Tracking Human Pose by Tracking Symmetric Parts

Varun Ramakrishna, Takeo Kanade, Yaser Sheikh
Robotics Institute, Carnegie Mellon University
{vramakri, tk, yaser}@cs.cmu.edu

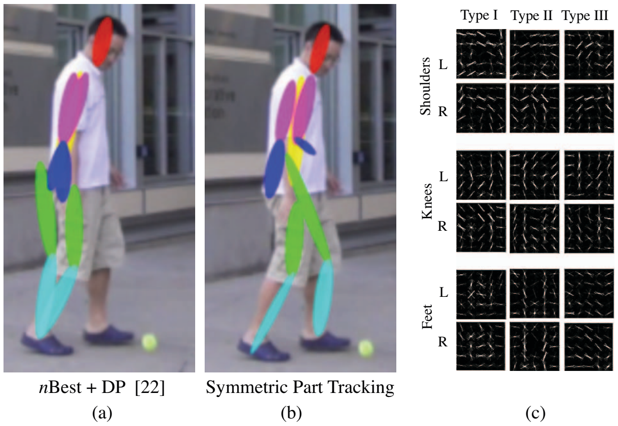
Abstract

The human body is structurally symmetric. Tracking by detection approaches for human pose suffer from double counting*, where the same image evidence is used to explain two separate but symmetric parts, such as the left and right feet. Double counting, if left unaddressed can critically affect subsequent processes, such as action recognition, affordance estimation, and pose reconstruction. In this work, we present an occlusion aware algorithm for tracking human pose in an image sequence, that addresses the problem of double counting. Our key insight is that tracking human pose can be cast as a multi-target tracking problem where the "targets" are related by an underlying articulated structure. The human body is modeled as a combination of singleton parts (such as the head and neck) and symmetric pairs of parts (such as the shoulders, knees, and feet). Symmetric body parts are jointly tracked with mutual exclusion constraints to prevent double counting by reasoning about occlusion. We evaluate our algorithm on an outdoor dataset with natural background clutter, a standard indoor dataset (*HumanEva-I*), and compare against a state of the art pose estimation algorithm.*

1. Introduction

As far back as Gibson [11, 12], researchers have noted the importance of having a representation for occlusion to reason about motion. Representing occlusion is particularly important in estimating human motion because, as the human body is an articulated structure, different parts occlude each other frequently. The human body is structurally symmetric and parts tend to be occluded by their symmetric counterparts, such as left knees by right knees (Figure 1). This occurs because the viewer's optical axis is often perpendicular to the body's bilateral plane of symmetry.

During occlusions, the appearance symmetry of the human body can cause *double counting*: the same image evidence is used to explain the location of both symmetric parts. If left unaddressed, double counting can critically affect subsequent processes, such as action recognition [37], affordance estimation [9], and pose reconstruction [24]. In action recognition and affordance estimation, these errors can substantially change the semantic interpretation of an action or scene. Double counting occurs when *symmetric part pairs* have high detection scores at the same locations in the image (Figure 2). This happens in two cases: (1) when image cues for one part of a symmetric pair dominate the other, and (2) in occlusion scenarios, in which the image only contains evidence for one part, such as profile views of a person. Thus, dealing with double counting requires a representation for occlusion, as well as relationships between symmetric parts that enforce mutual exclusion.



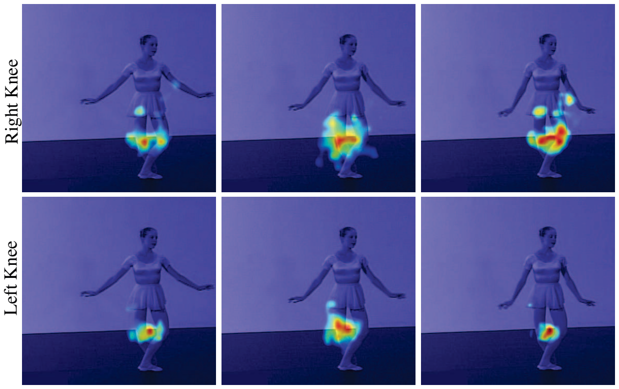
*n*Best + DP [22]
(a)

Symmetric Part Tracking
(b)

(c)

Figure 1: Symmetric Parts. Symmetric parts tend to cause *double counting* errors (a) in tree-structured models because they have similar appearance models as shown for a set of parts in (c). Our method reasons about occlusions and tracks symmetric parts jointly, and thereby reduces double counting errors shown in (b).

Spatial representations for reasoning about occlusion require evaluating a large set of possible spatial configurations [31], which scales combinatorially as we move from images to videos. Spatial representations also rely on weak cues; for example, the location and appearance of a shoulder provides only a weak cue as to whether the elbow is occluded. Temporal representations can make use of strong temporal continuity priors to reason about occlusions. It has been noted that even in the human visual system [28], temporal motion continuity serves occlusion reasoning. A part that is visible and has a smooth trajectory before and after a period of non-visibility must be occluded for that period. If



(a) Frame 10 (b) Frame 15 (c) Frame 20

Figure 2: Double Counting. The max marginals for symmetric parts (left and right knees) score highly on the same locations in the image because of the similar appearance of symmetric parts. We show three frames of a *ballet* sequence with the max-marginals of the left and right knees overlaid on the top and bottom rows respectively.

a system cannot reason about occlusion temporally, motion consistency will force it to struggle to find image evidence to support a smooth path when occlusion occurs. This can corrupt tracking even outside the duration of occlusion.

In this work, we argue that temporal reasoning about occlusion is essential to tracking human pose and handling double counting. We divide the body into a set of singleton parts and pairs of symmetric parts. Our key insight is that tracking human pose can be cast as a multi-target tracking problem where the ”targets” are related by an underlying articulated structure. Our contributions are: (1) an occlusion-aware model for tracking human pose that enforces both spatial and temporal consistency; (2) a method for jointly tracking symmetric parts that is inspired by optimal formulations for multi-target tracking. We evaluate our method on an outdoor pose dataset and report results on two standard datasets. We outperform a state-of-the-art baseline [23] and demonstrate a marked reduction in double counting errors.

2. Relevant Work

There exists a large body of work that tackles the problem of human pose estimation. Early methods [29, 22] used model-based representations to track human pose in video sequences, however these methods usually required good initializations and strong dynamic priors such as [18]. These methods usually require knowledge of the action being performed a priori, although some methods [2] exist which attempt to estimate the dynamical model online.

There is also a large body of work that looks at the problem of directly estimating 3D human pose from video sequences. These methods, while attractive for reasoning about occlusion in 3D, tend to require strong priors due to the larger set of possible configurations in 3D and do not generalize to arbitrary actions easily. We refer the interested reader to [32, 10, 20] for a survey of methods in this area.

There has been a recent thrust in methods that aim to detect people in a single image. Pictorial structure models [4, 3, 36, 25, 16], model the human body as a tree-structured graphical model with kinematic priors that couple connected limbs. These methods have typically been successful on images where all the limbs of the person are visible. However, they struggle on images where the subject is undergoing self-occlusion and suffer from double counting of image evidence.

Fully connected models have been employed by [35, 31, 19, 17] to estimate pose in a single image. These models augment the tree-structure to capture occlusion relationships between parts not connected in the tree. These models often require loopy belief-propagation for inference. Recently [33] used branch and bound to perform exact inference on a loopy graph, however these models do not have a representation for occlusion. For the single image case, the work by Jiang [15, 13] enforces exclusion constraints by decoding trellis graphs for each part with constraints between the graphs to enforce mutual exclusion. Our method can be viewed as the temporal dual of this approach and makes use of strong temporal continuity cues to exclude inconsistent configurations.

In the video domain, [26] found a frame with an easily detectable canonical pose to build up an appearance model of the person that can be used to aid tracking in the rest of the frames. While this method has been effective, finding a detectable canonical pose is usually difficult in short videos and in videos of non-standard actions. Sapp et al. [30] decomposed a full model for video into a set of tree-structured subgraphs, on which inference is performed separately and agreement is enforced between the solutions. Park et al. [23] generated multiple diverse high-scoring pose proposals from a tree-structured model and used a chain CRF to track the pose through the sequence. Recent approaches have also looked to track extremities of multiple interacting people using a branch and bound framework on AND-OR graphs [21] and quadratic binary programming [34].

We cast the problem of tracking human pose as a multi-target tracking problem where the ”targets” are related by an articulated skeleton. Our formulation for the simultaneous tracking of symmetric parts draws inspiration from recent advances in the area of multiple target tracking. Several linear programming formulations [14, 7, 5] have been proposed that allow a variable number of objects to be tracked in a globally optimal fashion. We choose to adopt an LP formulation as it allows for us to easily incorporate constraints that are specific to our problem.

3. Tracking Human Pose

The (u, v) location of a part p in a frame at time instant f is denoted by x_p^f. We denote by $\mathbf{x}_p = [x_p^1 \ \dots \ x_p^F]$, the locations of part p in frames 1 to F and by $\mathbf{x}$ the set of

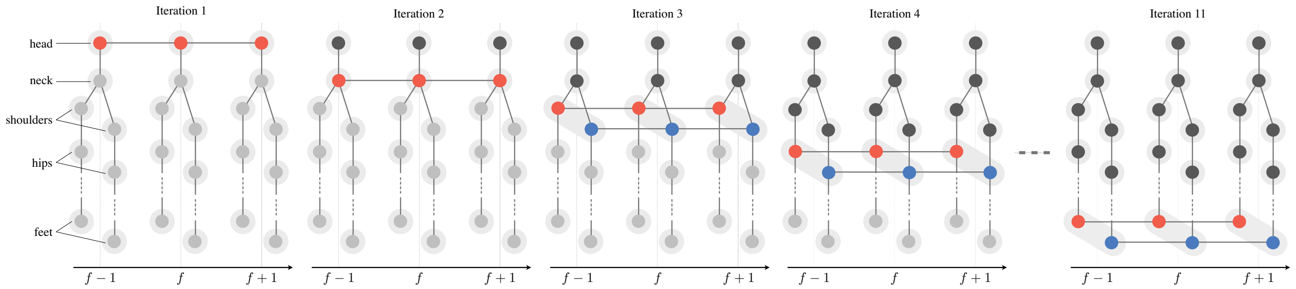


Figure 3: Graphical representation of the algorithm. We use a tree-structured deformable parts model in each frame to generate proposals for each part. In the first iteration, we track the head node using an LP tracking formulation. Proposals for the next symmetric pair in the tree are generated by conditioning each tree on the tracked locations computed in the previous iteration. Symmetric parts are tracked simultaneously with mutual exclusion constraints. The method proceeds by sequentially conditioning the tracking of parts on their parents until all the parts are tracked.

Algorithm 1 Tracking Human Pose by Tracking Symmetric Parts

```
Compute max-marginals and generate detections for root part (head).
Track root part.
while In breadth first fashion, select next part(s) do
    Compute max-marginals for current part(s) conditioned on the tracked locations of parent parts.
    if is_symmetric(part) then
        Track symmetric parts using LP multi-target tracking.
    else
        Track part using LP tracking.
    end if
end while
```

tracks for all parts $(1, \dots, P)$. A symmetric part pair is a pair of parts (p, q) that share the same appearance. The goal of human pose tracking is to estimate the location of each part of the person in every frame of the image sequence. We write this as maximizing the following scoring function over the full model:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}}\ \mathbf{E}(\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_P). \tag{1}$$

Optimizing the above scoring function over the full model requires a search over an exponential number of configurations and is NP-hard in general.

To bypass the intractability of the objective, we proceed by approximating the function and making stage-wise locally optimal decisions (see Figure 3). We begin with a root node for which the false positive rate is the lowest [36]. For human pose, this root node is the head for which we are able to get reliable detections. Given a set of proposals for the location of the head in each frame (Section 3.4), we solve for the optimal track $\mathbf{x}_1^*$,

$$\mathbf{x}_1^* = \underset{\mathbf{x}_1}{\operatorname{argmax}}\ \mathbf{E}(\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_P). \tag{2}$$

3.1. Tracking a Singleton Part

Given a set of proposals denoted by $\mathcal{X}_p^f$ for part p in the image at each frame f, we first augment the proposal sets with an occlusion state o_p^f for each frame. We form tracklets ${}^p t_{ijk}$ for each part by combining triplets $({}^i x_p^{f-1}, {}^j x_p^f, {}^k x_p^{f+1})$ where ${}^i x_p^f \in \mathcal{X}_p^f$ is a proposal at location i in the image or an occlusion state o_p^f.

We denote by ${}^p\mathbf{X}_{ijk}^f$ the indicator variable that is associated with tracklet ${}^p t_{ijk}$ that takes values $\in \{0, 1\}$ corresponding to the tracklet being selected or not. We associate with each tracklet, a score u_{ijk}^f based on appearance, detection, and foreground likelihood cues, which is described in Section 3.5. Our goal then is to maximize the following objective subject to constraints:

$$\begin{aligned}
&\max_{\{{}^p\mathbf{X}\}} && \sum_{\forall i,j,k,f} {}^p u_{ijk}^f \, {}^p\mathbf{X}_{ijk}^f \\
&\text{s.t.} && \{\mathbf{X}_{ijk}^f\} \in \{0,1\} \\
& && \forall f, \forall (j,k) \ \sum_i {}^p\mathbf{X}_{ijk}^f = \sum_l {}^p\mathbf{X}_{jkl}^{f+1} \qquad (\textit{Continuity}) \\
& && \forall f, \ \sum_{i,j,k} {}^p\mathbf{X}_{ijk}^f = 1 \qquad (\textit{Uniqueness})
\end{aligned} \tag{3}$$

The above optimization problem corresonds to finding the single best path in a lattice graph and can be solved efficiently using dynamic programming.

Continuity Constraints enforce conservation of flow by stating that the flow entering the nodes j and k should be equal to the flow emanating from those nodes. These constraints essentially encode the connectivity of a track, preventing fragmented tracks.

Uniqueness Constraints limit the flow at each time instant to be 1. This implies that one object is being tracked in the network graph.

3.2. Conditioned Tracking

Once the optimal track $\mathbf{x}_1^*$ has been obtained (Section 3.1), we generate proposals and track the next set of nodes

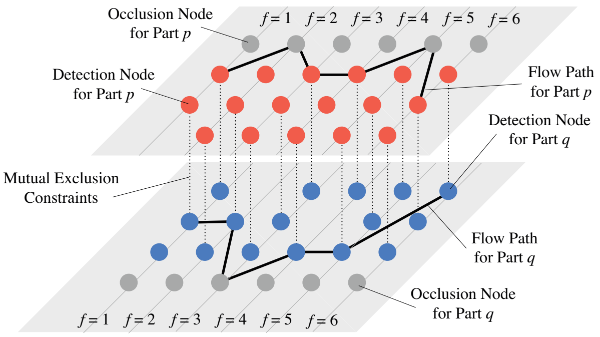

Occlusion Node for Part p
$f=1$
$f=2$
$f=3$
$f=4$
$f=5$
$f=6$
Detection Node for Part p
Flow Path for Part p
Detection Node for Part q
Mutual Exclusion Constraints
Flow Path for Part q
Occlusion Node for Part q
$f=1$
$f=2$
$f=3$
$f=4$
$f=5$
$f=6$

Figure 4: Max-flow formulation for symmetric part tracking. The blue and red dots denote detections for each of the parts separately in each frame. The gray nodes denote occlusion nodes for each frame. The dotted lines depict mutual exclusion constraints between certain sets of nodes. The symmetric tracking problem is to find the best scoring path in each of these graphs subject to the mutual-exclusion constraints.

conditioned on the optimal parent track $\mathbf{x}_1^*$.

$$(\mathbf{x}_2^*) = \underset{\mathbf{x}_2}{\operatorname{argmax}}\, \mathbf{E}(\mathbf{x}_1 = \mathbf{x}_1^*, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4 \ldots \mathbf{x}_P). \tag{4}$$

We use the same formulation as in Section 3.1 to obtain the optimal track $\mathbf{x}_2^*$.

Next, for a symmetric pair of parts whose tracks are given by $(\mathbf{x}_3, \mathbf{x}_4)$ we simultaneously estimate the optimal tracks (See Section 3.3):

$$(\mathbf{x}_3^*, \mathbf{x}_4^*) = \underset{\mathbf{x}_3,\mathbf{x}_4}{\operatorname{argmax}}\, \mathbf{E}(\mathbf{x}_1 = \mathbf{x}_1^*, \mathbf{x}_2 = \mathbf{x}_2^*, \mathbf{x}_3, \ldots \mathbf{x}_P). \tag{5}$$

Tracking is conditioned on the optimal parent track by fixing the location of the parent in each of the frames to the tracked locations and re-running dynamic programming inference in each of the trees in each frame (Section 3.4).

We proceed in this manner, by conditioning the tracking of the child nodes on the optimal tracks of their parents and by tracking symmetric parts using a joint formulation, until all the parts have been tracked.

3.3. Tracking a Pair of Symmetric Parts

Our approach treats the problem of tracking symmetric pairs of parts as a multi-target tracking problem. In multi-target tracking, the goal is to track multiple objects that share the same appearance and hence the same generic detector (typically pedestrians). The objects move in the scene in an unconstrained fashion with mutual occlusions. Recent methods have modeled multi-target tracking as a network flow problem [5, 14, 7] where finding tracks is equivalent to pushing K-units of flow through a graph where K is the number of objects to be tracked.

Our formulation is as follows: we denote by ${}^p\mathbf{X}$ and ${}^q\mathbf{X}$ the set of all indicator variables for tracklets p and q respectively. Our full objective is now the following optimization problem:

$$\begin{aligned}
\max_{\{{}^p\mathbf{X},{}^q\mathbf{X}\}} \quad & \sum_{i,j,k,f} {}^pu^f_{ijk}\,{}^p\mathbf{X}^f_{ijk} + \sum_{i,j,k,f} {}^qu^f_{ijk}\,{}^q\mathbf{X}^f_{ijk} \\
\text{s.t.} \quad & \{\mathbf{X}^f_{ijk}\} \in \{0,1\} \\
& \forall f,\ \sum_i {}^p\mathbf{X}^f_{ijk} = \sum_l {}^p\mathbf{X}^{f+1}_{jkl} \qquad (\textit{Continuity}) \\
& \forall f,\ \sum_i {}^q\mathbf{X}^f_{ijk} = \sum_l {}^q\mathbf{X}^{f+1}_{jkl} \\
& \sum_{i,k} {}^p\mathbf{X}^f_{ijk} + \sum_{i,k} {}^q\mathbf{X}^f_{ijk} \leq 1 \qquad (\textit{Mutual Exclusion}) \\
& \forall f,\ \sum_{i,j,k} {}^p\mathbf{X}^f_{ijk} = 1 \qquad (\textit{Uniqueness}) \\
& \forall f,\ \sum_{i,j,k} {}^q\mathbf{X}^f_{ijk} = 1
\end{aligned} \tag{6}$$

Mutual Exclusion Constraints. We enforce mutual exclusion constraints that prevent the symmetric parts from occupying the same location in the image. In a typical self-occlusion scenario the score of a particular location in the image will be high for both the symmetric parts. In such a case the mutual-exclusion constraints enforce that only one part can occupy the location, while the symmetric counterpart is either pushed to an occlusion node or to another location in the image that is consistent with the constraints and has a high score. We enforce these constraints by limiting the total flow at nodes in both networks that share the same location in the image.

This formulation corresponds to maximizing the flow through two separate networks that interact via the mutual exclusion constraints. The above optimization problem is an integer linear program and solving it is NP-complete. However, we can relax the problem by replacing the integral constraints by allowing $0 \leq {}^p\mathbf{X}^f_{ijk} \leq 1$ and $0 \leq {}^q\mathbf{X}^f_{ijk} \leq 1$. The relaxation can be shown to be tight for most practical cases [5].

We solve this linear program using a commercially available solver [1]. In the case of non-integral solutions, we use a branch and cut method to find the integral optimum as suggested in [5].

Occlusion Interpolation Once a solution is obtained, the location of the occluded part is estimated by interpolating between the image location of the node preceding and following occlusion using cubic B-spline interpolation.

3.4. Generating Part Proposals via Max-Marginals

Human pose in a frame at each time instant is modeled with a tree-strutured deformable part model as in recent work by [36]. A deformable part model is a tree-structured CRF that maximizes the following score, given an image:

$$\mathbf{S}(\mathbf{x}_f) = \sum_{i=1} w_i^F \phi(I_t, x_i^f) + \sum_{i,j} w_{ij} \psi(x_i^f, x_j^f) \tag{7}$$

where $\mathbf{x}_t = [x_1^f \ldots x_P^f]$ is the pose in frame f, $\phi(I_f, x_i^f)$ are a set of image features computed at location x_i^f, $\psi(x_i^f, x_j^f)$

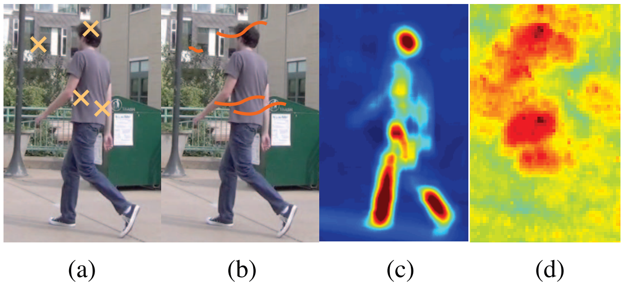


Figure 5: Scoring Tracklets. (a) Proposals for the head are generated from the max-marginal score map shown in (d). (b) Proposal sets are augmented by tracking each proposal forwards and backwards to ensure smooth tracks. (c) Foreground likelihood used to score tracklets (d) The detection likelihood for the head part.

is a quadratic function that measures the displacement between parts i and j. The weights w_i and w_{ij} are the parameters of the CRF that are learned as described in [36].

To generate proposals for part locations in each frame, we compute the max-marginal of the above scoring function at each part. The max-marginal for part i in frame f is given by:

$$\mu^*(x_i^t = s) = \max_{x^t : x_i^t = s} \mathbf{S}(\mathbf{x}_t), \tag{8}$$

which is the maximum of the scoring function with the part i clamped to location s. The max-marginal provides a peaky approximation of the true marginal distribution. We compute max-marginals for each tree in each frame separately. The max-marginals for a tree-structured graphical model can be computed efficiently for all the parts by performing two passes of max-sum message passing inference. We perform non-maxima suppression on the max-marginal score map for each part to generate a set of location proposals in each frame.

We expand the proposal set by tracking each proposal forwards and backwards using a Lucas-Kanade template tracker [6] to obtain extended proposal sets $\mathcal{X}_i^t$. This ensures smoother tracks and makes the proposal generation robust to frame-to-frame inconsistencies of the detector.

Once a parent part has been tracked, the max-marginals for the child nodes are recomputed by conditioning on the tracked locations of the parent nodes. The conditioned max-marginals for part i in frame f with a set of parent nodes $pa(i)$ with tracked locations $\mathbf{x}^*_{pa(i)}$ can be written as:

$$\mu^*(x_i^f = s) = \max_{\substack{x^f : x_i^f = s, \\ \forall j \in pa(i),\ x_j^f = x_j^{f*}}} \mathbf{S}(\mathbf{x}_f). \tag{9}$$

This can be efficiently computed for a tree, as before, by performing dynamic programming max-sum inference.

3.5. Scoring Part Tracklets

Each tracklet is assigned a likelihood score that consists of terms that measure the detection likelihood, the foreground likelihood and motion prior:

$$\begin{aligned} u_{ijk}^f = \ & \alpha_s \mathbf{s}_{\text{fore}}(\mathbf{X}_{ijk}^f) + \alpha_f \mathbf{s}_{\text{det}}(\mathbf{X}_{ijk}^f) \\ & + \alpha_m \mathbf{s}_{\text{mot}}(\mathbf{X}_{ijk}^f). \end{aligned} \tag{10}$$

The weighting co-efficients of the different terms were set by performing a grid search on validation data.

Detection Likelihood. The likelihood of detection for a particular part is obtained by using the max-marginal score of the tree-structured CRF model. We normalize the max-marginal score and obtain a likelihood of detection of part p at location i as:

$$\mathbf{l}_{det}({}^i x_p^f) \propto \frac{\exp(-\mu^*(x_p^f = i))}{\sum_{s=1}^{L} \exp(-\mu^*(x_p^f = s))}. \tag{11}$$

For a tracklet with occlusion nodes we assign a constant score for the occlusion nod $\mathbf{l}_{det}({}^i o_p^f) \propto p_{det}^o$. This constant needs to be calibrated in relation to the scores of the detector and is found by performing a grid search on validation data. The detection score for the tracklet $\mathbf{X}_{ijk}^f$ is obtained as:

$$\mathbf{s}_{det}(\mathbf{X}_{ijk}^f) = \mathbf{l}_{\text{det}}({}^i x_p^{f-1}) \cdot \mathbf{l}_{\text{det}}({}^j x_p^f) \cdot \mathbf{l}_{\text{det}}({}^k x_p^{f+1}). \tag{12}$$

Motion Likelihood. We use a constant velocity motion model. In order to check for constant velocity, we require two motion vectors, and therefore we use three consecutive sites in our formulation (similar to [5]). We denote the two motion vectors as $\mathbf{v}_{ij} = x_i^{f-1} - x_j^f$ and $\mathbf{v}_{jk} = x_j^f - x_k^{f+1}$. Our motion score is now given by:

$$\mathbf{s}_{mot}(\mathbf{X}_{ijk}^f) = e^{-\left(\frac{\|\mathbf{v}_{ij} - \mathbf{v}_{jk}\|}{\sigma_m}\right)^2}. \tag{13}$$

The constant velocity model allows us to enforce smoother trajectories and penalize large deviations.

Foreground Likelihood. The foreground likelihood is estimated by computing a background model by median filtering the image sequence. The foreground likelihood is estimated as:

$$\begin{aligned} \mathbf{s}_{mot}(\mathbf{X}_{ijk}^f) = (1 - p_b(x_i^{f-1})) \cdot (1 - p_b(x_j^f)) \\ \cdot (1 - p_b(x_k^{f+1})) \end{aligned}$$

where $p_b(x_j^f)$ denotes the probability of the location x_j^f of belonging to the background, as given by:

$$p_b(x) = \frac{1}{\sqrt{2\pi}\sigma_b} e^{-\left(\frac{\|I(x) - I_b(x)\|}{\sigma_b}\right)^2} \tag{14}$$

where I_b is the computed background model. As before, we assign a constant score to occlusion nodes.

4. Evaluation

We perform qualitative and quantitative experiments on two challenging datasets to determine the performance of

Metric	Method	Head	Torso	U.L.	L.L.	U.A.	L.A.
PCP	Ours [23]	0.99 0.99	0.86 0.83	0.95 0.92	0.96 0.86	0.86 0.79	0.52 0.52
KLE	Ours [23]	0.39 0.44	0.58 0.58	0.48 0.55	0.48 0.69	0.88 1.03	1.42 1.65

Table 1: PCP scores and keypoint localization error for the six sequences of the outdoor pose dataset. We obtain a significant improvement over the baseline due to better temporal consistency and occlusion handling.

the proposed algorithm. In order to test the tracking method we model human pose with the state-of-the-art tree-structured CRF model of [36]. For all experiments, we train the model on the PARSE dataset introduced in [27]. We model the human body with 26 parts as in [36]: 2 singleton parts for the head and neck and a total of 12 symmetric pairs of parts for the shoulders, torso, legs, and upper arms.

Comparisons. As our baseline, we compare the method of [23] that also uses a detector for pose in each frame [36] that is trained on the same training data. The *n*-Best pose configurations are generated for each frame and tracking is performed by modeling pose tracking with a chain-CRF and performing viterbi-decoding like inference.

4.1. Datasets.

We test our method on a variety of challenging datasets consisting of both indoor and outdoor sequences.

Human Eva-I: We evaluate our method on a standardized dataset that comprises of sequences of actors performing different actions in a indoor motion capture environment. We report results on the 250 frames each of the sequences *S1_Walking, S1_Jog, S2_Jog* for camera 1. We show qualitative results in Figure 8.

Outdoor Pose Dataset: This dataset consists of 6 sequences collected by us comprising of 4 different actors performing varied actions outdoors with a natural cluttered background. The actors perform complex actions and switch between actions within the same video. The poses they assume include many with significant self-occlusion. We have annotated close to 1000 frames of data and will be making this data available to the community for future evaluation. We show qualitative results in Figure 7.

Sequences from [23]: We also test our method on the *walkstraight* and *baseball* sequences used in [23] for evaluation and report PCP scores on these videos. We show qualitative results in Figure 6.

4.2. Detection Accuracy

We use two metrics to evaluate our algorithm. We use the *PCP* criterion as in [8] and *keypoint localization error* (KLE). Keypoint localization error measures the average euclidean distance from the ground truth keypoint normalized scaled by the size of the head in each frame to correct for scale changes. As our method (and most 2D pose estimation methods) cannot distinguish between left and right limbs we report the score of the higher scoring assignment. We obtain significantly better results than our baseline [23] on the outdoor pose dataset as reported in Table 1. The main improvements are in the tracking of the lower limbs which are especially susceptible to double counting errors. Our method reduces the double counting artifacts and enforces temporal smoothness for each part resulting in smoother and more accurate tracks. We also show improvments on the sequences used in [23], PCP and KLE accuracies are reported in Table 2.

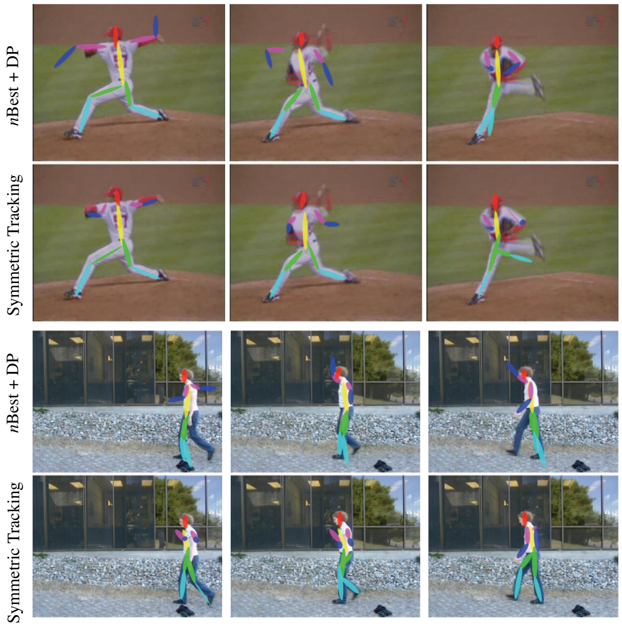


Figure 6: Qualitative Comparison. We show improvement frames on two of the sequences used in [23].

Metric	Method	Head	Torso	U.L.	L.L.	U.A.	L.A.
PCP	Ours [23]	1.00 1.00	0.69 0.61	0.91 0.86	0.89 0.84	0.85 0.66	0.42 0.41
KLE	Ours [23]	0.53 0.54	0.88 0.74	0.67 0.80	1.01 1.39	1.70 2.39	2.68 4.08

Table 2: PCP scores and keypoint localization error for the *baseball* and *walking* videos. We outperform the baseline due to better temporal consistency and occlusion handling.

4.3. Double counting errors

We observe a significant decrease in the number of double counting errors of our method over the baseline (Figure 9). In the outdoor pose dataset we reduce the number of double counting errors by substantially by around 75 %, while we observe a decrease of approximately 41 % on the HumanEva-I sequences.

5. Discussion

We have presented an occlusion-aware model for tracking human pose in video which addresses the problem of double-counting by explicitly modelling the tracking of symmetric parts as a multi-target tracking problem. We

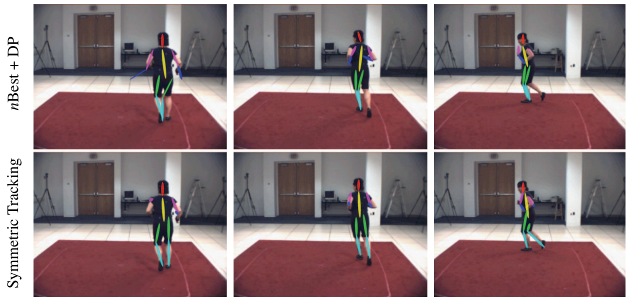


Figure 8: Qualitative Comparison. We show improvement frames on a sequence from the HumanEva-I dataset. We reduce double counting errors by reasoning about occlusion and enforcing mutual exclusion constraints.

Metric	Method	Head	Torso	U.L.	L.L.	U.A.	L.A.
PCP	Ours [23]	0.99 0.97	1.00 0.97	0.99 0.97	0.98 0.90	0.99 0.83	0.53 0.48
KLE	Ours [23]	0.27 0.23	0.48 0.52	0.13 0.24	0.22 0.35	1.14 1.10	1.07 1.18

Table 3: HumanEvaI evaluation. PCP scores and keypoint localization error for sequences from the HumanEva-I dataset. We obtain significant improvement over the baseline due to better temporal consistency and occlusion handling. We particularly perform well on the lower and upper legs which typically are difficult because of mutual occlusions.

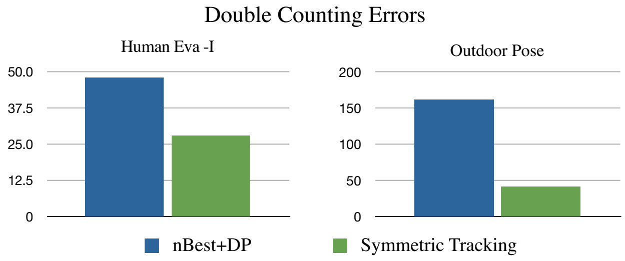


Figure 9: Reduction in double counting. We achieve a reduction in double counting errors on both our evaluation datasets due to better occlusion reasoning and mutual exclusion constraints.

argue that temporal continuity is a strong cue for understanding occlusion and therefore propose to track parts individually or in symmetric pairs resulting in a significant reduction in false positive occlusions and double counting. Future work will aim to infer a depth ordering between mutually occluding parts that enables 3D understanding and modelling the probability of transition into an occlusion state using appearance cues such as occlusion edges.

References

[1] The MOSEK optimization software. 4

[2] A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. *ECCV*, 2004. 2

[3] M. Andriluka, S. Roth, and B. Schiele. Pictorial Structures Revisited: People Detection and Articulated Pose Estimation. *CVPR*, 2009. 2

[4] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. *CVPR*, 2010. 2

[5] A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. *ECCV*, 2010. 2, 4, 5

[6] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 2004. 5

[7] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *PAMI*, 2011. 2, 4

[8] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008. 6

[9] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single-view geometry. *ECCV*, 2012. 1

[10] D. Gavrila. The visual analysis of human movement: A survey. *Computer vision and image understanding*, 1999. 2

[11] J. Gibson. What gives rise to the perception of motion? *Psychological Review*, 1968. 1

[12] J. Gibson, G. Kaplan, H. Reynolds, and K. Wheeler. The change from visible to invisible. *Attention, Perception, & Psychophysics*, 1969. 1

[13] H. Jiang. Human pose estimation using consistent max-covering. In *ICCV*, 2009. 2

[14] H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. In *CVPR*, 2007. 2, 4

[15] H. Jiang and D. Martin. Global pose estimation using non-tree models. *CVPR*, 2008. 2

[16] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 2

[17] L. Karlinsky and S. Ullman. Using linking features in learning non-parametric part models. *ECCV*, 2012. 2

[18] X. Lan and D. Huttenlocher. A unified spatio-temporal articulated model for tracking. In *CVPR*, 2004. 2

[19] X. Lan and D. Huttenlocher. Beyond trees: Common-factor models for 2d human pose recovery. In *ICCV*, 2005. 2

[20] T. Moeslund, A. Hilton, V. Krüger, and L. Sigal. *Visual analysis of humans: looking at people*. Springer, 2011. 2

[21] V. Morariu, D. Harwood, and L. Davis. Tracking people's hands and feet using mixed network and/or search. *PAMI*, 2012. 2

[22] D. Morris and J. Rehg. Singularity analysis for articulated object tracking. In *CVPR*, 1998. 2

[23] D. Park and D. Ramanan. N-best maximal decoders for part models. In *ICCV*, 2011. 2, 6, 7, 8

[24] H. Park and Y. Sheikh. 3d reconstruction of a smooth articulated trajectory from a monocular image sequence. *ICCV*, 2011. 1

[25] D. Ramanan. Learning to parse images of articulated bodies. *NIPS*, 2007. 2

[26] D. Ramanan, D. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *CVPR*, 2005. 2

[27] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *PAMI*, 2007. 6

[28] D. Remus and S. Engel. Motion from occlusion. *Journal of Vision*, 2003. 1

[29] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP-Image Understanding*, 1994. 2

[30] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *CVPR*, 2011. 2

[31] L. Sigal and M. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR*, 2006. 1, 2

[32] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Bm^{3}e: Discriminative density propagation for visual tracking. *PAMI*, 2007. 2

[33] M. Sun, M. Telaprolu, H. Lee, and S. Savarese. An efficient branch-and-bound algorithm for optimal human pose estimation. In *CVPR*, 2012. 2

[34] H. Trinh, Q. Fan, P. Gabbur, and S. Pankanti. Hand tracking by binary quadratic programming and its application to retail activity recognition. 2012. 2

[35] Y. Wang and G. Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. *ECCV*, 2008. 2

[36] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 2, 3, 4, 5, 6

[37] B. Yao and L. Fei-Fei. Action recognition with exemplar based 2.5 d graph matching. *ECCV*, 2012. 1

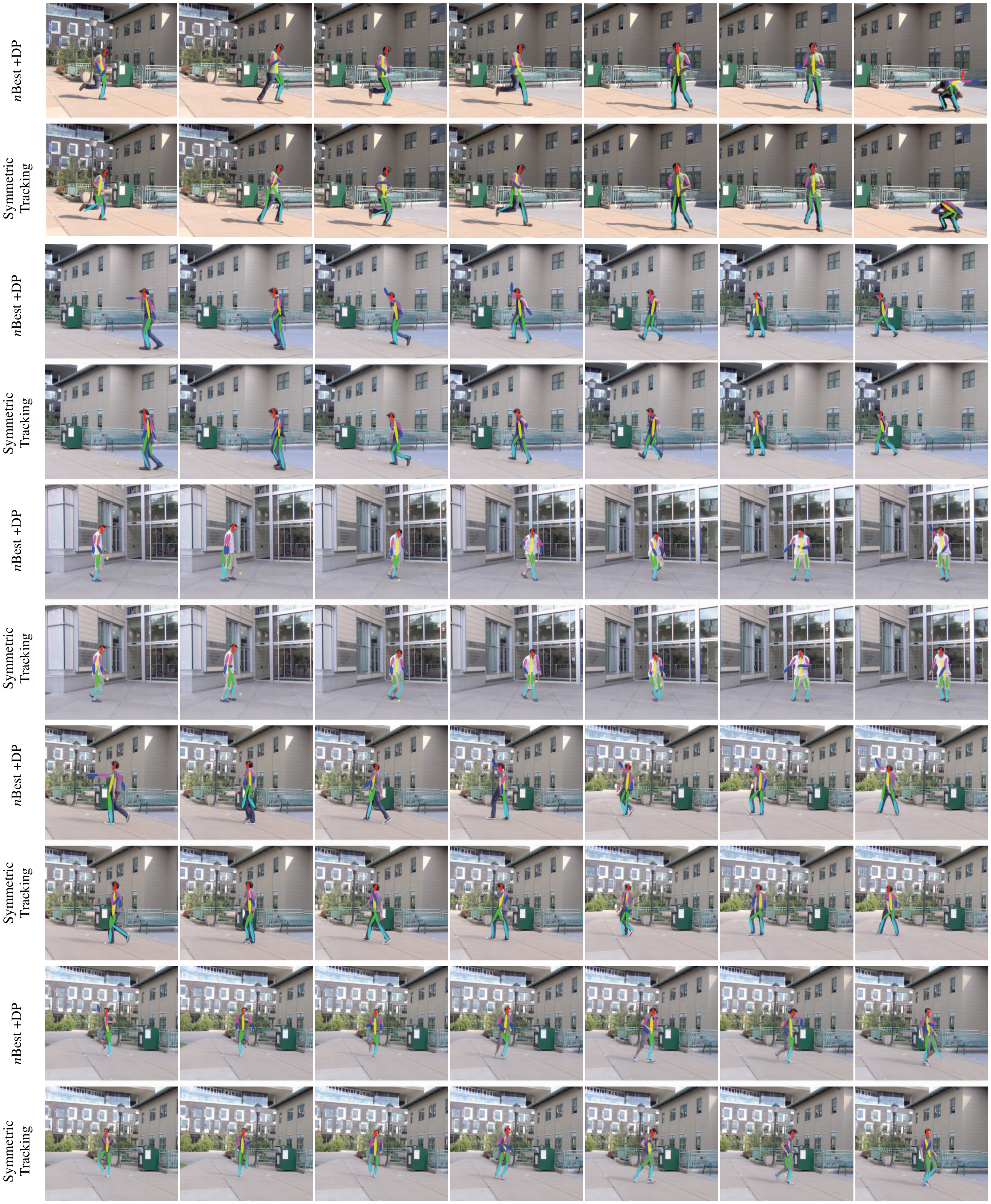


Figure 7: Qualitative Comparison. We show frames of symmetric tracking of human pose in comparison to the baseline [23] on outdoor pose dataset. Note that our method reduces double counting errors especially on frames when the person is entering a profile view with mutual occlusion.