This is an author produced version of a paper published in:
*Image and Vision Computing*

Cronfa URL for this paper:
http://cronfa.swan.ac.uk/Record/cronfa49667

**Paper:**
http://dx.doi.org/10.1016/j.imavis.2016.05.010

This item is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence. Copies of full text items may be used or reproduced in any format or medium, without prior permission for personal research or study, educational or non-commercial purposes only. The copyright for any work remains with the original author unless otherwise specified. The full-text must not be sold in any format or medium without the formal permission of the copyright holder.

Permission for multiple reproductions should be obtained from the original author.

Authors are personally responsible for adhering to copyright and publisher restrictions when uploading content to the repository.

http://www.swansea.ac.uk/library/researchsupport/ris-support/
Fixing the Root Node: Efficient Tracking and Detection of 3D Human Pose through Local Solutions

Ben Daubney, Member, IEEE, Xianghua Xie, Member, IEEE, Neil Mac Parthaláin, Member, IEEE, and Reyer Zwiggelaar, Member, IEEE

Abstract—3D human pose estimation is a very difficult task. In this paper we propose that this problem can be more easily solved by first finding the solutions to a set of easier sub-problems. These are to locally estimate pose conditioned on a fixed root node state, which defines the global position and orientation of the person. The global solution can then be found using information extracted during this procedure. This approach has two key benefits: The first is that each local solution can be found by modeling the articulated object as a kinematic chain, which has far less degrees of freedom than alternative models. The second is that by using this approach we can represent, or support, a much larger area of the posterior than is currently possible. This allows far more robust algorithms to be implemented since there is less pressure to prune the search space to free up computational resources. We apply this approach to two problems: The first is single frame monocular 3D pose estimation, where we propose a method to directly extract 3D pose without extracting any intermediate 2D representation or being dependent on strong spatial prior models. The second is multi-view 3D tracking where we show that using the above technique results in an approach that is far more robust than current approaches, without relying on strong temporal prior models. In both domains we demonstrate the strength and versatility of the proposed method.

Index Terms—3D pose estimation, tracking, local solutions, root node.

1 INTRODUCTION

There is currently much interest in tracking and estimating the pose of articulated objects, particularly if this articulated object is a human. The principal difficulties with this task is the high dimensionality of the solution space and noisy, often ambiguous observations. The high dimensionality of the solution space can be overcome by, for example, using iterative approaches that allow a coarse to fine resolution search to be performed. Whilst effective at finding strong minima, these approaches are susceptible to ambiguous observations; only a small area of the posterior is supported making these methods prone to unrecoverable failure. This is in contrast to methods such as the Pictorial Structures Model (PSM), where the entire search space is evaluated over a discrete grid. By calculating the full posterior distribution ambiguities can be overcome as either more observations become available or by integrating further high-level a priori information. Whilst the PSM is applicable to 2D pose estimation, when applied to 3D pose estimation the search space becomes too large to exhaustively search without resorting to an unsatisfactorily coarse grid.

The approach we propose in this work is to combine the advantages of both methods. On the one hand we want to provide much wider support across the solution space, whilst on the other hand keep computational costs low by actively pruning areas of little interest to enable finer resolution searches to be performed over a continuous, rather than discrete, state space. To accommodate these seemingly conflicting ideas, instead of estimating pose as a single optimization problem, we attempt to solve a set of easier, more constrained intermediate problems. By allowing each intermediate problem to be solved independently, we can ensure that these are broadly distributed across the domain, thus increasing support. Whilst at a local level to each sub-problem, we can more confidently reduce the search space to provide a set of local solutions.

Each local solution is provided by estimating the conditional posterior distribution. This is a distribution over human pose conditioned on a fixed state of the root node. The root node represents the position and orientation of the person in the global frame of reference. The principal assumption we exploit in this work is that it is much easier to correctly estimate 3D pose if the correct state of the root node is known a priori. To exploit this assumption we only require that: 1. The correct state of the root node is contained within the solution set; 2. Given (1), that the correct root node state can then be identified. The first condition is met by ensuring local solutions are found for a broad range of root node states. The
second condition is achieved by using knowledge gained through estimating each local solution, for example, by finding the local solution with the highest likelihood.

This work is motivated by the intuition that articulated models are made from parts of fixed lengths with fixed joint positions and as such the limbs operate as kinematic chains. Consider some observed image feature (e.g. a binary silhouette) and a 3D object configured to the optimal pose to explain this. If we were to apply a small perturbation to the state of the root node the entire configuration must then alter to achieve the maximum cost or ‘fit’ given the new state of the root node. It is this dependence that motivates this work.

Since each local solution is conditioned on a fixed root node, we can model the articulated object as a kinematic chain, which is far more constrained and therefore efficient to optimize over compared to alternative models, such as the loose-limbed model [1], [2]. For example, a loose-limbed model requires twice as many degrees of freedom for each part than a kinematic chain. This is as for each part both the orientation and position must be known, whereas for a kinematic chain only the orientation is required, since its position is implicitly defined by the parts to which it is connected.

However, the technical challenge in using this methodology is not to allow the computational cost of finding multiple solutions to be greater than competing methods that estimate pose as a single optimization problem. Through this work we show this can be achieved and present a number of novel approaches to attain this whilst applying a local-solution approach to two different problems. The first is direct 3D pose estimation from single monocular images and the second is 3D multiple-view tracking of pose. It is shown that in both scenarios state-of-the-art results are achieved without providing our method with additional computational resources, where we assume the computational bottleneck is in the required number of image likelihood evaluations.

The remainder of the paper is structured as follows: In the next section we provide a review of current approaches to pose estimation of humans from images. Then in Section 3 we describe the framework used in the remainder of this paper. Following this we then apply this framework to two different applications. The first is single image monocular 3D pose estimation and the second is multiple-view tracking. For each application, whilst the framework used remains constant, the implementations are very different. For example we use different graphical representations, different likelihood functions (both generative and discriminative) and different methods of searching the state space to find each local solution. We hope these differences in each implementation highlight the strength and versatility of the framework. Finally, in Section 6 we provide a discussion and draw conclusions from our work.

2 BACKGROUND

Human pose estimation can be broadly split into two categories, detection and tracking. The key difference between the two is in the source of prior information used by each. In tracking this information is provided through observations made in the previous time steps and a temporal prior that describes how a part is expected to move. For pose detection it is provided by a spatial prior that describes the relationships between connected parts. The prior effectively adds a set of constraints. In general the more constrained the prior the better the method will work given noisy data, though this is at the expense of its generality.

Simple and unconstrained priors include zero-mean Gaussian diffusion for tracking [3] and a uniform prior over all possible configurations for pose detection [4]. However, the focus of much recent work has been on developing stronger priors. For tracking, action specific models are learned using methods such as Gaussian Process Dynamical Models (GPDM) [5] or Mixture of Factor Analyzers [6]. These methods effectively reduce the dimensionality of the pose space by exploiting repeated patterns of motion in actions such as walking or running. A benefit is that they can learn correlations between unconnected parts of the body allowing a part to be localized even if it is occluded, though this is at the expense that this approach will deteriorate for unseen motions or poses. This limitation can be overcome by learning a range of priors of different motions and extracting the required prior at runtime [7]. In this work for our tracking approach, as we use a multiple camera setting we use a zero mean diffusion model as a temporal prior. We opt for this weak temporal prior since the resulting method is then most dependent on the observations available, which are often noisy and incomplete. This exposes the performance of the underlying tracking methods in coping with this noise, not the strength of the prior model or data it has been learned from.

In pose detection it is often desirable to keep the model as general as possible so it is applicable to a variety of poses. For example a single gaussian may represent the prior between connected parts [1]. Correlations between unconnected parts are modeled by adding latent variables [8] or learning a set of more constrained individual priors by first clustering training data and then learning a model from each cluster [9], [10]. However, we note that the clusters in these approaches, which are 2D approaches, often appear to represent 2D projections of a 3D articulated object viewed at different orientations. We suggest in this work a more direct method is simply to learn a 3D prior in the first instance. This can then be projected into the image at arbitrary positions and orientations.
allowing greater flexibility without having to learn multiple priors.

Whilst still an open problem, some have attempted to combine tracking and detection using both strong temporal and spatial priors [2], [4], [11]–[13]. However, directly combining the two often results in an untractable optimization problem where the global solution can not be guaranteed [2]. A more popular method is to effectively treat the two as independent problems [4], [11], [13]. Temporal priors are used to reduce occurrences of false positives, whilst improving true positive rates. A further benefit is that temporal consistency of the appearance of parts across a sequence can also be exploited.

In addition to a prior, another key component needed for estimating human pose is a method of optimization or inference. Currently, for single image 2D pose estimation a popular method is the Pictorial Structures Model (PSM) [1], [14]. This is a part based approach, where each part is detected independently and then these detections are assembled into the most likely configuration using a spatial prior and Dynamic Programming. This approach assumes a tree structure, where nodes represent the parts of the model and physically connected parts are joined by edges. The search space for each part is defined by a uniformly sampled grid that covers all permissible orientations and positions. The benefit of a uniformly sample grid is that the maximum coverage of the search space is achieved given the available resources, there is no bias as a result of initialization. Additional edges can be added to the model to represent temporal connections, however, often the problem then becomes intractable and methods such as Loopy Belief Propagation [15] or using a combination of trees [12] can be used to find a local solution.

Whilst popular for 2D pose estimation it is not obvious how to apply these uniformly sampled grid approaches to 3D pose estimation. The main difficulty is how to discretize the search space of a more complex and higher dimensional object and negate the additional computation cost of exploring this space. For this reason stochastic approaches are popular for 3D pose estimation [2], [3], [16]–[19]. Each stochastic sample may represent the entire state of the body [3], [16], [17], [19] or an individual part [2], [18], [20]. Intuitively, estimating the entire state is more computationally intensive since the size of the search space is exponential with the number of parts, though methods have been developed to improve the efficiency of this task. For example, Annealing can be used to gently guide the samples to ensure areas of the posterior with a higher likelihood are searched more carefully on subsequent iterations [3] or by clustering the samples and searching along their axes of uncertainty [21]. An alternative is to optimize individual components of the object’s state, for example using Partitioned Sampling [17] or Markov Chain Monte Carlo [16]. A limitation with these approaches is that they are iterative and need convergence for a solution to be found. As noted in [22], [23], this convergence happens in a particular order for objects modeled as a kinematic chain. Typically, those parts nearer a fixed node must converge before parts further down the model can do so. Intuitively, this is as any uncertainty in a given part will be propagated down the kinematic chain.

To overcome this problem, stochastic part based methods can be used, such as Non-Parametric Belief Propagation [2], [24] or Variational MAP [18]. These approaches do not model an articulated model as a kinematic chain but as a loose-limbed model, where the joint between connected parts is soft and allowed to deform. However, as the connection between parts is soft, the model is less constrained and slippage can occur, where two limbs can be joined at a very unlikely location or may not even be physically joined. It has been shown that given a known root location, models that have fixed joint positions outperform loose-limbed models at estimating 3D pose [23].

A popular alternative to direct 3D pose estimation is to first estimate 2D pose and then ‘lift’ this to 3D using a low dimensional embedding of the action you are observing [5], [13]. However, the limitation of this is that whilst the 2D prior is likely to be very general the mapping between 2D and 3D will most likely not be.

The final component needed to estimate pose is an image likelihood distribution. This provides a data cost of placing the articulated object in a particular configuration given an observed image. Perhaps the most popular method is the binary silhouette, where the object of interest is pre-segmented from the remainder of the image (e.g. [1], [25]). The cost is then typically maximized by covering the largest region of the foreground object. However, in many scenes a good segmentation is difficult to achieve and is limited to only a stationary camera. Other features typically used include the Chamfer match (e.g. [11]) or discriminatively trained part detectors using Histogram of Orientated Gradients (HOG) features [10] or Shape Contexts [26]. Some approaches use a combination of features to achieve the best performance [12] or adaptively learn the appearance model online using clustering [4]. In this work we explore using both generative and discriminative detectors.

The method we propose in this work allows the benefits of a part based approach to be exploited, whilst still modeling the body as a kinematic chain. This results in a method that is constrained, allowing accurate pose estimation to be performed, yet efficient. This is achieved by fixing the root node state for each local solution and finding the conditional posterior for each. All probability density functions are represented by parametric models making the representation extremely efficient compared to purely
particle based approaches.

The approach developed in this work is applied to two problems, tracking and monocular pose estimation. The tracking method has been previously been published in a conference proceedings [27], however, in this work we significantly strengthen the principal and theoretical grounding for the approach. This permits us to develop a much more general framework and we demonstrate this by also applying it to the problem of unconstrained monocular pose estimation. The tracking method described can be seen as a single implementation, or incarnation, of the framework described herein.

3 Approach

Pose estimation can be described as an inference problem; infer the most likely pose $X$ given some observations $Z$. By Bayes’ rule, the posterior distribution of a pose $X$ given an observation $Z$ is

$$p(X|Z) \propto p(Z|X)p(X),$$

where $p(Z|X)$ is the data cost or observational likelihood of making the observation $Z$ given the pose $X$ and $p(X)$ is the prior distribution over $X$, which describes those poses that are more likely than others. In the case of tracking, the prior will be derived from observations made in a previous time step and can be written as $p(X|\{Z_{t-1},..,Z_0\})$. In single image pose estimation the prior will be learned from training data, be independent of time and take the form $p(X|\Phi)$, where $\Phi$ represents the model parameters of the prior.

A popular method to solve Equation (1) is to represent the human body as a graph consisting of $n$ hidden and $n$ observable nodes. The hidden nodes represent the state of individual parts, $X = \{x_1,..,x_n\}$, and the observable nodes represent the observation for each part, $Z = \{z_1,..,z_n\}$. Each hidden node is connected to a single observable node and each pair of physically connected parts are joined by an edge resulting in a graph that is a tree. In this work, our approach is to solve Equation (1) in two steps. The first step is to estimate a set of local solutions. Each local solution is given by first calculating the conditional probability distribution over pose given a fixed value for the root node state $x_r$:

$$p(X|Z, x_r) \propto p(Z|X)p(X|x_r),$$

where $X = \{x_1,..,x_{n-1}\}$ represents the state of all nodes other than the root node and $Z = \{z_1,..,z_{n-1}\}$ represents all observations excluding that for the root node. The states can thus be decomposed as $X = \{X, x_r\}$ and $Z = \{Z, z_r\}$.

A local solution is then given by the pose that maximizes the conditional distribution

$$X^*(x_r) = \arg\max_X p(X|Z, x_r).$$

Whilst this estimate only represents a local maximum, since the graph used to describe the body is a tree, the problem is convex and, for example, the Maximum a Posteriori (MAP) solution could then be found by finding the local solution that maximizes Equation (1). Another approach we explore in this work is instead to train discriminative detectors to locate the correct local solution.

3.1 Probability Density Function Representation

Current techniques to estimate human pose represent Probability Density Functions (PDF) using a set of discrete sample points. The distribution or density of the samples may represent the PDF if using a method such as the Particle Filter [19] or, alternatively the samples may be taken uniformly over a grid and weighted by the PDF at that position, this is particularly popular if using a method such as Dynamic Programming or Belief Propagation [1]. We refer to these samples as delta-samples since they only represent the PDF at a single position. In this work we also represent the PDF using a set of samples, though contrary to the previous work each sample approximates an entire hyperplane of the PDF using a parametric model, the result is that fewer samples can be used to represent a much larger proportion of the PDF providing more support. We refer to these as hyper-samples, which we now describe.

The PDF over pose $X$ is approximated by a set of $M$ hyper-samples

$$p(X) \approx \left[\sum_{m=1}^{M} S^m\right],$$

where each hyper-sample represents the PDF over all parts conditioned on a given root node state:

$$S^m = p(X|x^m_r).$$

Each hyper-sample can further be decomposed to a distribution over each part, excluding the root node,

$$p(X|x^m_r) = \{p(x_1|x^m_r),..,p(x_{n-1}|x^m_r)\}.$$  

Each of these distributions is represented by a parametric model $p(x_i|x^m_r) = F(x_i, \Theta^m_i)$, where $\Theta^m_i$ is the set of model parameters. An additional term $p(x^m_r)$, which is the marginal over all parts excluding the root node, is also appended to the hyper-sample. This is a scalar and can be viewed as the hyper-sample’s weight. A hyper-sample is therefore defined parametrically as $S^m = \{x^m_r, \Theta^m_1,..,\Theta^m_{n-1}, w^m\}$, where $w^m$ is the weight and $x^m_r$ is the root node value on which the distribution is conditioned. Whilst there are many choices of parametric model, we use a single Gaussian or a Gaussian Mixture Model (GMM).

Each hyper-sample $S^m$ represents a hyper-plane of the PDF over all parts, except the root node, as a parametric model. Each hyper-plane is parallel to the axes of the root node state and passes through the point $x^m_r$. This method of approximating a PDF is
illustrated in Figure 1 and compared to using a set of delta-samples. As can be seen a small set of hyper-samples can represent a large area of the PDF and are much more informative. For example each mode is easily accessible through the parameters of $S^m$, whereas further analysis, such as clustering, would need to applied to extract the modes of the representation depicted in Figure 1 (a). The distribution over the root node is provided by the root node state and weight of each Hyper-Sample, $p(x_r) \approx [x_r^m, w^m]_{m=1}^M$.

### 3.2 Estimating a Local Solution

In this section, we describe how given a hyper-sample representing a conditional prior distribution, $S^m \approx P(X|x_r^m)$, we can estimate the conditional posterior $P(X|Z, x_r^m)$. Firstly, we present a method when the graph can be simplified beyond a tree to a star. In this graph all nodes are directly connected to the root node and can therefore be solved independently given that the root node is fixed. We employ this graph to represent the problem of single frame pose estimation.

Following this we explore how local solutions for general trees can be found. We employ a more complex tree as the graphical model used in our tracking approach.

Given a hyper-sample representing the conditional prior $S^m = \{x_r^m, \Theta_r^m, \ldots, \Theta_r^{m-1}, w^m\}$, where the prior over each part (excluding the root node) is provided by $p(x_i|x_r^m) = F(x_i, \Theta_r^m)$. We estimate the conditional posterior by decomposing Equation (2) so that the conditional distribution for each node can be calculated independently:

$$p(x_i|z_i, x_r^m) \propto p(z_i|x_i)p(x_i|x_r^m).$$

The distribution $p(x_i|z_i, x_r^m)$ is calculated using Importance Sampling. Initially $d$ delta-samples are drawn from the prior distribution,

$$p(x_i|x_r^m) \approx [x_i^s, \pi_i^s]_{s \in D_i}^d,$$

where each delta-sample has uniform weight $\pi_i^s$. The delta-sample is then used to articulate the model, which is projected into the image to sample the observational likelihood function. The value of the likelihood function is used to weight the delta-sample ($\pi_i^s \propto p(z_i|x_i^s)$). The samples and their weights can then be used to update the model parameters $\Theta_r^m$. This can be achieved, for example, using the Maximum Likelihood Estimate (MLE). This is repeated for each node so that the conditional posterior is represented as $S^{m+s} = \{x_r^m, \Theta_r^m, \ldots, \Theta_r^{m-1}, w^{m+s}\}$, where $w^{m+s} = \prod_{i=1}^n \sum_{s \in D_i} \pi_i^s$ and now $p(x_i|z_i, x_r^m) = F(x_i, \Theta_r^m)$.

The method used to extract the delta-samples from each distribution is dependent on the parametric model being used to represent the PDF over each node. In this work we use both an existing stochastic approach (for single image pose estimation) and a novel deterministic approach (for multi-view tracking) to select the delta-samples. However, note that the distribution for each part is always represented using a parametric model, sampling of these distributions is only used to take a measurement of the observational likelihood, not to act as its principal representation as in the other particle based approaches (e.g. [3]).

If the graph used to represent a person is a tree the method to calculate the conditional posterior is more complex. Whilst a fixed root node allows the PDF for each branch or sub-tree to be calculated independently. Each branch may consist of more than a single node. As such Equation (2) must be decomposed so that the marginal or belief at each node is now calculated as:

$$p(x_i|Z, x_r^m) \propto p(z_i|x_i) \prod_{v_j \in E(i)} p(x_j|z_j, \ldots, z_T),$$

where $v_j \in E(i)$ defines the set of edges connected to $i$ and $z_j, \ldots, z_T$ represents the set of observations for the subtree containing $v_j$, created by removing the edge $\{v_i, v_j\}$. The terms $p(x_j|z_j, \ldots, z_T)$ in Equation (9) represent messages passed from connected nodes in the graph. The exact details of how these messages are calculated is described in Section 5 since it is dependent on the parametization of the PDF used.

In the following sections, we describe two applications using local solutions implemented through hyper-samples. The first is single frame monocular pose estimation and the second is tracking a person using a multiple camera environment. For tracking, the hyper-samples are distributed over $x_r$ stochastically, similar to existing particle filtering approaches, though in contrast the delta-samples are selected deterministically making the approach extremely efficient. For single frame pose estimation the hyper-samples are distributed uniformly over $x_r$ as in the PSM model to ensure maximum coverage of the pose state space, though in contrast, now the delta-samples are drawn stochastically. For monocular pose estimation we use a discriminative likelihood function, $p(z_i|x_i)$, whilst for tracking it is generative.
Furthermore, to estimate the correct root node state, $x^*_r$, a discriminatively trained detector is used for monocular pose estimation, whereas for tracking we use a generative model. The PDF over each individual part, $p(x_i | x^m)$, is modeled using a single Gaussian for tracking and a Gaussian Mixture Model for single frame monocular pose estimation. This is as we expect much more uncertainty in the pose when we use single images, compared to multiple view tracking, which also integrates observations from previous time steps.

The graphical model used to represent the human body for each task is shown in Table 1 (top row). The node labels for Monocular Pose estimation are Torso, Head (H), Left Arm (LA), Left Leg (LL) etc. The labels for Multi-View Tracking correspond to the Hip (Hip), Torso (Tor), Upper Left Arm (ULA), Lower Left Arm (LLA), Upper Left Leg (ULL) etc. As can be seen for Monocular Pose estimation a single node is used to represent an entire limb, whereas for tracking a node defines a single part. Whilst the implementation details of the two approaches are very different the fundamental approach is the same and this highlights the benefit of using a fixed root node approach when tracking or detecting the pose of articulated objects. A comparison of the two approaches can be seen in Table 1 and a full description of the model used for each is provided in the following sections.

4 Application 1: Monocular 3D Pose Detection

In this section, we apply our method to the problem of estimating 3D pose from single monocular images. Human pose estimation benefits from a fixed root node approach in a number of ways. Firstly, using our method we ensure maximum coverage of the search space is achieved given a fixed set of resources, by distributing the hyper-samples uniformly across the state space of the root node. Secondly, we use limb likelihood estimates for a given local solution to train discriminative human detectors to improve detection rates. Finally, using our method, an accurate solution can be located without the requirement of convergence as the hyper-samples can easily represent multiple modes, even when conditioned on a single root node value. This is as the prior is modeled using a Gaussian Mixture Model (GMM) and we show that each component, learned in quaternion space, represents an independent volume when projected into Euclidian space.

Further contributions in this section are that we address how to accommodate deterministic part detectors applied at arbitrary orientations and scales in the image. This is in contrast to typical 2D approaches that simply pre-rotate and scale the image before part detection commences to detect parts over a very small range of scales and orientations [10]. This allows us to search for pose over a continuous state space, which allows more accurate poses to be extracted.

4.1 Model Representation and Sampling from the Prior

The graphical model used in this section is a star (as shown in Table 1), consisting of 6 nodes; the root node, which represents the torso and 5 nodes representing each of the main limbs. We assume the position of the ground plane is known, which is a common assumption for 3D pose estimation and tracking [11], [28], though methods do exist that could be used to automate this process (e.g. [29]). We also assume that the subject remains upright so their orientation is defined by the direction that they are facing across the ground plane. Hence, the state of the root node is parametrized as $x_r = (d_r, q_r)$, where $d_r \in \mathbb{R}^2$ defines the position on the ground plane and $\{q_r \in \mathbb{R}, 0 \leq q_r < 2\pi\}$ defines the orientation of the subject. Searches over more complex orientations could be accommodated by expanding the degrees of freedom of the root node state.

The prior $p(X|\Psi)$ is represented using a set of GMM’s. One is learned for each limb, hence $p(x_i | \phi_i) = \sum_{k=1}^{K} \lambda_k N(x_i; \mu_k^i, \Sigma_k^i)$, so that $\phi_i = \{\lambda_k, \mu_k^i, \Sigma_k^i\}_{k=1}^{K}$, where $K$ is the number of components in the model, and $\lambda_k, \mu_k^i$ and $\Sigma_k^i$ represent the $k$th component’s weight, mean and covariance respectively. Each Hyper-sample is initialized with this same prior distribution. However, the root node state is different for each. The root node state is taken from a three dimensional grid over $x_r$, representing locations on the ground plane and at each position a set of discrete orientations. This is depicted in Figure 2 (c) where we visualize a subset of the initial hyper-samples used to estimate pose. In practice this is sampled much more densely and at each sample point there are also hyper-samples with different orientations. Each hyper-sample is therefore parameterized as $S^m = \{x_r^m, \Theta^m, w^m\}$, where $\Theta^m = \{\lambda_k^i, \mu_k^i, \Sigma_k^i\}_{k=1}^{K}$.

Each distribution is learned over possible limb rotations, which are represented as unit quaternions. To approximate a Gaussian distribution over quaternions we use an approach similar to [2]. Each unit quaternion is represented by two parts a scalar and vector part $q = q_0 + \mathbf{q}$. By ensuring the scalar component is positive a quaternion can be represented in $\mathbb{R}^3$ using only the vector part. To reduce the likelihood of training data being located across the edge of the unit sphere, the training data is used to estimate a ‘safe’ quaternion space by rotating the data so that the sum of the scalar component across all data is maximal [27]. A GMM can then be learned directly in this space for each limb independently.

Each model is learned over an entire limb, which is represented by a single node. This may represent a distribution over more that a single part, i.e. a
distribution over the left leg models that over both
the lower and upper leg, hence \( x_i \in \mathbb{R}^6 \) (since each
part has three degrees of freedom), except for the head
which is modeled as a single part, \( x_{\text{head}} \in \mathbb{R}^3 \). The
covariance for each part is diagonal and can be written
as \( \Sigma_i = \text{diag}(x_{1i}^{k}, ..., x_{pi}^{k}) \), where \( p \) is the number of
parts for a given limb. For the arms and legs \( p = 2 \),
and for the head \( p = 1 \). The rotations are defined in
the frame of reference of the root node, not the part
to which they are physically connected.

Delta-samples are drawn from each GMM by first
picking a component with likelihood \( k_i \propto \Sigma_i \), following
which a sample is drawn from the selected component
\( (x_{1i}^{k}, ..., x_{pi}^{k}) \sim N((x_{1i}^{k}, ..., x_{pi}^{k})^T : \mu_{ij}^{k} : \Sigma_{ij}^{k}) \). The
rotations described by the sample can then be applied
to each limb and the kinematic chain assembled. The
root node state, \( x_{r}^{m} \), gives the pose its global position
and orientation.

Though a single delta-sample may represent the
state of more than one part, an observational likelihood
model is learned for each part independently. Therefore,
the observational likelihood for a given delta-sample is given by combining the likelihood of each part assuming conditional independence, hence
\( p(z_i | x_{r}^{m}) = \prod_{j=1}^{p_i} p(z_{ij} | x_{ij}^{m}) \). Thus, the weight of each
delta-sample is \( \pi_i = \prod_{j=1}^{p_i} \pi_{ij}^{m} \). These weights are then
used to estimate \( p(x_i | z_i, x_{r}^{m}) \) by updating the covari-
ance and mean of the component they were sampled
from using the Maximum Likelihood Estimate. Note
that if all weights were uniform the covariance and
mean would be unchanged. The MAP estimate for a
local solution is approximated by finding the GMM
component with the highest likelihood for each node
and using the mean.

An example of the prior used in this work is shown
in Figure 2. In (a) we show delta-samples drawn from
the prior for a fixed root node position and in (b), for
visualization a Gaussian has been fitted in euclidian
space to the delta-samples drawn from each component
in quaternion space. As can be seen a different
component learned in quaternion space appears to
 correspond to an independent area in Euclidian space.

The benefit of this approach is that unlike many other
particle based methods (e.g. [2], [18]), it does not need
to fully converge to provide a meaningful solution.
Effectively, each individual component of the GMM
is responsible for representing a small independent
volume of the posterior distribution. This is only possible since the GMM for each part is learned in
the frame of reference of the root node, not the limb
to which it is attached.

### 4.2 Part Detection and Feature Extraction

To detect individual parts we learn discriminative part
detectors. The features used are based on the His-
togram of Orientated Gradient (HOG) [30], where the
gradient magnitudes in a small rectangular region of
the image are binned depending on their orientation
to form a histogram. A set of these features, which
define the detection window, are then concatenated
together to form a vector that can be used for training.
Each component of this vector is referred to as an
attribute.

In previous approaches applied to 2D pose esti-
mation, a small number of orientations and scales
are searched over for each part (e.g. [31]) allowing
the image to be pre-rotated and scaled before feature

<table>
<thead>
<tr>
<th>Graphical Representation</th>
<th>Monocular Pose Estimation</th>
<th>Multi-View Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Graphical Representation} )</td>
<td>( \text{Stochastic} )</td>
<td>( \text{Deterministic} )</td>
</tr>
<tr>
<td>Estimate ( p(X</td>
<td>Z, X_{r}^{m}) )</td>
<td>( \text{Deterministic} )</td>
</tr>
<tr>
<td>Hyper-sample distribution over ( x_r )</td>
<td>( \text{Discriminative} )</td>
<td>( \text{Generative} )</td>
</tr>
<tr>
<td>Estimate ( x_r )</td>
<td>( \text{Discriminative} )</td>
<td>( \text{Generative} )</td>
</tr>
<tr>
<td>Estimate ( p(z_i</td>
<td>x_r) )</td>
<td>( \text{Gaussian Mixture Model (GMM)} )</td>
</tr>
<tr>
<td>Parametrization of ( p(x_i</td>
<td>z_i, x_{r}^{m}) )</td>
<td>( \text{Gaussian Mixture Model (GMM)} )</td>
</tr>
</tbody>
</table>

Fig. 2. Example of samples drawn from the model prior
(a). In (b) the GMM components have been visualized
by fitting a covariance to the samples drawn from each.
In (c) the model is shown projected to different root
positions, though in practice this would be much more
dense and at each position multiple orientations would
be projected.
4.3 Orientation and Position Estimation

In this section, we describe the features used to train discriminative detectors. These are then used to estimate the global position and orientation of a person, \( x^*_i \). The method is similar to that presented in [13], in that we use a measure similar to limb marginal likelihoods to train a Support Vector Machine (SVM). For each hyper-sample, \( S^m \), a feature vector can be constructed using the weights for each delta-sample extracted to update it, \( Y^m = (b^m_1, ..., b^m_k) \), where \( k \) is the number of parts and

\[
  b^m_{ij} = \frac{1}{d} \sum_{s \in T_i} \pi^s_{ij} \quad (10)
\]

where \( d \) is the number of delta-samples and \( \pi^s_{ij} \) is the sample’s weight for the \( j \)th part of the \( i \)th node (i.e. limb). In total there are ten parts, two for each of the main limbs one for the head and one for the torso, hence \( Y^m \in \mathbb{R}^{10} \).

As previously described the set of hyper-samples, \( S^m \in \mathcal{S} \), are initially the same except that their root node states, \( x^m_r = (d^m_r, q^m_r) \), are distributed uniformly across a grid that describes different locations on the ground plane and a set of different orientations. At each ground plane location, \( d_{gp} \), there is therefore a set of \( n \) hyper-samples, \( V = \{ S^m | d^m_r = d_{gp} \} \), with the same ground plane position, where \( n \) is the number of discrete orientations. The feature we use to both detect a person’s position and orientation is constructed by concatenating the likelihoods of all these hyper-samples together. Hence, \( V(V) = (Y^1, ..., Y^n)^T \). Since we define 16 different orientations, \( V(V) \in \mathbb{R}^{160} \). These are used as features for training detectors. All detectors are trained using the LIBSVM toolbox [35].

4.4 Experiments

We use two datasets to train and test our human pose estimation method. The HumanEva dataset [25] provides ground truth motion capture data so that the accuracy of the pose estimation can be quantified and we also use the TUD Multiview Pedestrians Dataset [13] to test our approach on more unconstrained and cluttered scenes.

When performing monocular 3D pose estimation there will always be a reconstruction ambiguity in absolute scale unless further information is provided, for this reason we provide an estimate of the subject’s height (in mm). Note that their scale in the image plane will still change as they are closer or further from the camera. We also label the ground plane. The hyper-samples are distributed uniformly over the ground plane with a spatial resolution of 100mm and angular resolution of 1/8\( \pi \). We see this as being the natural equivalent to 2D approaches that uniformly distribute the samples across the image plane. To estimate the local solution of each hyper-sample 600
TABLE 2
Description of detectors used. The first column shows the size of grid of HOG features used \((w \times l)\). The second shows the total number of attributes this grid produces \((i.e \ w \times l \times \text{No. Bins})\). The third is the number of attributes selected for the final detector. The fourth shows the total number of rules learned.

<table>
<thead>
<tr>
<th>Part</th>
<th>No. HOG</th>
<th>No. att.</th>
<th>Reduced no. att.</th>
<th>No. rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head</td>
<td>9 \times 8</td>
<td>648</td>
<td>33</td>
<td>14</td>
</tr>
<tr>
<td>Lower Arm</td>
<td>5 \times 9</td>
<td>405</td>
<td>41</td>
<td>18</td>
</tr>
<tr>
<td>Lower Leg</td>
<td>7 \times 14</td>
<td>882</td>
<td>49</td>
<td>16</td>
</tr>
<tr>
<td>Torso</td>
<td>13 \times 14</td>
<td>1638</td>
<td>38</td>
<td>14</td>
</tr>
<tr>
<td>Upper Arm</td>
<td>5 \times 7</td>
<td>315</td>
<td>48</td>
<td>17</td>
</tr>
<tr>
<td>Upper Leg</td>
<td>7 \times 11</td>
<td>693</td>
<td>43</td>
<td>18</td>
</tr>
</tbody>
</table>

TABLE 3
Classification rates using JRIP classifier and ten fold cross validation. TP - True Positive, FN - False Negative, TN - True Negative, FP - False Positive.

<table>
<thead>
<tr>
<th>Part</th>
<th>TP</th>
<th>FN</th>
<th>TN</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head</td>
<td>0.958</td>
<td>0.042</td>
<td>0.968</td>
<td>0.032</td>
</tr>
<tr>
<td>Lower Arm</td>
<td>0.913</td>
<td>0.087</td>
<td>0.863</td>
<td>0.137</td>
</tr>
<tr>
<td>Lower Leg</td>
<td>0.924</td>
<td>0.076</td>
<td>0.933</td>
<td>0.067</td>
</tr>
<tr>
<td>Torso</td>
<td>0.940</td>
<td>0.060</td>
<td>0.958</td>
<td>0.042</td>
</tr>
<tr>
<td>Upper Arm</td>
<td>0.840</td>
<td>0.160</td>
<td>0.933</td>
<td>0.067</td>
</tr>
<tr>
<td>Upper Leg</td>
<td>0.913</td>
<td>0.087</td>
<td>0.941</td>
<td>0.059</td>
</tr>
</tbody>
</table>

delta-samples are drawn, where a single delta sample represents the state of only a single part.

Often there are several positive detections where a person is located at small perturbations in position and scale relative to the true location of the person. Therefore, after applying our detector we apply the mean-shift algorithm to cluster the detections. The location of the subject is then taken as the position of the nearest hyper-sample to the expectation value of the cluster using the marginal likelihoods of each detector in the cluster. Given the set of hyper-samples at the located position (one for each orientation), the correct orientation is extracted using the SVM detector.

The same JRIP detector is used on both data sets. Each HOG feature consists of 9 unsigned orientation bins. The number of HOG features used for each detection window can be seen in Table 2. The resultant number of attributes this creates for each detection window is also shown.

A training set for each part was created using data taken over all actions and subjects from the Train partition of the HumanEva dataset [25]. MoCap data was used to select positive examples and negative examples. In total 1000 positive examples were selected and 2000 negative. The resultant JRIP detectors learned are described in Table 2. An average reduction of 93% in the number of attributes is achieved. This makes the approach far more efficient since on average only 7% of the features needs to be extracted from the image. Each classifier has less than 20 rules and the maximum rule length was just 5 conditions. The number of rules gives some indication to how difficult each classification problem is. For example it seems the head and torso are much easier to detect than all other parts.

In Table 3 the classification rates for these detectors are shown using ten fold cross validation. The most difficult part to detect accurately is the upper arm, this is most likely as it is often occluded by the lower arm.

To improve the accuracy of the extracted pose we further iterate the proposed method using additional image features. We use a publicly available skin detec-
Using 10 fold cross validation we achieve an accuracy so that $X \in \mathbb{R}^{10}$ and $\mathcal{Y}^m \in \mathbb{R}^{10}$. The ROC curve is shown in Figure 4 and as can be seen the performance of the detector significantly improves if the correct orientation is known. Whilst this is of limited practical use in detector design it suggests that the features for the positive examples more tightly cluster if the correct orientation is known. The next approach was to use the concatenated vector $V(\mathcal{V}) \in \mathbb{R}^{160}$. This significantly outperformed using a side view detector and also the detector that assumes the correct orientation as the person being detected. A rank the correct root position. A rank $R$ is anything worse than this. Over the 50 test images the ranking scores are 24, 12 and 14 for $R_1$, $R_2$ and $R_3$, respectively. Example frames showing the detector’s performance for each are presented in Figure 5. The conditions we use to rank the detections are quite stringent since a small error in the estimated position of the root node can make a significant difference in the resultant pose.

In Figure 6 we show some examples of the extracted poses. In (a), (b) and (d) we visualize the conditional posterior distribution for the optimal hyper-sample, $p(X|Z, z^{true}_m)$, by extracting delta-samples ((a) and (d)) and by plotting the GMM components (b). As is clearly shown the resultant posterior is still highly multimodal allowing the opportunity for further optimization based on higher level priors or temporal integration. In particular notice in the example in the top row that when the samples are rotated slightly as shown in (d), it can be seen that the front leg in the image is represented by both a mode for the left leg and the right. Note also that although we only visualize the distribution for the most likely hyper-sample, all other hyper-samples for all positions and orientations are still maintained and can be accessed if needed. In (c) the MAP estimates are shown for each image, as can be seen these closely relate to the images shown.

In Figure 7, we illustrate the most common cause of errors. As the model is represented using a tree in (a) we see two limbs fitting to the same mode. This is a common problem with all tree based methods. Errors shown in (b)-(d) are all as a result of incorrect root node estimation, whilst (b) and (c) are due to poor position estimation, (d) shows the wrong orientation has been detected. However, it would be expected that by increasing the resolution of the search over the root node’s position and orientation some of these errors would be reduced.

4.4.2 HumanEva Dataset
The HumanEva dataset is used to provide quantitative results of the extracted poses. Detectors are trained using the train partition of the dataset taken across all subjects. The prior is learned for each subject using motion capture data from the corresponding train partition for that person. For training and testing we use Camera 3 since this is the only view where the subject remains fully in view throughout the sequence and therefore can guarantee we have equal number of poses for all orientations. This is done to ensure no bias is added to the learned orientation detector.

Example extracted poses are shown in Figure 8. As can be seen the poses shown closely match those of the subject depicted in each sequence. Quantitative results from the Validate partition of the HumanEva dataset, using every fourth frame of the walking sequence are presented in Table 4. These show the error averaged across all joints of the model. As the method is monocular, we present both the relative and absolute error. The absolute error is dominated by errors in estimating the root node state; often the
Fig. 5. Example frames showing detector bounding box and ranking. (a) Examples with ranking $R_1$. (b) Examples with ranking $R_2$. (c) Examples with ranking $R_3$.

Fig. 6. Examples of estimated pose and conditional posterior distribution, $p(X|Z,z^{*}_{m})$. (a) Projection of delta-samples drawn from conditional posterior distribution. (b) Visualization of GMM modes. (c) MAP estimate of pose. (d) Visualizing delta-samples from alternative view.

Fig. 7. Examples of typical failure cases. (a) Over-counting - both legs are attracted to the same mode. (b) Poor depth estimation. (c) Incorrect root node position estimation. (d) Incorrect root node orientation estimation.

Fig. 8. Examples of extracted 3D poses. A different subject is shown in each row.

The hardest component to extract is the correct depth. However, even if the depth is underestimated a good representation of pose can still often be extracted resulting in a reduced relative pose error. Also for comparison we present the error when the position of the root node is given, and only the orientation and pose is unknown. As can be seen the relative error increase by only about 20mm when the position is unknown. For comparison we compare our method with [13], who use a 2D Pictorial Structure and then ‘lift’ this to 3D using exemplars, they also have a temporal prior modeled using a hierarchical Gaussian Process Latent Variable model. Tested on walking in a similar sequence they reported a 3D reconstruction error of 104mm and we achieve an error of 104.5mm averaged over all subjects. However, their approach has a much stronger prior distribution and they use observations made over multiple frames, where as currently we use only a single image. The benefit of our approach is that it is less dependent on the initial prior model, for example no correlations are learned between unconnected parts making it more general.
can be added by inflating the covariance of each distribution and messages between connected parts can be computed as a product of Gaussian’s. This makes performing inference for each hyper-sample very efficient. Furthermore, we present a method to deterministically extract a sparse set of delta-samples from each distribution. This is motivated by minimizing the KL-divergence between the distribution of the delta-samples and the PDF they are used to approximate. Using this approach each hyper-sample is updated using the equivalent number of image likelihood evaluations as just seven delta-samples in a typical particle filtering approach.

5.1 Tracking and Pose Estimation

Performing a joint optimization over both time and space using a part based approach results in a complex graphical model that is difficult to solve. We take a common approach and assume that tracking can be performed independently to pose estimation and each can be performed in turn.

The graphical model used to represent a person is shown in Table 1. The state of each part is again represented by a quaternion rotation $q_t$ that describes the orientation of each part in the frame of reference of the root node. The root node $x_r$ does not explicitly represent a part, its state represents the position $d_r \in \mathbb{R}^3$ and orientation $\theta_r \in \mathbb{R}^3$ of the body in the global frame of reference, i.e. that of the motion capture suite.

The PDF for each individual node of a hyper-sample is modeled using a Gaussian distribution, so that $p(x_i) = \mathcal{N}(x_i; \mu_i, \Sigma_i)$, where $\mu_i$ and $\Sigma_i$ represent the Gaussian’s mean and covariance respectively. Therefore, each hyper-sample is parameterized by $S^m = \{x^m_1, \mu^m_1, \Sigma^m_1, ..., \mu^m_{n-1}, \Sigma^m_{n-1}, w^m_m\}$.

The posterior at time $t-1$ is represented by a set of $M$ hyper-samples, so that $p(x_{t-1}|z_{t-1}, ..., z_t) \approx \sum_{m=1}^{M} \mathcal{N}(x_{t-1}|z_{t-1}, ..., z_t) \approx \sum_{m=1}^{M} \mathcal{N}(x_{t-1}^m|z_{t-1}^m, ..., z_t^m)$.

5 Application 2: Multiple View 3D Tracking

In this section, we apply our framework to the problem of tracking a person in 3D using multiple views. The benefit of our approach in this setting is that we can use a set of hyper-samples to represent a much larger volume of the state space than existing methods that typically converge to a single solution or represent very few modes (e.g. [3]). In effect a separate mode is represented by each of our hyper-samples making it a very rich representation. The advantage of this is that it enables our approach to be much more robust to tracking failure. Tracking failure typically occurs when the incorrect mode is tracked. This ‘incorrectness’ is not the fault of the algorithm and does not imply it has failed to track the global maximum. The problem is that observations, even using multiple views, are ambiguous and noisy, therefore it is conceivable that often the global maximum of the posterior is incorrect (i.e. it does not correspond to the true pose). Therefore, if only a single or very few modes are being tracked failure is very likely. Our contribution therefore, is not to design an approach that can most efficiently find the global maximum, which is the focus of much of the tracking literature, but to develop an approach that can support a much larger area of the posterior without further computational cost (i.e. without extra likelihood function evaluations). Therefore, if the global mode is incorrect due to noisy observations the approach is less likely to suffer catastrophic tracking failure. This is achieved by broadly distributing the root node states of the hyper-samples. The effect of this is that it permits greater uncertainty to be represented over the state of the root node, though this is achieved without then propagating this uncertainty to the remaining parts of the model. We show this not to be the case if using existing standard methods, where adding uncertainty to the root node also inflates the uncertainty of all remaining parts of the model.

Further benefits of our approach is that as the PDF over the state space, excluding the root node, is represented parametrically these can be updated in closed-form. For example temporal diffusion across frames

<table>
<thead>
<tr>
<th>Subject</th>
<th>Known root position</th>
<th>Rel. 3D error</th>
<th>Abs. 3D error</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>77.7</td>
<td>106.1</td>
<td>256.9</td>
</tr>
<tr>
<td>S2</td>
<td>62.1</td>
<td>83.3</td>
<td>200.1</td>
</tr>
<tr>
<td>S3</td>
<td>110.3</td>
<td>124.2</td>
<td>203.7</td>
</tr>
<tr>
<td>Average</td>
<td>83.3</td>
<td>104.5</td>
<td>238.8</td>
</tr>
</tbody>
</table>
is connected. Hence, \( p(\tilde{x}_{ij}) = \mathcal{N}(\tilde{x}_{ij}, \mu_{ij}, \Sigma_{ij}) \), where this distribution is learned over \( q_{ij} = (q_{ij}^{-1})^T q_{ij}^{t+1} \).

Given a value for \( x_j \) this can then be transformed to a distribution over \( x_i \) by:

\[
p(x_{i,t}|x_{i,t-1}) = \mathcal{N}(x_i, \mu_i, \Sigma_i) \approx \mathcal{F} \left( x_j, \mathcal{N}(x_{ij}, \mu_{ij}, \Sigma_{ij}) \right).
\]

(11)

The transformation, \( \mathcal{F} \left( x_j, \mathcal{N}(x_{ij}, \mu_{ij}, \Sigma_{ij}) \right) \), is non-linear and is performed using the Unscented Transform [37]. This method decomposes the covariance into a set of 2D sigma points \( \Sigma = \{\sigma_1, \ldots, \sigma_D\} \), where \( D \) is the dimension of the covariance. Each sigma point is then translated by the mean to generate a set of points that represent the mean and covariance of the original distribution. Each sigma point is calculated as

\[
\sigma_d = \mu + \sqrt{D_v} e_d, \\
\sigma_{D+d} = \mu - \sqrt{D_v} e_d,
\]

(12)

where \( v_d \) and \( e_d \) represents the \( d \)th eigenvalue and eigenvector of the covariance matrix. Each sigma point is then propagated through the non-linear function (i.e. \( \sigma'_m = q_i \sigma_m \)) and the mean and covariance calculated from them. In Figure 9 we show temporal diffusion, using this method, applied to a single hyper-sample for three consecutive frames. The covariance growth across the frames is cumulative.

![Fig. 9. Example showing temporal diffusion applied to the covariances of a single hyper-sample.](image)

During resampling, methods from annealing are used to adjust the weight of the hyper-samples, such that \( w' = (w)^\beta \). A value of \( \beta \) is selected such that the particle survival rate \( \alpha \) can be estimated over the entire set of hyper-samples as described in [3]. Given a set of samples tracked over \( t \) frames the survival rate will decrease according to \( \alpha^t \). To allow the same survival rate to be maintained over a fixed time interval, \( \alpha \) is set according to \( \alpha = \exp \frac{\ln \alpha_c}{N_t} \), where \( \alpha_c \) is the desired cumulative survival rate per second and \( N_t \) is the frame rate. This is used so that the uncertainty over the root node can be consistent regardless of the frame rate. A larger value of \( \alpha_c \) will allow the distribution of the hyper-samples to spread over a larger area of the root node state space, since more of the sample population will be maintained. This will provide wider support of the posterior distribution.

The process outlined in this section has described how a new set of hyper-samples are drawn from an old set to provide a prior for the current frame, the second stage in our tracking approach is to update each hyper-sample given local observations. It is this process that we describe next.

### 5.2 Local Solution Estimation

In this section we describe how a single hyper-sample is optimized to find a local solution. Since we assume this can be performed independent of time we drop the temporal indices for brevity. Though note the process described must be performed for each hyper-sample in turn.

#### 5.2.1 Limb Conditionals

As the graph used for monocular pose estimation was a star, limb conditionals only needed to be learned between the root node and each node connected to it. This was implicitly represented by the prior distribution, \( P(X|\Phi) \). However, for tracking we use a more complex tree, as such limb conditionals are required for all connected parts. These describe how two connected parts can deform relative to one another and are described by the distribution \( p(x_j|x_i, \theta_{ij}) \), where \( \theta_{ij} \) is the connection parameter. Rather than learning a full limb conditional over \( x_i \) and \( x_j \) we follow the approximation in [2] and learn a distribution over \( x_{ij} \) (i.e. \( p(x_{ij}|\theta_{ij}) \)). This distribution is also learned over unit quaternions, where \( q_{ij} = q_i^{-1} q_j \), and the connection parameters are defined as the mean, \( \mu_{ij} \), and covariance, \( \Sigma_{ij} \), of a Gaussian distribution. Given a state for \( x_i \) a PDF over \( x_j \) can be estimated by propagating the distribution, \( \mathcal{N}(x_{ij}; \mu_{ij}, \Sigma_{ij}) \), through the rotation \( q_i \). This is also performed using the Unscented Transform, used in the previous section, and is described by

\[
p(x_j|x_i, \theta_{ij}) \approx \mathcal{F} (q_i, \mathcal{N}(x_{ij}; \mu_{ij}, \Sigma_{ij})).
\]

(13)

An example of the PDFs for the lower legs given the state shown of the upper legs are presented in Figure 10. As would be expected the greatest uncertainty is along the direction that the lower leg can rotate about the knee.

![Fig. 10. Visualizing the PDF of the lower legs’ state given that of the upper legs.](image)
5.2.2 Extracting delta-samples

As described in Section 3.2, each local solution is found by extracting a set of \( d \) delta-samples from the PDF. This could be achieved by drawing random delta-samples as used for monocular pose estimation. However, many samples may be required to give confidence that the sampled distribution is accurately represented by this sample set. One method to provide a confidence in the ability of a sample set to represent the PDF is to measure the KL-divergence between the covariance and mean of the samples and that of the original distribution. The closer to zero this measure is, the more confidence we have. Instead of this, Equation (12), used to select a set of Sigma points, provides a means to deterministically select a set of delta-samples that exactly represent the covariance and mean of the original distribution, ensuring the KL-divergence between them is zero. We therefore sample from \( p(x_i | x_i^m) \) by decomposing the distribution \( N(x_i | \mu_i^m, \Sigma_i^m) \) into a set of sigma points using Equation 12, except that a copy of the mean is also maintained. So instead \( 2D + 1 \) sigma points are selected and each scaled by \( \sqrt{(D + 1/2)v_d} \).

These samples can then be used to update the model parameters by calculating the likelihood at each sigma point, \( \pi_i^s = p(z_i | x_i^s) \) and using these to re-estimate the mean and covariance, \( (\mu_i^{m*}, \Sigma_i^{m*}) \). An example of the delta-samples used to represent a single hyper-sample is shown in Figure 11, projected into two different camera views. The benefit of this approach is that it requires just 7 delta-samples to be extracted for each node of each hyper-sample. This makes the approach extremely efficient, since the bottleneck in pose estimation is typically the evaluation of the observational likelihood.

Fig. 11. An example of a set of sample points used to estimate observational likelihood distributions projected into two views. They represent the distributions shown on the left.

The observational likelihood used for tracking is based on the binary silhouette. Given a silhouette \( B \) and the set of image pixels \( P \), pixels classified as the foreground are set to one \( B(P_{fg}) := 1 \) and those classified as the background are set to zero \( B(P_{bg}) := 0 \). The appearance of a part is dependent on \( x_i^s \), since this will cause changes in scale due to depth or foreshortening due to orientation. The projection of the part consists of the pixels \( L(x_i^s) \subset P \) and the cost is defined as \( p(z_i | x_i^s) \propto \sum_{l \in L(x_i^s)} B(l) \).

To prevent different limbs being assigned to the same mode (over counting), each constructs a version of the binary silhouette for the opposing part \( B_{opp(i)} \), given by

\[
B_{opp(i)}(L(x_i^s) \cap P_{fg}) := 0.5,
\]

This makes it preferable for a limb to be located where the opposing limb is not predicted to be, whilst preferring this over locating a limb to a region of the image classified as the background.

5.2.3 Calculating Beliefs

In this section, we describe how the states of the nodes are updated for each hyper-sample \( S^m \) using message passing between nodes. As described in Section 5.2.2, the belief for the \( j \)th node conditioned on all observations and a given root node state is calculated as

\[
p(x_j | Z, x_i^m) = p(x_j | z_j) \prod_{v_i \in \mathcal{E}(j)} p(x_j | z_i, ..., z_T),
\]

where \( v_i \in \mathcal{E}(j) \) defines the set of edges connected to \( j \) and \( \{z_i, ..., z_T\} \) represents the set of observations for the subtree containing \( v_i \), created by removing the edge \( \{v_i, v_j\} \). We presume the observational likelihood function has already been computed as described in Section 5.2.2, and is given by \( p(x_j | z_j) = N(x_j; \mu_j^{mix}, \Sigma_j^{mix}) \). Since all distributions are modeled as Gaussians the above products can be calculated in a closed form through

\[
N(x_j; \mu_j, \Sigma_j) = N(x_j; \mu_j^{mix}, \Sigma_j^{mix}) \prod_{v_i \in \mathcal{E}(j)} N(x_j; \mu_j^{\tilde{i}}, \Sigma_j^{\tilde{i}}),
\]

where \( N(x_j; \mu_j^{\tilde{i}}, \Sigma_j^{\tilde{i}}) \) represents the message from \( i \) to \( j \) and the product of two Gaussian distributions results in a Gaussian with parameters

\[
\Sigma_k = (\Sigma_i^{-1} + \Sigma_j^{-1})^{-1}, \quad \mu_k = \Sigma_k (\Sigma_i^{-1} \mu_i + \Sigma_j^{-1} \mu_j).
\]

Messages are calculated in two steps. Firstly, incoming messages from all nodes other than the node to which the message is being passed are combined with the observational likelihood function

\[
N(x_i; \mu_i^{\tilde{j}}, \Sigma_i^{\tilde{j}}) = \prod_{v_k \in \mathcal{E}(i) / j} N(x_i; \mu_i^{\tilde{k}}, \Sigma_i^{\tilde{k}}).
\]

Secondly, this distribution is propagated through the limb conditional \( p(x_j | \theta_{i,j}) \), so that the message is a distribution over \( x_j \). This is performed using the Unscented Transform by propagating \( N(x_i; \mu_i^{\tilde{j}}, \Sigma_i^{\tilde{j}}) \) through the rotation defined by the mean of the limb conditional. Hence,

\[
N(x_j; \mu_j^{msg}, \Sigma_j^{msg}) = \mathcal{F} \left( N(x_i; \mu_i^{\tilde{j}}, \Sigma_i^{\tilde{j}}), \mu_j \right).
\]
Whilst this propagates the uncertainty in the initial message, the uncertainty in the limb conditional $\Sigma_{ij}$ must also be passed. This is achieved again using the Unscented Transform and propagating the limb conditional thorough the mean of the initial message $\mu_{ij}$,

$$\mathcal{N}(\mathbf{x}_j; \mu_j^{mod}, \Sigma_j^{mod}) = \mathcal{F} \left( \mu_{ij}, \mathcal{N}(\mathbf{x}_{ij}; \mu_{ij}, \Sigma_{ij}) \right).$$

The final message is then given by the convolution of the two of these distributions, setting $\mu_{msg} := 0$. Therefore the message parameters in Equation (16) are given by $\mu_{ij} = \mu_{ij}^{mod}$ and $\Sigma_{ij} = \Sigma_{ij}^{msg} + \Sigma_{ij}^{mod}$. The posterior for each node can then be updated using (16). The root node does not send messages since it is fixed, however, the marginal for a given root node state is approximated as

$$p(\mathbf{x}_m^r | \mathbf{Z}) \approx \prod_{i=1}^{n-1} \sum_{s=1}^{7} \pi_i^s,$$

where $\pi_i^s$ is the weight of the delta-sample drawn for the $i$th part. A hyper-sample $S^m$ then consists of a root node state, a set of updated Gaussian distributions and a weight. The Maximum A Posterior (MAP) pose $X^{MAP}$ is given by the set of Gaussian centers of the hyper-sample with the highest weight, $X^{MAP} = \{x_m^*, \mu_1^m, .., \mu_n^m \}$, where $m^* = \text{argmax}_m \ p(\mathbf{x}_m^m | \mathbf{Z})$.

Once each hyper-sample has been updated they are then propagated through the temporal priors described in Section 5.1. The new distribution then acts as a prior for the following frame.

5.3 Experiments and Results

The presented method was tested using the HumanEva dataset. The ‘Train’ partition of walking and jogging, consisting of only motion capture data, was used to learn all model parameters. The first 300 frames of the ‘Validation’ partition was used for testing. Three camera views (C1, C2 and C3) were used and foreground/background segmentation was performed using the Matlab code provided with the data set using default settings.

The presented approach was tested against two existing methods, the Annealed Particle Filter (APF) and the Sequential Importance Resampling Particle Filter (SIR-PF). All methods use the same model parameters, however, whilst the presented method adds temporal diffusion by directly inflating the covariance of each part the alternative methods perform this step stochastically. The APF allows the presented method to be tested against an approach that converges to a single mode. Whilst the SIR-PF can be used to examine how existing approaches behave when permitted to support a larger area of the posterior. This is controlled by adjusting the particle survival rate. To make sure the APF converge to a single mode we use a survival rate per frame of 0.03. To allow both the SIR-PF and the presented method to support a larger
area of the posterior, we use a survival rate per frame of 0.93, whilst tracking from video captured at 60Hz.

To ensure the computational cost of each method is the same, all methods use the same number of image likelihood evaluations. The APF uses 5 layers of 160 particles and the SIR-PF use a single layer of 800 particles. The presented method used 114 hyper-samples, since calculating the posterior for each hyper-sample requires the equivalent image likelihood evaluations as 7 SIR-PF/APF particles.

For the APF, pose was estimated using the expectation value of the samples and for the SIR-PF and the proposed method the MAP estimate was used. Limb limits were learned from the training data and used to discard unlikely poses for all methods. It was noted that often the error was dominated by left/right leg ambiguities, to overcome this a mutation was occasionally applied to a particle to swap the legs. This step was performed stochastically during resampling and applied to all methods.

In Figure 12, the set of particles shown represent the posterior for the proposed method and the SIR-PF. As can be seen if the SIR-PF is used to represent a large uncertainty, this uncertainty is present in all parts of the model. This is in contrast to the proposed method where the posterior for each part is updated conditioned on the root node value of the particle, allowing the uncertainty in these parts to remain small. This allows a large region of the root node state to be supported without increasing the uncertainty of the remaining parts. Example frames showing the estimated pose using the presented method are shown in Figure 13, as can be seen the estimated pose closely resembles that of the subject in each frame.

In Table 5, the average error is shown for each subject whilst walking. Over all subjects the proposed method outperforms both the APF and the SIR-PF. Our claim in this work is that in comparison to the APF our method is more robust to tracking failure since we support a larger area of the posterior. However, we note that if the APF can accurately track the root node it is likely to be comparable to our method since the APF uses 800 particles compared to the presented approach that uses the equivalent of 7 to update the posterior for a given root node hypothesis. This is observed in Table 5 where for S2 and S3 the APF outperforms the presented method. However, as can be seen for S1 the APF fails and our method significantly outperforms all others demonstrating its robustness. This is further illustrated in Figure 14

<table>
<thead>
<tr>
<th>Method</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>APF</td>
<td>194.2</td>
<td>79.0</td>
<td>87.7</td>
<td>118.9 ± 65.5</td>
</tr>
<tr>
<td>SIR-PF</td>
<td>105.1</td>
<td>93.0</td>
<td>109.2</td>
<td>102.5 ± 8.4</td>
</tr>
<tr>
<td>Proposed</td>
<td>87.5</td>
<td>95.2</td>
<td>98.5</td>
<td>93.7 ± 5.8</td>
</tr>
</tbody>
</table>

Fig. 14. Tracking error in each frame for the APF (blue) and the proposed method (red).

<table>
<thead>
<tr>
<th>Method</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>APF</td>
<td>200.7</td>
<td>120.0</td>
<td>117.9</td>
<td>146.2 ± 47.2</td>
</tr>
<tr>
<td>SIR-PF</td>
<td>105.1</td>
<td>105.2</td>
<td>120.7</td>
<td>110.4 ± 8.9</td>
</tr>
<tr>
<td>Proposed</td>
<td>89.3</td>
<td>108.7</td>
<td>113.5</td>
<td>103.8 ± 12.8</td>
</tr>
</tbody>
</table>

where an example of the tracking error in each frame is shown for both the proposed method and the APF. During the first 60 frames when the root node is accurately tracked the error is slightly lower for the APF, however, beyond this the APF fails whilst the proposed method is able to continue tracking the subject.

To further illustrate the robustness of the presented method we examine using just 2 camera views. Fewer camera views will result in more ambiguous observations and in these circumstances it will be beneficial to support the posterior over a larger area of the state space until these ambiguities can be resolved. The results are presented in Table 6. As expected the APF is more prone to tracking failure and our method outperforms both the SIR-PF and the APF.

We further experimented using three cameras but at different frame rates. For all frame rates the annealing rate is adjusted for the SIR-PF and presented method to maintain $\alpha_c = 0.01$, as described in Section 5.1. The annealing for the APF is unchanged to ensure it converges to a single mode. The error for each averaged across all subjects are presented in Figure 15. At lower frame rates, when there is greater movement by the subject across consecutive frames, the APF becomes more prone to tracking failure and the presented method continues to outperform both techniques across all frame rates, highlighting its superiority. In Figure 16 we show some example frames of the MAP pose and the distribution of samples used to represent the posterior, whilst tracking at 30Hz. Compared to Figure 12 operating at 60Hz, we note that the distribution is slightly broader, this is due to the weakness of the simplistic temporal prior and as fewer observations are integrated since the data has been down sampled.

Whilst in some instances the quantitative errors between the proposed method and the SIR-PF are rel-
attractively close, qualitatively the tracking is significantly poorer for the SIR-PF. In Figure 17, example frames are shown comparing the MAP solution using the SIR-PF compared to the proposed method. As can be seen the poses estimated by the SIR-PF are notably worse than those estimated by the proposed method. We observed that in general unrecoverable tracking failure for the APF resulted from poorly estimating the state of the root node, for example by estimating the incorrect orientation. This observation highlights the importance of representing greater uncertainty over the root node to develop robust tracking algorithms for articulated objects.

6 CONCLUSIONS

In this paper we have presented two novel approaches to extract 3D human pose. The first was from a single monocular image and the second was applied to multi-view tracking. Both approaches were designed to exploit the key assumption that it is easier to estimate pose if the root node state is known a priori. This was achieved by extracting a set of local solutions through the use of hyper-samples. There are two key benefits to this approach that we have exposed. The first is that using a fixed root node allows the human body to be modeled as a kinematic chain that can more efficiently be optimized than alternative representations. The second is that the presented approach allows more of the posterior to be supported than current methods allow. By exploiting the first benefit we have shown it is possible to extract an entire set of solutions using the same computational cost as competing methods would require to find a single solution. The second benefit has been used to engineer a tracking method that is robust in the presence of noisy, ambiguous observations, and to design a single image monocular approach that is not dependent on initialization.

For tracking it was shown that more robust performance can be achieved by providing greater support over the state of the root node. This was particularly emphasized at lower frame rates, where noise and missing data becomes much more detrimental as the weakness of the simple temporal prior becomes exposed. The philosophy of this approach is far removed from the most common to assume the answer lies in strengthening the temporal prior, we believe the solution lies in strengthening the support over the posterior distribution until stronger, more informative observations become available.

REFERENCES

Fig. 16. Example frames showing the MAP estimate of pose (top) and distribution of the samples used to represent the posterior (bottom) whilst tracking at 30Hz.

Fig. 17. Comparison of pose estimation between the SIR-PF (top row) and proposed method (bottom row).