Summarised Hierarchical Markov Models for Speed-Invariant Action Matching

J.Kilner, J-Y.Guillemaut, A.Hilton
Center for Vision, Speech and Signal Processing
University of Surrey, Guildford, United Kingdom
{J.Kilner,J.Guillemaut,A.Hilton}@surrey.ac.uk

Abstract

Action matching, where a recorded sequence is matched against, and synchronised with, a suitable proxy from a library of animations, is a technique for generating a synthetic representation of a recorded human activity. This proxy can then be used to represent the action in a virtual environment or as a prior on further processing of the sequence. In this paper we present a novel technique for performing action matching in outdoor sports environments. Outdoor sports broadcasts are typically multi-camera environments and as such reconstruction techniques can be applied to the footage to generate a 3D model of the scene. However due to poor calibration and matting this reconstruction is of a very low quality. Our technique matches the 3D reconstruction sequence against a predefined library of actions to select an appropriate high quality synthetic representation. A hierarchical Markov model combined with 3D summarisation of the data allows a large number of different actions to be matched successfully to the sequence in a rate-invariant manner without prior segmentation of the sequence into discrete units. The technique is applied to data captured at rugby and soccer games.

1. Introduction

Sports media production has begun to make use of multi-camera techniques in the production environment. Examples include the use of free-viewpoint video to render novel viewpoints, such as EyeVision and LiberoVision [6, 18], and analysis of some part of the action in a virtual environment such as with HawkEye[12].

Due to the unconstrained nature of the capture environment most multi-camera techniques, which are developed in the constrained environment of a special-purpose studio[14], struggle to achieve good results. Errors in calibration and matting alongside other domain-specific challenges such as moving cameras, unconstrained illumination, multi-body occlusion, rapid motion, zooming and low resolution images can degrade existing techniques to the point of complete failure.

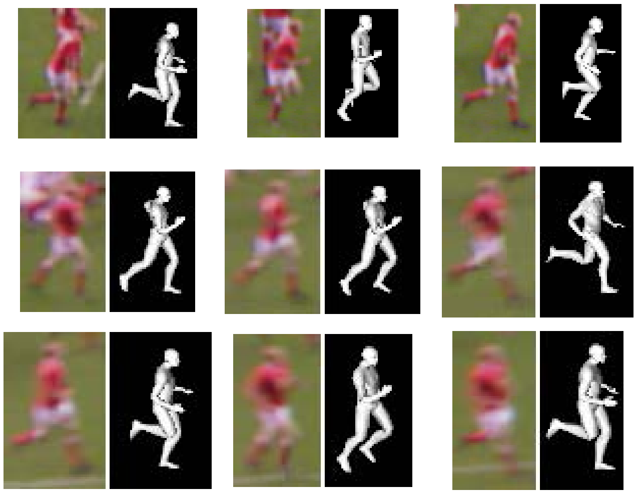

Figure 1. Original images from a sequence and matched poses.

In order to address these various issues this paper attempts to leverage the fact that the recorded data consists of human motion by matching the recorded shapes against a library of human actions to generate a synthetic approximation of the original sequence (as shown in Figure 1). This synthetic approximation can then either be used directly as a proxy for the recorded data, to produce a virtual representation such as VirtualReplay[1], or as a semantic prior on further processing (for example helping to constrain the final reconstruction to contain two legs where one has been truncated by calibration errors). This technique of action matching is thus a combination of action recognition, pose alignment and synchronisation.

By working in the 3D domain, many problems such as occlusions and blurring that would make any individual camera unsuitable for use can often be overcome, and a single exemplar of any motion can be used (as 3D matching is invariant to the viewpoints of the cameras).

Prior to matching, animations are summarised to pick out the significant poses in the sequence which can then be compared in a rate-independent manner. The summarised sequences are then matched to a library of prerecorded actions using a hierarchical Markov model.

Section 2 presents the background to this work. Section 3 describes the pre-processing required to generate shape-from-silhouette data and to track the players. The shape matching framework and the proposed action matching technique are then presented in Section 4. Section 5 contains the results obtained from real footage captured at an outdoor sporting event, and the paper concludes with a summary and further avenues of investigation in Section 6.

2. Background

Action synthesis is typically achieved in a custom-built multi-camera studio by means of motion capture, usually using commercial marker-based systems or more recently using pose-recognition based markerless motion capture systems[21]. Action synthesis and pose recognition in unconstrained crowded scenes such as video of outdoor team sports remains an open and challenging problem.

2.1. Pose Recognition

Recent work by Ferrari *et al*. used progressive search space reduction to estimate body pose in TV and film data[7], Dimitrijevic *et al*. used Bayesian templates to recognise walking poses in natural scenes[4], and Gammeter *et al*. used a statistical model of human pose to refine a pedestrian tracking system[9]. For sports applications, Lu and Little used Histograms of Orientated Gradients to perform action recognition on low resolution soccer and hockey video[20], Efros *et al*. recognised low resolution video sequences of soccer players using optic flow[5] and Weinland *et al*. used a volumetric exemplar representation to perform camera-pose invariant pose recognition with matching performed in 2D[28].

2.2. 3D in Sports Production

Since Kanade *et al*. developed EyeVision for use at the SuperBowl[6] there has been considerable interest in application of 3D computer vision techniques to the field of sports production. Innamoto *et al*. demonstrated a system for 3D playback of a recorded soccer game [13], Connor and Reid demonstrated 3D reconstruction of a soccer game from multiple cameras [3] and Loy *et al*. reconstructed 3D motion from monocular video using pose templates[19]. Multi-camera systems were used in recent work by Guillemaut *et al*. where a Graph-Cut optimisation generated high-quality 3D scene reconstructions and matting refinement[11].

While progress is being made in this field most techniques either require specialist equipment, manual intervention or rely on assumptions about camera pose or player motion. A general solution has yet to be found. Existing systems used in sports production rely on the time-consuming and expensive manual placement of avatars into a virtual environment to achieve motion synthesis.

2.3. Hierarchical Markov Models

The Hidden Markov Model (HMM) has been extended in many ways to allow it to deal with large numbers of states and to exploit high level relationships between states. A typical application is gesture recognition where child models will model individual actions and parent models will model the action sequence. Layered Hidden Markov Models[22], stochastic context-free grammars [24], and Hierarchical HMMs[8] have all been proposed as ways of structuring Markov models.

Layered structures of temporal processes typically have to deal with the problem of how to manage re-initialisation of the child models. The parent model decides which child model to activate at any given time based on the responses of the children to the data. If the child model is re-initialised at every time step, it cannot correctly model long-term temporal relationships and will become dominated by noise. However if it is not re-initialised then recognition will become degraded by poor performance during those periods of the sequence where it is inactive.

This issue is typically solved by quantising or clustering events at the lowest time granularity and re-initialising each child model after the cluster of observations is processed, as in work by Oliver *et al*. [22]. Typically parent models then consume the relative likelihoods of each child as a new observation vector, working at a different temporal granularity. This introduces temporal quantisation artefacts as each level of models is limited by the temporal granularity at which it operates. An alternative approach is that employed by Hierarchical Hidden Markov Models introduced by Fine *et al*. [8] where control flows up and down the hierarchy with each level of the hierarchy yielding to a higher level as it reaches a production state. This type of model cannot model arbitrary transitions between looping behaviours as a child model can not be pre-empted by another child model - it must yield on reaching a production state.

This work uses a novel formulation similar to the Layered Markov Models used in the work of Oliver *et al*. [22]. Unlike previous work, our formulation allows all levels of the model to work at the lowest level of temporal granularity and does not limit action transitions to any specific states of the modeled behaviours.

3. Pre-Processing

This work attempts to solve the problem of action matching between a library of pre-generated synthetic animations and a sequence of images recorded at a sporting event. This process can be considered in two stages: pre-processing where recorded images of the match are converted to per-player 3D model sequences, and action matching where the per-player model sequences generated in the pre-processing stage are matched against a library of synthetic animations using a Summarised Hierarchical Markov Model. The following section describes the pre-processing stage in this pipeline.

Pre-processing begins with the capture of images during a sporting event. These are calibrated and segmented to provide an input to a shape-from-silhouette technique. A robust shape-from-silhouette technique produces a 3D scene representation per frame of the recorded video, generating a sequence of 3D models which describe the recorded event. The scene models are then divided into individual player models by a player tracking algorithm which generates a number of per-player 3D model sequences. The per-player sequences are then used as the input to the action matching stage.

3.1. Calibration and Matting

For a sports broadcast of soccer or rugby there are approximately 15 manually-operated cameras in the stadium. Of these, 6-8 cameras are typically following the action of interest and capture footage suitable for calibration. The other cameras include slow motion cameras, cameras tightly focused on a single player and cameras giving an overall view of the stadium or watching the crowd.

Video sequences are captured from multiple cameras arranged around a stadium. The cameras are calibrated by detecting pitch markings in the image and comparing them to a pre-generated model[27] as lack of access to the field and the cameras prohibits traditional calibration techniques. While this technique produces relatively accurate results, the resulting calibration contains significant errors of the order of 1-2 pixels.

The images captured during the match are segmented using a combination of difference-keying and chroma-keying[10]. An example segmentation is shown in Figure 2. This technique typically achieves a segmentation with 1-2 pixels of accuracy.

3.2. Robust Multi-View Reconstruction

The calibration data and silhouettes are used as input to a robust shape-from-silhouette technique to calculate the visual hull[17] of the foreground region of the scene. As the calibration and matting contain combined errors of up to 3 or 4 pixels (which is of similar magnitude to the representation of player limbs) the volume generated by applying a straightforward shape-from-silhouette technique will be severely truncated.

In the standard model of multi-view geometry, a camera is a mapping between the $\mathbb{R}^2$ image domain to $\mathbb{R}^3$. An image which has been segmented into foreground and background is a labelling on $\mathbb{R}^2$ defining a set of points f which consists of all the foreground pixels and the entirety of $\mathbb{R}^2$ that lies outside the image. Shape-from-silhouette techniques then involve mapping this labelling from $\mathbb{R}^2$ to $\mathbb{R}^3$ using the multi-camera geometry of the scene. This defines the set of points F in $\mathbb{R}^3$ which map to f in all cameras. The Visual Hull(VH) is then the boundary of F.

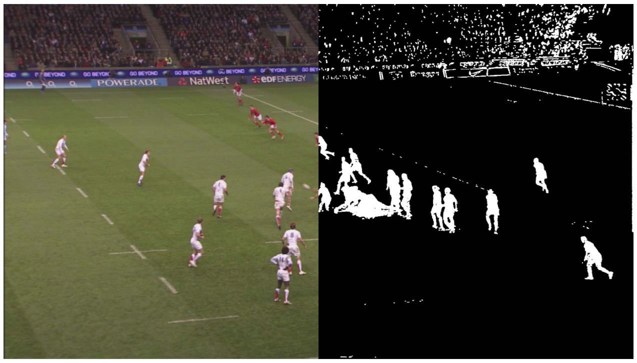

Figure 2. An example of an image and segmentation.

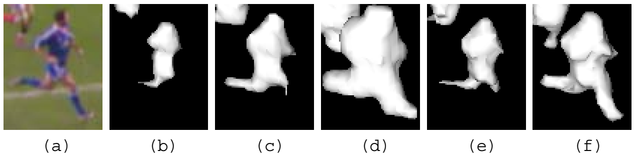

Figure 3. A comparison of various reconstruction techniques. a) the original image, b) VH (note extensive truncation), c) CVH with an error tolerance of 1 pixel (note right leg remains truncated) d) CVH with an error tolerance of 4 pixels (note truncation is eliminated but shape is severely distorted) e) CH at 90%, f) CCH at 90% with error tolerance of 1 pixel (note that truncation is eliminated while distortion is minimised)

In the presence of matting and calibration errors F will become underestimated resulting in a truncated visual hull. One way of reducing the error is to dilate the labelling f and so expand F as in the Conservative Visual Hull (CVH) technique introduced by Kilner *et al.* [15]. However, this can lead to a loss of detail in the recovered shape. A different approach is to relax the constraint that points in F must map to f in all cameras. Instead of the Visual Hull then a Consensus Hull(CH) can be taken where the CH is an iso-surface of labelling consensus. As the level of required consensus is reduced from 100% (the VH) the CH quickly generates a lot of phantom volumes and noise, but at $80\% - 90\%$ shape is retained while truncation is significantly reduced as shown in figure 3.

For this work a Conservative Consensus Hull (CCH) was used where a 90% CH was calculated using a 1 pixel re-projection error tolerance. This generates a complete reconstruction of the scene at the cost of some loss of accuracy. However the loss of accuracy is considerably less than if the CVH is used alone which would require a re-projection threshold of up to 4 pixels in order to generate a complete reconstruction.

3.3. Tracking

The CCH is calculated for each input frame and generates a 3D triangle mesh representing the entire scene within the volume of interest. This single mesh contains reconstructions of all the players, the referee and the ball.

The connected components of the scene mesh are calculated to produce a set of discrete sub-meshes for each frame. A greedy algorithm then concatenates these sub-meshes over the sequence to generate per-player tracks over the entire sequence. The algorithm seeks to maximise spatial locality and minimise changes in volume between frames in the sequence.

The per-player mesh sequences generated by this tracking phase are then used as the input to the action matching algorithm.

4. Action Matching

The per-player 3D mesh sequences generated by the pre-processing stage are then matched against a library of synthetic animations to generate a representation of the recorded action.

A library of 3D animations representing human actions are reduced to their distinctive key-frames using 3D animation summarisation. These actions and their key-frames are then used to construct a hierarchical Markov model. An input sequence of meshes is also summarised using 3D animation summarisation. The path through the hierarchical Markov model which best matches the input sequence is then calculated. This path represent the key-frames in the library which best represent the input sequence. These key-frames can then be mapped back to the original library sequences to generate a synthetic representation of the input sequence.

This section first describes the shape similarity measure which is used as a basis for this technique and then describes the animation summarisation technique and finally the hierarchical Markov model used for action matching.

4.1. Shape Similarity

Volumetric shape histograms of the input and library sequences are used for comparison. First the mesh sequences are aligned frame by frame such that the direction of motion is always along the z-axis. Meshes are then scaled and translated to fall within a unit sphere at the origin. A volumetric representation V is then generated (where $V(x) = 1$ for points within the volume). Points x are then sampled on a regular grid and the Shape Histogram H is obtained as:

$$H_{i,j,k} = \sum_x V(x) B(i,j,k,x), \tag{1}$$

where B is the bin-membership function:

$$B(i,j,k,x) = \begin{cases} 1 & \begin{pmatrix} i\delta r < r_x < (i+1)\delta r \\ j\delta\theta < \theta_x < (j+1)\delta\theta \\ k\delta\phi < \phi_x < (k+1)\delta\phi \end{pmatrix} \\ 0 & \qquad otherwise \end{cases} \tag{2}$$

with x expressed in spherical co-ordinates (r_x, θ_x, ϕ_x) and histogram quantisation steps $(\delta r, \delta\theta, \delta\phi)$.

The distance between two Shape Histograms a and b is then calculated using the Kullback Leibler Distance K which is a symmetric measure based on the Kullback Leibler Divergence k [16]

$$k(a,b) = \sum_i a_i \log(a_i/b_i) \tag{3}$$

$$K(a,b) = \frac{k(a,b) + k(b,a)}{2} \tag{4}$$

4.2. Animation Summarisation

Summarisation allows long, complicated sequences of data to be represented by a sub-set of the data which captures the salient details of the original sequence. Key-frame extraction for video summarisation has long been studied in the field of video analysis and retrieval - a review of the state of the art can be found in[2]. As data has progressed from 2D to 3D, the concept of summarisation has also been adapted to 3D in work carried out by Huang *et al.* [23].

Action matching attempts to model an input sequence in terms of a set of library actions represented as states within the model. A simple approach is to represent each frame of the sequence as a separate state in the model, however this causes two problems.

Firstly many of the states are similar. As the likelihood of an observation originating from a state is related to the distance between feature vectors, many states that are close to a given observation will lead to similar observation likelihoods. Numerically this then generates an under-constrained Markov model which can quickly become swamped by noise.

Secondly, the structure of an animation (the sequential ordering of the frames) is encoded in the transition function $T(i,j)$ which gives the probability of transitioning from state i to state j. In order to encode the sequential nature of an animation, T should return a high value when j follows directly from i, and a low value otherwise. However strictly enforcing this constraint (*e.g.* $T(i, i+1) = 1$ otherwise $T(i,j) = 0$) also rigidly encodes the recording rate. So if someone is recorded as running with a period of 20 frames, the model will only match someone running with a period of 20 frames. Weakening the constraints (i.e. $T(i, i+1) = 1-\delta$ otherwise $T(i, j) = \delta$) allows for repeating and skipping frames. However, adding enough flexibility to properly allow for time-warping quickly leads to the model settling on degenerate solutions involving excessive skipping or repeating of frames.

A solution to this problem is to break the animation down into a discontinuous, high-level representation that seeks to capture only the salient features of the action. By representing an action in these terms the tight coupling to the recording rate is broken and the number of similar states is greatly reduced.

One way of obtaining such a representation is by clustering the shape histograms either in the feature space or some dimensionally-reduced space (using projective dimensionality reduction techniques such as PCA or manifold learning techniques such as Isomap[26]). However, such transformations lose the temporal sequential structure of the original animation and mapping back from the discovered states to an original pose can be ambiguous. These techniques also require an arbitrary cut-off as to which dimensions are significant and the number of clusters to be extracted as no clearly defined separation emerges from the data itself. Finally, the precise nature of the reduced dimensions and the clusters extracted from the data alter with the library of animations used which means that including extra animations may require re-parameterising and fine-tuning of the model.

An alternative solution that retains the structure of the animation is to make use of a technique known as animation summarisation. This technique attempts to represent an animation by way of a sub-set of its frames known as the key-frames (as shown in Figure 4). If an animation consists of n frames $f_0 \dots f_n$ and the summary consists of m frames $f'_0 \dots f'_m$ where $0 \leq m \leq n$ then a set of key-frames ν and a mapping μ between original frames and key-frames is chosen which minimise the rate r and the distortion d of the summary where:

$$r \propto m \tag{5}$$

$$d = \sum_{i=0}^{n} D(f_i, f'_{\mu(i)}) \tag{6}$$

with D being some distance metric.A variation of the 3D animation summarisation technique introduced in [23] is used. The technique uses the Kullback Leibler distance between shape histograms as the distance metric D, and measures the rate as $r = m$. A brute force search is then performed to determine the optimal values for ν and μ. Each key-frame then represents an interval in the animation and a direct mapping between intervals in the library and recorded sequences can be established. In this way a Markov model which is much more numerically stable can be constructed while retaining the direct mapping between recorded frames and proxy frames and allowing for variations in the rate of motion.

4.3. Hierarchical Markov Model

To model the activity in the input sequence a Markov model is constructed. Each state in the model represents a key-frame from the library sequences and each key-frame in the input sequence is an observation to be explained by the model. State transitions probabilities represent constraints on transitions between key-frames.

Constructing a single model to represent all possible states and all possible state transitions can become unwieldy and numerically unstable. One way to avoid this is to reduce the problem to a hierarchy of models - parent models to handle the transitions between library actions and child models to handle the structure within the actions themselves (i.e. the sequential nature of the key-frames including looping of a sequence).

Hierarchical Markov Models as used in the field of gesture recognition encounter problems when applied to the recognition of continuous actions. While an individual child model can be chosen at any point (either by looking ahead at the sequence or by consuming pre-segmented sections of the observation sequence), it is not possible to dynamically change between child models due to the re-initialisation problem. A child model must explain the current state and a set of previous states while a re-initialised child model only has to explain the current state. Comparing against re-initialised models will almost always favour the re-initialised model and lead to instability. Standard solutions are to allow re-initialisation only at segment boundaries or to allow all models to re-initialise when one model finishes. The problem of arbitrary re-initialisation is not handled.

This problem is solved by representing the system through a single parent and multiple child models. The parent model represents the progression of the sequence in terms of the most appropriate action to represent the sequence at any one time. Each child model matches the action it represents to the observed sequence on a frame-by-frame basis.

Both child and parent models are represented in the standard manner as a state transition probability matrix A, an observation probability matrix B and an initial state probability vector π, and the maximum likelihood state sequence is calculated using the Viterbi algorithm[25].

At each time t the window of observations $(t, t+1)$ is evaluated using each child model. The child model for action n is evaluated in two ways - once with π representing a uniform distribution over all states to give the re-initialised

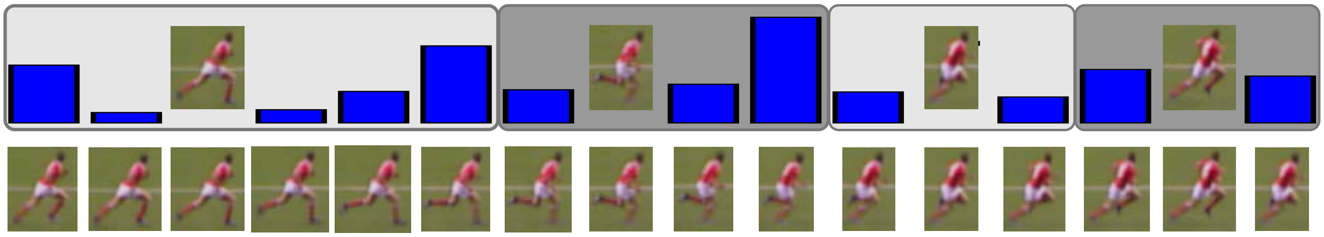

Figure 4. An animation summary. The bottom row is the original animation. The top row shows the key-frames in the summarised representation. The gray boxes indicate the frames of the animation represented by each key-frame. The blue bars above each frame are a plot of the distortion introduced by representing the frame with the respective key-frame.

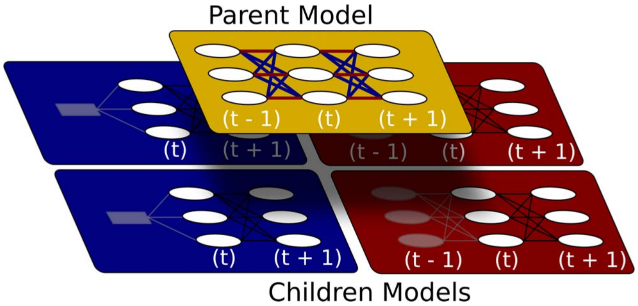

Figure 5. The system is represented by a single high-level model and multiple children models. Each state in the parent model corresponds to a child model (only the children of two parent states are shown in this diagram). Each child model is evaluated twice - once with re-initialisation at every time step (blue) and once retaining state from time step to time step (red). Colour coding of the transitions in the parent model show that the output of the blue evaluation is used when the parent model transitions between states over time, and the red evaluation is used when the parent model remains in the same state.

likelihood $P_n^r(t)$, and once using π calculated from the evaluation of the child model at $t-1$ giving the continuous likelihood $P_n^c(t)$ (see Figure 5). In both cases $A(i,j) = 1$ if $(j-i)\%m = 1$ otherwise $A(i,j) = 0$ (where m is the number of states in the model), and $B(i,j) = -\sqrt{K(i,j)}(K$ being described in equation 4). B is then normalised such that for each row $B(i)$, $min(B(i)) = 0$ and $max(B(i)) = 1$. In order to avoid weighting towards the re-initialised model π is always normalised so that all likelihoods sum to 1.

The parent model is then evaluated to determine the child model that will be active at any given time. A transition probability matrix is used such that $A(i,i) = 0.8$ and $A(i,j) = \frac{0.2}{m-1}$ (where m is the number of states in the model). This distribution is used to avoid over-fitting the model to the relatively small amount of data available. With larger data sets these likelihoods could be learned using Baum-Welch[25] or similar techniques.

B is then calculated such that $B(i,i) = P_i^c(t)$ and $B(i,j) = P_i^r(t)$ - i.e. if the parent model changes action then the observation probability is calculated using the re-initialised child model, and if the model stays on the same action then the child model is used without re-initialisation. In this way a path through the parent model can be calculated allowing for the re-initialisation of any child model at any time and allowing for the correct simultaneous evaluation of multiple possible paths through the model without the need for any up-front segmentation of the observation sequence into actions.

With this model it is easy to maintain the child-model state sequence Q_n for each child model n that accompanies the high-level state sequence $\mathbb{Q}$. At each time t the state sequences $Q_n^r(t)$ and $Q_n^c(t)$ are obtained from the child model. If the parent model calculates that the maximum likelihood path through n at t is a transition from another state, then $Q_n^r(t)$ is appended to Q_n, if the parent model indicates the maximum likelihood path through the model at t is a self-transition from the same state, then $Q_n^c(t)$ overwrites the previous entries in Q_n. In this way the final output state sequence can be calculated simply as $Q_n(t), n = \mathbb{Q}(t)$ giving the set of key-frames that best describe the sequence.

5. Results

The technique was evaluated on footage of a rugby and soccer match. Data was recorded using standard broadcast equipment. The recording equipment consisted of standard HD broadcast cameras. 12 cameras were recorded for the rugby data - 6 static cameras and 6 operator-controlled cameras. At any given time roughly half of these cameras were suitable for use by the system due to issues of framing and motion blur. The cameras were distributed around the pitch as shown in Figure 7. 16 cameras recorded the football footage - 3 were static and the rest were operator-controlled. Similar restrictions on usability meant that approximately 6 cameras were usable for reconstruction.

A library of 16 actions was generated by applying motion capture to a skeleton which was skinned to a human model. Actions consisted of jogging, 2 types of jumping, running at 4 speeds, skipping, sprinting at 3 speeds, standing, turning left while running, turning right while running, walking and walking backwards.

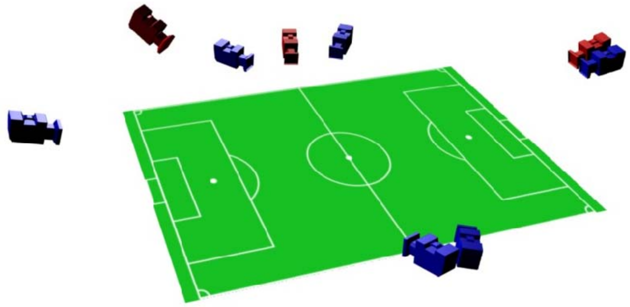

Figure 6. Arrangement of cameras used for the rugby data set. Static cameras are coloured blue while moving cameras are coloured red.

	Matches			%Matched	
Sequence	Exact	Near	Miss	Exact	Near
Rugby	294	198	245	40%	67%
Football 1	103	37	70	49%	67%
Football 2	101	43	38	55%	79%

Table 1. Evaluation of the generated pose estimates. Note that "% Matched Near" includes both Near and Exact matches and indicates all matches that are within an acceptable threshold of the original action.

To provide a quantitative analysis, every tenth frame of each sequence was examined and standing players assessed as either an exact match, a near match or a miss (players lying on the ground were not considered). Players were considered an exact match if the estimated pose was in the correct location with the correct orientation and in the correct phase of an appropriate action. If the pose was correctly located and oriented, but the phase was slightly incorrect then the match was considered a near match. Anything else was considered a miss. The results are shown in Table 1.

The action matching including shape histogram generation took approximately 0.2 seconds per-player per-frame on a 2GHz Intel Core2Duo T7300-based laptop. For comparison, a reference Matlab-based pose estimation algorithm took between 5 and 30 seconds per-player per-frame to run on a 3.3GHz Intel Xeon-based server. All code was run on a single CPU and was written in Python.

6. Conclusions

In this paper we have presented a fully automatic technique to match recorded 3D sequences against a library of actions. The technique provides an alternative to the more accurate but time-consuming process of manually placing and configuring avatars in a virtual environment. The system can be extended to cover a large library of actions and is robust to changes in camera arrangement, appearance of modeled players and action rates, allowing a single action library to be re-used in all scenarios.

Better pre-processing of the data to further reduce the noise in the system would improve performance. Access to larger data sets would allow training to replace the prior on transition probabilities with a learnt model. A richer library of human activities would also further improve matching as the current library only covers a limited number of possible actions.

7. Acknowledgements

This work was supported by the DTI Technology programme under Free-viewpoint video for interactive entertainment production TP/3/DSM/6/I/ 15515 and EPSRC Grant EP/D033926. For further details visit the iview project (http://www.bbc.co.uk/rd/iview).

References

[1] BBC Sports Virtual Replay Website. *http://www.bbc.co.uk/sports/virtualreplay*, 2008.

[2] M. Barbieri, L. Agnihotri, and N. Dimitrova. Video summarization: methods and landscape. *Internet Multimedia Management Systems IV*, 5242(1):1–13, 2003.

[3] K. Connor and I. Reid. A multiple view layered representation for dynamic novel view synthesis. *British Machine Vision Conference*, 2003.

[4] M. Dimitrijevic, V. Lepetit, and P. Fua. Human body pose detection using bayesian spatio-temporal templates. *Computer Vision and Image Understanding*, 104(2):127–139, 2006.

[5] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *IEEE International Conference on Computer Vision*, pages 726–733, 2003.

[6] Eye-Vision. Carnegie Mellon goes to the Super Bowl. *http://www.ri.cmu.edu/events/sb35/tksuperbowl.html*, 1996.

[7] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[8] S. Fine and Y. Singer. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, pages 41–62, 1998.

[9] S. Gammeter, A. Ess, T. Jaeggli, K. Schindler, B. Leibe, and L. J. V. Gool. Articulated multi-body tracking under egomotion. *European Conference on Computer Vision*, pages 816–830, 2008.

[10] O. Grau, A. Hilton, J. Kilner, G. Miller, T. Sargeant, and J. Starck. A free-viewpoint video system for visualisation of sport scenes. *SMPTE Motion Imaging*, pages 213–219, 2007.

[11] J.-Y. Guillemaut, J. Kilner, and A. Hilton. Robust graph-cut scene segmentation and reconstruction for free-viewpoint video of complex dynamic scenes. *IEEE International Conference on Computer Vision*, 2009.

[12] HawkEye. HawkEye Technology Website. *http://www.hawkeyetechnology.co.uk/*, 2008.

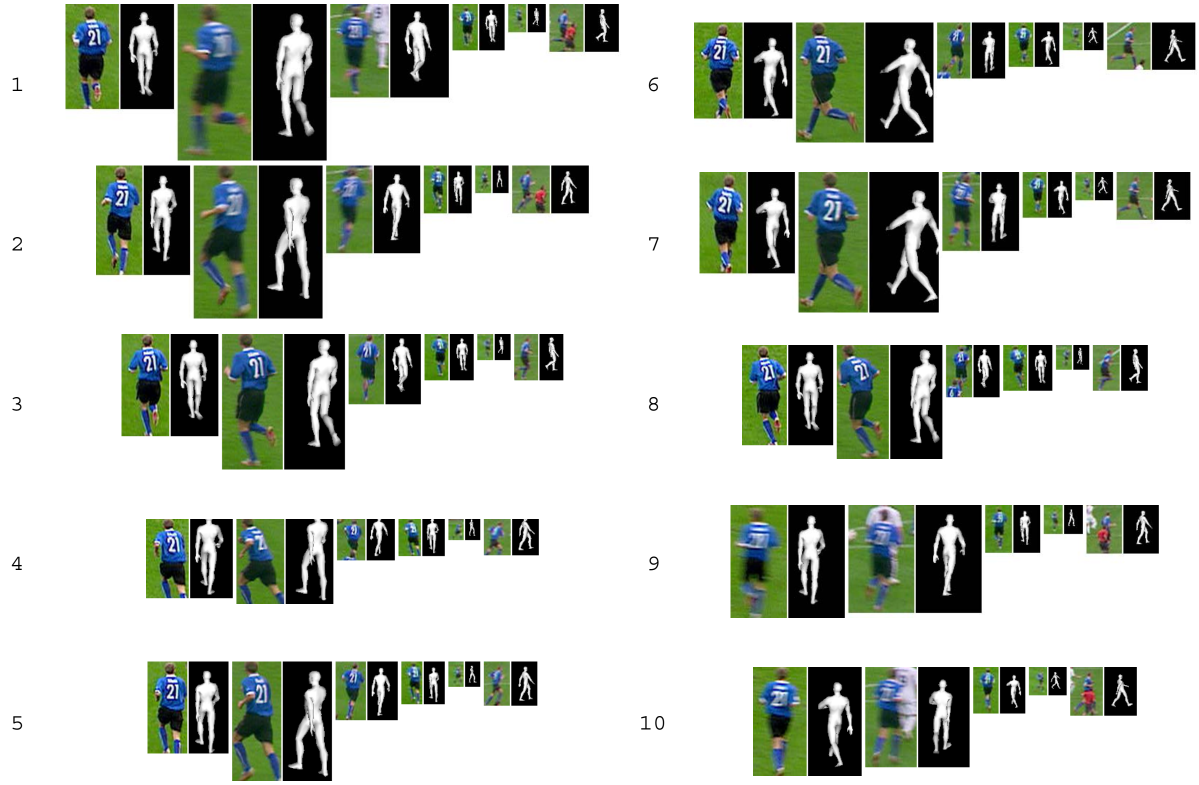

Figure 7. An example set of results as used in the evaluation described in Section 6. Images are taken at 10 frames intervals. The image labelled 7 is an example of what would be classified as an "exact" match, while 2 is an example of a "near" match as measured in the evaluation.

[13] N. Inamoto and H. Saito. Arbitrary viewpoint observation for soccer match video. *European Conference on Visual Media Production*, pages 21–30, 2004.

[14] T. Kanade, P. Rander, and P. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, 4(1):34–47, 1997.

[15] J. Kilner, J. Starck, A. Hilton, and O. Grau. Dual-mode deformable models for free-viewpoint video of sports events. *International Conference on 3-D Digital Imaging and Modeling*, pages 177–184, 2007.

[16] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[17] A. Laurentini. The visual hull concept for silhouette based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[18] LiberoVision. LiberoVision GmBH Website. *http://www.liberovision.com/*, 2008.

[19] G. Loy, M. Eriksson, J. Sullivan, and S. Carlsson. Monocular 3d reconstruction of human motion in long action sequences. *European Conference on Computer Vision*, 4:442–455, 2004.

[20] W.-L. Lu and J. Little. Simultaneous tracking and action recognition using the pca-hog descriptor. *Canadian Conference on Computer and Robot Vision*, pages 6–6, June 2006.

[21] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[22] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. *IEEE International Conference on Multimodal Interfaces*, 0:3, 2002.

[23] P.Huang, A.Hilton, and J.Starck. Automatic 3d video summarization: Key frame extraction from self-similarity. *International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.

[24] D. Pynadath and M. Wellman. Generalized queries on probabilistic context-free grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):65–77, 1998.

[25] L. Rabiner. A tutorial on hmm and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[26] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[27] G. Thomas. Real-time camera pose estimation for augmenting sports scenes. *European Conference on Visual Media Production*, pages 10–19, 2006.

[28] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. *IEEE International Conference on Computer Vision*, pages 1–7, 2007.