Multiview 3D Markerless Human Pose Estimation from OpenPose Skeletons

Maarten Slembrouck[1](✉) , Hiep Luong[1], Joeri Gerlo[3], Kurt Schütte[2],
Dimitri Van Cauwelaert[1], Dirk De Clercq[3], Benedicte Vanwanseele[2],
Peter Veelaert[1], and Wilfried Philips[1]

[1] TELIN-IPI, Faculty of Engineering and Architecture, Ghent University-imec,
Ghent, Belgium
`maarten.slembrouck@ugent.be`
[2] Human Movement Biomechanics Research Group, Department of Kinesiology,
KU Leuven, Leuven, Belgium
[3] Department of Movement and Sport Sciences, Ghent University, Ghent, Belgium
`https://ipi.ugent.be`

Abstract. Despite the fact that marker-based systems for human
motion estimation provide very accurate tracking of the human body
joints (at mm precision), these systems are often intrusive or even impos-
sible to use depending on the circumstances, e.g. markers cannot be put
on an athlete during competition. Instrumenting an athlete with the
appropriate number of markers requires a lot of time and these mark-
ers may fall off during the analysis, which leads to incomplete data and
requires new data capturing sessions and hence a waste of time and effort.
Therefore, we present a novel multiview video-based markerless system
that uses 2D joint detections per view (from OpenPose) to estimate their
corresponding 3D positions while tackling the people association prob-
lem in the process to allow the tracking of multiple persons at the same
time. Our proposed system can perform the tracking in real-time at 20–
25 fps. Our results show a standard deviation between 9.6 and 23.7 mm
for the lower body joints based on the raw measurements only. After
filtering the data, the standard deviation drops to a range between 6.6
and 21.3 mm. Our proposed solution can be applied to a large number
of applications, ranging from sports analysis to virtual classrooms where
submillimeter precision is not necessarily required, but where the use of
markers is impractical.

Keywords: Markerless human motion · Joint detection · Multiview

1 Introduction

Current experiments in the field of human motion analysis are often analysed
with marker-based systems such as Qualisys or Vicon. The biggest drawback of
these systems is the time needed to instrument a person with reflective markers.
Moreover, during the data capturing process markers may fall off, rendering that

recording useless. Beside this, such a system with markers cannot be deployed in numerous applications such as virtual classrooms, athlete analysis during competition and many more.

Recent advances in markerless monocular skeleton detection enable new applications that require semi-accurate tracking of body parts. Such markerless systems provide the solution for the above mentioned drawbacks of marker-based systems. Whereas marker-based systems claim submillimeter accuracy for the markers, markerless systems only obtain an accuracy up to a few centimeters. The reason is that a joint (e.g. an ankle) is not always detected at the anatomically correct position. Depending on the clothing, joints might not even be visible and even humans would have a hard time to locate the exact position of the joints from the videos only.

The changes in planar joint angles are often used for movement analysis, e.g. technical performance in sports or basic clinical gait analysis. For this reason a markerless system has its value despite the fact that it cannot accurately measure rotations along the limbs axis. Markerless systems have been around for a while now. Since the early 2000s, research has been going on to find the location of joints in RGB videos. Most of these approaches relied on shape-from-silhouettes and tried to match a detailed kinematic model. Positional errors were typically larger than 10 cm [6,7,15,16]. Later advances obtained 51 to 100 mm positional errors on the joints [12,13]. More recently, the shift to monocular pose extraction enabled more flexible camera setups [8,9]. However, the reported positional errors of these systems are all typically between 56 and 140 mm. The multiview system that we propose goes one step further. Our system can accurately detect 2D joints, which enables us to obtain positional errors between 24.2 and 49.2 mm.

Apart from obtaining a better accuracy, we also aim at improved robustness. To obtain this goal, we use the existing 2D pose extractor of OpenPose [3] and triangulate the joint positions in 3D. However, we noticed a number of issues, such as self-occlusion, switching limbs and misdetected joints by the 2D pose extractor. We handle all three issues in this paper and present a robust system that can be applied in a wide range of applications due to its flexibility in the number of cameras and the scale in which it can be applied. In the results section, we will show promising results that demonstrate this flexibility.

The outline of the paper is as follows. In Sect. 2 we discuss the pose extraction from monocular video. In Sect. 3 triangulation is explained for at least two image points from different cameras. We explain how we match different persons in the people association step in Sect. 4 and in Sect. 5 the fusion from 2D to 3D at a skeleton level is explained. In Sect. 6 we discuss two use cases where a single person is analysed with 3 and 8 cameras.

2   Real-Time Skeleton Detection

Our goals is to have an accurate robust system that runs in real-time, i.e. at a framerate of at least 20 fps. OpenPose [3,4] is one of the deep learning algorithms that provides real-time skeleton data. OpenPose is currently the only

framework to support 25 joint points per person, which makes it very useful to analyse human motion. Our system uses the detected skeletons from OpenPose to reason further about the 3D position of each joint using multiple camera views. Alternative pose extractors were also researched, such as Alpha Pose [10], Cascaded Pyramid Networks [5], Dense Pose [1], PoseFlow [17] and SMPL [2]. However, they only support a limited number of joints to be detected. VNect on the other hand is a close competitor to OpenPose and it also supports for instance the detection of the foot tip, but we were unable to obtain the needed source code to use this framework in our experiments [14]. OpenPose is able to run on an NVidia 1080Ti GPU card at almost 30 fps. We employ two graphics cards in parallel and equally distribute the load over both cards. By stitching multiple images together before feeding it to the OpenPose neural network, we are able to process multiple camera images at a rate of 20–25 fps. More convenient GPUs will run the proposed method between 10 and 15 fps.

In the next section we will discuss how we estimate 3D position from a set of 2D points using least squares triangulation. We need this step for the people association in Sect. 4 as well as the fusion of skeleton point into their 3D positions, taking into account possible misdetections in 2D in Sect. 5.

3   Triangulation

Traditional cameras observe the 3D world by light ray projection on a 2D plane. During this process depth information is lost, which is one of the reasons why it is hard to accurately reconstruct a 3D scene from a single captured image. The use of multiple cameras from multiple viewpoints has proven that 3D reconstruction is possible by using the content of the image and calibration data of the different cameras [11].

The mathematical conversion from a set of 2D points from multiple cameras into a 3D location is often referred to as triangulation. An example of triangulation is shown in Fig. 1. The idea is to estimate the position of point X based on the 2D image positions p_i and fixed known camera calibrations (intrinsic and extrinsic). Due to inaccuracies in the camera calibration and the discretization of the image sensor, the lines will rarely intersect. Therefore, we need to apply an approximated model. In the following paragraphs, we will briefly discuss how we perform triangulation using the minimization of least squares of the distances between the point X and the lines defined by the camera position and the pixel location.

For a single joint, we define two vectors: $\boldsymbol{c}_i$ which represents the camera position of camera C_i and vector $\boldsymbol{a}_i$ which is the vector between the unknown 3D position of the joint $\boldsymbol{x}$ and $\boldsymbol{c}_i$: $\boldsymbol{a}_i = \boldsymbol{x} - \boldsymbol{c}_i$.

Let the ith line be defined by $\boldsymbol{c}_i$ and a unit vector $\boldsymbol{d}_i$. Given the principal point (u_0^i, v_0^i) of camera i, we can calculate $\boldsymbol{d}_i$ as follows:

$$\widetilde{\boldsymbol{d}}_{t,i} = R_i^T \begin{pmatrix} m_{ix}(p_{i,u} - u_0^i) \\ m_{iy}(p_{i,v} - v_0^i) \\ f_i \end{pmatrix},$$
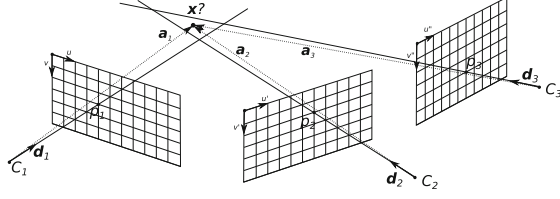
Fig. 1. Triangulation example where we need to estimate the 3D position of point X based on 2D points from multiple cameras. Ideally vectors $\boldsymbol{a}_i$ and $\boldsymbol{d}_i$ would coincide.

$$\boldsymbol{d}_i = \frac{\widetilde{\boldsymbol{d}}_{t,i}}{||\widetilde{\boldsymbol{d}}_{t,i}||},$$

where R_i is the rotation matrix, f_i is the focal length (mm), m_{ix} and m_{iy} represent the horizontal and vertical pixel size on the sensor (mm) and $(p_{i,u}, p_{i,v})$ the projected joint position.

The point to line distance is given by:

$$||\boldsymbol{w}_i|| = \sqrt{\boldsymbol{w}_i \cdot \boldsymbol{w}_i} \quad \text{where} \quad \boldsymbol{w}_i = \boldsymbol{d}_i \times \boldsymbol{a}_i. \tag{1}$$

We now determine the single 3D point $\boldsymbol{x}$ that minimizes the sum of squared point to line distances $\sum_i ||\boldsymbol{w}_i||^2$. This minimum occurs where the gradient is the zero vector ($\boldsymbol{0}$):

$$\nabla \left(\sum_i ||\boldsymbol{w}_i||^2 \right) = \boldsymbol{0}.$$

Expanding the gradient,

$$\sum_i \left(2\boldsymbol{d}_i(\boldsymbol{d}_i \cdot \boldsymbol{a}_i) - 2(\boldsymbol{d}_i \cdot \boldsymbol{d}_i)\boldsymbol{a}_i \right) = \boldsymbol{0}.$$

We found that the coordinates of $\boldsymbol{x}$ satisfy a 3×3 linear system,

$$M\boldsymbol{x} = \boldsymbol{b}, \tag{2}$$

where the kth row (a 3-element row vector) of matrix M is defined as

$$M_k = \sum_i \left(d_{ik}\boldsymbol{d}_i - (\boldsymbol{d}_i \cdot \boldsymbol{d}_i)\boldsymbol{e}_k \right)^T$$

with vector $\boldsymbol{e}_k$ the respective unit basis vector, and

$$\boldsymbol{b} = \sum_i \boldsymbol{d}_i(\boldsymbol{c}_i \cdot \boldsymbol{d}_i) - \boldsymbol{c}_i(\boldsymbol{d}_i \cdot \boldsymbol{d}_i).$$

In practice M is almost always not singular. However, in rare circumstances, M could be singular. For example, we could have a system with only two cameras facing each other and a joint close to the line joining the 2 projection centres.

Since this situation is highly unlikely, and can be easily avoided, we exclude it. We use Gaussian elimination to find x in Eq. 2.

Note that self-occlusion of a joint is automatically solved by not taking into account a viewpoint that did not detect the joint. However, at least two cameras need to observe a joint to report a position.
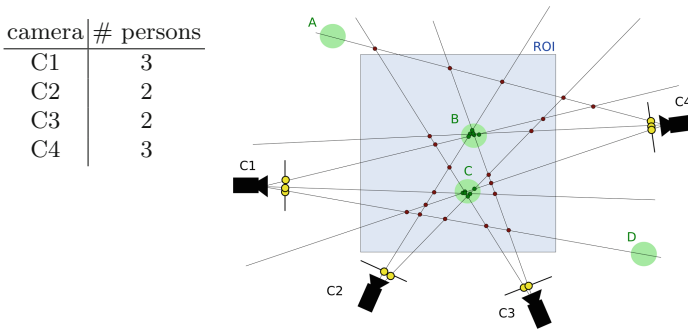

camera	# persons
C1	3
C2	2
C3	2
C4	3

Fig. 2. Clustering of pairwise camera point reconstructions to find different people in the scene. The table indicates the number of detected skeletons in each view. Persons B and C are found, Persons A and D are located outside the region on interest (ROI).

4   People Association

Triangulation of a person's joints relies heavily on the assumptions that the 2D point correspondences in multiple views belong to the same 3D point. However, in case of multiple persons in the area of interest, these correspondences are not so easy to find due to occlusion and possible inaccurate 2D joint positions. As illustrated in Fig. 2.

Let us consider a simple camera network with 4 cameras (Fig. 2). For clarity, we have reduced all skeletons to a single point in the figure. Note that each of these points actually represent 25 joints. The number of possible matches between the different 2D skeletons becomes $(3 + 1)(2 + 1)(2 + 1)(3 + 1) = 144$. We add 1 for each view because none of the skeleton views may correspond to a 3D skeleton in the area of interest. From the 144 combinations, there are also some combinations that cannot result in a skeleton reconstruction since a minimum of two points from different views is required for triangulation.

There is no need to test all 25 individual joints from the same skeleton combinations. To tackle the people association problem, we calculate pairwise correspondences between skeletons from different viewpoints. We assume that joints from a detected skeleton belong to the same person and the spine is correctly detected which is the case in our applications. Therefore, we only consider the spine of a person, defined by the neck and midhip joint (other joints may be chosen for a different application). Valid combinations are those that correspond to

a low point to line distance between the triangulated point and the line defined by its 2D detection and the camera position for both the neck and the mid-hip (in the example, two non-parallel lines will always interest. In 3D however, these lines rarely intersect resulting in a reprojection error). In a second phase, the valid matches are clustered based on the 3D distances between the detected spines. 2D skeletons that have been clustered are removed from the search space to avoid that these combinations are again matched with other skeletons later on. Therefore, multiple persons can be calculated from the same set of frames. Additional constraints concerning the region of interest (ROI) may be used to reject persons that are detected in multiple cameras, but are not inside the ROI.

5   Fusion of 2D Joint Projections into 3D Joints

The conversion to a 3D point from multiple 2D joint projections in different views has been discussed in detail above. However, a number of difficulties needs to be addressed, specifically when it comes to pose estimation.

Triangulation supposes that the detected 2D points are accurate. A pose extractor, such as OpenPose, provides a confidence score in range 0.0–1.0 for each of the detected joints. Usually the position of joints is rather accurate. However, when the confidence score of joints is low (e.g. below 0.2), we noticed that we better discard these points in the triangulation process. Another issue with the pose extractor of OpenPose is confusion in a sense of the left and right extremities of a person's body. We especially noticed this problem with the legs. In most cases, the left leg is detected on the left side of the person, but sometimes the left leg is confused with the right leg, or both legs are detected inside the same leg depending on the pose of the person of interest. Both issues demand a suitable solution to avoid discrete changes in the spatio-temporal domain. With a limited number of cameras, it is not always clear what the correct solution should be, especially when multiple cameras suffer from left/right confusion at the same time. We need to be careful to swap the limbs in the correct view and not to pose swapping on correct views. In that case the 3D positions of both legs are correct, with a low point to line distances, but the label left/right might be switched, causing inferior results. Therefore spatio-temporal tracking offers a suitable solution. Figure 3 shows errors that occur frequently in our datasets.

5.1   Handling Limb Ambiguities

After reconstructing a skeleton, we calculate the point to line distances between the reconstructed joint position X_j and the line defined by the camera position C_j and the image location on the image sensor p_{ij} and store them in matrix D which has n rows and m columns where n is the number of cameras and m the number of joints. Only the following are considered in this matrix because they handle switching legs: LHip, LKnee, LAnkle, LHeel, LBigToe, LSmallToe, RHip, RKnee, RAnkle, RHeel, RBigToe and RSmallToe. The same can be done for the arms of a person. We define matrix D as
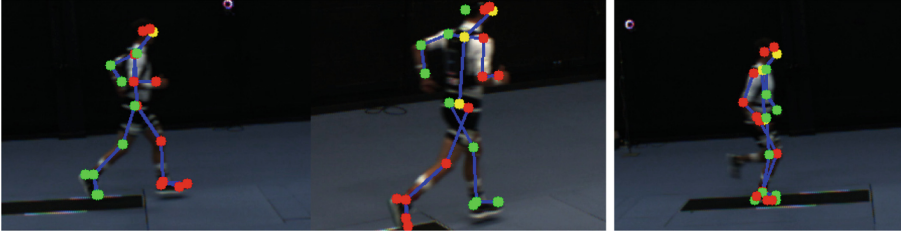
Fig. 3. Misdetected joints causing difficulties in the 3D matching. OpenPose confused between left (red) and right (green) in two different views captured at the same time and both the left and right leg are detected inside the physical right leg, while the physical left leg remains undetected. (Color figure online)

$$D = \begin{bmatrix} ||\boldsymbol{w}_{11}||^2 & ||\boldsymbol{w}_{12}||^2 & ||\boldsymbol{w}_{13}||^2 & \dots & ||\boldsymbol{w}_{1m}^2|| \\ ||\boldsymbol{w}_{21}||^2 & ||\boldsymbol{w}_{22}||^2 & ||\boldsymbol{w}_{23}||^2 & \dots & ||\boldsymbol{w}_{2m}^2|| \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ ||\boldsymbol{w}_{n1}||^2 & ||\boldsymbol{w}_{n2}||^2 & ||\boldsymbol{w}_{n3}||^2 & \dots & ||\boldsymbol{w}_{nm}^2|| \end{bmatrix},$$

where $||\boldsymbol{w}_{ij}||^2$ is the squared point to line distance between the 3D joint position $\boldsymbol{x}_j$ and its detected position on camera j (cfr. Eq. 1). We choose point to line distances over projection errors because the former takes into account the distance between the camera and the detected point.

5.2   Minimizing the Squared Point to Line Distances in Matrix D

Our goal is to remove all limb ambiguities so that we obtain the minimum $\sum_i \sum_j ||\boldsymbol{w}_{ij}||^2$. In order to facilitate real-time processing, we limit the combinations. Some are more likely than others e.g. chances are rather small that the limbs of a skeleton have been switched by more than half of the cameras. Therefore, we constrain the search space to reduce the number of limb reassignments. We are satisfied when all values in D are below a certain threshold T_m (maximum allowed point to line distance). We also use an additional threshold T_u with $T_m < T_u$ to detect and to cope with extreme cases. Both thresholds are not very sensitive. We found that $T_m = 100$ mm and $T_u = 500$ mm correctly fixed the issues with the 2D skeleton detection.

Figure 4 illustrates how the reconstruction is made while coping with switching limbs and misdetected joints. The algorithm starts with the calculation of matrix D. We first check if the maximum value in D is lower than T_u^2. If this is the case, we immediately arrive at the final reconstruction. If this is not the case, we verify if the number of iterations t is smaller than $I_{\max}$. If that is the case, we decide not to investigate any longer to maintain the real-time processing time of the algorithm. Depending on the application, we choose to leave out the joints from the final reconstruction to make sure that no faulty positions are returned.
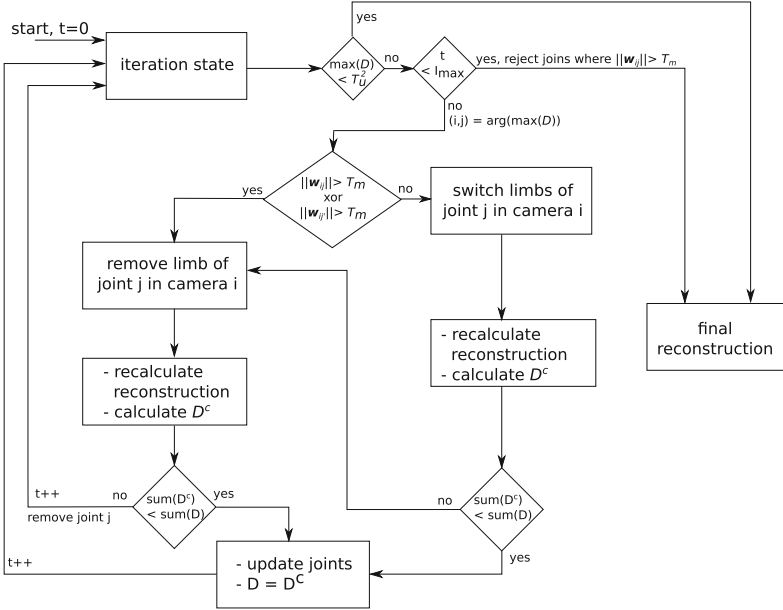
Fig. 4. Flowchart of the proposed algorithm. Joint j' represents the opposite joint j.

At each iteration the following steps are taken: when only one of the two legs has a point to line distance higher than T_m^2, we are almost certain that both legs are detected inside the same leg, meaning one correct location and one incorrect location. Therefore, we decide to test the point to line distances for the reconstruction without the positions of this leg of that particular camera. If both x_{ij} and $x_{ij'}$ are higher than T_m^2, we switch the limbs first. If that does not lead to a better reconstruction, we ignore the limb as well. In all cases, the reconstruction is only accepted, when the sum of the errors of the new candidate joints D^C decreases: $\mathrm{sum}(D^C) < \mathrm{sum}(D)$.

6   Experiments

Two experiments were conducted on two different locations which both were recorded with vision-based cameras and infra-red cameras. The marker-based camera systems (Qualisys and Vicon) have a theoretical submillimeter precision for the marker positions. However, we should keep in mind that due to marker/soft tissue movement it is unlikely that submillimeter precision is reached for the calculation of joint center positions.

The first dataset was recorded at the Sports Science Laboratory Jacques Rogge (SSL-JR) at Ghent University. The vision-based system consisted out of

seven 4.5 MP cameras (Manta G-046C, AVT, Stadtroda, Germany). The person of interest ran in a straight line, always in the same direction at different speeds ranging from 2.1 to 5.1 m/s (Fig. 5). The camera images are captured synchronously by two computers at 67 Hz. The running length that can be captured is around 11 m. The infra-red based motion capture system consisted out of ten 1.3 MP cameras (Oqus3+, Qualisys AB, Göteborg, Sweden) operating at a frame rate of 250 Hz. The cameras were fixed to the lab walls, uniformly distributed to measure 4 m of the running length, with a distance to the center of the volume ranging from 3.5 to 7 m. In total 88 Passive IR-reflective 12 mm-sized spherical markers were attached to the subject body and used for full body modeling in Visual 3D software (C-Motion Inc., Germantown, USA). Joint center coordinates of the ankles, knees and hips were exported for comparison. The wand calibration of the setup showed a standard deviation on measured distances of 0.4 mm.
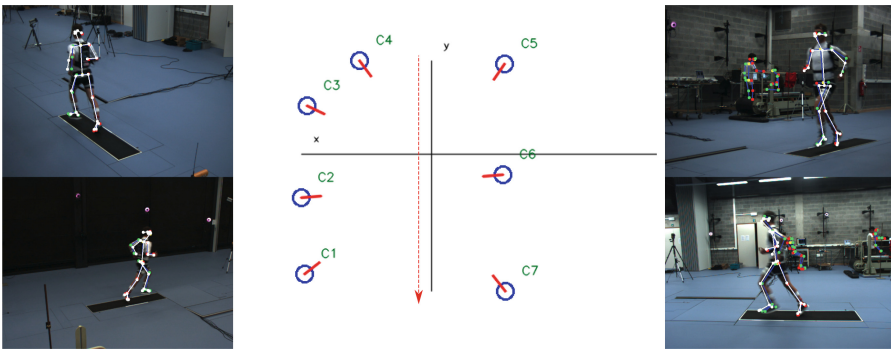

Fig. 5. Camera setup in the Sport Science Lab Jacques Rogge in Ghent on the left (SSL-JR dataset). The red arrow indicated the running direction. On the right we show two camera images with detected 3D skeleton (white) on top. The yellow and cyan lines represent the people association. (Color figure online)

The second dataset was recorded in Leuven, Belgium. Only three 4.5 MP cameras (Manta G-046C, AVT, Stadtroda, Germany) were used operating at a frame rate of 50 Hz. The cameras were located closer to the person of interest in comparison with the first dataset because the measuring volume was only $3 \times 3 \times 2$ m. The person in this dataset is executing stationary movements such as squats and clocks (Fig. 6). A fixed ten camera Vicon system (Vicon MX T20, VICON Motion Systems Ltd., Oxford, UK) supplemented with three additional portable Vicon Vero cameras (Vicon Vero v1.3, VICON Motion Systems Ltd., Oxford, UK) were used. All cameras were sampled at 100 Hz and utilizing a measurement error of 1 mm. In addition, ground reaction forces were collected using two AMTI OR 6 Series force plates sampled at 1000 Hz (Optima, Advanced Mechanical Technology, Inc., Watertown, USA). These force plates were used to determine initial ground contact during the side cut maneuver and check

the execution of the clock for the Vicon system. A single researcher placed 39 retro-reflective markers on the participant using palpation to identify the correct attachment site. Markers were placed as shown in Fig. 6 on the trunk, pelvis and both legs and feet, in order to collect kinematic data for the trunk, hip, knee and ankle.
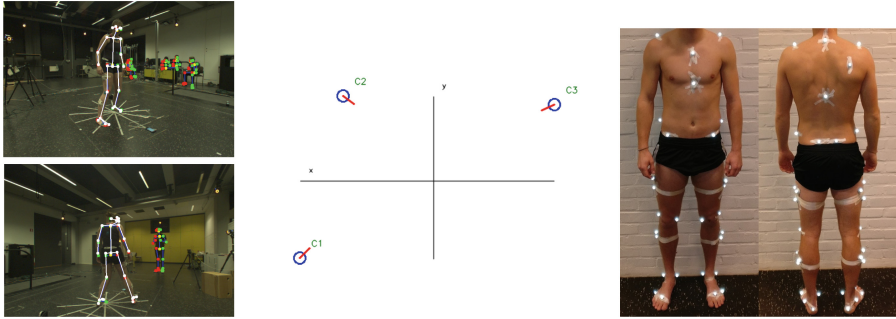

Fig. 6. 3-camera setup in Leuven positioned approximately 2.5 m from the person.

First, we take a look at the graphical representation of the individual results, after which we will evaluate results averaged over all sequences and the spread that can be found in these measurements. We filtered the raw measurements in the spatio-temporal domain from the frame by frame results with a Hanning window of length 7. Such operation slightly improves the results. Figure 7 shows a typical graph produced by the proposed system. We see that the proposed system follows the marker-based positions rather accurately.

Table 1 shows the accuracy averaged per dataset, while Fig. 8 shows the distribution of these numbers. We may conclude that spatio-temporal filtering improves the results by decreasing the standard deviation and average positional error between 1 and 3 mm. For the second dataset we notice an offset in positional errors for a number of joints. The limited number of cameras is most likely the cause of this. Also the cameras in this setup are not entirely evenly distributed around the person of interest. However, the experiments show that even with a limited number of cameras, the proposed method performs well. The offsets between the marker-based and proposed system, bares little significance because the standard deviation is small we reliably detects position changes of the different joints even in case of self-occlusion.
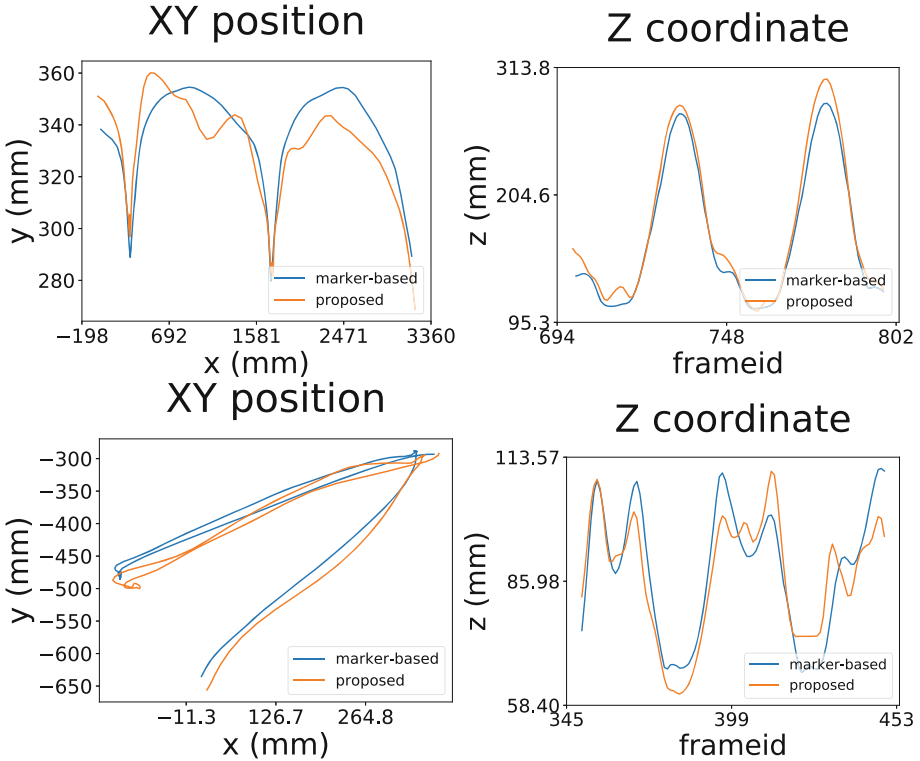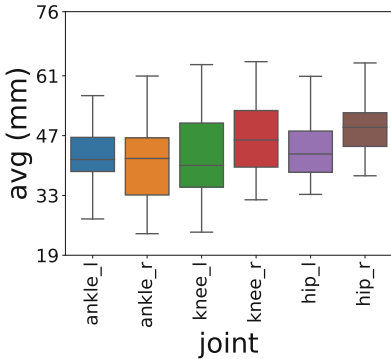
Fig. 7. Typical result of the positions of an ankle joint (top row: SSL-JR dataset, bottom row IPLAY-Leuven dataset). Note the different scales in each of the graphs.
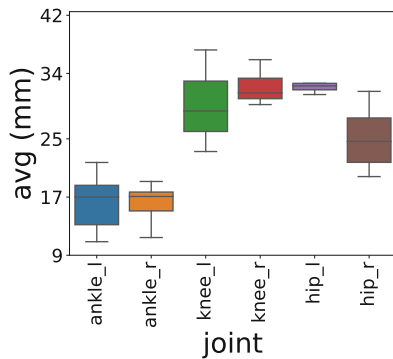

Fig. 8. Average positional error between the marker-based positions and the estimated 3D position for all 33 sequences (SSL-JR) and 9 sequences (IPLAY).

Table 1. Positional errors and standard deviation averaged over 33 sequences (SSL-JR) and 9 sequences (IPLAY-Leuven dataset). All measurements are in mm.

	joint	Unfiltered		Filtered	
		avg	stddev	avg	stddev
S	ankle_l	42.1	23.7	40.6	21.3
S	ankle_r	41.2	21.7	38.8	19.4
L	knee_l	42.2	20.8	40.4	19.1
J	knee_r	46.3	19.9	44.8	18.3
R	hip_l	44.4	18.8	41.6	15.5
	hip_r	50.7	15.2	49.2	12.5
I	ankle_l	19.4	11.9	18.7	8.2
P	ankle_r	16.7	9.6	15.9	6.6
L	knee_l	30.3	10.6	29.8	7.9
A	knee_r	32.6	12.4	32.1	10.0
Y	hip_l	34.2	15.7	33.5	10.9
	hip_r	26.1	14.7	25.1	11.0

7   Conclusion

In this paper we presented a fast and reliable way to convert 2D OpenPose skeleton detections from multiple camera views into 3D skeletons. Our proposed method copes with misdetected joints and switching limbs to extract reliable 3D tracking data for 25 joints of the human body. During our experiments we found that the positional error for the lower limbs are between 15.9 and 49.2 mm and the standard deviation between 6.6 and 21.3 mm. We compared our system to marker-based systems, which claim submillimeter accuracy. The reported accuracy is not as precise as marker-based systems, but much more flexible and can be used in applications which are satisfied with an accuracy of a few millimeter such as entertainment applications, macro body analysis, virtual classrooms...

References

1. Alp Güler, R., Neverova, N., Kokkinos, I.: DensePose: dense human pose estimation in the wild. In: CVPR, pp. 7297–7306 (2018)

2. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it SMPL: automatic estimation of 3D human pose and shape from a single image. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 561–578. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_34

3. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. arXiv (2018)

4. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: CVPR, pp. 7291–7299 (2017)

5. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: CVPR, June 2018

6. Cheung, G.K.M., Baker, S., Hodgins, J., Kanade, T.: Markerless human motion transfer. In: 3DPVT, pp. 373–378, September 2004

7. Cheung, K., Baker, S., Kanade, T.: Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In: CVPR. IEEE (2003)

8. Du, Y., et al.: Marker-less 3D human motion capture with monocular image sequence and height-maps. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_2

9. Elhayek, A., Kovalenko, O., Murthy, P., Malik, J., Stricker, D.: Fully automatic multi-person human motion capture for VR applications. In: Bourdot, P., Cobb, S., Interrante, V., kato, H., Stricker, D. (eds.) EuroVR 2018. LNCS, vol. 11162, pp. 28–47. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01790-3_3

10. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: RMPE: regional multi-person pose estimation. In: ICCV (2017)

11. Hartley, R.I., Sturm, P.: Triangulation. In: Hlaváč, V., Šára, R. (eds.) CAIP 1995. LNCS, vol. 970, pp. 190–197. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60268-2_296

12. Hofmann, M., Gavrila, D.M.: Multi-view 3D human pose estimation combining single-frame recovery, temporal integration and model adaptation. In: CVPR, pp. 2214–2221, June 2009

13. Huo, F., Hendriks, E., Paclik, P., Oomes, A.H.: Markerless human motion capture and pose recognition. In: 2009 10th Workshop on Image Analysis for Multimedia Interactive Services, pp. 13–16. IEEE (2009)

14. Mehta, D., et al.: VNect: Real-time 3D human pose estimation with a single RGB camera. ACM Trans. Graph. **36**(4), 44 (2017)

15. Rosenhahn, B., Kersting, U.G., Smith, A.W., Gurney, J.K., Brox, T., Klette, R.: A system for marker-less human motion estimation. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 230–237. Springer, Heidelberg (2005). https://doi.org/10.1007/11550518_29

16. Saboune, J., Charpillet, F.: Markerless human motion capture for gait analysis. arXiv (2005)

17. Xiu, Y., Li, J., Wang, H., Fang, Y., Lu, C.: Pose flow: efficient online pose tracking. In: BMVC (2018)