Multi-person 3D Pose Estimation and Tracking in Sports

Lewis Bridgeman
l.bridgeman@surrey.ac.uk

Marco Volino
marco.volino@surrey.ac.uk

Jean-Yves Guillemaut
j.guillemaut@surrey.ac.uk

Adrian Hilton
CVSSP
University of Surrey
a.hilton@surrey.ac.uk

Abstract

We present an approach to multi-person 3D pose estimation and tracking from multi-view video. Following independent 2D pose detection in each view, we: (1) correct errors in the output of the pose detector; (2) apply a fast greedy algorithm for associating 2D pose detections between camera views; and (3) use the associated poses to generate and track 3D skeletons. Previous methods for estimating skeletons of multiple people suffer long processing times or rely on appearance cues, reducing their applicability to sports. Our approach to associating poses between views works by seeking the best correspondences first in a greedy fashion, while reasoning about the cyclic nature of correspondences to constrain the search. The associated poses can be used to generate 3D skeletons, which we produce via robust triangulation. Our method can track 3D skeletons in the presence of missing detections, substantial occlusions, and large calibration error. We believe ours is the first method for full-body 3D pose estimation and tracking of multiple players in highly dynamic sports scenes. The proposed method achieves a significant improvement in speed over state-of-the-art methods.

1. Introduction

The problem of estimating 3D pose from video is a well-explored one. There has been significant research activity into calculating 3D pose from both monocular [4, 8, 23, 33] and multi-view video [6, 13, 20, 30], yet few existing methods have been crafted for the sports domain. Sports datasets are especially challenging for computer vision algorithms due to: player contact and fast motion; similar player appearance; heavy occlusion; moving, low resolution and very wide-baseline cameras; and poor calibration. However, the potential applications of estimating the 3D pose of players in sports are wide-reaching. These include performance analysis, motion capture, and novel applications in broadcast and immersive media.

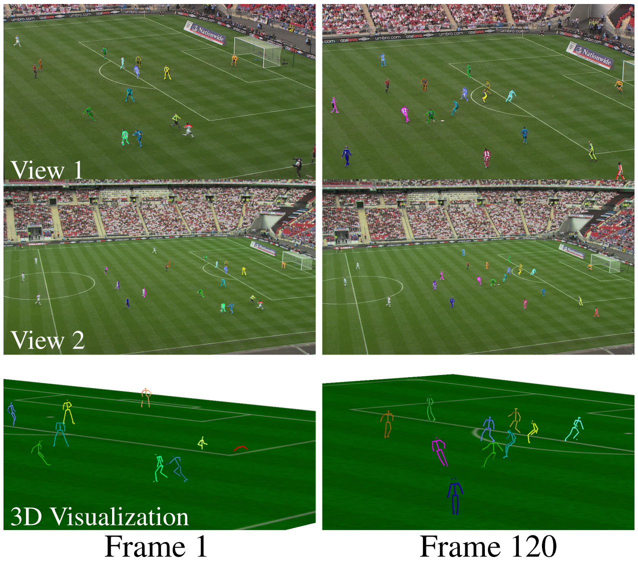


Figure 1: Results of our proposed method. The first two rows show our results overlaid on the input images, and the bottom row shows the results from a new camera view.

Estimating the 2D poses of multiple people in images is a well-understood problem [10, 18]. The task of combining multi-person 2D detections from multiple views to generate 3D skeletons has also been explored [6, 12, 20]. However, none of these methods are especially applicable to sports due to lengthy processing times, reliance on appearance models, or sensitivity to calibration error and noisy pose detections. In this paper we propose a greedy algorithm to find correspondences between 2D poses in multiple views, employing them to generate 3D skeletons. By maintaining the connectivity of 2D poses, the resulting method provides a significant improvement in speed versus recent methods. We introduce techniques to correct the errors associated with multi-person pose detectors; split poses, fused poses and swapped joints. Finally, we introduce an algorithm to track our generated 3D skeletons, and hence their

2D counterparts, throughout a sequence. Our algorithm has been shown to work on a variety of sports datasets with poor calibration, erroneous pose detections, and substantial occlusion. Example results can be seen in Figure 1.

Our contributions include:

- A method for identifying and correcting errors in a multi-person pose detector output by employing multi-view information.
- A fast greedy algorithm for identifying correspondences between 2D poses in multi-view video.
- A method for tracking 3D skeletons in sequences with missing and noisy joint estimations.

2. Related Work

The literature on estimating 2D human pose from monocular images can be categorized into single-person [15, 34, 37] and multi-person [10, 18, 21, 28] methods. Before the uptake in convolutional neural network (CNN) methods, the state-of-the-art employed generative models. Pictorial structures model a pose as a collection of connected parts, with priors constraining the relative positions or angles of each part. The parts are aligned to image data in an energy minimization. Pictorial structures were originally applied to monocular 2D pose estimation [15, 27], but have also been extended to multi-view 3D pose estimation [1, 9].

CNNs have successfully been applied to pose estimation [29, 34, 37]. In [34], a regressor is trained to directly return the joint coordinates. An end-to-end architecture that learns spatial models for pose estimation was presented in [37].

CNN-based pose estimators result in a significant increase in accuracy, and provide a basis for more difficult pose estimation tasks such as multi-person 2D pose estimation [10, 18, 28]. In [10], a method is introduced to estimate poses of multiple people in real-time by fusing joint confidence maps, and a learned vector field that defines the relationship between joints. Monocular 3D pose estimation has been approached using pictorial structures [4], fitting 3D skeletons to 2D joints [11, 23], and using convolutional architectures [33]. The methods in [26, 31] extend convolutional pose estimation to video, using the temporal information to overcome the challenges of estimating pose from a single frame.

Markerless motion capture tracks the motion of the human skeleton in 3D without using traditional optical markers and specialized cameras. This is essential in sports capture, where players cannot be burdened with additional performance capture attire. There has been extensive research into markerless motion capture of a single subject [5, 22, 30, 32]. The method in [36] fuses 2D pose detections and data from inertial measurement units (IMUs) to recover 3D pose for multiple people. In [22] IMUs and 2D pose are combined to capture the 3D pose of a single person in real-time. The pictorial structure model is extended in [9] for use in estimating the 3D pose of a single person from multiple views. An early model-based tracking method is presented in [25], which uses multi-view silhouettes and color information to track up to two people in a studio environment. This method is effective, but requires manual initialization of the geometry of each actor. The method in [32] introduces a sum of Gaussians appearance model for near video-rate motion capture of a single subject, but also requires initialization.

Multi-person markerless motion capture methods are required in order to capture team sports. Markerless motion capture of multiple people in multi-view video has been investigated in [5, 6, 13, 14, 20]. The approaches in [14, 20] take 2D pose detections from multiple views and use volumetric voting to find all 3D joint locations. The method in [14] clusters the 3D joints, and a 3D Pictorial Structure (3DPS) model is used to generate 3D poses for each cluster. In [20], the 3D joints are grouped into body parts, which are then grouped into full skeletons; the trajectory of each skeleton is also tracked. Volumetric voting is an effective way of estimating the 3D joints of multiple people, provided there are enough cameras and the scene is not crowded. However, it would be expensive if applied to a large capture area such as a soccer stadium. Additionally, volumetric voting is sensitive to poor calibration and erroneous pose detections, both of which are common in sports datasets.

The method in [6] attempts to increase the speed of multi-person 3DPS models by reducing the state-space to all pairwise joint triangulations, although the method only runs at 1fps for a single subject. A model-based tracking algorithm and 2D pose detections are fused in [13] to achieve multi-person motion capture from minimal viewpoints, although their algorithm requires initialization of every person in the scene. Recent work [12] finds correspondences between 2D poses in multiple views in an optimization framework, combining epipolar geometry costs and CNN appearance descriptors. A second stage fits a 3DPS model to each person individually. The use of appearance models is not applicable in sports, where players wear matching outfits. The method runs at 10fps on 3-person datasets; significantly faster processing times are required for sports broadcast applications.

Our method employs a greedy search to find correspondences between 2D poses in different camera views, and is able to achieve this at video-rate speeds. It relies solely on geometry terms, rather than appearance models. The method can compensate for some of the typical errors that arise in multi-person 2D pose estimation. The 2D pose associations can be used to generate 3D skeletons, which are tracked temporally, even in sequences with heavy occlusion and erroneous pose detections.

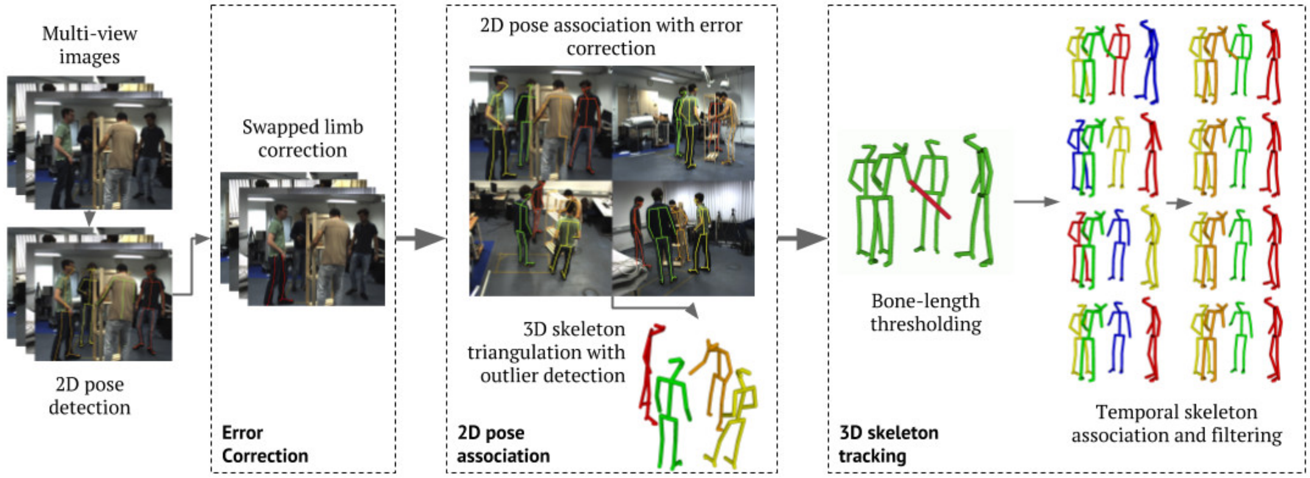


Figure 2: An overview of the pipeline. Our method finds correspondences between 2D pose estimations in multiple views. We compute 3D skeletons each frame, identify tracks of 3D skeletons and filter the results.

3. Methodology

The proposed framework takes as input multi-view video of multiple people and camera calibration. The multi-view videos are passed through a pose detector [10] providing unsorted 2D pose estimations each frame. Three successive processes are applied to the data: the first step corrects some of the errors in the output of the pose detector; the second step applies a label to every 2D pose, ensuring consistency between views; finally, the labelled 2D poses each frame are used to produce a sequence of tracked 3D skeletons. A system overview is presented in Figure 2.

2D Pose Error Correction: Single and multi-view information is used to correct some of the errors found in the pose detector output: part flipping, single-person splitting, and multiple-person fusion. Flipped body parts are corrected by comparing the correspondence scores of the original and reversed poses. Candidate split poses are identified and subsequently corrected in the pose association stage.

Per-frame 2D Pose Association: Associations between 2D poses in differing camera views are found using a greedy algorithm. These associations are used to generate a set of labels, such that poses belonging to a single person share a common label.

3D Skeleton Tracking: The labelled 2D poses are used to generate 3D skeletons each frame. Associations are found between skeletons in consecutive frames, resulting in tracked 3D skeletons to which filtering is applied.

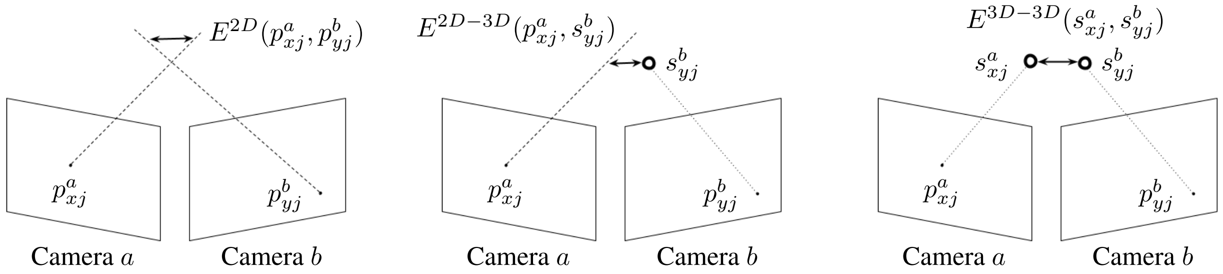
3.1. Per-frame 2D Pose Association

The aim of this stage is to find a label for each 2D pose whereby all poses that correspond to one person share a label. A cost is assigned to all pairs of poses between views that measures the likelihood of their being in correspondence; this is used as the heuristic in finding associated poses in a greedy algorithm. The algorithm takes a cyclic approach, whereby new associations provide additional information about the location of skeletons in 3-space, thus the correspondence costs can be refined before the next iteration.

The input to this stage is the 2D pose detections, which have passed through the error correction process described in section 3.2. The i-th pose in camera c is given by $p_i^c \in \mathbb{R}^{50}$, which comprises the coordinates of 25 joint detections. The pose detector also provides a confidence for each joint, $\alpha_i^c \in \mathbb{R}^{25}$. The j-th joint in p_i^c is given by p_{ij}^c, and similarly α_{ij}^c is the confidence of joint j. A confidence of zero signifies that the joint was undetected in the image. The algorithm outlined in this section produces labels for each 2D pose; these can be used to generate 3D skeletons, $s \in S$ as described in section 3.3. 3D skeletons may be indexed either by their associated 2D poses (s_i^c is the skeleton associated with p_i^c), or by their assigned label (s_I is the skeleton generated from all poses with label I).

3.1.1 Correspondence Costs

Upon the first iteration of the algorithm, correspondence costs are computed between all pairs of 2D poses in different views. The pose correspondence cost comprises per-joint costs for every joint the two poses have in common (an undetected joint has a confidence of zero). The per-joint cost could be estimated as the deviation of the detections from their respective epipolar lines, however, we opt to use the distance of the common perpendicular vector between the two rays extending from the centres-of-projection (COPs) through the joint detections. The advantages are twofold: unlike an epipolar or reprojection error, this metric is invariant to both the distance of the joint from the COP, and the resolution of the camera; secondly, the cost is measured in 3-space, and thus can be compared to 2D-3D and 3D-3D joint correspondences, which occur in later



(a) E^{2D} (Equation 1) (b) E^{2D-3D} (Equation 2) (c) E^{3D} (Equation 3)

Figure 3: Visualization of the three types of joint correspondence scores.

iterations of the algorithm. The cost of associating joint j in pose x of camera a and pose y of camera b is shown in Equation 1, and illustrated in Figure 3a.

$$E^{2D}(p^a_{xj}, p^b_{yj}) = \frac{\min_{m,n} \|P_a^{-1}(p^a_{xj}, m) - P_b^{-1}(p^b_{yj}, n)\|}{\sqrt{\alpha^a_{xj}\alpha^b_{yj}}} \tag{1}$$

where P_a^{-1} is the inverse projection function of camera a, thus $P_a^{-1}(p^a_{xj}, m)$ is the point at distance m along the ray extending from the coordinates of joint j in pose p^a_x.

After each iteration the newly associated 2D poses are used to infer 3D joint locations. 3D joint positions are subsequently used over 2D joint coordinates where possible, thus the correspondence costs will include 2D-3D (Equation 2, Figure 3b), and 3D-3D correspondences (Equation 3, Figure 3c):

$$E^{2D-3D}(p^a_{xj}, s^b_{yj}) = \frac{\min_{m} \|P_a^{-1}(p^a_{xj}, m) - s^b_{yj}\|}{\alpha^a_{xj}} \tag{2}$$

$$E^{3D}(s^a_{xj}, s^b_{yj}) = \|s^a_{xj} - s^b_{yj}\| \tag{3}$$

where s^b_y is the 3D skeleton associated with 2D pose p^b_y. Finally, the complete correspondence score between the full poses p^a_x and p^b_y is calculated as:

$$\sigma(p^a_x, p^b_y) = \frac{\sum_j E^X}{\sum_j 1}, \quad \{\alpha^a_{xj}, \alpha^b_{yj} > 0\} \tag{4}$$

where E^X represents, in order of preference, Equation 3, 2 then 1, as shown in Equation 5:

$$E^X = \begin{cases} E^{3D}(s^a_{xj}, s^b_{yj}), & \text{if } A(p^a_{xj}) = A(p^b_{yj}) = 1 \\ E^{2D-3D}(p^a_{xj}, s^b_{yj}), & \text{if } A(p^a_x) \neq A(p^b_y) \\ E^{2D}(p^a_{xj}, p^b_{yj}), & \text{if } A(p^a_{xj}) = A(p^b_{yj}) = 0 \end{cases} \tag{5}$$

$A(p^a_{xj}) = 1$ if s^a_{xj} has been computed, or 0 otherwise. The 3D-3D and 2D-3D scores are favoured over 2D-2D scores for their reduced ambiguity.

3.1.2 Pose Association Algorithm

The process for finding correspondences and allocating labels is outlined in algorithm 1.

```
Data: 2D poses
Result: Label associated with every 2D pose
while new valid correspondence do
    re-issue pose labels;
    calculate new 3D joint locations;
    recompute correspondence scores;
end
```

Algorithm 1: Per-frame pose association

New Correspondences: New pose correspondences are sought in a greedy fashion; pairs of poses are ranked according to:

$$r(p^a_x, p^b_y) = \frac{\sigma(p^a_x, p^b_y)}{\omega(p^a_x, p^b_y)\gamma(p^a_x)\gamma(p^b_y)} \tag{6}$$

where ω denotes the number of shared joints between two poses, and $\gamma(p^a_x)$ is the number of poses already associated with p^a_x. The ordered list is traversed and the first valid association is found. An association is deemed valid if the correspondence score σ is below the empirically estimated correspondence threshold $\tau_c = 0.4$. Furthermore, all dependent associations must be considered simultaneously. For example, if there is an existing association between (p^a_1, p^b_3), and we wish to associate (p^b_3, p^c_2), then (p^a_1, p^c_2) must also be associated. Therefore, in order to ensure a new association is valid, the average correspondence score of all dependent associations must be below threshold τ_c.

If a valid association is found, the labels are updated such that all associated poses share a common label, and all poses without associations have a unique label. For any poses with updated labels, the 3D joints are recomputed using the method in section 3.3. Finally, the correspondence scores are updated as per section 3.1.1, using newly computed 3D joints where possible, and the list is recomputed. The algorithm repeats until no more valid pose associations are possible.

3.2. 2D Pose Error Correction

The pose detector provides 2D pose estimations and associated joint confidences. Several common errors pass through this stage with high confidence: (1) left-right limb swaps; (2) single-person division into multiple poses; (3) multiple-person fusion into a single pose. Temporal filtering and tracking at the pose estimation stage could help to rectify these errors; this is an active research area [2, 3, 19]. Instead, we aim to correct these errors on a per-frame basis, maintaining the applicability of our pose labelling algorithm to short sequences and individual frames.

Joint Swaps: We observe that true correspondence scores are lowered when a pose contains swapped limbs. To this end, we propose a heuristic to determine likelihood of a pair of limbs being incorrectly swapped based on the average minimum correspondence score with other poses. A pair of limbs is deemed incorrectly flipped if Equation 7 is satisfied:

$$\frac{L(\hat{p}_x^a)}{L(p_x^a)} < \tau_f \tag{7}$$

where p_x^a and $\hat{p}_x^a$ respectively are the original and flipped version of pose x in camera a, and τ_f is an empirically estimated threshold to determine whether a flip is necessary. L is defined as:

$$L(p_x^a) = \frac{\sum_b M(p_x^a, b)\chi_{\tau_c}\langle M(p_x^a, b)\rangle}{\left(\sum_b \chi_{\tau_c}\langle M(p_x^a, b)\rangle\right)^2}, \quad \{b \neq a, b \in C\} \tag{8}$$

$L(p_x^a)$ is the average minimum correspondence score for all other camera views b. It excludes cameras where no pose correspondences are below threshold τ_c. Squaring the denominator favours correspondences in multiple camera images. $M(p_x^a, b)$ is the minimum correspondence score between p_x^a and all poses in camera b, and is defined as:

$$M(p_x^a, b) = \min_y \sigma_{2D}(p_x^a, p_y^b) \tag{9}$$

where σ_{2D} is the pose correspondence score seen in Equation 4, using only 2D joint correspondences. χ_{τ_c} is a thresholding function used to eliminate pose correspondences below threshold $\tau_c = 0.2$, and is given by:

$$\chi_{\tau_c}\langle l\rangle = \begin{cases} 1, & \text{if } l < \tau_c \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Single-person Division: In some cases, the joint detections for a single person will be incorrectly split into two or more poses. In cases where a single person is associated with multiple separate 2D poses in a single image, they will typically have no common joints between them. If the poses do share a joint, it will have the same image coordinates. For all pairwise combinations of poses in a single image, the number of common joints is computed; this sum disregards any common joints for which the distance between the coordinates is below a threshold - *i.e.* nominally identical coordinates. In the pose association stage (section 3.1), the association of multiple poses within the same image is allowed, provided they have zero joints in common and both associate well with a tertiary pose in another image.

Multiple-person fusion: A single pose-detection may sometimes span multiple people in the scene, identifying a subset of joints of each. Two problems must be solved in correcting this category of error: how to recognize when it has occurred; how to identify the subsets of the joints that correspond to different people. The large number of possible divisions of a 25-joint pose into an indeterminate number of subsets belonging to separate people means identifying and correcting this error is highly challenging; it may not be possible to achieve in real-time.

The result of the 2D pose association stage is a label for each pose. We define rules for creating ground truth labels that handle instances of multiple-person fusion. In cases where a large majority (>70%) of joints in a single pose belong to one person, it is assigned the majority ground truth label. In other cases, assigning a ground truth label to a pose is ambiguous, so it is given a unique label that disassociates it from other poses. No changes are made to the 2D pose association stage. In instances of multiple-person fusion, the pose is either disassociated from all other poses due to a bad correspondence score, or associated with the person whose joints make up a majority. This allows for cases where a minority of joints in a pose are associated incorrectly; these will be disregarded using outlier detection in the subsequent triangulation stage.

3.3. 3D Skeleton Tracking

The output from section 3.1 is a label per 2D pose. Where multiple poses have the same label, it is possible to estimate a 3D skeleton. The 3D location of each joint s_{Ij} in a skeleton with label I is optimised per:

$$\arg\min_{s_{Ij}} \sum_c \sum_i \alpha_{ij}^c \|P_c(s_{Ij}) - p_{ij}^c\|, \quad \{p_i \in I, c \in C\} \tag{11}$$

where $P_c(s_{Ij})$ is a is the projection of s_{Ij} in camera c. This results in a set of 3D skeletons per frame. RANSAC is used during the triangulation process to eliminate outlier pose detections. The bone lengths of the resultant skeleton are thresholded to remove any remaining outlier 3D joints.

Following this step, our notation is adjusted to no longer consider cameras, and instead consider multiple frames; s_I^t now represents skeleton I in frame $t \in T$. S^t is the set of all skeletons in frame t. $\alpha_{Ij}^t \in \{0, 1\}$ now represents whether joint j in skeleton I at frame t exists.

The 3D skeleton tracking stage takes unsorted 3D skeletons for all frames and returns sequences of 4D skeletons and their component 2D poses. A greedy algorithm is used to first find correspondences between skeletons in successive frames. The search is then extended to increasingly separated frames up to a maximum of τ_{max}. This allows tracking of skeletons that are not present in every frame, due to error in the association process or occlusion. This process is outlined in algorithm 2.

```
Data: 3D skeletons every frame
Result: Sequences of 3D skeletons
for $i = 1$ to $\tau_{max}$ do
    list correspondences;
    for $t \in T$ do
        for $s_a^t \in S^t$ and $s_b^{t-i} \in S^{t-i}$ do
            correspondences.add($s_a^t$,$s_b^{t-i}$);
        end
    end
    correspondences.sort();
    for c in correspondences do
        if c.cost<$T_s$ then
            c.conjoin();
        end
    end
end
```

Algorithm 2: Per-frame pose association

Correspondence Cost: The correspondence costs between pairs of skeletons is the same as described in section 3.1.1, using only the 3D-3D joint terms:

$$\sigma^{3D}(s_I^{t_1}, s_J^{t_2}) = \frac{\sum_j E^{3D-3D}(s_{Ij}^{t_1}, s_{Jj}^{t_2})\alpha_{Ij}^{t_1}\alpha_{Jj}^{t_2}}{\sum_j \alpha_{Ij}^{t_1}\alpha_{Jj}^{t_2}} \tag{12}$$

where E^{3D-3D} is the 3D joint correspondence cost described in Equation 3. Correspondence costs are computed for all pairwise combinations of skeletons in neighbouring frames. All pairs of skeletons in the sequence are sorted by increasing value of σ^{3D}/ρ, where ρ represents the intersection over union of the joints in each skeleton.

Selecting Correspondences: After ranking all candidate skeleton correspondences, true correspondences can be defined. The list is traversed, and if a correspondence cost σ^{3D} is below threshold T_s the two skeletons are connected. T_s is an empirically obtained threshold defining the maximum allowable difference in σ^{3D} between successive frames. The process is repeated for increasing time differences, up to τ_{max}.

Skeleton Tracks: After all possible skeleton correspondences have been made, the result is tracks of skeletons throughout the video sequence. Each track is traversed, and both the length of the track and the average number of joints in each frame are computed. Tracks shorter than T_l frames and with fewer than T_j joints on average are culled. Both thresholds are chosen empirically; we choose values 30 and 5 respectively. T_s is re-employed to identify and remove noisy joints in each skeleton track with respect to neighbouring frames. Missing joints are linearly interpolated provided that the correspondence of the same joint in the two neighbouring frames falls below threshold T_s. A 3-frame triangle filter is applied to all skeleton tracks to smooth the final result. We compute the tracking stage offline, however the time taken to complete this process is negligible, so it could be computed in real-time with a latency of τ_{max} frames.

4. Results & Evaluation

We test our method on a variety of datasets, both sports and otherwise. We separately evaluate the results of both the 2D pose association algorithm and the 3D skeleton estimation. We assess the accuracy of the 2D pose association method on synthetic multi-view multi-person images. For the evaluation of the 3D skeleton estimation we use the Campus [7] and Shelf [5] datasets, and compare our results to state-of-the-art methods. We also present results of the entire pipeline on a selection of sports datasets.

4.1. 2D Pose Association

We create synthetic multi-view images of multiple people using tools provided with the SURREAL dataset [35]; the scenes comprise textured models in a variety of poses. We create image sets with varying numbers of people and cameras, and with both narrow and wide-baseline camera arrangements. The subjects are contained within a circle of radius 2.5m, and the cameras on a circle of radius 5m. The narrow-baseline cameras have a spacing of 10-degrees, and the wide-baseline cameras are equispaced around the circle. We run the pose detector on the images, and assign a ground truth label to each detected pose. For poses where the joints belong to two or more people we assign the label of the person whose joints are the majority (>70%).

We run our algorithm on each set of images, and generate a binary matrix where each cell represents a pair of poses; 1 represents a correspondence, and 0 otherwise. We evaluate the accuracy over the matrix, using our ground truth labels. The results can be seen in Table 1. The algorithm achieves a higher accuracy with the wide-baseline camera arrangement for a smaller number of cameras, and a larger number of people. This is due to improved robustness to occlusion and triangulation accuracy with wide-baseline views. Notably, the algorithm achieves 100% on scenes with two people for all camera setups.

4.2. 3D Skeleton Estimation

As there are no public domain multi-view sports datasets or existing algorithms applied to multi-person tracking in sports, we evaluate on two public datasets with 3-5 cameras and 3 people: Campus [7] and Shelf [5]. We calculate the

People	Cameras		
	2	4	8
	Narrow-baseline		
2	100.00%	100.00%	100.00%
4	96.50%	97.55%	99.08%
6	95.78%	96.33%	98.15%
8	94.60%	98.41%	98.34%
	Wide-baseline		
2	100.00%	100.00%	100.00%
4	97.78%	96.36%	98.47%
6	97.71%	98.03%	98.56%
8	97.68%	98.55%	98.44%

Table 1: The accuracy of pose correspondences in a narrow and wide-baseline arrangement.

percentage of correct parts (PCP) for each actor. The PCP denotes a body part as correct if the two estimated component joints are less than 50% of the true body-part length away from their ground-truth locations. The alternative definition used by [12] uses the average of the distance of the two joints; we also compute this metric, which we denote by (A). We compare to the methods in [1, 6, 12, 14]. These methods are all designed for general scenes and employ 3DPS models to refine the final skeletons; our method is designed to work on challenging sports scenes, and uses triangulation to estimate the 3D skeleton for speed. We compare the results of 3D skeleton estimation per-frame, and also following the skeleton tracking and temporal filtering stage (section 3.3) which we denote by (ST).

The provided ground truth joints are for the skeleton used in [6], whereas our pose estimator uses a different skeleton. Thus we compute the PCP over all body parts except for the head. The results of the Campus and Shelf dataset are shown in Tables 2a and 2b. Comparing our scores to previous methods, the state-of-the-art achieves a higher performance on the 3-view Campus dataset. However, all other methods use a 3DPS model to constrain the final joint positions; pictorial structure models have been shown to result in more accurate joint estimations than triangulation when the number of views is small [12]. On the 5-view Shelf dataset we achieve a score that is comparable to the state-of-the-art, despite not employing pose priors. Although the performance of our direct triangulation is typically lower than using methods with priors, our algorithm also outputs temporally and spatially corresponding 2D poses for the entire sequence, allowing any method to be substituted for estimating the 3D skeletons.

Campus Dataset [7]			
Method	Actor 1	Actor 2	Actor 3
Amin et al. [1]	85.00	76.56	73.70
Belagiannis et al. [6]	93.45	75.65	84.37
Ershadi-Nasab et al. [14]	**94.18**	**92.89**	84.62
Proposed	85.26	88.54	89.77
Proposed (ST)	86.62	89.01	**90.66**
Dong et al. [12]	**97.60**	**93.30**	**98.00**
Proposed (A)	91.84	92.48	92.83
Proposed (A, ST)	91.84	92.71	93.16

(a)

Shelf Dataset [5]			
Method	Actor 1	Actor 2	Actor 3
Amin et al. [1]	72.42	69.41	85.23
Belagiannis et al. [6]	75.26	69.68	87.59
Ershadi-Nasab et al. [14]	93.29	75.85	94.83
Proposed	98.25	81.68	**97.10**
Proposed (ST)	**98.77**	**85.89**	**97.10**
Dong et al. [12]	98.80	**94.10**	**97.80**
Proposed (A)	99.28	91.59	97.58
Proposed (A, ST)	**99.68**	92.79	97.72

(b)

Table 2: Comparison of the PCP on the Campus (a) and Shelf (b) datasets.

4.3. Skeleton Tracking

We present qualitative results on a number of internal sports datasets that are summarized in Table 3. We apply the full pipeline (error correction, 2D pose association, 3D skeleton tracking) to these datasets, and overlay the final skeletons on the original images. The soccer dataset is particularly challenging, due to poor calibration and the small size of the players in the image; the average bounding box for each player is only 44×79 pixels. To produce 2D pose estimations of the soccer players, we first detect their bounding boxes using [17], then run the pose detector on the cropped images. The 2D pose detections on images this small are frequently erroneous or missing, especially in cases of overlapping people. Selected frames from the final results can be seen in Figure 4 and a video of the results is included in the supplementary material.

Dataset	C	P	R	D	F
Table-tennis [22]	6	4	720p	2.92m	5308
Boxing	8	5	2160p	6.38m	1000
Karate	16	2	2160p	2.28m	1012
Soccer [16]	6	24	1080p	∼48m	120

Table 3: Properties of the datasets used for qualitative evaluation: number of cameras (C); number of people (P); camera resolution (R); average camera distance from origin (D); and number of frames (F).

To assess the quality of the tracking, we compute the number of ID switches, a metric commonly used in multi-object tracking [24] that counts the number of times a tracked object is assigned a new identity. These scores are shown in Table 4.

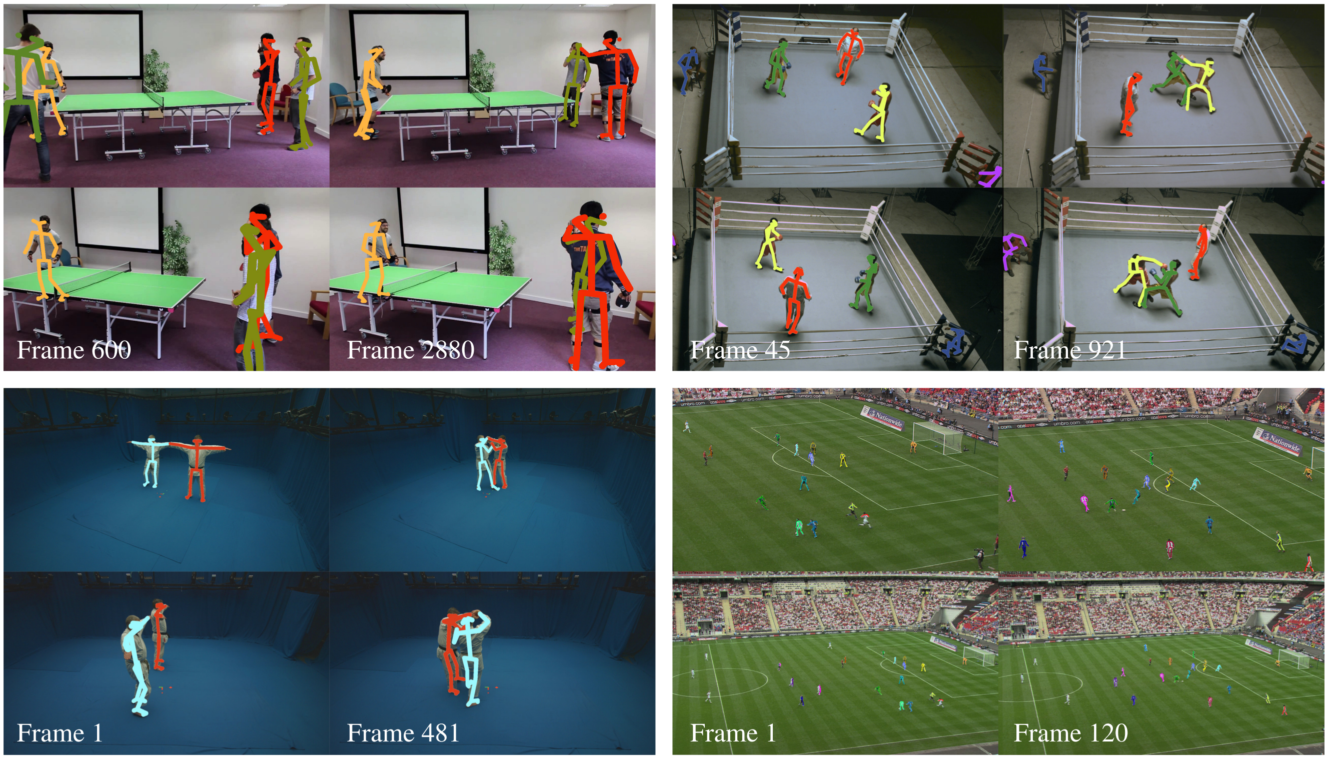


Figure 4: A selection of frames from the table-tennis, boxing, karate and soccer datasets showing results of 3D skeleton estimation plus tracking.

Dataset	F	TP	IDS	Norm. IDS
Table-tennis	5308	4	7	3.3×10^{-4}
Boxing	1000	5	0	0
Karate	1012	2	0	0
Soccer	120	19	1	4.3×10^{-4}

Table 4: The number of frames (F), tracked people (TP), ID switches (IDS), and ID switches after adjusting for number of frames and number of tracked people (Norm. IDS) in each dataset.

In the boxing and karate datasets, all subjects maintain their tracking for the duration of the sequence, even during close contact. In the table-tennis dataset five ID switches are due to subjects leaving the capture volume, and two are due to tracking failures. This may be due in part to poorly estimated distortion parameters, which are observable in the results in Figure 4. In the soccer dataset, 18 out of 19 reconstructed players maintain their tracking over the sequence, despite extended spells of missing frames due to absent 2D pose detections.

4.4. Processing Time

We run our method on a desktop with an Intel i7 3.2GHz processor and 64GB of RAM. All stated speeds are given pre-computed 2D pose estimation. Our parallelized implementation runs at over 110fps on the Shelf dataset. The 2D pose association stage is the most computationally expensive, but the time taken to track the 3D skeletons is negligible - however it does introduce a latency due to the temporal filtering. The methods in [6] and [12], which both use pictorial structure models, run at approximately 1fps and 10fps respectively.

5. Conclusions

In this paper we presented a new method for computing tracked 3D skeletons of people from multi-view sports video. Our pose-matching algorithm compensates for errors in the pose detections, and can identify correspondences between 2D poses in different viewpoints. The algorithm is capable of running in real-time on the Campus and Shelf datasets. Our tracking algorithm has been shown to effectively track players a crowded soccer scene with missing and noisy detections. Our method requires no modelling of actor appearance, and manages to perform well with poor camera calibration and erroneous pose detections.

Acknowledgements

This work was funded by EPSRC Grant EP/N50977/1.

References

[1] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele. Multiview pictorial structures for 3d human pose estimation. In *BMVC*, 2013. 2, 7

[2] M. Andriluka, U. Iqbal, A. Milan, E. Insafutdinov, L. Pishchulin, J. Gall, and B. Schiele. Posetrack: A benchmark for human pose estimation and tracking. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5167–5176, 2018. 5

[3] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 5

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

[5] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3d pictorial structures for multiple human pose estimation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1669–1676, 2014. 2, 6, 7

[6] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3d pictorial structures revisited: Multiple human pose estimation. *IEEE transactions on pattern analysis and machine intelligence*, 38 10:1929–42, 2016. 1, 2, 7, 8

[7] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1806–1819, 2011. 6, 7

[8] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science. Springer International Publishing, Oct. 2016. 1

[9] M. Burenius, J. Sullivan, and S. Carlsson. 3d pictorial structures for multiple view articulated pose estimation. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3618–3625, 2013. 2

[10] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 1, 2, 3

[11] C.-H. Chen and D. Ramanan. 3d human pose estimation = 2d pose estimation + matching. pages 5759–5767, 07 2017. 2

[12] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou. Fast and robust multi-person 3d pose estimation from multiple views. *CVPR*, 2019. 1, 2, 7, 8

[13] 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. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3810–3818, 2015. 1, 2

[14] S. Ershadi-Nasab, E. Noury, S. Kasaei, and E. Sanaei. Multiple human 3d pose estimation from multiview images. *Multimedia Tools and Applications*, 77:15573–15601, 2017. 2, 7

[15] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79, 2004. 2

[16] J.-Y. Guillemaut and A. Hilton. Joint multi-layer segmentation and reconstruction for free-viewpoint video applications. *International Journal of Computer Vision*, 93:73–100, 2010. 7

[17] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 7

[18] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016. 1, 2

[19] H. Izadinia, I. Saleemi, W. Li, and M. Shah. (mp)2t: Multiple people multiple parts tracker. In *ECCV*, 2012. 5

[20] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. S. Godisart, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1, 2

[21] M. Kocabas, S. Karagoz, and E. Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *ECCV*, 2018. 2

[22] C. Malleson, M. Volino, A. Gilbert, M. Trumble, J. Collomosse, and A. Hilton. Real-time full-body motion capture from video and imus. In *2017 Fifth International Conference on 3D Vision (3DV)*, 2017. 2, 7

[23] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: real-time 3d human pose estimation with a single rgb camera. *ACM Trans. Graph.*, 36:44:1–44:14, 2017. 1, 2

[24] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking. *CoRR*, abs/1603.00831, 2016. 7

[25] J. R. Mitchelson and A. Hilton. Simultaneous pose estimation of multiple people using multiple-view cues with hierarchical sampling. In *BMVC*, 2003. 2

[26] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1913–1921, 2015. 2

[27] L. Pishchulin, M. Andriluka, P. V. Gehler, and B. Schiele. Poselet conditioned pictorial structures. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2013. 2

[28] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4929–4937, 2016. 2

[29] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014. 2

[30] L. Sigal, M. Isard, H. W. Haussecker, and M. J. Black. Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision*, 98:15–48, 2011. 1, 2

[31] J. Song, L. Wang, L. V. Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5563–5572, 2017. 2

[32] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. *2011 International Conference on Computer Vision*, pages 951–958, 2011. 2

[33] D. Tome, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2

[34] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014. 2

[35] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *CVPR*, 2017. 6

[36] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll. Recovering accurate 3D human pose in the wild using IMUs and a moving camera. In *European Conference on Computer Vision (ECCV)*, volume Lecture Notes in Computer Science, vol 11214, pages 614–631. Springer, Cham, Sept. 2018. 2

[37] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732, 2016. 2