3D Multi-Person Pose Estimation from Multiple Views using Graph Partitioning

Julian Tanke Juergen Gall
University of Bonn
tanke@informatik.uni-bonn.de

Abstract

In this work we address the problem of estimating multiple 3D human poses from a set of calibrated cameras. Estimating 3D human poses from multiple views has several compelling properties: humans can be projected into a joint global coordinate space, sets of cameras can cover a much larger area than a single camera could, and ambiguities, occlusions and motion blurs can be resolved by other views. We propose a bottom-up approach where we first triangulate geometrically reasonable 2D joint pairs from which we then build a sparse graph with edges between close-by joints of the same type and edges between joints that form limbs. Partitioning this graph yields a set of 3D human pose estimations. Our approach is robust and can handle situations where each camera only has a small field of view with people only partially visible. We obtain competitive results for single person 3D pose estimation and state-of-the art results for multi-person 3D pose estimation.

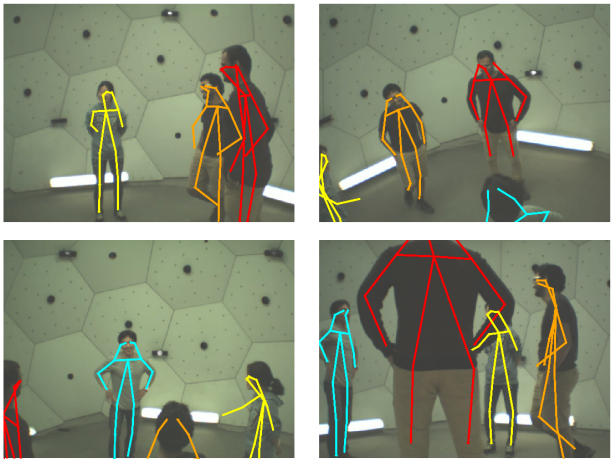

Figure 1. Projections of 3D human poses estimated by our model on a challenging configuration of the CMU Panoptic Dataset [16] with only four calibrated camera views. The humans occlude each other and individuals are only partially visible in some views. Our bottom-up approach manages to separate the different individuals and estimates their poses.

1. Introduction

Estimating 3D human body poses from images is a difficult task which has applications in many areas such as sports [7], human-computer interfacing [23], and surveillance [27]. Most approaches [14, 18, 20, 25] address 3D human pose estimation from single images while multi-view 3D human pose estimation [7, 17, 3, 4, 11] remains comparatively unexplored. Utilizing multiple views has several advantages over monocular 3D human pose estimation: ambiguities introduced by foreshortening as well as body joint occlusions or motion blurs can be resolved using other views. Furthermore, human poses can be projected into a joint global coordinate system when using calibrated cameras. Current proposed solutions [3, 4] employ a top-down approach which suffers from early commitment as multiple views induce difficult ambiguities.

We propose a bottom-up approach that extracts 3D human poses for an arbitrary number of people in unconstrained multiple views using graph partitioning. We utilize a pre-trained neural network to extract 2D joint candidates and 2D part affinity fields [8] for each camera. 3D joint candidates are generated by triangulating the 2D joints from all cameras. We employ two strategies to reduce the number of 3D points: first we only select 2D pairs for triangulation if geometrically reasonable (see Figure 3) and second we cluster the 3D points using mean shift [9]. We then construct a graph where the edges that represent limbs are weighted by their projections to the cameras part affinity fields. The resulting graph is sparse as we operate in world coordinates where we can reason about sizes and thus drop any edge with unreasonable length. Graph partitioning is applied to extract 3D human pose candidates on which we then apply another graph partitioning over their 2D projections to resolve any ambiguities. Our model can be applied to unseen data without the need to fine-tune the 2D pose estimation neural network. To the best of our knowledge we present the first complete bottom-up approach to solve multi-person pose estimation from multiple cameras.

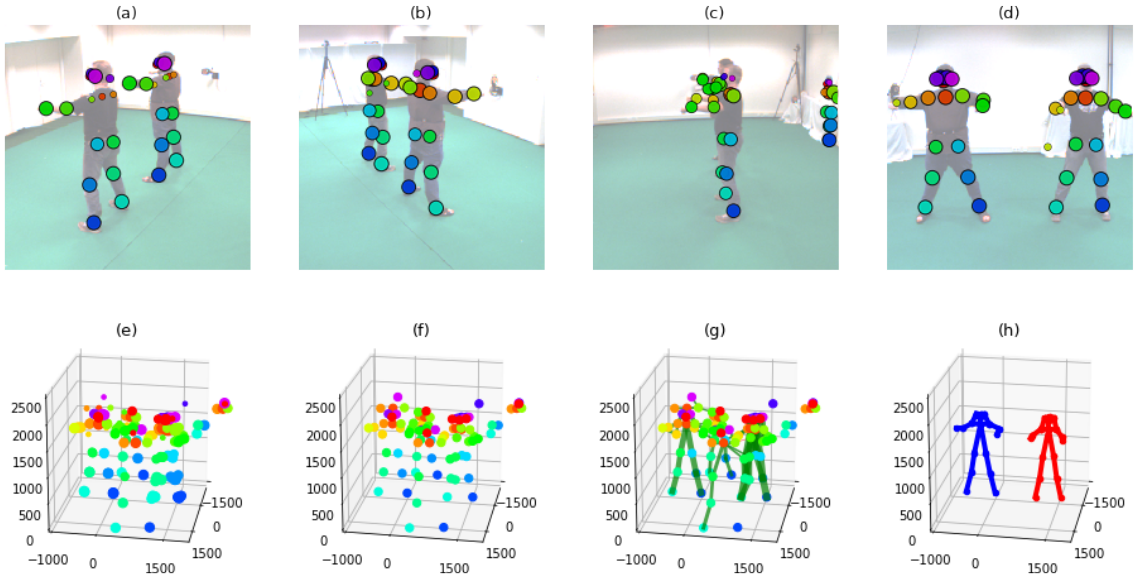

Figure 2. Step-by-step description of our model: (a) - (d) calculate 2D joint candidates for every camera view separately. (e) Extract 3D candidate points by triangulating all 2D joint candidates that are consistent with the epipolar geometry in all cameras. (f) Merge 3D point clusters using the mean shift [9] algorithm. (g) Build graph structure where the limb weights are defined by the projections into all camera part affinity fields. (h) Solve the graph partitioning and estimate 3D human poses in world coordinates.

2. Related Work

Our model is built on advancements in the field of 2D multi-person pose estimation. Convolutional pose machines [26] is a sequential architecture composed of repeated convolutional neural networks who refine body joint confidence maps. Part affinity fields [8] are 2D vector fields who represent associations between body joints which form limbs. Both architectures utilize a greedy bottom-up approach which is robust to early commitment and which decouples the runtime complexity from the number of people on the image, yielding real-time performance.

Martinez et al. [18] split the problem of inferring 3D human poses from single images into estimating a 2D human pose and then regressing the 3D pose on the low-dimensional 2D representation. Though 3D human pose estimation approaches from single images yield impressive results they do not generalize well to unconstrained data. Multiple views [21, 22] can be used to guide the training.

A common technique to estimate a single 3D human pose from multiple views is to extend the well-known pictorial structure model [12] to 3D [2, 6, 7, 17, 21]. Burenius et al. [7] utilize a 2D part detector based on the HOG-descriptor [10] while Kazemi et al. [17] use random forests. Pavlakos et al. [21] outperform all previous models by utilizing the well-known Stacked Hourglass network [19] to extract human joint confidence maps from the camera views. However, these models have to discretize their solution space resulting in either a very coarse result or a very large state space making them impractical for estimating 3D poses of multiple people. Furthermore, they restrict their solution space to a 3D bounding volume around the subject which has to be known in advance. Estimating multiple humans from multiple views was first explored by Belagiannis et al. [3, 4]. Instead of sampling from all possible translations and rotations they utilize a set of 3D body joint hypotheses which were obtained by triangulating 2D body part detections from different views. However, these methods rely on localizing bounding boxes using a person tracker for each individual in each frame to estimate the number of persons that has to be inferred from the common state space. This will work well in cases where individuals are completely visible in most frames but will run into issues when the pose is not completely visible in some cameras as shown in Figure 1. A CNN-based approach was proposed by Elhayek et al. [11] where they fit articulated skeletons using 3D sums of Gaussians [24] and where body part detections are estimated using CNNs. However, the Gaussians and skeletons need to be initialized independently beforehand for each actor in the scene. Our approach does not suffer from any of the aforementioned drawbacks as our model works off-the-shelf and poses are constructed in a bottom-up approach.

3. Model

In this section we detail the steps to build our graph to jointly extract multi-person poses from multiple views via

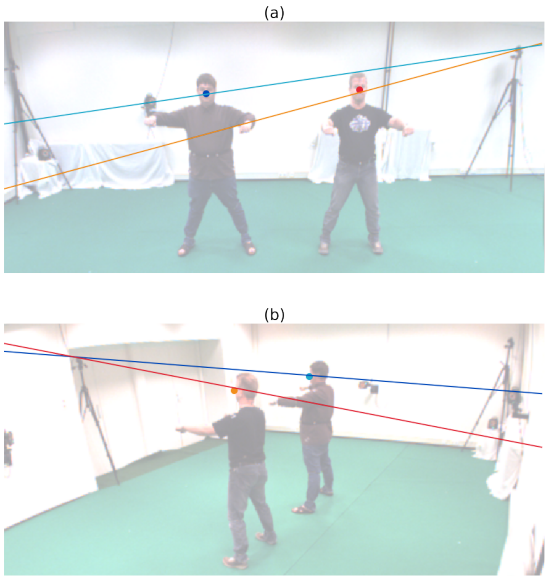

Figure 3. Epipolar lines for two camera views. The blue and the red dot in image (a) are projected as blue (red) epipolar lines in the second image (b) while the orange and light-blue dot from image (b) are projected onto image (a).

graph partitioning, as illustrated in Figure 2. First we discuss how we obtain 3D joint candidates which form the vertices of our graph. Then we proceed with defining the edges and constraints that the graph has to adhere to. Finally we discuss the post-processing where human poses are extracted from the connected components of the partitioned graph and where multi-view ambiguities are being resolved.

3.1. 3D Joint Candidates

For each calibrated camera $i \in \{1, \dots, n\}$, where n is the total number of cameras, we estimate a joint confidence map S_j^i for each joint type $j \in \{1, \dots J\}$ and a part affinity field L_l^i, which encodes location and orientation of limbs in the image domain, for each limb $l = (j, \hat{j}) \in \mathcal{L}$ [1] as described by Cao et al. [8]. We then extract 2D joint candidates D_j^i for each joint type j in each view i by selecting pixel locations where the confidence is a local maximum and higher then a threshold ϵ_{conf}, as shown in Figure 2 (a)-(d). For each joint type j we now generate 3D candidates K_j by triangulating the 2D detections D_j^i in each view with the 2D detections in all other views where the confidence $\Pr(A)$ of the 3D point A is the average confidence of its two 2D source points, as shown in Figure 2 (e). To avoid triangulating point pairs that clearly do not fit together geometrically, we project the epipolar line of each point into each others views. Two points are then only triangulated if both their pixel distances to their respective epipolar lines are below a threshold ϵ_{epi}, as illustrated in Figure 3. We use the threshold rather than only selecting the pair with closest distances as, due to noise in detection and calibration and due to the tendency of people to flock together, *wrong* pairs might have a closer distance than the *correct* one. Depending on the number of cameras, many redundant 3D points will be generated. As the 3D points represent our graph vertices we would like to reduce their number as much as possible to simplify our graph partitioning. To do so we apply the well-known mean shift [9] algorithm with a Gaussian kernel, parameterized by $\sigma_{\mathbf{ms}}$ and weighted by confidence $\Pr(A)$ of each 3D point A, to fuse 3D points in close proximity to each other. The modes found by the mean shift operation define the new, reduced 3D candidates $\mathcal{K}_j$ for each joint type j, which is shown in Figure 2 (f).

[1] $\mathcal{L}$ represents the limbs defined for the MSCOCO keypoint challenge

3.2. Graph Partitioning

To extract human poses from the data, we utilize graph partitioning, similar to PoseTrack [13, 15]. For this, we define a graph structure $G = (D, E)$ with vertices $D = \bigcup_j \mathcal{K}_j$ $\forall j \in \{1, \dots, J\}$ and with two sets of edges $E = E_\iota \cup E_\lambda$. The first set is defined over all joints of the same type that are in close vicinity, defined by a threshold δ, to each other. This is motivated by the observation that, due to noise, some points might end in different clusters during the mean shift step even though they belong to the same ground-truth joint. In order to correct this, we define intra-joint edges for every joint type j:

$$E_\iota = \left\{ (A_j, B_j) : A_j \neq B_j \wedge ||A_j - B_j||_2 < \delta \right\} \quad (1)$$

where $A_j, B_j \in \mathcal{K}_j$. The second set of edges is defined over the limbs $\mathcal{L}$:

$$E_\lambda = \left\{ (A_j, B_{\hat{j}}) : l = (j, \hat{j}) \in \mathcal{L} \wedge \mathrm{s}(l, A_j, B_{\hat{j}}) \right\} \quad (2)$$

where $s(l, A_j, B_{\hat{j}}) := \epsilon^l_{\min} \leq ||A_j - B_{\hat{j}}||_2 \leq \epsilon^l_{\max}$ ensures that only limbs with sensible length are being considered. We obtain a graph partitioning by maximizing to following cost function:

$$\underset{\nu,\iota,\lambda}{\operatorname{argmax}} \; \phi_\iota + \phi_\lambda \quad (3)$$

where $\nu \in \{0,1\}^{|D|}$ represents a set of binary variables that enable or disable certain 3D joint candidates and where $\iota \in \{0,1\}^{|E_\iota|}$ and $\lambda \in \{0,1\}^{|E_\lambda|}$ represent sets of binary variables that control if an intra-joint or limb edge should be removed or not. The binary term for intra-joint edges

ϕ_ι penalizes distance between points while the binary term for limb edges ϕ_λ measures the consistency between a limb edge and the part affinity field:

$$\phi_\iota = \sum_{(A_j,B_j)\in E_\iota} \iota(A_j,B_j)\big(-\tanh(||A_j - B_j||_2)\big) \quad (4)$$

$$\phi_\lambda = \sum_{(A_j,B_{\hat{j}})\in E_\lambda} \lambda(A_j,B_{\hat{j}}) \log \frac{\Psi(A_j,B_{\hat{j}})+1}{1-\Psi(A_j,B_{\hat{j}})} \quad (5)$$

where Ψ measures the angular difference of the limb with the underlying part affinity field averaged over all cameras using the dot product. To apply this, we need to project the 3D joint candidates $A_j, B_{\hat{j}}$ into all views i which we denote by $a_j^i, b_{\hat{j}}^i$.

$$\Psi(A_j,B_{\hat{j}}) = \frac{1}{n}\sum_{i=1}^{n}\int_{u=0}^{u=1} L_l^i(p(u)) \cdot \frac{b_{\hat{j}}^i - a_j^i}{||b_{\hat{j}}^i - a_j^i||_2}\,\mathrm{d}u \quad (6)$$

where $p(u)$ interpolates the two 2D joint locations $a_j^i, b_{\hat{j}}^i$, $p(u) = (1-u)a_j^i + ub_{\hat{j}}^i$, and where $L_l^i(x)$ yields the 2D vector from the part affinity field for a limb $l = (j,\hat{j}) \in \mathcal{L}$ in the i-th camera at the 2D location x. We clamp this vectors so that Ψ yields values between -1, meaning that the limb points in the inverse direction of the part affinity field, and 1, meaning that the limb perfectly fits the part affinity field.

To obtain a partitioning that represents valid human 3D poses we need to enforce a set of constraints. First we need to ensure that only those edges are considered where both end points are valid:

$$2\lambda(A,B) \le \nu(A) + \nu(B), \quad (7)$$
$$2\iota(A,B) \le \nu(A) + \nu(B)\,. \quad (8)$$

Furthermore, we need to ensure transitivity between edges:

$$\lambda(A,C) + \lambda(B,C) - 1 \le \iota(A,B), \quad (9)$$
$$\lambda(A,C) + \iota(A,B) - 1 \le \lambda(B,C), \quad (10)$$
$$\iota(A,C) + \iota(A,B) - 1 \le \iota(B,C). \quad (11)$$

Conversely, if a vertex A is connected to vertices B and C but there is no edge between B and C either the edge AB or AC has to be removed:

$$\iota(A,B) + \iota(A,C) \le 1, \quad (12)$$
$$\lambda(A,B) + \lambda(A,C) \le 1, \quad (13)$$
$$\lambda(A,B) + \iota(A,C) \le 1. \quad (14)$$

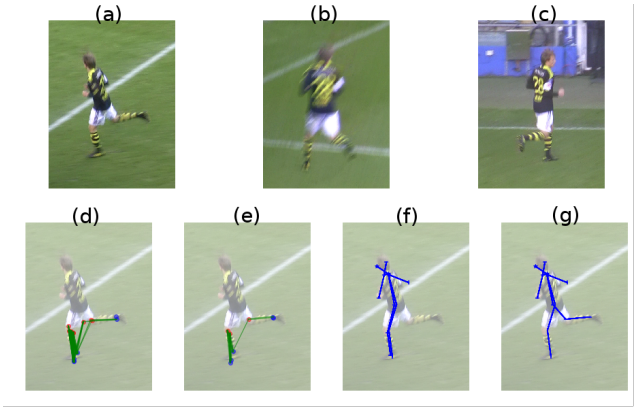

Figure 4. Enforcing intersection constraints: by avoiding intersection of different joint types we extract more robust poses. This is especially noticeable in the case of symmetric joint types like left/right hand, etc. (a) - (c): person from the *KTH Multiview Football Dataset II* [17] with motion blur and body twist. The part affinity model struggles to decide what body joints are *left* and which ones are *right*. The graph representation of the left lower leg is represented in (d) where the actual left lower leg gets a high response (thick green lines). However, the actual right leg also gets a slightly smaller response for being a left leg. The edges for the right leg are depicted in (e) where the response is higher on the actual left lower leg than on the actual right lower leg. If we do not enforce intersection constraints (f) the left and right leg will collapse into the same limb as this will give the highest response. When enforcing the constraints (g) the right leg cannot be put onto the left leg anymore. However, the actual right leg still yields a positive response resulting in a more accurate pose estimation.

Last but not least we introduce intersection constraints which ensure that different joint types do not occupy the same space:

$$\nu(A) + \nu(B) \le 1 \;\text{ if }\; ||A - B||_2 \le \epsilon_{\text{intersection}}. \quad (15)$$

This is useful especially in the case of symmetric limbs like left/right lower arm or lower leg, as the model might confuse those, especially in cases with motion blur or large body twists, as depicted in Figure 4.

3.3. Post Processing

Following the steps given in Section 3.2, we obtain a reduced graph where some vertices and edges are removed, based on ν, ι and λ and where 3D human poses are encoded as connected components. As each connected component can contain multiple vertices who represent the same joint type, we use the weighted average to create a single 3D location for each joint. After dropping every connected component that has less than ϵ_{joints} joints, a set $\mathcal{H} = \{H_k\}_{k=1}^m$ of m surviving 3D human pose candidates emerges. However, some configurations of camera pairs and 3D points result in a setup where the 3D positions are ambiguous even

Figure 5. View ambiguities introduced by persons in similar poses result in an effect we call *ghosting* where valid projections hallucinate a non-existing 3D human pose on the UMPM dataset [1]. This is especially prevalent when two cameras face each other. When projecting this pose into other views, the error becomes evident. However, we do not expect a person to be visible in all views so any person that is visible in at least two cameras must be considered valid. To resolve the ghosting, we utilize another graph partitioning explained in Section 3.3.

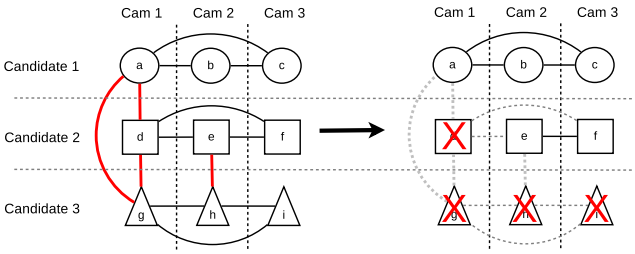

Figure 6. Post-processing step to resolve conflicting human pose candidates: The left graph represents a sample with three human pose candidates and three camera views. In Camera 1, all human candidates are overlapping each other resulting in conflict edges (red edges, E_κ) while in Camera 2 only candidate 2 and 3 conflict. Last but not least, Camera 3 is free of conflict. To guide the algorithm towards rewarding candidates with the least conflicts, we add edges between the same candidate in different views (black edges, E_χ). The right hand side shows the result of our graph partitioning algorithm where all conflicts are resolved. Candidate 1 is visible in all three views, candidate 2 is visible in camera 2 and 3 and candidate 3 is being removed.

when projected into both camera views. This problem was already observed by Belagiannis et al. [3, 4] and can be seen in Figure 5.

To extract our final 3D human pose estimations from the candidate set $\mathcal{H}$, we utilize another graph partitioning over a graph $G_p = (D_p, E_p)$ where we optimize for a configuration $\pi = \{0,1\}^{|\mathcal{H}|}$ that controls whether a 3D human pose candidate is removed or not. To resolve the ghosting we check how well every 3D human candidate H_k projects into each camera view i. When two or more candidates overlap in a camera view, we mark them as conflicting, which has to be resolved by removing all but one conflicting candidate from the camera view, as shown on the right-hand side in Figure 6. We furthermore encourage 3D pose candidates to be visible in as many views as possible. A vertex $h_k^i \in D_p$ is defined for every 3D human pose k and every camera i:

$$D_p = \left\{ h_k^i : \forall i \in \{1,\dots,n\},\ \forall k \in \{1,\dots,|\mathcal{H}|\} \right\}. \quad (16)$$

Again, we distinguish between two different edge types $E_p = E_\kappa \cup E_\chi$ where E_κ represents conflicts when two or more persons overlap in a camera view and where E_χ counts the number of valid projections of a single 3D candidate. In Figure 6, the red edges represent E_κ while the black ones represent E_χ. They are defined as follows:

$$E_\kappa = \left\{ (h_{k_1}^i, h_{k_2}^i) : k_1 \neq k_2 \wedge \mathrm{c}(h_{k_1}^i, h_{k_2}^i) \geq \epsilon_{\text{covering}} \right\} \quad (17)$$

$$E_\chi = \left\{ (h_k^{i_1}, h_k^{i_2}) : i_1 \neq i_2 \right\} \quad (18)$$

where $\mathrm{c}(h_{k_1}^i, h_{k_2}^i)$ calculates the maximum covering of two bounding boxes:

$$\mathrm{c}(h_1, h_2) = \max\left(\frac{|h_1 \cap h_2|}{|h_1|}, \frac{|h_1 \cap h_2|}{|h_2|} \right) \quad (19)$$

where $|h_k^i|$ is the size of the bounding box of the projected person k into view i where the corner points of the bounding box are defined by the most left/right/top/bottom 2D locations of the projected joints in h_k^i. Similarly, $|h_{k_1}^i \cap h_{k_2}^i|$ calculates the intersecting area between the two bounding boxes of person k_1 and k_2 in view i. This function yields more stable results under large variation in bounding box sizes than intersection over union. To find the set of candidates that are best described by the cameras without conflict, we maximize a cost function over $\pi = \{0,1\}^{|\mathcal{H}|}$, which enables or disables 3D human candidates, $\rho \in \{0,1\}^{|D_p|}$, which enables or disables visibility of a 3D pose candidate in a certain camera view and $\chi \in \{0,1\}^{|E_\chi|}$, which connects the camera views:

$$\underset{\pi,\rho,\chi}{\operatorname{argmax}}\Big(\sum_{h\in D_p}\rho(h)\Upsilon(h)\Big)+\Big(\sum_{(a,b)\in E_\chi}\chi(a,b)\Big) \quad (20)$$

where the function $\Upsilon(h^i) = \frac{1}{J}\sum_{j=1}^{J} S_j^i(h^i(j)) \in [0,1]$ sums the confidence scores at all projected joints in camera i. To ensure that conflicts are resolved, the following constraint is added for all vertices $a \in D_p$:

$$\rho(a) + \sum_{b\,:\,(a,b)\in E_\kappa}\rho(b) \leq 1. \quad (21)$$

This means that if multiple person candidates are in conflict in a camera view, only one of them can be selected. Counting only valid projections of 3D candidates in the cost function is ensured by the following constraint:

$$2\chi(a,b) \leq \rho(a) + \rho(b) \quad (22)$$

where $a, b \in D_p$. The optimization has to ensure that all projections of a 3D pose candidate k are disabled if the 3D pose itself is removed:

$$\sum_{a\in D_p(k)}\rho(a) \leq n\pi(k) \quad (23)$$

where $D_p(k)$ returns all vertices $\in D_p$ that are from the same 3D human pose k. On top of that, a valid 3D pose must be visible in at least two views to ensure triangulation:

$$2\pi(k) \leq \sum_{a\in D_p(k)}\rho(a) \quad (24)$$

3.4. Pose Completion

Due to the bottom-up nature of our approach, a predicted 3D human pose might have missing limbs if a body joint is not visible in at least two views due to occlusion. We devise a simple approach that regresses the missing body joints by learning common human poses. We reduce the number of joints per 3D pose to 13 by removing all head joints as different datasets differ greatly in head annotations and as problems of missing limbs usually affect the extremities such as arms and legs. The model is defined as follows:

$$p = W_2 \tanh(W_1\hat{p}) \quad (25)$$

where $\hat{p} \in \mathbb{R}^{52}$ is a vector representing the body pose with missing joints where each of the 13 joints takes four components with the first three elements representing the (x, y, z) coordinates and where the forth element is either 1 if the joint is visible or 0 otherwise. The output vector $p \in \mathbb{R}^{39}$ represents the regressed 3D body pose where every joint takes three elements representing the new (x, y, z) coordinates. $W_1 \in \mathbb{R}^{52\times64}$ projects the input data to a high-dimensional space which is then regressed by $W_2 \in \mathbb{R}^{64\times39}$. As we use the mean squared error as loss function this model can be iterpreted as kernelized linear regression which jointly learns the kernel projection and linear regression. As training data, we utilize 3D poses from the UMPM [1] *p2_ball_1* dataset which we center at the origin. At inference time, we shift a 3D pose at the origin by subtracting its mean before passing it to the model. The original position is recovered by adding the mean back to the regressed 3D pose. In difficult settings with few cameras and many occlusions, this method helps to recover complete human poses.

	[7]*	[17]*	[21]*	[3]	[4]	Ours	Ours+
ua	.60	.89	1.0	.68	.98	.96	.96
la	.35	.68	1.0	.56	.72	.87	.91
ul	1.0	1.0	1.0	.78	.99	.96	.97
ll	.90	.99	1.0	.70	.92	.95	.96
avg	.71	.89	1.0	.68	.90	.94	**.95**

Table 1. Quantitative comparison of methods to solve single human 3D pose estimation from multiple views on the KTH Football II [17] dataset. The numbers are the PCP score in 3D with $\alpha = 0.5$. Methods annotated with * can only estimate single human poses and rely on being provided with a tight 3D bounding box centered at the true 3D location of the person. *Ours*+ and *Ours* describe our method with and without pose completion (see Section 3.4). *ul* and *la* show the scores for upper and lower arm respectively while *ul* and *ll* represent upper and lower legs.

4. Experiments

We evaluate our model on two tasks, single person 3D pose estimation and multi-person 3D pose estimation, and compare it with state-of-the art methods. Percentage of correct parts (PCP) in 3D as described in [7] is used to quantify the experiments. Last but not least we assess our models hyperparameters. The source code of the approach will be released upon acceptance.

4.1. Single Person 3D Pose Estimation

Naturally first works on 3D human pose estimation from multiple views cover single humans. Typical methods [7, 17, 21] find a solution over the complete discretized state space which is intractable for multiple persons. However, we report their results for completeness. All models were evaluated on the complete first sequence of the second player on the KTH Football II [17] dataset. Our results are reported in Table 1. Our model outperforms all

Campus dataset

	[3]			[4]			Ours			Ours$^+$		
Actor	1	2	3	1	2	3	1	2	3	1	2	3
upper arms	.83	.90	.78	.97	.97	.90	.81	.99	.96	.82	.99	.97
lower arms	.78	.40	.62	.86	.43	.75	.61	.60	.73	.63	.60	.78
upper legs	.86	.74	.83	.93	.75	.92	1.0	1.0	.99	1.0	1.0	.99
lower legs	.91	.89	.70	.97	.89	.76	1.0	.997	.99	1.0	.997	.99
average	.85	.73	.73	.93	.76	.83	.86	.90	.92	.86	.90	.93
average*		.77			.84			.89			**.90**	

Shelf dataset

	[3]			[4]			Ours			Ours$^+$		
Actor	1	2	3	1	2	3	1	2	3	1	2	3
upper arms	.72	.80	.91	.82	.83	.93	.97	.81	.94	.98	.81	.94
lower arms	.61	.44	.89	.82	.83	.93	.90	.39	.95	.93	.46	.95
upper legs	.37	.46	.46	.43	.50	.57	.99	.96	1.0	.99	.96	1.0
lower legs	.71	.72	.95	.86	.79	.97	.99	.96	.997	.99	.96	.997
average	.60	.61	.80	.73	.74	.85	.96	.78	.97	.97	.80	.97
average*		.67			.77			.90			**.91**	

Table 2. Quantitative comparison of multi-person 3D pose estimation from multiple views on the evaluation frames of the annotated [4] Campus and Shelf dataset [5]. The numbers are the PCP score in 3D with $\alpha = 0.5$. *Ours*$^+$ and *Ours* describe our method with and without pose completion (see Section 3.4). We show results for each of the three actors separately as well as averaged for each method (*average**).

Shelf dataset						**UMPM dataset**			
Ours			Ours$^+$			Ours		Ours$^+$	
1	2	3	1	2	3	1	2	1	2
.89	.23	.87	.89	.23	.87	.94	.93	.94	.94
.69	.18	.84	.70	.18	.84	.91	.89	.91	.89
.97	.80	.98	.97	.80	.98	.99	.98	.99	.98
.97	.82	.91	.97	.82	.96	.95	.97	.97	.97
.88	.51	.90	.88	.51	.91	.95	.94	.95	.94
.76			.77			.95		.95	

Table 3. PCP scores on Shelf and UMPM dataset with $\alpha = 0.3$. The table structure is identical to Table 2.

other multi-person multiple view setups and gets close to the state-of-the art results on single human pose estimation models which utilize strong assumptions and are much more constrained. As expected, the model has the most difficulty with lower arms (*la*) which experience strong deformation and high movement speed and which are comparatively small.

4.2. Multi-Person 3D Pose Estimation

To evaluate our model on multi-person 3D pose estimation, we utilize the Campus [3] and Shelf [3] dataset. The difficulty of the Campus dataset lies in its low resolution (360×288 pixel) which makes accurate joint detection hard. Furthermore, small errors in triangulation or detection will result in large PCP errors as the final score is calculated on the 3D joint locations. Clutter and humans occluding each others make the Shelf dataset challenging. Nevertheless, our model managed to achieve state-of-the art results on both datasets by a large margin which can be seen in Table 2. This can be attributed on the one hand to our strong human part detector and on the other hand to the bottom-up nature of our approach, especially in the case of the Shelf dataset.

The PCP score with $\alpha = 0.5$ became the standard evaluation practice in previous work [3, 4, 7, 17, 21]. However, the parameter $\alpha = 0.5$ allows a large margin of error which is sensible when the data is noisy and has low resolution. However, datasets with higher resolution can be evaluated with a more aggressive parameter of $\alpha = 0.3$ to get a better understanding of the actual quality of the prediction. We show our results with $\alpha = 0.3$ on the complete Shelf dataset as well as on every 5-th frame of the UMPM video *p2_free_1* in Table 3.

4.3. Hyperparameters

Our model utilizes a set of parameters on which we perform ablation studies by fixing all but one parameter. We run the experiments on the KTH Football II and Campus dataset, our results can be seen in Figure 7. First, we evaluate the spread σ_{ms} of the mean shift kernel (a). We notice that the optimal value for KTH Football II is at $\sigma_{\text{ms}} = 400$ while the Campus dataset is not affected as much. This can be explained by the noisy camera calibration of the KTH Football datasets which uses an idealized orthographic pro-

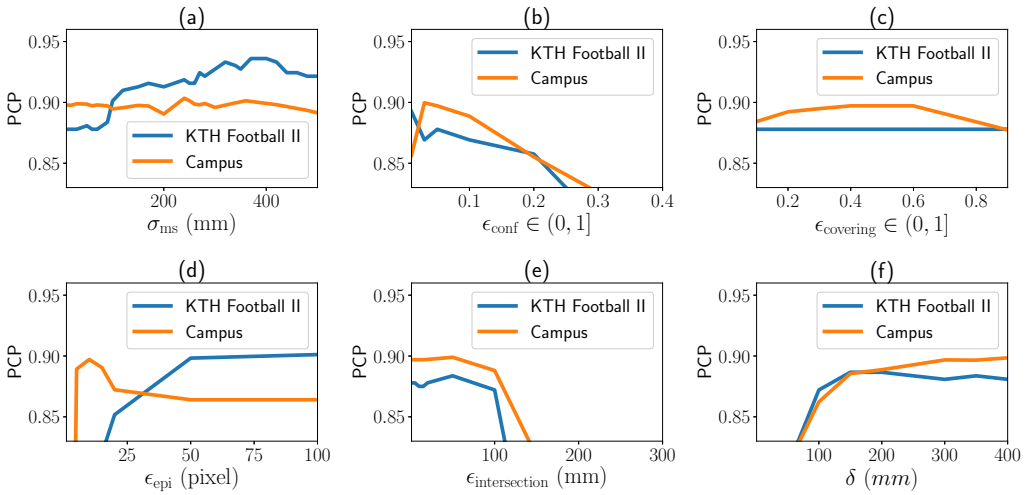

Figure 7. Evaluation of different model parameters on the KTH Football II and Campus dataset where the y-axis corresponds to the PCP score and where the x-axis corresponds to the respective parameter value. Higher PCP scores are better.

jection, which only approximates the true moving camera setup, making exact triangulation more difficult. A wider spread of triangulated points is thus expected and a high value in σ_{ms} helps to associate points that would otherwise end up in different clusters. The Campus dataset, on the other hand, uses fixed projective cameras where triangulation is less noisy.

In Figure 7 (b) we study the threshold ϵ_{conf} to extract 2D joint candidates D_j^i from the confidence maps in each camera i. The KTH Football II dataset suffers from a lot of motion blur due to fast movements resulting in confidence maps with low beliefs, where setting $\epsilon_{\text{conf}} = 0.01$ yields the best results. However, such a low threshold also increases the number of false positive 2D joint candidates which creates problems in multi-person setups, such as the Campus dataset, as seen in the graph.

We evaluate $\epsilon_{\text{covering}}$ in Figure 7 (c), which determines when two 3D pose candidates are in conflict in a view. Too low values result in a too sensitive selection where conflicts are marked even when two candidates barely intersect, resulting in unnecessarily culling correct 3D pose candidates. Too high values, on the other hand, result in ghosting, as described in Section 3.3. As expected, this parameter only affects the multi-person dataset Campus.

The maximal pixel distance ϵ_{epi} from two 2D joint candidates in two different views to their respective epipolar line is explored in Figure 7 (d). Low values ensure less false positive 3D joint candidates but require accurate camera calibration. As the KTH Football II dataset has less accurate calibration it requires a larger parameter.

Figure 7 (e) explores the minimal distance between two different joints in world coordinates as defined in Equation (15). As expected, the parameter behaves very similar in both datasets, where setting the value too large results in joints being wrongfully removed.

Last but not least we evaluate δ in Figure 7 (f) which controls if two 3D joint candidates have an edge in E_ι or not. The main motivation of this parameter is performance as a low value reduces the number of edges in the graph, leaving out more noisy but potentially better point combinations. However, a too low value associates only close-by vertices.

Motivated by our findings we define the default parameters as $\sigma_{\text{ms}} = 400$, $\epsilon_{\text{conf}} = 0.05$, $\epsilon_{\text{covering}} = 0.5$, $\epsilon_{\text{epi}} = 10$, $\epsilon_{\text{intersection}} = 50$ and $\delta = 150$ for all our experiments if not denoted otherwise. When evaluating the KTH Football II dataset, we set $\epsilon_{\text{epi}} = 50$ to compensate for the inaccurate camera model.

5. Conclusion

We introduce the first complete bottom-up approach for multi-person 3D pose estimation from multiple calibrated cameras by utilizing two repeated graph partitionings. The first partitioning associates 3D joint candidates to form human pose candidates while the second partitioning resolves view ambiguities. We achieve competitive results on single human 3D pose estimation without imposing strong restrictions like a pre-defined 3D bounding box while outperforming state-of-the art solutions for multiple humans. Our model works off-the-shelf without the need to fine-tune the CNN for 2D pose estimation on the data.

References

[1] N. v. d. Aa, X. Luo, G. Giezeman, R. Tan, and R. Veltkamp. Utrecht Multi-Person Motion (UMPM) benchmark: a multi-person dataset with synchronized video and motion capture data for evaluation of articulated human motion and interaction. In *Workshop on Human Interaction in Computer Vision*, 2011.

[2] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele. Multi-view Pictorial Structures for 3D Human Pose Estimation. In *British Machine Vision Conference*, 2013.

[3] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3d pictorial structures for multiple human pose estimation. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[4] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3d pictorial structures revisited: Multiple human pose estimation. *Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[5] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[6] M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnörr. A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision*, 2010.

[7] M. Burenius, J. Sullivan, and S. Carlsson. 3d pictorial structures for multiple view articulated pose estimation. In *Conference on Computer Vision and Pattern Recognition*, 2013.

[8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[9] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Transactions on Pattern Analysis and Machine Intelligence*, 2002.

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

[11] A. Elhayek, E. de Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt. Efficient ConvNet-based marker-less motion capture in general scenes with a low number of cameras. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[12] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 2005.

[13] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. Arttrack: Articulated multi-person tracking in the wild. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[14] U. Iqbal, A. Doering, H. Yasin, B. Krüger, A. Weber, and J. Gall. A dual-source approach for 3d human pose estimation from single images. *Computer Vision and Image Understanding*, 2018.

[15] U. Iqbal, A. Milan, and J. Gall. PoseTrack: Joint Multi-Person Pose Estimation and Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[16] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic Studio: A Massively Multiview System for Social Motion Capture. In *International Conference on Computer Vision*, 2015.

[17] V. Kazemi, M. Burenius, H. Azizpour, and J. Sullivan. Multi-view body part recognition with random forests. In *British Machine Vision Conference*, 2013.

[18] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *International Conference on Computer Vision*, 2017.

[19] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, 2016.

[20] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[21] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Harvesting Multiple Views for Marker-less 3D Human Pose Annotations. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[22] H. Rhodin, J. Spörri, I. Katircioglu, V. Constantin, F. Meyer, E. Müller, M. Salzmann, and P. Fua. Learning Monocular 3D Human Pose Estimation from Multi-view Images. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[23] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Conference on Computer Vision and Pattern Recognition*, 2011.

[24] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *International Conference on Computer Vision*, 2011.

[25] D. Tome, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *Conference on Computer Vision and Pattern Recognition*, 2017.

[26] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[27] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *International Conference on Computer Vision*, 2015.