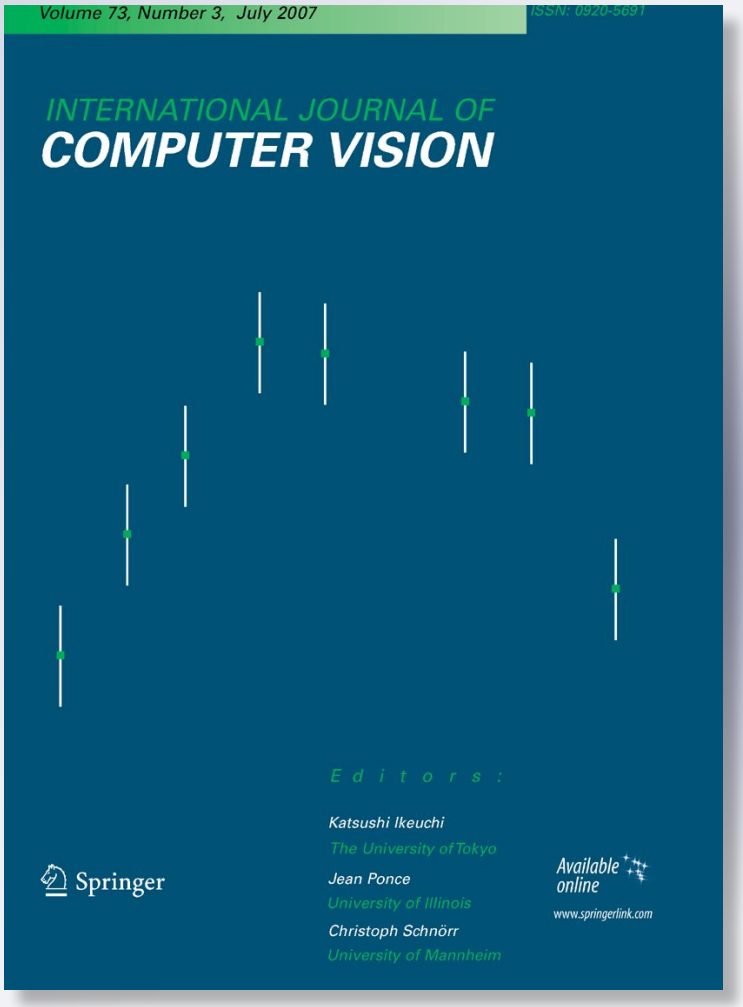
# *Multi-view 3D Human Pose Estimation in Complex Environment*

**M. Hofmann & D. M. Gavrila**

**International Journal of Computer Vision**

ISSN 0920-5691
Volume 96
Number 1

Int J Comput Vis (2012) 96:103-124
DOI 10.1007/s11263-011-0451-1




Springer

Int J Comput Vis (2012) 96:103–124
DOI 10.1007/s11263-011-0451-1

# Multi-view 3D Human Pose Estimation in Complex Environment

**M. Hofmann · D.M. Gavrila**

Received: 9 February 2010 / Accepted: 10 April 2011 / Published online: 1 May 2011


**Abstract** We introduce a framework for unconstrained 3D human upper body pose estimation from multiple camera views in complex environment. Its main novelty lies in the integration of three components: single-frame pose recovery, temporal integration and model texture adaptation. Single-frame pose recovery consists of a hypothesis generation stage, in which candidate 3D poses are generated, based on probabilistic hierarchical shape matching in each camera view. In the subsequent hypothesis verification stage, the candidate 3D poses are re-projected into the other camera views and ranked according to a multi-view likelihood measure. Temporal integration consists of computing K-best trajectories combining a motion model and observations in a Viterbi-style maximum-likelihood approach. Poses that lie on the best trajectories are used to generate and adapt a texture model, which in turn enriches the shape likelihood measure used for pose recovery. The multiple trajectory hypotheses are used to generate pose predictions, augmenting the 3D pose candidates generated at the next time step.

We demonstrate that our approach outperforms the state-of-the-art in experiments with large and challenging real-world data from an outdoor setting.

**Keywords** Human motion capture · Articulated pose recovery · Human computer interaction · Surveillance

Most research was carried out while the first author was with TNO Defence, Safety & Security, The Hague, The Netherlands.

M. Hofmann · D.M. Gavrila (✉)
Intelligent Autonomous Systems Group, Informatics Institute, University of Amsterdam, Science Park 107, 1098 XG Amsterdam, The Netherlands
e-mail: d.m.gavrila@uva.nl

M. Hofmann
e-mail: mhofmann.uva@gmail.com

## 1 Introduction

The recovery of 3D human pose is an important problem in computer vision with many potential applications in human computer interfaces, motion analysis (e.g. sports, medical) and surveillance. 3D pose also provides informative, view-invariant features for a subsequent activity recognition step. Despite the considerable advances that have been made over the past years (see next section), 3D human pose recovery remains essentially unsolved for unconstrained movement in dynamic and cluttered environment. The challenges involve estimating articulated motion of bodies of which the exact proportions are not known in advance, dealing with the underconstrained nature of the problem due to loss of depth information and/or (self) occlusion, and performing foreground-background segmentation.

This paper presents a framework for the estimation of 3D human upper body movement from multiple views, which entails the combination of probabilistic single-frame[1] pose recovery, temporal integration and texture model adaptation. Using input from three calibrated cameras, we are able to handle arbitrary movement—i.e. not limited to walking and running—in cluttered scenes with non-stationary backgrounds (see Fig. 3). By integrating single-frame pose recovery within a tracking and prediction mechanism, there is no need to rely on specific initial poses to jump-start the pose estimation. The framework thus also entails automatic re-initialization after a period of failure.

Efficiency is an important design consideration. Single-frame pose recovery is implemented by a multi-stage cascade architecture, where candidate poses are increasingly

[1] "Single-frame" in this paper denotes multi-view image data collected at a particular time instant.

pruned by the successive processing stages. Early processing stages are characterized by lighter computational costs per candidate pose, and a higher degree of inherent parallelism. For example, the pose hypothesis stage is performed by each camera independently. The computational burden is shifted as much as possible to an off-line stage, by matching a set of pre-rendered 2D pose (shape) exemplars, compactly organized in a pre-computed tree structure. Later processing stages are characterized by higher computational costs per remaining candidate pose and a stronger interaction across camera views. Take for example, the use of inverse kinematics for local pose optimization.

The current system also has some limitations. Like previous 3D pose recovery systems, it currently cannot handle a sizable amount of external occlusion. It furthermore assumes the existence of a 3D human model that roughly fits the person in the scene (we are able to use the same generic model for different male adults in the experiments). Lastly, the system expects subjects to be roughly in a standing position for pose initialization, for computational cost reasons.

## 2 Previous work

There is meanwhile an extensive literature on 3D human pose estimation. We refer to a selection of papers here which we consider most relevant to this paper. For a more exhaustive listing, see surveys by Forsyth et al. (2005), Gavrila (1999), Moeslund et al. (2006), Sigal and Black (2010).

One line of research has focused on 3D model-based tracking; i.e. given a reasonably accurate 3D human model and an initial 3D pose, predict the pose at the next time step using a particular dynamical and observation model (Balan and Black 2006; Brubaker et al. 2010; Deutscher and Reid 2005; Drummond and Cipolla 2001; Fossati et al. 2009; Gavrila and Davis 1996; Hasler et al. 2009; Kakadiaris and Metaxas 2000; Lee and Elgammal 2010; Li et al. 2010; Ong et al. 2006; Peursum et al. 2010; Roberts et al. 2006; Rosenhahn and Brox 2007; Stenger et al. 2006; Vondrak et al. 2008; Xu and Li 2007). Multi-hypothesis approaches based on particle filtering (Brubaker et al. 2010; Deutscher and Reid 2005; Ong et al. 2006; Peursum et al. 2010; Xu and Li 2007) or non-parametric belief propagation (Sigal et al. 2004) are used for increased robustness. However, the high dimensionality of the pose parameter space necessitates researchers to employ strong motion priors (i.e. known action classes such as walking, running) and/or various sequential sampling techniques. In practice, tracking soon goes astray if no recovery mechanism is added.

Another line of research has dealt with 3D pose initialization. Work in this category can be distinguished by the number of cameras used. Multi-camera systems for 3D pose initialization were so far applied in controlled indoor environments. The near-perfect foreground segmentation resulting from the stationary background, together with the many cameras used ($> 5$), allows to recover pose by Shape-from-Silhouette techniques (Cheung et al. 2005a, 2005b; Corazza et al. 2010; Kehl and Gool 2006; Mikic et al. 2003; Starck and Hilton 2003; Sundaresan and Chellappa 2009). A new line of research goes beyond the recovery of pose parameters to the estimation of the non-rigid surface of the 3D human model (Balan et al. 2007; Gall et al. 2009).

Single camera systems for 3D pose initialization can be sub-divided whether they use generative or learning-based approaches. Learning-based approaches construct a mapping between 3D pose and 2D image observables using machine learning techniques (Agarwal and Triggs 2006; Bo and Sminchisescu 2010; Bissacco et al. 2007; Kanaujia et al. 2007; Rogez et al. 2008; Shakhnarovich et al. 2003). These approaches are conceptually appealing and fast, but questions still remain regarding their scalability to arbitrary poses. Certainly, a large number of examples would be needed in that case to allow for successful regression, given the ill conditioning and high dimensionality of the problem (most experimental results involve restricted movements, i.e. walking). Furthermore, learning-based approaches tend to rely on good foreground segmentation.

On the other hand, pose initialization using 3D generative models (Kohli et al. 2008; Lee and Cohen 2006) involves finding the best match between model projections and image, and retrieving the associated 3D pose. 3D generative models typically involve compositions of volumetric primitives like ellipsoids or cones (Forsyth et al. 2005; Gavrila 1999; Moeslund et al. 2006). Alternatives involve linear subspace models, derived from a training set of 3D human body scans (SCAPE, Balan et al. 2007) and mesh models (e.g. Hasler et al. 2009).

Pose initialization using 2D generative models involves 2D pose recovery (Andriluka et al. 2009; Ferrari et al. 2009; Mori and Malik 2006; Ramanan et al. 2007) followed by a 3D inference step (Lee and Nevatia 2009) with respect to the joint locations. In order to reduce the combinatorial complexity associated with pose recovery, previous generative approaches apply part-based techniques (Sigal et al. 2004; Bergtholdt et al. 2010). As far as these involve search space decomposition (i.e. searching first for the torso, then arms and legs) (Mori and Malik 2006; Navaratnam et al. 2005; Ramanan et al. 2007), they are error prone; estimation mistakes made early on based on partial model knowledge cannot be corrected later on. In practice, this means that instances with an appreciable amount of torso movement and rotation are difficult to handle. It proves difficult to combine an efficient inferencing mechanism on body parts with the enforcement of multi-part constraints (e.g. dynamics, appearance); Sigal and Black (2010) suggest that part-based

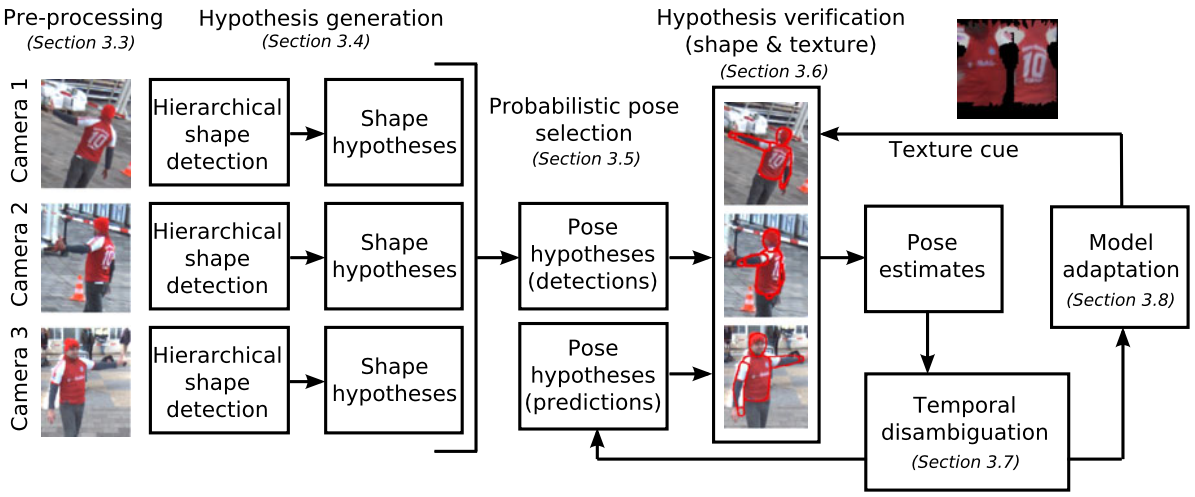



**Fig. 1** Framework overview. See Sect. 3.1

approaches are probably best to provide an efficient proposal function for estimation with a more centralized representation of the body (e.g. a kinematic tree).

Methods for pose initialization can serve to initialize the above-mentioned trackers. An increasingly popular alternative is their use in "tracking-as-recognition" approaches, especially when no strong motion priors are available. Here, pose estimates obtained independently at each time instant are integrated to consistent trajectories, taking into account a more generic motion model. This is typically achieved by Markov chain optimization (Fossati et al. 2007; Lee and Nevatia 2009; Lv and Nevatia 2007; Navaratnam et al. 2005; Peursum et al. 2007).

Given the extensive amount of work done on human 3D pose estimation, it has been difficult to assess how the various approaches stack against each other. In an upcoming special issue of this journal, Sigal et al. (2010) make an important contribution in this regard, by providing the "HumanEva" data set. It contains a sizeable amount of multi-video and synchronized motion capture data of humans performing a set of predefined actions in a controlled indoor environment. Together with the provided motion capture ground truth and baseline algorithm, it allows other researchers to benchmark their 3D pose estimation systems on a common data set (Bergtholdt et al. 2010; Bo and Sminchisescu 2010; Brubaker et al. 2010; Corazza et al. 2010; Gall et al. 2010; Peursum et al. 2010; Lee and Elgammal 2010; Li et al. 2010).

## 3 3D Human Pose Estimation

### 3.1 Overview and Contribution

Figure 1 presents an overview of the proposed framework. Image pre-processing determines a rough region of interest in the 3D scene and in the various camera views, based on foreground segmentation and volume carving (Sect. 3.3).

In the hypothesis generation stage (Sect. 3.4), candidate 3D poses are obtained by matching a pre-computed library of 2D pose exemplars containing silhouette data in the individual camera views. For efficiency, matching is performed hierarchically using a tree structure; the latter was constructed on top of the exemplar library off-line. A pose selection step follows, that maps matched 2D pose exemplars to 3D poses and estimates the corresponding posteriors (Sect. 3.5).

In the subsequent hypothesis verification stage (Sect. 3.6), the candidate 3D poses are projected to all camera views and ranked according to a multi-view likelihood measure. This involves three sub-stages. In the first sub-stage, the projections of candidate 3D poses into the camera views are approximated with the above-mentioned library of 2D pose exemplars, assuming orthographic projection. Because these 2D poses exemplars are already computed, the projection involves a simple table look-up operation. The multi-view likelihood measure involves shape only and can thus be computed very fast. The second sub-stage uses perspective projection and graphical rendering; this enables the use of both shape and texture in the likelihood measure (apart from the first frames, when a texture model is not available). In the last sub-stage, a gradient-based procedure optimizes the pose parameters in continuous parameter space.

Temporal integration consists of computing $K_{traj}$ best trajectories in batch mode using a Viterbi-style maximum likelihood approach (Sect. 3.7). Poses that lie on the best trajectories are used to generate and adapt a texture model (Sect. 3.8), which provides the above-mentioned texture component in the multi-view matching likelihood of hypothesis verification. The multiple trajectory hypotheses are also used to generate pose predictions, augmenting the 3D pose

candidates generated by single-frame pose recovery at the next time step.

Pose estimation involves a multi-stage recovery process, where an increased computational effort is spent in the later stages, as the 3D pose space is successively pruned. We address the issue of the potentially unfavorable combinatorics of our exemplar-based pose representation by reducing the number of solutions at each process stage by ranking, non-maxima suppression and truncation. First, obtained solutions are ranked according to their likelihood/posterior values. A non-maxima suppression procedure (implemented as a single pass over the solutions) removes solutions which have lower likelihood/posterior values than similar solutions, given a criterion for similarity (e.g. distance in image or pose space); it is applied to ensure diversity in the solutions produced. The remaining list of solutions is truncated by size, maintaining the most likely/probable solutions.

The contributions of this paper are two-fold. The main contribution is a framework for estimating unconstrained 3D human movement in complex environment using a moderate number of cameras. It consists of single-frame pose recovery, temporal integration and texture-based model adaptation components, as described above. The way multiple pose trajectories are used goes beyond previous "tracking-as-recognition" approaches (see Sect. 2), where the computation of a (single) best pose trajectory is solely a post-processing step, decoupled from the estimation process. Model adaptation in our approach, furthermore, does not require a pre-defined key pose (i.e. feet apart) (Fossati et al. 2007; Ramanan et al. 2007) or a scripted initialization movement (Kakadiaris and Metaxas 2000). To reduce the chance of a wrong model update, we update only for those poses which lie on the most likely pose trajectory, i.e. we perform batch-mode temporal integration before model adaptation, rather than model adaptation at each time instant independently (Balan and Black 2006). We do not use strong motion priors (e.g. Lee and Elgammal 2010; Sigal et al. 2004).

The second contribution concerns the way multi-camera pose recovery is performed. The error-prone foreground segmentation resulting from operating in dynamic outdoor environments together with the lower number of cameras used prevents solving matters by Shape-from-Silhouette techniques outright (see Sect. 2). Inverse kinematics techniques (Kakadiaris and Metaxas 2000; Knossow et al. 2008), on the other hand, require close initial estimates. We also do not wish to rely on feature correspondences across cameras (i.e. wide-baseline stereo), as this will be difficult to achieve robustly. Instead, we propose to perform 3D pose detection for each camera independently and fuse information at the pose parameter level by means of the efficient multi-stage recovery process described above. Fusing the information at the pose level improves the scalability with respect to the number of cameras (e.g. allowing optimized per-camera matching, improved algorithm parallelism). We introduce a probabilistic pose selection criterion which implicitly performs viewpoint selection by evaluating and ranking by a pose posterior term. An advantage of our exemplar-based approach is that it describes the articulations of the upper-body as a whole. This ensures that upon matching, all available model knowledge is used at the same time, avoiding some of the drawbacks of the part-based decomposition approaches discussed in Sect. 2. Our framework offers two ways to go beyond the parameter discretization induced by the exemplar-based approach: by means of local pose optimization and pose prediction, both performed in continuous parameter space. This paper extends our earlier work Hofmann and Gavrila (2009a, 2009b).

In the following sections, main system parameters are denoted by variables. For an overview of the actual values used in the experiments, see Appendix A.

### 3.2 3D Human Shape Model

Our 3D upper body model uses tapered super-quadrics as body part primitives (Gavrila and Davis 1996); this yields a good trade-off between desired accuracy and model complexity (i.e. number of parameters). Each superquadric has parameters for length $(a_1, a_2, a_3)$, squareness $(e_1, e_2)$ and tapering $(t_x, t_y)$. The generating function for a vertex on the superquadric surface is

$$\zeta(u,v) = \begin{pmatrix} a_1 \cos^{e_1} u \cos^{e_2} v (t_x \sin^{e_1} u + 1) \\ a_2 \cos^{e_1} u \sin^{e_2} v (t_x \sin^{e_1} u + 1) \\ a_3 \sin^{e_1} u \end{pmatrix} \quad (1)$$

where the angle parameters $u, v$ are in the ranges $-\frac{\pi}{2} \leq u \leq \frac{\pi}{2}, -\pi \leq v \leq \pi$.

We use a model with generic parameters (averaged from limb lengths estimates of our actors) and do not tune for slight differences in body height or limb length among the actors in our data sequences (which are in the order of $<10$ cm w.r.t. height). Articulation at each joint is represented using transformations of homogeneous coordinates

$$\mathbf{x}' = H\mathbf{x}, \quad H = H(R(\phi, \theta, \psi), T) \quad (2)$$

where $R$ is a $3 \times 3$ rotation matrix determined by the Euler angles $\phi$, $\theta$, $\psi$, and $T$ a constant $3 \times 1$ translation vector. Transformations of limbs not at the model root are represented by a kinematic chain $\mathbf{H} = H_1 H_2 \cdots H_k$ along the respective joints. Given the body parts torso, neck, head, upper arm, lower arm and hand, we represent a 3D upper body pose as a 13-dimensional vector of joint angles

$$\boldsymbol{\pi} = (\phi_t, \theta_t, \psi_t, \ \phi_h, \psi_h, \ \phi_s^l, \theta_s^l, \psi_s^l, \theta_e^l, \ \phi_s^r, \theta_s^r, \psi_s^r, \theta_e^r) \quad (3)$$

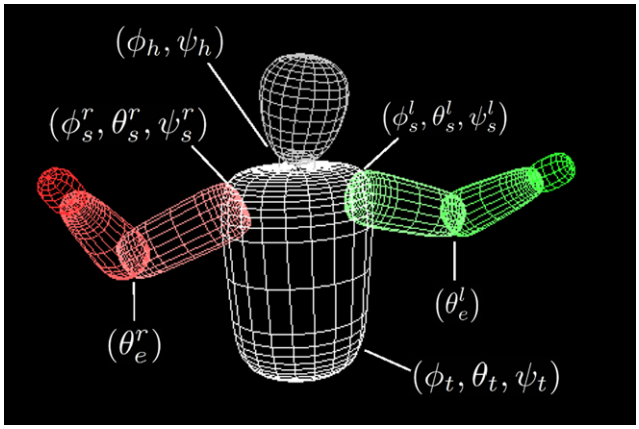



**Fig. 2** Visualization of the shape model, including a description of the joint angles in (3)

augmented by a three dimensional vector $\boldsymbol{x}$ denoting the position of the root of the articulated structure, which is, in our case, the torso center (depending on context, we use in this paper the term "pose" to denote either $\boldsymbol{\pi}$ or $(\boldsymbol{\pi}, \boldsymbol{x})$). Figure 2 provides a visualization of the model as well as the joint angles defined in (3).

### 3.3 Image Pre-processing

See Fig. 1. The aim of the pre-processing stage is to obtain a rough region of interest, both in terms of the individual 2D camera views and in terms of the 3D space. Due to the considered dynamic environment, segmenting the foreground by background modeling (Zivkovic 2004) in the separate camera views is error-prone; moving objects or people in the background might create spurious foreground blobs and therefore false regions of interest. See Fig. 3, top two rows.

Figure 3 also illustrates that the human silhouettes are not necessarily segmented in a quality suitable for solving pose recovery by Shape-from-Silhouette techniques outright (see Sect. 2). In order to take some advantage of the additional information provided by multiple camera views, nevertheless, we fuse the computed foreground masks by means of volume carving (Laurentini 1994). After the necessary morphological operations (dilation), connected voxel components of a minimum height and size give an estimate of the number of people and their rough 3D location in the scene. More specifically, given binary volume information, we compute the accumulated projection image along the ground plane normal direction and require a certain minimum "mass" in the resulting image to form a 3D blob that is recognized as an object. 3D blobs that are too small or that lie too far away from the ground plane are deleted; we also remove parts of the legs in our volume reconstruction by removing voxels less than $h_{torso} = 70$ cm above ground; body pose initialization is thus restricted to roughly standing position (a condition which can be relaxed, when allowing for a larger search space for the torso parameters at the next processing stage). Projecting the reconstructed voxels onto the camera images produces an improved foreground mask. See Fig. 3, third row, where various shadows, the train passing by, and the other 3D person blob, currently not under consideration, have been eliminated.

The voxel information in combination with our camera calibration yields information about the image scales and regions of interest to be used in the forthcoming hypothesis generation step (Sect. 3.4). The input images, together with the corresponding foreground edge- and a distance-transformed-image, are stored at the relevant scales for further processing (the 2D pose exemplars that the images are matched with are generated at a fixed scale).

### 3.4 Pose Hypotheses Generation

See Fig. 1. We generate pose hypotheses by processing the camera views individually. We follow an exemplar-based approach and match each camera view image with a pre-generated 2D pose exemplar library with known 3D articulation. The challenge in creating an exemplar library is to establish a reasonable trade-off between representation specificity (i.e. the number of 2D pose exemplars, the dissimilarity between neighboring 2D pose exemplars) and efficiency (both in terms of storage and matching speed).

#### *3.4.1 2D Pose Exemplar Generation*

To obtain a discretized 3D pose space representation, we first define a set of upper body poses by specifying lower and upper bounds for each joint angle separately and discretize each angle. The Cartesian product of these angles contains anatomically impossible poses; these are filtered by collision detection using the 3D shape model (Sect. 3.2) and through rule-based heuristics. See Appendix A for details.

The outcome, the set $\Pi$ of "allowable" 3D poses, is used to generate a 2D pose exemplar library $S$ by means of projection of the 3D shape model. In our case, the 2D pose exemplars contain silhouette information only. The main reason not to incorporate internal edges was that these are hard to detect in typical scenery (i.e. contrast between arm and torso for similar colored clothing) and we preferred in the early processing stages to concentrate on the robust features. This had also the beneficial effect to reduce the number of necessary exemplars. A further reduction in the latter is achieved by the use of orthographic projection in computing $S$.

The above-mentioned discretization of joint angle is chosen fine enough that the 3D pose space is adequately covered; in other words, the similarity in appearance of *any* projected 3D pose and the closest corresponding 2D pose exemplar should lie within a user-supplied matching tolerance. Doing so, the generated 2D pose exemplar library $S$

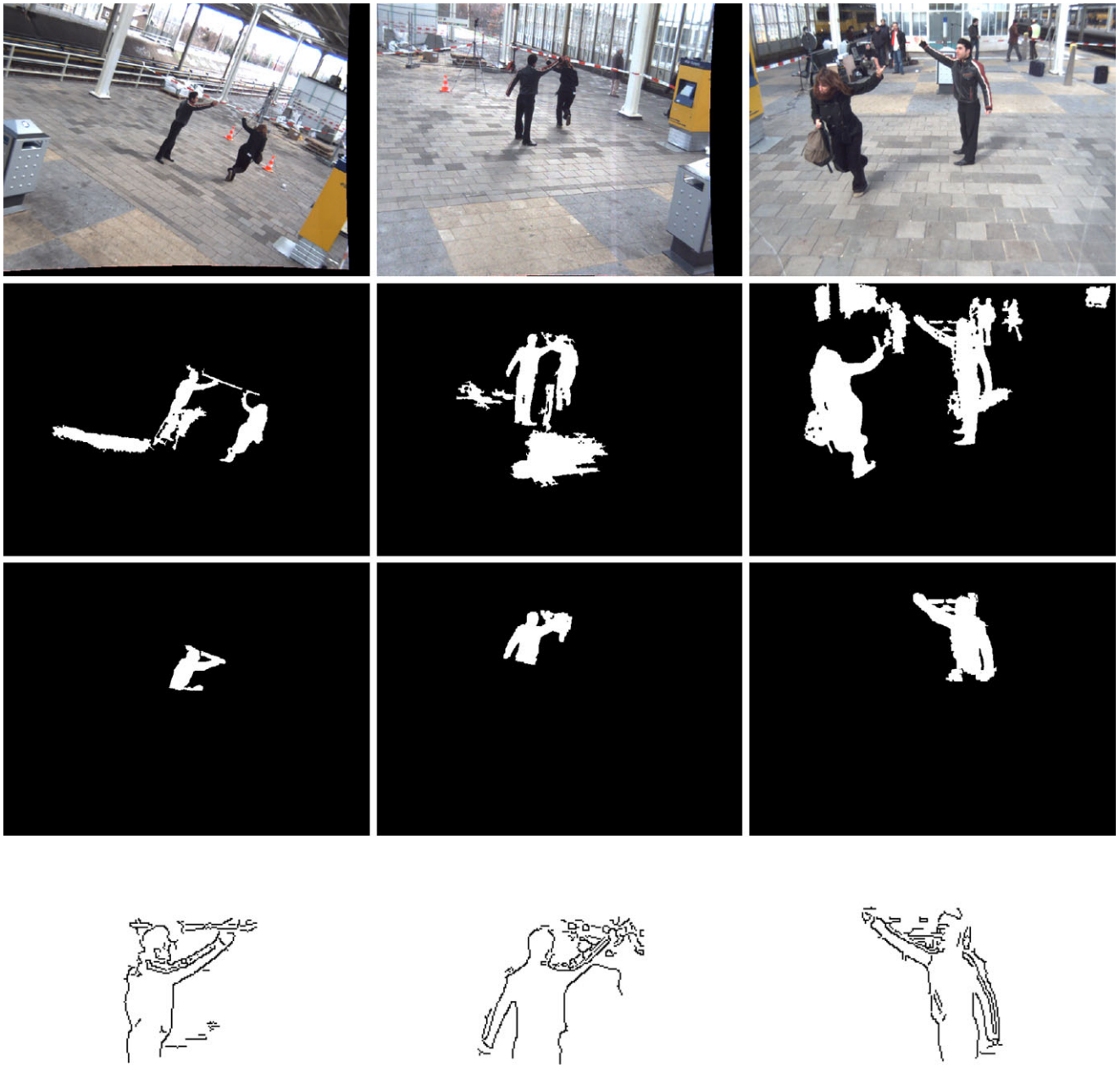


**Fig. 3** Image pre-processing in dynamic environment: (*top row*) input image (*second row*) foreground mask based on background modeling in the individual camera views. Note the artifacts introduced by *shadow*, and people and a train moving in the background (*third row*) improved foreground mask after volume carving, (re)projection and object selection, (*bottom row*) foreground edge image

(roughly of size $15 \times 10^6$) might contain a number of similar exemplars; this redundancy will be reduced in next subsection, where only a subset of $S$ will be actually used for matching.

### *3.4.2 2D Pose Exemplar Tree Construction*

2D pose exemplars are hierarchically organized in a tree structure, for efficient matching. See Fig. 4. The exemplars at the various levels of the tree, $S_l$, $l = 1, \ldots, L$, can be grouped based on their appearance (e.g. chamfer distance) (Gavrila 2007) or based on a decomposition of the underlying 3D joint angles (Stenger et al. 2006). The appearance-based grouping is attractive because of the compactness of the resulting representation; similar 3D pose projections (e.g. front and back views) can be grouped together even if they are distant in joint angle space. However, the appearance-based grouping approach of Gavrila (2007) is not directly applicable to our case since it has

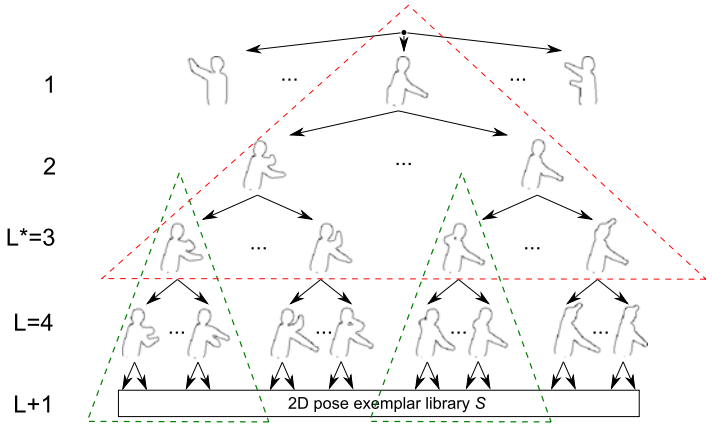



**Fig. 4** Schematized structure of the 4-level 2D pose exemplar tree and its construction. First, $N^*$ exemplars (i.e. the set $S_{L^*}$) are generated for subsequent appearance-based grouping of the levels $l = 1 \ldots L^*$ (*top triangle*). Second, each pose exemplar in $S$ is assigned to an exemplar in $S_{L^*}$ based on distance in appearance space. Third, $N^*$ subtrees are constructed by appearance-based grouping to build the levels $L^* + 1 \ldots L + 1$ (*lower triangles*); the last level, representing the allowable pose space $S$, is discarded. See Sect. 3.4.2

quadratic complexity in terms of the number of exemplars (recall that we are dealing with about $15 \times 10^6$ exemplars, see Sect. 3.4.1).

We therefore use a hybrid approach for tree construction, combining appearance-based grouping and discretization of underlying joint angles. See Fig. 4. The basic idea is to split the tree construction process in two steps: first, the creation of the top $L^*$ levels, and then the creation of the sub-trees that are attached to the nodes at level $L^*$.

First, we establish the maximum number of exemplars $N^*$ that is still practically manageable for an appearance-based grouping approach with quadratic complexity as in Gavrila (2007), given the processing environment at hand. We generate these $N^*$ (in our case, $N^* \approx 20000$) exemplars with a coarser discretization of the allowable joint angles. Assuming a roughly constant branch ratio $B$ throughout the tree, $N^*$ specifies a tree level $L^*$ (in our case, $L^* = 3$). For the root $l = 1$ up to level $L^*$, we will construct the tree by appearance-based grouping using the chamfer distance (Gavrila 2007).

As a second step, each of the 2D pose exemplars $S$ associated with the allowable poses are assigned to one of the $N^*$ prototypes of $S_{L^*}$, based on smallest distance in appearance space.

Finally, we construct the $N^*$ sub-trees one-by-one, as before, based on appearance-based grouping, similar to Gavrila (2007). For each sub-tree, the assigned exemplars of the previous step are used in its construction. The last level $L + 1$ of the tree is discarded, to reduce redundancy in 2D pose exemplar representation (e.g. poses with different arm positions occluded by torso). The final tree has level $L$ and each new leaf level node $\{s | s \in S_L\}$ contains a pointer to the 3D poses $\boldsymbol{\Pi}_s$ that corresponded to the discarded children nodes.

### 3.4.3 *Hierarchical Bayesian 2D Pose Exemplar Matching*

With the hierarchical 2D pose exemplar representation now in place (Fig. 4), we implement online 3D pose hypothesis generation by a tree traversal process, following Gavrila (2007). Tree traversal starts at the root. Processing a node involves matching the corresponding (prototype) 2D pose exemplar with the image at some interest locations. For the locations where the distance measure between 2D pose exemplar and image is below a user supplied threshold $\tau$, the child nodes are added to the list of nodes to be processed. For locations where the distance measure is above-threshold, search does not propagate to the subtree.

The above coarse-to-fine approach is combined with a coarse-to-fine approach over the transformation parameters (i.e., image translation). Image locations on a grid $\gamma_l$, where matching is successful for a particular non-leaf node, give rise to a new set of interest locations for the child nodes on a finer grid $\gamma_{l+1}$ in the vicinity of the original locations. At the root, the interest locations lie on a uniform grid over the image. The combined coarse-to-fine approach in pose and transformation space leads to massive efficiency gains (i.e. several order of magnitude) compared to the brute-force of matching the leaf level 2D pose exemplars ($S_L$) one-by-one at each image location. By following a path in the tree toward the leaf node, both exemplar suitability and exemplar localization increase. Final detections are the successful matches at the leaf level of the tree.

Gavrila (2007) sets matching thresholds $\tau$ based on the posterior for the existence of a correct match at a current node $s_l$ at level $l$, after a set of observations along the path from the root to that node. A match is defined correct, if that particular node (i.e. the associated 2D pose exemplar and associated image location) lies on the path from the root

to the best matching leaf level node (the "optimal" path). For notational simplicity, we do not include in the remainder the subscripts regarding to image location or an index denoting a particular node at level $l$. Let $s_l^+$ and $s_l^-$ denote the event outcome that the corresponding 2D pose exemplar $s_l$ matches correctly and incorrectly, respectively (i.e. $p(s_l^+) + p(s_l^-) = 1$). The observation obtained at the $l$-th level of the tree, is denoted by $O_l$; in our case, this is the uni-directional chamfer distance. Define $O_{1:l} = \{O_i\}_{i=1}^{l}$ to be the observations from the top level up to level $l$, along a particular path in the tree.

Under the Markov assumption along the path from the root to the current node (with $x_l$ denoting either $s_l^+$ or $s_l^-$)

$$p(O_l|O_{1:l-1}x_l) = p(O_l|O_{1-1}x_l) \tag{4}$$

and considering three possible transitions from a parent node at level $l-1$ to a current node at level $l$

1. $s_l^+ s_{l-1}^+$: both parent and current node lie on optimal path,
2. $s_l^- s_{l-1}^+$: parent lies on optimal path but current node does not, and
3. $s_l^- s_{l-1}^-$: parent does not lie on optimal path (and consequently, neither does current node),

Gavrila (2007) derives the following recursive form of the posterior

$$p(s_l^+|O_{1:l}) = \frac{1}{1+\alpha_l} \tag{5}$$

with ($l > 1$)

$$\alpha_l = \frac{p(s_{l-1}^+|O_{1:l-1})\; p(O_l|O_{l-1}s_l^- s_{l-1}^+)\; p(s_l^-|s_{l-1}^+)}{p(s_{l-1}^+|O_{1:l-1})\; p(O_l|O_{l-1}s_l^+ s_{l-1}^+)\; p(s_l^+|s_{l-1}^+)} + \frac{p(s_{l-1}^-|O_{1:l-1})\; p(O_l|O_{l-1}s_l^- s_{l-1}^-)}{p(s_{l-1}^+|O_{1:l-1})\; p(O_l|O_{l-1}s_l^+ s_{l-1}^+)\; p(s_l^+|s_{l-1}^+)}$$

$$\alpha_1 = \frac{p(s_1^-)}{p(s_1^+)}\,\frac{p(O_1|s_1^-)}{p(O_1|s_1^+)}$$

Equation (5) describes how the probability that a particular node provides a correct match during tree search ($p(s_l^+|O_{1:l})$) is based on the probability that the match is correct at the parent node ($p(s_{l-1}^+|O_{1:l-1})$), the observations made at the current and parent node ($O_l$ and $O_{l-1}$) and likelihood functions for the three possible transitions from the parent to the current node ($p(O_l|O_{l-1}x_l x_{l-1})$). The likelihood functions are derived from histogramming observations at the nodes of the template tree on a training set, where the correct solution is known. For example, $p(O_l|O_{l-1}s_l^+ s_{l-1}^+)$ is derived by collecting dissimilarity measurements along the path from the top to the best matching 2D pose exemplar at the leaf level. For details, see Gavrila (2007).

We use (5) to discontinue search below those tree nodes where the match is below a certain level-specific threshold $\tau_l$. The outcome of the pose hypothesis generation stage is a list of leaf-level 2D pose exemplars $s_L \in S_L$ with associated image locations and posterior probabilities $p(s_L^+|O)$. Non-maximum suppression of the results removes lower-ranked matches of the same 2D pose exemplar nodes in an image neighborhood of $u_{PosHyp}$ pixels in $x/y$ direction. About $K_{PosHyp} \approx 300{,}000$ pose solutions pass this stage on average in our experiments (derived from approximately 3000 leaf level 2D pose exemplars) and serve as input to the subsequent pose selection stage.

### 3.5 3D Candidate Pose Selection

See Fig. 1. Given a particular 2D pose exemplar $s_L \in S_L$ that is hypothesized by the previous processing stage, we now derive the posterior for a 3D pose $\boldsymbol{\pi}$ that it represents (i.e. recall last paragraph of Sect. 3.4.2). To emphasize the link between the 2D pose exemplar $s_L$ and underlying pose $\boldsymbol{\pi}$, we now denote in this section the former by $s_\pi$, changing the subscript. The posterior is given by (see Appendix B for details)

$$p(\boldsymbol{\pi}|O) = \frac{p(O|s_\pi^+)}{p(O|s_\pi^+)\,|\Pi_s| \;+\; p(O|s_\pi^-)\,|\Pi \setminus \Pi_s|} \tag{6}$$

where $|\Pi_s|$ is the number of 3D poses associated with the 2D pose exemplar $s$, and $|\Pi|$ the number of all poses. Regarding observables $O$, we shift from the uni-directional chamfer distance used for hierarchical 2D pose exemplar matching to the more accurate and expensive bi-directional chamfer distance, for all shape matching in the remaining process stages (thus $p(O|s_\pi^+)$ and $p(O|s_\pi^-)$ cannot be re-used from previous subsection).

In order to experimentally determine $p(O|s_\pi^+)$ and $p(O|s_\pi^-)$, we make a number of simplifying assumptions, mainly to handle the scarcity of available training data. We first assume that $p(O|s_\pi^+)$ does not depend on the exact pose exemplar $s^+$; the contributions of all $s^+$ in the training set are aggregated for the purpose of estimating the underlying probability density function (PDF). We do however differentiate between $p(O|s_\pi^-)$ at different $s_\pi^-$ in order to account for different degrees of saliency of the 2D pose exemplars (e.g. "arms outstreched" more discriminative than "arms along body"). We aggregate the contributions of all $s^-$ corresponding to a particular first-level ancestor in the tree, and thus maintain $|S_1|$ separate distributions.

The various distributions obtained by histogramming are subsequently fitted. $p(O|s_\pi^+)$ is fitted by a log-normal distribution and $p(O|s_\pi^-)$ by both log-normal and normal distributions. For the latter case, we choose the distribution

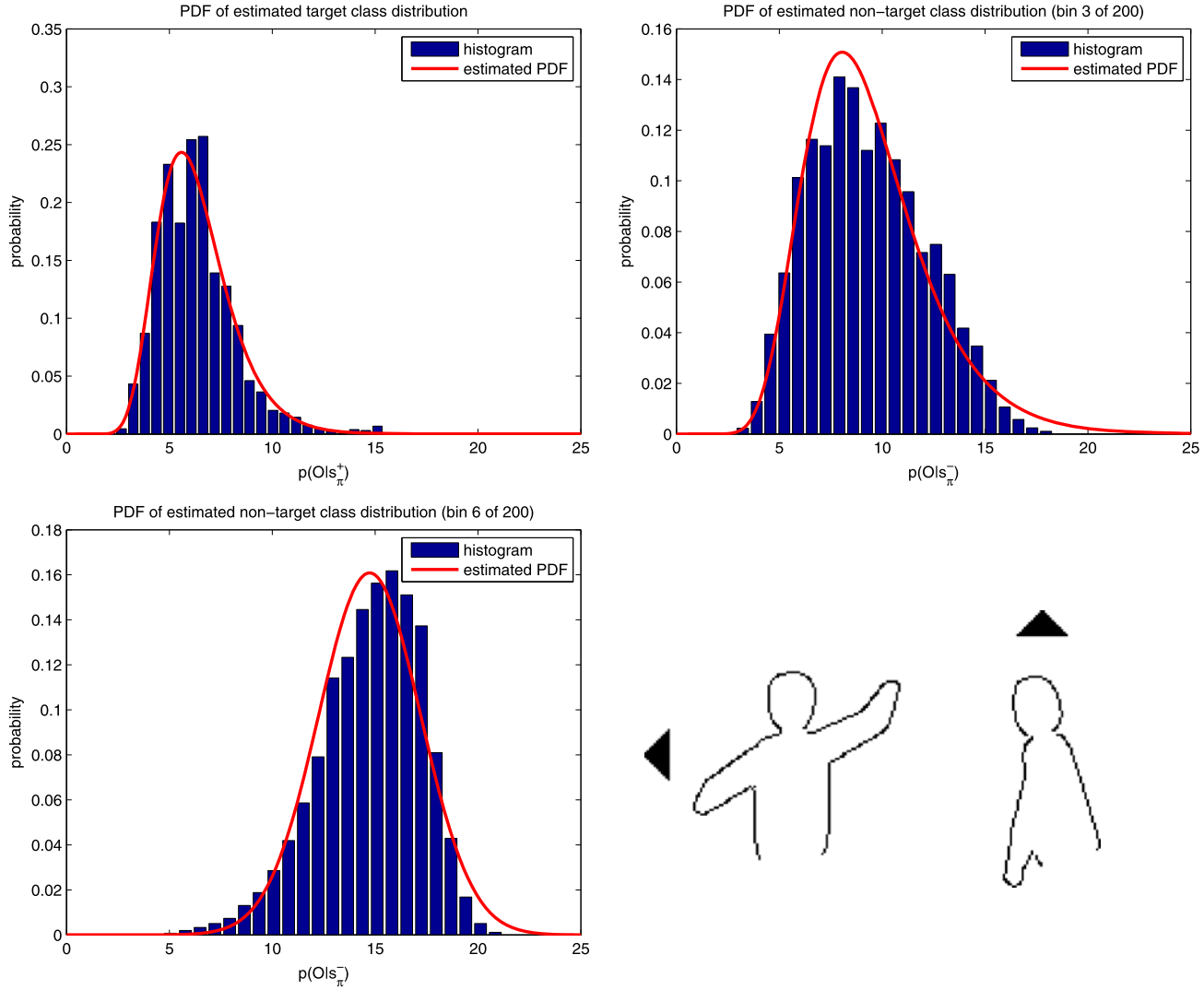



**Fig. 5** Example distribution $O|s_\pi^+$ (aggregated for first-level exemplars) and example distributions $O|s_\pi^-$ with associated first-level exemplars. For the latter, the expected value of $O|s_\pi^-$ for the more salient exemplar (*left*) is noticeably larger than that for the less salient exemplar (*right*)

class with the lowest mean squared error with respect to the learned histograms. The histograms associated with less salient templates exhibit a right-tailed shape and are well-fitted by log-normal distributions, while histograms associated with more salient templates are more symmetric or rather slightly left-tailed and are better fitted by normal distributions. Figure 5 shows the estimated distribution for the target class, as well as two of the estimated distributions for the non-target class, for a less salient and a more salient exemplar respectively.

Pose selection is implemented by extracting the poses of all matched exemplars for each camera (Sect. 3.4), evaluating the pose posterior probability (6) and ranking the aggregated list. Non-maximum suppression of the results removes lower ranked poses, where the similarity with a higher ranked pose is below a threshold $u_{PosSel}$. The pose similarity measure we use is the mean distance between corresponding 3D locations of the human body model

$$d_x(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2) = \frac{1}{|B|} \sum_{i \in B} d_e(\boldsymbol{v}_1^i, \boldsymbol{v}_2^i) \tag{7}$$

where $B$ is a set of locations on the human body model, $|B|$ the number of locations, $\boldsymbol{v}^i$ is the 3D position of the respective location in a fixed Euclidean coordinate system, and $d_e(.)$ is the Euclidean distance. For the set of locations, we choose torso and head center as well as shoulder, elbow and wrist joint location for each arm. After non-maxima suppression, the $K_{PosSel}$ best remaining solutions pass to the next processing stage.

Observe that pose selection by (6) has the desirable property that camera viewpoint selection is implicitly performed; "bad" viewpoints, which capture ambiguous poses

(e.g. hands self-occluded by torso), result in 2D pose exemplars with comparatively large $|\Pi_s|$ and thus in decreased 3D pose posteriors.

### 3.6 Multi-camera Hypothesis Verification and Optimization

See Fig. 1. After the previous pose selection stage, which ranks pose hypotheses $\boldsymbol{\pi}$ generated from 2D matches from individual cameras, we continue by verifying these over multiple cameras and placing them in a 3D world coordinate system (i.e. adding $\boldsymbol{x}$). This is implemented by the three steps described in the following subsections.

#### *3.6.1 Pose Verification Using 2D Shape Exemplars*

In this first—still entirely shape-based—step, we map a 3D pose generated by one camera to the corresponding 2D pose exemplars of the other cameras and match these onto their respective images. Due to the used orthographic projection, the mapping from a pose as observed in camera $c_i$ to the corresponding pose in camera $c_j$ is achieved by modifying the torso rotation angle $\psi_{torso}$ relative to the projected angle between cameras $c_i$ and $c_j$ on the ground plane. The mapping from a 3D pose to a 2D pose exemplar is then easily retrieved from a look-up table after re-discretizing the angle to one of the values present in the poses of our library $S$.

For each pose $\boldsymbol{\pi}$, we also need to obtain a 3D position $\boldsymbol{x}$ in the world coordinate system from the 2D location of the match on the image plane. We therefore backproject this location and sample 3D points at various depths, which then we project in the other camera images and match the corresponding exemplars at the locations where there is foreground support. The multi-view pose probability for a pose $\boldsymbol{\pi}$ is modeled as

$$p(\boldsymbol{\pi}|\boldsymbol{O}) = \prod_{c=1}^{C} p(\boldsymbol{\pi}|O_c) \tag{8}$$

where $\boldsymbol{O}$ is the set of observations from the three cameras and $p(\boldsymbol{\pi}|O_c)$ is the posterior term from (6). For each pose $\boldsymbol{\pi}$, the 2D location with the highest probability per camera is kept; triangulation of point pairs and averaging then yields a 3D position $\boldsymbol{x}$ in the world coordinate system. In case of an inconsistent triangulation being generated, i.e. if the distance between triangulated point pairs exceeds a certain threshold, the respective pose hypothesis is discarded.

To account for the error made by the orthographic projection assumption, we add a correction angle $\psi_t^{corr}$ to the torso rotation angle $\psi_t$, when converting the pose representation from orthographic to perspective projection, see Fig. 6. We obtain a ranked list of candidate 3D poses $\{\boldsymbol{\pi}, \boldsymbol{x}\}$ and, as in Sect. 3.5, perform non-maximum suppression

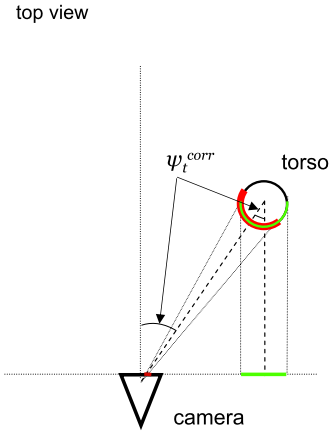



**Fig. 6** Correction angle $\psi_t^{corr}$ to the torso rotation angle $\psi_t$ when transferring poses from orthographic to perspective projection (Sect. 3.6.1)

based on threshold $u_{PosVerEx}$ for pose similarity (7). The best $K_{PosVerEx}$ remaining candidate poses pass to the next processing stage.

#### *3.6.2 Pose Verification by Rendering (Shape, Texture)*

In this processing stage, the candidate 3D poses are rendered on-line, assuming perspective projection, and ranked according to a multi-view likelihood based on both shape and texture cues

$$p_{ST}(\boldsymbol{O}|\boldsymbol{\pi}, \boldsymbol{x}) = p_S(\boldsymbol{O}|\boldsymbol{\pi}, \boldsymbol{x}) \times p_T(\boldsymbol{O}|\boldsymbol{\pi}, \boldsymbol{x}) \tag{9}$$

where $p_S(.)$ is the product $p(\boldsymbol{O}|\boldsymbol{\pi}, \boldsymbol{x}) = \prod_{c=1}^{C} p(O_c|\boldsymbol{\pi}, \boldsymbol{x})$ of chamfer distance-based likelihoods per camera, and $p_T(.)$ is a respective texture likelihood term (for a description of the distance measure, see (20)). While no texture model is available during the first frames, the multi-view matching likelihood is based on the shape component only; the texture likelihood is taken constant in this case.

Equation (9) represents a computationally expensive step in the evaluation cascade due to on-line graphical rendering across multiple camera views and the additional evaluation of the texture likelihood, but provides an accurate likelihood evaluation; poses are not approximated by a subset of shapes anymore, and the assumption of perspective projection is more realistic. As before, we perform non-maximum suppression based on threshold $u_{PosVerRdr}$ for pose similarity (7). The best $K_{PosVerRdr}$ remaining candidate poses pass to the next processing stage.

#### *3.6.3 Local Pose Optimization*

We overcome the limitation of our discrete, exemplar-based representation by performing a local optimization of the pose parameters in continuous space using the gradient $\nabla p(O|\boldsymbol{\pi}, \boldsymbol{x})$. For efficiency reasons, optimization

is only performed on a subset of the best-ranked results ($K_{PosVerRdrOpt}$) of the preceding processing stage. Rendering makes it difficult to compute gradients, due to the contour discretization to pixel level, which makes $\nabla p(O|\boldsymbol{\pi}, \boldsymbol{x})$ non-differentiable with respect to its parameters. In previous work (Hofmann and Gavrila 2009b), we computed a local gradient approximation using central differences. This has the disadvantage of making gradient computation costly and requiring the suitable setting of the difference delta constant (i.e. such that the discretized contour output changes yet the finite difference result is sufficiently accurate). In this paper, we perform a local shape-based optimization using the analytical gradient of a close approximation of the model contour, i.e. of densely sampled vertices on the contour.

Linear sampling of the two angle parameters in (1) results in a set of very unevenly spaced vertices, due to the varying local curvature of the object. However, we can obtain a regular sampling of vertices on a superquadric surface by using a first-order differential model (Pilu and Fisher 1995). Two neighboring vertices are defined to be part of the contour, if a sign change occurs in the dot product between surface normal and a ray through vertex and camera center. We exclude vertices that are occluded by other body parts in our model by making use of the inside-outside function of a superquadric.

Let $C$ denote the set of camera views, and $V^c_{\text{rim}}$ a set of model vertices on the projected contour in view $c$. Furthermore, let $d_{\text{ch}}(S_c, p)$ be a differentiable function that computes the real-valued chamfer distance between an image projection $p = (p_x, p_y)$ of a point on the model surface and the closest edge point of the silhouette in the scene image $S_c$, and $P^c$ be the camera projection matrix of camera view $c$. The average chamfer distance between model vertices and image silhouette contours is then computed as

$$D_C = \frac{1}{|C|} \sum_{c \in C} \frac{1}{|V^c_{\text{rim}}|} \sum_{v \in V^c_{\text{rim}}} d_{\text{ch}}\left(S_c, P^c \mathbf{H}^v \zeta^v\right) \tag{10}$$

$\mathbf{H}^v \zeta^v$ (see Sect. 3.2) denotes the 3D world coordinate position of vertex $v$ after transformation along the articulated chain. In our implementation, we compute $d_{\text{ch}}(S_c, p)$ as the bilinear interpolation of the distance transform image $T_c$ of $S_c$ (we use interpolation instead of accessing $T_c(p_x, p_y)$ directly in order to keep the operation differentiable).

$$\begin{aligned} d_{\text{ch}}(S_c, p) = {} & (1 - f_x)(1 - f_y) T_c(i_x, i_y) \\ & + f_x (1 - f_y) T_c(i_x + 1, i_y) \\ & + (1 - f_x) f_y T_c(i_x, i_y + 1) \\ & + f_x f_y T_c(i_x + 1, i_y + 1); \end{aligned} \tag{11}$$

where $(i_x, i_y)$ are the integer parts and $(f_x, f_y)$ are the fractional parts of $p$. The gradient of (10) with respect to the pose parameters is computed as follows:

$$\frac{\partial D_C}{\partial \{\boldsymbol{\pi}, \boldsymbol{x}\}} = \frac{1}{|C|} \sum_{c \in C} \frac{1}{|V^c_{\text{rim}}|} \sum_{v \in V^c_{\text{rim}}} J_d J_P J_{H^v} \zeta^v \tag{12}$$

Here, $J_d$ is the $1 \times 2$ Jacobian of the bilinear chamfer distance lookup function $d_{\text{ch}}(S_c, p)$ while $J_P \equiv J_P(P)$ is the $2 \times 3$ Jacobian of the projection operation, i.e. the multiplication with the camera matrix. $J_{H^v}$ is the Jacobian containing the partial derivatives of the entries of the cumulative transformation matrix $\mathbf{H}^v$ with respect to the pose parameters.

Because we sum over a number of sampled vertices in (10) and (12), both functions are not smooth over the whole parameter space, despite being differentiable at each point. Although second-order optimization methods such as Levenberg-Marquardt or quasi-Newton methods give the promise of fast convergence, they cannot cope with noisy gradients sufficiently well, leading to convergence failure. We instead optimize the value of (10) using a first-order Gradient Descent method with local step-size adaptation, as outlined in Bray et al. (2007). The Gradient Descent update equation for a parameter vector $\mathbf{q}_i$ of a function $E(\mathbf{q}_i)$ from iteration $i$ to iteration $i + 1$ is

$$\mathbf{q}_{i+1} = \mathbf{q}_i - \mathbf{a}_i \otimes \mathbf{g}_i, \quad \mathbf{g}_i = \frac{\partial E(\mathbf{q}_i)}{\partial \mathbf{q}_i} \tag{13}$$

where $\otimes$ denotes a component-wise product and $\mathbf{a}_t$ is a step-size vector of local learning rates which is updated by a meta-level descent on the step-sizes.

$$\mathbf{a}_i = \mathbf{a}_{i-1} \otimes \exp(\mu_g \mathbf{g}_i \otimes \mathbf{v}_i) \tag{14}$$

$$\mathbf{v}_{i+1} = \lambda_g \mathbf{v}_i + \mathbf{a}_i \otimes \left(\mathbf{g}_i - \lambda_g \mathbf{v}_i\right) \tag{15}$$

For efficiency reasons, we opt to update the gradient trace $\mathbf{v}_i$ as an exponential average of past gradients, as opposed to evaluating a multiplication of the Hessian with a vector (Bray et al. 2007) (the additional cost would be comparable to an additional gradient evaluation). The optimization process terminates after a maximum number of iterations is reached (in our case, 80) or if $\|\mathbf{g}_i\| < \epsilon$. For each optimized hypothesis, the shape-texture likelihood of (9) is computed and the set of all hypotheses is re-ranked accordingly.

### 3.7 Temporal Integration and Prediction

See Fig. 1. After executing the single-frame pose recovery stages outlined in Sects. 3.4 through 3.6.3, we obtain a number of pose hypotheses ranked by their multi-view observation likelihood (see (9)). The following step disambiguates these multiple hypotheses over time and determines pose trajectories that both match the observations well and exhibit coherent motion. We formulate this as the

following optimization task: given a sequence of observations $\boldsymbol{O}_0, \ldots, \boldsymbol{O}_T$ up to time step $T$, find the pose sequence $(\boldsymbol{\pi}_0, \boldsymbol{x}_0), \ldots, (\boldsymbol{\pi}_T, \boldsymbol{x}_T) \in \Pi_0 \times X_0, \ldots, \Pi_T \times X_T$ that maximizes

$$p(\boldsymbol{\pi}_{0:T}, \boldsymbol{x}_{0:T} | \boldsymbol{O}_{0:T}) \propto \prod_{t=1}^{T} p(\boldsymbol{\pi}_t, \boldsymbol{x}_t | \boldsymbol{\pi}_{t-1}, \boldsymbol{x}_{t-1}) \prod_{t=0}^{T} p(\boldsymbol{O}_t | \boldsymbol{\pi}_t, \boldsymbol{x}_t). \quad (16)$$

$p(\boldsymbol{\pi}_t, \boldsymbol{x}_t | \boldsymbol{\pi}_{t-1}, \boldsymbol{x}_{t-1})$ denotes the pose transition likelihood as a first-order Markov chain, while $p(\boldsymbol{O}_t | \boldsymbol{\pi}_t, \boldsymbol{x}_t)$ is the observation likelihood of (9). For the transition model, we evaluate the distances of a set of 3D locations on the model between a pose hypothesis and its predecessor in the previous frame. More specifically, let $d_k$ be the magnitude of displacement of body part $k$, from $t-1$ to $t$; we then compute the transition likelihood as follows:

$$\begin{aligned} &p(\boldsymbol{\pi}_t, \boldsymbol{x}_t | \boldsymbol{\pi}_{t-1}, \boldsymbol{x}_{t-1}) \\ &\quad = p(d_{\text{torso}}) \times p(d_{\text{l.shoulder}}) \times p(d_{\text{l.elbow}} | d_{\text{l.shoulder}}) \\ &\qquad \times p(d_{\text{l.wrist}} | d_{\text{l.elbow}}) \times p(d_{\text{r.shoulder}}) \\ &\qquad \times p(d_{\text{r.elbow}} | d_{\text{r.shoulder}}) \times p(d_{\text{r.wrist}} | d_{\text{r.elbow}}) \end{aligned} \quad (17)$$

For improved accuracy, we condition the displacement of the elbow on that of the adjoining shoulder. Note that the above decomposition only relates to the computation of the transition probability, in order to cope with the limited amount of training data (see Sect. 4); all pose parameters are estimated jointly in the current exemplar-based approach.

The type of problem formulated in (16) is solved by application of the Viterbi algorithm (Rabiner 1989) on the input data, classically used in a post-processing step for state disambiguation. In our case, we compute (16) "on-line" for each frame in a sliding window over the last $T$ frames. Furthermore, we use a parallel List Viterbi Algorithm (LVA) (Seshadri and Sundberg 1994) implementation to compute not only the optimal, but the $K_{traj}$ best trajectories through the Viterbi trellis at each time step. For our experiments, we chose $K_{traj} = 500$; we found that this is large enough a number to ensure sufficient trajectory diversity and small enough to keep all trajectories in system memory.

We generate $\frac{1}{2} K_{PosVerRdr}$ pose predictions at every time step that augment the detections of the next time step (i.e. to obtain half as many predictions as detections), as indicated in Fig. 1; these are generated using whole trajectory information. To generate predictions at time step $t$, the desired number of trajectories is sampled with replacement from the set of best trajectories with a probability proportional to the trajectory likelihood determined by the LVA algorithm.

For each trajectory sample, all 3D joint locations are independently filtered over the current length of the trajectory using a Kalman filter and a constant acceleration dynamical model. Given the current state, we perform the prediction update of the Kalman filter for each joint location; a new joint location is sampled from the predicted states and their estimated covariances, correspondingly. We obtain a pose prediction $(\tilde{\boldsymbol{\pi}}_{t+1}^k, \tilde{\boldsymbol{x}}_{t+1}^k)$ from the set predicted 3D joint locations by inverse kinematics and constrained nonlinear optimization (Marquardt 1963), with the last trajectory pose as an initialization.

Note that state estimation is no longer constrained to the discrete space, due to the local pose estimation of Sect. 3.6.3 and the above-mentioned prediction mechanism.

Given the high dimensionality of the state space and the weak motion model (arbitrary human movement), we opted for the above sliding window batch-mode framework rather than a recursive framework, because of increased estimation stability. A temporary breakdown in detection can be better bridged by the selected approach. Furthermore, the approach allows an automatic (re-)initialization of system, after a prolonged failure of detection, in cases where recursive filtering frameworks such as particle filtering would completely lose track.

### 3.8 Model Adaptation Using Texture Information

See Fig. 1. An additional component in our system is dedicated to augmenting our shape model with texture information in order to increase the discriminative power of hypothesis verification (see Sect. 3.6.2). Generally, the quality of the learned texture model is sensitive to the estimated pose of the shape model, and matching with a wrong texture model can be damaging for pose estimation. In order to avoid incorrect texture model updates as much as possible, we decided not to perform these based on pose estimates at a single time instant, but rather based on the more reliable trajectory information computed in the previous section. We currently maintain a texture model learned from the optimal trajectory at each time step and start acquiring the model after a constant, user-defined number of frames (15 frames in our experiments). To acquire the input as exactly as possible, we perform local pose optimization prior to acquisition in case this had not been done before, i.e. if the pose on the best trajectory at the current time step was not among the $K_{PosVerRdrOpt}$ best poses as described in Sect. 3.6.3.

We obtain a texture map for each major body part (torso, head, upper arm, lower arm) by sampling the visible area of our model primitives (see Sect. 3.2) for each camera view and storing the color values in a 2D texture image. During acquisition we ensure that we do not sample in areas of self-occlusion through other body parts by performing collision detection on the ray from camera center to the points on the respective superquadric. The texture images from each camera are then combined by choosing for each pixel the sampled value for which the angle between the superquadric

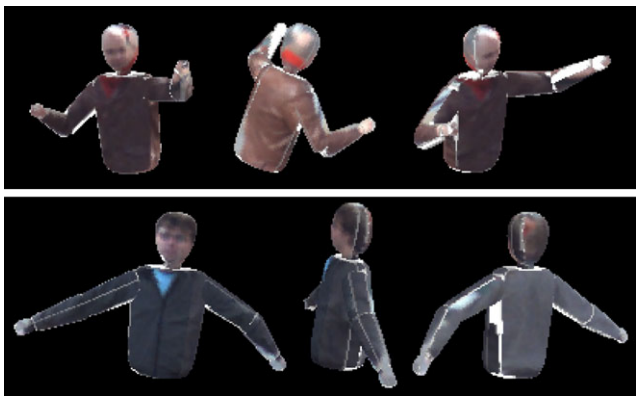


**Fig. 7** Two examples of shape model enriched with texture information, rendered from various viewpoints. Parts of the body that are occluded in all cameras stay untextured and are shown *in white*. Depicted color space is non-normalized RGB

normal vector and the ray from the camera center is smallest. Figure 7 shows an example of a reprojected texture map acquired from the depicted pose. Because images from different cameras are effectively stitched together during the acquisition of a texture map, there will be differences in luminance due to camera properties and scene illumination. We reduce the variation induced by global indirect illumination by working in a normalized RGB color space $r = R/L$, $g = G/L$, $b = B/L$, where $L := \frac{1}{|K|}\sum_{k\in K}(R_k + G_k + B_k)$ is the average luminance over the scene pixels $K$. This normalization can account mainly for differences in color calibration of the cameras or global illumination.

Each color pixel of a texture map is represented by a multivariate isotropic normal distribution whose parameters are updated by causal filter equations

$$\tilde{\mu}_t = (1-\alpha)\tilde{\mu}_{t-1} + \alpha X_t \tag{18}$$

$$\tilde{\sigma}_t^2 = (1-\alpha)\tilde{\sigma}_{t-1}^2 + \alpha(X_t - \tilde{\mu}_t)^T(X_t - \tilde{\mu}_t) \tag{19}$$

where $X_t$ is the measurement from the acquired texture map at time $t$ and $\alpha$ is a learning rate factor. Figure 8 shows an example of the texture model adaptation over time. The filtering of the texture map allows incorrect estimates to be smoothed out. Although the resulting texture map is quite blurred, it is nevertheless beneficial for pose recovery, as we will see in the experiments.

In the hypothesis verification stage (see Sect. 3.6), the texture map of a pose hypothesis is compared to that of the model by computing the average pixel-wise Mahalanobis distance:

$$d_{\text{texture}} = \frac{1}{R_b}\sum_{i=1}^{R_b}\sqrt{(X-\mu)^T\Sigma(x-\mu)} \tag{20}$$

where $R_b$ is the resolution of the texture map associated with body part $b$.

**Fig. 8** Progression of texture model adaptation over time (frames 15, 20, 50, 250) on an example sequence; shown is the pixel-wise mean of the model components. The pose used for the first acquisition (frame 15) is *depicted below*; in this case, each pixel of the model is initialized with a user-defined variance

## 4 Experiments

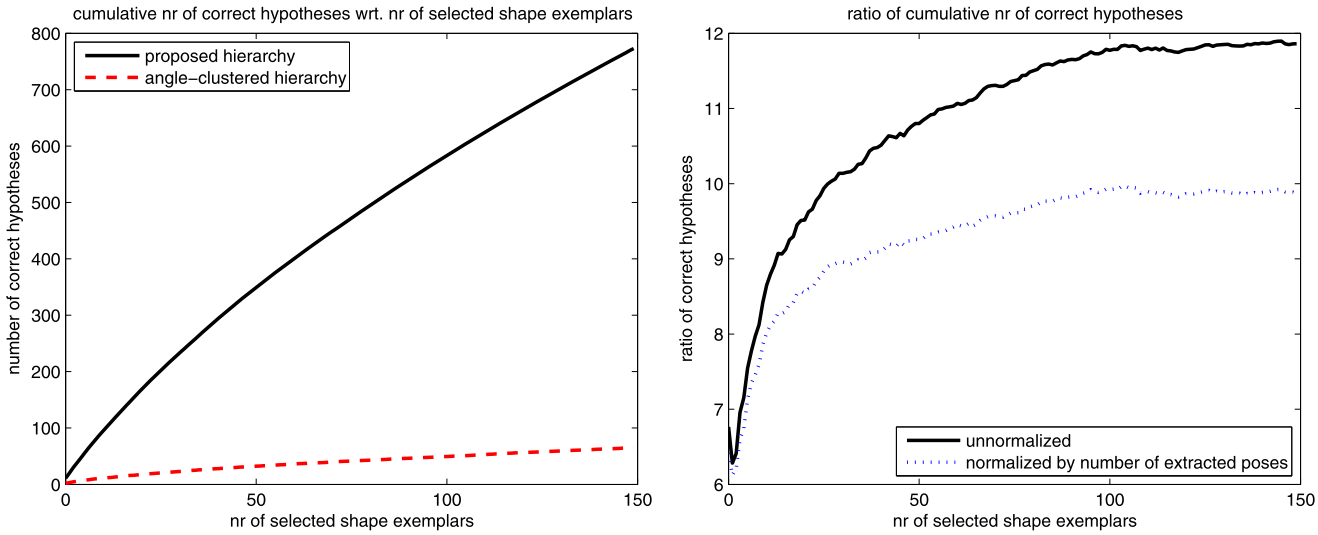
Our experimental data consists of recordings from three synchronized color CCD cameras looking over a train station platform. In 12 sequences (about 10 s on average, captured at 20 Hz), various actors perform unscripted movements, such as walking, gesticulation and waving. The setting is challenging; the movements performed contain a sizable amount of torso turning, the background is cluttered and non-stationary (people are walking in the background, trains are passing by), furthermore, there are appreciable lighting changes. The realism of the dataset in the context of surveillance was the key motivation for using it as our primary dataset for evaluation. The evaluation methodology we use is similar to Sigal et al. (2010), both in terms of the 3D pose error metric (7) and the baseline algorithm (Deutscher and Reid 2005). Our data is made public to facilitate benchmarking.[2]

Cameras were calibrated using (Bouguet 2003); this enabled the recovery of the ground plane. Ground truth pose was manually labeled for all frames of the data set. Considering the quality of calibration and labeling, we estimate the ground truth accuracy to be within 4 cm. We used a single generic human model to capture the three male adults in the scene (one male wore two different outfits). To obtain an indication of the shape variations involved, we handfitted personalized human-outfit models and found the average vertex distance between the used generic and the tuned reference models to be 3.2 cm, 2.6 cm, 2.9 cm and 3.7 cm, respectively, in a canonical pose (i.e. arms stretched laterally, at 90° elevation).

All distributions described in Sects. 3.4.3, 3.5 and 3.6 were learned by a leave-one-out approach for each test sequence. The generic motion model (Sect. 3.7) was derived from the aggregated CMU *MoCap* data[3]; after some conver-

[2] The data set is made available for non-commercial research purposes. Please follow the links from http://isla.science.uva.nl/ or contact the second author.

[3] http://mocap.cs.cmu.edu/.




(a) (b)

**Fig. 9** (**a**) Number of correct pose hypotheses with respect to the number of top 2D pose exemplars. (**b**) Ratio of the number of correct pose hypotheses between both trees. Both (**a**) and (**b**) obtained by averaging over all frames and cameras

sions to the degrees of freedom of our shape model and our frame rate, the latter yielded 756,844 frames for training.

### 4.1 Evaluation of 2D Pose Exemplar Tree Representation

We first evaluate the quality of the constructed 2D pose exemplar tree, which was created using the hybrid clustering approach discussed in Sect. 3.4.2. For this, we compare it to a tree clustered in joint angle parameter space by means of a hierarchical $k$-means algorithm, as proposed for example in Stenger et al. (2006) in the context of hand tracking. The trees contain the same exemplars at the leaf level $S_L$ and have the same number of exemplars $|S_l|$ at each level $l$. The quality of the tree structures is assessed by performing the single-view pose hypothesis generation stage (Sect. 3.4) with the same tree-level specific thresholds $\tau_l$, and by evaluating the 3D pose hypotheses associated with the top ranked 2D pose exemplars. In the following comparisons, we regard a 3D pose hypothesis as "correct" if the average pose error to the ground truth (7) is less than 10 cm.

Figure 9(a) shows the number of correct pose hypotheses in relation to the number of top ranked 2D pose exemplars for both tree structures. The benefit of our hybrid tree clustering algorithm is clear: we obtain about one order of magnitude more correct poses compared to the tree clustered in joint angle parameter space. To elaborate on this comparison, Fig. 9(b) shows the ratio of the number of correct poses between both trees (i.e. nr. of correct poses in proposed tree divided by nr. of correct poses in angle-clustered tree); it saturates at a value of about 12. We additionally plot the same ratio normalized by the number of extracted poses associated with the top ranked 2D pose exemplars in either tree. The proposed tree structuring still generates about 9.5 times more correct hypotheses.

The considerably worse performance of tree construction by clustering joint angles can be explained by the fact that equal distance in joint angle space does not imply equal appearance similarity. Small changes of some angles, in particular of the torso twist angle $\psi_{torso}$, will have a large effect on the projected silhouette given the arms that are extended and visible. On the other hand, there will be no effect on the projected silhouette if the extended arms are self-occluded by the torso. Appearance-based clustering results in a representation that covers pose parameters space non-uniformly to account for these effects.

### 4.2 Evaluation of Pose Hypothesis Verification

We proceed with a quantitative analysis of the pose hypothesis verification component of our system (see Sect. 3.6). Figure 10 shows the average pose error of the most correct solution among the $K$ best-ranked solutions, averaged over the frames of the data set (the most correct solution minimizes the pose error, i.e. similarity to ground truth by (7)). We see the benefit of our cascaded pose recovery approach (Sects. 3.6.1–3.6.3) in the successive decrease of the average pose error. Figure 10 also shows that, if we are able to select the correct solution among the 10 (50) best ranked solution, we have the potential to reduce pose error to 8 (7) cm. Of course in practice, we do not know which solution is most correct, and pose errors produced by the overall system tend to be larger, see next subsection.

We now evaluate the performance of the local pose optimization stage (see Sect. 3.6.3) separately. To this end,

we created 2200 test input poses from 10 different images of our data set by random perturbations $\boldsymbol{\pi}_{\mathrm{GT}} + N(0, \Sigma)$ of the ground truth pose $\boldsymbol{\pi}_{\mathrm{GT}}$, with varying covariances $\Sigma$. See Fig. 11, we observe a clear benefit of pose optimization in reducing the mean pose error.

### 4.3 Evaluation of Overall System

Figure 12 depicts examples of estimated poses after running the whole system using shape and texture cues, taken from the best trajectory. A quantitative evaluation in terms of the deviation between estimated and ground truth 3D pose over the entire dataset is given in Table 1. As can be seen, by adding pose hypothesis predictions (Sect. 3.7) and texture-based model adaptation (Sect. 3.8) to the pose recovery framework, we achieve a reduction of the mean pose error over our dataset from 10.7 cm to 9.5 cm, on average. Table 2 shows the average pose error at particular joint locations. It can be observed that the pose error increases from the root of the articulated structure (torso) to the extremities (elbows).

We also compared the different instantiations of our system with the hierarchical Partitioned Annealed Particle Filter (PAPF) (Deutscher and Reid 2005), a state-of-the-art technique for tracking high-DOF (unconstrained) articulated movement. Unlike Shape-from-Silhouette approaches, it does not require perfect silhouette segmentation. In order to focus on the essential differences, we implemented the PAPF using the same foreground segmentation (Sect. 3.3), shape-based likelihood computation (9) and motion model data (CMU *MoCap*) for initializing the diffusion covariance. After some tuning, we selected a parameterization with 4 layers for our 13 DOF model (cf. 10 layers for a 30 DOF model in Deutscher and Reid 2005) and 200 particles per layer, as in Deutscher and Reid (2005). The PAPF was initialized with the ground truth in the first frame of each sequence, while our system does not rely on manual initialization.

In our experiments, we observed a good performance of the PAPF on many sequences. However, for more diffi-

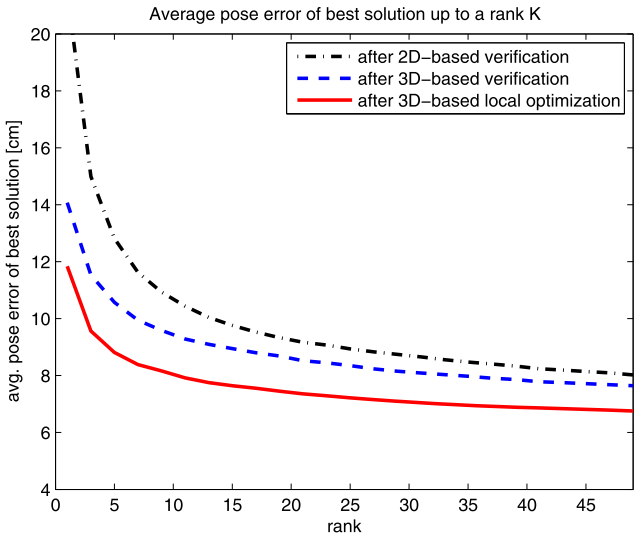
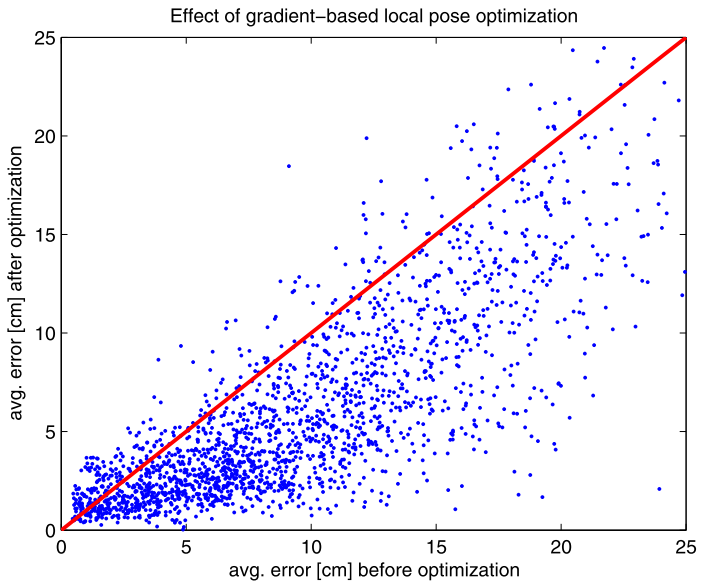



**Fig. 10** Mean pose error of the best solution among the $K$ best-ranked, averaged over the frames of the data set, after processing stages of Sects. 3.6.1–3.6.3




(a)

(b)

**Fig. 11** (**a**) Plot of the mean pose error in cm (7) before and after gradient-based local pose optimization (Sect. 3.6.3). (**b**) Example of local pose optimization, before (*top row*, avg. error 8.7 cm) and after (*bottom row*, avg. error 5.9 cm)

**Fig. 12** Examples of recovered poses in the three camera views for multiple persons (best trajectory, shape and texture cues). Shown are image cut-outs

**Table 1** Average mean pose error ((7), in cm) and standard deviation over 12 test sequences (S: shape, T: texture)

| | Avg. mean pose error | Std. dev. |
|---|---|---|
| Our system (S & T, det. & pred.) | 9.5 | 4.1 |
| Our system (S, det. & pred.) | 10.3 | 4.8 |
| Our system (S, detections only) | 10.7 | 5.4 |
| PAPF (Deutscher and Reid 2005) | 14.4 | 7.5 |

cult sequences (appreciable background clutter, ambiguous poses, fast torso turning), we observed that the PAPF particles diverted away from the correct solution after a while, with little chance for recovery. Unlike the particle filtering approach, our system is inherently able to re-initialize after temporary likelihood ambiguities, due to the single-frame pose detection component that yields candidate poses independently generated at every time step. On average, our proposed approach outperformed PAPF considerably (avg. pose error down to 9.5 cm vs. 14.4 cm), even though the latter had been initialized with the ground truth pose.

Figure 13 provides a closer look at the timeline of a challenging tracking sequence with two 360° torso turns in short succession (frames 70–150 and 180–340). The figure shows the pose error (7) over time for the best trajectory using the system configurations listed in Table 1, as well as a comparison with the trajectory obtained by the PAPF approach. The greyish background captures the distribution of the pose error over the poses obtained (i.e. by detection, or by prediction from a previous time step); lighter shades indicate higher densities. For example, one observes in Fig. 13 a whitish band for the first 150 frames for pose error interval 35–45 cm; this corresponds to a cluster of solutions

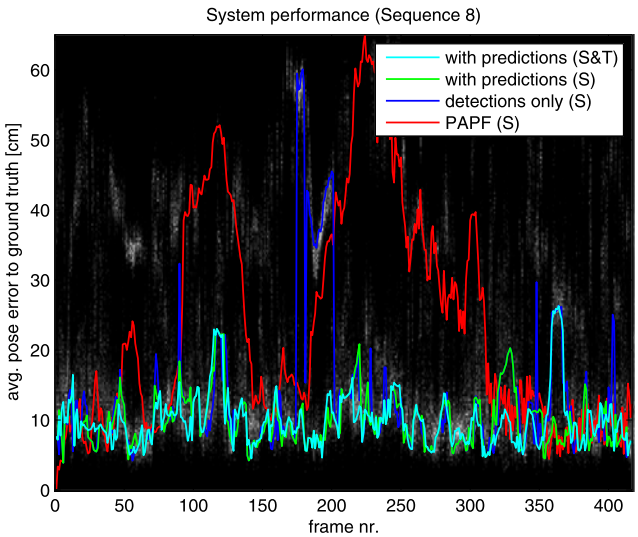



**Fig. 13** Pose error (in cm) for best trajectory for three system configurations (with and without prediction generation; S: shape, T: texture) and for the PAPF. The background shows histograms of the pose distance of the single-frame detections per time step (*lighter shades* indicate higher densities)

for which the torso twist is misaligned by 180°. In this sequence, the Viterbi-based approaches are able to track the two 360° torso turns, whereas the PAPF estimates the torso orientation almost unchanged. It is noteworthy that no other parameterization of the PAPF algorithm we tried was able to improve on this; for example, scaling up the diffusion covariance to generate more diverse particles leads to loss of track even earlier due to drift. We take this as an example of the increased robustness of the proposed trajectory-based estimation which combines multi-hypothesis detection and prediction. Without the additional prediction gen-

**Table 2** Average pose error (summands of (7), in cm) and standard deviation over 12 test sequences using our proposed system, for each measured location

| | Torso ctr. | Head ctr. | R. shoulder | R. elbow | R. hand | L. shoulder | L. elbow | L. hand |
|---|---|---|---|---|---|---|---|---|
| Avg. pose error | 5.53 | 6.35 | 7.00 | 11.73 | 15.18 | 6.91 | 10.79 | 12.32 |
| Std. dev | 2.50 | 2.96 | 3.59 | 8.33 | 11.09 | 3.47 | 6.58 | 8.49 |

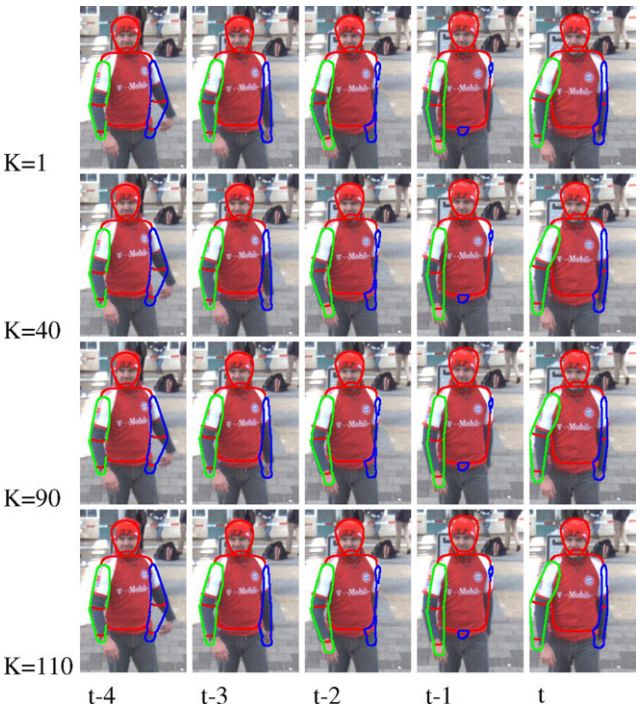



**Fig. 14** Visualization of variously ranked Viterbi trajectories within the $K_{traj}$ best trajectories computed, at time step $t = 189$ (one camera view only). *Left arm*: *green*, *right arm*: *blue*. *From left to right*: poses at various time steps. *From top to bottom*: poses at various ranks

eration mechanism (Sect. 3.7), our system can only use the detections provided by our single-frame pose recovery. Using this setting, we can see a temporary starvation of correct detections around frame 175 (and, to a lesser degree, at frames 350 and 400) due to ambiguous likelihood measurement in the hypothesis generation stage; using the two system configurations that include predictions, these frames are correctly "bridged".

Figure 14 shows Viterbi trajectories at different ranks ($K = 1, 40, 90, 110$) within the $K_{traj}$ best trajectories computed. Shown is time step $t = 189$ of the same sequence, during a torso turn where there is considerable ambiguity regarding the torso orientation (see also Fig. 13). Our multiple trajectory representation captures the pose diversity associated with this ambiguity and more. Note that the "best" Viterbi trajectory at this time step represents an incorrect motion with the upper body turned around 180°. The trajectory with the smallest pose error appears at rank 110 (observe the color coding of the arms). The ambiguity is

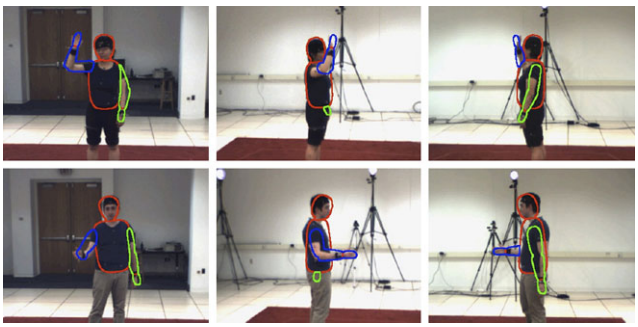


**Fig. 15** Subjects S1 and S2 from the HumanEva-I data set (3 camera views shown each)

eventually removed after processing the observations of further time steps.

We also performed experiments with our system on the HumanEva-I data set (Sigal et al. 2010), which involves an indoor setting and one subject per scenario. The two subjects S1 and S2 of this data set (see Fig. 15) differ appreciably in appearance, and while we could use our generic male adult shape model for male subject S2, we had to manually adjust the model to a smaller figure for female subject S1 in order to produce usable tracking results (average vertex distance of S1 and S2 to our generic male adult shape model was 9.5 cm and 4.2 cm, respectively, in canonical pose with hands stretched out). We kept the appearance-based grouping of the tree hierarchy exemplars as described in Sect. 3.4.2 and just re-rendered the exemplars using the adapted shape model (as opposed to a complete new clustering). We kept the various probability models that were learned from our own data set and did not re-estimate them for the HumanEva-I data.

Table 3 contains quantitative results from these experiments. As can be seen, the errors achieved are slightly lower than those obtained on our own data set. We attribute this to the controlled indoor conditions and to the closer camera positioning to the subjects. Tracking errors of 8.1 cm and 11.2 cm were reported in Peursum et al. (2010) for the respective walking sequences of S1 and S2 using Annealed Particle Filtering (Deutscher and Reid 2005). Further literature listing errors on either HumanEva-I and II data sets include Brubaker et al. (2010), Corazza et al. (2010), where errors of approx. 6 cm and 8 cm are reported. Gall et al. (2010) and Lee and Elgammal (2010) report significantly lower errors around 3–4 cm; however in the latter paper, a strong motion model for walking is enforced.

**Table 3** Quantitative results of evaluation on HumanEva-I data

| Subject | Sequence | Mean pose error [cm] |
|---|---|---|
| S1 | Gestures-1 | 6.3 |
| S1 | ThrowCatch-1 | 5.8 |
| S1 | Walking-1 | 4.7 |
| S2 | Gestures-1 | 6.5 |
| S2 | ThrowCatch-1 | 8.9 |
| S2 | Walking-1 | 7.4 |

**Table 4** Approximate breakdown of computation time onto the separate system stages (see Sect. 5)

| Stage | Section | Time spent |
|---|---|---|
| Preprocessing | 3.3 | 2% |
| 2D exemplar matching | 3.4.3 | 22% |
| Pose selection | 3.5 | 2% |
| Exemplar-based verification | 3.6.1 | 13% |
| Rendering-based verification | 3.6.2 | 25% |
| Local pose optimization | 3.6.3 | 32% |
| Temporal integr. & model adapt. | 3.7, 3.8 | 4% |

## 5 Discussion

The experiments demonstrated a new quality to 3D pose estimation for arbitrary (single) human movement in a noisy and uncontrolled environment. The remaining failure mode of the system concerns prolonged "ambiguous" poses with the hands close to the torso; the silhouette-based approach stands little chance in recovering exact hand position, furthermore, most clothing does not contain appreciable texture differences between torso and arms. Further work could add the segmentation-of and matching-with the weak internal edges to the later process stages.

We implemented the system in unoptimized $C++$ code; it currently requires about 30–40 s per frame (i.e. total over three cameras at a timestamp) to recover 3D pose on a 3 GHz Intel PC. This is not fast in absolute terms, but it seems to compare favorably with previously reported processing speeds in literature concerning 3D pose recovery with generative models against non-stationary background (e.g. Balan and Black 2006; Gall et al. 2010; Kohli et al. 2008; Lee and Cohen 2006; Lee and Nevatia 2009), yet direct comparisons are difficult due to the differing types of movement considered (for example, unconstrained upper body movement vs. whole-body walking). Computing time approximately breaks down onto the separate processing stages as given in Table 4. The main processing bottlenecks are the 2D pose exemplar matching (Sect. 3.4.3), the pose verification by graphical rendering (Sect. 3.6.2) and local pose optimization (Sect. 3.6.3) stages. These stages could be parallelized, allowing for a near-linear reduction of processing time with available CPU/GPU cores. However, we did not exploit parallelism in our implementation, neither at the camera-, nor at the pose solution- or vertex-level. The only exception to this was local pose estimation, where the pose solutions were distributed over the available four processor cores.

Another way to improve system efficiency is by parameter setting. We did not spend major time on parameter tuning and selected conservative truncation thresholds (e.g. $\tau_1, \ldots, \tau_L$, $K_{PosSel}$, $K_{PosVerEx}$, $K_{PosVerRdr}$) for the various processing stages. Gavrila and Munder (2007) describe an automatic procedure to optimize parameters in a multi-stage cascade architecture, based on successive ROC optimization. All in all, significant optimization potential remains.

Future work involves the recovery of whole-body pose and that of multiple, potentially occluding, people. An extension of the chosen exemplar-based approach to whole-body recovery (or to multiple body models for capturing a larger variety of people) is straightforward. It will however, at least for the near future, require coarser joint angle discretizations, in order to keep the additional memory requirements in check. Its effect on pose recovery accuracy remains to be investigated. We do expect a future adaptation of a generic 3D human shape model (see work by Balan et al. 2007; Gall et al. 2009) to the particular person in the scene to result in more accurate likelihood estimation, alleviating some of the above-mentioned pose discretization effect.

Multi-person pose recovery could be achieved by executing the system multiple times on all recognized 3D blobs in the image pre-processing stage (see Sect. 3.3). The system could be built on top of a more sophisticated (multi-) 3D blob tracking approach, e.g. Fleuret et al. (2008), Liem and Gavrila (2009), to better handle track consistency in case of multiple persons being in close vicinity, i.e. avoidance of ID changes.

## 6 Conclusion

We presented a framework for unconstrained 3D human upper body pose estimation from multiple camera views in a complex environment. The main novelty lies in the integration of three components: single-frame pose recovery, temporal integration and model texture adaptation. The second contribution concerns the way single-frame pose recovery is performed: hypotheses are generated in each camera independently based on probabilistic hierarchical shape matching, and information is fused at the pose parameter level in an efficient, multi-stage process.

We demonstrated an improvement versus the state-of-the-art in a dozen of challenging real-world sequences depicting different actors performing unscripted movements. Further work will be necessary to deal with multiple people under occlusion.

**Acknowledgements** The authors would like to thank John Schavemaker (TNO) for stimulating discussions throughout this research. This research was in part funded by the MultimediaN project.



## Appendix A: Main Systems Parameters and Their Settings

### A.1 Overall

- $|C| = 3$. Number of cameras

### A.2 Image Pre-processing (Sect. 3.3)

- $h_{torso} = 70$ cm

### A.3 Pose Exemplar Representation (Sects. 3.4.1 and 3.4.2)

- Table 5 lists upper- and lower- bounds, and step sizes for the pose space discretization (Sect. 3.4.1)
- Non-colliding 3D poses are considered "allowable" (Sect. 3.4.1) if they fulfill following inequalities

$$\theta_s \leq 1.5\,\theta_e + 310^\circ \tag{21}$$
$$\theta_s \leq -0.5\psi_s + 110^\circ \quad \text{if } \psi_s > 40^\circ$$
$$\phi_s \geq 3\,\psi_s - 160^\circ$$

where $(\phi_s, \theta_s, \psi_s)$ are the Euler angles (in degrees) of the shoulder joint and $(\theta_e)$ is the elbow joint angle, see Fig. 2. The above bounds were determined experimentally

**Table 5** Pose space discretization

| Angle | Upper bound | Lower bound | Nr steps |
|---|---|---|---|
| $\phi_t$ | 75 | 90 | 3 |
| $\theta_t$ | 80 | 100 | 3 |
| $\psi_t$ | −180 | 157.5 | 16 |
| $\phi_h$ | −30 | 0 | 2 |
| $\psi_h$ | −30 | 30 | 3 |
| $\phi_s^l, \phi_s^r$ | −45 | 144 | 8 |
| $\theta_s^l, \theta_s^r$ | 15 | 148 | 8 |
| $\psi_s^l, \psi_s^r$ | −80 | 100 | 9 |
| $\theta_e^l, \theta_e^r$ | 0 | 140 | 6 |

- $N^* \approx 20000$ Maximum number of 2D pose exemplars that can still be practically handled with a partitional clustering algorithm of quadratic complexity, on a particular hardware (Sect. 3.4.2)
- $B = 10$ Branching ratio of tree of 2D pose exemplars (non-leaf level)
- $L^* = 3$ Level of tree that is constructed with a partitional clustering algorithm of quadratic complexity. Value is determined by $N^*$ and $B$
- $|S_l|$ Number of 2D pose exemplars at tree level $l$. $|S_1| \approx 200$, $|S_2| \approx 2000$, $|S_3| \approx 20000$, and $|S_4| \approx 150000$ (Sect. 3.4.2)

### A.4 Single Frame Pose Recovery (Sects. 3.4.3–3.6.3)

- $\gamma_l$ Image grid size (in pixel) for interest locations for matching at tree level $l$. $\gamma_1 = 6$ px, $\gamma_2 = 3$ px, $\gamma_3 = 1$ px, $\gamma_4 = 1$ px (Sect. 3.4.3).
- $\tau_l$ Threshold on posterior for nodes at tree level $l$. $\tau_1 = 0.01$, $\tau_2 = 0.016$, $\tau_3 = 0.02$, $\tau_4 = 0.024$ (Sect. 3.4.3).
- $u_x$ Area of non-maximum suppression at process stage $x$, before truncation. In particular,
  $u_{PosHyp} = 2$ pixels (Sect. 3.4.3),
  $u_{PosSel} = 1.25$ cm (Sect. 3.5),
  $u_{PosVerEx} = 1.75$ cm (Sect. 3.6.1)
  $u_{PosVerRdr} = 0$ cm (no non-maximum suppression) (Sect. 3.6.2)

- $K_x$: Number of hypotheses that are generated by process stage $x$, after truncation. In particular,
  $K_{PosHyp} \approx 300000$ value is determined by $\tau_1 \cdots \tau_4$ (Sect. 3.4.3),
  $K_{PosSel} = 30000$ (Sect. 3.5),
  $K_{PosVerEx} = 2000$ (Sect. 3.6.1),
  $K_{PosVerRdr} = 800$ (Sect. 3.6.2), and
  $K_{PosVerRdrOpt} = 50$ (Sect. 3.6.3)

### A.5 Temporal Integration (Sect. 3.7)

- $T = 50$ Number of image frames of time interval for which the best trajectories are computed
- $K_{traj} = 500$ Number of best trajectories obtained by List-Viterbi algorithm

### A.6 Texture-Based Model Adaptation (Sect. 3.8)

- $\alpha = 0.1$ Learning rate pixel-based texture adaptation
- $R_b$ Texture model resolution (in pixels) for different body parts. $R_{torso} = 32$ px $\times$ 32 px, $R_{head} = R_{u.arm} = R_{l.arm} = 16$ px $\times$ 16 px, $R_{neck} = R_{hand} = 8$ px $\times$ 8 px

## Appendix B: Derivation of Pose Selection Posterior Probability

Let $\Pi$ be the set of "allowable" 3D poses and $S$ the associated 2D pose exemplar library (see Sect. 3.4.1). Let $s_\pi$ be the 2D pose exemplar associated with the pose $\boldsymbol{\pi}$ and $\Pi_s$ the set of poses associated with the 2D pose exemplar $s$, at the leaf level of the tree.

We expand the pose posterior $p(\boldsymbol{\pi}|O)$ over all 2D pose exemplars $\boldsymbol{S}$ and consider correct and incorrect matches ($s^+$ and $s^-$, respectively)

$$\begin{aligned} p(\boldsymbol{\pi}|O) &= \sum_{s\in\boldsymbol{S}} p(\boldsymbol{\pi}, s^+|O) = \sum_{s\in\boldsymbol{S}} p(\boldsymbol{\pi}|s^+)p(s^+|O) \\ &= \sum_{s\in\boldsymbol{S}} p(\boldsymbol{\pi}|s^+)\frac{p(O|s^+)p(s^+)}{p(O)} \\ &= \sum_{s\in\boldsymbol{S}} p(\boldsymbol{\pi}|s^+)\frac{p(O|s^+)p(s^+)}{p(O|s^+)p(s^+)+p(O|s^-)p(s^-)} \end{aligned} \tag{22}$$

where $p(s^+)+p(s^-)=1$.

Each pose is associated with exactly one 2D pose exemplar at the leaf level of the tree, and one 2D pose exemplar $s$ is associated with a number of poses $\Pi_s$. Therefore we model:

$$p(s^+|\boldsymbol{\pi}) = \begin{cases} 1 & \text{if } \boldsymbol{\pi}\in\Pi_s \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

and analogous

$$p(s^-|\boldsymbol{\pi}) = \begin{cases} 1 & \text{if } \boldsymbol{\pi}\in\Pi\setminus\Pi_s \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

It follows that

$$\begin{aligned} p(\boldsymbol{\pi}|s^+) &= \frac{p(s^+|\boldsymbol{\pi})p(\boldsymbol{\pi})}{p(s^+)} = \frac{p(s^+|\boldsymbol{\pi})p(\boldsymbol{\pi})}{\sum_{\boldsymbol{\pi}'\in\Pi} p(s^+|\boldsymbol{\pi}')p(\boldsymbol{\pi}')} \\ &= \begin{cases} \frac{p(\boldsymbol{\pi})}{\sum_{\boldsymbol{\pi}'\in\Pi_s} p(\boldsymbol{\pi}')} & \text{if } \boldsymbol{\pi}\in\Pi_s \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{25}$$

Note that in (22), the sum is only non-zero for $s=s_{\boldsymbol{\pi}}$, the shape exemplar associated with pose $\boldsymbol{\pi}$. By expanding $p(s^+)$ and $p(s^-)$ over all poses $\Pi$, it also follows that

$$p(s^+) = \sum_{\boldsymbol{\pi}'\in\Pi} p(s^+|\boldsymbol{\pi}')p(\boldsymbol{\pi}') = \sum_{\boldsymbol{\pi}'\in\Pi_s} p(\boldsymbol{\pi}') \tag{26}$$

$$p(s^-) = \sum_{\boldsymbol{\pi}'\in\Pi} p(s^-|\boldsymbol{\pi}')p(\boldsymbol{\pi}') = \sum_{\boldsymbol{\pi}'\in\{\Pi\setminus\Pi_s\}} p(\boldsymbol{\pi}') \tag{27}$$

Therefore, by substituting $p(\boldsymbol{\pi}|s^+)$, $p(s^+)$ and $p(s^-)$, the posterior can be written as

$$p(\boldsymbol{\pi}|O) = \frac{p(O|s_\pi^+)p(\boldsymbol{\pi})}{p(O|s_\pi^+)(\sum_{\boldsymbol{\pi}'\in\Pi_s} p(\boldsymbol{\pi}')) + p(O|s^-)(\sum_{\boldsymbol{\pi}'\in\{\Pi\setminus\Pi_s\}} p(\boldsymbol{\pi}'))} \tag{28}$$

Assuming a uniform pose prior, i.e. $p(\boldsymbol{\pi}) \equiv \frac{1}{|\Pi|}$, the term for the posterior simplifies to

$$p(\boldsymbol{\pi}|O) = \frac{p(O|s_\pi^+)}{p(O|s_\pi^+)|\Pi_s| + p(O|s_\pi^-)|\Pi\setminus\Pi_s|} \tag{29}$$

## References


Agarwal, A., & Triggs, B. (2006). Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(1), 44–58.

Andriluka, M., Roth, S., & Schiele, B. (2009). Pictorial structures revisited: people detection and articulated pose estimation. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Balan, A., & Black, M. (2006). An adaptive appearance model approach for model-based articulated object tracking. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Balan, A. O., Sigal, L., Black, M. J., Davis, J. E., & Haussecker, H. W. (2007). Detailed human shape and pose from images. In: *CVPR* (pp. 1–8).

Bergtholdt, M., Kappes, J., Schmidt, S., & Schnörr, C. (2010). A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision*, *87*(1–2), 93–117.

Bissacco, A., Yang, M. H., & Soatto, S. (2007). Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Bo, L., & Sminchisescu, C. (2010). Twin Gaussian processes for structured prediction. *International Journal of Computer Vision*, *87*(1–2), 28–52.

Bouguet, J. Y. (2003). Camera calibration toolbox for Matlab.

Bray, M., Meier, E. K., Schraudolph, N. N., & Gool, L. J. V. (2007). Fast stochastic optimization for articulated structure tracking. *Image and Vision Computing*, *25*(3), 352–364.

Brubaker, M., Fleet, D., & Hertzmann, A. (2010). Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, *87*(1–2), 140–155.

Cheung, K. M., Baker, S., & Kanade, T. (2005a). Shape-from-silhouette across time—part I. *International Journal of Computer Vision*, *62*, 221–247.

Cheung, K. M., Baker, S., & Kanade, T. (2005b). Shape-from-silhouette across time—part II. *International Journal of Computer Vision*, *63*(3), 225–245.

Corazza, S., Mündermann, L., Gambaretto, E., Ferrigno, G., & Andriacchi, T. (2010). 3D human motion tracking with a coordinated mixture of factor analyzers. *International Journal of Computer Vision*, *87*(1–2), 156–169.

Deutscher, J., & Reid, I. (2005). Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, *61*(2), 185–205.

Drummond, T., & Cipolla, R. (2001). Real-time tracking of highly articulated structures in the presence of noisy measurements. In *Proc. of the IEEE international conference on computer vision (ICCV)* (pp. 315–320).

Ferrari, V., Marin-Jimenez, M., & Zisserman, A. (2009). Pose search: retrieving people using their pose. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Fleuret, F., Berclaz, J., Lengagne, R., & Fua, P. (2008). Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(2), 267–282.

Forsyth, D. A., Arikan, O., Ikemoto, L., O'Brien, J., & Ramanan, D. (2005). Computational studies of human motion. *Foundations and Trends in Computer Graphics and Vision*, *1*(2–3), 77–254.

Fossati, A., Dimitrijevic, M., Lepetit, V., & Fua, P. (2007). Bridging the gap between detection and tracking for 3D monocular video-based motion capture. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Fossati, A., Salzmann, M., & Fua, P. (2009). Observable subspaces for 3D human motion recovery. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Gall, J., Stoll, C., de Aguiar, E., Theobalt, C., Rosenhahn, B., & Seidel, H. P. (2009). Motion capture using joint skeleton tracking and surface estimation. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Gall, J., Rosenhahn, B., Brox, T., & Seidel, H. P. (2010). Optimization and filtering for human motion capture. *International Journal of Computer Vision*, *87*(1–2), 75–92.

Gavrila, D. M. (1999). The visual analysis of human movement: a survey. *Computer Vision and Image Understanding*, *73*(1), 82–98.

Gavrila, D. M. (2007). A Bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(8), 1408–1421.

Gavrila, D. M., & Davis, L. (1996). 3-D model-based tracking of humans in action: a multi-view approach. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Gavrila, D. M., & Munder, S. (2007). Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision*, *73*(1), 41–59.

Hasler, N., Rosenhahn, B., Thormählen, T., Wand, M., Gall, J., & Seidel, H. P. (2009). Markerless motion capture with unsynchronized moving cameras. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Hofmann, M., & Gavrila, D. M. (2009a). Multi-view 3D human pose estimation combining single-frame recovery, temporal integration and model adaptation. In: *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Hofmann, M., & Gavrila, D. M. (2009b). Single-frame 3D human pose recovery from multiple views. In *Proc. of the DAGM symposium on pattern recognition*.

Kakadiaris, I., & Metaxas, D. (2000). Model-based estimation of 3-D human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(12), 1453–1459.

Kanaujia, A., Sminchisescu, C., & Metaxas, D. (2007). Semi-supervised hierarchical models for 3D human pose reconstruction. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Kehl, R., & Gool, L. V. (2006). Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding*, *103*(2–3), 190–209.

Knossow, D., Ronfard, R., & Horaud, R. (2008). Human motion tracking with a kinematic parametrization of extremal contours. *International Journal of Computer Vision*, *79*, 247–269.

Kohli, P., Rihan, J., Bray, M., & Torr, P. H. S. (2008). Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *International Journal of Computer Vision*, *79*, 285–298.

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

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

Lee, M. W., & Cohen, I. (2006). A model-based approach for estimating human 3D poses in static images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(6), 905–916.

Lee, M. W., & Nevatia, R. (2009). Human pose tracking in monocular sequence using multilevel structured models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(1), 27–38.

Li, R., Tian, T. P., Sclaroff, S., & Yang, M. H. (2010). 3d human motion tracking with a coordinated mixture of factor analyzers. *International Journal of Computer Vision*, *87*(1–2), 170–190.

Liem, M., & Gavrila, D. M. (2009). Multi-person tracking with overlapping cameras in complex, dynamic environments. In *Proc. of the British machine vision conference (BMVC)*.

Lv, F., & Nevatia, R. (2007). Single view human action recognition using key pose matching and Viterbi path searching. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, *11*, 431–441.

Mikic, I., Trivedi, M., Hunter, E., & Cosman, P. (2003). Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, *53*(3), 199–223.

Moeslund, T. B., Hilton, A., & Kruger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, *103*(2–3), 90–126.

Mori, G., & Malik, J. (2006). Recovering 3D human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(7), 1052–1062.

Navaratnam, R., Thayananthan, A., Torr, P. H. S., & Cipolla, R. (2005). Hierarchical part-based human body pose estimation. In *Proc. of the British machine vision conference (BMVC)*.

Ong, E. J., Hilton, A., & Micilotta, A. S. (2006). Viewpoint invariant exemplar-based 3D human tracking. *Computer Vision and Image Understanding*, *104*, 178–189.

Peursum, P., Venkatesh, S., & West, G. (2007). Tracking-as-recognition for articulated full-body human motion analysis. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

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

Pilu, M., & Fisher, R. B. (1995). Equal-distance sampling of superellipse models. In *Proc. of the British machine vision conference (BMVC)*.

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

Ramanan, D., Forsyth, D. A., & Zisserman, A. (2007). Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(1), 65–81.

Roberts, T. J., McKenna, S. J., & Ricketts, I. W. (2006). Human tracking using 3D surface colour distributions. *Image and Vision Computing*, *24*(12), 1332–1342.

Rogez, G., Rihan, J., Ramalingam, S., Orrite, C., & Torr, P. H. (2008). Randomized trees for human pose detection. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Rosenhahn, B., & Brox, T. (2007). Scaled motion dynamics for markerless motion capture. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Seshadri, N., & Sundberg, C. (1994). List Viterbi decoding algorithms with applications. *IEEE Transactions on Communications*, *42*, 313–323.

Shakhnarovich, G., Viola, P., & Darrell, T. (2003). Fast pose estimation with parameter-sensitive hashing. In *Proc. of the IEEE international conference on computer vision (ICCV)* (pp. 750–757).

Sigal, L., & Black, M. (2010). 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.

Sigal, L., Bhatia, S., Roth, S., Black, M. J., & Isard, M. (2004). Tracking loose-limbed people. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Sigal, L., Balan, A., & Black, M. (2010). Humaneva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, *87*(1–2), 4–27.

Starck, J., & Hilton, A. (2003). Model-based multiple view reconstruction of people. In *Proc. of the IEEE international conference on computer vision (ICCV)* (pp. 915–922).

Stenger, B., Thayananthan, A., Torr, P. H. S., & Cipolla, R. (2006). Model-based hand tracking using a hierarchical Bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(9), 1372–1384.

Sundaresan, A., & Chellappa, R. (2009). Multicamera tracking of articulated human motion using shape and motion cues. *IEEE Transactions on Image Processing*, *18*(9), 2114–2126.

Vondrak, M., Sigal, L., & Jenkins, O. C. (2008). Physical simulation for probabilistic motion tracking. In *Proc. of the IEEE conf. on computer vision and pattern recognition (CVPR)*.

Xu, X., & Li, B. (2007). Learning motion correlation for tracking articulated human body with a Rao-Blackwellised particle filter. In *Proc. of the IEEE international conference on computer vision (ICCV)*.

Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. In *Proc. of the international conference on pattern recognition (2)* (pp. 28–31).