K-Means clustering was 1 level only and the MGMM was run for 200 iterations.

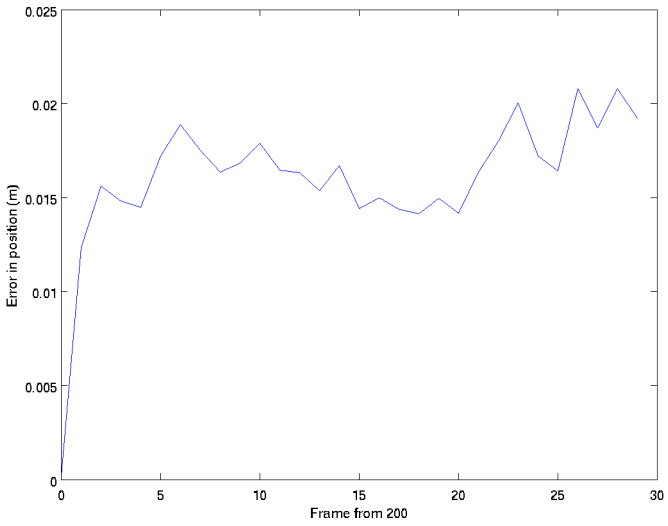

Fig. 1. Plot of the error in position vs frame number for the tracked index finger in 'Kate'

References

1. K. M. Cheung, S. Baker and T. Kanade: Shape from silhouette across time part I: theory and algorithms. Int. J. Comput. Vision, 62:221–247 (2005).
2. C. L. Zitnick, S. B. Kang and M. Uyttendaele: High-quality video view interpolation using a layered representation. ACM Trans. Graphics, 23:600–608 (2004).
3. J. Mitchelson and A. Hilton: Hierarchical tracking of multiple people. Proc. BMVC-03, Norwich, (2003).
4. A. Bowen, A. Mullins, R. Wilson and N. Rajpoot: Video based Rendering Using Surface Patches. Proc. IEEE 3DTV Conf., Kos, (2007).
5. A. Mullins: Stochastic Geometry Estimation for Video based Rendering. PhD Thesis, University of Warwick, (2008).
6. A. Bowen: Video based Rendering and Coding using a Planar Patch Model. PhD Thesis, University of Warwick, (2008).
7. X. Mo and R. Wilson: Video modelling and segmentation using Gaussian mixture models. Proc. ICPR-04, Cambridge, (2004).
8. G. McLachlan and D. Peel: Finite Mixture Models. Wiley, New York (2000).
9. http://research.microsoft.com/vision/InteractiveVisualMediaGroup/
10. J. M. Kotecha and P. M. Djuric: Gaussian particle filtering. IEEE Trans. Sig. Proc., 51:2592–2601 (2003).
11. http:www.dcs.warwick.ac.uk/ fade/

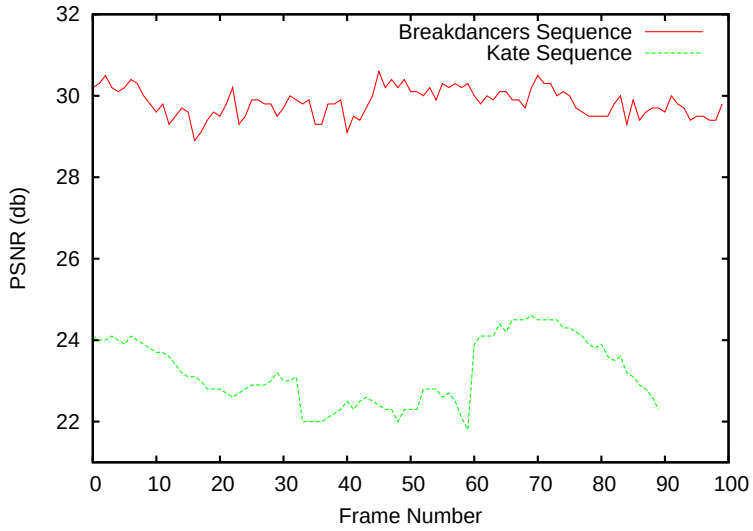

Fig. 2. Plot of the PSNR vs. frame number for the synthesised views.

(a) Skeleton at time 0 (b) Skeleton at time 80

Fig. 3. Skeleton, animated using motion field from 'Kate'.

(a) Coloured centroids for the breakdancers sequence

(b) Coloured centroids for the 'Kate' sequence

(c) The clustering of 'breakdance' at t=0.

(d) The clustering of 'Kate' at t=0.

Fig. 4. Clustering the 'Breakdancers' and 'Kate' data sets using uniform and seeded priors.

(a) Frame 0 (b) Frame 15

(c) Frame 30 (d) Frame 45

(e) Frame 60 (f) Frame 75

Fig. 5. Synthesised views of the 'Kate' sequence at different time instants.

(a) Frame 15 (b) Frame 30

(c) Frame 45 (d) Frame 60

(e) Frame 75 (f) Frame 90

Fig. 6. Synthesised views of the 'Ballet' sequence at different time instants.