Recognition-Based Motion Capture and the HumanEva II Test Data

Nicholas R. Howe
Smith College
Northampton, Massachusetts
nhowe@cs.smith.edu

Abstract

Quantitative comparison of algorithms for human motion capture have been hindered by the lack of standard benchmarks. The development of the HumanEva I & II test sets provides an opportunity to assess the state of the art by evaluating existing methods on the new standardized test videos. This paper presents a comprehensive evaluation of a monocular recognition-based pose recovery algorithm on the HumanEva II clips. The results show that the method achieves a mean relative error of around 10-12 cm per joint.

1. Introduction

A number of papers have proposed or tested human pose recovery systems based on *recognition* of known poses [15, 6, 19, 16]. Others have focused on techniques for pose regression [1, 2, 18]. More recent efforts have developed combinatorial methods for inferring the presence and pose of human figures based upon configurations of features like parallel edges, detected faces, etc. [20, 17, 13]. The newer combinatorial approaches may impose differing (and perhaps less restrictive) conditions on pose recovery. Nevertheless, advances in other areas may render seeming restrictions moot, and a full understanding of the state of the art requires quantitative comparisons. The HumanEva data provide a common basis by which the performance of all algorithms may be compared, and this paper will therefore present results for one recognition-based family of pose recovery algorithms.

This paper does not attempt to survey the full state of the art in human pose recovery, as others have done so already [14, 4]. It does not even encompass the growing subset of recognition-based motion capture algorithms [19, 16]. Rather it seeks to provide a comprehensive data point for a single method, as described in Section 2.

The system tested here is based upon earlier work by Howe [6, 7, 9], and makes the following set of assumptions. Like many methods of its ilk, it relies on accurate segmentation of the moving subjects from the background to yield silhouettes. This typically requires that the camera and background both remain static, although some work has developed methods that can segment people automatically when these assumptions are relaxed [22, 5]. Second, it requires offline access to a body of motion-captured training data containing examples of the sorts of movements and poses to be recovered. The system can recover arbitrary novel sequences of movements, so long as they do not include poses that stray too far from poses in the training set.

Under the two assumptions listed above, the system initializes itself without human assistance. It recovers an approximation of the subject's pose and motion without use of detailed camera calibration parameters or specific knowledge of the subject's measurements. Of course, such information where available can improve the accuracy of recovered poses and provide absolute spatial localization. (Note that the results presented in this paper do not actually rely upon camera calibration or subject size information for pose recovery, even where the HumanEva distribution includes those data.)

2. Algorithm

The general structure of a recognition-based pose tracker may be summarized as follows. First, the video input undergoes preprocessing to extract some useful set of features from each image frame. These features become the keys used to retrieve known poses from a library compiled from the training data. Because the library will typically not contain an exact match to the observed pose, and because the extracted features may not clearly differentiate the true pose from other poses with similar feature values, a collection of candidate poses should be retrieved for each frame [6]. This guards against situations where the correct pose may not be the top-ranked hit using the chosen feature set. Once the pool of candidate poses has been identified for each frame, the collection of observations forms a temporal Markov chain with a finite number of possible states, and forward-backward dynamic programming (sometimes called the Viterbi algorithm) can find the sequence of poses that minimizes an objective function. Typically, the objective function chosen will have both "smoothness" and "data" terms, to discourage solutions that change pose sharply between adjacent frames or do not closely match the observations. The remainder of this section describes each of these steps in further detail.

2.1. Feature Extraction

The method evaluated here uses two sorts of features: foreground silhouettes recovered via background subtraction, and optical flow in the foreground area obtained via Krause's algorithm [12]. These are complementary, the one giving precise information about the position of body parts visible in silhouette, the other giving information about movements inside the silhouette, yet less affected by clothing choices than the internal edges.

Krause's optical flow algorithm runs quickly but gives less accurate results than more recent, computation-intensive methods. Masking the flow by the foreground silhouette mitigates flow errors measured in the background due to noise. The flow in the foreground area is ultimately converted to ten simple low-degree moments in each flow component, as described in prior work [7]. Use of rotation-variant moments here reflects the expectation that the orientation of the subjects to be tracked will match that of the training data. This assumption applies to most video produced for human consumption, where the vertical world axis nearly always coincides with the vertical axis in the image plane. It may require reevaluation in other contexts, such as security camera video feeds, which will therefore require different sorts of training. All of the HumanEva videos use a standard vertical orientation.

The foreground segmentation used here does not employ any assumptions specific to the pose estimation task. Recent work has shown that performing segmentation and pose recovery simultaneously may improve the segmentation in difficult cases [11], but the staged approach used here suffices for good segmentation on the HumanEva data. The foreground segmentation employs background models trained on each pixel, building a model for each color plane in HSV space. For the HumanEva data, a single robust Gaussian per plane suffices, computed on the first 300 frames of each test clip using the trim mean and variance on the middle 20% of the data.[1] This procedure assumes that the background remains static and that the subject does not obscure any pixel in more than 40% of the frames, which is true on the HumanEva II clips and the single HumanEva I clip with results reported here. Clips not meeting these standards would require alternate model-building methods. Note that none of the results here use the background models supplied with the HumanEva data sets, as they contain subtle dissimilarities to the test clips that impair background subtraction quality.

For each frame, the ordinary scaled deviation from the model is simply the deviation from the mean, divided by the standard deviation. Experimentally, it turns out that each of the three HSV color planes requires slightly different treatment for best results. Hue can be noisy at low saturation. Saturation exhibits lower signal-to-noise than the other two planes. Value is generally quite accurate, except in the presence of shadows. These considerations lead to the adjusted computations below.

$$\Delta_H^*(x,y) = |H(x,y) - \mu_H(x,y)| \cdot \min(S(x,y), \mu_S(x,y)) \tag{1}$$

$$\Delta_H(x,y) = \frac{\max\left(0, 2\pi \cdot \Delta_H^*(x,y) - z_H\right)}{\sigma_H(x,y)} \tag{2}$$

$$\Delta_S(x,y) = \frac{|S(x,y) - \mu_S(x,y)|}{\sigma_S(x,y)} \tag{3}$$

$$\Delta_V(x,y) = \frac{\max\left(0, |V(x,y) - \mu_V(x,y) + \frac{z_V}{2}| - \frac{z_V}{2}\right)}{\sigma_S(x,y)} \tag{4}$$

$$\Delta(x,y) = w_H \Delta_H(x,y) + w_S \Delta_S(x,y) + w_V \Delta_V(x,y) \tag{5}$$

The HumanEva II videos all use these parameter values: $z_H = z_V = 0.1$; $(W_H, W_S, W_V) = (0.4, 0.2, 0.4)$.

Foreground segmentation is modeled informally as a Markov Random Field problem and solved by finding the minimal graph cut on an appropriate graph [8, 21]. The composite scaled deviations $\Delta(x,y)$ become edge weights in the graph. The graph cut minimizes an objective function on segmentations L that also includes a fixed penalty Δ_{FG} for assigning a pixel to the foreground and penalties for differing assignments on neighboring pixels.

$$\begin{aligned} E(L) \quad = \quad & \sum_{p:L(p)=1} \Delta_{FG} + \sum_{p:L(p)=0} \Delta(x_p, y_p) \\ & + \nu \sum_p \sum_q C(p,q)(L(p) \neq L(q)) \end{aligned} \tag{6}$$

Here ν controls the importance of connections between neighboring pixels, and hence the smoothness of the segmentation. $C(p,q)$ ranges from 0 to 1 and indicates the

[1] Because hue is an angular quantity, its mean is ill-defined. Expediency suggests introducing a discontinuity at some point far from observed values and computing an ordinary mean. The discontinuity goes opposite the "center of mass" of the angular values in a polar view. For simplicity of presentation, the remainder of this section assumes that all hue values are pre-linearized and mapped onto the range (0,1).

Figure 1. Sample foreground segmentation results. Note the detail visible in most of the boundary, including markers on the hands and near the shoulders. Shadow artifacts appear near the feet.

degree to which two pixels are considered neighbors. Four-connected pixels will normally have $C(p, q) = 1$, unless an edge appears in the image frame that is not present in the background model: $|I(p) - I(q)| - |\mu(p) - \mu(q)| > \tau$, for 4-neighbors p and q. Diagonally connected pixels are connected with a discount $C(p, q) = .3204$, designed to make diagonal and straight boundaries equally attractive.

The best parameter choice varies somewhat with different cameras. For the HumanEva II videos, all shot with similar equipment, the same parameters apply throughout: $\Delta_{FG} = 1.2$ and $\nu = 3$. These generate mostly clean segmentations; often the quality is high enough that the external markers used for the motion capture system can be clearly discerned (Figure 1). Some compromises are necessary; a lower value of ν or higher value of z_V would avoid shadow artifacts around the feet at the expense of occasional missed body sections.

Once computed, a chain code stores the foreground silhouette boundary at moderate resolution (200-300 boundary points). The chain code affords easy computation of the turning angle and half-chamfer distance metrics used below.

2.2. Pose Retrieval

The pose library comes from the HumanEva I training set for subjects S1, S2, and S3. Subject S2 also appears in a separate sequence in the HumanEva II test data, but subject S4 stands as a control for any undue advantage from this factor. Each training motion-capture clip is processed sequentially, with a frame selected for the library if it differs sufficiently from those already present. The library stores the chain-code boundary of the rendered silhouette of selected poses, as well as the flow moments computed from the instantaneous rendered flow. The library used in these experiments represents the union of the Jog and Walking training clips for three subjects, and has 1711 distinct frames. (Fewer frames would have been selected if the library processed all the data as a group instead of individually, because more duplicates would have been rejected. However, it is more convenient to simply combine libraries for different activity types.)

For each frame, several similarity measures retrieve their top poses from the pose library. Multiple measures may be combined using the sums of their individual rankings of the poses as a new composite score [3]. The pool of candidate poses for each frame comprises the following:

- 35 poses retrieved using a composite of flow moments, turning angle, and half-chamfer distance. 25 of these come from poses close to one of the last frame's candidate poses, 10 are chosen openly.
- 15 poses retrieved using flow moments alone. 10 of these come from poses close to one of the last frame's candidate poses, 5 are chosen openly.
- 15 poses retrieved using a composite of flow moments, turning angle, and half-chamfer distance. 10 of these come from poses close to one of the last frame's candidate poses, 5 are chosen openly.

Due to overlap between the different categories, the candidate pool for a frame usually has around 20-30 members. A full chamfer match registers each candidate with the silhouette observations, and the candidate pool is supplemented with the mirrored LOS-inverse poses. (The mirror LOS-inverse swaps the left and right sides of the body and simultaneously inverts along the camera line-of-sight axis; the result has the same silhouette as the original, and similar optical flow [6].) Poses whose chamfer match scores lag the leader's by more than 50% are pruned at this point, unless the pool would be left with fewer than ten candidates as a result.

2.3. Temporal Chaining

Regarding the video observations as a Markov process inspires the method for linking poses into a coherent temporal sequence. Unfortunately, the probabilities required for standard Markov analysis cannot be estimated directly. The linkage step therefore minimizes a heuristic objective function with data and smoothness terms.

$$E = \sum_{f=0}^{n} E_{data}\left(\Theta_f, I_f\right) + \sum_{f=2}^{n} E_{smooth}\left(\Theta_f, \Theta_{f-1}, \Theta_{f-2}\right) \tag{7}$$

Here $E_{data}(\Theta_f, I_f)$ is simply the symmetric chamfer distance computed in the previous section, and $Esmooth(\Theta_f, \Theta_{f-1}, \Theta_{f-2})$ represents the smoothness term. The latter in turn consists of two summed subterms: conservation-of-momentum [10] and match to flow observations [7]. Physical kinematics formulae on the articulated body model give the change in momentum, while the flow match computes at low resolution the mean error between the observed flow and the rendered flow from Θ_f to Θ_{f-1} to Θ_{f-2}. In the equations below, let body part j have mass M_j, moment of inertia I_j, translation $\dot{x}_j$ and rotation $\dot{\varphi}_j$.

Further, let P^* be the set of points in the intersection of a low-resolution grid with the subject foreground, ϕ_θ the flow rendered from θ_{f-1} and θ_f, and phi_{obs} the observed image flow. $|P^*| \approx 200$.

$$E_{smooth} = \lambda_1 E_{mom} + \lambda_2 E_{flow} \tag{8}$$

$$\begin{aligned} E_{mom} = & \\ & \sum_{j \in Parts} M_j \left[\dot{x}_j(\Theta_f, \Theta_{f-1}) - \dot{x}_j(\Theta_{f-1}, \Theta_{f-2})\right]^2 \\ & \quad + I_j \left[\dot{\varphi}_j(\Theta_f, \Theta_{f-1}) - \dot{\varphi}_j(\Theta_{f-1}, \Theta_{f-2})\right]^2 \end{aligned} \tag{9}$$

$$E_{flow} = \sum_{p \in P^*} \|\vec{\phi}_\theta(x_p, y_p) - \vec{\phi_{obs}}(x_p, y_p)\| \tag{10}$$

This work uses $\lambda_1 = 0.01$ and $\lambda_2 = 100$. Prior work has noted problems with the Markov optimization selecting solutions that abruptly shift between poses facing opposite directions [9]. Ideally the pose energy term should force alternate solutions, but this does not always happen, particularly when the arms and legs all lie close to the body axis. This work solves the problem in effective if somewhat *ad hoc* manner: $\Delta_\Theta(\Theta_f, \Theta_{f-1}, \Theta_{f-2}) \doteq \infty$ for any pair of successive frames whose pelvis facing differs by more than $90°$. With this restriction in place, the previously observed instabilities disappear.

3. Experiments

Results appear below for nine clips: four simultaneous color views of *S2-Combo-1*, four simultaneous color views of *S4-Combo-4*, and one color view of *S1-Walking-1* validation data. All come from the HumanEva II data set except for the last, which is included for purposes of comparison with HumanEva I results. All are processed identically save for the use of $\Delta_{FG} = 0.7$ for *S1-Walking-1* in compensation for camera differences between HumanEva I & II. The results below treat each camera viewpoint as monocular data, without considering information available in the other clips.

Analysis of the results appears in several forms. The reconstructed poses are registered to the 2D image frame coordinate system. Therefore, 2D error is measured in pixels using the absolute image coordinates. One pixel of error may correspond to varying world distance depending on the proximity of the subject to the camera. In three dimensions, distance from the camera is unknowable without knowledge of the camera parameters (which are assumed unavailable in general, even though they are provided for HumanEva II). Thus 3D error is computed up to an arbitrary translation of the body root, in millimeter units. Note that this still requires application of an unknown body scaling factor, which is supplied by using the median subject height from the training data. The 3D result is transformed into world coordinates using the camera calibration only to facilitate evaluation, as the online evaluation form expects answers in this reference frame.

Clip		Walking		Jogging	
Take	Cam.	2D	3D	2D	3D
S2 Combo 1	C1	19	120	17	116
S2 Combo 1	C2	19	115	17	109
S2 Combo 1	C3	26	166	17	99
S2 Combo 1	C4	18	145	18	144
S4 Combo 4	C1	25	218	18	129
S4 Combo 4	C2	19	193	16	120
S4 Combo 4	C3	16	188	15	106
S4 Combo 4	C4	22	219	18	155
S1 Walking 1	C4	15	99	N/A	N/A

Figure 2. Summary of mean tracking error for sequences in the HumanEva II comparison set. Walking includes frames 1–350; jogging includes frames 351–700. 2D error is absolute in image coordinates and measured in pixels. 3D error is relative to the body root (pelvis) and measured in millimiters.

Each clip comprises three parts: a walking section (frames 1 to 350), a jogging section (frames 351-700) and a balancing section (remaining frames). HumanEva I includes training data for walking and jogging motions, but not for balancing. Since a recognition-based method cannot handle the balancing sequence without going outside the HumanEva data for training data, results appear here only for walking and jogging.

Figure 2 summarizes the results in tabular form, while Figures 3 through 7 plot the frame-by-frame error. Analysis of the results shows several trends. Excluding the clips with evident gross errors, the 2D error appears remarkably stable, remaining mostly in the 15-19 pixel range. The 3D error also displays an apparent “baseline” error somewhere around 10-12 centimeters, but more often makes occasional forays above this level. The observed error baseline reflects the degree to which the training data can model the actual observation. Two strategies could lower this floor. Increasing the density of training data would allow the algorithm to retrieve poses closer to the observations, albeit at the price of a larger pose library to search. Alternately, if the preliminary solution provided by pose recognition could be further optimized to match the observed motion more closely, one could reduce the error without expanding the library. The trick is to optimize without losing the implicit prior on human poses embodied in the pose library.

The various peaks visible in the different plots appear where the result contains an obvious qualitative error, with corresponding effect on the quantitative results. These errors may be grouped according to their nature and severity.

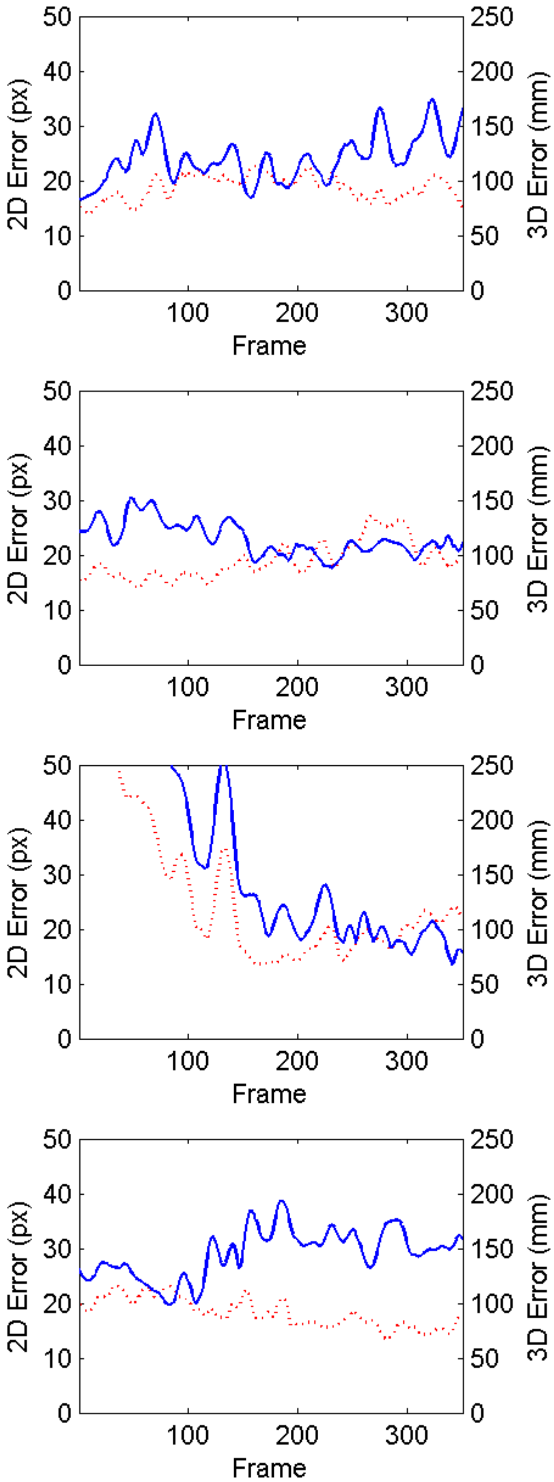

Figure 3. Plots of tracking error by frame for subject S2 walking (frames 1–350 of the S2-Combo-1 clip) for four cameras (C1 to C4, top to bottom). Absolute 2D error is shown dotted (red) and measured in pixels. Relative 3D error is shown solid (blue) and measured in millimeters.

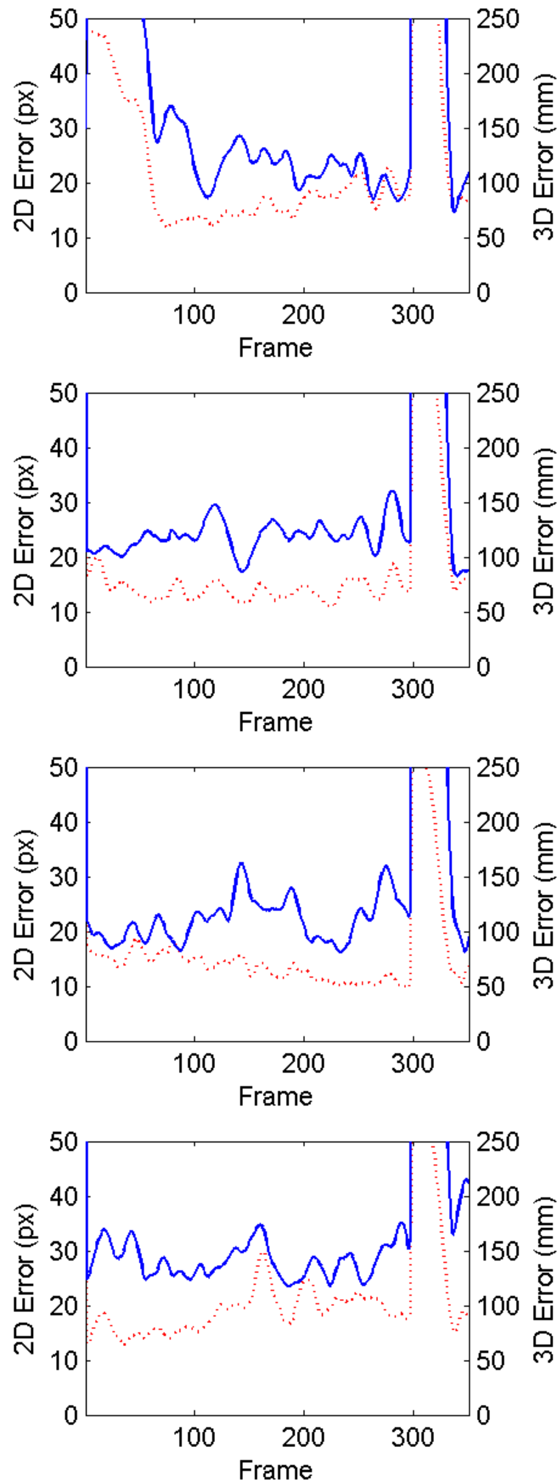

Figure 4. Plots of tracking error by frame for subject S4 walking (frames 1–350 of the S4-Combo-4 clip) for four cameras (C1 to C4, top to bottom). Absolute 2D error is shown dotted (red) and measured in pixels. Relative 3D error is shown solid (blue) and measured in millimeters. The peak in error around frames 298–336 is not visually apparent in the reconstruction and may indicate a problem with the ground truth.

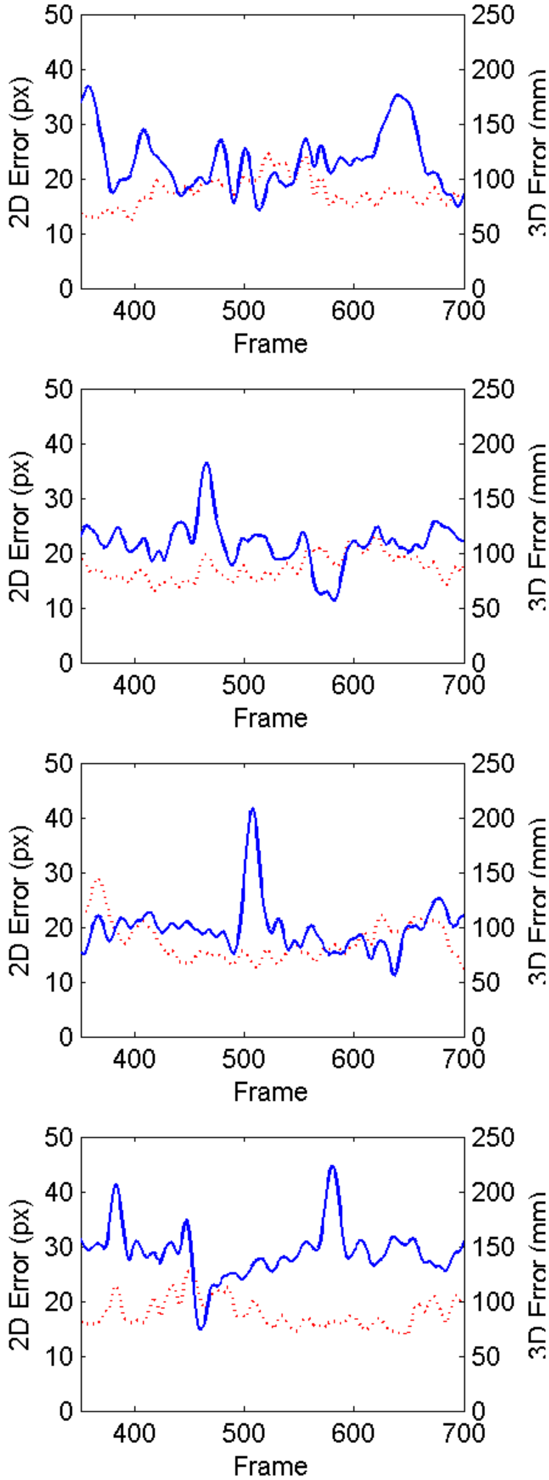

Figure 5. Plots of tracking error by frame for subject S2 jogging (frames 351–700 of the S2-Combo-1 clip) for four cameras (C1 to C4, top to bottom). Absolute 2D error is shown dotted (red) and measured in pixels. Relative 3D error is shown solid (blue) and measured in millimeters.

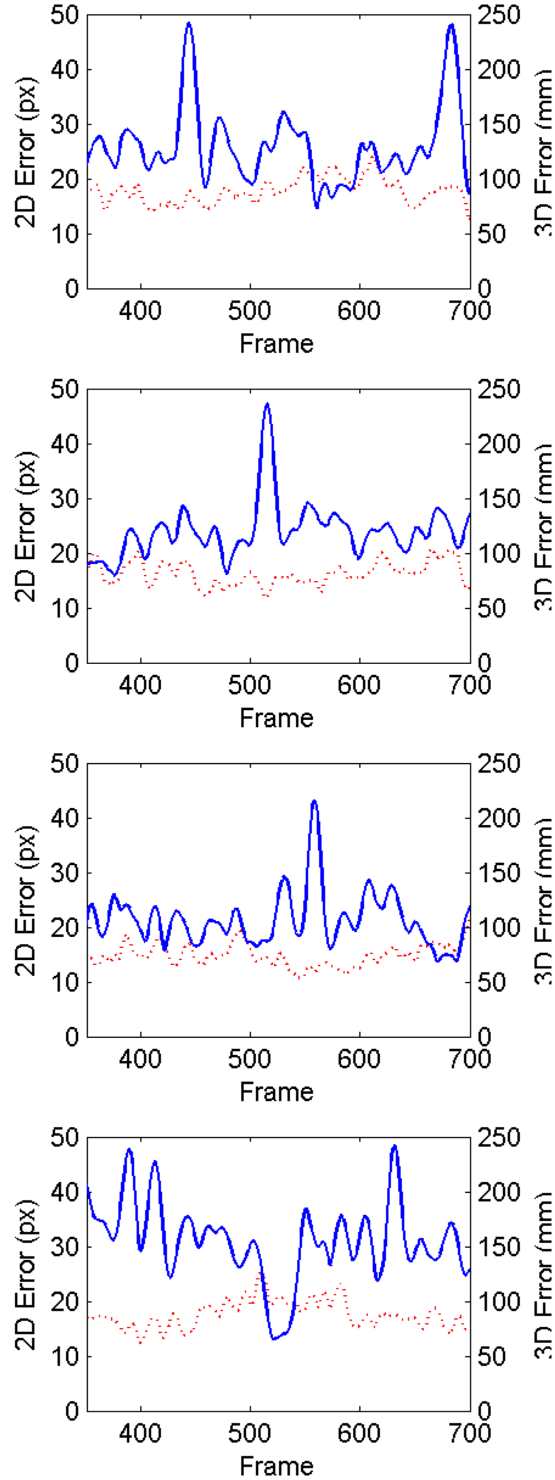

Figure 6. Plots of tracking error by frame for subject S4 jogging (frames 351–700 of the S4-Combo-4 clip) for four cameras (C1 to C4, top to bottom). Absolute 2D error is shown dotted (red) and measured in pixels. Relative 3D error is shown solid (blue) and measured in millimeters.

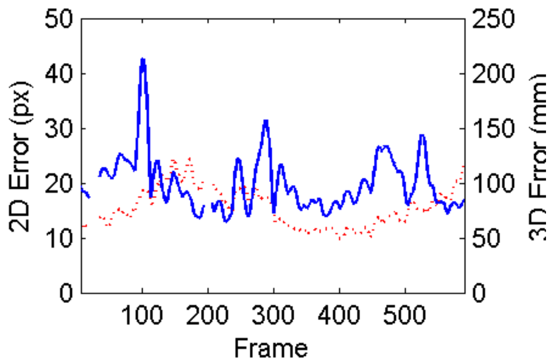

Figure 7. Plot of tracking error by frame for subject S1 walking validation sequence for one camera, C1 (HumanEva I data set). Absolute 2D error is shown dotted (red) and measured in pixels. Relative 3D error is shown solid (blue) and measured in millimeters.

A *stutter-step* represents a temporary switching of the feet in the reconstruction. This can happen when the recognition/retrieval step does not include a suitable correct candidate pose for some frame. A *slide* occurs when the feet stop moving for some number of frames as the figure continues moving forward. These are most commonly observed when the figure is moving either toward or away from the camera and the separation of the feet cannot be discerned in the silhouette. Although they appeared fairly frequently in early experiments on the HumanEva data, increasing the flow-matching weight λ_2 during chaining has largely eliminated the problem. A *reversal* error occurs when the turning direction of the reconstructed pose does not match reality; i.e., the subject actually turns $180°$ counter-clockwise while walking in a circle, but the reconstruction turns $180°$ clockwise instead. Partial reversals appear at the start of two of the walking clips (S2-Combo-1-C3 and S4-Combo-4-C1), reflecting difficult initial pose configurations for those clips. Erroneous pose reconstructions of this sort are consistent with the silhouette observations, but not with the flow observations. However, flow-based cues tend to be weaker than silhouette cues, and the ends of the Markov chain can be more difficult to solve when there is not a strongly identified pose serving to pin down the solution.

One set of peaks in the error does not correspond to any readily visible mistake in the reconstruction. The high error in frames 298-336 for all four views of S4-Combo-4 may be an artifact, because the reconstructed solutions appear normal. The most likely explanation is some flaw in the ground truth data for these frames. Omitting these questionable frames, the mean error for the affected walking sequences becomes 21, 15, 14, and 19 pixels (2D), and 146, 118, 111, and 144 mm (3D).

4. Lessons Learned

Evaluating the recognition-based motion capture algorithm on HumanEva data has provided valuable insight into its virtues and flaws. Simply running it on the eight 1200+ frame sequences in HumanEva II has helped to illuminate common failure modes and improve default parameter settings. On the other hand, the results also show it recovering successfully from such errors to return to the correct pose on subsequent frames. Furthermore, the observed accuracy shows the technique to be sufficient for many purposes: under good conditions, this form of recognition-based motion capture achieves relative 3D error around 10 centimeters per joint compared to the ground truth.

Acknowledgement

This manuscript is based upon work supported by the National Science Foundation under Grant No. IIS-0328741.

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *International Conference on Computer Vision & Pattern Recognition*, volume II, pages 882–888, 2004. 1

[2] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1), January 2006. 1

[3] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Information Processing and Management*, 31(3):431–448, 1995. 3

[4] D. Forsyth, O. Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan. Computational studies of human motion: Part 1, tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision*, 1(2/3), 2006. 1

[5] A. Fusiello, M. Aprile, R. Marzotto, and V. Murino. Mosaic of a video shot with multiple moving objects. In *IEEE International Conference on Image Processing*, volume II, pages 307–310, 2003. 1

[6] N. Howe. Silhouette lookup for automatic pose tracking. In *IEEE Workshop on Articulated and Nonrigid Motion*, 2004. 1, 3

[7] N. Howe. Flow lookup and biological motion perception. In *International Conference on Image Processing*, 2005. 1, 2, 3

[8] N. Howe and A. Deschamps. Better foreground segmentation through graph cuts. Technical report, Smith College, 2004. http://arxiv.org/abs/cs.CV/0401017. 2

[9] N. R. Howe. Evaluating lookup-based monocular human pose tracking on the humaneva test data. Technical report, Smith College, 2006. Extended abstract for EHUM 2006 workshop. 1, 4

[10] N. R. Howe. Silhouette lookup for monocular 3d pose tracking. *Image and Vision Computing*, 25(3):331–341, March 2006. Articulated and Nonrigid Motion. 3

[11] p. Kohli, P. Torr, and M. Bray. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *European Conference on Computer Vision*, pages 642–655, 2006. 2

[12] E. Krause. *Motion Estimation for Frame-Rate Conversion*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1987. 2

[13] C. McIntosh, G. Hamarneh, and G. Mori. Human limb delineation and joint position recovery using localized boundary models. In *IEEE Workshop on Motion and Video Computing*, 2007. 1

[14] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, March 2001. 1

[15] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision*, 2002. 1

[16] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1052–1062, 2006. 1

[17] D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 271–278, 2005. 1

[18] R. Rosales and S. Sclaroff. Combining generative and discriminative models in a framework for articulated pose estimation. *International Journal of Computer Vision*, 67(3):251–276, 2006. 1

[19] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *International Conference on Computer Vision*, pages 750–757, 2003. 1

[20] L. Sigal and B. M. J. Predicting 3d people from 2d pictures. In *IV Conference on Articulated Motion and Deformable Objects*, pages 185–195, 2006. 1

[21] Y. Sun, B. Yuan, Z. Miao, and C. Wan. Better foreground segmentation for static cameras via new energy form and dynamic graph-cut. In *ICPR (4)*, pages 49–52, 2006. 2

[22] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic, textured background via a robust Kalman filter. In *International Conference on Computer Vision*, pages 44–50, 2003. 1