3D Pictorial Structures for Multiple View Articulated Pose Estimation

Magnus Burenius, Josephine Sullivan, Stefan Carlsson
CVAP, KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

We consider the problem of automatically estimating the 3D pose of humans from images, taken from multiple calibrated views. We show that it is possible and tractable to extend the pictorial structures framework, popular for 2D pose estimation, to 3D. We discuss how to use this framework to impose view, skeleton, joint angle and intersection constraints in 3D. The 3D pictorial structures are evaluated on multiple view data from a professional football game. The evaluation is focused on computational tractability, but we also demonstrate how a simple 2D part detector can be plugged into the framework.

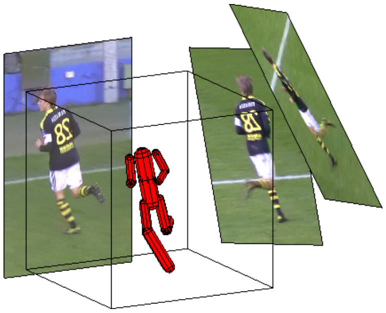

Figure 1. We discretize the space of human 3D poses and find the pose that best fits the images from a set of calibrated cameras, using dynamic programming.

1. Introduction

Human pose estimation is an important problem in computer vision [11]. It comes in many different flavors depending on the final goal and the assumptions made:

- Estimate pose in 2D or 3D.
- Estimate pose from a single time frame or a sequence.
- Estimate pose from a single camera view or multiple.
- Impose a weak or strong prior on the pose.

In this paper we focus on human pose estimation in 3D, at a single time frame, using multiple views, imposing a weak pose prior. We explore how pictorial structures can be used to solve this problem.

From a wider perspective, pictorial structures are interesting since they might provide a unifying framework for general pose estimation and object detection in both 2D and 3D. They are also interesting from a practical point of view, due to their efficiency. Pictorial structures simplify the inference over the high-dimensional space of human poses, by modeling the dependencies between body parts as a tree structure, as opposed to a general graph.

Pictorial structures in 2D typically discretize the search space. Using dynamic programming over the tree graph a global optimum of a cost function is computed. This is the state-of-the-art for single view human 2D pose estimation [9, 8, 16, 1]. The pictorial structures framework also works well for general 2D object detection. The deformable part model [7], which fits this framework, provides state-of-the-art performance for this problem. Recently this type of model has also been extended to 3D pose estimation of general objects [12], where in this case pose corresponds to the single overall rotation of the object relative to the camera.

However, pictorial structures have not been used as much for 3D pose estimation of humans, or articulated objects in general. Bergtholdt et al. [2] do multiple view 3D pose estimation, by first inferring the 2D pose in each view. They couple the inference over the different views by enforcing soft epipolar constraints. In this way 3D information is taken into account although the search is done in 2D. A disadvantage with this approach is that the coupling of views cannot be implemented in a tree graph. By using a general graph the inference of a global optimum is not tractable.

Sigal et al. [15] on the other hand perform the search in 3D. They argue that while efficient 2D pose estimation relies on a discretization, this is not practical in 3D. Therefore they use a stochastic algorithm to perform inference over a continuous space. This has two disadvantages compared to the discretized pictorial structures, commonly used in 2D. The stochastic algorithm is more complicated and it cannot give the same guarantee of global optimality as dynamic programming over a discrete space.

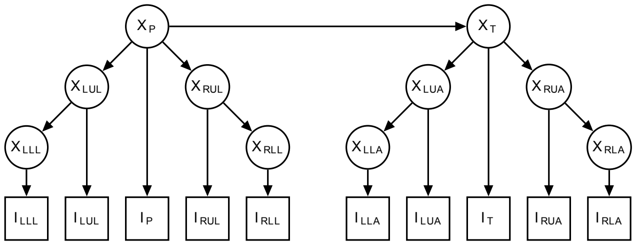

Figure 2. The Bayesian network of our model. The body parts are in topological order: Pelvis, Torso, Left Upper Leg, Right Upper Leg, Left Upper Arm, Right Upper Arm, Left Lower Leg, Right Lower Leg, Left Lower Arm, Right Lower Arm. The square nodes represent measured variables.

Discretizing the space of 3D poses is difficult for many reasons. 2D rotations are simply described by a single angle, which can be used to create a grid of evenly spread rotations, such that two discrete rotations can be composed to another discrete rotation. The space of 3D rotations is more complicated and has no gold-standard parametrization. It is not obvious how to create a discrete set of 3D rotations that are evenly spread and can be easily composed. Also, in 2D the general distance transform is used to give efficient inference [8]. It is not clear how to generalize this to 3D rotations. Furthermore, the space of translations and rotations in 2D together form a 3D space, whereas the space of translations and rotations in 3D together form a 6D space. A discretization of 3D poses would therefore require considerably more points. It is unclear whether dynamic programming is tractable over this larger space. The goal of this paper is to address these issues. We aim to show that discrete pictorial structures in 3D are practical and tractable.

Our model is described in section 2. We first describe the general framework, which is more or less the same in 2D and 3D, and then describe the aspects unique to 3D. In section 2.1 we discuss weak pose priors leading to tractable inference. These impose skeleton and joint angle constraints. In section 2.2 we discuss how to create a discrete search grid over 3D poses. The problem of double-counting, typical for tree-based models, is discussed in section 2.3. In the experiments section 3 we evaluate our model on multiple view data from a professional football game. First the tractability is evaluated and then we implement and evaluate a simple HOG-based part detector.

2. Model

In this section we initially present a general overview of our model and framework that is consistent with pictorial structures in 2D. The details specific to a 3D implementation are then discussed.

The human body is modeled as a collection of N body parts. The state $X_n = (T_n, R_n)$ of each part n is defined by its global translation T_n and global rotation R_n in 3D. Each is considered as a discrete random variable. Outcomes of these random variables are denoted by $x_n = (t_n, r_n)$ and assumed to be elements of the discrete set $\Omega_X = \Omega_T \times \Omega_R$. In section 2.2 we discuss how the space of translations and rotations in 3D are discretized to give $\Omega_T \subset \mathbb{R}^3$ and $\Omega_R \subset \mathbb{SO}(3)$. The state of all parts is represented by $X = (X_1, \ldots, X_N)$. We assume the parts are connected in a tree graph and the state of part n only depends on the state of its parent $pa(n)$:

$$P_{X_n|X}(x_n \mid x) = P_{X_n|X_{pa(n)}}(x_n \mid x_{pa(n)}) \tag{1}$$

The joint distribution of all parts then factorizes as:

$$P_X(x) = \prod_n P_{X_n|X_{pa(n)}}(x_n \mid x_{pa(n)}) \tag{2}$$

Our goal is to estimate the state of the parts from image measurements. Let $I_n = (I_n^1, \ldots, I_n^C)$ be a random variable representing the image evidence from C views of part n. We assume the evidence from different views are independent and therefore the likelihood of part n in state x_n generating the image evidence i_n can be written in terms of the likelihood functions for each view:

$$P_{I_n|X_n}(i_n \mid x_n) = \prod_c P_{I_n^c|X_n}(i_n^c \mid x_n) \tag{3}$$

This likelihood provides an image matching score or a goodness-of-fit to all camera views for a part given its state, thereby imposing view constraints. If $I = (I_1, \ldots, I_N)$ is the image evidence of all parts and we assume I_n is conditionally independent of all $I \setminus I_n$ given X_n, the full joint distribution over all the random variables factorizes as:

$$P_{X,I}(x, i) = \prod_n P_{I_n|X_n}(i_n \mid x_n)\; P_{X_n|X_{\mathrm{pa}(n)}}(x_n \mid x_{\mathrm{pa}(n)}) \tag{4}$$

The Bayesian network in figure 2 displays the assumed dependency structure of the variables in our model. We want to find the most probable state x^* of the parts given measurements of their images i. This corresponds to solving the discrete optimization problem:

$$x^* = \arg\max_x P_{X|I}(x \mid i) = \arg\max_x P_{X,I}(x, i) \tag{5}$$

Since the objective function is factorized over a tree graph, the global maximum can be found using the max-product algorithm [3]. See algorithm 1 for its application to our problem. We assume the parts are ordered topologically, i.e. the index of a child is always greater than the index of its parent and we let the root have index 1. The costly part of the algorithm is the optimization problem in the innermost loop:

$$\max_{x_n} \Big(\ln P_{X_n|X_{pa(n)}}(x_n \mid x_p) + m_n(x_n) \Big) \tag{6}$$

Algorithm 1 Max-product for our model

```
$m_n(x_n) := \ln P_{I_n|X_n}(i_n \mid x_n) \quad \forall n$
for $n := N$ to 2
    $p := pa(n)$
    for $x_p \in \Omega_X$
        $\tilde{m} := \max_{x_n} \left(\ln P_{X_n|X_p}(x_n \mid x_p) + m_n(x_n)\right)$
        $m_p(x_p) := m_p(x_p) + \tilde{m}$
    end
end
$x_1^* := \operatorname{argmax}_{x_1} m_1(x_1)$
for $n := 2$ to $N$
    $p := pa(n)$
    $x_n^* := \operatorname{argmax}_{x_n} \left(\ln P_{X_n|X_p}(x_n \mid x_p^*) + m_n(x_n)\right)$
end
```

for all $x_p \in \Omega_X$. The time complexity of this is in general $O(|\Omega_X|^2) = O(|\Omega_T|^2|\Omega_R|^2)$. We consider N to be constant. In section 2.1 we show how to reduce the complexity to $O(|\Omega_T||\Omega_R|)$ and $O(|\Omega_T||\Omega_R|^2)$, by choosing a pose prior $P_{X_n|X_{pa(n)}}$ which exploits the fact that we are modeling a 3D human skeleton.

2.1. Skeleton Model

As we model the human as a kinematic tree, the state $X_n = (T_n, R_n)$ of a child depends only on that of its parent. The global translation T_n and rotation R_n of each part can then be defined recursively in terms of local translations ΔT_n and local rotations ΔR_n of the part and the global translation and rotation of its parent:

$$R_n = R_{pa(n)} \Delta R_n \tag{7}$$

$$T_n = T_{pa(n)} + R_{pa(n)} d_n + \Delta T_n \tag{8}$$

where d_n is a constant vector offset of part n from its parent. We assume the global translation and rotation of the root is uniformly distributed and view the conditional probability $P_{X_n|X_{pa(n)}}$ as a prior on the pose of part n given the pose of its parent. If all the local translations and rotations are assumed to be independent of one another, then T_n and R_n are independent given the parent state:

$$P_{X_n|X_{pa(n)}} = P_{T_n|X_{pa(n)}} P_{R_n|X_{pa(n)}} \tag{9}$$

From equation (7) we see that the rotation of a child is independent of the translation of its parent and R_n is deterministically defined by $R_{pa(n)}$ and ΔR_n. Therefore we have:

$$\begin{aligned} P_{R_n|X_{pa(n)}}(r_n \mid (t_p, r_p)) &= P_{R_n|R_{pa(n)}}(r_n \mid r_p) \\ &= P_{\Delta R_n}(r_p^T r_n) \end{aligned} \tag{10}$$

as $\Delta R_n = R_{pa(n)}^T R_n$. Similarly, the translation prior, by exploiting equation (8), can be expressed as:

$$P_{T_n|X_{pa(n)}}(t_n \mid (t_p, r_p)) = P_{\Delta T_n}(t_n - t_p - r_p\, d_n) \tag{11}$$

The total pose prior thus factorizes as:

$$\begin{aligned} &P_{X_n|X_{pa(n)}}((t_n, r_n) \mid (t_p, r_p)) = \\ &\qquad P_{\Delta T_n}(t_n - t_p - r_p\, d_n)\; P_{\Delta R_n}(r_p^T r_n) \end{aligned} \tag{12}$$

where $P_{\Delta T_n}$ is a prior over the local translations and $P_{\Delta R_n}$ is a prior over the local rotations.

Translation Prior We propose three alternatives for the translation prior. Each alternative provides potential opportunities for speeding up the general max-product algorithm 1. The simplest corresponds to modeling the skeleton as a chain of limbs of fixed length and is expressed with:

$$P_{\Delta T_n}(\Delta t_n) = \begin{cases} 1 & \text{if } \Delta t_n = (0,0,0)^T \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

One can also allow each limb some small degree of flexibility in its length by defining a set M_n of possible deformations such that:

$$P_{\Delta T_n}(\Delta t_n) \propto \begin{cases} 1 & \text{if } \Delta t_n \in M_n \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Another possibility is to use a loose chain model as in standard 2D pictorial structures:

$$P_{\Delta T_n}(\Delta t_n) \propto \mathcal{N}(\Delta t_n \mid (0,0,0)^T, \sigma_n^2\, \mathbb{I}_3) \tag{15}$$

The local translations are then described by a discretized normal distribution with zero mean and isotropic covariance. With this prior the inference can be made efficient using the distance transform [8].

In this work we only explore the fixed length constraint and while it is somewhat restrictive, it is not an unreasonable assumption to make in 3D. This is not the case in 2D where limbs in the image can go through extreme foreshortening due to projection and it is therefore a necessity to allow the length of limbs to vary.

Rotation Prior The distribution $P_{\Delta R_n}$ describes the possible rotations of the joint connecting two parts. In this paper we consider in detail two possibilities. The first is simply a uniform distribution:

$$P_{\Delta R_n}(\Delta r_n) \propto 1 \tag{16}$$

The second alternative we examine is one which enforces hard limits on joint angles:

$$P_{\Delta R_n}(\Delta r_n) \propto \begin{cases} 1 & \text{if } \Delta r_n \in Q_n \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

This type of prior can be expressed conveniently for humans as hard constraints in the Twist-swing parametrization

Algorithm 2 Max-product imposing view and skeleton constraints

```
$m_n(x_n) := \ln P_{I_n|X_n}(i_n \mid x_n) \quad \forall n$
for $n := N$ to 2
    for $t_n \in \Omega_T$
        $\tilde{m}(t_n) := \max_{r_n} m_n((t_n, r_n))$
    end
    $p := pa(n)$
    for $t_p \in \Omega_T$
        for $r_p \in \Omega_R$
            $t_n := t_p + r_p d_n$
            $m_p((t_p, r_p)) := m_p((t_p, r_p)) + \tilde{m}(t_n)$
        end
    end
end
$x_1^* := \operatorname{argmax}_{x_1} m_1(x_1)$
for $n := 2$ to $N$
    $p := pa(n)$
    $t_n^* := t_p^* + r_p^* d_n$
    $r_n^* := \operatorname{argmax}_{r_n} m_n((t_n^*, r_n))$
end
```

of 3D rotations [10]. One could of course learn an arbitrary distribution for $P_{\Delta R_n}$ from training data, however, we discount this alternative in this work as we want to impose as few priors as possible on the expected pose of the subject.

Tractable Max-Product In general, the max-product algorithm 1 has a time complexity of $O(|\Omega_X|^2) = O(|\Omega_T|^2|\Omega_R|^2)$. However, each of the pose prior we suggested allows a speed up of the costly innermost loop maximization (6).

The fixed length prior is deterministic. Thus when looking for the optimal state $x_n = (t_n, r_n)$, we know the translation t_n and only need to search over all rotations r_n. Also, if there is a uniform prior on rotation, we can ignore the constant normalization factor. Using algorithm 2 it is then possible to speed up the optimization to $O(|\Omega_T||\Omega_R|)$.

If we still assume fixed limb lengths but a hard rotation prior we can use algorithm 3, with time complexity $O(|\Omega_T||\Omega_R|^2)$. This is the worst complexity of any combination of the suggested translation and rotation priors.

2.2. Discrete Search Grid

Using dynamic programming to search for the optimal pose requires a discretization of the state space. We have two requirements for this discretization. Firstly, the points should be evenly spread. Secondly, if we add translations or compose rotations, it should be easy to find the resulting discrete point. It is easy to construct such a discretization for the translations $\Omega_T \subset \mathbb{R}^3$, but not as easy for the rotations $\Omega_R \subset \mathbb{SO}(3)$.

Algorithm 3 Max-product imposing view, skeleton and joint angle constraints

```
$m_n(x_n) := \ln P_{I_n|X_n}(i_n \mid x_n) \quad \forall n$
for $n := N$ to 2
    $p := pa(n)$
    for $t_p \in \Omega_T$
        for $r_p \in \Omega_R$
            $t_n := t_p + r_p d_n$
            $\tilde{m} := \max_{\Delta r_n \in Q_n} m_n((t_n, r_p \Delta r_n))$
            $m_p((t_p, r_p)) := m_p((t_p, r_p)) + \tilde{m}$
        end
    end
end
$x_1^* := \operatorname{argmax}_{x_1} m_1(x_1)$
for $n := 2$ to $N$
    $p := pa(n)$
    $t_n^* := t_p^* + r_p^* d_n$
    $r_n^* := \operatorname{argmax}_{\Delta r_n \in Q_n} m_n((t_n^*, r_p^* \Delta r_n))$
end
```

Translation Discretization We assume the subject is roughly localized by a bounding rectangle in each image. We also assume that the cameras are calibrated. Therefore we can compute a bounding cube (fig. 1). The discrete set of translations Ω_T is created as a grid covering this cube (fig. 3).

Rotation Discretization We use best-candidate sampling [13] to generate a discrete set of rotations Ω_R that are evenly spread. First a large set of candidates are generated by sampling rotations uniformly. Then only the candidates furthest away from each other are kept.

For this process we use the unit quaternion representation of rotations [6]. It describes a rotation as a point on the hypersphere $\mathbb{S}^3$ embedded in $\mathbb{R}^4$. It is possible to sample uniformly from $\mathbb{SO}(3)$ by sampling points on $\mathbb{S}^3$ uniformly. To do this simply sample a vector in $\mathbb{R}^4$ from an isotropic and zero mean normal distribution and normalize this vector.

After many candidates have been generated we want to retain those samples furthest away from each other. This requires measuring distances between points in $\mathbb{SO}(3)$. We use the geodesic distance $d(q_1, q_2) = 2 \arccos(|q_1 \cdot q_2|)$, where $q_1 \cdot q_2$ is the ordinary dot-product of the unit quaternions and not the quaternion product. Finally, we convert the rotations from unit quaternions to rotation matrices. The discrete set of rotations Ω_R now fulfills our first requirement of being evenly spread (fig. 3).

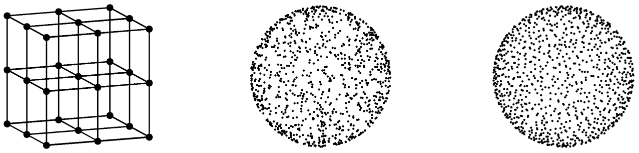

Figure 3. The discrete set of translations Ω_T is generated as a grid covering a bounding cube. To the left we show an example with $|\Omega_T| = 3^3$. The discrete set of rotations Ω_R is generated by sampling unit quaternions, i.e. points on a hypersphere. In the middle we show $|\Omega_R| = 10^3$ samples from a uniform distribution and to the right we show the same number of best-candidate samples.

Ideally, we would like the composition of two rotations in Ω_R to be also in Ω_R. This will, in general, not be the case. We would then like to use the closest grid point. How can we know which grid point that is? A simple solution is to precompute a table with this information. If we have $|\Omega_R|$ rotation states we precompute a $|\Omega_R| \times |\Omega_R|$ table where the element with indices i and j is the index to the rotation in Ω_R that is closest to the composition of the rotations with indices i and j. This matrix can potentially be precomputed in $O(|\Omega_R|^3)$ time, by comparing the distances to all grid points. In section 3 we explore the tractable number of grid points.

2.3. Avoiding Self-Intersections

Using a tree graph and the max-product algorithm to find the solution, is a double-edged sword. On the one hand, it allows us to find the global optimum in a tractable way. But on the other hand, assuming the dependencies between the variables in the model form a tree has its limitations. A typical problem is the double counting of image evidence. If some parts have a similar appearance, typically e.g. the left and right arms and legs, the optimal score often has them placed at the same position.

2D Pose Estimation This problem is especially prevalent in 2D where there is an inherent ambiguity, since two parts may very well project to the same image area even if they do not occupy the same volume in 3D. To reason in this case one needs to recognize whether the two parts really occlude each other or not. Researchers have addressed this problem, but frequently it involves dropping the tree assumption and using a global objective function which couples all parts [2, 14, 1]. The optimization then becomes difficult and the solution found may not be the global optimum.

3D Pose Estimation In 3D we do not have the same ambiguity as in 2D. Whereas the parts should be allowed to intersect in 2D, they should never be allowed to intersect in 3D. However, preventing all parts from intersecting each other would still require a full graph instead of a tree. Instead, we propose a two-step algorithm that prevents a subset of the parts from intersecting. First we find the global optimum of the original objective function, which does not take intersections into account. To deal with the possible intersection of e.g. the legs, we then consider the hypotheses:

1. Left leg has been estimated correctly.
2. Right leg has been estimated correctly.

We then evaluate each of these hypotheses in turn by running the algorithm a second time. In this second stage we fix the part which is assumed to be correctly estimated. The corresponding mirror part is then prevented to intersect the fixed part. This can be done by modifying the appearance scores $P_{I_n|X_n}$. A part can be fixed by setting all states, except the fixed one, to have a zero probability. Similarly, we can prevent a part from intersecting its fixed mirror part by zeroing out all states where this happen. To allow simple and fast intersection tests we model the parts as capsules, i.e. cylinders with spherical ends.

We then find the global optimum to this modified cost function using the max-product algorithm over the same tree graph. This gives a new pose whose parts do not intersect and its associated score. We then choose the hypothesis with the highest score (fig. 5).

3. Experiments

To test our algorithm in a realistic scenario, we recorded a sequence from a professional football game using three cameras, each having a resolution of 1920 × 1080 pixels and a frame-rate of 25Hz. The cameramen followed the same player as he moved around the pitch. We annotated the 2D pose in each view for 214 consecutive frames. Using these 2D measurements the cameras were synchronized and calibrated and the pose was reconstructed in 3D, using affine factorization [4]. We use these 2D measurements and 3D reconstruction as the ground truth to evaluate our algorithm. Our primary questions are:

- Are pictorial structures in 3D a practical solution?
- What is the necessary level of discretization needed to represent human poses in 3D?
- Is this discretization level computationally tractable?

To answer these questions we first investigate what levels of discretizations are tractable in terms of memory consumption. We then consider the computation time for these discretizations. Finally, we evaluate if these discretizations can represent poses with the desired accuracy. These experiments are discussed in 3.1.

Our next set of experiments focus on applying the algorithm to measurements extracted automatically from each view, using 2D part detectors. These experiments are discussed in 3.2.

$\mid \mathbf{\Omega_R} \mid$ $\mid \mathbf{\Omega_T} \mid$	$\mathbf{4^3}$	$\mathbf{8^3}$	$\mathbf{16^3}$
$\mathbf{16^3}$	10 MB	84 MB	670 MB
$\mathbf{32^3}$	84 MB	670 MB	5.4 GB
$\mathbf{64^3}$	670 MB	5.4 GB	43 GB

Table 1. Memory consumption for different discretizations.

	View & Skeleton Constraints			**View, Skeleton & Joint Angle Constraints**		
$\mid \mathbf{\Omega_R} \mid$ $\mid \mathbf{\Omega_T} \mid$	$\mathbf{4^3}$	$\mathbf{8^3}$	$\mathbf{16^3}$	$\mathbf{4^3}$	$\mathbf{8^3}$	$\mathbf{16^3}$
$\mathbf{16^3}$	0.021 s	0.14 s	1.0 s	0.041 s	2.4 s	5.5 min
$\mathbf{32^3}$	0.14 s	1.0 s	8.4 s	0.42 s	23 s	69 min
$\mathbf{64^3}$	1.1 s	8.7 s		5.1 s	4.9 min	

Table 2. Computation time for different discretizations.

3.1. Tractability

A key factor that affects the tractability of all the considered max-product algorithms (1, 2, 3) is the memory used to store all scores/messages, i.e. the m-array. It has $N \times |\Omega_T| \times |\Omega_R|$ elements. In our implementation $N = 10$ and 4 bytes are used for each element. In table 1 we list the memory requirements for this array for different translation Ω_T and rotation Ω_R discretizations. All discretizations listed in the table, except the bottom right corner, fit into the 16 GB RAM of our test system.

We next look at the computation time for running algorithm 2 and 3 for different discretizations. The algorithms were implemented in C++ using OpenMP to parallelize the for-loops over Ω_T. The computations were run on an Intel Core2 Quad processor with four 2.8 GHz cores. The result is summarized in table 2. Algorithm 2 imposes view and skeleton constraints. Its time complexity of $O(|\Omega_T||\Omega_R|)$ is confirmed by the table. Algorithm 3 additionally imposes joint angle constraints. Its time complexity of $O(|\Omega_T||\Omega_R|^2)$ is approximately matched by the table.

Finally, we explore what level of discretization that is necessary to obtain an acceptable estimate of the 3D pose. To perform this evaluation we use synthetically generated scores for $P_{I_n^c|X_n}$. This avoids conflating inaccuracies in the measurement process with the coarseness of the grid discretization, when analyzing the cause of errors in the final 3D pose estimate. The synthetic scores are computed from 2D pose annotations. Each part is modeled as a line segment. Let the annotated start and end points of part n in view c be denoted by $\hat{s}(i_n^c)$ and $\hat{e}(i_n^c)$. If the part is in state x_n the projected start and end points are denoted $s_n^c(x_n)$ and $e_n^c(x_n)$. Our synthetic appearance score is then the difference between the projected and annotated end points:

$$\ln P_{I_n^c|X_n}(i_n^c \mid x_n) = -\left\|s_n^c(x_n) - \hat{s}(i_n^c)\right\|^2 + \\ -\left\|e_n^c(x_n) - \hat{e}(i_n^c)\right\|^2 \qquad (18)$$

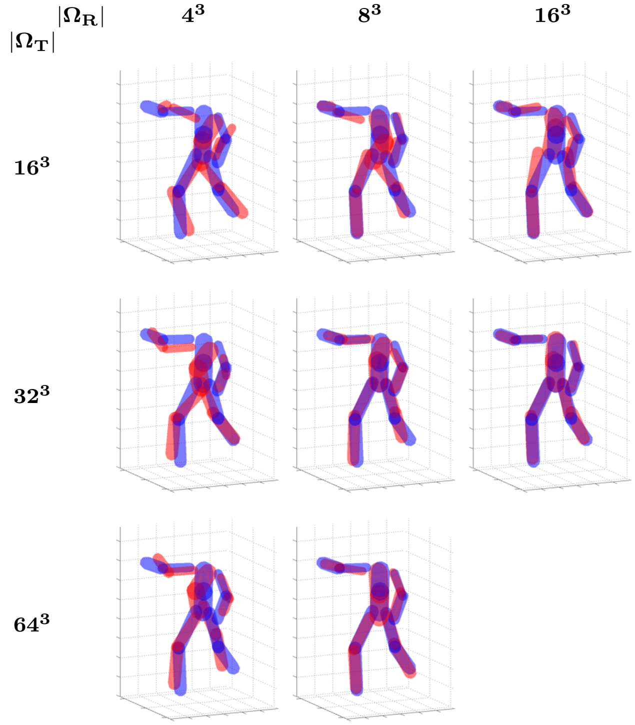

Figure 4. Evaluation of the necessary detail required for the discretization grid. Synthetic appearance scores are used. The estimated 3D pose (red) is the pose closest to the ground truth pose (blue), that is possible to represent with the given discretization.

Figure 4 shows the result of running algorithm 2 with these synthetic scores, for different levels of discretizations. We conclude that having $|\Omega_T| \geq 32^3$ and $|\Omega_R| \geq 8^3$ gives enough detail. Since this is tractable both in terms of memory and speed we conclude that algorithm 2 is practical and tractable.

We have observed that algorithm 3 seems to require a finer discretization, $|\Omega_R| \geq 16^3$. This is on the border of being tractable in terms of speed of our current implementation. We believe this extra level of detail is needed since the hard joint angle constraints remove some of the local rotations of each part. More specifically, it removes some of the rotations that approximately rotate the part around its own axis, but result in slightly different end positions. This loss of precision needs to be compensated by having more global rotations.

3.2. Automatic Part Detection

These experiments test automatic pose estimation using algorithm 2. To do this we implemented simple 2D part detectors based on the HOG-descriptor [5]. We model each part as a cylinder and approximate its projection to an image as a rectangle. Each 3D rotation then corresponds to a 2D rotation and change of aspect ratio of this rectangle. To be invariant to this effect, we warp the rectangle to a canon-

	View & Skeleton Constraints						View, Skeleton & Intersection Constraints					
Parts	**C=1**		**C=2**		**C=3**		**C=1**		**C=2**		**C=3**	
Pelvis	97	57	97	35	100	50	97	57	97	35	100	55
Torso	87	40	90	48	100	65	87	38	90	48	100	55
Upper Arms	14	2	55	8	55	15	14	2	53	8	60	20
Lower Arms	6	0	30	6	35	18	6	0	28	7	35	15
Upper Legs	62	8	87	26	90	45	63	9	88	19	100	48
Lower Legs	33	5	68	35	70	57	41	7	82	38	90	60
All Parts	41	13	67	23	70	39	43	13	69	23	77	40

Table 3. A quantitative summary of the results of our pose estimation to real images from 20 different frames. PCP scores in % with $\alpha = 0.5$ and $\alpha = 0.2$ (in blue) are used to measure performance of pose estimation using 1, 2 or 3 cameras. We first only impose view and skeleton constraints. We then add intersection constraints for the lower legs.

ical square. We let the HOG of this square represent the appearance of the part.

Using 2D pose annotations we train a binary logistic regression classifier [3], to allow a probabilistic interpretation, for each part. We use 100 frames from the 3 camera views for training. When testing on an image we evaluate the detector for each 2D position and each 2D rotation and aspect ratio of the rectangle.

After this has been done for all views we use these response scores in look-up tables when evaluating the score for each 3D position and 3D rotation. Each 3D position and rotation of the part corresponds to a 2D position, rotation and aspect ratio of the rectangle in each view. The scores from the different views are aggregated using equation 3.

The quantitative results in this section are reported in terms of PCP scores: percentage of correctly estimated parts. A part is declared correctly estimated if:

$$\frac{\|\hat{s}_n - s_n\| + \|\hat{e}_n - e_n\|}{2} \leq \alpha \|\hat{s}_n - \hat{e}_n\| \qquad (19)$$

where $\hat{s}_n$ and $\hat{e}_n$ represent the ground truth 3D coordinates of the start and end point of part n and s_n and e_n the algorithm's estimate. We report scores for $\alpha = 0.2$ and $\alpha = 0.5$ in table 3. The PCP score is more informative than one based on the Euclidean distance, given the the difficulty of the data set and the precision of our simple 2D part detectors. We test with and without the 3D intersection constraints and using 1, 2 or 3 camera views.

Table 3 and figure 5 show that our simple 2D part detectors are not very accurate. However, designing accurate 2D part detectors has not been our focus. The frame-work supports any such detector. More importantly, the table and figure show that given a 2D part detector, the 3D pictorial structures frame-work can improve the accuracy of the estimation by imposing view, skeleton and intersection constraints in 3D.

4. Conclusions and Future Work

We have described and implemented a frame-work for 3D pictorial structures that can be used for multiple view articulated pose estimation. Thanks to the discretization of the search space a globally optimal pose can be computed. We implemented two algorithms. The first algorithm (2) imposes view and skeleton constraints. The second algorithm (3) also imposes joint angle constraints. We have shown that the first algorithm is tractable, whereas our implementation of the second algorithm is on the border of being tractable in terms of speed, on our test system. We also demonstrated how the problem of intersecting parts, common for tree-based models, can be dealt with more easily in 3D than 2D.

We see several interesting directions for future research. Finding an efficient way of computing max-convolutions over discrete subsets of $\mathbb{SO}(3)$ would speed up the second algorithm, imposing joint angle constraints. A coarse-to-fine or branch and bound approach could also help to reduce the search in general. One could also utilize the parallel nature of the max-product algorithm by exploring GPU implementations.

In our implementation we compute the image evidence of the individual parts using 2D part detectors that are rather basic and not that accurate. Better performance can be expected if this frame-work independent component is instead based on a state-of-the-art 2D pose estimator. Now that the tractability of the frame-work has been shown, we plan to refine this appearance component and thouroughly compare the performance with alternative 3D pose estimators. Another interesting direction for future research is how to automatically calibrate the cameras.

Acknowledgement This work was supported by the FP7 project ”Free-viewpoint Immersive Networked Experience”. The authors would like to thank AIK Football Club and Hego Tracab for help with collecting the football footage.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Discriminative appearance models for pictorial structures. *International Journal of Computer Vision*, 99(3):259–280, 2012.

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

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[4] M. Burenius, J. Sullivan, and S. Carlsson. Motion capture from dynamic orthographic cameras. In *4DMOD - ICCV Workshop*, 2011.

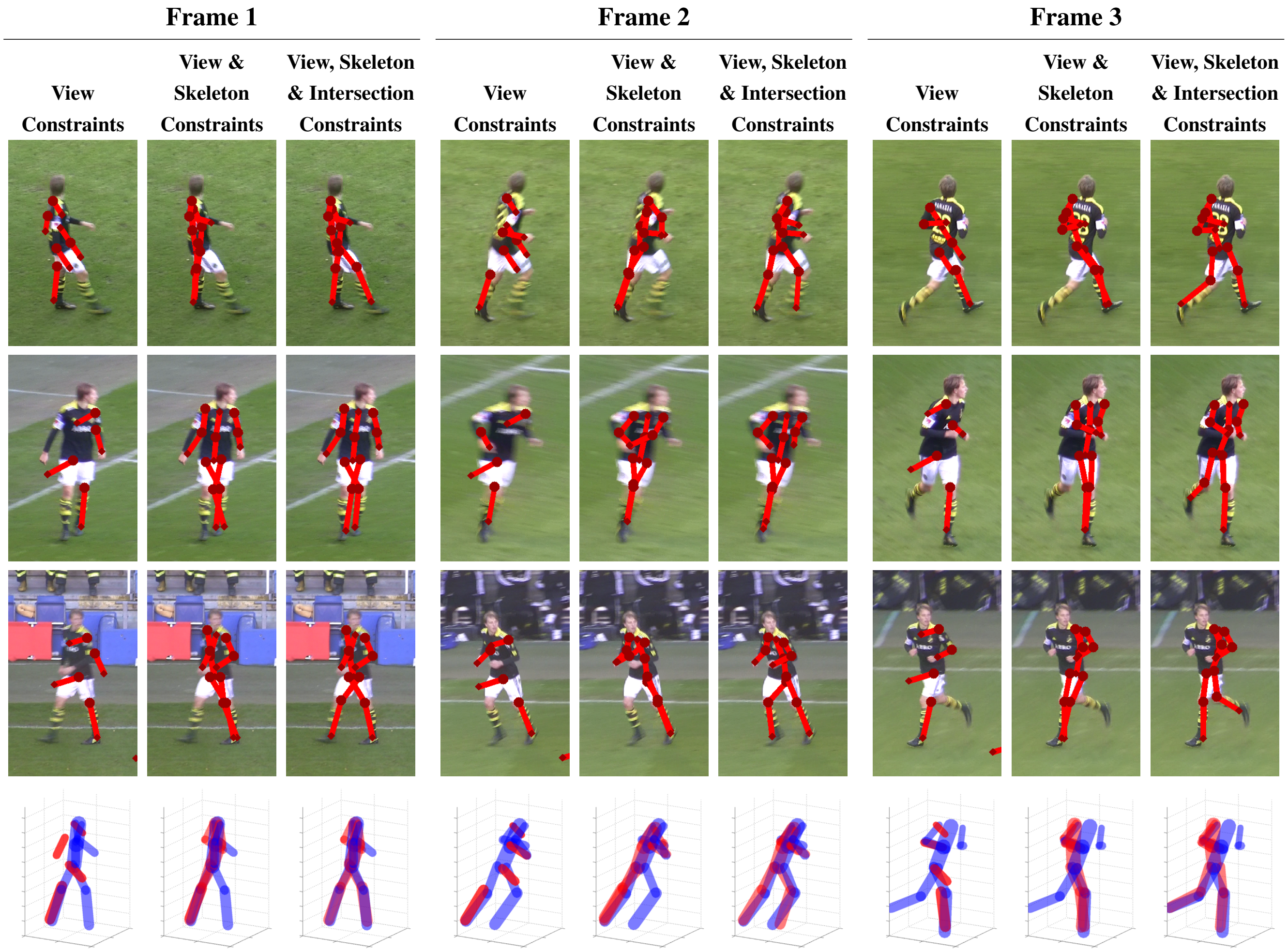
Frame 1
Frame 2
Frame 3
View Constraints
View & Skeleton Constraints
View, Skeleton & Intersection Constraints
View Constraints
View & Skeleton Constraints
View, Skeleton & Intersection Constraints
View Constraints
View & Skeleton Constraints
View, Skeleton & Intersection Constraints

Figure 5. Multiple view 3D pose estimation imposing different types of constraints. In the first column only view constraints are imposed. The second column adds skeleton constraints. The third column also adds intersection constraints. The rows show the different camera views and the bottom row shows the reconstruction from a new view. The reconstruction is drawn in red and the ground truth in blue.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2005.

[6] E. B. Dam, M. Koch, and M. Lillholm. Quaternions, interpolation and animation. Technical report, 1998.

[7] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627 –1645, sept. 2010.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, Jan. 2005.

[9] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, C-22(1):67 – 92, jan. 1973.

[10] F. S. Grassia. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3):29–48, 1998.

[11] T. Moeslund, A. Hilton, V. Krüger, and L. Sigal. *Visual Analysis of Humans: Looking at People*. Springer, 2011.

[12] B. Pepik, P. Gehler, M. Stark, and B. Schiele. 3d2pm - 3d deformable part models. In *Proceedings of the European Conference on Computer Vision*, 2012.

[13] M. Pharr and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2010.

[14] L. Sigal and M. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2006.

[15] L. Sigal, M. Isard, H. Haussecker, and M. Black. Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision*, 98(1):15–48, 2012.

[16] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.