Human Activity Recognition for Synthesis

Yisheng Chen Rick Parent Raghu Machiraju Jim Davis
The Ohio State University
{chenyis, parent, raghu, jwdavis}@cse.ohio-state.edu

1. Introduction

Given a monocular video input which contains a moving human being, our goal is to automatically recognize the action and then track the articulated pose. Distinct from other example-based approaches such as estimating poses from still images [7] and using multiple cameras [6], our system uses a single camera to reconstruct human motions, combining knowledge from images and motion capture data together. Other model-based motion reconstruction systems are presented in [3, 4, 8].

Figure 1 illustrates the structure of our activity recognition and reconstruction system. We first create a motion capture database containing multiple subjects and action types, and then render the silhouettes from various view points (step A). Based on the training images, we build a hierarchical reliable-inference (RI) classifier [2] (step B). Once the preprocessing is done, we use the RI framework to classify observed motion into known actions by sequentially evaluating posterior class ratios (step C). Next several matches are picked from the database for each test image, based on image features, estimated pose configuration and camera angle transitions (step D). A dynamic programming technique is used to generate the sequences of natural and complex human motion from the database (step E). The pose parameters are further refined [1] to recover global translation and body orientation and improve the match with the original sequence (step F).

2. Action Recognition for Synthesis

2.1. Preprocessing and Training

Using the CMU Motion Capture Database, we include 4 actions, including jumping, punching, running and walking. All global translations and yaw transformations are removed to allow the focus to be on the essential characteristics of observed motion. Then we use the motion data to create synthetic examples and extract silhouettes from the rendered images (step A in Figure 1).

Then we use a RI classifier to recognize human actions, which are inferred from a posteriori probability model. We use feature vectors composed of similitude moments called pseudo Zernike moments, which possess the following characteristics: resilience to noise, information redundancy, and rank-preserving reconstruction capability. However a single image alone is not always sufficient to be classified as a particular action class. The use of additional motion images helps alleviate ambiguities and we use a multi-level sequential RI method [2] that automatically determines an appropriate threshold for each action exposure from negative examples. The motion history image representation is used to collapse multi-frame exposures into a single 2D template, and the likelihood distribution for each action class is modelled as a Gaussian mixture model. The training process is marked as step B in Figure 1.

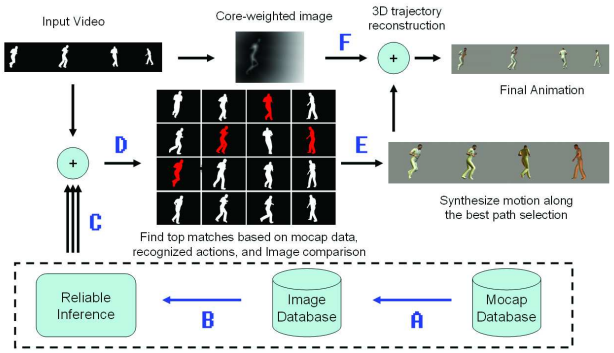

Figure 1. Diagram of our activity recognition and reconstruction method. The preprocessing stage is enclosed by the rectangle with dashed lines.

2.2. Recognizing Actions

In our framework, given an input test video, the multi-level sequential RI framework continually incorporates the subsequent motion image into a longer video exposure until a valid action class is found, using the thresholds determined in the training stage (step C in Figure 1).

2.3. Estimating Poses

After the input image has been classified, we try to find the pose that produces this image using the training data set. Given the hundreds images in the test video and dozens of close matches for each test image, we need to find a sequence of matched frames as to produce a good human motion. A good match is determined by the following factors (step D in Figure 1).

First, we keep the similarity between the silhouette from the dataset and the captured person in the test image. Here we use pseudo Zernike moments to measure the difference between two images. Second, we minimize differences in body orientation between successive frames. Since we re-

move all body yaw in the training dataset, this problem is equivalent to minimizing the transition of camera angles. We use Kalman filtering to track body rotation and eliminate unlikely matches. Third, we guarantee the smoothness of the joint angles with the progression of the test sequence. Thus we create motion graphs [5] from the training sequences. We find all local minima that are below a pre-determined threshold in those motion graphs, and allow transitions between those local minima. Using strongly connected component algorithm, we remove all dead ends in the graph to make it fully connected.

Inspired by Viterbi algorithm, our algorithm always tries to trace the best path while pruning the unlikely. Specifically, we only consider matches with the same recognized action. After examining the whole sequence, we only select the first K ranked paths, and then continue to process the next test image. Finally we backtrack the whole sequence to choose a path with the least cost (step E in Figure 1).

2.4. Reconstructing Translation and Yaw

When we acquire a good match sequence, the pose configurations and their associated cameras are available too. However, these poses do not have any translation or yaw, since the information has already been removed in the preprocessing stage. Core-weighted *XOR* comparison is used to retrieve the translation (step F in Figure 1).

We use the same 3D human model that renders the training images, and we build a silhouette-based objective function to match the 2D projection of the model to the input video. In our implementation, given a silhouette, a core-weighted image is a weighted sum of its Euclidean distance transform and that of its inverse image. Our cost function calculates the difference between two core-weighted images, generated from the test silhouette image and the model projection respectively. Since the objective function disallows the computation of analytic derivatives in terms of motion parameters, we employ the downhill simplex method to minimize the proposed cost function and retrieve the x-y-z translation.

Then, the new orientation of the body can be recovered by combining the 'yaw-less' body orientation of the example in the mocap database and the camera angle associated with the image.

3. Experimental Results

To illustrate our reconstruction method, several full body tracking sequences combining different actions are used. The upper row of Figure 2 shows a subject jumping from left to right and then turning and walking away. Our system not only labels the motion frames into correct action class, but also captures the arm and legs motion and the turning of the body, which is difficult for many tracking algorithms to follow. Furthermore, we reconstruct a smooth transition from jumping to walking, which is absent in our database. The lower row of Figure 2 shows our success

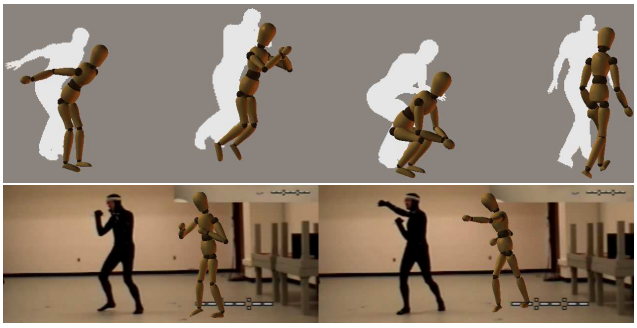

Figure 2. Upper row: *Jump* clip and its reconstructed pose. Lower row: *Boxing* clip and its reconstructed pose.

in tracking boxing motion. Although the punching examples in our database are performed by an actor standing still, our trajectory reconstruction algorithm rebuilds the vivid movement of the boxer. More reconstruction results can be found at http://www.cse.ohio-state.edu/~chenyis/ActionSynthesis.

4. Conclusions

We presented a framework for rapid-and-reliable action detection and robust 3D human motion reconstruction using image silhouettes. Based on a motion capture database, the approach builds an probabilistic action classification framework based on sequential reliable-inference and a sequence of poses close to the input sequence that is extracted from the database and then refined. The resulting system recognizes actions correctly and efficiently, and our reconstruction results not only show similarity to the original motion but also look natural. However, when the input video is not synchronized with the motions in the dataset, our system may choose a wrong path in the motion graph and generate unmatched motions. In the future, we expect an automated time-warping technique to solve the asynchrony problem and add more motions into the database to make our system more general.

References

[1] Y. Chen, J. Lee, R. Parent, and R. Machiraju. Markerless Monocular Motion Capture Using Image Features and Physical Constraints. In *Computer Graphics International*, 2005. 1

[2] J. Davis. Sequential reliable-inference for rapid detection of human actions. In *IEEE Workshop on Detection and Recognition of Events in Video*, 2004. 1

[3] D. DiFranco, T. Cham, and J. Rehg. Reconstruction of 3-D Figure Motion from 2-D Correspondences. In *CVPR*, 2001. 1

[4] I. Kakadiaris and D. Metaxas. Model-Based Estimation of 3D Human Motion. In *IEEE PAMI*, volume 22, 2000. 1

[5] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *Proceedings of SIGGRAPH 2002*. 2

[6] L. Ren, G. Shakhnarovich, J. Hodgins, H. Pfister, and P. Viola. Learning silhouette features for control of human motion. In *ACM Transactions on Graphics*, volume 24, 2005. 1

[7] G. Shakhnarovich, P. Viola, and T. Darrell. Fast Pose Estimation with Parameter-Sensitive Hashing. In *ICCV*, 2003. 1

[8] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3d human tracking. In *CVPR*, 2003. 1