4D Association Graph for Realtime Multi-person Motion Capture Using Multiple Video Cameras

Yuxiang Zhang[1]*, Liang An[1]*, Tao Yu[1], Xiu Li[1], Kun Li[2], Yebin Liu[1,3]
[1]Department of Automation, Tsinghua University [2]Tianjin University
[3]Institute for Brain and Cognitive Sciences, Tsinghua University

Abstract

This paper contributes a novel realtime multi-person motion capture algorithm using multiview video inputs. Due to the heavy occlusions and closely interacting motions in each view, joint optimization on the multiview images and multiple temporal frames is indispensable, which brings up the essential challenge of realtime efficiency. To this end, for the first time, we unify per-view parsing, cross-view matching, and temporal tracking into a single optimization framework, i.e., a 4D association graph that each dimension (image space, viewpoint and time) can be treated equally and simultaneously. To solve the 4D association graph efficiently, we further contribute the idea of 4D limb bundle parsing based on heuristic searching, followed with limb bundle assembling by proposing a bundle Kruskal's algorithm. Our method enables a realtime motion capture system running at 30fps using 5 cameras on a 5-person scene. Benefiting from the unified parsing, matching and tracking constraints, our method is robust to noisy detection due to severe occlusions and close interacting motions, and achieves high-quality online pose reconstruction quality. The proposed method outperforms state-of-the-art methods quantitatively without using high-level appearance information.

1. Introduction

Markerless motion capture of multi-person in a scene is important for many industry applications but still challenging and far from being solved. Although the literatures have reported single view 2D and 3D pose estimation methods [41, 36, 11, 12, 18, 17, 28, 34, 44, 45, 33], they suffer from heavy occlusions and produce low-fidelity results. Comparably, multi-view cameras provide more than one views to alleviate occlusion, as well as stereo cues for accurate 3D triangulation, therefore are indispensable inputs for markerless motion capture of multi-person scenes. While remarkable advances have been made in many kinds of multi-camera motion capture systems for human [30, 31, 24] or even animals [4], most of them fail to achieve the goals of realtime performance and high quality capture under extremely close interactions.

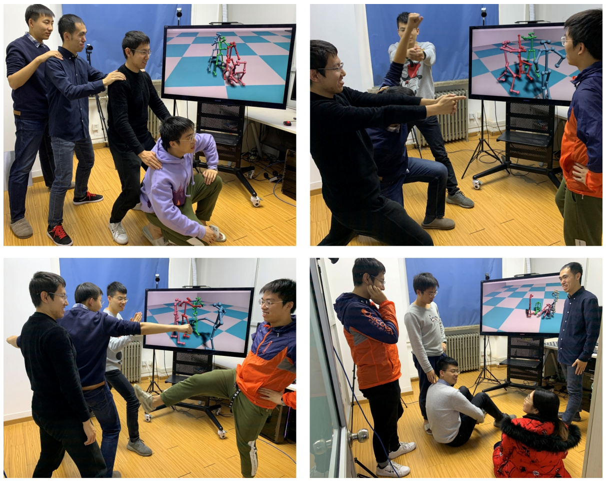


Figure 1. Our method enables multi-person motion capture system working at 30fps for 5 persons using 5 RGB cameras, while achieving high quality skeleton reconstruction results.

Given the 4D (2D spatial, 1D viewpoint and 1D temporal) multiview video input, the key to the success of realtime and high quality multi-person motion capture is how to leverage the rich data input, *i.e.*, how to operate on the 4D data structure to achieve high accuracy while maintaining realtime performance. Essentially, based on the human body part features pre-detected in the separate 2D views using state-of-the-art CNN methods [11], three kinds of basic associations can be defined on this 4D structure. These include single image association (*i.e.*, **parsing**) [11, 20] to form human skeletons in a single image, cross-view association (*i.e.*, **matching**) to establish correspondences among different views, and temporal association (*i.e.* **tracking**) to build correspondences between sequential frames.

Existing methods struggle to deal with all these association simultaneously and efficiently. They consider only

*Equal contribution

parts of these associations, or simply operate them in a sequential manner, resulting in failure to be a high quality and realtime method. For example, the state-of-the-art methods [14, 10, 39] share a similar high-level framework by first performing per-view person parsing, followed by cross-view person matching, and temporal tracking sequentially. They usually assume and rely on perfect per-view person parsing results in the first stage. However, this can not be guaranteed in crowded or close interaction scenarios. Temporal extension [8, 7] of the 3D pictorial structure (3DPS) model [6] apply temporal tracking [23], followed with cross-view parsing using the very time-consuming 3DPS structure optimization. The Panoptic Studio [24] addresses these associations in a sequential manner, by first matching (generate node proposals), then tracking (generate trajectories), and finally assemble the 3D human instances. As it tracks over the whole sequence, it is impossible to achieve realtime performance.

In this paper, we formulate parsing, matching, and tracking in a unified graph optimization framework, called **4D association graph**, to simultaneously and equally addressing 2D spatial, 1D viewpoint and 1D temporal information. By regarding the detected 2D skeleton joint candidates in the current frame and the 3D skeleton joints in the former frame as graph nodes, we construct edges by calculating confidence weights between nodes. Such calculation jointly takes advantage of feature confidences in each individual image, epipolar constraints and reconstructed skeletons in the temporal precedent frame. Compared with [14, 24, 8, 7] which adopt sequential processing strategy on image space, viewpoint, and time dimensions, our 4D graph formulation enables unified optimization on all these dimensions, thereby allowing better mutual benefit among them.

To realize realtime optimization on the 4D association graph, we further contribute an efficient method to solve the 4D association by separating the problem into a 4D limb parsing step and a skeleton assembling step. In the former step, we propose a heuristic searching algorithm to form 4D limb bundles and a modified minimum spanning tree algorithm to assemble the 4D limb bundles into skeletons. Both of these two steps are optimized based on an energy function designed to jointly consider the image feature, stereo and temporal cues, thus optimization quality is guaranteed while realtime efficiency is achieved. We demonstrate a realtime multi-person motion capture system using only 5 to 6 multiview video cameras, see Fig. 1 and the supplemental video. Benefiting from this unified strategy, our system succeeds even in the close interaction scenarios (Video 02:55-03:30). Finally, we contribute a multiview multi-person close interacting motion dataset synchronized with marker-based motion capture system.

2. Related Work

We briefly overview literature on multi-person skeleton estimation according to the dimension of input data.

2.1. Single Image Parsing

We restrict our single image parsing to the work that addresses multi-person pose estimation in 2D and 3D. As there are close interactions in the scene, they all need to consider skeleton joint or body part detection and their connection to form skeletons. Parsing methods can be typically categorized into two classes: bottom-up method and top-down method. In general, top-down methods [26, 17, 12, 18, 43, 28] demonstrate higher average precision benefiting from human instance information, and bottom-up methods [20, 11, 35, 27, 38] tend to propose pixel-aligned low-level feature positions while assembling them is still a great challenge. Typically, a state-of-the-art bottom-up method, OpenPose [11], introduces part affinity field (PAF) to assist parsing low-level keypoints on limbs, obtaining realtime performance with high accuracy.

2.2. Cross-view Matching

Matching finds correspondences across views, no matter on high level features (human instances) or low-level features (keypoints). Previous work [6, 8, 7, 16] implicitly solves matching and parsing using 3D pictorial structure model. However, such method is time-consuming due to large state space and iterative belief propagation. Joo *et al.* [24] utilize detected features from dense multi-view images to vote for possible 3D joint positions, which does matching in another implicit way. Such voting method only works well with enough observation views. Most recent work [14] matches per-view parsed human instances cross view with convex optimization method constrained by cycle-consistency. Though fast and robust, such method relies on appearance information to ensure good results, and could be affected by possible parsing error (*e.g.* false positive human instance and wrong joint estimation).

2.3. Temporal Tracking

Tracking is one key step towards continuous and smooth motion capture, and helps solve current pose ambiguity according to history results. Tracking could be done either in 2D space or 3D space. Many works have addressed 2D tracking, known as pose-tracking tasks [3, 37, 22, 19]. For 3D tracking, motion capture of multiple closely interacting persons [31, 30] has been proposed through joint 3D template tracking and multi-view body segmentation. Li *et al.* [29] propose a spatio-temporal tracking for closely interacting persons from multi-view videos. However, these pure tracking algorithms are easy to fail because of temporal error accumulation. Elhayek *et al.* [15] track 3D articulated model to 2D human appearance descriptor (Sum of

Gaussian), achieving markerless motion capture for both indoor and outdoor scenes. However, it does not demonstrate multi-person case (more than 3 persons). Belagiannis *et al*. [8] also utilize tracking information, but they derive human tracks in advance as prior to reduce state space, instead of solving tracking and matching simultaneously. Bridgeman *et al*. [10] contribute a real time method, yet it adopt a sequential processing of image parsing, cross-view correction and temporal tracking. In Panoptic Studio [24], after temporal tracking of 3D joint proposals on the whole sequence, optimization is started for human assembling.

3. Overview

Our 4D association graph considers the information in two consecutive frames. We first use the off-the-shelf bottom-up human pose detector [11] on each input view of the current frame to generate low-level human features on each view. Our 4D association graph takes as input multi-view human body part candidates (2D heatmaps position) and connection confidence (PAF [11] score ranging between 0 and 1) between body parts (see Fig. 2(a)), together with the former reconstructed 3D skeletons. By regarding body parts and the 3D joints in the former frame as graph nodes, we construct edges with significant semantic meaning between nodes. Specifically, as shown in Fig. 2(b), there exist three kinds of edges: per-view parsing edges connecting adjacent body parts in each image view, cross-view matching edges connecting the same body part across views, and temporal tracking edges connecting history 3D nodes and 2D candidates. The construction of these edges will be elaborated in Sect. 4.

Based on the input graph in Fig. 2(b), this 4D association problem can be described as a minimum-cost multi-cut problem, *i.e*., a 0-1 integer programming problem to select those edges that belong to the real skeletons and the physically real temporal and cross-view edges, see Fig. 2(c). Actually, our graph model is similar to the available single view association problem [11, 20], except that it is more complex. As it is a NP-hard problem, we split it to 4D limb parsing (Sect. 5.1) and a skeleton assembling (Sect. 5.2) problems. Our proposed solving method can guarantee realtime performance while obtaining robust results. Here, it is worth mentioning that, our graph model and the solving method also work for special cases when there is no temporal edges, *i.e*., at the first frame of the whole sequence, or when new persons entering the scene.

4. 4D Association Graph

For each image view $c \in \{1, 2, ..., N\}$ at the current frame t, the convolutional pose machine (CPM) model [41, 11] is first applied to get the heatmaps of keypoints and their part affinity fields (PAFs). Denote $\mathcal{D}_j(c) = \{\mathbf{d}_j^m(c) \in \mathbb{R}^2\}$ as the candidate positions of the skeleton joints $j \in \{1, 2, ..., J\}$, with m as candidate index. Here, t is ignored by default as processing the current frame. Denote $f_{ij}^{mn}(c)$ as PAF score connecting $\mathbf{d}_i^m(c)$ and $\mathbf{d}_j^n(c)$, where $\{ij\} \in \mathcal{T}$ is a limb on the skeleton topology tree $\mathcal{T}$.

With both the candidate positions $\mathcal{D}_j(c)$ and the skeleton joints reconstructed in former frame seen as graph nodes, we have three kinds of edges: per-view parsing edges $\mathcal{E}_P$ connecting nodes in the same view, cross-view matching edges $\mathcal{E}_V$ connecting nodes in different views geometrically, and temporal tracking edges $\mathcal{E}_T$ connecting nodes temporally. The solving of this association graph is equivalent to determining bool variable $z \in \{0, 1\}$ for each edge, where $z = 1$ means connected nodes are associated in the same human body, $z = 0$ otherwise. Note that $z = 0$ means the two nodes are linked with two different bodies, or are linked with a false position (a fake joint that is not on a real body). The connecting weight on edges is written as $p(z) = p(z = 1)$. In the following, the weights of each edge is defined in the 4D association graph.

4.1. Parsing Edges and Matching Edges

Without considering the temporal tracking edges introduced by the former reconstructed 3D skeletons, the parsing edges and the matching edges forms a 3D association graph $\mathcal{G}_{3D}$. This case happens when processing the first frame of the whole sequence or when a new person is entering in the scene. The graph $\mathcal{G}_{3D}$ directly extends the original per-view multiple person parsing problem [11] with cross view geometric matching constraints. With these geometric constraints, false limb connections in single view case may have good chance to be distinguished and corrected during joint 3D association.

Denote $z_{ij}^{mn}(c_1, c_2)$ as bool variable on edge connecting $\mathbf{d}_i^m(c_1)$ and $\mathbf{d}_j^n(c_2)$. Obviously, a feasible solution $\{z_{ij}^{mn}(c_1, c_2)\}$ on $\mathcal{G}_{3D}$ must conforms to the following inequalities

$$\begin{aligned} \forall m, \sum_n z_{ij}^{mn}(c, c) \leq 1 \\ \forall c_2 \neq c_1, m, \sum_n z_{ii}^{mn}(c_1, c_2) \leq 1 \end{aligned} \tag{1}$$

Specifically, the top one forces that no two edges share a node, i.e., no two limbs of the same type (e.g., left forearm) share a part. The bottom one forces that no joint from one view connects to two joints of the same type from another view. Note also here c_1 and c_2 represent all possible combinations of view pairs.

For the per-view parsing edge $\mathcal{E}_P$, we directly define the input edge weight as its PAF score:

$$p(z_{ij}^{mn}(c) = 1) = f_{ij}^{mn}(c) \tag{2}$$

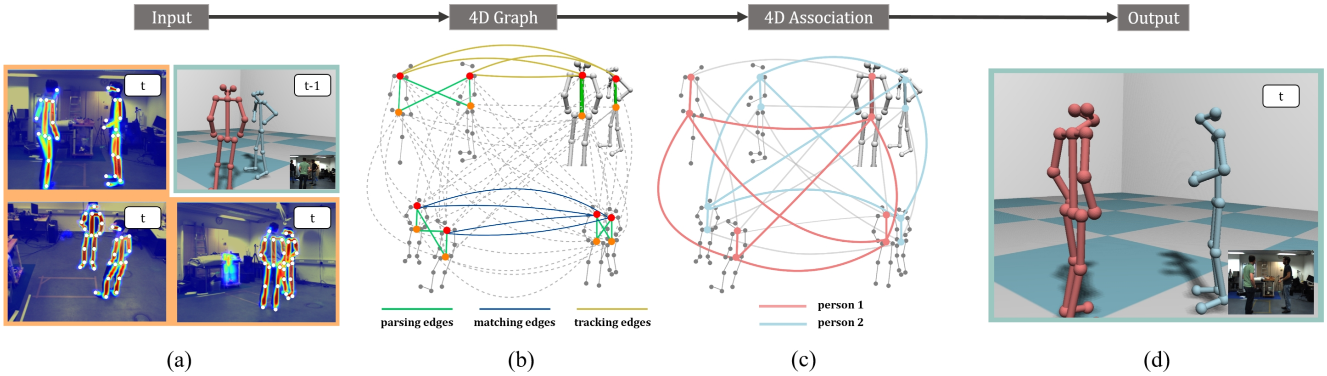


Figure 2. Method overview. (a) We input body part positions and connection confidence of different views at time t, together with 3D person of last time. We use 3 views for example. (b) The 4D association graph. For clarity, we only highlight the association of the torso limb with three types of edges (**parsing** edges, **matching** edges and **tracking** edges) with different colors. (c) From the initial graph (b), our association method outputs the assembling results. (d) We optimize the assembled multiview 2D skeletons (c) to form 3D skeletons of current frame t.

For cross-view matching edge $\mathcal{E}_V$, the weight is defined based on the epipolar distance, written as line-to-line distance in 3D space:

$$p(z_{ii}^{mn}(c_1, c_2)) = 1 - \frac{1}{Z}\mathbf{d}_i^m(c_1) \oplus \mathbf{d}_i^n(c_2) \tag{3}$$

$$\mathbf{d}(c_1) \oplus \mathbf{d}(c_2) = d(K_{c_1}^{-1}\tilde{\mathbf{d}}(c_1), K_{c_2}^{-1}\tilde{\mathbf{d}}(c_2)) \tag{4}$$

where $\tilde{\mathbf{d}} = [\mathbf{d}^{\mathrm{T}}, 1]^{\mathrm{T}}$, K_c is intrinsic matrix of view c, $d(\cdot,\cdot)$ means line-to-line distance between two rays emitting from the camera centers of view c_1 and c_2. Z is an empirically defined normalization factor, which adjusts epipolar distance to range $[0, 1]$. Note that we only build edges for those cross-view nodes sharing the same joint index.

4.2. Tracking Edges

Although solving $\mathcal{G}_{3D}$ at each time instant could provide good association in most cases, failures may happen for very crowded scene or severe occlusions. To improve skeleton reconstruction robustness, we take advantage of the temporal prior, i.e., the reconstructed skeletons at the former frame for regularization of the association problem, which forms the **4D association graph** $\mathcal{G}_{4D}$. We restrict the connecting edge between the former frame skeletons and the current frame joint features, by requiring the two nodes of the edge to be the same skeleton joint (can be on different persons). Denote $z_i^{mk}(c)$ as the final optimized bool variable for edge connecting image joint feature $\mathbf{d}_i^m(c)$ and skeleton joint $\mathbf{X}_i^k$. We define tracking edge connecting probability as

$$p(z_i^{mk}(c)) = 1 - \frac{1}{T}d'(\mathbf{X}_i^k, K_c^{-1}\mathbf{d}_i^m(c)) \tag{5}$$

where $d'(\mathbf{X}, \mathbf{d})$ indicates point-to-line distance between 3D point $\mathbf{X}$ and 3D line emitting from camera center to $\mathbf{d}$, and T is normalization factor, ensuring $p(z_i^{mk}(c))$ to be in range $[0, 1]$. Similarly, we have inequality conditions hold for the feasible solution space:

$$\forall i, c, \sum_m z_i^{mk}(c) \leq 1, \sum_k z_i^{mk}(c) \leq 1 \tag{6}$$

This constraint forces that each 3D joint at the last frame matches no more than one 2D joint on each view at the current frame, and vice versa.

4.3. Objective Function

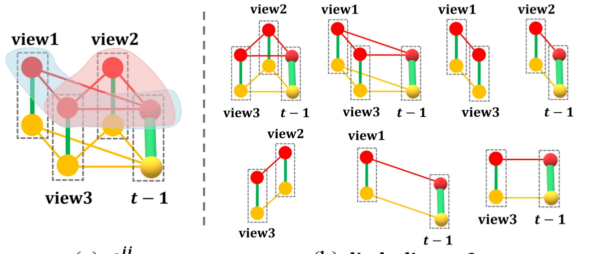
Based on the predefined probabilities for the parsing edges, matching edges and tracking edges, our 4D association optimization can be formulated as an edge selection problem to maximize an objective function under conditions 1 and 6. Specifically, let $q(z) = p(z) \cdot z$ denote the final energy of an edge, where z is a boolean variable, and then our objective function can be written as the summation of energies of all the selected edges in $\mathcal{E}_P$, $\mathcal{E}_M$ and $\mathcal{E}_T$:

$$\begin{aligned} E(\mathcal{Z}) =& w_p \sum q(z_{ij}^{mn}(c,c)) + w_m \sum q(z_{ii}^{mn}(c_1, c_2)) \\ &+ w_t \sum q(z_i^{mk}(c)) \end{aligned} \tag{7}$$

Note here $\sum$ would traverse all the possible edges, i.e., all feasible values of variables (i,j,m,n,k,c,$c1$,$c2$) by default. w_p, w_m and w_t are empirically defined weighting factors for edges $\mathcal{E}_P$, $\mathcal{E}_M$ and $\mathcal{E}_T$, respectively. With $w_t = 0$, it degenerates to the objective function for solving association graph $\mathcal{G}_{3D}$. Notice that, both $\mathcal{G}_{3D}$ and $\mathcal{G}_{4D}$ can be solved with the same procedure, as described in Sect. 5.

5. Solving 4D Association

Solving the 4D Association graph means maximizing the objective function Eqn. 7 under constraints Eqn. 1 and



(a) $\mathcal{G}_{4D}^{ij}$ (b) **limb cliques** $\mathcal{G}_C$

Figure 3. Illustration of limb cliques. (a) A sample 4D graph on limb $\{ij\}$ denoted as $\mathcal{G}_{4D}^{ij}$. Two cliques are marked as red area and blue area. (b) Limb cliques of different sizes could be proposed from the 4D graph on limb. Joints of the same type (same color in the above figure) on a limb clique form a clique, and joints of different types on each view must share a green parsing edge.

Eqn. 6. Traversing the huge association space in a brute force manner is infeasible for realtime systems. Instead, inspired by the realtime but high quality parsing method [11] that assembles 2D human skeleton in a greedy manner, we propose a realtime 4D association solver. The key difference between our 4D association and the previous 2D association is that: the limb candidates scatter not only in a single image but in the whole space and time, and some limbs represent the same physical limbs. Therefore, we need to first associate those limbs that are likely to be the same limb bundle across views and times, before 4D skeletons assembling. Based on this idea, our realtime solution can be divided into two steps: 4D limb bundle association (Sect. 5.1), and 4D human skeleton association by the bundle Kruskal's algorithm (Sect. 5.2). It is worth noting that, both of these two steps rely on the objective function Eqn. 7 for optimization.

5.1. 4D Limb Bundle Parsing

To extract limb bundles across view and time, we first restrict $\mathcal{G}_{4D}$ on a limb $\{ij\}$ (two adjacent types of joint) as $\mathcal{G}_{4D}^{ij}$. Since there are multiple persons in the scene, graph $\mathcal{G}_{4D}^{ij}$ may contain multiple real limb bundles. In theory, each real limb bundle contains two joint cliques. For clarity, a clique means a graph where every two nodes are connected [42], see Fig. 3(a) for example. This implies that every two joints of the same type in the limb bundle must share a cross-view edge or a temporal edge. By further considering the parsing edges, a correct 4D limb bundle consists of two joint clique connected with parsing edges on each view. We call such limb bundle candidate as *limb clique*. Fig. 3(b) enumerates all the possible limb cliques of Fig. 3(a). Consequently, our goal in this step is to search all possible limb cliques $\{\mathcal{G}_C | \mathcal{G}_C \subset \mathcal{G}_{4D}^{ij}\}$ for the real limb bundles.

We measure each limb clique with $E(\mathcal{Z}_{\mathcal{G}_C})$ based on the objective function Eqn. 7. However, directly maximizing $E(\mathcal{Z}_{\mathcal{G}_C})$ would always encourage as many edges as possible to be selected in a clique, even false edges. Hence, we normalize $E(\mathcal{Z}_{\mathcal{G}_C})$ with clique size $|\mathcal{V}_C|$ of $\mathcal{G}_C$, and add a penalty term to balance the clique size and the average probability. Overall, the objective function for a limb clique is

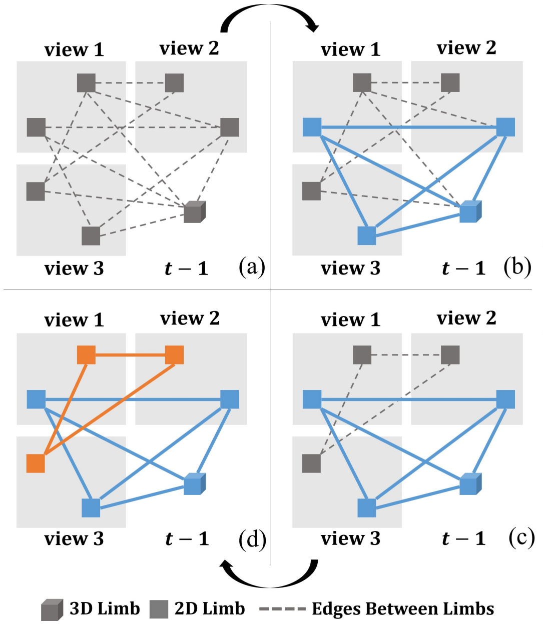


Figure 4. Illustration of limb bundle parsing procedure. (a) Initial graph $\mathcal{G}_{4D}^{ij}$. A square/cube represents a limb (2D or 3D), and each grey dash line means an edge. (b) A best clique (limb bundle) detected from (a) is shown in blue. (c) Then, we remove both limbs and edges related to the best clique, and extract next best one. (d) Finally, all cliques are detected. We could extract cliques without temporal edges, like the orange one.

$$E(\mathcal{G}_C) = E(\mathcal{Z}_{\mathcal{G}_C})/|\mathcal{V}_C| + w_v \rho(|\mathcal{V}_C|) \tag{8}$$

where w_v is balancing weight, and ρ is a Welsch robust loss[13, 5] defined as

$$\rho(x) = 1 - \exp\left(-\frac{1}{2}(x/c)^2\right) \tag{9}$$

Here, $c = (N-1)/2$ is a parameter depending on the total number of views.

Fig. 4 illustrates the limb bundle parsing procedure. After selecting a limb clique and marking it as a limb bundle, we remove it from $\mathcal{G}_{4D}^{ij}$ (Fig. 4(b)), together with all other edges connected with any joint in this clique (Fig. 4(c)). By doing this, our solution always conforms to feasibility inequalities (1,6). This selection process is iterated until $\mathcal{G}_{4D}^{ij}$ is empty (Fig. 4(d)).

5.2. 4D Skeleton Assembling

After generating all the 4D limb bundles, we need to assemble them into multiple 4D human skeletal structures. We first sort all the 4D limb bundles based on their scores, and build a priority queue to store them. In each iteration, we pop a 4D limb bundle from the queue with the maximum score (based on Eqn. 8), and merge it into the 4D skeletons. In this merging process, all the 2D joints (belongs to this bundle, from different views) should have a same labeled person ID. However, since a newly added limb bundle may share the same 4D joint as some limb bundles that are already assigned, conflicts would arise when these 2D joints have already been labeled with different person IDs on different views in the previous iterations, see Fig. 5(a). To eliminate this conflict, we propose a simple yet effective way by splitting the newly added limb bundles to small limb bundles according to the persons whose joints are assigned to (Fig. 5(b)). We then re-compute the objective function of each small bundle and push back to the prior queue for further assembling. If there is no conflict, we merge the bundle into the skeleton and label the 2D joints. We iterate popping and merging until the queue is empty (Fig. 5(c)).

We call the above method bundle Kruskal's algorithm. In the single view case, there would be no conflicts, and our method degenerates to traditional Kruskal's algorithm, which is a famous minimum spanning tree (MST) algorithm used in OpenPose [11].

5.3. Parametric Optimization

Based on 4D skeleton assembling results on the 2D view images, we can further optimize the full 3D body pose by embedding a parametric skeleton. We minimize the energy function

$$E(\Theta) = w_{2D}E_{2D} + w_{shape}E_{shape} + w_{temp}E_{temp} \quad (10)$$

where E_{2D} is the data term aligning 2D projections on each view to the detected joints, E_{shape} penalizes human shape prior (*e.g.* bone length and symmetry), and E_{temp} is temporal smoothing term (w_{2D}, w_{shape} and w_{temp} are balancing weights, $w_{temp} = 0$ if no temporal information exists). As this fitting process is a classic optimization step, please refer to [9, 44, 29] for details. Temporally, we track each person and use the average bone lengths of the first five frames with high confidence (visible in more than 3 cameras) as the bone length prior for the person in the later frames. If the person is lost and re-appear, we simply regard him/her as a new person and re-calculate the bone lengths.

6. Results

In Fig. 6, we demonstrate the results of our system. Using only geometry information from sparse view points, our method enables realtime and robust multi-person motion capture under severe occlusions (Fig. 6(a)), challenging poses (Fig. 6(b)) and subtle social interactions (Fig. 6(c)).

6.1. Implementation Details

The multi-view capture system consists of 5 synchronized industrial RGB cameras (with resolution 2048×2048) and a single PC with one 3.20 GHz CPU and one NVIDIA TITAN RTX GPU. Our system achieves 30 fps motion capture for 5 persons. Specifically, for each frame, the preprocessing step (including demosaicing, undistortion and resizing for multi-view inputs) takes less than 1 ms, the CNN inference step takes 22.9 ms in total for 5 images, the 4D association step takes 11 ms, and the parametric optimization step takes less than 4 ms. Moreover, we ping-pong the CNN inference and the 4D association for achieving realtime performance with affordable delay (60 ms). More details about the optimization parameters are provided in the supplementary material.

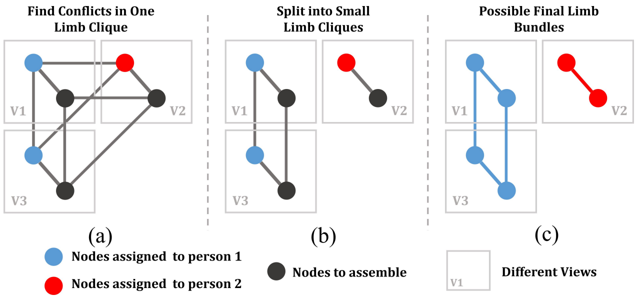
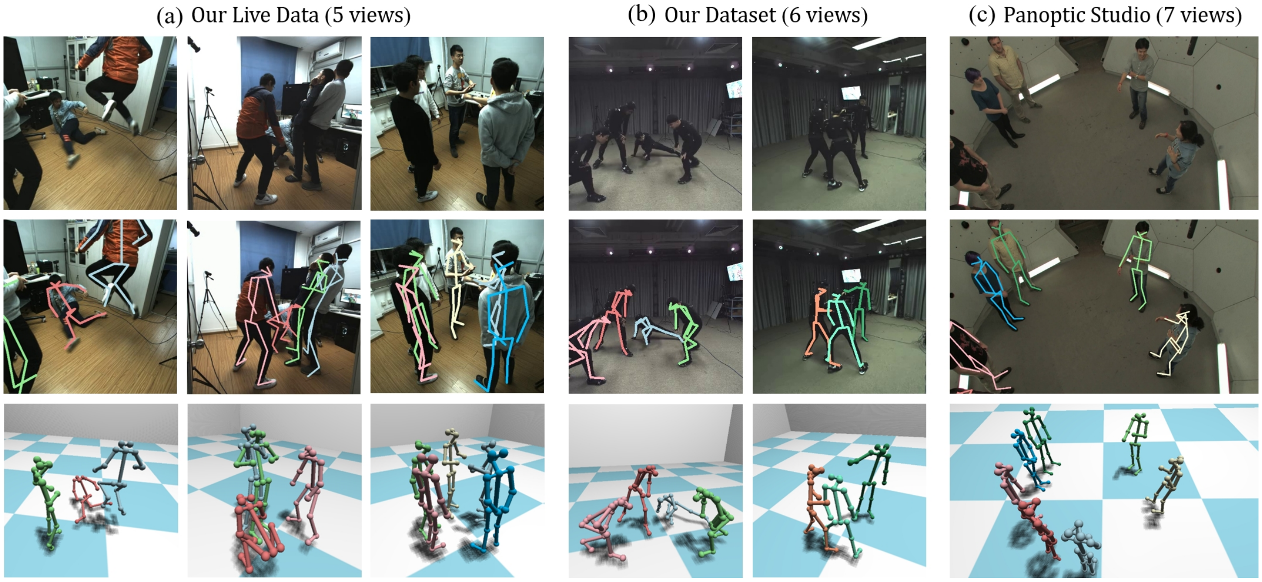

Find Conflicts in One Limb Clique
Split into Small Limb Cliques
Possible Final Limb Bundles
V1
V2
V3
V1
V2
V3
V1
V2
V3
(a)
(b)
(c)
Nodes assigned to person 1
Nodes assigned to person 2
Nodes to assemble
V1
Different Views

Figure 5. Conflicts handling in our skeleton assembling step. (a) A limb bundle to be added. It contains 3 parsing edges on 3 views. In this case, each parsing edge contains a joint to be assembled (black node) and a joint already assembled (blue or red nodes) in previous iterations. Here conflict arises as blue and red belong to different person IDs. (b) We split original limb bundle into small bundles according to the existing person IDs. (c) A possible final assembling result.

Note that the 4D association pipeline is fully implemented on CPU. Also, in the CNN inference step, the input RGB images are resized to 368 × 368, and the CNNs for keypoints and PAFs are re-implemented using TensorRT [40] for further acceleration.

6.2. Dataset

We contribute a new evaluation dataset for multi-person 3D skeleton tracking with ground truth 3D skeletons captured by commercial motion capture system, OptiTrack [1]. Compared with previous 3D human datasets [25, 21, 32, 24, 8, 2], our dataset is mainly focusing on the more challenging scenarios like close interactions and challenging motion. Our dataset contains 5 sequences with each around 20-second long capturing a 2-4 person scene using 6 cameras. Our actors all wear black marker-suit for ground truth skeletal motion capture. With ground truth 3D skeletons, our dataset enables more effective quantitative evaluations for both 2D parsing and 3D tracking algorithms. Note that besides evaluating our method using the proposed dataset, we also provide evaluation results using Shelf and Panoptic Studio dataset following previous works [8, 7, 14].



(a) Our Live Data (5 views)

(b) Our Dataset (6 views)

(c) Panoptic Studio (7 views)

Figure 6. Results of our system. From top to bottom: input images, reprojection of 3D human, and 3D visualization respectively. (a) Our live captured data with fast motion (left), severe occlusion (middle) and crowded scene (right). 5 views used. (b) Our dataset with textureless clothing and rich motion. 6 views used. (c) Panoptic studio dataset with natural social interaction. 7 views used.

6.3. Quantitative Comparison

We compare with state-of-the-art methods quantitatively using both the Shelf dataset and our testing dataset. The quantitative comparison on Shelf dataset is shown in Table. 1. Benefiting from our 4D association formulation, we achieve more accurate results than both temporal tracking methods based on 3DPS ([8, 6, 7, 16]) and appearance-based global optimization methods [14].

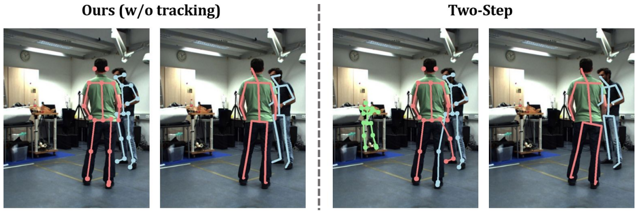
We also compare with [14] on our testing dataset according to 'precision' (the ratio of correct joints in all estimated joints) and 'recall' (the ratio of correct joints in all ground truth joints). A joint is correct if its Euclidean distance to the ground truth joint is less than threshold $0.2m$. As shown in Tab. 2, our method outperforms [14] under both metrics.

Shelf	A1	A2	A3	Avg
Belagiannis *et al.* [6]	66.1	65.0	83.2	71.4
†Belagiannis *et al.* [8]	75.0	67.0	86.0	76.0
Belagiannis *et al.* [7]	75.3	69.7	87.6	77.5
Ershadi-Nasab *et al.* [16]	93.3	75.9	94.8	88.0
Dong *et al.* [14]	97.2	79.5	96.5	91.1
*Dong *et al.* [14]	98.8	94.1	**97.8**	96.9
†# Tanke *et al.* [39]	**99.8**	90.0	98.0	96.0
†Ours(final)	99.0	**96.2**	97.6	**97.6**

Table 1. Quantitative comparison on Shelf dataset using percentage of correct parts (PCP) metric. '*' means method with appearance information, '†' means method with temporal information, '#' means accuracy without head. 'A1'-'A3' correspond to the results of three actors, respectively. The averaged result is in column 'Avg'.

Our Dataset	Dong[14]	Ours(final)
Precision(%)	71.0	**88.5**
Recall(%)	80.2	**90.2**

Table 2. Comparison with [14] using our testing dataset.



Ours (w/o tracking)

Two-Step

Figure 7. Comparison with two-step pipeline. Top figures are association result, bottom figures are reprojection of 3D pose. Notice that, reprojection of 3D pose generated by two-step pipeline obviously deviates from correct position due to false parsing.

6.4. Qualitative Comparison

To further demonstrate the advantages of our bottom-up system, we perform qualitative comparison with the state-of-the-art method [14], which utilizes top-down human pose detector [12] to perform single view parsing. The qualitative results is shown in Fig. 8, from which we can see that top-down method depends heavily on instance proposals, and may generate false positive human pose detection to deteriorate the cross-view matching performance (left case). Furthermore, per-view parsing would fail to infer correct human poses under severe occlusion, deteriorating pose reconstruction results (right). Instead, thanks to relatively precise low-level features (e.g. keypoints) and robust 4D asso-

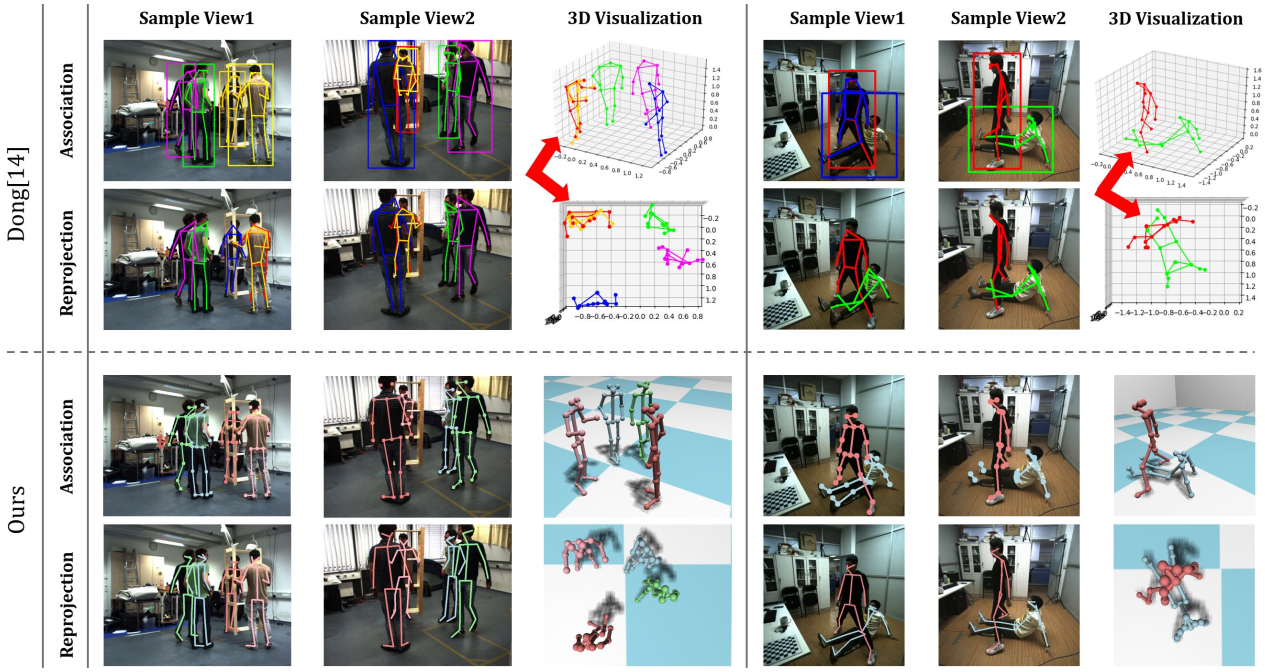


Figure 8. Qualitative comparison with Dong[14] on Shelf (left figure) and our captured data (right figure), both with 5 cameras. For each case, we show association results and reprojection of 3D pose on two sample views. For 3D visualization, we show a side view rendering and a top view rendering for clear comparison.

ciation algorithm, the joints are associated more accurately in our results.

Shelf	A1	A2	A3	Avg
two-step	98.1	83.8	**97.6**	93.1
w/o tracking	96.5	86.8	97.0	93.4
Ours(final)	**99.0**	**96.2**	**97.6**	**97.6**

Table 3. Ablation study on Shelf dataset. 'two-step' means first per-view parsing and then cross-view matching. 'w/o tracking' means we solve $\mathcal{G}_{3D}$ in each frame. Both 'two-step' and 'w/o tracking' use triangulation to infer 3D poses. Numbers are percentage of correct parts(PCP).

6.5. Ablation Study

With/Without tracking. We first evaluate tracking edges in the 4D graph. By triangulating 2D bodies into 3D skeletons directly using $\mathcal{G}_{3D}$, we eliminate the usage of tracking edges. The result is labeled as 'w/o tracking' in Table. 3. Without using tracking edges, our method still exhibits competent result and out-performs state-of-the-art method [14] (93.4% vs 91.1%). Moreover, our 4D association method is more robust in messy scenes ('Ours(final)' as shown in Table. 3).

Compare with two-stage pipeline. We implement a two-step pipeline for comparison, by using [11] to parse human in each view, followed with human matching using clique searching method with objective function defined on the parsed bodies. Note that no temporal information is used, and 3D poses are obtained by triangulation. Result is shown as 'two-step' in Table. 3. As shown in Table. 3, our per-frame $\mathcal{G}_{3D}$ solution 'w/o tracking' performs better than two-step pipeline, especially on actor 'A2'. To show our robustness to per-view parsing ambiguity, we use only 3 views to reconstruct 2 persons (Fig. 7). Wrong parsing result on one view would harm the inferred 3D pose, especially when very sparse views are available.

7. Conclusion

We proposed a realtime multi-person motion capture method with sparse view points. Build on top of the low-level detected features directly, we formulated parsing, matching and tracking problem simultaneously into a unified 4D graph association framework. The new 4D association formulation not only enabled realtime motion capture performance, but also achieved state-of-the-art accuracy, especially for crowded and close interaction scenarios. Moreover, we contributed a new testing dataset for multi-person motion capture with ground truth 3D poses. Our system narrowed the gap between laboratory markerless motion capture system and industrial applications in real world scenarios. Finally, our novel 4D graph formulation may stimulate future research in this topic.

Acknowledgements. This paper is supported by the National Key Research and Development Program of China [2018YFB2100500] and the NSFC No.61531014 and No.61861166002.

References

[1] Optitrack marker mocap. `https://www.optitrack.com`.

[2] Nvd 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 *ICCV Workshop HICV*, 2011.

[3] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018.

[4] Praneet C Bala, Benjamin R Eisenreich, Seng Bum Michael Yoo, Benjamin Y Hayden, Hyun Soo Park, and Jan Zimmermann. Openmonkeystudio: Automated markerless pose estimation in freely moving macaques. *bioRxiv*, 2020.

[5] Jonathan T Barron. A general and adaptive robust loss function. In *CVPR*, 2019.

[6] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. In *CVPR*, 2014.

[7] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures revisited: Multiple human pose estimation. *TPAMI*, 2016.

[8] Vasileios Belagiannis, Xinchao Wang, Bernt Schiele, Pascal Fua, Slobodan Ilic, and Nassir Navab. Multiple human pose estimation with temporally consistent 3d pictorial structures. In *ECCV Workshop*, 2014.

[9] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, 2016.

[10] Lewis Bridgeman, Marco Volino, Jean-Yves Guillemaut, and Adrian Hilton. Multi-person 3d pose estimation and tracking in sports. In *CVPR Workshop*, 2019.

[11] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2d pose estimation using part affinity fields. *TPAMI*, 2019.

[12] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded Pyramid Network for Multi-Person Pose Estimation. In *CVPR*, 2018.

[13] John E Dennis Jr and Roy E Welsch. Techniques for nonlinear least squares and robust regression. *Communications in Statistics-Simulation and Computation*, 1978.

[14] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *CVPR*, 2019.

[15] Ahmed Elhayek, Edilson de Aguiar, Arjun Jain, J Thompson, Leonid Pishchulin, Mykhaylo Andriluka, Christoph Bregler, Bernt Schiele, and Christian Theobalt. Marconiconvnet-based marker-less motion capture in outdoor and indoor scenes. *TPAMI*, 2017.

[16] Sara Ershadi-Nasab, Erfan Noury, Shohreh Kasaei, and Esmaeil Sanaei. Multiple human 3d pose estimation from multiview images. *Multimedia Tools and Applications*, 2018.

[17] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[19] Eldar Insafutdinov, Mykhaylo Andriluka, Leonid Pishchulin, Siyu Tang, Evgeny Levinkov, Bjoern Andres, and Bernt Schiele. Arttrack: Articulated multi-person tracking in the wild. In *CVPR*, 2017.

[20] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016.

[21] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 2013.

[22] Umar Iqbal, Anton Milan, and Juergen Gall. Posetrack: Joint multi-person pose estimation and tracking. In *CVPR*, 2017.

[23] Engin Turetken Pascal Fua Jerome Berclaz, Francois Fleuret. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 2011.

[24] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, et al. Panoptic studio: A massively multiview system for social interaction capture. *TPAMI*, 2019.

[25] Vahid Kazemi, Magnus Burenius, Hossein Azizpour, and Josephine Sullivan. Multi-view body part recognition with random forests. In *BMVC*, 2013.

[26] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. In *ECCV*, 2018.

[27] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *ECCV*, 2018.

[28] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *CVPR*, 2019.

[29] Kun Li, Nianhong Jiao, Yebin Liu, Yangang Wang, and Jingyu Yang. Shape and pose estimation for closely interacting persons using multi-view images. In *CGF*, 2018.

[30] Yebin Liu, Juergen Gall, Carsten Stoll, Qionghai Dai, Hans-Peter Seidel, and Christian Theobalt. Markerless motion capture of multiple characters using multiview image segmentation. *TPAMI*, 2013.

[31] Yebin Liu, Carsten Stoll, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *CVPR*, 2011.

[32] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, 2017.

[33] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *3DV*, 2018.

[34] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *ICCV*, 2019.

[35] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *ECCV*, 2018.

[36] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016.

[37] Yaadhav Raaj, Haroon Idrees, Gines Hidalgo, and Yaser Sheikh. Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields. In *CVPR*, 2019.

[38] Jie Song, Bjoern Andres, Michael Black, Otmar Hilliges, and Siyu Tang. End-to-end learning for graph decomposition. In *ICCV*, 2019.

[39] Julian Tanke and Juergen Gall. Iterative greedy matching for 3d human pose tracking from multiple views. In *GCPR*, 2019.

[40] Han Vanholder. Efficient inference with tensorrt, 2016.

[41] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[42] Robin J. Wilson. Introduction to graph theory, fourth edition, 1996.

[43] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018.

[44] Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. In *CVPR*, 2018.

[45] Andrei Zanfir, Elisabeta Marinoiu, Mihai Zanfir, Alin-Ionut Popa, and Cristian Sminchisescu. Deep network for the integrated 3d sensing of multiple people in natural images. In *NIPS*, 2018.