# A Recognition-Based Motion Capture Baseline on the HumanEva II Test Data

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

January 11, 2011

**Abstract**

The advent of the HumanEva standardized motion capture data sets has enabled quantitative evaluation of motion capture algorithms on comparable terms. This paper measures the performance of an existing monocular recognition-based pose recovery algorithm on select HumanEva data, including all the HumanEva II clips. The method uses a physically-motivated Markov process to connect adajacent frames and achieve a 3D relative mean error of 8.9 cm per joint. It further investigates factors contributing to the error, and finds that research into better pose retrieval methods offers promise for improvement of this technique and those related to it. Finally, it investigates the effects of local search optimization with the same recognition-based algorithm and finds no significant deterioration in the results, indicating that processing speed can be largely independent of the size of the recognition library for this approach.

## 1 Introduction

Hundreds of papers in recent years have considered the problem of automated human pose recovery. This large body of research comprises assorted methods working towards various goals and making different assumptions. The late dearth of standardized test sets means that many papers include no quantitative results, and those that do mostly employ proprietary data. These confusing conditions have held back progress in the field, making it difficult to discern the strengths of different techniques. Fortunately, the debut of the HumanEva test data [33] offers a framework for clean comparison and experimentation by providing a high quality, public test set. HumanEva consists of multicamera video and synchronized motion capture (*mocap*) of multiple motion types performed by multiple actors, with designated training, validation and testing splits, and third-party evaluation of test results.

This paper addresses the task of three-dimensional pose recovery from a static uncalibrated single camera. The scientific literature describes a number of approaches to this problem, summarized in the section below. The experimental results presented herein use the entire HumanEva II test set plus an additional validation clip from HumanEva I to evaluate a recognition-based method drawn from previous work [10, 15]. In addition to testing the performance of the base algorithm and a local variant, the experiments herein also examine factors contributing to the error to infer promising research areas and evaluate the potential of recognition-based approaches in general.

## 1.1 Related Work

The first journal publications to use the HumanEva datasets were relatively sparse on numeric analysis, some not even giving numeric figures [41, 29, 39]. While this paper was under review many more results have been released; the HumanEva team has written a survey placing them in context [34]. Of these, many are not directly comparable to the results reported here because they employ multiple and/or calibrated cameras. Of the most relevant methods, several report very good results on HumanEva I [5, 20]. However, one that gives numbers for both HumanEva I and II does worse than the method herein on the comparable portion, and offers some evidence that portions of HumanEva I may be easier to accurately recover than HumanEva II [26]. Lee and Elgammal report excellent results but their method relies on strong motion priors [19].

Two slightly more dated surveys admirably categorize and attempt to make sense of the state of knowledge in the processing of human pose and motion [22, 8]. The subset of papers attempting to recover full-body pose in three dimensions from monocular input cleaves roughly according to their use of either *generative* or *discriminative* approaches, although some recent work has attempted to combine the two in order to capitalize on the distinct advantages of each [30, 37].

Generative methods can predict image appearance from pose and other parameters, allowing them to treat pose recovery as an optimization problem that seeks parameter values offering the best match to observations [32]. Despite the appeal of this approach, the many degrees of freedom in a human body and other scene considerations make tractability quite challenging. Most current work on generative approaches develops new tools for handling the complex optimizations required. Recent work has considered techniques including combinatorial methods [28], belief propagation [35], local gradient descent [21], and better statistical models [40].

Discriminative methods avoid the optimization problem by attempting to learn a direct mapping between image observations and underlying pose. Constructing such a mapping requires training data of some sort; these may consist of paired images and poses, or perhaps are synthetically generated from motion-captured pose data alone. Some discriminative approaches learn a regression from appearance to pose [1, 7, 2, 30], possibly neglecting the fact that dissimilar poses can have similar featural representations in most systems.

By contrast, *recognition-based* or lookup-based approaches simply retrieve stored or previously synthesized poses in response to image stimuli [23, 10, 31, 24]. In this manner, prior knowledge about human pose is embodied in the pose database rather than a learned regression from image to pose. Some closely related methods begin with retrieval from a database but use this to influence a density model propagated from frame to frame [36, 25].

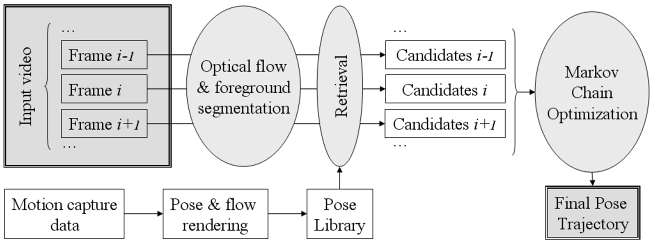

Figure 1: Data flow for the recognition-based pose recovery algorithm evaluated in this paper.

Recognition-based techniques can exploit a number of different image cues for pose retrieval. Published methods most commonly use silhouettes [23, 10, 24], but also employ edges [31, 3], histograms of gradients [27] and optical flow [11]. The specific features chosen help to determine the generality and reliability of a pose recovery system. For example, edges may not always appear on a subject in predictable locations, and accurate foreground segmentation to get a silhouette can prove problematic under adverse conditions. The next section discusses these matters in greater detail.

# 2 Algorithm

This paper evaluates a recognition-based pose recovery system based in general form upon earlier work by Howe [10, 11, 14, 12], with various modifications to improve the accuracy as detailed herein. First, the video input undergoes preprocessing to extract a feature set from each image frame. These features become the keys used to retrieve known poses from a library compiled out of the training data. Because the library typically will 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, the system retrieves a collection of candidate poses for each frame [10]. This guards against situations where the best 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 (the Viterbi algorithm) identifies the sequence of poses that minimizes an objective function. The objective function includes both "smoothness" and "data" terms, to discourage solutions that change pose sharply between adjacent frames or do not closely match the observations. Figure 1 summarizes the data flow of the system.

Successful pose recovery rests on a number of assumptions. For example, automated techniques must reliably extract the chosen features from the image data. The use of silhouette features typically requires that the camera and scene remain static, and even then errors will occur without sufficient figure/ground contrast. Although important, this limitation carries less force than in the past. This paper describes a foreground segmentation method that generates excellent results with the HumanEva data. Figure 2 shows the qualitative improvement possible over a simplistic background subtraction, suggesting that high-quality silhouettes can be achieved under some operating circumstances. Furthermore, the scope where accurate silhouette extraction can be performed continues to expand; other promising research uses feedback from the body model and recovered pose to enforce realistic segmentation results [42]. Although not applied in the present work, pose-model feedback seems promising as a natural augmentation of the techniques described here. Research has also begun to mitigate the requirements for a static camera [9] and static background [42].

Recognition-based methods also assume 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. This restriction suggests possible challenges for generality and scalability, since a system capable of recognizing arbitrary unrestricted poses would require a very large library that would take too long to search. Of course, retrieval speedup is an old problem, and some work has explored sublinear retrieval methods for pose recovery [31]. This paper tests a simple local retrieval mechanism that theoretically decouples retrieval speed from overall library size by searching only a small relevant subset of the library (although the research implementation does not exploit the potential speedup).

Given the assumptions listed above, the system initializes itself without human assistance and recovers a close approximation of the subject's pose and motion in three dimensions over time. Although the HumanEva distribution includes camera calibration parameters, the techniques presented in this paper do not rely upon camera calibration or subject size information for pose recovery. Of course, such information where available could potentially improve the accuracy of recovered poses and provide absolute spatial localization.

## 2.1 Feature Extraction

The method under evaluation employs two sorts of features: foreground silhouettes recovered via background subtraction, and optical flow in the foreground area obtained via Krause's algorithm [18]. 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 a feature like internal edges would be.

Krause's optical flow algorithm runs quickly but gives less accurate results than more computation-intensive methods. Masking the flow by the foreground silhouette therefore mitigates flow errors measured in the background due to noise. As described in prior work, twelve simple low-degree moments describe the optical flow in the foreground area [11]. 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 revision in other contexts, such as security camera video feeds, which will need correspondingly different sorts of training. All of the HumanEva videos use a standard vertical orientation.

The foreground segmentation used here broadly resembles work recently reported elsewhere [38], but differs in its details as described below. Recent work has suggested that performing segmentation and pose recovery simultaneously may improve the segmentation in difficult cases [17], but the decoupled approach used here provides sufficiently good segmentation on the HumanEva data. The segmentation begins by training background color models on each pixel for hue, saturation, and value color planes. For the HumanEva II 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 clips used for the experiments. Clips not meeting these standards would require alternate model-building methods, and continuous operation would require adaptive background modeling. Note that none of the results here employ the background models supplied with the HumanEva data sets, as those contain subtle dissimilarities to the test clips that reduce the quality of the ultimate foreground segmentation.

For each frame, the ordinary scaled deviation from the model would equal simply the deviation from the mean $\mu$, divided by the standard deviation $\sigma$. Experimentally, it turns out that each of the three HSV color planes requires a slight variant of this 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 heuristic considerations motivate the adjusted computations below.

$$\Delta(x,y) = w_H \Delta_H(x,y) + w_S \Delta_S(x,y) + w_V \Delta_V(x,y) \tag{1}$$

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

$$\Delta_H^*(x,y) = \|H(x,y) - \mu_H(x,y)\| \ \cdot \min(S(x,y), \mu_S(x,y)) \tag{3}$$

$$\Delta_S(x,y) = \frac{|S(x,y) - \mu_S(x,y)|}{\sigma_S(x,y)} \tag{4}$$

$$\Delta_V(x,y) = \frac{\max\left(0, |V(x,y) - \mu_V(x,y) + \frac{z_V}{2}| - \frac{z_V}{2}\right)}{\sigma_V(x,y)} \tag{5}$$

Equation 3 weights hue differences by the lesser of the two saturations. Equation 2 further ignores small hue differences below threshold $z_H$. Equation 5 discounts differences in value below threshold $z_V$ but only if they are darker than the mean. The following parameter values apply to all HumanEva II videos: $z_H = z_V = 0.1$;

[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 to a range surrounding the mean hue $\mu_H$.

$(w_H, w_S, w_V) = (0.4, 0.2, 0.4)$, counting saturation half as much as the other components.

Foreground segmentation is modeled informally as a Markov Random Field problem and solved in practice by finding the minimal graph cut on an appropriate graph [13, 38]. 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 cost $\Delta_{FG}$ for assigning a pixel to the foreground and penalties for differing assignments on neighboring pixels.

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

Here $\nu$ 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 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 $q$. (Here the norm is taken in RGB color space, and $\mu$ is the aforementioned mean of the pixel background model; $\tau = 0.05$.) Diagonally connected pixels are connected with a discount $C(p, q) = .3204$, a value chosen to make diagonal and straight boundaries equally attractive.[2] All other pixels are disconnected, $C(p, q) = 0$.

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 mocap system can be discerned (Figure 2). Not all compromises can be avoided: a lower value of $\nu$ or higher value of $z_V$ would eliminate shadow artifacts around the feet at the expense of occasional missed body sections. Postprocessing on the segmentation result selects the largest time-space connected component, eliminating transient foreground detections not attached to the subject.

Numeric measures confirm the high quality of the foreground segmentation results. Rendering motion-capture data on the HumanEva I validation sequence described in the experiments provides an approximate ground truth segmentation.[3] Compared with the background subtraction code provided with the HumanEva data, the approach described above incorrectly labels fewer pixels (1.2% of image area vs. 1.9%) and produces boundaries that are simpler (less than half the length) and closer to the motion-capture rendering (mean distance of 3.4 pixels vs. 7.8 pixels).

Once computed, a chain code represents the segmented foreground silhouette boundary. The chain code affords easy computation of the turning angle and half-chamfer

[2] A vertical boundary and a stairstep diagonal of equal length $l$ differ in the proportion of orthogonal to diagonal neighbor links crossing them. The vertical boundary is crossed by $l$ orthogonal connections and $2l$ diagonal links, while $l\sqrt{2}$ orthogonal connections and $l/\sqrt{2}$ diagonal connections span the diagonal boundary. To assign equal weight to the sum of the links across both, the diagonal links must be discounted by a factor of $(2\sqrt{2} - 2)/(4 - \sqrt{2})$.

[3] Although the rendered mocap data is not subject to gross errors, visual inspection of the images suggests that its boundaries may actually be less accurate than the segmentation result because the body model lacks perfect realism. Nevertheless it serves as a point of comparison.

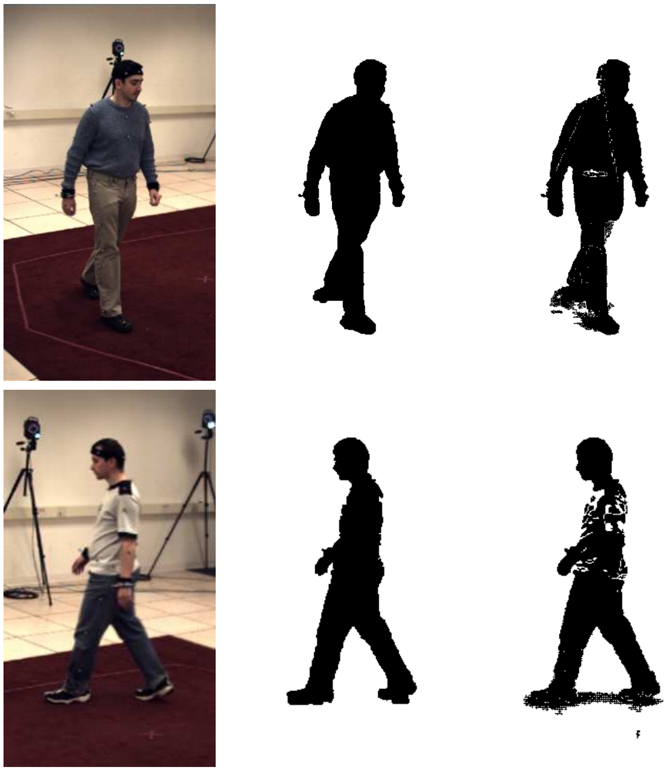

Figure 2: Sample foreground segmentation results (middle) for input images (left) using the improved algorithm in this paper. Note the detail visible in most of the boundary, including markers used to get ground truth on the hands and near the shoulders. Small shadow artifacts appear near the feet. The HumanEva baseline result based on straightforward Gaussian modeling [33] appears (right) for qualitative comparison. Improved silhouette finding allows for reconstruction even under difficult conditions.

distance metrics used below. In pathological cases where the foreground segmentation is disconnected, gaps may be artificially bridged at their narrowest points before computing the chain code. In practice the segmentation algorithm used here very seldom produces disconnected segments, and none of the test data required bridging.

## 2.2 Pose Library

The pose library draws its entries from synthesized views of the HumanEva I mocap on subjects S1, S2, and S3, performing jogging and walking motions. Subject S2 appears as an actor in both training and test data, but subject S4 appears only in testing and thus stands as a control for any undue advantage from this factor (which seems negligible in practice). The training process examines each motion-capture clip sequentially a frame at a time, selecting a pose for the library $\mathcal{L}$ if it differs sufficiently from those already present. Added poses are scaled to a standardized length (e.g., torso is always 40 units), rotated to a consistent pelvis orientation, and rendered under orthogonal projection. This viewpoint is the *library coordinate frame*. If $\{J_i(\psi)\}$ are the joint coordinates of a pose $\psi$, the difference between two poses $D_\psi$ is taken as the maximal change in position over all the joints in this frame. Selected poses must differ from all previous poses by more than $d_\mathcal{L} = 4$ units (around 5.5 cm); this corresponds to selecting every third frame or so from a novel motion sequence.

$$D_\psi(\psi, \psi') = \max_i \|J_i(\psi) - J_i(\psi')\| \tag{7}$$

Mocap data with a rendered body model generate the library of linked poses and silhouettes needed for retrieval-based reconstruction. Wireframe renderings of the body model used appear in Figure 6. It comprises rigid solids for each of 15 body segments (torso, neck, head, upper and lower arms and legs, plus hands and feet). Each segment has ellipsoidal endpoints, possibly of different dimensions, and a smoothly interpolated center section, to give a realistic appearance. Because the HumanEva motion data give positions of only six main body segments (the torso, head, and upper/lower arms/legs), positions of hands, feet, and neck are estimated in the rendered silhouettes based on the neighboring body parts.

The library stores the chain-code boundary of silhouettes of the selected poses rendered under orthographic projection from $n_A = 36$ viewpoints equally distributed in azimuth, as well as the flow moments as computed from the rendered flow. Flow renderings compute the image-plane motion of points on the body model surface visible at each pixel, using two adjacent frames of mocap data. An example appears in Figure 3. These experiments build libraries separately for the *Jog* and *Walking* training clips, selecting $|\mathcal{L}| = 1711$ distinct frames for inclusion. (Fewer frames would have been selected if the library processed all the data as a group instead of individually, because more duplicate poses would have been passed over. However, it is convenient simply to combine libraries for different activity types.)

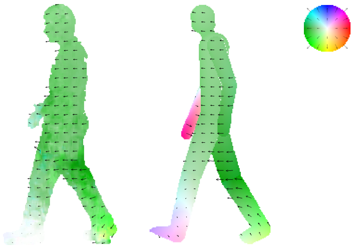

Figure 3: Sample flows: observed (masked Krause result, left) and rendered (right)

## 2.3 Pose Retrieval

For each frame, several similarity measures help to retrieve poses from the library, specifically the turning angle distance $D_\theta$, half-chamfer distance $D_\chi$, and flow moments $D_F$ [15]. For silhouette $S$, define $\{{}^S_q\vec{P}_i\}, i \in 1...q$ as a set of points spaced evenly on the boundary of $S$, such that ${}^S_q\vec{P}_1$ is at the topmost point and the indices progress clockwise numerically around the border. Also, let $\theta(\vec{P})$ denote the turning angle (a.k.a. tangential angle) of the silhouette border at $\vec{P}$.[4]

$$D_\theta(S, S') = \frac{1}{q}\sum_{i=1}^{q}\left|\theta({}^S_q\vec{P}_i) - \theta({}^{S'}_q\vec{P}_i)\right| \tag{8}$$

$$D_\chi(S, S') = \frac{1}{q}\sum_{i=1}^{q}\min_{j=1}^{q}\|{}^S_q\vec{P}_i - {}^{S'}_q\vec{P}_j\| \tag{9}$$

In practice, to compare an observed silhouette to a stored library pose, the library silhouette points ${}^{S'}_q\vec{P}_j$ are heuristically scaled and translated into approximate alignment as described below, and lookups on a distance transform of $\{{}^S_q\vec{P}_i\}$ allow efficient computation of Equation 9. The library points $\{{}^L_q\vec{P}_i\}$ are scaled so that the height of their bounding rectangle matches that of the observed silhouette, and translated so that the centers of the bounding rectangles coincide. Let ${}^S_q x_\downarrow = \min_{i\in 1...q} {}^S_q\vec{P}^x_i$, i.e., the minimum of the $x$ coordinates of the points, ${}^S_q y_\uparrow = \max_{i\in 1...q} {}^S_q\vec{P}^y_i$, the maximum of the $y$ coordinates, etc.

$${}^{S'}_q\vec{P}^x_i = \frac{({}^S_q y_\uparrow - {}^S_q y_\downarrow)}{({}^L_q y_\uparrow - {}^L_q y_\downarrow)}\left[{}^L_q\vec{P}^x_i - \frac{1}{2}({}^L_q x_\downarrow + {}^L_q x_\uparrow)\right] + \frac{1}{2}({}^S_q x_\downarrow + {}^S_q x_\uparrow) \tag{10}$$

$${}^{S'}_q\vec{P}^y_i = \frac{({}^S_q y_\uparrow - {}^S_q y_\downarrow)}{({}^L_q y_\uparrow - {}^L_q y_\downarrow)}\left[{}^L_q\vec{P}^y_i - \frac{1}{2}({}^L_q y_\downarrow + {}^L_q y_\uparrow)\right] + \frac{1}{2}({}^S_q y_\downarrow + {}^S_q y_\uparrow) \tag{11}$$

[4] By definition $\theta(\vec{P})$ varies smoothly, with discontinuities no greater than $\pm\pi$ between ${}^S_q\vec{P}_i$ and ${}^S_q\vec{P}_{i+1}$. Its value will exceed $2\pi$ in case of spirals, etc.

Let $F(x, y)$ represent an optical flow, with horizontal and vertical components $F^x$ and $F^y$. The vector of moments $\mathcal{M}$ used for retrieval consists of the components $\{M^x_{ij}, M^y_{ij} | i, j \geq 0, i + j \leq 2\}$.

$$M^x_{ij} = \frac{\sum_{(x,y) \in S} (x - \bar{x})^i (y - \bar{y})^j F^x(x, y)}{\sum_{(x,y) \in S} |x - \bar{x}|^i |y - \bar{y}|^j} \tag{12}$$

$$D_F(S, S') = \|\mathcal{M} - \mathcal{M}'\|_2 \tag{13}$$

Multiple measures may be combined using the sums of their individual rankings of the poses as a new composite score, after Belkin et. al. [4]. Retrieval may be *local*, coming from the set of poses close to one of the last frame's candidate poses, or *global*, retrieved from the entire pose library. Although local retrieval should allow efficient search via tree-based structures, in practice the prototype implementation simply identifies candidates within $D_\psi < 6$ units of the previous frame's picks via brute force search over the entire library. By default, the algorithm retrieves candidates according to the plan described below, which may be succinctly represented in two ordered triples giving the respective number of picks made via each of the three approaches shown: $N_L = (25, 10, 10)$ for local picks and $N_G = (10, 5, 5)$ for global picks.

- 35 poses retrieved using a composite of flow moments, turning angle, and half-chamfer distance. 25 of these are local and 10 are global.
- 15 poses retrieved using flow moments alone. 10 of these are local and 5 are global.
- 15 poses retrieved using a composite of turning angle, and half-chamfer distance. 10 of these are local and 5 are global.

Retrieval in multiple categories as described above provides redundancy in the case of bad silhouette or flow data. Due to overlap between the different categories, the candidate pool for a frame usually has around 20-30 members. The next step registers each candidate pose with the silhouette observations in the image frame. Translation of the projected library silhouette is initialized to match centroids with the observation, and further optimized by gradient ascent on the symmetric chamfer match score $D^*_\chi$. Scale is set to match the bounding box heights.

$$D^*_\chi(S, S') = D_\chi(S, S') + D_\chi(S', S) \tag{14}$$

Following retrieval, the candidate pool is supplemented with additional poses generated from the original pool by swapping the left and right sides of the body and simultaneously mirroring the pose along the camera line-of-sight axis (a *mirror-LOS* transform) [10]. Geometric arguments show that this generates a realistic pose with the same silhouette as the original, and similar optical flow. Adding these additional candidate poses at this point is equivalent in effect to having them in the pose library from the start, but achieves this benefit at little extra cost.

As a final heuristic efficiency, 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.

## 2.4 Temporal Chaining

Without any constraints, ambiguities in pose retrieval mean that the top candidate can flip abruptly between different modes from frame to frame. Treating the video observations as a Markov process provides 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 one data and two smoothness terms, computed efficiently via forward-backward dynamic programming.

$$Q = \sum_{f=1}^{n} Q_{fr}\left(\psi_f, I_f\right) + \lambda_1 \sum_{f=2}^{n} Q_{fl}\left(\psi_f, \psi_{f-1}\right) + \lambda_2 \sum_{f=3}^{n} Q_{mom}\left(\psi_f, \psi_{f-1}, \psi_{f-2}\right) \tag{15}$$

Here $\psi_f$ represents the 2D-registered pose at frame $f$. $Q_{fr}(\psi_f, I_f)$ represents the match to observations in frame $f$, computed as the symmetric chamfer distance (Equation 14) between the observed silhouette and the registered silhouette projected from $\psi_f$. $Q_{fl}(\psi_f, \psi_{f-1})$ measures the match between the rendered optical flow and actual flow observations [11]. $Q_{mom}(\psi_f, \psi_{f-1}, \psi_{f-2})$ penalizes reconstructions that violate conservation of momentum [15].

The flow match term $Q_{fl}$ computes at low resolution a rendered flow $F_\psi$ from $\psi_{f-1}$ to $\psi_f$. This is compared to the observed optical flow $F_{obs}$ for the corresponding frames, again at low resolution. Let $P^*$ be the set of points ($|P^*| \approx 200$) in the intersection of a low-resolution grid with the subject foreground $S_f$.

$$Q_{fl} = \frac{1}{|P^*|} \sum_{p \in P^*} \|\vec{F}_\phi(x_p, y_p) - \vec{F}_{obs}(x_p, y_p)\| \tag{16}$$

Physical kinematics formulae on the articulated body model give the change in momentum (neglecting contact forces). In the equations below, let body part $j$ have mass $M_j$ and moment of inertia $I_j$, with translation $\dot{x}_j$ and rotation $\dot{\varphi}_j$ computed from the three frames' poses. The mass and moment of inertia used are computed from the limb shapes in the graphically rendered body model, assuming uniform density throughout the body.

$$\begin{aligned} Q_{mom} = & \\ & \sum_{j \in Parts} M_j \left[\dot{x}_j(\psi_f, \psi_{f-1}) - \dot{x}_j(\psi_{f-1}, \psi_{f-2})\right]^2 \\ & \qquad + I_j \left[\dot{\varphi}_j(\psi_f, \psi_{f-1}) - \dot{\varphi}_j(\psi_{f-1}, \psi_{f-2})\right]^2 \end{aligned} \tag{17}$$

This work uses $\lambda_1 = 0.01$ and $\lambda_2 = 100$. Prior work notes occasional problems with the Markov optimization selecting solutions that abruptly shift between poses facing opposite directions [14]. Ideally the momentum term should select against such errors, but to definitively rule out any problems of the sort, this work adopts an *ad hoc* restriction: set $Q_{motion}\left(\psi_f, \psi_{f-1}, \psi_{f-2}\right) \doteq \infty$ for any pair of successive frames whose pelvis facing differs by more than 90°.

Markov optimization finds the most continuous sequence of library poses it can, but the resulting motion will appear jerky at times when the library does not contain a smoothly interpolating pose. A final smoothing operation eliminates this source of jitter [10]. It requires a pose parameterization, chosen such that no parameter includes a discontinuity within the human range of motion. A low-pass filter smoothes each parameter over time, eliminating sharp changes between frames. Comparison of pose results before and after the smoothing operation reveals that it tends to increase accuracy, but only slightly. The results appear in Tables 1 and 2 below.

# 3 Experiments

The experiments presented below primarily use the HumanEva II data set, focusing on the jogging and walking segments in accordance with the priority recommendations of the HumanEva creators. HumanEva II comprises four simultaneous color views of *S2-Combo-1* and four simultaneous color views of *S4-Combo-4*. For comparison purposes the experiments also use one color view of *S1-Walking-1* validation data from HumanEva I. Parameter settings remain fixed throughout save for one exception: $\Delta_{FG} = 0.7$ for *S1-Walking-1* in compensation for camera differences between HumanEva I & II. All results treat each camera viewpoint as monocular data, without utilizing information from the other clips.

The clips from HumanEva II include three distinct parts: a walking segment (designated as frames 1 to 350), a jogging segment (frames 351-700) and a balancing segment (remaining frames). The HumanEva team identifies the walking and jogging segments as the priority test set, and the results given herein and cited from other research as a comparison all refer to these two segments only. In practice it would be difficult to conduct a standardized trial of the third segment, because a recognition-based method cannot properly handle the balancing motion without going outside the HumanEva data for training data. Note that although the walking and jogging are analyzed separately for evaluation purposes, the same system runs on all 700 frames without being told which activity is being performed.

## 3.1 Error Evaluation

The HumanEva designers provide an automated evaluation system to allow assessment of reconstruction quality without revealing the secret motion-captured ground truth and thus compromising the test suite. All numeric results in this paper were computed through this system, whose design imposes constraints on the error analysis. The evaluation module compares the reconstructed coordinates of 20 specified joint markers (to be provided by the algorithm under consideration) to the ground truth coordinates of those points, computing and reporting for each frame a single number: the mean distance between all corresponding pairs of points [33]. (Although it would be useful to compute the error separately for each individual joint, the HumanEva designers chose not to make this information available, probably to avoid compromising the secrecy of the ground truth data.) The evaluation can assess the error in both 2D and 3D reference frames, with certain differences in handling between the two cases as noted below.

While the evaluation module stores ground truth data in world coordinates, the retrieval-based algorithm produces results in library coordinates, the frame implied by the orthogonal projection described in Section 2.2. Before points can be compared they must be transformed into the same frame of reference by translation, rotation, and scaling.[5] Let $X = \{x_1, x_2, ..., x_{20}\}$ be the 3D reconstructed coordinates of the joint positions in library coordinates, and $\hat{X} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_{20}\}$ the 3D ground truth positions of the corresponding points in world coordinates. Further, define transformation functions to various common reference frames: $\Phi_I(x)$ projects library coordinates into 2D and then scales and translates them to conform to image frame coordinates, according to the registration computed during pose retrieval (Section 2.3). $\hat{\Phi}_I(\hat{x})$ converts world coordinates into the same 2D image frame via the camera calibration. $\Phi_G(x)$ converts library to world coordinates, up to an arbitrary translation as described below.

Using these definitions, the system computes absolute 2D error in pixels and relative 3D error in centimeters as follows:

$$ERR_{2D}(X, \hat{X}) = \frac{1}{20} \sum_{m=1}^{20} \left\| \Phi_I(x_m) - \hat{\Phi}_I(\hat{x}_m) \right\| \tag{18}$$

$$ERR_{3D}(X, \hat{X}) = \frac{1}{20} \sum_{m=1}^{20} \|(\Phi_G(x_m) - \Phi_G(x_1)) - (\hat{x}_m - \hat{x}_1)\| \tag{19}$$

The expression for 3D error compares displacements relative to the body roots ($x_1$ and $\hat{x}_1$) because the absolute translational relationship between the library and world reference frame origins remains unknown. Their rotational relationship is known because the library coordinate frame is aligned with the image frame, and HumanEva data provides the camera calibration necessary to rotate into the world frame. (Note that the calibration is only used for evaluation; the reconstruction itself assumes a general case where calibrations are unavailable.) The relative scale of library to world coordinate systems is approximated using the median subject height from the training data. Knowing the height and limb lengths of subjects would no doubt improve the results, but as with the camera calibration, reconstruction methods must presume such data are unavailable in general. Methods to automatically recover limb lengths from the video could prove useful, but are not investigated here.

A subtle correction to the rotation applied in $\Phi_G$ reduces the computed error values as compared with previously reported results [12].) The pose library renders silhouettes in orthographic perspective, but a real camera is subject to perspective effects. This means that the rotation indicated in the calibration parameters applies only to the central point of the image; when the subject appears either to the left or the right of center the rotation must adjust accordingly in $\Phi_G$, as illustrated in Figure 4. A similar consideration holds for camera pitch, but it has negligible effect because pitch is minimal in the HumanEva camera views, and the reconstruction processing uses a zero-pitch pose library and produces zero-pitch solutions.

[5] Because of this requirement the HumanEva evaluation tool cannot be considered fully relative. One would prefer a system that automatically matches position, rotation, and scale between any pose input and ground truth, and returns an error measure normalized to overall body size.

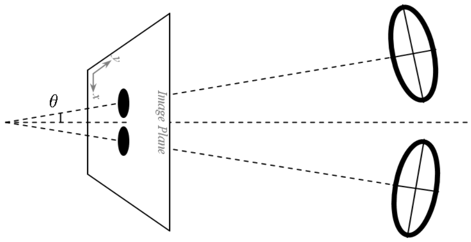

Figure 4: A correction to body rotation (top view) is computable from the horizontal coordinate in the image plane and the camera parameters. Under perspective projection, different rotations in world coordinates will generate the same silhouette depending on whether their projection is to the left or right of the center. A correction for this rotation is incorporated into $\Phi_G$ before comparison of the reconstructions with the HumanEva ground truth.

### 3.2 Baseline Results

Table 1 summarizes the mean joint position error in tabular form, while Figure 5 provides a visual comparison to other results: a local-only variant described below in Section 3.5 and two previously reported efforts. Analysis of the results shows several trends. The mean 2D joint error stands at around 14 pixels. The relative 3D joint error has a median by frame at 7.9 cm across all the clips, but more often contains peaks significantly above this level; the mean by frame is 8.9 cm. This level of accuracy improves on results for HumanEva II published in related workshops, which had been above 10 cm [27, 6, 16]. Full videos of each reconstruction appear in the supplementary files for this paper.

Peaks in the error rate correspond to obvious qualitative mistakes, some examples of which appear in Figure 6. These errors may be grouped according to their nature and severity. A *stutter-step* represents a temporary switching of the feet in the reconstruction. This can occur if 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 slides appeared fairly frequently in early experiments on the HumanEva data, increasing the flow-matching weight $\lambda_2$ in Equation 15 has largely eliminated the problem. A *reversal* error occurs when the reconstructed pose stands in mirror opposition to the reality; i.e., the subject actually turns left instead of right, or has the arms and legs backwards. 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 no strongly identified pose serving to pin down the solution.

Table 1: Mean tracking error. Walking includes frames 1–350 (but omits frames 298–336 for *S4-Combo-4* due to missing ground truth). 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 centimeters.

| Clip | | Walking | | Jogging | | Walk/Local | | Jog/Local | |
|---|---|---|---|---|---|---|---|---|---|
| Take | Cam. | 2D | 3D | 2D | 3D | 2D | 3D | 2D | 3D |
| S2 Combo 1 | C1 | 16 | 8.1 | 15 | 8.5 | 15 | 7.8 | 15 | 8.1 |
| S2 Combo 1 | C2 | 16 | 7.3 | 14 | 7.3 | 17 | 8.7 | 14 | 7.3 |
| S2 Combo 1 | C3 | 23 | 14.3 | 14 | 7.7 | 21 | 13.3 | 14 | 8.0 |
| S2 Combo 1 | C4 | 15 | 7.8 | 15 | 8.4 | 16 | 8.0 | 15 | 8.6 |
| Mean S2 | All | 17 | 9.3 | 14 | 8.0 | 18 | 9.5 | 15 | 8.0 |
| S4 Combo 4 | C1 | 17 | 10.9 | 13 | 10.2 | 17 | 11.1 | 13 | 10.5 |
| S4 Combo 4 | C2 | 11 | 8.3 | 12 | 8.8 | 14 | 10.8 | 12 | 8.7 |
| S4 Combo 4 | C3 | 9 | 7.9 | 10 | 9.4 | 9 | 7.9 | 11 | 8.7 |
| S4 Combo 4 | C4 | 14 | 9.0 | 13 | 9.9 | 14 | 9.6 | 13 | 10.0 |
| Mean S4 | All | 13 | 9.0 | 12 | 9.6 | 14 | 9.8 | 12 | 9.5 |
| S1 Walking 1 | C1 | 12 | 7.2 | N/A | N/A | 13 | 7.5 | N/A | N/A |

The results do not show a statistically significant difference between subject 2, whose motions from a different take are included in the training set, and subject 4, who is previously unseen. Subject-specific effects may depend on the type of motion performed, since the walking test shows a smaller difference than the jogging. In any case, one cannot draw strong conclusions from a study with only two test subjects (as provided in HumanEva II).

The prototype implementation consists of mostly unoptimized Matlab code, including file I/O operations to load and save intermediate results. Reconstruction under these conditions takes about five seconds per frame on a desktop PC from pose retrieval to final product. An efficient C implementation would doubtless cut processing time substantially. Real-time operation might be achievable by implementing portions on a GPU or other parallel architecture, but this can be addressed in future work.

## 3.3 Error Analysis

A mean error of 8.9 cm per joint may suffice for many pose recovery applications, but further improvement would be welcome. What factors contribute to the observed error rate? A certain amount of error is systemic: the implementation of the algorithm described in the previous section uses an internal representation of pose slightly different from that used by the HumanEva data. In particular, it replaces the limb dimensions of a particular subject with mean values obtained from the training data. Also, the conversion between formats may introduce errors due to differing interpretation of the control point positions. These mismatches add up: for the *S1-Walking-1* clip validation data, converting to the internal pose format and back introduces a mean error per joint

Figure 5: Bar plot of mean error for 17 clip segments, with framewise standard deviation shown where available. Within each grouping, results appear as follows from left to right: global+local retrieval result (default), local-only retrieval result, results reported by Poppe [27] and by Husz et. al. [16].

of 3.7 ± .2 cm. Although significant, this factor alone does not suffice to explain the rates of error observed in the experiments.

Insufficient library coverage could potentially cause elevated error. Since the initial phases of the algorithm limit the solutions considered to candidate poses retrieved from the library, the lowest error achievable will be limited by the library's best match to the actual ground truth pose. Smoothing may improve the result somewhat, since it can generate new poses beyond those found in the library, but in practice smoothing tends to exert rather small influence on the error, giving improvements on the order of a few millimeters.

Despite the considerations above, several observations suggest that the current density of coverage in the pose library could support lower error, and therefore insufficient library coverage also cannot explain the observed error rates. The ground truth validation data in the *S1-Walking-1* clip provide one test. Searching the library for the closest match to ground truth in each frame reveals that the algorithm could achieve 4.2 cm mean error per joint if it consistently identified the best available pose. Even a much more sparsely filled library, built using $d_{\mathcal{L}} = 6$ cm, $n_A = 24$ and containing only 822 poses still achieves 4.8 cm mean error per joint under these ideal circumstances. What is more, mean error per joint remains unchanged in actual practice with the sparser library, at 8.9 cm over all clips.

Given these observations, it appears that suboptimal retrieval from the pose library deserves the most scrutiny in the error analysis. Indeed, further investigation reveals that the retrieval step returns the optimal pose within the candidate pool on only 27 of 557 possible frames for the standard library, and 57 frames for the sparse library. In part this occurs because the video input does not contain 3D information used to determine the optimal pose match. But it appears likely that the features and measures used for retrieval in these experiments discriminate poorly between close matches to the actual pose, and this confusion increases with library coverage density.

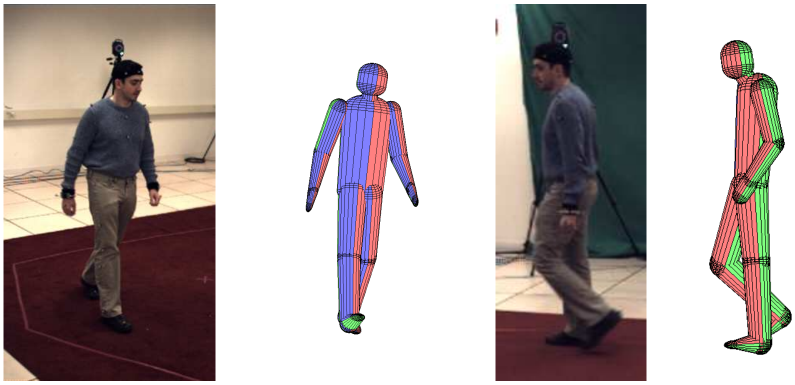

Figure 6: Sample erroneous reconstructions. A small slide occurs on *S2-Combo-1-C1* around frame 100 (left). Poor initialization causes a reversal on *S2-Combo-1-C3*, shown right at frame 114. The reversal is corrected soon after via a stutter-step.

Despite suboptimal retrieval, the candidates identified still suffice to produce a consistent final reconstruction. This is good news in one sense: recognition-based pose recovery works even with imperfect pose retrieval, but better retrieval methods might substantially improve the error. Some other mechanisms are already in use [31, 27]; determining which of these or others do the best job should become a near term priority for research in recognition-based pose recovery.

### 3.4 Parameter Sensitivity

The system as described depends upon a number of parameter settings. Many of these serve relatively minor purposes, and the exact values need not be tuned carefully. A few require more attention, as detailed in this section. Evaluation of parameter sensitivity is somewhat difficult as many of the crucial parameters exhibit a "cliff effect": changing them moderately results in little or no change in results, but at some critical threshold the outcome can degrade suddenly and dramatically, as for example a large area of background is redesignated as foreground, or vice versa, or the pose reconstruction gets stuck in a local minimum.

The most sensitive set of parameters concern the foreground segmentation, upon which the remaining steps depend heavily. Not only can mistakes here doom subsequent pose recovery [14], but the different parameters interact and so cannot easily be varied independently. The values chosen for $w_H$, $w_S$, $w_V$, $z_H$ and $z_V$ all seek to maximize the contrast between the difference signal in the foreground pixels and any noise present in the background. With this signal-to-noise maximized, $\Delta_{FG}$ may be lowered as much as possible so as to detect low-contrast body parts. Likewise, $\nu$ controls the sensitivity of the foreground outline to noise. Changes to any of the former five parameters will necessitate corresponding adjustments of the latter two. Experience has shown that the optimal parameter values usually lie near the values reported for this

Table 2: Results of parameter-variation studies for *S4-Combo-4-C4* clip. All numbers are relative 3D error, in centimeters.

| Variant: | Walk | Jog |
|---|---|---|
| A. $\lambda_1 = 0.001$ | 8.9 | 14.3 |
| B. $\lambda_1 = 0.1$ | 24.3 | 15.1 |
| C. $\lambda_2 = 10$ | 8.8 | 9.8 |
| D. $\lambda_2 = 1000$ | 9.0 | 9.9 |
| E. Smaller pool: $N_L = (15, 6, 6)$; $N_G = (6, 3, 3)$ | 24.2 | 20.2 |
| F. Larger pool: $N_L = (32, 16, 16)$; $N_G = (16, 8, 8)$ | 8.7 | 14.0 |
| G. All global: $N_L = (0, 0, 0)$; $N_G = (35, 15, 15)$ | 24.5 | 15.1 |
| H. Composite only: $N_L = (45, 0, 0)$; $N_G = (20, 0, 0)$ | 8.7 | 10.2 |
| I. Sparse library: $d_{\mathcal{L}} = 6$ cm, $n_A = 24$ | 9.0 | 14.2 |
| J. Dense library: $d_{\mathcal{L}} = 2$ cm, $n_A = 48$ | 8.3 | 9.5 |
| K. Poor segmentation: $\Delta_{FG} = 2.4$ | 24.3 | 16.5 |
| L. Poor segmentation: $\nu = 6$ | 9.0 | 16.5 |
| M. No smoothing | 9.3 | 10.5 |

work, although the differing instrinsic noise levels between cameras may necessitate some changes, particularly to $\Delta_{FG}$. Table 2 shows the results of more conservative settings for $\Delta_{FG}$ and $\nu$.

A second group of parameters centers around the pose library construction and retrieval. To help understand the sensitivity to these parameters, a set of results is presented under numerous variations for the *S4-Combo-4-C1* clip, chosen because its performance is near the mean on both walking and jogging activities. Table 2 summarizes these results. In general, the retrieval design provides multiple paths to the correct pose, aimed at providing redundancy should any single retrieval type fail. Thus eliminating one path may not change the results much unless it happens to prevent the retrieval of a key pose in some frame, in which case a completely different (and incorrect) result may be chosen. Indeed, as shown in Figure 7, the final solutions tend to lie close to either the true solution or its mirror-LOS inversion, which has identical silhouette and similar flow. These represent two local minima for the system, and the gross differences in the numbers of Table 2 depend on the number of frames spent following each one. One conclusion of this experiment is that parameter sensitivity would decrease dramatically with a reliable technique for ruling out the false solution, as all methods then might converge near the true one. Modeling human motion dynamics might provide one way to achieve this.

## 3.5 Local Pose Retrieval

The success of recognition-based motion capture relies on the premise that the pose library contains training data for the target motion. When considering recognition of unrestricted motion, scalability concerns arise because the pose library must include

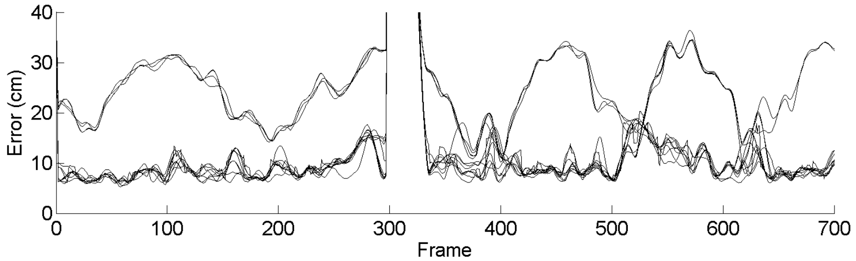

Figure 7: Plot of relative 3D tracking error by frame for the various parameter variations listed in Table 2. For most frames, the solutions lie near one of two local minima: the true solution or the mirror-LOS inverse. (The spike near frame 300 is an artifact of a ground-truth error.)

a vast number of prospective poses. Beyond the logistical challenge of collecting the data (probably surmountable with modern motion-capture technology), enormous pose libraries present computational difficulties as well. Linear search over the entire collection becomes infeasible. Various sublinear methods have been proposed for search and retrieval in high dimensional space, such as hashing and k-d trees, although there exists limited work on applying such techniques to pose tracking [31]. Accordingly, this section describes and evaluates a variant of the baseline pose recognition algorithm that searches only a small subset of the library at every frame but the first, or when recovering from a catastrophic tracking error.

Consider that Viterbi optimization will not normally select any solution where consecutive frames differ too greatly. Thus for frame $i+1$, one need not search the library poses outside some distance ($D_\psi > d_{loc}$) around the pool of candidate poses for frame $i$ because too-distant poses will be rejected by Viterbi in any case. The library itself can hold the means to easily find this neighborhood; simply store for each pose a reference to all other poses within a desired $d_{loc}$ as computed using Equation 7. With this technique there is some theoretical risk that a solution that accidentally leaves the neighborhood of the correct pose will become permanently lost in a bad region of pose space. Although this might be detected by monitoring the frame fitting error $Q_{fr}$ and performing a full (non-local) retrieval to recover, there is absolutely no indication of a drift problem in the experiments. On the contrary, for the two clips that get initialized in an incorrect pose, both the local and the global search converge on the correct pose in approximately the same time.

Table 1 shows in the four right-hand columns the results of pose recognition using $d_{loc} = 12$ (approximately 16 cm). The numbers reveal only slightly worse performance than the results for full global search. Local search yields a mean error over all frames in all clips of just 9.0 cm, compared with 8.9 cm using full search. Despite this, the recognition step only checks around 4000 poses per frame on average, compared with over 60,000 for the full search.

Larger libraries containing many different motion types will cover greater regions of pose space, but should not exhibit great increases in the number of poses within any local neighborhood. Thus exploiting locality appears to offer large computational gains with very little penalty, as evidenced by these experiments. Error recovery may prove more difficult for the local algorithm, but testing this speculation will require more difficult motion sequences where significant errors occur more frequently.

# 4 Conclusion

This paper makes two main contributions toward greater understanding of pose recovery methods. First, it establishes the performance of a fully described recognition-based pose recovery system on the benchmark HumanEva II data, adding to the body of results for these data. Since the results here use relatively straighforward temporal Markov inference, the numbers may perhaps serve as a baseline for more complicated inference methods. Furthermore, results presented here for the local retrieval variant demonstrate its viability for this particular recognition-based approach and add to the limited amount of hard performance data for local methods. Although local search has been used before for body tracking [31], it is arguably not well known given the number of reviewers who cite search scalability as a limitation of recognition-based systems.

The observed accuracy improves on results published in the two HumanEva workshops [27, 6, 16], and a comparable journal publication [29]. The median frame error of 7.9 cm gives qualitatively satisfactory reconstructions and probably suffices for use in some human-computer interaction contexts, including gaming control (in a manner similar to Microsoft's Kinect product), activity recognition, monitoring of the elderly, security, and perhaps physiological studies. It may also prove valuable in analyzing archival footage that lacks markers and multiple viewpoints. Further improvements to these reported error rates appear likely with research into better pose retrieval mechanisms, which play a limiting role. Although the HumanEva data sets currently represent the best widely available test sets for pose recovery, application-specific test cases may become available in the future to better address questions of suitability for a particular task.

An additional contribution of this work lies in its implications for recognition-based pose recovery in general. Recognition represents a low-hurdle approach to human pose recognition, requiring neither camera calibration nor multiple viewpoints, and thus applicable in more casual settings. The results using local retrieval show that an implementation could be made fast, a possible advantage over other methods (although it is hard to be sure since much work in this area does not report computation time). On the other hand, even the best-case retrieval results suggest that recognition-based approaches alone may not suffice for applications needing high-quality motion capture, which requires at least an order of magnitude reduction in the error. Meeting these needs will require an annealing/optimization step that tunes the retrieved poses to more closely match the observations. The experimental error analysis also points to the importance of modeling individual subject limb lengths; not doing so here accounts for the bulk of the error outside that caused by retrieval problems. Although more work remains, particularly on finding the best features for lookup, for simple applications recognition-based methods offer attractive results for pose recovery on monocular image sequences.

## Acknowledgement

Portions of this manuscript are 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. 2

[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. 2

[3] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003. 3

[4] 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. 10

[5] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *International Journal of Computer Vision*, 87(1-2):28–52, March 2010. 2

[6] S. Cheng and M. Trivedi. Articulated body pose estimation from voxel reconstructions using kinematically constrained gaussian mixture models: Algorithm and evaluation. In *EHuM2: 2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation*, 2007. http://www.cs.brown.edu/~ls/ehum2/schedule.html. 14, 20

[7] A. Elgammal and C. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume II, pages 681–688, 2004. 2

[8] 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. 2

[9] 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. 4

[10] N. Howe. Silhouette lookup for automatic pose tracking. In *IEEE Workshop on Articulated and Nonrigid Motion*, 2004. 2, 3, 10, 12

[11] N. Howe. Flow lookup and biological motion perception. In *International Conference on Image Processing*, 2005. 3, 4, 11

[12] N. Howe. Recognition-based motion capture and the humaneva ii test data. In *EHuM2: 2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation*, 2007. http://www.cs.brown.edu/~ls/ehum2/schedule.html. 3, 13

[13] N. Howe and A. Deschamps. Better foreground segmentation through graph cuts. Technical report, Smith College, 2004. http://arxiv.org/abs/cs.CV/0401017. 6

[14] 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. 3, 11, 17

[15] N. R. Howe. Silhouette lookup for monocular 3d pose tracking. *Image and Vision Computing*, 25(3):331–341, March 2006. 2, 9, 11

[16] Z. Husz, A. Wallace, and P. Green. Evaluation of a hierarchical partitioned particle filter with action primitives. In *EHuM2: 2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation*, 2007. `http://www.cs.brown.edu/~ls/ehum2/schedule.html`. 14, 16, 20

[17] 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. 5

[18] E. Krause. *Motion Estimation for Frame-Rate Conversion*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1987. 4

[19] C.-S. Lee and A. Elgammal. Coupled visual and kinematic manifold models for tracking. *International Journal of Computer Vision*, 87(1-2):118–139, March 2010. 2

[20] R. Li, T.-P. Tian, and S. Sclaroff. 3d human motion tracking with a coordinated mixture of factor analyzers. *International Journal of Computer Vision*, 87(1-2):170–190, March 2010. 2

[21] 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. 2

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

[23] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision*, 2002. 2, 3

[24] 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. 2, 3

[25] R. Navaratnam, A. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *iccv*, pages 1–8, 2007. 2

[26] P. Peursum, S. Venkatesh, and G. West. A study on smoothing for particle-filtered 3d human body tracking. *International Journal of Computer Vision*, 87(1-2):53–74, March 2010. 2

[27] R. Poppe. Evaluating example-based pose estimation: Experiments on the humaneva sets. In *EHuM2: 2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation*, 2007. `http://www.cs.brown.edu/~ls/ehum2/schedule.html`. 3, 14, 16, 17, 20

[28] 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. 2

[29] G. Rogez, C. Orrite-Uruñuelaa, and J. Martínez-del Rincón. A spatio-temporal 2d-models framework for human pose recovery in monocular sequences. *Pattern Recognition*, 41(9):2926–2944, 2008. 2, 20

[30] 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. 2

[31] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *International Conference on Computer Vision*, pages 750–757, 2003. 2, 3, 4, 17, 19, 20

[32] H. Sidenbladh, M. J. Black, and D. A. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, pages 702–718, 2000. 2

[33] L. Sigal, A. Balan, and M. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1):4–27, March 2010. 1, 7, 12

[34] L. Sigal and M. Black. Guest editorial: State of the art in image- and video-based human pose and motion estimation. *International Journal of Computer Vision*, 87(1-2):1–3, March 2010. 2

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

[36] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 390–397, 2005. 2

[37] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Learning joint top-down and bottom-up processes for 3d visual inference. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1743–1752, 2006. 2

[38] 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. 5, 6

[39] A. Sundaresan and R. Chellappa. Model driven segmentation and registration of articulating humans in laplacian eigenspace. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):1771–1785, 2008. 2

[40] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 238–245, 2006. 2

[41] X. Zhao and Y. Liu. Generative tracking of 3d human motion by hierarchical annealed genetic algorithm. *Pattern Recognition*, 41(8):2470–2483, 2008. 2

[42] 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. 4