Cross-View Tracking for Multi-Human 3D Pose Estimation at over 100 FPS

Long Chen[1] Haizhou Ai[1] Rui Chen[1] Zijie Zhuang[1] Shuang Liu[2]
[1]Department of Computer Science and Technology, Tsinghua University [2]AiFi Inc.

Abstract

Estimating 3D poses of multiple humans in real-time is a classic but still challenging task in computer vision. Its major difficulty lies in the ambiguity in cross-view association of 2D poses and the huge state space when there are multiple people in multiple views. In this paper, we present a novel solution for multi-human 3D pose estimation from multiple calibrated camera views. It takes 2D poses in different camera coordinates as inputs and aims for the accurate 3D poses in the global coordinate. Unlike previous methods that associate 2D poses among all pairs of views from scratch at every frame, we exploit the temporal consistency in videos to match the 2D inputs with 3D poses directly in 3-space. More specifically, we propose to retain the 3D pose for each person and update them iteratively via the cross-view multi-human tracking. This novel formulation improves both accuracy and efficiency, as we demonstrated on widely-used public datasets. To further verify the scalability of our method, we propose a new large-scale multi-human dataset with 12 to 28 camera views. Without bells and whistles, our solution achieves 154 FPS on 12 cameras and 34 FPS on 28 cameras, indicating its ability to handle large-scale real-world applications. The proposed dataset will be released at https://github.com/longcw/crossview_3d_pose_tracking.

1. Introduction

Multi-human 3D pose estimation from videos has a wide range of applications, including action recognition, sports analysis, and human-computer interaction. With the rapid development of deep neural network, most of the recent efforts in this area have been devoted to monocular 3D pose estimation [25, 26] . However, despite much progress, the single-camera setting is still far from being resolved due to the large variations of human poses and partial occlusion in the monocular views. A natural solution for these problems is to recover the 3D poses from multiple camera views.

Recent multi-view approaches generally employ the detected 2D body joints from multiple views as inputs with the advance of 2D human pose estimation [9, 11, 35], and address the 3D pose estimation in a two-step formulation [2, 13]. Specifically, the 2D joints of the same person are first matched and associated across views, the 3D location of each joint is subsequently determined by a multi-view reconstruction method. In this formulation, the challenge comes from three parts: 1) the detected 2D joints are noisy and inaccurate since the pose estimation is imperfect; 2) the cross-view association is ambiguous when multiple people interacting with each other in crowded scenes; 3) the computational complexity explodes as the number of people and number of cameras increase.

To tackle the problem of cross-view association, 3D pictorial structure model (3DPS) is widely used in some previous methods [2, 8], where the 3D poses are recovered from 2D joints in a discretized 3-space. In this formulation, the likelihood of a joint belonging to a spatial bin is given by the geometric consistency [16], along with a pre-defined body structure model. A severe problem of 3DPS is the expensive computational cost due to the huge state space with multiple people in multiple views. As an improvement, Dong *et al*. [13] propose solving the cross-view association problem at the body level in advance before applying 3DPS. They associate 2D poses of the same person from different views as clusters and estimate 3D poses from the clusters via 3DPS. Nevertheless, matching 2D poses between all pairs of views still makes the computational complexity explode as the number of cameras increases.

In contrast to previous methods that process inputs from multiple cameras simultaneously, we propose a new solution with an iterative processing strategy. Specifically, we propose exploiting the temporal consistency in videos to match 2D poses of each view with 3D poses directly in 3-space, where the 3D poses are retained and updated iteratively by the cross-view multi-human tracking. There are two advantages in our formulation. Firstly, for the accuracy, matching in 3-space is expected to be robust to partial occlusion and inaccurate 2D localization, as the 3D poses consist of multi-view information. Secondly, for the efficiency, processing camera views iteratively makes the

computational complexity varies only linearly as the number of cameras changes, enabling the applications on large-scale camera systems. To verify the effectiveness, we compare our method with state-of-the-art approaches on several widely-used public datasets, and moreover, we test it on a self-collected dataset with more than 12 cameras, as shown in Figure 1. With the proposed solution, we are able to estimate 3D poses accurately in 12 cameras at over 100 FPS.

Below, we review related work in multi-human 3D pose estimation and multi-view tracking, and then we present the details of our new approach, which contains an efficient geometric affinity measurement for tracking in 3-space, along with a novel 3D reconstruction algorithm that designed for iterative processing in videos. In the experimental section, we perform the evaluation on three public datasets: Campus [2], Shelf [2], and CMU Panoptic [18], demonstrating both state-of-the-art accuracy and efficiency of our method. We also propose a new dataset that collected from large-scale camera systems, to verify the scalability of our method for real-world applications as the number of cameras increases.

Figure 1: Multi-human multi-view 3D pose estimation. The triangles in the 3D view represent camera locations.

2. Related work

Multi-human 3D pose estimation. The problem of 3D human pose estimation has been studied from monocular [26, 1, 21, 25, 12] and multi-view perspectives [8, 4, 13, 32].

Most of the existing monocular solutions are designed for the single-person cases [28, 21, 12], where the estimated poses are relatively centered around the pelvis joint, and the absolute locations in the environment are unknown. Such a relative coordinate setting limits the application of these methods in surveillance scenarios.

To estimate multiple 3D poses from a monocular view, Mehta *et al*. [22] use the location-maps [23] to infer 3D joint positions at the respective 2D joint pixel locations. Moon *et al*. [25] propose a root localization network to estimate the camera-centered coordinates of the human roots. Despite lots of recent progress in this area, the task of monocular 3D pose estimation is inherently ambiguous as multiple 3D poses can map to the same 2D joints. The mapping result, unfortunately, often has a large deviation in practice, especially when occlusion or motion blur occurs in images.

On the other hand, multi-camera systems are becoming progressively available in the context of various applications such as sport analysis and video surveillance. Given images from multiple camera views, most previous methods [27, 29, 8, 2] are generally based on the 3D Pictorial Structure model (3DPS) [8], which discretizes the 3-space by an $N \times N \times N$ grid and assigns each joint to one of the N^3 bins (hypothesis). The cross-view association and reconstruction are solved by minimizing the geometric error [16] between the estimated 3D poses and 2D inputs among all the hypotheses. Considering all joints of multiple people in all cameras simultaneously, these methods are generally computational expensive due to the huge state space. Recent work from Dong *et al*. [13] propose to solve the cross-view association problem at the body level first. 3DPS is subsequently applied to each cluster of the 2D poses of the same person from different views. The state space is therefore reduced as each person is processed individually. Nevertheless, the computational cost of cross-view association of this method is still too high to achieve the real-time speed.

Multi-view tracking for 3D pose estimation. Multi-view tracking for 3D pose estimation is not a new topic in computer vision. However, it is still nontrivial to combine these two tasks for fast and robust multi-human 3D pose estimation, as facing the challenges mentioned above.

Markerless motion capture, aiming at 3D motion capturing for a single person, has been studied for a decade [33, 14, 34]. Tracking in these early works is developed for joint localization and motion estimation. As the recent progress in deep neural network, temporal information is also investigated with the recurrent neural network [30, 20] or convolutional neural network [28] for single-view 3D pose estimation. However, these approaches are generally designed for well-aligned single person cases, where the critical cross-view association problem is neglected.

As for the multi-human case, Belagiannis *et al*. [4] propose employing cross-view tracking results to assist 3D pose estimation under the framework of 3DPS. It introduces the temporal consistency from an off-the-shelf cross-view tracker [5] to reduce the state space of 3DPS. This approach separates tracking and pose estimation into two tasks and

runs at 1 fps, which is far from being applied to the time-critical applications. There is also a very recent tracking approach [7] that uses the estimated 3D poses as inputs of the tracker to improve the tracking quality, while the pose estimation is rarely benefited from the tracking results. Tang *et al*. [32] propose to jointly perform multi-view 2D tracking and pose estimation for 3D scene reconstruction. The 2D detections are associated using a ground plane assumption, which is efficient but limits the accuracy. In contrast, we couple cross-view tracking and multi-human 3D pose estimation in a unified framework, making these two tasks benefit from each other for both accuracy and efficiency.

3. Method

In this section, we first give an overview of our framework with iterative processing, then we detail the two components of our framework, that is, cross-view tracking in 3-space with geometric affinity measurement and incremental 3D pose reconstruction in videos.

3.1. Iterative processing for 3D pose estimation

Given an unknown number of people interacting with each other in the scene covered by multiple calibrated cameras, our approach takes the detected 2D body joints as inputs. We aim at estimating the 3D locations of a fixed set of body joints for each person in the scene. Particularly, our approach differs from previous methods in the way they process frames from different cameras. In contrast to taking all camera views at a time in a batch mode, here we assume each camera streams frames independently, where the frames are collected in chronological order and fed into the framework one-by-one iteratively.

With iterative processing, the overall computational cost increases only linearly as the number of cameras increases, and the strict synchronization between cameras is no longer required, making the solution have the potential to be applied to large-scale camera systems. Such a modification is straightforward, but not that easy to achieve, as the cross-view association is generally ambiguous, especially when only one view is observed at one time. Another challenge, in this case, is to reconstruct 3D poses from different cameras when these cameras are not strictly synchronized.

To solve the problems, we construct our framework from two components: 1) cross-view tracking for body joint association, and 2) incremental 3D pose reconstruction for unsynchronized frames. Given a frame from a particular camera, the task of tracking is to associate the detected 2D human bodies with tracked targets. Here, we represent the targets in 3-space using historically estimated 3D poses. The cross-view association is therefore performed between 2D joints and 3D poses in 3-space, as detailed in Section 3.2. Subsequently, based on the association results, each 2D human body is assigned to a target or labeled as unmatched. The 3D pose of each target is incrementally updated when combining the newly observed and previously retained 2D joints. Since these joints are from different times, conventional reconstruction method such as triangulation [16] is prone to inaccurate 3D locations. To deal with the unsynchronized frames, we present our incremental triangulation algorithm in Section 3.3.

3.2. Cross-view tracking with geometric affinity

In multi-view geometry, reconstructing the location of a point in 3-space requires knowing the 2D locations of the point in at least two views. Thus in our case, in order to estimate the 3D poses, we have to associate the detected 2D joints across views first. Similar to [13], we associate the joints at the body level, but not just across views, also across times. This forms the cross-view tracking problem, as discussed in this section.

Problem statement. We retain historical states of persons in the scene as tracked targets, the problem becomes associating these targets with the newly detected human bodies, while the detections come from a different camera in every iteration. Here, we begin with some notations and definitions. We use $\mathbf{x} \in \mathbb{R}^2$ to represent 2D point in camera coordinate, and $\mathbf{X} \in \mathbb{R}^3$ for 3D point in global coordinate. For a frame from camera c at time t, a detected human body D is denoted as 2D points $\mathbf{x}^k_{t,c}$ of a fixed set of human joints with indices $k \in \{1, ..., K\}$. Meanwhile, a target T is represented in 3-space using points $\mathbf{X}^k_{t'} \in \mathbb{R}^3$ of the same set of human joints, where t' stands for the last updated time of the joint. The historical 2D joints are also retained in the corresponding targets.

Then, supposing there are M detections $\{D_{i,t,c} | i = 1, ..., M\}$ in the new frame, we need to associate these detections to the last N tracked targets $\{T_{i,t'} | i = 1, ..., N\}$, and afterwards update the 3D locations of targets based on the matching results. Technically, this is a weighted bipartite graph matching problem, where the graph is determined by the affinity matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$ between targets and detections. Once the graph is determined, the problem can be solved efficiently with the Hungarian algorithm [19]. Therefore, our major challenge is to measure the affinity of each pair of targets and detections accurately and efficiently.

Affinity measurement. Given a pair of target and detection $\langle T_{t'}, D_{t,c} \rangle$, the affinity is measured from both 2D and 3D geometric correspondences:

$$A(T_{t'}, D_{t,c}) = \sum_{k=1}^{K} A_{2D}(\mathbf{x}^k_{t'',c}, \mathbf{x}^k_{t,c}) + A_{3D}(\mathbf{X}^k_{t'}, \mathbf{x}^k_{t,c}), \tag{1}$$

where $\mathbf{x}^k_{t'',c}$ is the last matched joint k of the target from camera c. For each type of human joints the correspondence is computed independently, thus we omit the index k in the following discussion for notation simplicity.

As shown in Figure 2a, the 2D correspondence is computed based on the distance of detected joint $\mathbf{x}_{t,c}$ and previously retained joint $\mathbf{x}_{t'',c}$ in the camera coordinate:

$$A_{2D}(\mathbf{x}_{t'',c}, \mathbf{x}_{t,c}) = w_{2D}(1 - \frac{\|\mathbf{x}_{t,c} - \mathbf{x}_{t'',c}\|}{\alpha_{2D}(t - t'')}) \cdot e^{-\lambda_a(t-t'')}. \tag{2}$$

There are three hyper-parameters w_{2D}, α_{2D}, and λ_a, standing for the weight of 2D correspondence, threshold of 2D velocity, and the penalty rate of time interval, respectively. Note that $t > t''$ since frames are processed in chronological order. $A_{2D} > 0$ indicates these two joints may come from the same person, and vice versa. The magnitude represents the confidence of the indication, which decreases exponentially as the time interval increases.

2D correspondence is the most basic affinity measurement that exploited by single-view tracking methods. In order to track people across views, a 3D correspondence is introduced, as illustrated in Figure 2b. We suppose that cameras are well calibrated and the projection matrix of camera c is provided as $\mathrm{P}_c \in \mathbb{R}^{3\times4}$. We first back-project the detected 2D point $\mathbf{x}_{t,c}$ into 3-space as a ray:

$$\tilde{\mathbf{X}}_t(\mu; \mathbf{x}_{t,c}) = \mathrm{P}_c^{+}\tilde{\mathbf{x}}_{t,c} + \mu\tilde{\mathbf{X}}_c, \tag{3}$$

where $\mathrm{P}_c^{+} \in \mathbb{R}^{4\times3}$ is the pseudo-inverse of P_c and $\mathbf{X}_c$ is the 3D location of the camera center. The symbol with superscript tilde denotes the corresponding homogeneous coordinate. The 3D correspondence is then defined as:

$$A_{3D}(\mathbf{X}_{t'}, \mathbf{x}_{t,c}) = w_{3D}(1 - \frac{d_l(\hat{\mathbf{X}}_t, \mathbf{X}_t(\mu))}{\alpha_{3D}}) \cdot e^{-\lambda_a(t-t')}, \tag{4}$$

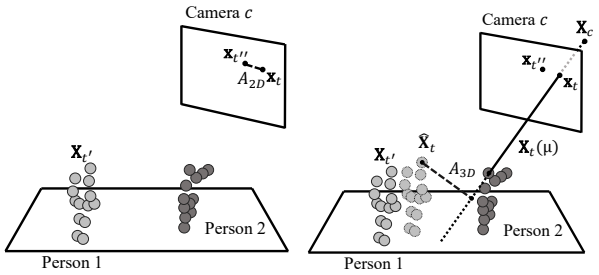
where $d_l(\cdot)$ denotes the point-to-line distance in 3-space and α_{3D} is the threshold of distance. Note that in this formulation, the detected point is compared with a predicted point $\hat{\mathbf{X}}_t$ at the same time t. A linear motion model is introduce to predict the 3D location at time t:

$$\hat{\mathbf{X}}_t = \mathbf{X}_{t'} + \mathbf{V}_{t'} \cdot (t - t'), \tag{5}$$

where $t \geq t'$ and $\mathbf{V}_{t'}$ is 3D velocity estimated via a linear least-square method.

Here, for the purpose of verifying the iterative processing strategy, we only employ the geometric consistency in the affinity measurement for simplicity. This baseline formulation already achieves state-of-the-art performance for both human body association and 3D pose estimation, as we demonstrated in experiments. The key contribution comes from Equation 4, where we match the detected 2D joints with targets directly in 3-space.

Compared with matching in pairs of views in the camera coordinates [13], our formulation has three advantages: 1) matching in 3-space is robust to partial occlusion and inaccurate 2D localization, as the 3D pose actually combines the information from multiple views; 2) motion estimation in 3-space is more feasible and reliable than that in 2D camera coordinates; 3) the computational cost is significantly reduced since only one comparison is required in 3-space for each pair of target and detection. To verify this, a quantitative comparison is further conducted in ablation study.



(a) 2D correspondence (b) 3D correspondence

Figure 2: Geometric affinity measurement. (a) 2D correspondence is computed within the same camera. (b) 3D correspondence is measured between the predicted location and the projected line in 3-space.

Target update and initialization. With previous affinity measurement, this section describes how we update and initialize targets in a particular iteration. Firstly, we compute the affinity matrix between targets and detections using Equation 1 and solve the association problem in bipartite graph matching. Each detection is either assigned to a target or labeled as unmatched based on the association results. In the former case, if a detection is assigned to a target, the 3D pose of the target will be updated gradually with the new detection, as the 2D information is observed over time. Thus, 3D pose reconstruction in our framework is an incremental process, as detailed in Section 3.3.

As for the target initialization, we collect unmatched detections from different cameras and associate them across views using epipolar constraint [16]. Here for each camera, only the most recent frame is retained, thus we assume all detections are from very similar times and can be matched directly. Particularly, we solve the association problem in weighted graph partitioning [31, 10], to comply the cycle-consistency constraint as there are multiple cameras [13]. Body pose of a new target is initialized in 3-space from the detections when at least two views are matched. The overall procedure of cross-view tracking is shown in Algorithm 1.

3.3. Incremental 3D pose reconstruction

Generally, given 2D poses of the same person at a time in different views, the 3D pose can be reconstructed using triangulation. However, with the iterative processing, 2D poses in our framework may come from different times, raising the incremental triangulation problem.

Supposing the new frame is from camera c at time t, for a target $T_{t'}$ with the matched detection $D_{t,c}$ we collect 2D points from different cameras for each type of human joints:

$$\mathbf{J}_t^k = \{\mathbf{x}_{t,c}^k\} \cup \{\mathbf{x}_{t_i,c_i}^k | c_i \neq c\}, \tag{6}$$

where $\mathbf{x}_{t,c}^k$ is the new point in camera c, and $\mathbf{x}_{t_i,c_i}^k$ denotes the last observed point in camera c_i. For each joint, the 3D location is estimated independently, thus we omit the index k in the following discussion for clarity. Here we aim at estimating the 3D location $\mathbf{X}_t$ from the point collection $\mathbf{J}_t$, where the points are from different times.

We first briefly introduce the linear algebraic triangulation algorithm and then explain our improvement that designed for this problem. For each camera, the relationship between 2D point $\mathbf{x}_{t,c}$ and 3D point $\mathbf{X}_t$ can be written as:

$$\tilde{\mathbf{x}}_{t,c} \times (\mathrm{P}_c \tilde{\mathbf{X}}_t) = \mathbf{0}, \tag{7}$$

where $\times$ is the cross product, $\tilde{\mathbf{x}}_{t,c} \in \mathbb{R}^3$ and $\tilde{\mathbf{X}}_t \in \mathbb{R}^4$ are the homogeneous coordinates, and $\mathrm{P}_c \in \mathbb{R}^{3\times4}$ denotes the projection matrix. Writing Equation 7 out on multiple cameras gives the equation of the form:

$$\mathbf{C}\tilde{\mathbf{X}}_t = \mathbf{0}, \tag{8}$$

with

$$\mathbf{C} = \begin{bmatrix} x_1\mathbf{p}_1^{3\mathrm{T}} - \mathbf{p}_1^{1\mathrm{T}} \\ y_1\mathbf{p}_1^{3\mathrm{T}} - \mathbf{p}_1^{2\mathrm{T}} \\ x_2\mathbf{p}_2^{3\mathrm{T}} - \mathbf{p}_2^{1\mathrm{T}} \\ y_2\mathbf{p}_2^{3\mathrm{T}} - \mathbf{p}_2^{2\mathrm{T}} \\ \cdots \end{bmatrix}, \tag{9}$$

where (x_c, y_c) denotes the 2D point $\mathbf{x}_{t,c}$, and $\mathbf{p}_c^{i\mathrm{T}}$ is the i-th row of P_c. If there are at least two views, Equation 8 is overdetermined and can be solved via singular value decomposition (SVD). The final non-homogeneous coordinate $\mathbf{X}_t$ can be obtained by dividing the homogeneous coordinate $\tilde{\mathbf{X}}_t$ by its fourth value: $\mathbf{X}_t = \tilde{\mathbf{X}}_t / (\tilde{\mathbf{X}}_t)_4$.

The conventional triangulation algorithm assumes that 2D points of different views are from the same time and independently of each other. However, in our case the points are collected from different times (Equation 6). The time difference between points varies from 0 to 300 ms in practice, according to the frame rate and temporary occlusion.

Aiming at estimating the 3D point $\mathbf{X}_t$ for the newest time t, we argue that points from different times should have different importance when solving Equation 8. To this end, we add weights $\mathbf{w_c}$ to the coefficients of $\mathbf{C}$ corresponding to different cameras:

$$(\mathbf{w_c} \circ \mathbf{C})\tilde{\mathbf{X}}_t = \mathbf{0}, \tag{10}$$

where $\mathbf{w_c} = (w_1, w_2, w_3, w_4, ...)$ and $\circ$ denotes Hadamard product. This is a similar formulation to that in [17], where

Algorithm 1: Tracking procedure for each iteration

Input: New 2D human poses $\mathbb{D}_{t,c} = \{D_{j,t,c} | j = 1, ..., M\}$
Previous targets $\mathbb{T}_{t'} = \{T_{i,t'} | i = 1, ..., N\}$ at time t'
Previous unmatched detections $\mathbb{D}_u = \{D_{t_i,c_i}\}$
Output: New targets with 3D poses $\mathbb{T}_t = \{T_{i,t}\}$ at time t

Initialization: $\mathbb{T}_t \leftarrow \emptyset$; $\mathbf{A} \leftarrow \mathbf{A}_{N\times M} \in \mathbb{R}^{N\times M}$
/* cross-view association */
foreach $T_{i,t'} \in \mathbb{T}_{t'}$ **do**
  foreach $D_{j,t,c} \in \mathbb{D}_{t,c}$ **do**
    $\mathbf{A}(i,j) \leftarrow A(T_{i,t'}, D_{j,t,c})$
  end
end
$Indices_T, Indices_D \leftarrow \mathrm{HungarianAlgorithm}(\mathbf{A})$
/* target update */
foreach $i, j \in Indices_T, Indices_D$ **do**
  $T_{i,t} \leftarrow \mathrm{Incremental3DReconstruction}(T_{i,t'}, D_{j,t,c})$
  $\mathbb{T}_t \leftarrow \mathbb{T}_t \cup \{T_{i,t}\}$
end
/* target initialization */
foreach $j \in \{1, ..., M\}$ *and* $j \notin Indices_D$ **do**
  $\mathbb{D}_u \leftarrow \mathbb{D}_u \cup \{D_{j,t,c}\}$
end
$\mathbf{A}_u \leftarrow \mathrm{EpipolarConstraint}(\mathbb{D}_u)$
foreach $\mathbb{D}_{cluster} \in \mathrm{GraphPartition}(\mathbf{A}_u)$ **do**
  if $\mathrm{Length}(\mathbb{D}_{cluster}) \geq 2$ **then**
    $T_{new,t} \leftarrow \mathrm{3DReconstruction}(\mathbb{D}_{cluster})$
    $\mathbb{T}_t \leftarrow \mathbb{T}_t \cup \{T_{new,t}\}$
    $\mathbb{D}_u \leftarrow \mathbb{D}_u - \mathbb{D}_{cluster}$
  end
end

$\mathbf{w_c}$ is estimated by a convolution neural network for the confidences of 2D points. Differently, our method is designed for incremental processing on time series:

$$w_i = e^{-\lambda_t(t-t_i)} / \left\|\mathbf{c}^{i\mathrm{T}}\right\|_2, \tag{11}$$

where λ_t is the penalty rate, $t_i \leq t$ is the timestamp of the point, and $\mathbf{c}^{i\mathrm{T}}$ denotes the i-th row of $\mathbf{C}$. In this case, the importance of the point increases as its timestamp closes to the last time, making the estimated 3D point $\mathbf{X}_t$ closer to the actual joint location at time t. The second term of L^2-norm is written to eliminate the bias from different 2D locations in different views, as introduced in Equation 9.

4. Experiments

We perform the evaluation on three widely-used public datasets: Campus [2], Shelf [2], and CMU Panoptic [18], and compare our method with previous works in terms of both accuracy and efficiency. We also propose a new dataset with 12 to 28 camera views, to verify the scalability of our method as the numbers of cameras and people increase.

4.1. Datasets

We first briefly introduce the public datasets and evaluation metric for multi-human 3D pose estimation. Then we present the detail of our proposed dataset and compare it with existing public datasets.

Campus and Shelf. The Campus is a small-scale dataset that captured by three calibrated cameras. It consists of three people interacting with each other on an open outdoor square. The Shelf dataset is captured by five cameras with a more complex setting, where four people are interacting and disassembling a shelf in a small indoor area. The joint annotations of these two datasets are provided by Belagiannis *et al*. [2] for evaluation. We follow the same evaluation protocol as in previous works [2, 3, 15, 13] and compute the PCP (percentage of correctly estimated parts) scores to measure the accuracy of 3D pose estimation.

CMU Panoptic. The CMU Panoptic dataset [18] is captured in a closed studio with 480 VGA cameras and 31 HD cameras. The hundreds of cameras are distributed over the surface of a geodesic sphere with about 5 meters of width and 4 meters of height. The studio is designed to simulate and capture social activities of multiple people and therefore the space inside the sphere is built without obstacle. For the lack of the ground truth of 3D poses of multiple people, only qualitative results are presented on this dataset. In contrast to previous works [13, 17] that exploit only a few cameras (about two to five) for 3D pose estimation, we present analysis with different numbers of cameras in our ablation study.

Our dataset. Our dataset, namely Store dataset, is captured inside two kinds of simulated stores with 12 and 28 cameras, respectively. Different from CMU Panoptic that uses hundreds of cameras for a small closed area, we evenly arrange the cameras on the ceiling of the store to simulate the real-world environment. Each camera works independently without strict synchronization, as we discussed in Section 3.1. Moreover, there are lots of shelves inside the second store, serving as obstacles, making the scene more complex than previous datasets. A detailed comparison is presented in Table 1. We use the Store dataset along with the CMU Panoptic dataset to verify the scalability of our method on the large-scale camera systems.

4.2. Comparison with state-of-the-art

We first present the quantitative comparison with other state-of-the-art methods in Table 2. Belagiannis *et al*. introduced 3DPS for multi-view multi-human 3D pose estimation in [2]. Afterwards, they extended 3DPS for videos by exploiting the temporal consistency in [4]. These early works have a huge state space with a very expensive computational cost. Dong *et al*. [13] propose to cluster joints at the body level to reduce the state space. An appearance model [36] is also investigated in their work to mitigate the ambiguity of the body-level association. Their approach takes about 25 ms on a dedicated GPU to extract appearance features and 20 ms for the body association, and 60 ms for the 3D reconstruction in 3DPS. Without bells and whistles, our geometric-only method outperforms pervious 3DPS-based models and achieves competitive accuracy with the very recent work [13], while our method is much faster with only a single laptop CPU. Note that, for the fair comparison, we use the same 2D pose detections for the experiments as that in [13], which are provided by an off-the-shelf 2D pose estimation method [11].

Dataset	Cameras	People	Area	Obstacle
Campus	3	3	43	None
Shelf	5	4	19	Shelf
CMU Panoptic	480+31	7	17	None
Store layout1 (ours)	12	4	12	None
Store layout2 (ours)	28	16	23	Shelves

Table 1: Comparison of datasets. The area is computed in square meters using convex hull of camera locations.

	PCP(%)				
Campus	Actor1	Actor2	Actor3	Average	FPS
CVPR14 [2]	82.0	72.4	73.7	75.8	-
ECCVW14 [4]	83.0	73.0	78.0	78.0	1
TPAMI16 [3]	93.5	75.7	85.4	84.5	-
MTA18 [15]	94.2	92.9	84.6	90.6	-
CVPR19 [13]	**97.6**	93.3	98.0	96.3	9.5
Ours	97.1	**94.1**	**98.6**	**96.6**	**617**
Shelf	Actor1	Actor2	Actor3	Average	FPS
CVPR14 [2]	66.1	65.0	83.2	71.4	-
ECCVW14 [4]	75.0	67.0	86.0	76.0	1
TPAMI16 [3]	75.3	69.7	87.6	77.5	-
MTA18 [15]	93.3	75.9	94.8	88.0	-
CVPR19 [13]	98.8	**94.1**	**97.8**	**96.9**	9.5
Ours	**99.6**	93.2	97.5	96.8	**325**

Table 2: Quantitative comparison on the Campus and Shelf datasets. FPS of other methods is the average speed taken from the papers, as per-dataset speed is not provided.

4.3. Ablation study

To further verify the effectiveness of our solution, ablation study is conducted to answer the following questions: 1) Whether matching in 3-space has achieved better results comparing to its 2D counterparts? 2) How much is the contribution of the incremental triangulation, is it really necessary? 3) What is the speed of our method on large-scale camera systems and how much is the contribution of the iterative processing? 4) How is the quality of the tracking?

Matching in 2D or 3D? As described in Section 3.2, we argue that matching in 3-space leads to more accurate association results, since it robust to partial occlusion and inaccurate 2D localization. To verify that, instead of comparing the final PCP score, we measure the association accuracy directly and compare our method with four baselines, as shown in Figure 3. The association accuracy is computed for each camera based on the degree of agreement between clustered 2D poses and annotations. This formulation

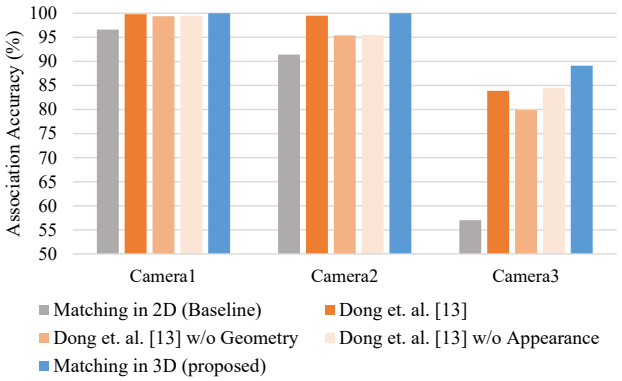


Figure 3: Association accuracy on the Campus dataset.

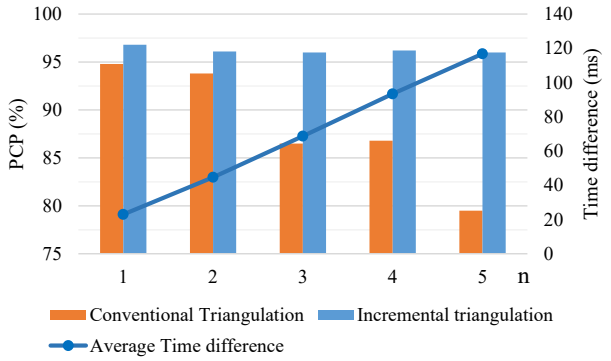


Figure 4: PCP score in terms of input frame rate on the Shelf dataset. The original frame rate is 25 FPS, therefore the actual frame rate of each trail is $25/n$.

removes the impact of different reconstruction algorithms. The first baseline is matching joints in pairs of views in the 2D camera coordinates via epipolar constraint. The following three baselines are taken from the official implementation of [13], which employs geometric information and human appearance features for matching 2D poses between camera views. As seen in the figure, all these approaches achieve good performance in Camera1 and Camera2 of the Campus dataset, while the gap is revealed in the more difficult Camera3, which is placed closer to the people and suffers more from occlusion. Our method that matching in 3-space outperforms the baselines with 32%, 5.2%, 9.2%, 4.6% association accuracy in Camera3, respectively.

Different 3D reconstruction methods. Cross-view association is the first step of 3D pose estimation, while 3D reconstruction is also critical. Here, we retain the association results of our method and estimate the 3D poses using different reconstruction algorithms. As presented in Table 3 four algorithms are considered: 3DPS, conventional triangulation, incremental triangulation without normalization, and our proposed. We select torso, upper arm, lower arm for comparison because these body parts have different motion amplitudes that can evaluate for different cases. All the four reconstruction algorithms achieve good performance on the torso as it has a small range of motion and is easy to detect. As for the lower arm, which can generally move quickly, our incremental triangulation improves about 3% to 5% PCP score compared with conventional triangulation.

To further verify if the incremental triangulation has the ability to handle unsynchronized frames, we analyze the performance drop when the input frame rate decreases. The original Shelf dataset was captured with 25 FPS. We construct datasets with different frame rates by sampling one frame from every n frames in each camera. The comparison between incremental and conventional triangulation is shown in Figure 4. Average time differences within every 2D joint collection $\mathbf{J}_t^k$ are also recorded in the figure. As

Campus	Torso	Upper arm	Lower arm	Whole
3DPS	100.0	**99.1**	82.5	96.0
Triangulation	100.0	95.4	79.1	94.4
Ours, w/o norm	100.0	95.6	81.7	95.4
Ours, proposed	100.0	98.6	**84.6**	**96.6**
Shelf	Torso	Upper arm	Lower arm	Whole
3DPS	100.0	98.1	**88.4**	96.6
Triangulation	100.0	97.0	84.5	94.8
Ours, w/o norm	100.0	**98.7**	87.7	**96.9**
Ours, proposed	100.0	**98.7**	87.7	96.8

Table 3: PCP scores of different 3D reconstruction algorithms on the Campus and Shelf datasets.

the input frame rate decreases and the time differences increase, the performance of conventional triangulation drops significantly, while that of ours keeps stable, indicating the effectiveness of our method in handling the unsynchronized frames. Therefore, we confirm that incremental triangulation is essential for the iterative processing.

Speed on large-scale camera system. As already seen in Table 2, our method is about 50 times faster than others on the small-scale datasets Campus and Shelf. We further test the proposed method on the large-scale Store dataset as demonstrated in Figure 5. It finally achieves 154 FPS for 12 cameras with 4 people and 34 FPS for 28 cameras with 16 people. Note that when counting the running speed, we follow the common practice that one frame represents that all cameras are updated once.

Indeed, different implementation and hardware environment affect the running speed a lot. Our algorithm is implemented in C++ without multi-processing and evaluated on the laptop with an Intel i7 2.20 GHz CPU. In order to verify the efficiency more fairly and understand the contribution of iterative processing, we construct a baseline method that matches joints in pairs of views in the camera coordinates with the same testing environment. The comparison is con-

Figure 5: Qualitative result on the Store dataset (layout 2). There are 28 cameras and 16 people in the scene and different people are represented in different colors. The camera locations are illustrated in the 3D view as triangles in blue.

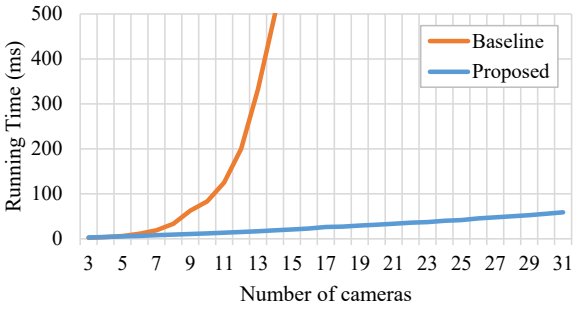


Figure 6: Average running time of one frame with different numbers of cameras on the CMU Panoptic dataset.

Method	Camera	MOTA	IDF1	FP	FN	IDS
Single-view	Camera1	86.7	81.7	32	34	2
	Camera2	97.6	63.9	4	4	4
	Camera3	97.3	98.6	7	7	0
	Camera4	68.8	41.8	77	79	3
	Camera5	79.0	69.0	51	51	5
Cross-view	Camera1	98.8	99.4	3	3	0
	Camera2	99.2	99.6	1	1	2
	Camera3	98.4	99.2	4	4	0
	Camera4	97.6	98.8	6	6	0
	Camera5	97.6	98.8	6	6	0

Table 4: Tracking performance on the Shelf dataset.

ducted on the CMU Panoptic dataset with its 31 HD cameras, as the cameras are all placed in a closed small area that changing the number of cameras does not affect the number of people observed. As shown in Figure 6, the running time of the baseline method explodes as the number of cameras increases, while that of ours varies almost linearly. The result verifies the effectiveness of the iterative processing strategy and demonstrates the ability of our method to work with large-scale camera systems in real-world applications.

Tracking quality. We measure the tracking quality using the Shelf dataset. Particularly, we project the estimated 3D poses onto each camera and follow the same evaluation protocol as MOTChallenge [24]. We compare our result with a simple single-view tracking baseline [6] as shown in Table 4. In some easy cases, e.g. Camera2, the baseline single-view tracker achieves similar performance as cross-view tracking. But for the difficult cases such as Camera4, which contain severe occlusion, our cross-view tracking outperforms its single-view counterpart significantly. The result verifies that, in our framework, multi-human tracking can be also boosted by multi-view 3D pose estimation.

5. Conclusion

We have presented a novel solution for multi-human 3D pose estimation from multiple camera views. By exploiting the temporal consistency in videos, we propose to match the 2D inputs with 3D poses in 3-space directly, where the 3D poses are retained and iteratively updated by a cross-view tracking. In experiments, we have achieved state-of-the-art accuracy and efficiency on three public datasets. The comprehensive ablation study demonstrates the effectiveness of each component in our framework. Given its simple formulation and efficiency, our solution can be extended easily by other techniques such as appearance features, and applied directly to other high-level tasks. In addition, we propose a new large-scale Store dataset to simulate the real-world scenarios, which verifies the scalability of our solution and may also benefit future researches in this area.

6. Acknowledgement

This work was supported by the Natural Science Foundation of China (Project Number 61521002).

References

[1] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 623–630. IEEE, 2010. 2

[2] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1669–1676, 2014. 1, 2, 5, 6

[3] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures revisited: Multiple human pose estimation. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):1929–1942, 2016. 6

[4] Vasileios Belagiannis, Xinchao Wang, Bernt Schiele, Pascal Fua, Slobodan Ilic, and Nassir Navab. Multiple human pose estimation with temporally consistent 3d pictorial structures. In *European Conference on Computer Vision Workshop*, pages 742–754. Springer, 2014. 2, 6

[5] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1806–1819, 2011. 2

[6] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 8

[7] Lewis Bridgeman, Marco Volino, Jean-Yves Guillemaut, and Adrian Hilton. Multi-person 3d pose estimation and tracking in sports. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 3

[8] Magnus Burenius, Josephine Sullivan, and Stefan Carlsson. 3d pictorial structures for multiple view articulated pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3618–3625, 2013. 1, 2

[9] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017. 1

[10] Long Chen, Haizhou Ai, Rui Chen, and Zijie Zhuang. Aggregate tracklet appearance features for multi-object tracking. *IEEE Signal Processing Letters*, 26(11):1613–1617, 2019. 4

[11] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018. 1, 6

[12] Yu Cheng, Bo Yang, Bo Wang, Wending Yan, and Robby T Tan. Occlusion-aware networks for 3d human pose estimation in video. In *ICCV*, 2019. 2

[13] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7792–7801, 2019. 1, 2, 3, 4, 6, 7

[14] Ahmed Elhayek, Edilson de Aguiar, Arjun Jain, Jonathan Tompson, Leonid Pishchulin, Micha Andriluka, Chris Bregler, Bernt Schiele, and Christian Theobalt. Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3810–3818, 2015. 2

[15] Sara Ershadi-Nasab, Erfan Noury, Shohreh Kasaei, and Esmaeil Sanaei. Multiple human 3d pose estimation from multiview images. *Multimedia Tools and Applications*, 77(12):15573–15601, 2018. 6

[16] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 2, 3, 4

[17] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *ICCV*, 2019. 5, 6

[18] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015. 2, 5, 6

[19] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 3

[20] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–135, 2018. 2

[21] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017. 2

[22] 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 *2018 International Conference on 3D Vision (3DV)*, pages 120–130. IEEE, 2018. 2

[23] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 2

[24] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 8

[25] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *ICCV*, 2019. 1, 2

[26] Francesc Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2823–2832, 2017. 1, 2

[27] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6988–6997, 2017. 2

[28] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 2

[29] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *ICCV*, 2019. 2

[30] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–84, 2018. 2

[31] Ergys Ristani and Carlo Tomasi. Tracking multiple people online and in real time. In *Asian conference on computer vision*, pages 444–459. Springer, 2014. 4

[32] Zheng Tang, Renshu Gu, and Jenq-Neng Hwang. Joint multi-view people tracking and pose estimation for 3d scene reconstruction. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018. 2, 3

[33] Graham W Taylor, Leonid Sigal, David J Fleet, and Geoffrey E Hinton. Dynamical binary latent variable models for 3d human pose tracking. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 631–638. IEEE, 2010. 2

[34] Denis Tome, Matteo Toso, Lourdes Agapito, and Chris Russell. Rethinking pose in 3d: Multi-stage refinement and recovery for markerless motion capture. In *2018 International Conference on 3D Vision (3DV)*, pages 474–483. IEEE, 2018. 2

[35] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018. 1

[36] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camera style adaptation for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5157–5166, 2018. 6