A Robust Likelihood Function for 3D Human Pose Tracking

Weichen Zhang, *Student Member, IEEE,* Lifeng Shang, *Member, IEEE,* Antoni B. Chan, *Member, IEEE,*

***Abstract*—Recent works on 3D human pose tracking using unsupervised methods typically focus on improving the optimization framework to find a better maximum in the likelihood function (i.e., the tracker). In contrast, in this work, we focus on improving the likelihood function, by making it more robust and less ambiguous, thus making the optimization task easier. In particular, we propose an exponential Chamfer distance for model matching that is robust to small pose changes, and a part-based model that is better able to localize partially occluded and overlapping parts. Using a standard annealing particle filter and simple diffusion motion model, the proposed likelihood function obtains significantly lower error than other unsupervised tracking methods on the HumanEva dataset. Noting that the joint system of the tracker's body model is different than the joint system of the mocap ground-truth model, we propose a novel method for transforming between the two joint systems. Applying this bias correction, our part-based likelihood obtains results equivalent to state-of-the-art supervised tracking methods.**

***Index Terms*—Pose estimation, Human Tracking, Exponential Chamfer distance, Part-based model, Joint system correction**

I. Introduction

Multi-view 3D human pose tracking is still a challenging problem in computer vision. The human pose configuration is a high-dimensional state space, and the goal is to recover the pose from a set of images taken from different viewpoints. Typical image features such as silhouettes and edges, are not robust to partial self-occlusions and noises, making occluded and overlapping body parts hard to localize. The motion of a human is complex; limbs move with a wide range of motion and speeds, causing frequent occlusions of body parts. Because of these confounding factors, pose tracking is a high-dimensional optimization problem with multiple local maxima.

To deal with these problems, recent approaches use supervised learning to simplify and constrain the problem. One line of work is to learn strong action-specific motion priors [1–4] that can well predict the next pose from previous poses, thus narrowing the search space and making it easier to recover the optimal solution. Other approaches use action recognition as a prior for poses [5], or learn a low-dimensional state space to better model correlations in the pose configuration [6]. Finally, [7] learns a twin Gaussian process to directly map from image features to the pose.

Weichen Zhang, Lifeng Shang and Antoni B. Chan are with the Dept. of Computer Science, City University of Hong Kong.

This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 110610 and CityU 123212).

Although supervised methods achieve very good tracking performance, they have several limitations. First, the supervised methods are data-driven, and hence will have difficulty recovering poses that are far from those present in the training set. The models must be retrained to recognize new poses or new actions. Second, the quality of tracking depends highly on the quality of the training data, in particular the ground-truth poses. For datasets with noisy or biased ground-truth poses, supervised methods inadvertently learn this bias. For example, the HumanEva dataset [8] defines the ground-truth joints as the mocap markers placed on the surface of the person, and hence there is a systematic bias between the ground-truth joints and the real joints of the person.

In contrast to previous supervised learning approaches, which aim to learn strong motion priors to constrain the search space, in this paper we focus on the unsupervised setting and propose a strong likelihood function that is robust to partial-occlusion and noise. The contributions of this paper are four-fold: 1) we propose distance function between silhouettes, which is robust to small pose changes, and is based on the exponential Chamfer distance; 2) we propose a likelihood function based on part-based matching of silhouettes and edges, which is robust to partial self-occlusions; 3) in experiments, we demonstrate that our proposed part-based likelihood function, coupled with a simple diffusion motion model, achieves state-of-the-art results compared to other unsupervised and supervised pose tracking methods; 4) we propose a method to perform correction between the 3d-body joint system of the tracker and the mocap joint system of the ground-truth – applying the correction uniformly improves the error metrics with respect to the mocap ground truth.

The remainder of this paper is organized as follows. In Section II, we present a detailed literature review on previous work on human pose estimation and tracking. In Section III, we describe our tracking framework, along with the body and motion models. In Section IV, we propose our robust likelihood function, and in Section V we propose a method to estimate the systematic bias of mocap ground truth, thus allowing comparisons between unsupervised and supervised tracking methods. Finally, in Section VI we present experiments using our robust likelihood on the HumanEva dataset.

II. Related work

Early methods used Kalman filters to perform pose tracking [25], but the linear and Gaussian assumptions of the Kalman filter cannot handle multiple solutions well. The multiple hypothesis tracker [10] and particle filter [9] were

TABLE I
PREVIOUS WORK ON UNSUPERVISED METHODS FOR 3D HUMAN POSE TRACKING. FOR "LIKELIHOOD FUNCTION", OVERLAP, DISTANCE OR SSD (SUM-SQUARED DIFFERENCE) DESCRIBES THE RELATIONSHIP BETWEEN THE BODY PROJECTION AND IMAGE FEATURES IN THE LIKELIHOOD FUNCTION.

unsupervised methods					
Ref	body model	image descriptor	likelihood function	motion model	optimization
[9]	2D shape	contour	distance	linear stochastic differential equation	factored sampling
[10]	2D cylinder	silhouette	overlap	Gaussian diffusion	Gauss-Newton method
[11]	2D cylinder	shape and appearance	grayvalue differences and probability of occlusion	constant velocity and learned walking motion	particle filter
[12]	3D kinematic	3D markers projected on 2D image	distance	Gaussian diffusion	hybrid Monte Carlo filter
[13]	3D ellipse	edge and intensity	residual error for visible nodes & constant error otherwise	Gaussian diffusion	scaled covariance sampling
[14]	3D mesh	silhouette and color	distance	reconstruction	stochastic meta descent
[15]	3D ellipse	silhouette and edge	SSD	Gaussian diffusion	annealing particle filter
[16]	2D cylinder	silhouette	Chamfer distance	-	sample from posterior
[17]	2D cylinder	edge + patch	SSD	transition kernel	optimized unscented particle filter
[18]	3D curved surfaces	contour	SSD	extremal-contour point velocity with zero-reference kinematic model	Gauss-Newton method
[19]	3D ellipse	silhouette	distance	-	EM algorithm
[20]	3D truncated cone model	silhouette and edge	SSD	Gaussian diffusion	hierarchical particle swarm optimization
[21]	3D mesh	silhouette and color	Chamfer and Bhattacharya distances	Gaussian diffusion	interacting simulated annealing
[22]	3D mesh	color	Gaussian distribution of model on image	-	gradient descent
[23]	3D cylinder	silhouette and edge	region-specific probability and distance on edge	Gaussian diffusion	importance sampling
[24]	3D mesh	silhouette	overlap	self-trained	branched iterative hierarchical sampling
Ours	3D cylinder	silhouette, edge and color	part-based exponential Chamfer distance	Gaussian diffusion	annealing particle filtering

then proposed to enable both multimodal solutions and non-linear components. Later, [26] proposed a covariance scaled sampling method for prediction of human motion, which is the motion model we use in this paper. Over the past two decades, there have been a large number of papers published on human pose tracking and estimation. [27] gives a good overview of pose tracking works before 2007.

Tables I and II summarizes recent works on human pose tracking and estimation. They can be divided into two groups: unsupervised (generative) approaches and supervised (discriminative) approaches.

A. Unsupervised methods

Unsupervised approaches typically use a Bayesian framework for tracking. Since there is no specific motion prediction model, most works use a Gaussian diffusion model to get the prior pose [12, 13, 15, 21, 23], which is also used by our work. Many previous works use edges and silhouettes as the image descriptor [10, 15, 16], while in recent years, color is used to handle occlusions [21, 24]. In our work, we use color to identify parts, from which we extract edge and silhouette features.

To calculate the likelihood function, there are two typical methods: 1) calculate the Chamfer distance between the projection of the 3d model and the edge image [15, 17, 20]; 2) calculate the region size of the model projection overlapping the silhouettes [10, 15]. [16] used silhouette overlap as the likelihood in their single-view pictorial structures model. To estimate the pose, sample poses were drawn from the posterior distribution, and the sample with smallest Chamfer distance to the binary input image was selected in a *post-processing* step. From [16], the Chamfer distance has been used for silhouettes to localize the model more robustly [21].

Unsupervised methods also focus on improving the tracking/optimization framework. The HumanEva dataset [8] provides an unsupervised baseline algorithm, which uses a likelihood function consisting of bi-directional silhouette and edge matching, and the annealing particle filter (APF) [15] as the tracker. [23] learns a graphical model for the body structure, and performs tracking with importance sampling. [21] proposes interacting simulated annealing (ISA) to help search for the global maximum of the likelihood, and uses a smoothing tracker to achieve state-of-art results on HumanEva-II, but needs the future video frames to run the smoother. [24] proposes a branched iteration hierarchical sampling (BIHS) method to improve the solutions found by the particle filter,

TABLE II
PREVIOUS WORK ON SUPERVISED 3D HUMAN POSE TRACKING. FOR SUPERVISED MOTION MODELS, THE PREVIOUS POSE IS MAPPED TO THE NEXT POSE, AND ADDITIONAL IMAGE FEATURES ARE USED FOR TRACKING. FOR SUPERVISED METHODS THAT MAP FROM IMAGE TO POSE DIRECTLY, THE INPUTS ARE IMAGE FEATURES AND OUTPUTS ARE POSES.

Supervised methods					
Ref	Type	Input	Output	Feature for tracking	Model
[28]	motion model	high-dimensional pose	low-dimensional pose	image intensity	GPLVM
[29]	motion model	shape vector	pose vector	shape context	RVM and SVM
[2]	motion model	training pose	latent position	2D image location of 3D body point	GPDM
[30]	motion model	high-dimensional pose	low-dimensional pose	-	rank prior
[6]	motion model	vector of joint angles	latent space points	silhouette	CMFA-VB
[4]	motion model	pose history	current pose and latent variable	silhouette and edge	CRBM
[31]	image to pose	silhouettes	body pose parameters	-	specialized mapping architecture with EM
[32]	image to pose	silhouettes	pose vector	-	GPLVM
[33]	image to pose	HoG	pose vector		examplar-based
[34]	image to pose	histograms of shape context, silhouette	3D pose state	-	mixtures of experts
[7]	image to pose	HoG or HMAX	3D pose vectors	-	TGP
[5]	image to pose	colour, dense optical flow, spatiotemporal gradients	relative pose feature	-	Hough forest classifier
[35]	image to pose	depth feature	body joint position	-	randomized decision forests
[36]	motion model, image to pose	observation-state pairs	pose state	shape context	Bayesian mixture of experts
[37]	motion model, image to pose	High-dimensional pose and image descriptor	low-dimensional predicted pose, image descriptor	silhouette	LLE, RVM, Binary PCA

while also introducing self-training motion model to improve the speed.

[38] generates a 3D visual hull from multiple-cameras and then finds the best 3d body model. [22] proposes a region-based pose estimation, which finds the best-fitting projected model to the image, and uses 2d-3d correspondences to back-project the optimum to the 3D pose.

In our work we mainly focus on improving the *objective function*, i.e., the likelihood function. In particular, we use color feature to segment the silhouette into parts, and match the silhouette parts to the model parts with the exponential Chamfer distance. In this way, the penalty terms are smoothened and the number of local maxima is reduced, resulting in a more robust and less ambiguous likelihood function, which is easier to optimize using the standard APF.

B. Supervised methods

In recent years, supervised methods have been employed in pose tracking to obtain promising results. The supervised methods can be categorized into two types: prediction from the previous pose (i.e., a motion model), and prediction directly from the image. In the former category, probabilistic latent variable models are widely used for learning motion priors, e.g., GPLVM [1, 39], GPDM [2, 3], and CRBM [4]. In these works, the latent variables represent the relationships between poses in consecutive frames. The motion models are trained on mocap data, and used in tracking to predict the next pose to narrow the search space of the tracker. While they obtain good results, they can only be applied to the specific actions learned.

In contrast to learning a motion model, [7, 29, 40, 41] directly learn a mapping from image descriptors to poses, using a regression function, e.g., the twin Gaussian process (TGP) [7]. Image descriptors, such as silhouettes, edges, segmentations, or HOGs, are used. Action recognition [5, 42] can also serve as a prior on the pose, using the detected action in the image. Alternatively a low-dimensional manifold can be learned to represent the state space [6, 43].

For supervised methods, there are two main concerns. First, the poses and motions that can be estimated are largely dependent on those seen in the training set, and hence these methods do not typically generalize well to unseen test situations (i.e., a training set bias). Second, supervised methods cannot be directly compared to unsupervised methods when the ground-truth annotations have a systematic bias from the actual human pose. For example, in the HumanEva dataset, the ground-truth joint positions are determined by mocap markers that are placed on the surface of the person, whereas the actual joints are inside the person. In other words, there is a systematic offset (bias) between the real joint positions of the person and the ground-truth positions. Hence, the accuracy results (e.g., error to the biased ground-truth data) of supervised methods, which directly map to the biased ground-truth data, are not always comparable with unsupervised methods, which predict real joint positions of the human. In this paper, we propose a method for correcting the ground-truth bias when comparing unsupervised and supervised methods.

C. Single camera methods

Previous work has also focused on 3d human pose tracking using a single camera. However, localizing the global position of the body is difficult, given the single view constraint. Hence, most methods are supervised in order to better learn the 2d to 3d mapping. [7] directly maps from image features to the

positions of joints. [44] uses an exemplar-based method to lift 2D tracklets into 3D. [4] learns a motion model to predict body positions of the next frame. [45] uses one-class SVM to find correct models from a set of ambiguous shapes. [46] tracks the body with at least 2 views, using APF with a quaternion representations of nodes. In our experiments, we also test on single and double view to illustrate the robustness of our likelihood function.

III. Pose tracking framework

In this work, we use a standard online Bayesian tracking framework. Denote x_t as the pose configuration (state) in frame t, y_t as the image observation, $y_{1:t} = [y_1, \cdots, y_t]$ as the sequence of images from time 1 to t, and $\ell = -\log p(y_t|x_t)$ as the negative observation log-likelihood function, or penalty function. Our interest is in the posterior $p(x_t|y_{1:t})$, which recovers the pose at time t from the images seen so far,

$$p(x_t|y_{1:t}) \propto p(y_t|x_t)p(x_t|y_{1:t-1}) \tag{1}$$

where $p(y_t|x_t)$ is the likelihood, which measures how well the image y_t matches the pose x_t. The prediction (prior) is $p(x_t|y_{1:t-1})$, and using a 1st-order Markov assumption,

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{2}$$

where $p(x_t|x_{t-1})$ is the motion model, which predicts the current pose given the previous pose, and $p(x_{t-1}|y_{1:t-1})$ is the posterior at time $t-1$.

Since the posterior is potentially multimodal and the motion model and likelihoods are non-linear and non-Gaussian, we resort to sequential Monte Carlo theory, and approximate the posterior using a set of weighted samples $\{\omega_t^{(i)}, x_t^{(i)}\}_{i=1}^N$, where $x_t^{(i)}$ is the ith sample or particle, and $\omega_t^{(i)}$ is the corresponding weight, calculated from the likelihood of the sample. Similar to previous work [8], we adopt the annealing particle filtering (APF) [15] as the optimization algorithm to determine the optimal particles for the current likelihood function. APF is a layer-based particle filtering framework, where the diffusion width is reduced in each layer, making the samples converge to local maxima of the likelihood. Finally, the estimated state is approximated by the weighted sample mean, $\hat{\mathscr{X}} = \sum_{i=1}^N x_t^{(i)} \omega_t^{(i)}$, where the weight $\omega_t^{(i)}$ is the normalized likelihood of sample $x_t^{(i)}$.

In the remainder of this section we present the human body model (state space) and motion model used by our tracker. Our robust likelihood function is proposed in Section IV.

A. Human body model

We use the body model provided by the HumanEva dataset [8]. The body skeleton is constructed as a 3D kinematic tree, and the limbs are represented as cylinders, which look like rectangles when projected to 2D image space (e.g., top-left of Fig. 1). The body consists of 15 parts: head, torso, pelvis, upper arms, lower arms, hands, upper legs, lower legs, and feet. The hips, shoulders, pelvis, thorax and head are modeled as ball and socket joints (3 DoF), while knees, elbows and the clavicles are allowed 2 DoFs. The ankles and wrists are assumed to be hinge joints with 1 DoF. These DoFs are the relative joint angles to the parent parts. With an additional 3 parameters to represent the global position of the pelvis, the whole human body is modeled by 40 parameters. The lengths, radii of cylinders and joint offsets are provided by the HumanEva dataset, and the shape is fixed during the tracking process.

B. Motion model

In our work, we do not use a trained motion model to predict the pose from the previous state. Using a trained motion model restricts the tracker to only work in specific situations or with specific actions. On the other hand, by using a simple motion model and a strong likelihood function, we obtain a tracker that works on a wider range of action and poses. The motion model that we use is a *simple diffusion process* [15], $x_t = x_{t-1} + \varepsilon_t$, where the noise ε_t is a multivariate Gaussian distribution with zero mean and diagonal covariance. The purpose of the noise is to diffuse the particles to cover more of the search space, where the covariance determines the range of diffusion. Given the previous particles $\{x_{t-1}^{(i)}, \omega_{t-1}^{(i)}\}_{i=1}^N$, we obtained the predicted states $\{\hat{x}_t^{(i)}, \omega_{t-1}^{(i)}\}_{i=1}^N$ using the diffusion process. Finally, after diffusion, the tracker removes particles with impossible pose configurations, according to the HumanEva joint angle limits and penetrating part detections.

IV. Robust part-based likelihood function

The main focus of our work is to construct a robust likelihood function from features extracted from multi-view images. The previous work [8] proposes a baseline likelihood function that uses silhouettes, bi-directional silhouettes and edges. However, in conjunction with APF, the performance is worse than action-specific supervised approaches. Therefore, more robust distance functions and more informative image features are required.

Fig. 1 summarizes our proposed robust part-based likelihood function. In the tracker, we first diffuse the posterior states (particles) into predicted states, and then project the corresponding 3d body parts into the camera views. For each image view, background subtraction is used to detect the foreground silhouette. Next, the silhouette is roughly segmented into body parts using a GMM color model. The negative log-likelihood function is composed of three penalty terms based on the projected body parts and the segmented silhouette: 1) the exponential Chamfer distance between the visible projected parts and the corresponding silhouette segment (ℓ_{pECD}); 2) the fraction of silhouette pixels that are not covering the projected body model (ℓ_b); 3) the exponential Chamfer distance between the visible edges of the projected parts and those of the silhouette segment (ℓ_{edge}).

A. Silhouettes and exponential Chamfer distance

The silhouette is a very important cue for human pose tracking, since it can determine the basic outline of the human and is easily calculated using background subtraction. Almost all works use this feature [5, 8, 21, 23, 24]. In our work, we

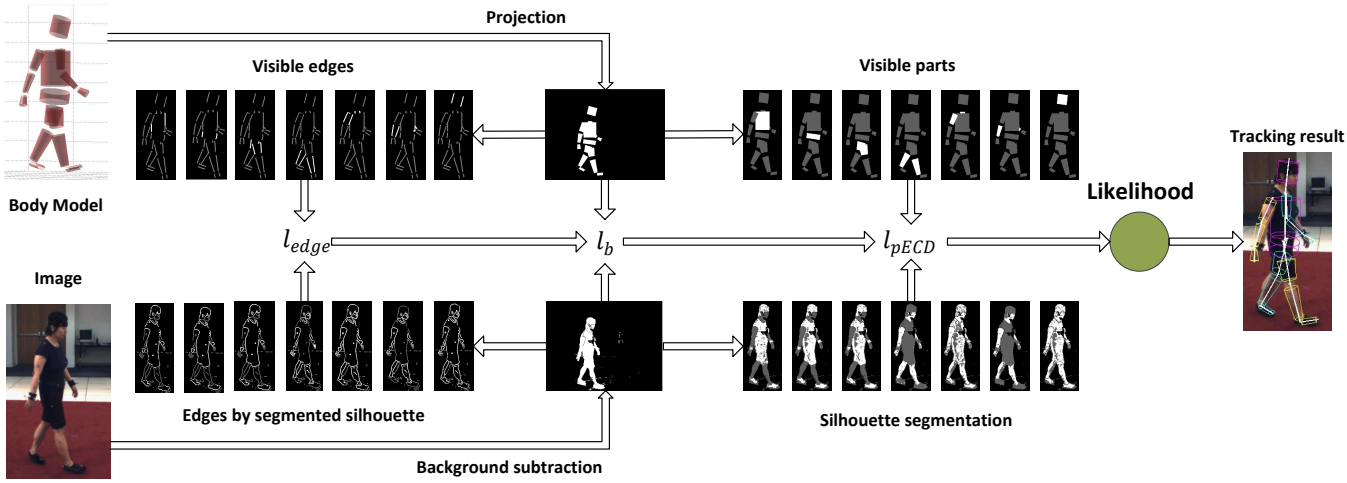

Fig. 1. The framework of the robust part-based likelihood function. The silhouette is roughly segmented into body parts using a GMM color model. The negative log-likelihood function is composed of three penalty terms based on the projected body parts and the segmented silhouette: 1) the exponential Chamfer distance between the visible projected parts and the corresponding silhouette segment (ℓ_{pECD}); 2) the fraction of silhouette pixels that are not covering the projected body model (ℓ_b); 3) the exponential Chamfer distance between the visible edges of the projected parts and those of the silhouette segment (ℓ_{edge}).

obtain the silhouette image using a standard GMM background model provided by HumanEva [8].

Denote the binary silhouette image as $S(i)$, where the pixels are indexed by i, and the foreground and background pixels are set to 1 and 0, respectively. Similarly, denote $P(i)$ as the projection image of the 3D human body model to the current camera view. A standard method for calculating the difference between the projected pose and the silhouette is to measure the fraction of pose pixels that are not in the silhouette [8, 10, 15, 24],

$$\ell_f = \frac{1}{|P|} \sum_{\{i|P(i)=1\}} [1 - S(i)], \tag{3}$$

where $|P|$ is the number of non-zero pixels in P. A disadvantage of the likelihood in (3) is that, in some situations, two candidate poses will have the same ℓ_f value, even though one pose is actually closer to the real pose. An example is given in Fig. 2, where the three poses have the same ℓ_f, but the rightmost pose is probably the best candidate if these are limbs.

To remove the ambiguity of the overlap ratio, we consider using the distance of the pose pixels to the silhouette, rather than just the amount of overlap. In particular, we calculate the Chamfer distance transformation, which is normally applied to edge maps, to the silhouette image $D(i)$. Each value $D(i)$ is the distance of pixel i to the closest foreground pixel in $S(i)$. We then put the distance values through a generalized normal function, $f(x) = \exp(-(\frac{|x|}{\alpha})^{\beta})$, where $\{\alpha, \beta\}$ are parameters. Finally, the *exponential Chamfer distance* (ECD) penalty term is the average distance of all the pose pixels,

$$\ell_{ECD} = \frac{1}{|P|} \sum_{\{i|P(i)=1\}} [1 - f(D(i))]. \tag{4}$$

Fig. 3 plots the individual penalty term $[1 - f(D(i))]$ versus distance $D(i)$. The penalty is near zero for small distances, and gradually increases to 1 as the distance increases. The parameter α controls the width of the zero-region of the penalty function (i.e., the region where non-zero Chamfer distance is given low penalty), while the parameter β controls the sharpness of the transition (see Fig. 3). For comparison, ℓ_f is equivalent to setting $f(x)$ to a step function at 1 (i.e., the minimum non-zero Chamfer distance)[1], but yields a non-smooth likelihood function. When $f(x) = 1 - x$, i.e., a linear function, the corresponding penalty function directly uses the Chamfer distance, similar to [16] (denoted as linear Chamfer distance, LCD). For LCD, the penalty for poorly matched pixels increases without limit, which makes the likelihood function peakier and leads to degenerate particles in APF.

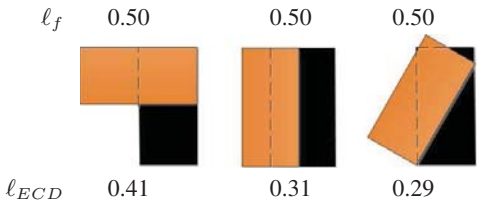

Fig. 2. Example of three poses with the same silhouette-overlap ℓ_f: the black rectangle is the image silhouette, and orange rectangle is the projected model. Using our proposed ECD removes the ambiguity and gives lowest penalty ℓ_{ECD} to the rightmost pose.

Our proposed penalty function in (4) has several important properties that improve its robustness. First, the penalty for a pose pixel that is not in the silhouette is based on its distance to the silhouette. Hence, even if two poses have the same amount of pixels overlapping the silhouette, ℓ_{ECD} will be lower for the pose with non-overlapping pixels that are closer to the silhouette. This is demonstrated in Fig. 2, where the rightmost pose has the lowest ℓ_{ECD}, and the leftmost has the highest. Second, when the projected pose is larger than the silhouette (e.g., due to background subtraction errors), the lowest ℓ_{ECD}

[1]Setting $\alpha = 1$ and $\beta \to \infty$ for ECD yields the step function for ℓ_f.

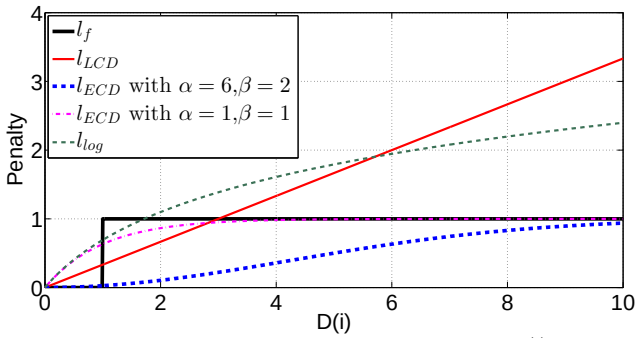

Fig. 3. Plots of penalty functions versus Chamfer distance $D(i)$ for different transformations.

pose is centered over the silhouette, which is illustrated in Fig. 4a. In contrast, ℓ_f will be 0 for any translation of the pose that covers the silhouette, which causes a ridge in the likelihood function with multiple optimal solutions. Third, when the silhouette and projected model have similar shapes, the ℓ_{ECD} is robust to small deviations (translations and rotations) of the projected model. This is illustrated in Fig. 4b, where a small change in rotation from the best pose maintains a similar ℓ_{ECD}. Compared to the ℓ_f, the first two properties of ℓ_{ECD} reduce the number of local maxima and smoothens the likelihood function, by removing poses with equal penalties. The third property adds robustness to small deviations in rotation and translations, which helps APF find the local maxima easier (e.g., a particle is selected if it is near enough to a maximum).

B. Part-based silhouette likelihood

One problem with matching the whole body model to the whole silhouette is that it is difficult to localize the parts inside the silhouette. While multiple views help to reduce the ambiguity, usually the problem persists, especially when the part is both occluded in one view, and inside the silhouette in another view (e.g., the arms in Fig. 7). To better locate these parts, we propose a part-based silhouette likelihood function, where each part in the body model is compared to the corresponding part segment in the silhouette.

Although there are several methods to obtain an appearance model of each limb from videos [47, 48], in our setting, the person silhouette is available from background subtraction. Hence, a simple GMM color model suffices to model each part in the silhouette. First, the silhouette is roughly segmented into its corresponding parts using color cues. Using the initial pose, a GMM color model with 3 mixture components is learned for each body part, from the corresponding pixels in all image views. The color features used are hue and saturation only. In the following frames, each GMM is used to segment its part from the silhouette using the color images. An example silhouette segmentation is shown in Fig. 5.

Given the segmented silhouette, the *visible* portion of each projected body part is compared with its corresponding segment. Since the hidden (occluded) portion of the projected body part is not expected to be seen in the image, it should not be used to influence the match. Fig. 6 illustrates part-based matching using hidden part removal for the torso and lower

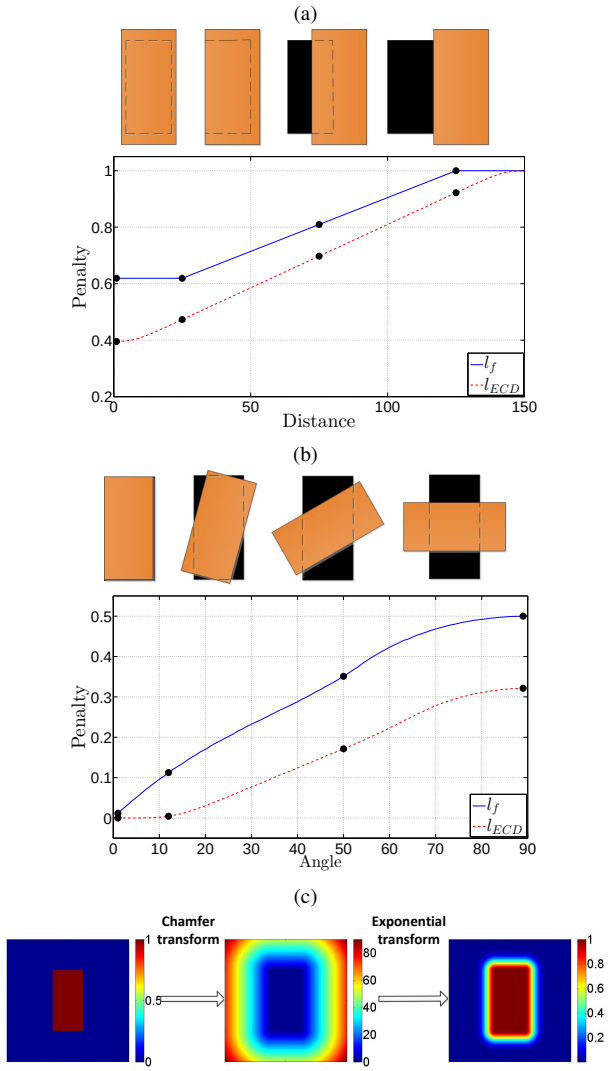

Fig. 4. Comparison of ℓ_{ECD} and ℓ_f: the black rectangle is the silhouette and the orange rectangle is the projected model. (a) when the projected model is larger than the silhouette, the pose with the lowest ℓ_{ECD} is centered over the silhouette; (b) ℓ_{ECD} is robust to small changes in rotation from the true pose. (c) the image transformation from silhouette to Chamfer distance map, and then to exponential Chamfer distance map.

arm parts. In both cases, a portion of the projected part is occluded. However, the visible part matches well to a portion of the silhouette segment.

If not applying ECD, the part-based silhouette likelihood function is,

$$\ell_p = \frac{1}{\sum_j |P_j|} \sum_j \sum_{\{i|P_j(i)=1\}} [1 - S_j(i)], \tag{5}$$

where P_j is the visible portion of the jth projected body part, and S_j is the silhouette segment for the jth part. Finally, when applying ECD, the part-based silhouette likelihood is defined as the sum of the ECD between each visible part and its silhouette segment,

$$\ell_{pECD} = \frac{1}{\sum_j |P_j|} \sum_j \sum_{\{i|P_j(i)=1\}} [1 - f(D_j(i))], \tag{6}$$

where D_j is the Chamfer distance transform of the silhouette segment for the jth part.

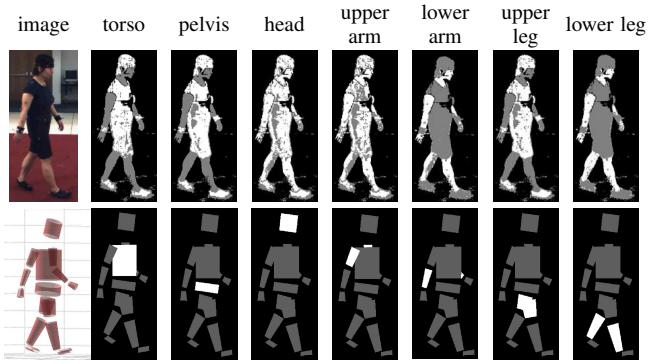

Fig. 5. (top) silhouette part segmentation; (bottom) model visible part segments. White is the part segment, and gray is the silhouette or model projection.

Note that the segmentation for each part is quite rough and contains pixels from other parts with similar colors (e.g., lower arm segment contains lower leg). However, our purpose here is not to obtain a perfect segmentation, but to reduce the ambiguity of the parts inside the silhouette. For those parts with distinct color that can be well-segmented (e.g., the lower arms), the segmentation helps to focus the part-based silhouette match onto particularly salient parts. If the part cannot be segmented well due to neighboring parts having similar colors, the segmented silhouette will contain the part along with neighboring parts. Hence, the corresponding body part is still matched to a subset of the original silhouette, and the search space for the part is still reduced (e.g., see Fig. 5). In the extreme case when all parts have the same color (e.g., all black) and part-segmentation is not possible, then each part will be matched against the whole silhouette, which is equivalent to performing matching between whole silhouettes. Hence, when the part segmentation fails, the part-based model essentially reverts to the whole silhouette matching.

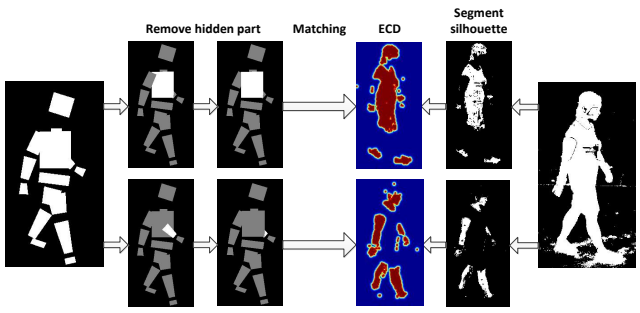

Fig. 6. Part-based silhouette matching: the visible portions of the projected parts are matched to the corresponding silhouette segment.

Fig. 7 compares the different silhouette-based likelihood functions. Using only silhouette overlap ℓ_f, the tracker crosses the two legs (Fig. 7, green arrow) because of a bad local minimum in the penalty function. Using the exponential Chamfer distance ℓ_{ECD} corrects the pose of the legs, but the arms still are not localized well (Fig. 7, red arrow) because they are inside the silhouette. Finally, using the part-based ECD ℓ_{pECD} fits the arms to the correct pose.

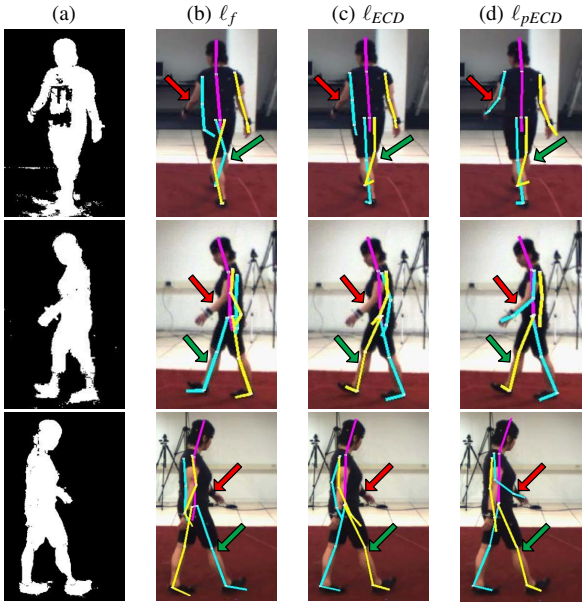

Fig. 7. Comparison of tracking with silhouette-based likelihoods: a) silhouette images, and tracking result using b) silhouette overlap, c) silhouette ECD, and d) part-based silhouette ECD. The body-overlap ℓ_b term is used for all. The video is HumanEva-I S1 Walking Train, and each row is a camera view.

C. Bi-directional silhouette

The penalty functions in the previous subsection are minimized when the projected body parts are covered by the silhouette. However, any parts inside the silhouette have a penalty of zero, so multiple optimal configurations are possible, some of which do not utilize the silhouette completely. To address this problem, [8, 49] propose an additional penalty term that measures how well the silhouette covers the projected model. Specifically, the fraction of silhouette pixels that are not in the projected model is calculated by swapping the projection and silhouette in (3),

$$\ell_b = \frac{1}{|S|} \sum_{\{i|S(i)=1\}} [1 - P(i)]. \tag{7}$$

[8, 49] combine (7) and (3) to obtain a bi-directional silhouette likelihood. In our work, we also form a bi-directional likelihood, where the forward term uses the part-based ECD ℓ_{ECD}, while the backward term uses the full body-overlap penalty ℓ_b.

We also considered using the ECD for the backward term. In this case, the ECD needs to be calculated for each candidate pose (i.e., each particle in the APF), and thus the computational complexity is high. However, the accuracy only increased slightly in preliminary experiments, and so did not justify the large increase in computational cost.

D. Part-based edge likelihood

The segmented silhouettes cannot reduce the ambiguity when two parts have similar colors. In this case, edges can help to reduce ambiguity and better localize the parts [23]. The Canny edge detector [50] is applied to the image. The standard edge likelihood function (without using part-based

segmentation) is

$$\ell_{EDGE} = \frac{1}{|E|} \sum_{\{i|E(i)=1\}} [1 - f(M(i))], \tag{8}$$

where E is the model projected edges, M is the Chamfer distance transform map of image edges.

For the part-based edge likelihood, the silhouette segments are used to obtain the edge images for each segment, and the Chamfer distance transform is calculated, resulting in the part-based edge map M_j. For the body model, the *visible* edges of the projected part form the projected part-edge image E_j. Finally, the two sets of edges for each part are compared using the ECD,

$$\ell_{pEDGE} = \frac{1}{\sum_j |E_j|} \sum_j \sum_{\{i|E_j(i)=1\}} [1 - f(M_j(i))]. \tag{9}$$

E. Combined likelihood function

The part-based silhouette and edge likelihoods and the body-overlap measure the fitness between the camera images and a candidate human pose. The three terms are combined to form the observation likelihood of the image, given the pose state,

$$-\log p(y_t|x_t) \propto \gamma_1 \ell_{pECD} + \gamma_2 \ell_b + \gamma_3 \ell_{pEDGE}, \tag{10}$$

where $\gamma = [\gamma_1, \gamma_2, \gamma_3]$ is a vector of weights. Finally, (10) is calculated for each camera view, and then summed together to obtain the likelihood function for the APF tracker.

V. Joint System Correction Using GP Regression

The widely used measurement to compare tracking methods is the mean error between the ground-truth and the tracked joint positions. The ground-truth annotations of the HumanEva dataset are obtained using a mocap system. The ground-truth joint positions are defined as the mocap marker positions, which are placed on the surface of the person (e.g., the outside of the elbow), whereas the real joint positions are inside the person (e.g., the actual elbow joint). For unsupervised methods (e.g., our tracking system), the definition of joint positions are based on the joints of the human model (e.g., the top centers of cylinder limbs), and are more similar to the real joint positions. Hence, when evaluating against the mocap ground-truth, unsupervised methods will appear to have larger error even though they may track the real joint positions well. These differences are illustrated in Fig. 8b. The mocap pose contains small errors compared to the actual pose (e.g., the thorax joint is outside the body). The ECPBL tracking result in Fig. 8a recovers the actual pose well, but nonetheless will have large error compared to the mocap ground-truth data. Furthermore, the lengths of the ground-truth limbs can change, whereas the model limbs are fixed lengths. Since the ground-truth joint annotations do not match the real human joints, we call this a "biased" ground-truth. The biased ground-truth makes the comparison between supervised methods and unsupervised methods unfair, since supervised methods directly map to the biased ground-truth joints, while unsupervised methods track the real joint positions. To compensate for the ground-truth bias, we introduce a joint system correction for unsupervised tracking results, in order to fairly compare them with supervised methods.

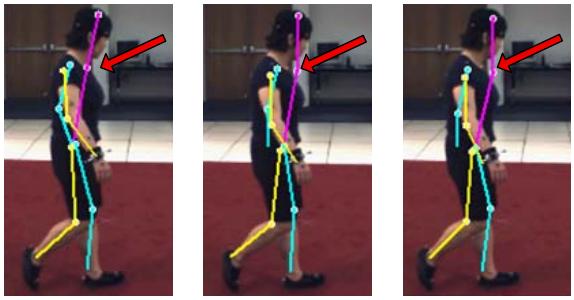

Fig. 8. Example differences between the tracker body joint system and mocap joint system, and pose correction using GP regression.

We propose to compensate for the systematic differences between the two joint systems, by using Gaussian process (GP) regression [51], a nonparametric Bayesian method that can robustly learn from small training sets. In similar work, [52] corrects a Kinect skeleton using cascade regression to map an estimated pose to the ground truth.

To perform the correction between the tracked joint system and the mocap joint system, we learn a GP that predicts the offset between the tracked joint position and corresponding mocap joint position. A GP is learned for each coordinate of each joint independently (45 GPs total). Using the tracked joint $\mathbf{x}_{jo}$ (e.g. elbow) as the reference point, each neighboring joint $\mathbf{x}_j$ (e.g., the shoulder and wrist) is mapped to the sphere $\hat{\mathbf{x}}_j = \frac{\mathbf{x}_j - \mathbf{x}_{jo}}{||\mathbf{x}_j - \mathbf{x}_{jo}||}$. The input vector of the GP function is the concatenation of the mapped neighborhood joints $\hat{\mathbf{x}}_j$ of the tracked model. The corresponding output value is the offset between the tracked joint and the mocap joint (ground-truth).

Once the GPs are learned, the tracking result can be transformed into the mocap joint system, by calculating the input vectors, predicting the joint offsets using the GPs, and performing the correction. Since the GP inputs only depend on the relative positions of the joints, it can be applied to any tracked person, regardless of absolute location. Note that we intentionally use a simple GP formulation, since we are only interested in modeling the systematic bias in the joint positions, and want to avoid overfitting problem when using more complex models.

Fig. 8 shows an example of pose correction, where Fig. 8a is the tracking result, Fig. 8b is the mocap ground truth, and Fig. 8c is the corrected pose. The red arrows indicates the thorax joint, which has a relatively large bias in the mocap ground truth (it is biased forwards). After GP regression, the tracked result better matches the biased ground-truth.

VI. Experiments

In this section, we present experiments testing our proposed likelihood function on the HumanEva datasets: 1) 3 subjects and 5 motions from HumanEva-I (3 color views); 2) the combo video of HumanEva-II (4 color views). Finally, we present experiments on pose estimation using a single and double view from HumanEva-I.

A. Experiment setup

We denote our part-based likelihood with exponential Chamfer distance as ECPBL. The parameters of the likelihood model were selected to minimize the average error on the *Train* videos of HumanEva-I S1 (walking, jogging, gesture and boxing). In particular, the optimal ECD parameters were $\{\alpha,\beta\} = \{6,2\}$, and the likelihood weights were $\gamma = [0.4, 0.3, 0.3]$ (see next subsection more details). After training, the likelihood parameters were fixed throughout the experiment. The APF tracker uses 200 particles and 5 layers, and the covariance of motion diffusion model is set using the HumeanEva baseline program [8]. The tracker was initialized using the mocap data of the first frame, as in [4, 8].

To compare likelihood functions, we use two baseline likelihoods from [8]: silhouette and edge overlap (SE) and bi-directional silhouette and visible edge (BiSE). Both use (3) and (7) to calculate silhouette overlap and bi-directional silhouette overlap. All baselines use the same APF and initialization as ECPBL.

Tracking algorithms were tested on the validation sequences in HumanEva-I. Tracking results were evaluated using the metric proposed in [8], which computes the Euclidean distance (in mm) between the tracked pose and the mocap ground-truth pose, averaged over 15 virtual markers. All results are averaged over 3 trials, using different random seeds for each trial, in order to test the robustness to different instantiations of APF. We also report the overall error, which is averaged over all subjects and motions. On HumanEva-II, we follow common practice [8, 21], and evaluate tracking on frames 2 to 296 and 335 to 1258, which excludes the bad ground-truth data. Video results are presented in the supplemental.

B. Parameter selection

We first present results on parameter selection and its effect on the likelihood function. Using the Chamfer distance on the silhouette helps to better localize and align the limbs. We compare different transformations of the Chamfer distance on the HumanEva-I S1 *Train* videos. The results are presented in Table III. First, the selection of ECD parameters has a large effect on the accuracy, with errors ranging from 69.7 to 50.7. The best performance is obtained with $\{\alpha,\beta\} = \{6,2\}$. From Fig. 3, when $\{\alpha,\beta\} = \{6,2\}$, the penalty function varies smoothly over Chamfer distances between 0 and 10 pixels. Note that it gives less penalty in the range from 0 to 5 pixels, and hence the tracker is tolerant to misalignments within this range. Second, compared to the unbounded penalty functions, x and $\log(1+x)$, the ECD has fewer errors on complex motions such as walking, jogging, and boxing.

Next we consider the effect of changing the likelihood term weights γ, which influence how the samples are propagated to the next layer in the APF. We first set the weights to the same value, and then test on increasing/decreasing the influence of each term. The tracking errors for different weights are shown in Table IV. The choice of likelihood weights also has large effect on the error rate, which ranged from 64.4 to 50.7. The best performance results from increasing the weight of only the part-based ECD, $\gamma = [0.4, 0.3, 0.3]$. Amplifying the weight of the second or third likelihood terms decreased the robustness.

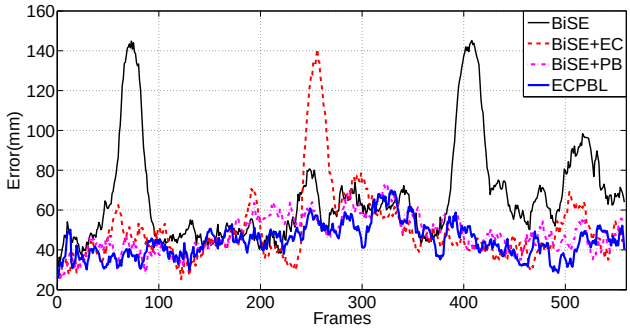

Fig. 9. Tracking error over time of our ECBPL versus the baseline method BiSE with exponential Chamfer distance (EC) and part-based model (PB) on HumanEva-I S1 walking validation video.

Finally, we consider the selection of the best combination of base features, bi-directional silhouette (BiS) or bi-directional silhouette with edges (BiSE), and elements in our proposed ECPBL, the exponential chamfer distance (EC) and the part-based model (PB). The results using various likelihood combinations on S1 Train videos are shown in Table V (top). The results are averaged over 3 trials. For both BiS and BiSE, using the combination of EC and PB components decreases the mean error and standard deviation, compared to other variants. The usage of edge features (BiSE) also shows slightly better tracking performance than without using edge features (BiS). Therefore, we select BiSE with EC and PB as our ECPBL likelihood function.

C. Baseline comparisons

We next evaluate how the baseline likelihood functions BiS and BiSE are improved when adding different elements from our proposed ECPBL, the exponential chamfer distance (EC) and the part-based model (PB). Table V (bottom) presents the tracking results on HumanEva-I Validation videos, averaged over 3 trials. Compared with the original baselines (BiS and BiSE), using the exponential chamfer distance (EC) or parts-based matching (PB) improves the overall performance. Looking at BiSE, the overall error drops about 5% (4mm) when using EC or 16% (13mm) when using PB. Using EC and PB together (i.e., ECPBL) further improves the performance, yielding an overall decrease in error of 26% (22mm). A similar trend is observed when using EC and PB with the BiS baseline (overall decrease of 22% (19mm) error). Also note that the standard deviation of the error drops significantly (58.6%, 14mm), which indicates that the ECPBL tracking results are more stable. The utilization of PB helps to better localize overlapping limbs. In the example in Fig. 10, the ECPBL distinguishes arms from the silhouettes, while BiSE misses arms when they are overlapped with the torso.

Next, we consider the effect of adding noise to the image to test the robustness of the likelihood function. To simulate a noisy background model, we add zero mean Gaussian noise to the image when computing silhouettes, resulting in noisy silhouettes used for tracking. Fig. 11 plots the mean error for ECPBL, BiS, and BiSE for different levels of added noise.

TABLE III
AVERAGE TRACKING ERROR (MM) WHEN USING DIFFERENT TRANSFORMATIONS OF CHAMFER DISTANCE x ON HUMANEVA-I S1 TRAIN VIDEOS. THE LIKELIHOOD WEIGHTS WERE $\gamma = [0.4, 0.3, 0.3]$. THE ECD PARAMETERS ARE $\{\alpha, \beta\}$. THE STANDARD DEVIATION OF THE TRACKING ERROR IS IN PARENTHESIS.

	ECD penalty: $1-\exp(-(\lvert x\rvert/\alpha)^{\beta})$				Unbounded penalty functions	
	$\{1,1\}$	$\{6,2\}$	$\{4,2\}$	$\{2,4\}$	x	$\log(1+x)$
Walking	57.3(16.6)	**42.4**(6.2)	55.6(8.7)	51.8(16.8)	164.8(78.2)	49.8(6.0)
Jogging	71.9(11.7)	**54.1**(6.1)	67.7(11.5)	72.3(11.5)	227.9(87.0)	59.6(5.9)
Gesture	53.9(4.4)	46.4(3.2)	54.7(4.8)	47.3(3.0)	47.5(2.6)	**45.7**(2.5)
Boxing	95.8(23.4)	**59.8**(6.7)	85.3(17.6)	65.9(10.4)	69.5(15.0)	63.6(11.0)
overall	69.7(14.0)	**50.7**(5.6)	65.8(10.7)	59.3(10.4)	127.4(45.7)	54.7(6.4)

TABLE IV
AVERAGE TRACKING ERROR (MM) USING DIFFERENT LIKELIHOOD WEIGHTS γ ON HUMANEVA-I S1 TRAIN VIDEOS WITH ECD TRANSFORMATION $\{\alpha, \beta\} = \{6, 2\}$. THE STANDARD DEVIATION OF THE TRACKING ERROR IS IN PARENTHESIS.

γ	[0.3, 0.3, 0.3]	[0.6, 0.2, 0.2]	[0.2, 0.6, 0.2]	[0.2, 0.2, 0.6]	[0.4, 0.3, 0.3]	[0.5, 0.25, 0.25]
Walking	49.3(6.5)	51.1(16.3)	62.6(29.0)	48.9(7.6)	**42.4**(6.2)	49.1(8.2)
Jogging	58.4(5.3)	57.1(7.1)	66.8(20.1)	59.1(7.5)	54.1(6.1)	**54.1**(4.4)
Gesture	47.1(2.4)	**45.8**(2.8)	46.0(2.9)	49.3(3.1)	46.4(3.2)	46.4(2.7)
Boxing	63.7(9.8)	60.2(10.0)	82.2(26.2)	65.3(9.5)	**59.8**(6.7)	66.3(11.5)
overall	54.6(6.0)	53.6(9.1)	64.4(19.6)	55.7(6.9)	**50.7**(5.6)	54.0(6.7)

TABLE V
AVERAGE TRACKING ERROR ON HUMANEVA-I TRAIN AND VALIDATION VIDEOS FOR BASELINE METHODS USING DIFFERENT ELEMENTS FROM ECPBL, EXPONENTIAL CHAMFER DISTANCE (EC) AND PART-BASED MATCHING (PB). OUR ECPBL IS THE SAME AS BISE+EC+PB. THE SECOND ROW SHOWS THE TERMS (EQUATION NUMBERS) USED IN EACH LIKELIHOOD FUNCTION. THE STANDARD DEVIATION OF THE TRACKING ERROR IS IN PARENTHESIS.

			BiS	BiS +EC	BiS +PB	BiS +EC+PB	BiSE	BiSE +EC	BiSE +PB	BiSE +EC+PB
		Likelihood	(3)+(7)	(4)+(7)	(5)+(7)	(6)+(7)	(3)+(7)+(8)	(4)+(7)+(8)	(5)+(7)+(9)	(10)
Train	S1	Walking	58.3(18.1)	60.8(23.5)	51.5(6.9)	44.6(7.7)	53.5(20.8)	49.0(15.4)	46.2(7.9)	**42.4**(6.2)
		Jogging	69.3(23.5)	64.6(21.2)	60.9(9.0)	57.4(6.6)	68.9(24.1)	67.7(21.2)	60.7(5.8)	**54.1**(6.1)
		Gesture	71.0(12.3)	55.5(7.5)	52.5(4.1)	**44.5**(2.3)	52.5(10.2)	53.9(4.5)	52.5(4.0)	46.4(3.2)
		Boxing	69.7(14.1)	68.1(13.3)	60.8(11.2)	61.9(15.5)	64.8(15.2)	61.5(10.1)	60.8(8.2)	**59.8**(6.7)
		Overall	67.1(17.0)	62.3(16.4)	56.4(7.8)	52.1(8.0)	59.9(17.6)	58.0(12.8)	55.1(6.5)	**50.7**(5.6)
Validation	S1	Walking	58.7(19.4)	61.2(24.4)	53.7(16.2)	45.9(10.5)	56.6(20.6)	53.8(19.8)	49.2(6.8)	**44.3**(8.2)
		Jogging	68.3(18.6)	65.4(20.1)	60.8(14.3)	58.0(8.4)	68.7(20.1)	65.7(19.2)	60.4(9.2)	**55.4**(9.9)
		Gesture	59.1(7.9)	58.9(8.9)	49.6(2.6)	**48.3**(2.5)	53.2(3.0)	51.4(2.8)	50.6(2.8)	48.9(2.3)
		Boxing	83.0(18.4)	77.7(19.7)	62.0(5.7)	61.4(5.1)	74.1(18.2)	73.5(21.1)	62.9(7.3)	**60.6**(7.0)
	S2	Walking	88.5(44.4)	88.5(36.0)	66.9(28.5)	63.8(18.9)	75.2(24.7)	68.0(23.3)	64.4(12.5)	**58.4**(9.2)
		Jogging	79.2(15.6)	81.2(15.9)	76.0(14.3)	71.0(14.8)	71.5(13.5)	70.1(13.3)	74.9(14.7)	**68.2**(10.5)
		ThrowCatch	92.8(21.1)	83.7(18.4)	65.9(10.5)	62.0(7.5)	95.9(26.3)	85.0(22.1)	73.3(19.8)	**57.9**(7.0)
		Gesture	76.0(16.6)	68.3(16.2)	69.3(15.2)	65.7(14.7)	74.8(19.7)	67.7(13.4)	62.0(7.5)	**59.4**(2.6)
		Boxing	129.9(49.2)	122.5(27.7)	88.1(16.7)	90.8(13.6)	135.6(52.9)	133.4(27.0)	90.0(17.3)	**80.2**(11.4)
	S3	Walking	85.4(20.7)	89.2(29.9)	74.0(9.1)	67.9(11.5)	85.3(23.2)	81.0(23.4)	71.8(8.2)	**66.0**(9.0)
		Jogging	90.0(17.3)	91.6(20.0)	89.5(20.2)	80.4(18.7)	87.9(17.3)	85.4(15.6)	82.6(14.3)	**57.5**(8.8)
		Gesture	64.0(11.2)	63.1(8.6)	55.7(6.4)	55.0(4.1)	57.5(10.2)	58.9(8.6)	51.1(4.2)	**50.2**(4.2)
		Boxing	138.4(60.4)	132.0(53.8)	135.4(53.6)	104.6(30.6)	149.9(60.6)	136.1(52.2)	122.2(42.4)	**98.4**(39.1)
		Overall	85.6(24.7)	83.3(23.0)	72.8(16.4)	66.6(12.4)	83.7(23.9)	79.3(20.1)	70.6(12.9)	**62.0**(9.9)

As the level of Gaussian noise increases, the mean error for BiS and BiSE increases by about 12mm, compared to when no noise is added. In contrast, the mean error for ECPBL increases by about 5mm when noise is added. These results suggest that our ECPBL likelihood function is more robust to silhouette noise (i.e., poor background modeling), due to the smoother likelihood of ECD and the removal of local maxima using part-based matching.

D. Results on HumanEva-I

The complete tracking results for the three subjects (S1, S2, and S3) on HumanEva-I are presented in Table VI. Compared with the traditional baseline likelihoods, our part-based likelihood (ECPBL) achieves the lowest overall error of 62.0 versus 75.7 and 92.7 for the baselines. ECPBL also has lower error than the loose-limb method [23] (average error of 52.7 vs 70.7 on 3 videos). In addition, the standard deviation of the error also decreases significantly on many videos. This suggests that the proposed likelihood is significantly more

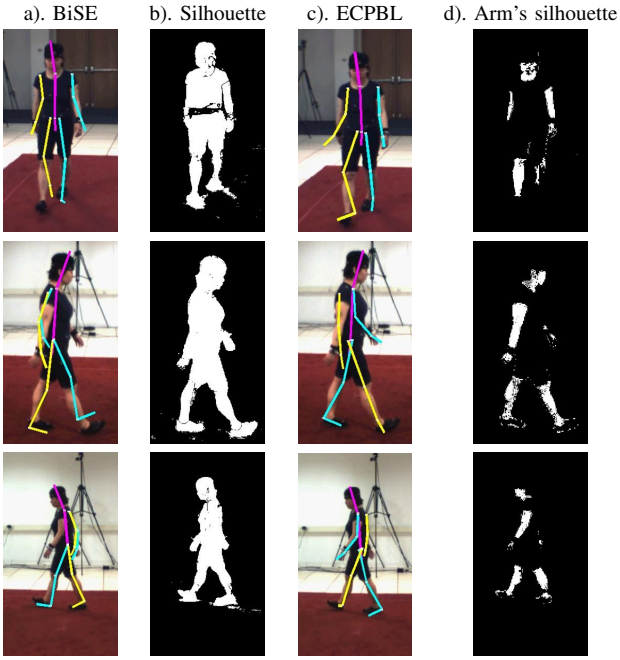

Fig. 10. A comparison of BiSE and ECPBL tracking results (frame 506 of S1-Walking): a) The results for BiSE, and b) the corresponding silhouettes used. c) The results for ECPBL, and d) the corresponding segmented silhouettes of the arm part. ECPBL better localizes the arm when it is overlapping the torso.

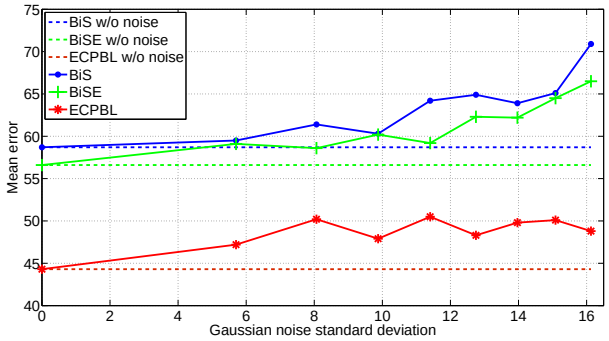

Fig. 11. The mean error with standard deviation versus the standard deviation of Gaussian noise added to images. The dashed lines show the performance of each method without noise added.

robust than the other likelihoods, and does not lose track as often. Plots of the tracking error over time in Fig. 12 confirm that ECPBL is more robust.

Next we applied the joint system correction from Section V. The GPs were trained using the tracking results on the training videos and the corresponding mocap ground-truth data. The learned GPs were then used on the tracking results of the validation sequences to correct for the mocap joint system bias. The resulting errors are shown in Table VI under ECPBL (joint corr.). Using the joint system correction, the overall error reduces to 44.6, an average reduction of 28%. The error reduction is consistent, and ranges from 6% to 79%, with the most improvement in the gesture and walking actions. The fact that there is consistent improvement suggests that indeed there is a systematic difference between the tracker joint system and mocap joint system. If this were not the case, we would expect inconsistent improvement, or no improvement at all. Note that this partially explains why supervised methods, i.e., those that map directly to the mocap pose (e.g. TGP), tend to have much lower error than unsupervised methods – such methods automatically correct for the joint system bias.

Compared with the supervised tracking methods, ECPBL with joint correction has equivalent results to the current state-of-the-art supervised method, TGPKNN. The overall error rate of ECPBL is slightly lower than TGPKNN (44.6 vs 45.6), and has lowest error on 6 out of 13 of the videos. ECPBL also achieves lower error than the other supervised methods (CRBM, GPLVM, and CMFA-VB), which learn strong motion priors or state spaces.

Finally, Fig. 13 shows the error reduction for individual joints when using the joint system correction. The most improvement in error is seen on the body (inner) joints (e.g., thorax, pelvis, hips, and shoulders), while in contrast, the reduction of error was less for the terminal joints (e.g., ankles, wrists). Averaging over all subjects and all motions, the errors of body joints were reduced by 38%, while those of the terminal joints were reduced by 19%. The relative positions of the inner joints are stable (e.g., in the walking motion), irrespective of the mocap or tracking joint system. Hence, the GP regression can effectively map between the joints in the two systems. On the other hand, the motion of the terminal joints is more complex and with a larger range, while also suffering from lower tracking quality. Thus, the GP regression is not able to find a stable mapping between the two systems. Fig. 14 shows another example of the systematic bias of the mocap ground truth. The tracking result matches well the image, with the elbow and thorax positioned inside the body (red circles). However, in the mocap ground-truth, the elbow and thorax are placed on the outside of the body. This is the systematic bias of the mocap ground-truth, whose influence can be removed after applying the GP joint correction (right column).

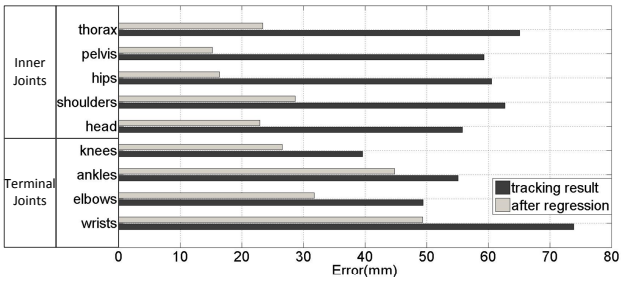

Fig. 13. Average tracking error (mm) for individual joints before and after joint system correction. The test video is HumanEva-I S2 Walking.

E. Results on HumanEva-II

The results on HumanEva-II (S4) are presented in Table VII. The error for ECPBL is 56.8, which is better than the BiS baseline (avg. error of 80). Compared to the baseline, ECPBL performs better on the jogging and balancing sequences (see Fig. 15). The most similar performing method is BISH [24], which is an unsupervised method that improves the optimization stage of the tracker. [24] also finds a systematic

TABLE VI
AVERAGE TRACKING ERROR (MM) ON THE HUMANEVA-I DATASET FOR UNSUPERVISED AND SUPERVISED METHODS. THE STANDARD DEVIATION OF THE TRACKING ERROR IS IN PARENTHESIS.

		Unsupervised methods				Supervised methods				
		SE	BiSE* [8]	Loose-limb [23]	**ECPBL**	**ECPBL** (joint corr.)	CRBM [4]	GPLVM [6]	CMFA-VB [6]	TGPKNN [7]
S1	walking	85.3(48.1)	61.2(21.3)	66.0(19.0)	**44.3**(8.2)	**27.5**(5.1)	48.8(3.7)	-	-	28.3(8.2)
	jogging	73.6(17.6)	63.6(22.4)	77.0(20.2)	**55.4**(9.9)	50.2(10.3)	-	-	-	**37.6**(19.8)
	gesture	65.1(4.4)	54.6(3.6)	-	**48.9**(2.3)	**10.1**(2.3)	-	-	-	19.1(3.5)
	boxing	81.2(17.1)	82.4(21.3)	-	**60.6**(7.0)	**49.7**(10.7)	75.4(9.7)	-	-	56.9(15.8)
S2	walking	76.1(25.4)	69.3(25.3)	69.0(18.8)	**58.4**(9.2)	30.8(6.1)	47.4(2.9)	88.4(25.7)	68.7(24.7)	**28.0**(12.8)
	jogging	81.4(15.2)	74.3(12.6)	-	**68.2**(10.5)	43.3(6.9)	-	91.7(26.0)	72.1(54.7)	**28.6**(6.0)
	throw-c	76.5(14.6)	71.8(22.4)	-	**57.9**(7.0)	52.6(9.6)	-	86.0(21.3)	68.0(22.2)	**40.7**(17.3)
	gesture	75.4(8.9)	70.7(11.6)	-	**59.4**(2.6)	**45.0**(8.5)	-	84.6(18.6)	67.7(23.9)	49.2(10.5)
	boxing	105.5(50.3)	95.4(32.9)	-	**80.2**(11.4)	**54.5**(11.5)	-	86.0(18.2)	70.0(22.7)	77.2(40.9)
S3	walking	162.5(112.3)	78.3(30.4)	-	**66.0**(9.0)	46.8(7.8)	49.8(2.2)	87.4(21.7)	69.6(22.2)	**31.5**(14.2)
	jogging	79.7(22.7)	68.8(14.3)	-	**57.5**(8.8)	44.1(7.8)	-	99.0(21.9)	70.1(21.3)	**33.4**(11.6)
	gesture	71.6(6.8)	58.9(5.6)	-	**50.2**(4.2)	**34.6**(7.4)	-	87.2(11.7)	50.6(18.5)	85.4(33.2)
	boxing	171.4(47.5)	134.3(57.2)	-	**98.4**(39.1)	92.9(28.2)	-	90.3(25.6)	**67.2**(23.0)	77.2(40.9)
	overall	92.7(30.1)	75.7(21.6)	-	**62.0**(9.9)	**44.6**(9.4)	-	89.0(21.2)	67.1(25.9)	45.6(18.1)

* Results using the BiSE implementation from [8], which uses a different edge detector than the BISE in Table V.

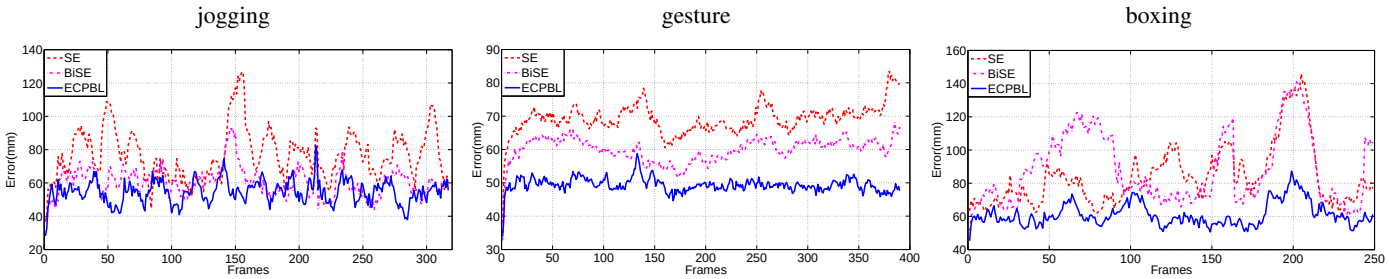

Fig. 12. Tracking error (mm) over time of ECPBL versus baselines on HumanEva-I S1 jogging, gesture and boxing. Invalid ground-truth frames are removed.

error (or mocap bias) in HumanEva-II, where the mean error rarely dropped below 50mm. Two unsupervised methods have lower error than ECPBL. The first is a region-based approach by [22], which has an error of 48.7. On the other hand, ECPBL has a lower standard deviation (9.3 vs 21.9) and lower maximum error (85.3 vs 156.5), suggesting that ECPBL is more stable and less prone to losing track. In contrast, [22] loses track in the jogging sequence. The best result on HumanEva-II is from [21], which improves the optimization stage of the tracker, and also uses a smoothing filter to reduce the error. The smoothing filter requires access to future video frames, and hence is not an online tracker and not directly comparable. Finally, note that besides BiS [8] and ECPBL, the other methods rely on a more accurate 3D mesh model, which improves the silhouette matching. Despite this fact, ECPBL still achieves comparable results using simple cylinder models. Future work will incorporate a 3D-mesh model into the ECPBL.

F. Tracking with a single view

We next present experiments on using ECPBL for single view tracking on HumanEva-I. Fig. 16 shows several examples of tracking using a single view (camera C1). When the person is near to the camera and all limbs are visible (e.g., Fig. 16a), ECPBL can obtain a good estimate of both the global position and body pose. However, because there is only one camera view, the ECPBL will be maximized by completely filling the silhouette with the body model. This causes errors in the global position when the body model is not a perfect replica of the person. For example, in Fig. 16b the model is placed farther away from the camera because the lateral width of the body model is larger than the person. Likewise, in Fig. 16c, the tracker estimates that the global position is closer to the camera because the frontal width of the body model is smaller than the person. Nonetheless, the body pose can still be estimated fairly well, as illustrated in Fig. 16d, where the estimated pose has been shifted to the true global position while keeping the relative limb angles unchanged. Finally, when some limbs are occluded, e.g., as in Fig. 16b, ECPBL cannot localize the occluded limbs since there are essentially no constraints on these limbs.

A quantitative comparison between our ECPBL and existing monocular or double view methods is presented in Table VIII. The error for ECPBL is relatively high (95.7mm), due to the aforementioned errors in the global position. Next, we evaluate the relative pose to the center joint (ECPBL translated), which shifts the root position to the ground-truth position without changing the limb angles relative to the root.[2] After translation, the error of single-view ECPBL is reduced significantly (62.5mm), and is comparable to the OA [45] and CRBM [4], which are both supervised methods. Finally, we evaluate the pose estimate on only the joints visible in the image, i.e., ignoring occluded joints, and the error of ECPBL is further reduced (56.7mm). These results demonstrate that ECPBL is capable of recovering the relative position of the

[2]This is the same evaluation method of TGP [7].

TABLE VII
AVERAGE TRACKING ERROR (MM) ON HUMANEVA-II S4. THE STANDARD DEVIATION OF THE TRACKING ERROR IS IN PARENTHESIS. * INDICATES THAT THE METHOD USES A 3D MESH MODEL.

Type	Methods	Average Error	Max error
Unsupervised	BiS [8]	80.0 (5.0)	∼150
	AICP with Rot. Bounds [38] *	80.0 (13.0)	-
	BISH [24] *	57.9 (13.2)	∼85
	ECPBL (ours)	56.8 (9.3)	85.3
	Region-based tracking [22] *	48.7 (21.9)	156.5
	ISA + Layer 2 smoothing [21] *	32.5 (5.2)	∼50
Supervised	Action recognition prior [5] *	45.2 (13.4)	-

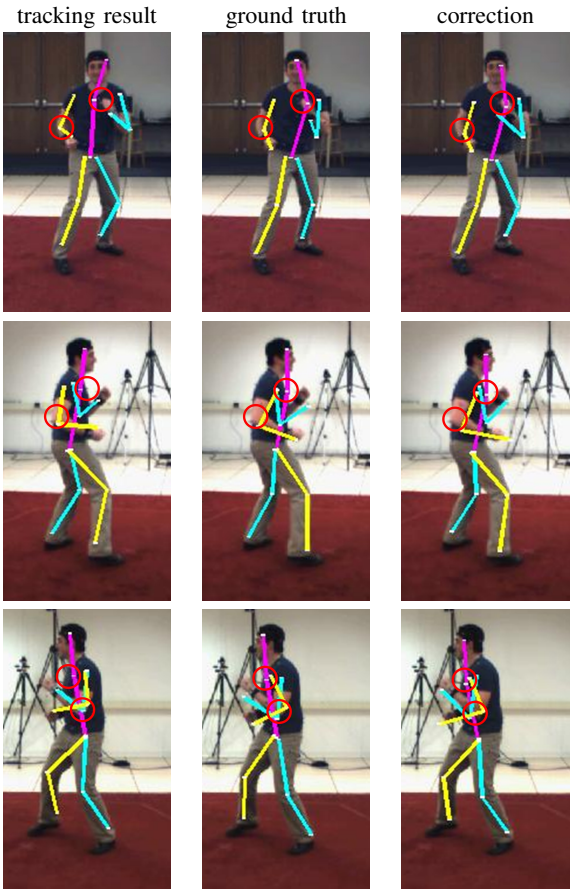

Fig. 14. Example of joint correction on S2 Boxing Validation: (left) tracking result, (middle) mocap ground truth, and (right) corrected pose. The mocap pose is slightly more forward than the image.

visible joints of the human pose from a single-view. When using two views, our ECPBL works well to localize both the global position of the body and local angles of limbs, and has lower error than the unsupervised method of [46].

Note that the errors on the global position and occluded limbs when using a single view are reasonable when considering that ECPBL is an *unsupervised* method that does not use a motion model. Using a supervised motion model with ECPBL would likely provide the necessary constraints to better localize the global position and to handle occluded limbs.

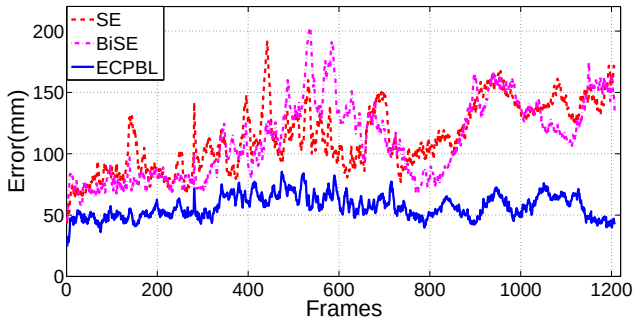

Fig. 15. Tracking error over time for ECPBL and the baseline methods on HumanEva-II.

G. Test on TUM kitchen

The TUM kitchen dataset [53] contains 20 videos of 4 subjects, recorded from 4 color views in a kitchen setting. It is used for evaluating pose estimation and action recognition. In each video, the actor moves back and forth, while performing tasks in the kitchen. Following the experiment setup in [5], we test our ECPBL on the 0-2, 0-4, 0-6, 0-8, 0-10, 0-11, and 1-6 episodes. The joint correction model was trained on a separate episode 1-0. The results on TUM are presented in Table IX. The joint-corrected ECPBL is comparable to a supervised method using an action recognition prior (AP) [5].

H. Test on Human3.6M

The Human3.6M dataset [54] is a large scale dataset for human pose estimation. The dataset contains 3.6 million images of 11 subjects performing 15 actions, taken from 4 color viewpoints, and the corresponding accurate 3D human poses. In our experiments, we perform *multi-view* pose estimation on the 6 training subjects (S1, S6, S7, S8, S9, S11) with provided ground-truth data. We test on the validation sequences of the 8 actions listed in Table X.[3] The results of multi-view pose estimation using ECPBL and the BiS/BiSE baseline algorithms are shown in Table X. ECPBL outperforms BiS and BiSE significantly, with the mean error reduced by about 40% (45mm).

We also compare with the supervised single-view method of [54], linear kernel dependency estimation (LinKDE). Using

[3]Because our body model is only a rough cylinder model, we did not test on actions involving large body contortions (e.g., Sitting Down) or interactions with a chair, which is considered as foreground in the dataset (e.g., Eating).

TABLE VIII
AVERAGE TRACKING ERROR (MM) FOR SINGLE VIEW AND DOUBLE VIEW TRACKING METHODS ON HUMANEVA-I S1 WALKING. ECPBL (TRANSLATED) SHOWS THE RESULT AFTER TRANSLATING THE BODY ROOT TO THE GROUND-TRUTH POSITION WITHOUT CHANGING LIMB ANGLES. ECPBL (VISIBLE) SHOWS THE RESULTS FOR ONLY THE VISIBLE JOINTS OF THE ECPBL (TRANSLATED) RESULT. ALL METHODS USE THE 'C1' OR 'C1' AND 'C2' VIEW.

Single view						Double view	
ECPBL	ECPBL (translated)	ECPBL (visible)	OA [45]	TGP [7]	CRBM [4]	ECPBL	Root Uncertainty [46]
95.7(42.6)	62.5(14.9)	56.7(15.9)	99.6(42.6)	38.1(21.4)	47.3(5.0)	54.6(11.3)	89.3(12.8)

TABLE IX
AVERAGE TRACKING ERROR (MM) ON 7 EPISODES OF THE TUM KITCHEN DATASET.

	0-2	0-4	0-6	0-8	0-10	0-11	1-6	Overall
AP [5]	47.8(18.1)	**60.6**(20.7)	69.1(29.3)	46.9(18.9)	**60.2**(18.4)	**74.0**(33.5)	80.2(35.7)	62.7(24.9)
ECPBL	56.8(11.2)	75.0(14.0)	78.8(23.1)	57.6(11.1)	68.2(22.5)	79.3(19.0)	**73.1**(19.4)	69.8(17.2)
ECPBL (joint corr.)	**42.8**(11.8)	61.9(13.2)	**66.5**(23.3)	**44.5**(11.6)	61.2(13.6)	77.0(17.8)	74.9(17.7)	**61.3**(15.6)

the code provided with [54], LinKDE was trained using the training sequences for the subjects/actions, and tested on the validation sequences. The four views were treated as distinct samples for training and testing. The results are presented in Table X.[4] The overall performance of LinKDE is better than two baseline algorithms, but worse than the ECPBL. Although LinKDE is a supervised method and has better accuracy than the two multi-view baseline methods (BiS/BiSE), it sometimes has difficulty when tracking occluded limbs, due to only using a single view, or when a test pose is not similar to any pose in the training set.

VII. CONCLUSION AND DISCUSSION

In this paper, we have proposed a robust part-based likelihood function, which is based on the exponential Chamfer distance between visible projected parts and silhouette segments, and between visible part edges and edges in the silhouette segment. The exponential transformation of the Chamfer distance better aligns the limbs, making the likelihood function smoother and easier to optimize using APF. Our part-based model helps to localize occluded parts by matching them only to the segmented visible parts. Our method benefits when a part can be segmented well, but is not greatly affected by poorly-segmented parts, since these will be matched to the original silhouettes.

Using the ECD and part-based model together, we obtain very robust tracking results on the HumanEva dataset. Our unsupervised part-based likelihood function performs significantly better than other unsupervised tracking methods on HumanEva-I. After correcting for the bias of the mocap joint system, the part-based likelihood function performs comparably to the current state-of-the-art supervised method, TGPKNN. Especially considering the standard deviation of the error, our robust likelihood function outperforms other methods in terms of stability. We hope that our work can renew interest in unsupervised methods, and serve as a new baseline for unsupervised methods.

Learning the color model for each part depends on the quality of the initial mocap pose. Automatically estimating an initial pose from the first frame, e.g., [21, 23], is a topic of future work. Meanwhile, quick motions like boxing, are difficult to track using the Gaussian diffusion motion model. How to handle the sample propagation between frames of quick motion is also an interesting future work. Finally, the current MATLAB implementation of ECPBL runs at 40s per frame. Optimizing the framework using GPU is another direction of future work.

[4]Since LinKDE predicts poses from cropped human images, the reported error is for the relative joint positions (i.e., relative to the root joint).

ACKNOWLEDGEMENTS

The authors thank L Sigal, AO Balan, and MJ Black for providing the HumanEva-I and HumanEva-II datasets and corresponding code. The authors also thank CE Rasmussen for the GPML Matlab Code from [51].

REFERENCES

[1] N. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *The Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.
[2] R. Urtasun, D. J. Fleet, and P. Fua, "3d people tracking with gaussian process dynamical models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 238–245.
[3] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30(2), pp. 283–298, 2008.
[4] G. W. Taylor, L. Sigal, D. J. Fleet, and G. E. Hinton, "Dynamical binary latent variable models for 3d human pose tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 631–638.
[5] A. Yao, J. Gall, and L. Van Gool, "Coupled action recognition and pose estimation from multiple views," *International Journal of Computer Vision*, vol. 100(1), pp. 16–37, 2012.
[6] R. Li, T.-P. Tian, S. Sclaroff, and M.-H. Yang, "3d human motion tracking with a coordinated mixture of factor analyzers," *International Journal of Computer Vision*, vol. 87(1), pp. 170–190, 2010.
[7] L. Bo and C. Sminchisescu, "Twin gaussian processes for structured prediction," *International Journal of Computer Vision*, vol. 87(1-2), pp. 28–52, 2010.
[8] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International Journal of Computer Vision*, vol. 87(1-2), pp. 4–27, 2010.
[9] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29(1), pp. 5–28, 1998.
[10] T.-J. Cham and J. M. Rehg, "A multiple hypothesis approach to figure tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.
[11] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3d human figures using 2d image motion," in *Computer VisionłECCV 2000*, 2000, pp. 702–718.

TABLE X
AVERAGE TRACKING ERROR (MM) FOR UNSUPERVISED MULTI-VIEW POSE ESTIMATION (BIS, BISE, AND ECPBL) AND SUPERVISED SINGLE-VIEW POSE ESTIMATION (LINKDE). THE RESULTS FOR EACH ACTION ARE AVERAGED OVER THE 6 TRAINING SUBJECTS OF HUMAN3.6M DATASET.

	Directions	Discussion	Greeting	Posing	Purchases	Waiting	Walking	WalkTogether	Overall
BiS	89.9(19.3)	113.2(27.3)	122.1(32.0)	107.4(31.4)	113.8(39.0)	107.6(29.7)	129.6(53.1)	91.4(23.3)	109.4(31.9)
BiSE	96.7(21.5)	112.1(27.0)	114.9(27.4)	110.9(36.2)	112.1(36.6)	110.7(30.0)	132.5(46.8)	93.7(25.0)	110.5(31.3)
ECPBL	57.6(11.9)	**65.3**(16.5)	**68.1**(17.9)	**60.9**(17.0)	**78.6**(28.6)	**71.3**(22.4)	**61.7**(12.2)	**61.0**(13.8)	**65.6**(17.5)
LinKDE [54]	**49.5**(25.8)	69.0(45.1)	92.9(65.1)	105.2(76.3)	87.3(60.1)	92.5(65.3)	63.1(30.0)	61.8(32.4)	77.7(50.0)

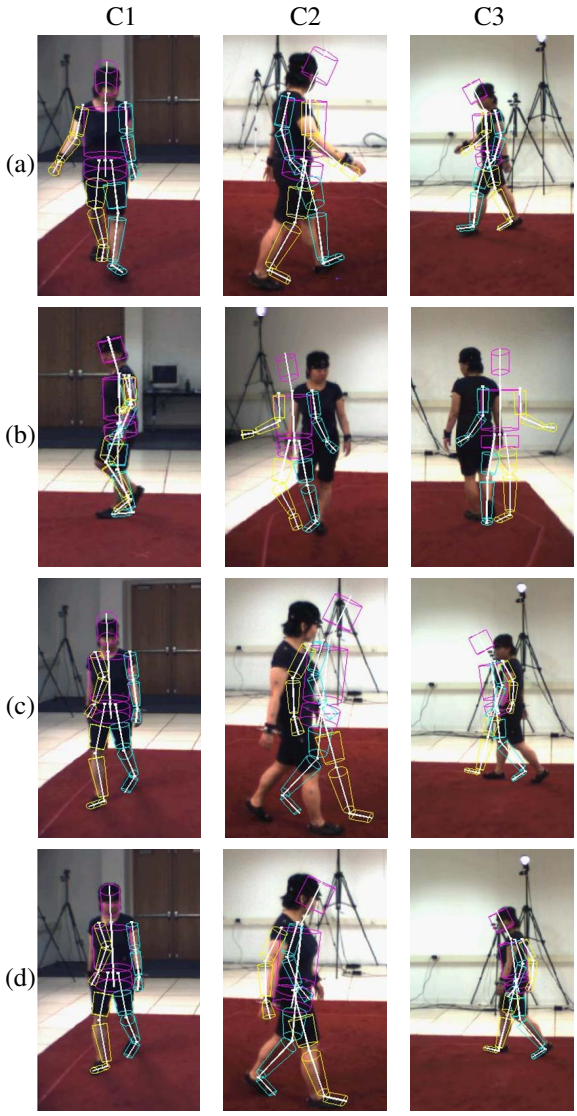

Fig. 16. Tracking results using single view C1: a) the 89th frame, b) the 409th frame, c) the 59th frame, d) the 59th frame after manually change the global position.

[12] K. Choo and D. J. Fleet, "People tracking using hybrid monte carlo filtering," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2001, pp. 321–328.
[13] C. Sminchisescu and B. Triggs, "Estimating articulated human motion with covariance scaled sampling," *The International Journal of Robotics Research*, vol. 22, no. 6, pp. 371–391, 2003.
[14] R. Kehl, M. Bray, and L. Van Gool, "Full body tracking from multiple views using stochastic sampling," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 129–136.
[15] J. Deutscher and I. Reid, "Articulated body motion capture by stochastic search," *International Journal of Computer Vision*, vol. 61(2), pp. 185–205, 2005.
[16] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61(1), pp. 55–79, 2005.
[17] P. Wang and J. M. Rehg, "A modular approach to the analysis and evaluation of particle filters for figure tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 790–797.
[18] D. Knossow, R. Ronfard, and R. Horaud, "Human motion tracking with a kinematic parameterization of extremal contours," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 247–269, 2008.
[19] R. Horaud, M. Niskanen, G. Dewaele, and E. Boyer, "Human motion tracking by registering an articulated surface to 3d points and normals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 158–163, 2009.
[20] V. John, E. Trucco, and S. Ivekovic, "Markerless human articulated tracking using hierarchical particle swarm optimisation," *Image and Vision Computing*, vol. 28, no. 11, pp. 1530–1547, 2010.
[21] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel, "Optimization and filtering for human motion capture," *International Journal of Computer Vision*, vol. 87(1-2), pp. 75–92, 2010.
[22] C. Schmaltz, B. Rosenhahn, T. Brox, and J. Weickert, "Region-based pose tracking with occlusions using 3d models," *Machine Vision and Applications*, pp. 1–21, 2011.
[23] L. Sigal, M. Isard, H. Haussecker, and M. J. Black, "Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation," *International Journal of Computer Vision*, vol. 98(1), pp. 15–48, 2012.
[24] J. Bandouch, O. C. Jenkins, and M. Beetz, "A self-training approach for visual tracking and recognition of complex human activity patterns," *International Journal of Computer Vision*, vol. 99(2), pp. 166–189, 2012.
[25] I. A. Kakadiaris and D. Metaxas, "Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996, pp. 81–87.
[26] C. Sminchisescu and B. Triggs, "Covariance scaled sampling for monocular 3d body tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. I–447.
[27] R. Poppe, "Vision-based human motion analysis: An overview," *Computer Vision and Image Understanding*, vol. 108(1), pp. 4–18, 2007.
[28] T.-P. Tian, R. Li, and S. Sclaroff, "Tracking human body pose on a learned smooth space," Boston University Computer Science Department, Tech. Rep., 2005.
[29] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28(1), pp. 44–58, 2006.
[30] A. Geiger, R. Urtasun, and T. Darrell, "Rank priors for continuous non-linear dimensionality reduction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 880–887.
[31] R. Rosales and S. Sclaroff, "Learning body pose via specialized maps." Advances in neural information processing systems, 2002.
[32] R. Navaratnam, A. W. Fitzgibbon, and R. Cipolla, "The joint manifold model for semi-supervised multi-valued regression," in *IEEE International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.
[33] R. Poppe, "Evaluating example-based pose estimation: Experiments on the humaneva sets," 2007.
[34] L. Sigal, A. Balan, and M. J. Black, "Combined discriminative and generative articulated pose and non-rigid shape estimation," *Advances in neural information processing systems*, vol. 20, pp. 1337–1344, 2007.
[35] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.
[36] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Discriminative density propagation for 3d human motion estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 390–397.

[37] T. Jaeggli, E. Koller-Meier, and L. Van Gool, "Learning generative models for multi-activity body pose estimation," *International Journal of Computer Vision*, vol. 83, no. 2, pp. 121–134, 2009.
[38] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, and T. P. Andriacchi, "Markerless motion capture through visual hull, articulated icp and subject specific model generation," *International Journal of Computer Vision*, vol. 87(1), pp. 156–69, 2010.
[39] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," *Advances in neural information processing systems*, pp. 329–336, 2004.
[40] C. Ionescu, F. Li, and C. Sminchisescu, "Latent structured models for human pose estimation," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2220–7.
[41] C. Sminchisescu, A. Kanaujia, and D. N. Metaxas, "Bm^{3}e: Discriminative density propagation for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29(11), pp. 2030–44, 2007.
[42] Z. L. Husz, A. M. Wallace, and P. R. Green, "Behavioural analysis with movement cluster model for concurrent actions," *Journal on Image and Video Processing*, vol. 2011, p. 2, 2011.
[43] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290(5500), pp. 2319–23, 2000.
[44] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3d pose estimation and tracking by detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 623–630.
[45] E. Simo-Serra, A. Ramisa, G. Alenya, C. Torras, and F. Moreno-Noguer, "Single image 3d human pose estimation from noisy observations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2673–2680.
[46] B. Daubney and X. Xie, "Tracking 3d human pose with large root node uncertainty," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1321–1328.
[47] H. Sidenbladh and M. J. Black, "Learning the statistics of people in images and video," *International Journal of Computer Vision*, vol. 54(1), pp. 183–209, 2003.
[48] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29(1), pp. 65–81, 2007.
[49] C. Sminchisescu and A. Telea, "Human Pose Estimation from Silhouettes. A Consistent Approach Using Distance Level Sets," in *WSCG '02*, 2002.
[50] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.
[51] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.
[52] W. Shen, K. Deng, X. Bai, T. Leyvand, B. Guo, and Z. Tu, "Exemplar-based human action pose correction and tagging," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1784–91.
[53] M. Tenorth, J. Bandouch, and M. Beetz, "The tum kitchen data set of everyday manipulation activities for motion tracking and action recognition," in *IEEE International Conference on Computer Vision (ICCV) Workshops*, 2009, pp. 1089–1096.
[54] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, jul 2014.

Weichen Zhang received the B.S. degree in Automation from Huazhong University of Science and Technology in 2011. He is currently a PhD candidate in Computer Science at the City University of Hong Kong. He is currently with the Video, Image, and Sound Analysis Laboratory , Department of Computer Science, City University of Hong Kong. His research interests include Computer Vision, Machine Learning and Pattern recognition.

Lifeng Shang is a Researcher of Huawei Noah's Ark Lab. He obtained his PhD degree in Computer Science from the University of Hong Kong in 2012. Before joining Huawei, he was a Postdoctoral Research Fellow at the City University of Hong Kong 2012-2013. His research interests include machine learning, computer vision, and information retrieval.

Antoni B. Chan received the B.S. and M.Eng. degrees in electrical engineering from Cornell University, Ithaca, NY, in 2000 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, San Diego (UCSD), San Diego, in 2008. From 2001 to 2003, he was a Visiting Scientist with the Vision and Image Analysis Laboratory, Cornell University, Ithaca, NY, and in 2009, he was a Postdoctoral Researcher with the Statistical Visual Computing Laboratory, UCSD. In 2009, he joined the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, as an Assistant Professor. His research interests include computer vision, machine learning, pattern recognition, and music analysis. Dr. Chan was the recipient of an NSF IGERT Fellowship from 2006 to 2008, and an Early Career Award in 2012 from the Research Grants Council of the Hong Kong SAR, China.