ElliPose: Stereoscopic 3D Human Pose Estimation by Fitting Ellipsoids

Christian Grund
AISC GmbH
cg@aisc.digital

Julian Tanke
University of Bonn
tanke@iai.uni-bonn.de

Juergen Gall
University of Bonn
gall@iai.uni-bonn.de

Abstract

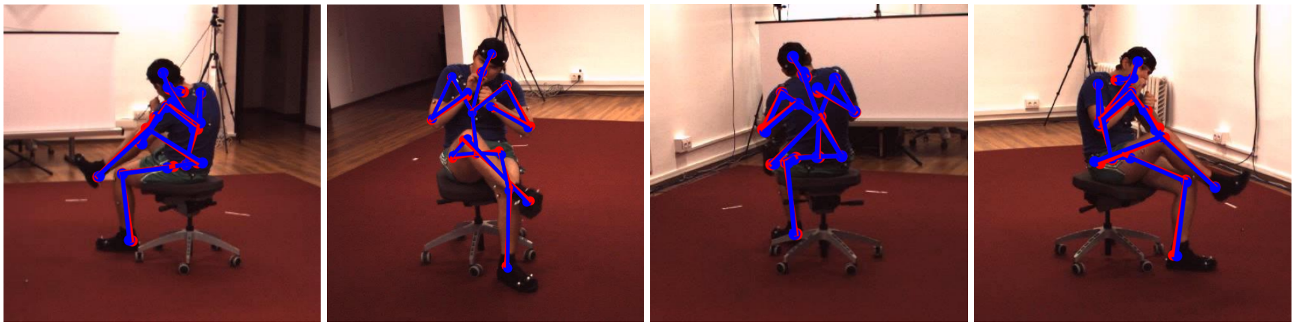
One of the most relevant tasks for augmented and virtual reality applications is the interaction of virtual objects with real humans which requires accurate 3D human pose predictions. Obtaining accurate 3D human poses requires careful camera calibration which is difficult for non-technical personal or in a pop-up scenario. Recent markerless motion capture approaches require accurate camera calibration at least for the final triangulation step. Instead, we solve this problem by presenting ElliPose, Stereoscopic 3D Human Pose Estimation by Fitting Ellipsoids, where we jointly estimate the 3D human as well as the camera pose. We exploit the fact that bones do not change in length over the course of a sequence and thus their relative trajectories have to lie on the surface of a sphere which we can utilize to iteratively correct the camera and 3D pose estimation. As another use-case we demonstrate that our approach can be used as replacement for ground-truth 3D poses to train monocular 3D pose estimators. We show that our method produces competitive results even when comparing with state-of-the-art methods that use more cameras or ground-truth camera extrinsics.

1. Introduction

Advances in augmented and virtual reality make this technology more and more prevalent in modern industry [33] and consumer products [1, 41]. One of the most relevant tasks for AR/VR applications is the interaction of virtual objects with real humans where 3D human poses can be obtained using consumer cameras (e.g. smartphones) in a stereoscopic setup. Recently, many markerless motion capture methods [2, 3, 9, 14, 27, 29, 30, 45, 47, 57, 58] have been proposed. However, these methods require accurate camera calibration which is difficult to obtain for laymen or in pop-up settings.

To alleviate this problem we present the ellipsoidal stereoscopic 3D human pose estimation algorithm ElliPose, a stereoscopic 3D pose estimation approach which iteratively and jointly estimates the human and camera pose. Given 2D human pose estimations [6, 8, 23, 26, 28] for uncalibrated cameras our approach is capable of generating 3D poses and calibrate the cameras. We exploit the fact that the length of a bone cannot change and thus the translation invariant vector describing this bone has to lie on the surface of a sphere when observed over time. Triangulating 2D poses from a stereoscopic setup produces distorted 3D poses, which arises from an inaccurate estimate of the camera extrinsics, especially when the cameras are positioned closer to each other and the objects lie along a steep viewing angle [16]. In contrast to current state-of-the-art approaches [25, 31], which require the full extrinsic calibration, our approach is faster and easier to perform. As our method is fully algorithmic it can also easily be transferred to other detection tasks such as for animals, given an appropriate detection framework [5].

In our work we evaluate the distortion and use this information to simultaneously correct the camera location and the 3D pose prediction. By analysing a sequence of poses, we can fit an ellipsoid to the relative trajectory of the body joints and thus estimate the global distortion. The detections can then be undistorted and the camera location can be corrected. Aside from the 2D pose estimation, where any off-the-shelf method is sufficient, our approach is purely algorithmic: For a stereoscopic setup we first estimate the Essential Matrix E using point correspondences of the 2D pose detections in each camera for each time step. We than use E to obtain 3D poses for each time frame. We subsequently estimate the relative trajectories of the body joints and use a multi-scale variation of the ellipsoid fitting algorithm by Turner *et al.* [48] to fit a scale invariant ellipsoid and thus estimate the global distortion. After deforming the coordinate system in such a way that the ellipsoid gets closer to a ball shape, the camera positions are fitted to the new 3D joint locations. Our experiments show that moving the 3D point locations only slightly and re-estimating the cameras multiple times performs better than warping the ellipsoid into a ball shape directly. We thus use an iterative approach by alternating between estimating ellipsoids and recalibrating the cameras. A predicted 3D pose can be seen in Figure 1.



(a) Camera 1: Right-Back (b) Camera 2: Right-Front (c) Camera 3: Left-Back (d) Camera 4: Left-Front

Figure 1: 3D human pose prediction of 1501st frame of subject 11 performing "Smoking 2" projected onto the real cameras on Human3.6M [19]. Predictions are blue, ground truth is red. The pose was triangulated using the front camera pair (cameras 2 and 4).

Our approach produces highly competitive 3D pose estimation results for stereoscopic pose estimation while not requiring calibrated cameras. Using the front camera set of Human3.6m [19], we are able to reduce the Mean-Per-Joint-Position-Error (MPJPE) from 75.7 mm (Baseline) to 40.9 mm (ElliPose) on the validation set. Furthermore we are able to reduce the position and scale invariant MPJPE (PMPJPE) of the state-of-the-art approach MetaPose [49] by over 30%. As another application we demonstrate that our approach can be used to acquire training data for neural 3D human pose estimation. We train the fully supervised approach by Pavllo *et al.* [39] using our 3D pose estimates instead of ground truth data where we greatly outperform any other semi-supervised approach by the authors [39].

2. Related Work

2D Human Pose Estimation: Recent neural 2D human pose estimation methods can be split into two categories: top-down and bottom-up. Top-down approaches first search for persons in the image using a person detector like Faster-RCNN [15, 24, 38, 44], R-FCN [11, 53], Feature Pyramid Networks [8, 28] or YoloV3 [24, 43]. Subsequently the image patch containing the detected person is used to estimate a heatmap for each joint type, which are merged into a skeleton model [8, 53].

Bottom-up approaches first detect probability heatmaps for each joint type for all persons in the image. Those heatmaps are then merged toward probably multiple person detections. For this association algorithms like linear programs [18, 20, 40], part affinity fields [6, 26] or graph clustering [23] are used.

Monocular 3D Human Pose Estimation: Since 2D images lack depth information, detecting 3D human poses from 2D images is an ill-posed problem [52]. Nonetheless modern algorithms are capable to recover 3D pose information. Library-based strategies use large databases to interpolate best 3D pose fits for predicted 2D poses [7, 56]. Surprisingly, simple MLPs [32] are capable of outperforming library-based approaches by a margin. Further improvements have been shown in algorithms optimizing the inner frame keypoint relations using Long-Short-Term-Memory (LSTM) models [37, 51], Euclidean Distance Matrices (EDM) [35] or Graph Neural Networks [10, 59]. Adding skip-connections over multiple frames in video sequences yield further improvements [17, 39]. To obtain more plausible results, Generative Adversarial Networks (GAN) [50, 55] as well as bone length constraints [21, 39] have been used. Due to the ill-posed nature of the problem, these learning-based methods all suffer when confronted with novel poses.

Multi-View 3D Human Pose Estimation: Many markerless motion capture methods [2, 3, 9, 14, 27, 29, 30, 45, 47, 57, 58] have been proposed. Many approaches are based on 2D predictions, thus perform a 2D-to-3D lifting. Using 2D point locations as input, some approaches use triangulation directly [22] while others perform cross-view optimization before triangulation [31]. Alternatively the probability maps predicted by the 2D human pose estimation approach are used as input and projected into a probability voxel grid [42, 47] which are further improved using Voxel-to-Voxel prediction networks [34]. Other approaches learn 3D poses directly from 2D pose [46] or image data [13, 54]. Most of these methods however require a calibrated camera setup which is difficult to obtain in some settings. Similar to our approach Kocabas [25] use multi-view geometry. Their approach is a monocular 3D pose estimator which is trained on 2D estimates from multiple cameras. Even though their monocular 3D pose estimation results are not comparable to our stereoscopic results, they provide a triangulation baseline including multi-view triangulation accuracies. In MetaPose, Usman *et al.* [49] use the monocular EpipolarPose estimates to generate 3D poses from multiple cameras individually, average the resulting poses and subsequently use a neural bundle adjustment to correct camera locations and pose prediction. TransFusion [31] is a cross-view 2D pose refinement network which is capable of improving the 2D pose predictions using multiple camera angles. Their transformer network is capable of improving other viewing angles also without knowing the camera extrinsics. In their publication they also use the improved 2D pose estimates to triangulate towards a 3D pose. However they use for triangulation the ground truth extrinsics.

3. Method

Inaccuracies in stereoscopic triangulation of 2D poses occur due to two types of error, 2D detection inaccuracies and camera calibration inaccuracies. We address both of these errors by presenting the novel ElliPose algorithm. Our algorithm consists of three stages: *First*, we triangulate detected 2D poses into error-prone 3D poses while simultaneously estimating the camera calibration matrix. For this we place the first camera at the origin and produce a relative estimate for the second camera using the essential matrix. *Second*, we use our novel multi-scale ellipsoid fitting algorithm to correct for camera miscalibration, and *third*, we minimize the bone length inconsistency while also minimizing the reprojection error to overcome 2D pose detection noise. An overview of the algorithm can be seen in Figure 2.

We define the 2D pinhole-camera and the 3D world-coordinate keypoints as

$$\begin{aligned} p_p^{k,t} &= (x_p^{k,t}, y_p^{k,t}) & (1)\\ \text{and } p_w^{k,t} &= (x_w^{k,t}, y_w^{k,t}, z_w^{k,t}), & (2) \end{aligned}$$

respectively, where t describes the timestamp, k the keypoint type and $x^{k,t}$, $y^{k,t}$ and $z^{k,t}$ the coordinates of the keypoint location in their respective coordinate system. Coordinates in world space are subscript with p_w while coordinates in pose space are subscript with p_p. Furthermore, we will define $\mathcal{K} = (k_1, ..., k_{|\mathcal{K}|})$ as the ordered set of all available keypoint types and T as the number of frames in the sequence. Since we are dealing with stereosopoic cameras we further define the first camera as C with camera matrix $\mathtt{P} = \mathtt{KQ}$ where $\mathtt{K}$ is the intrinsic and $\mathtt{Q}$ the extrinsic calibration matrix and the second camera as C' with $\mathtt{P}' = \mathtt{K}'\mathtt{Q}'$.

Stage 1 - Triangulation In the triangulation stage, the synchronized 2D poses are triangulated into 3D poses while simultaneously estimating the camera extrinsics. We first concatenate the points of all keypoint types at any timestamp into two large vectors of points, one corresponding to each camera (Fig. 2.a). To estimate the camera locations we determine the essential matrix $\mathtt{E} = \mathtt{K'}^\top \mathtt{FK}$ with $\mathtt{F}$ being the fundamental matrix. We solve [16]

$$\begin{bmatrix} x_{p'}^{(1)} x_p^{(2)} & x_{p'}^{(1)} y_p^{(1)} & x_{p'}^{(1)} & y_{p'}^{(1)} x_p^{(1)} & y_{p'}^{(1)} y_p^{(1)} & y_{p'}^{(1)} & x_p^{(1)} & y_p^{(1)} & 1 \\ x_{p'}^{(2)} x_p^{(2)} & x_{p'}^{(2)} y_p^{(2)} & x_{p'}^{(2)} & y_{p'}^{(2)} x_p^{(2)} & y_{p'}^{(2)} y_p^{(2)} & y_{p'}^{(2)} & x_p^{(2)} & y_p^{(2)} & 1 \\ & & & & \vdots & & & & \end{bmatrix} \mathtt{F}^{\mathtt{I}} = 0. \tag{3}$$

with $\mathtt{F}^{\mathtt{I}} = [\mathtt{F}^{11}, \mathtt{F}^{12}, \mathtt{F}^{13}, \mathtt{F}^{21}, \mathtt{F}^{22}, \mathtt{F}^{23}, \mathtt{F}^{31}, \mathtt{F}^{32}, \mathtt{F}^{33}]^\top$ and $\mathtt{E} = \mathtt{K'}^\top \mathtt{FK}$ (Fig. 2.b). From this we reconstruct the camera matrices as

$$\mathtt{P} = \mathtt{K}[\mathtt{I}_3|\mathbf{0}] \qquad\qquad \mathtt{P}' = \mathtt{K}'\mathtt{Q}', \tag{4}$$

where $\mathtt{I}_3$ is the 3×3 identity matrix, $\mathtt{Q}' = [R'|t']$ and $\mathtt{E} = [\![t']\!]_\times R'$ and where $[\![\cdot]\!]_\times$ denotes the cross product (Fig. 2.c). Finally we triangulate by solving for

$$\begin{bmatrix} y_p^i \mathtt{P}^{31} - \mathtt{P}^{21} & y_p^i \mathtt{P}^{32} - \mathtt{P}^{22} & y_p^i \mathtt{P}^{33} - \mathtt{P}^{23} & y_p^i \mathtt{P}^{34} - \mathtt{P}^{24} \\ \mathtt{P}^{11} - x_p^i \mathtt{P}^{31} & \mathtt{P}^{12} - x_p^i \mathtt{P}^{32} & \mathtt{P}^{13} - x_p^i \mathtt{P}^{33} & \mathtt{P}^{14} - x_p^i \mathtt{P}^{34} \\ y_{p'}^i \mathtt{P}'^{31} - \mathtt{P}'^{21} & y_{p'}^i \mathtt{P}'^{32} - \mathtt{P}'^{22} & y_{p'}^i \mathtt{P}'^{33} - \mathtt{P}'^{23} & y_{p'}^i \mathtt{P}'^{34} - \mathtt{P}'^{24} \\ \mathtt{P}'^{11} - x_{p'}^i \mathtt{P}'^{31} & \mathtt{P}'^{12} - x_{p'}^i \mathtt{P}'^{32} & \mathtt{P}'^{13} - x_{p'}^i \mathtt{P}'^{33} & \mathtt{P}'^{14} - x_{p'}^i \mathtt{P}'^{34} \end{bmatrix} \begin{bmatrix} x_w^i \\ y_w^i \\ z_w^i \\ w_w^i \end{bmatrix} = \mathbf{0} \tag{5}$$

(Fig. 2.d).

Stage 2 - Multi-scale Ellipsoid Fitting After triangulation we receive a distorted 3D human pose prediction due to an inaccurate $\mathtt{Q}'$. First we calculate the set of bone vectors $\mathcal{V}$ as we will show in Section 3.1 (Fig. 2.e). We then use RANSAC to fit a multi-scale ellipsoid to the bone vectors (see Section 3.1) (Fig. 2.f). The resulting ellipsoid $\mathbf{A}$ is now tested for sphericity. If the ellipsoid is close to a sphere ($\mathbf{A} \approx \mathtt{I}_3$) we exit the stage (Fig. 2.g) without (further) improvement. Otherwise we deform the world along the eigenvectors to clinch the ellipsoid towards a ball shape (Fig. 2.h). More details are provided in Section 3.2. The eigenvalues of $\mathbf{A}$ are the principle axes of the ellipsoid. To undistort the world and thus to bring the ellipsoid into a more spherical shape, we perform a basis transformation into an ellipsoid coordinate system where the principle axes of the ellipsoid are the coordinate system axes. Subsequently we multiply the weighted inverse of the square rooted eigenvalues $\text{EVal}(\mathbf{A})$ to the point coordinates and transform them back into the previous world coordinate system. More details on the undistortion process are provided in Section 3.3.

The now achieved pose is less distorted than the previous pose, but the cameras are not aligned to the pose anymore. Thus we use an iterative approach with the old camera positions as initial guess to realign the cameras. Subsequently we multiply the inverse of the first camera to both cameras and normalize the camera distance to the ground truth camera distance if provided (Fig. 2.i). After this we triangulate the 2D poses to a new 3D pose using the new camera locations (Fig. 2.j) and start over.

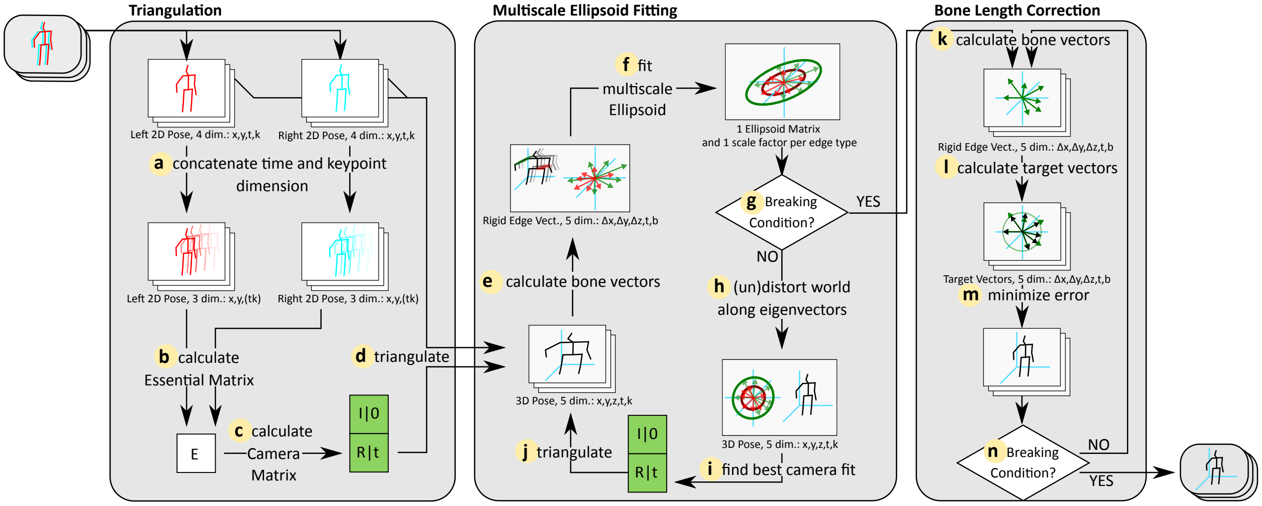

Triangulation
Left 2D Pose, 4 dim.: x,y,t,k
Right 2D Pose, 4 dim.: x,y,t,k
a concatenate time and keypoint dimension
Left 2D Pose, 3 dim.: x,y,(tk)
Right 2D Pose, 3 dim.: x,y,(tk)
b calculate Essential Matrix
d triangulate
E
c calculate Camera Matrix
I|0
R|t
Multiscale Ellipsoid Fitting
f fit multiscale Ellipsoid
1 Ellipsoid Matrix and 1 scale factor per edge type
Rigid Edge Vect., 5 dim.: Δx,Δy,Δz,t,b
g Breaking Condition?
YES
NO
e calculate bone vectors
h (un)distort world along eigenvectors
3D Pose, 5 dim.: x,y,z,t,k
j triangulate
I|0
R|t
3D Pose, 5 dim.: x,y,z,t,k
i find best camera fit
Bone Length Correction
k calculate bone vectors
Rigid Edge Vect., 5 dim.: Δx,Δy,Δz,t,b
l calculate target vectors
Target Vectors, 5 dim.: Δx,Δy,Δz,t,b
m minimize error
n Breaking Condition?
NO
YES

Figure 2: Overview of ElliPose. The approach consists of three stages: triangulation, multiscale ellipsoid fitting and bone length correction. The first stage estimates a pose and camera positions by triangulation, the second stage corrects camera miscalibration due to errors during triangulation and the third stage corrects errors due to 2D detection noise. The triangulation stage gets a pair of 2D poses as input, concatenates the keypoint and time dimension (a) and uses the two resulting 3-dimensional data matrices (x, y and k×t) to calculate the essential matrix (b), infer from this the camera extrinsics (c) and triagulate the pose (d). The second stage calculates relative bone vectors (e) and fits a multiscale ellipsoids to it using different scales for different bone types (f). The eigenvectors of the ellipsoid are used to indicate the worlds distortion and the world is undistorted accordingly (h), subsequently the camera extrinsics are recalibrated using the undistored pose (i) and the pose is retriangulated using the new camera poses. Then the stage repeats. The stage is left after the ellipsoid fitting without pose or camera updating if the estimated ellipsoid is close to a ball shape (g). The final stage corrects the remaining noise by estimating for each bone the current vector (k) and target vector representing this bone with correct length (l). Each point is then relocated such that the difference between new bone and target bone is minimized as well as the reprojection error (m). This procedure repeats for a given number iteration (n), then returns a final 3D pose.

Stage 3 - Bone Length Correction To enforce consistent bone length, we first calculate the bone vectors $\mathcal{V}$ as shown in Section 3.1 as well as their type-wise mean length $l{*}_b$ (Fig. 2.k). Using this information we calculate the target vectors $\vec{v}^{b,t} l^*_b / \|\vec{v}^{b,t}\|$ (Fig. 2.l). Subsequently we minimize the error function in Equation (19) (Fig. 2.m) detailed in Section 3.4. This stage is then repeated for a fixed number of iteration (Fig. 2.n).

3.1. Bone Vectors

We define as bone any pair of keypoints whose distance does not change over time. We identify for the keypoints in the Human3.6m dataset [19] (left hip, right hip), (left hip, center hip), (center hip, right hip), (knee, hip), (ankle, knee), (left shoulder, right shoulder), (left shoulder, center shoulder), (center shoulder, right shoulder), (shoulder, elbow), (elbow, wrist), as well as any pair of points on the head to have this property and define the *bone set* $\mathcal{B} = \{b_1, ..., b_{|\mathcal{B}|}\} \subseteq \mathcal{K}^2$ as the set of those pairs. Let $\mathtt{Z} \in \mathbb{R}^{3T \times |\mathcal{K}|}$ be the pose matrix with

$$\mathtt{Z}=\left[\begin{array}{ccc|c|ccc} x_w^{1,1} & y_w^{1,1} & z_w^{1,1} & & x_w^{1,T} & y_w^{1,T} & z_w^{1,T} \\ \vdots & \vdots & \vdots & \cdots & & \vdots & \\ x_w^{|\mathcal{K}|,1} & y_w^{|\mathcal{K}|,1} & z_w^{|\mathcal{K}|,1} & & x_w^{|\mathcal{K}|,T} & y_w^{|\mathcal{K}|,T} & z_w^{|\mathcal{K}|,T} \end{array}\right] \tag{6}$$

and $\mathtt{B}$ the *bone matrix* where the q-th row is defined so that for each $b_q = (k_i, k_j)$, the i-th column value is 1, the j-th column value is -1 and all other values are 0. The bone vector matrix $\mathtt{V} \in \mathbb{R}^{3T \times |\mathcal{B}|}$ containing all bone vectors is now calculated by

$$\begin{aligned} \mathtt{V} &= \mathtt{BZ} \\ &= \left[\begin{array}{c|c|c} [\ \vec{v}^{1,1\top}\ ] & & [\ \vec{v}^{1,T\top}\ ] \\ \vdots & \cdots & \vdots \\ [\ \vec{v}^{|\mathcal{B}|,1\top}\ ] & & [\ \vec{v}^{|\mathcal{B}|,T\top}\ ] \end{array}\right]. \end{aligned} \tag{7}$$

Furthermore we define the set $\mathcal{V}$ of all bone vectors as

$$\mathcal{V}=\left\{\vec{v}^{1,1},...,\vec{v}^{|\mathcal{B}|,1},\vec{v}^{1,2},...,\vec{v}^{|\mathcal{B}|,2},\ \ \ldots\ \ ,\vec{v}^{1,T},...,\vec{v}^{|\mathcal{B}|,T}\right\}. \tag{8}$$

3.2. Multiscale Ellipsoid Fitting

Our goal is to fit the same ellipsoid to multiple bone trajectories simultaneously despite them having different length. To do so we need the fitting algorithm to have different scaling factors for different bones. As Turner *et al.* [48] show, an origin centered ellipsoid can be expressed as

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} + L = 0 \quad \text{with } \mathbf{A} = \begin{bmatrix} A & D/2 & E/2 \\ D/2 & B & F/2 \\ E/2 & F/2 & C \end{bmatrix} \tag{9}$$

$$\begin{aligned}\Leftrightarrow\ & x^2{+}y^2{+}z^2{-}U(x^2{+}y^2{-}2z^2){-}V(x^2{-}2y^2{+}z^2) \\ & {-}4Mxy{-}2Nxz{-}2Pyz{-}T{=}0\end{aligned} \tag{10}$$

with $(x,y,z)^\top{=}\mathbf{x}$ and

$$\begin{aligned} M&{=}{-}\tfrac{3D}{4(A+B+C)} & N&{=}{-}\tfrac{3E}{2(A+B+C)} & P&{=}{-}\tfrac{3F}{2(A+B+C)} \\ T&{=}{-}\tfrac{3L}{(A+B+C)} & U&{=}{-}\tfrac{3\cdot(A-C)}{(A+B+C)} & V&{=}{-}\tfrac{3\cdot(A-B)}{(A+B+C)} \end{aligned} \tag{11}$$

which can be expressed as a linear equation

$$\mathbf{\Lambda}\mathbf{s}{=}\mathbf{e} \tag{12}$$

where

$$\begin{aligned} \mathbf{s}&{=}[U,V,M,N,P,T]^\top \\ \mathbf{e}&{=}\left[x_0^2{+}y_0^2{+}z_0^2, \dots, x_n^2{+}y_n^2{+}z_n^2\right] \\ \mathbf{\Lambda}&{=}\begin{bmatrix} x_0^2{+}y_0^2{-}2z_0^2,\ x_0^2{-}2y_0^2{+}z_0^2,\ 4x_0y_0,\ 2x_0z_0,\ 2y_0z_0,\ 1 \\ \vdots \\ x_n^2{+}y_n^2{-}2z_n^2,\ x_n^2{-}2y_n^2{+}z_n^2,\ 4x_ny_n,\ 2x_nz_n,\ 2y_nz_n,\ 1 \end{bmatrix} \end{aligned} \tag{13}$$

The entries in $\mathbf{e}$ are the bone vectors for all bones $|\mathcal{B}|$ over all time steps T. This linear system is over-determined and can be solved using least squares. To prevent 2D human pose mis-detections and other outliers from interfering with the result we utilize RANSAC with $n{=}|\mathcal{B}|{+}5$. By inspection of Equation (9) it can be seen that L is the scaling factor of the ellipsoid. Furthermore T is the only parameter in Equation (10) containing L, thus we extend $\mathbf{s}$ to introduce an individual T for each bone type, so

$$\mathbf{s}_{LS}{:=}\left[U,V,M,N,P,T_1,T_2\dots,T_{|\mathcal{B}|}\right]^\top. \tag{14}$$

Analogous we adapt $\mathbf{\Lambda}$ such that the i-th row of $\mathbf{\Lambda}$ is defined as

$$\begin{aligned}\Lambda_i{=}\big[& x_i^2{+}y_i^2{-}2z_i^2,\ \ x_i^2{-}2y_i^2{+}z_i^2,\ 4x_iy_i, \\ & 2x_iz_i,\ 2y_iz_i,\ \xi_{b_1}(w_i),\ \xi_{b_2}(w_i),\ \dots\ \xi_{b_{|\mathcal{B}|}}(w_i)\big]\end{aligned} \tag{15}$$

where the indicator function ξ is defined as

$$\xi_b(w_i){:=}\begin{cases} 1, \text{ if vector } w_i \text{ represents the bone type } b{\in}\mathcal{B} \\ 0, \text{ otherwise.} \end{cases} \tag{16}$$

The definition of $\mathbf{e}$ stays unchanged. Solving this linear system provides an ellipsoid $\mathbf{A}$ and a scale L for each bone.

3.3. Undistorting the World

The eigenvectors and the eigenvalues of the ellipsoid $\mathbf{A}$ are the directions of the principle axes and their respective quadratic length. Since we aim to use the fitted ellipsoid to undistort the world coordinates, we need to transform the world so that the lengths of those axes are normalized.

Let $\underline{\mathtt{Z}}{\in}\mathbb{R}^{T|\mathcal{B}|\times 3}$ be a matrix containing all keypoints as columns. We perform a base transformation on the pose coordinates by multiplying the inverse of the right eigenvector-matrix $\text{EVec}(\mathbf{A})$. Subsequently, we scale the coordinates by multiplying the weighted inverse of the square-rooted eigenvalues $\text{EVal}(\mathbf{A})$ to the corresponding coordinates and transform the coordinates back into the previous base. Thus, the corrected matrix containing all keypoints $\underline{\mathtt{Z}}^*$ is calculated by

$$\begin{aligned}\underline{\mathtt{Z}}^*{=}\text{EVec}(\mathbf{A}){\cdot}\Big((1{-}\gamma)\mathtt{I}_3{+}\gamma\text{diag}\Big(\sqrt{\text{EVal}(\mathbf{A})}\Big)\Big) \\ {\cdot}\text{EVec}(\mathbf{A})^{-1}{\cdot}\underline{\mathtt{Z}}.\end{aligned} \tag{17}$$

3.4. Bone Length Optimization

The length of a given bone should stay consistent for a given sequence. To enforce this we use a bone length consistency optimization. Simultaneously, we want to ensure that the pose still resembles the observed 2D poses, thus we want to keep the projection error low. Thus we want to

$$\begin{aligned}\underset{\{p_w^{k,t}:k\in\mathcal{K}\}}{\text{minimize}}&\left[\sum_{k\in\mathcal{K}}\|\mathtt{P}p_w^{k,t}{-}p_p^{k,t}\|^2{+}\sum_{k\in\mathcal{K}}\|\mathtt{P}'p_w^{k,t}{-}p'^{k,t}_p\|^2\right] \\ &{+}\alpha\left[\sum_{b\in\mathcal{B}}(\|\vec{v}^{b,t}\|{-}l_b^*)^2\right]\end{aligned} \tag{18}$$

where α is a weighting factor. Due to the non-convex nature of the second part of the equation this problem is NP-complete [36].

We circumvent this problem by using an iterative approach and assuming that the direction of the bone vector is approximately correct. Thus, for each iteration step, we define fixed target bone vectors $\vec{w}^{b,t}{=}\vec{v}^{b,t}l_b^*/\|\vec{v}^{b,t}\|$ as vectors with the same direction as the vectors from the previous iteration step but with corrected length.

$$\underset{\{p_w^{k,t}:k\in\mathcal{K}\}}{\text{minimize}}\left[\sum_{k\in\mathcal{K}}\|\mathrm{P}p_w^{k,t}-p_p^{k,t}\|^2+\sum_{k\in\mathcal{K}}\|\mathrm{P}'p_w^{k,t}-p'^{k,t}_p\|^2\right] +\alpha\left[\sum_{b\in\mathcal{B}}\left\|\vec{v}^{b,t}-\vec{w}^{b,t}\right\|^2\right] \qquad (19)$$

3.5. Implementation Details

Camera fitting: As 2D human pose backbone we use the CPN 2D predictions calculated by Pavllo *et al.* [39]. We predict the essential matrix using OpenCV [4] with ground truth intrinsic camera parameters using the RANSAC method with a threshold of 0.005 and a probability of 0.999. The pose is also triangulated using OpenCV [4] with a distance threshold of 1000.

Ellipsoidal Correction: For ellipsoid fitting we use the RANSAC algorithm combination with the convex optimizer CVXPY [12]. We iterate for 2500 steps while terminating early as soon as 100 *good samples* are found. A sample is called *good*, if its inlier score is above $\sqrt{0.25T|\mathcal{B}|}$. The inlier score is calculated as sum of the square roots of the per-bone-type inlier counts. A bone is considered as inlier if a full undistortion of the ellipsoid would result in its length lying within the boundaries of $\delta{=}1{\pm}0.2$.

For world rescaling along the eigenvectors we use a weighting factor of $\gamma{=}0.2$. The new camera positions are computed using the PnPRansac solver of OpenCV [4], iterating 10000 times. An ellipsoid is consideres as approximatly spherical if the elementwise square distance ϵ between $\mathbf{A}$ and the identity matrix is below 0.025. As an aditional breaking condition for stage 2 we stop after at most $n{=}20$ iterations. If this limit is reached we continue with the prediction which achieved the lowest ϵ during all iteration steps.

Bone Length Consistency: We solve the bone length minimization term (19) with a weighting constant of $\alpha{=}20$ using CVXPY [12] for $m{=}5$ iteration steps.

Runtime: Our non-optimized Python implementation takes on average 0.33ms per iteration on an Intel Core i7-7700 CPU. Since we set the maximum number of iterations to 20×2500, the runtime of ellipsoidal fitting for one video sequence is at most 16.5s, independent of the number of frames in the sequence.

4. Experiments

We evaluate our approach on the Human3.6m dataset [19] which consists of 15 actions performed by 7 actors. The actions are recorded in a studio using 4 cameras and a marker-based motion capture system. The four cameras are placed in the corners of an rectangular room so that the front cameras are approximately 9.5 meter from the back cameras apart and the left cameras 3.5 meter from the right. We follow the previous literature [32, 39, 55] by assigning actors S1, S5, S6, S7, and S8 to the training set and actors S9 and S11 to the test set. We skip the sequences "S9 Greeting", "S9 SittingDown1" and "S9 Waiting" for evaluation since those have corrupted poses [22].

Methods	**Used Estimator**	**MPJPE**	**PMPJPE**	**PCK$_{50}$**	**PCK$_{100}$**	**PCK$_{150}$**	**PPCK$_{50}$**	**PPCK$_{100}$**	**PPCK$_{150}$**
Kocabas^{4} [25]	Ground Truth	—	15.1	—	—	—	—	—	—
Kocabas^{4R} [25]	Ground Truth	4.4	2.1	—	—	—	—	—	—
Kocabas^{4R} [25]	ResNet+Deconv	28.4	25.2	—	—	—	—	—	—
TransFusion$^{2\text{Я}}$ [31]	Transformer	35.9	—	—	—	—	—	—	—
TransFusion^{2R} [31]	Transformer	25.8	—	—	—	—	—	—	—
MetaPose^{4} [49]	Kocabas [25]	—	32.0	—	—	—	—	—	—
MetaPose^{r} [49]	Kocabas [25]	—	44.0	—	—	—	—	—	—
Baselinerd	Ground Truth	5.0	2.4	100	100	100	100	100	100
Baselinerd	CPN [39]	38.2	32.0	79.6	96.2	98.4	87.8	97.4	98.9
Baseline^{ld}	CPN [39]	38.1	31.3	79.1	96.5	98.7	88.2	97.8	99.2
Baseline^{fd}	CPN [39]	75.7	32.7	49.2	80.3	90.3	86.3	97.6	99.3
Baseline^{bd}	CPN [39]	75.6	39.6	48.1	77.5	89.6	79.6	94.4	97.7
ElliPoserd	Ground Truth	5.0	2.4	100	100	100	100	100	100
ElliPoserd	CPN [39]	36.9	29.8	80.4	96.8	98.7	90.2	98.0	99.1
ElliPose^{ld}	CPN [39]	35.8	28.7	81.7	97.3	99.0	91.2	98.4	99.4
ElliPose^{fd}	CPN [39]	40.9	26.8	71.9	96.7	99.1	93.7	99.1	99.6
ElliPose^{bd}	CPN [39]	52.1	33.9	62.5	89.2	96.1	86.3	96.6	98.4

Table 1: Results after triangulating 3D points from 2D. MPJPE in mm. PCK$_n$: n in mm, PPCK aligned equivalent to PMPJPE.
^{l}Using cams 3 and 4 (left cams) ^{r}Using cams 1 and 2 (right cams)
^{f}Using cams 2 and 4 (front cams) ^{b}Using cams 1 and 3 (back cams)
^{2}Using any two cams ^{4}Using all cams
^{d}Using GT cam distance for scale ^{R}Using extrinsic cam parameters
$^{\text{Я}}$Using extrinsic cam parameters only for triangulation

As evaluation score we use *M*ean *P*er *J*oint *P*osition *E*rror (MPJPE) and *P*ercentage of *C*orrect *K*eypoints (PCK), following previous works [22, 32, 39, 55]. MPJPE describes the mean error after aligning the prediction with the ground truth data at the root node (usually center hip). Additionally, we report PMPJPE, which further aligns the poses using a rigid transformation. Congruently to the PMPJPE metric we also define a PPCK metric giving the PCK after rigid alignment.

4.1. Stereoscopic 3D Pose Reconstruction

In Table 1 we present results of our ElliPose algorithm in comparison to the state-of-the-art TransFusion [31] and MetaPose [49] as well as to the triangulation baseline of Kocabas [25]. Note that only TransFusion [31] and MetaPose [49] report results on camera pairs while Kocabas [25] uses all 4 cameras. On top of that we present a simple baseline which utilizes only our triangulation step to show the effectiveness of our ellipsoid fitting.

Our method performs best on the left and right camera pair since triangulation preforms better as the angle between the optical axes increases [16]. This also explains the bad performance of the front and back pairs using our baseline

		left	right	front	back
Breaking Condition	$\epsilon \leq 0.01$	35.8	36.7	**40.0**	54.1
	$\epsilon \leq 0.025^{*}$	35.8	36.9	40.9	**52.1**
	$\epsilon \leq 0.05$	**35.4**	36.3	42.9	53.2
	$\epsilon \leq 0.1$	35.9	36.2	49.8	54.0
	$\epsilon \leq 0.2$	36.1	**36.1**	56.7	60.6
	$\epsilon \leq \infty^{\dagger}$	36.2	**36.1**	72.7	72.8
Inlier Threshold	$\delta = 1 \pm 0.025$	36.5	37.6	44.2	54.9
	$\delta = 1 \pm 0.05$	36.0	36.5	42.8	52.7
	$\delta = 1 \pm 0.1$	**35.7**	37.0	41.6	**50.7**
	$\delta = 1 \pm 0.2^{*}$	35.8	36.9	40.9	52.1
	$\delta = 1 \pm 0.3$	35.9	**36.4**	**40.4**	54.2
Update Weight	$\gamma = 0.1$	**35.7**	**36.2**	40.6	**51.1**
	$\gamma = 0.2^{*}$	35.8	36.9	40.9	52.1
	$\gamma = 0.5$	36.3	37.4	**40.3**	53.5
	$\gamma = 1.0$	37.5	38.9	42.9	55.4
Rigid Edge Normalization Iterations	$m = 0$	37.5	38.4	42.5	54.0
	$m = 1$	36.0	36.9	41.0	53.2
	$m = 3$	**35.8**	**36.6**	41.0	**51.9**
	$m = 5^{*}$	**35.8**	36.9	40.9	52.1
	$m = 7$	**35.8**	36.7	**40.4**	**51.9**

Table 2: MPJPE results in mm for our ElliPose algorithm by altering single parameters from the proposed set of parameters using different camera pairs. The proposed parameters (gray) are an inlier threshold $\delta = 1 \pm 0.2$, an update weighting factor of $\gamma = 0.2$, a ellipse breaking condition of $\epsilon \leq 0.025$ and $m = 5$ iterations of bone length normalization.

[*] proposed parameter

[†] skipping ellipsoidal correction entirely thus excepting any error

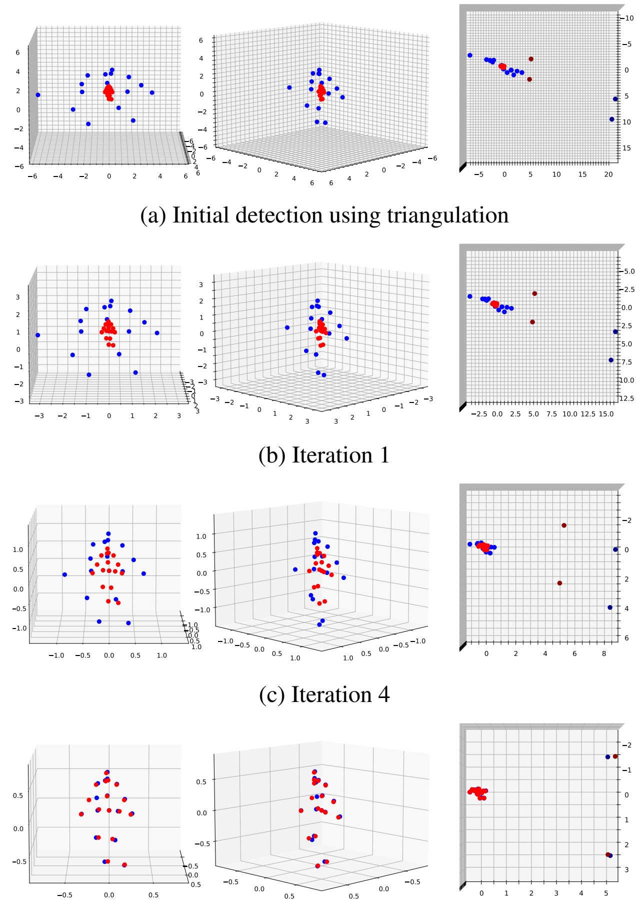
3D Data	**2D Data**	**MPJPE**	**NMPJPE**	**PMPJPE**
^{P}GT (All)	–	**46.8**	**47.1**	**36.5**
^{P}GT (S1,5,6)	CPN (S7,8)	57.7	53.8	—
^{P}GT (S1,5)	CPN (S6,7,8)	63.9	55.3	—
^{P}GT (S1)	CPN (S5,6,7,8)	64.7	61.8	—
ElliPose (All)	–	51.1	48.9	40.0

Table 3: Results on training the fully supervised network of Pavllo *et al.* [39] using only 3D poses generated with ElliPose in comparison to their original approaches using ground truth 3D poses for full-supervision and ground truth poses in combination with CPN estimates and a backprojection loss for semi-supervision.

[P] Results provided by Pavllo *et al.* [39]

approach.

Our ElliPose algorithm performs consequently better than our baseline approach, which proves its effectiveness. Using the right/left camera pairs our algorithm shows an improvement of 1.3/2.3 mm MPJPE and 2.2/2.6 mm PMPJPE. For the front/back pairs the improvement is even larger by 34.8/23.5 mm MPJPE and 5.9/5.7 mm MPJPE. In comparison to other publications our algorithm shows competing or better results using a less constrained setup.


(a) Initial detection using triangulation

(b) Iteration 1

(c) Iteration 4

(d) Iteration 9

Figure 3: Nine iterations of ellipsoidal pose correction. Ground truth is red, prediction is blue. The left and middle columns show different side perspectives of the predicted and the correct pose. The right column shows a top perspective with additional dark blue and dark red points corresponding to the ground truth and the predicted camera positions respectively. Over nine iterations the distortion have been dissolved while the camera positions have been corrected simultaneously.

Using ground truth 2D poses and 4 calibrated cameras Kocabas [25] achieves only 0.6 mm lower MPJPE than our approach. When using no camera extrinsics but still all four cameras they perform 12.7 % worse than our approach.

TransFusion [31] presents a learnable cross-view 2D pose prediction refinement algorithm. In Table 1 we report two versions: one, where camera extrinsics are explicitly provided to the model (TransFusion^{2R}) and one where the camera extrinsics are withheld during training (TransFusion$^{2\text{Я}}$). However, for obtaining the final 3D human pose the ground-truth camera extrinsics are used. Despite them using extrinsics for triangulation, our approach

Method	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing	Purchases	Sitting	Sit'Down	Smoking	Waiting	WalkDog	W'Together	Walking	**Average**
Pavllo^{FS}[39]	**45.2**	**46.7**	**43.3**	**45.6**	**48.1**	**55.1**	**44.6**	**44.3**	**57.3**	**65.8**	**47.1**	**44.0**	**39.0**	**32.8**	**33.9**	**46.8**
Ours	47.6	49.7	47.0	48.8	52.9	59.2	47.0	47.0	65.1	71.9	51.0	47.3	52.6	38.2	39.5	51.1

Table 4: MPJPE per action for training the fully supervised network of Pavllo *et al*. [39]. Pavllo *et al*. has a higher accuracy, however they use ground truth poses for training while we use the estimated poses for training. Semi-supervised results have not been reported per action by Pavllo *et al*.
^{FS} fully-supervised

performs on par with the left camera pair, and only 1.0 mm worse on the right camera pair (MPJPE).

MetaPose [49] is the only state-of-the-art approach which does not use ground-truth camera poses while lifting 2D estimates towards 3D poses. They present results using four cameras and two cameras. For the latter they used the right camera pair. Since they do not use camera extrinsics they are not able to scale the pose thus only provide PMPJPE metrics. We outperform their method on the right camera pair on PMPJPE by 14.2 mm.

We further present PCK results and show that 90.2 % or 91.2 % of our predicted joint locations lie within a margin of 5 cm when using the right or left camera pair and neglecting scale and orientation.

4.1.1 Ablations

In Table 2 we analyze the impact of all hyper-parameters of our method. If not noted otherwise, we set the inlier threshold range to δ=1±0.2, the update weighting factor to γ=0.2, the ellipse breaking condition to $\epsilon \leq 0.025$ and the number of iterations of bone length normalization to m=5. We observe that setting the breaking condition ϵ too large results in a significant performance drop on camera views with steep angles while setting the value lower than ϵ=0.025 does not improve the result. When setting the RANSAC inlier threshold δ we find that δ=1±0.1 performs better on the left/back cameras while δ=1±0.3 performs better on the right/front cameras. We thus choose δ=1±0.2. Setting the update weight δ=1 completely undistorts the ellispoid but results in over-correction. The smaller we set the update weight the more accurate are our results but the optimization requires more time. Thus, for performance reasons we set γ=0.2. Similarly, we set m=5 as more iterations improve the results only slightly but increase the runtime.

4.1.2 Qualitative Results

Qualitative results can be seen in Figure 1. Furthermore, Figure 3 shows multiple steps of the ellipsoidal correction in the ElliPose algorithm. Figure 3a shows the initial, highly distorted, triangulation as it is generated by the first stage, with predictions in blue and ground truth points in red. The initial prediction largely failed due to noisy 2D pose predictions. The right column shows the scene from the top view including dark blue and dark red points representing the estimated and the true camera locations, respectively. Figure 3b-d illustrate the progress of our ellipsoidal correction. As we can see the pose strongly improves while the camera locations get closer and closer to the ground truth location.

4.2. Monocular 3D Pose Estimation

We show that our estimated 3D poses can be straightforward used for training existing 3D monocular pose estimation networks if annotated 3D ground-truth human poses are missing. We train the 3D pose estimator by Pavllo *et al*. [39] by replacing the ground truth training data with the 3D poses generated by the ElliPose algorithm. The results can be seen in Table 3. Our approach outperforms any semi-supervised approach by Pavllo [39].

5. Conclusion

We presented ElliPose, a stereoscopic 3D human and camera pose estimation algorithm. It fits ellipsoids to detected bones over time to iteratively refine camera and 3D human pose. ElliPose performs competitively compared to state-of-the-art methods which either use more cameras or utilize ground-truth camera positions for 3D human pose estimation. Our approach can be easily set up by inexperienced users and it can even be used as a replacement for ground-truth 3D poses to train 3D monocular pose estimation models, outperforming existing semi-supervised methods.

Acknowledgment

The work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) GA 1927/5-2 (FOR 2535 Anticipating Human Behavior).

References

[1] Mir Suhail Alam, Malik Arman Morshidi, Teddy Surya Gunawan, Rashidah Funke Olanrewaju, and Fatchul Arifin. Pose estimation algorithm for mobile augmented reality based on inertial sensor fusion. *International Journal of Electrical & Computer Engineering (2088-8708)*, 12(4), 2022.

[2] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial

structures revisited: Multiple human pose estimation. *Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[3] 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*, 2014.

[4] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[5] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. In *International Conference on Computer Vision*, 2019.

[6] 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.

[7] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7035–7043, 2017.

[8] 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.

[9] Hau Chu, Jia-Hong Lee, Yao-Chih Lee, Ching-Hsien Hsu, Jia-Da Li, and Chu-Song Chen. Part-aware measurement for robust multi-view multi-human 3d pose estimation and tracking. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[10] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2262–2271, 2019.

[11] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29, 2016.

[12] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 2016. To appear.

[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/CVF Conference on Computer Vision and Pattern Recognition*, pages 7792–7801, 2019.

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

[15] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2334–2343, 2017.

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

[17] 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.

[18] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European conference on computer vision*, pages 34–50. Springer, 2016.

[19] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[20] Umar Iqbal and Juergen Gall. Multi-person pose estimation with local joint-to-person associations. In *European conference on computer vision*, pages 627–642. Springer, 2016.

[21] Umar Iqbal, Pavlo Molchanov, and Jan Kautz. Weakly-supervised 3d human pose learning via multi-view images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5243–5252, 2020.

[22] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7718–7727, 2019.

[23] Sheng Jin, Wentao Liu, Enze Xie, Wenhai Wang, Chen Qian, Wanli Ouyang, and Ping Luo. Differentiable hierarchical graph grouping for multi-person pose estimation. In *European Conference on Computer Vision*, pages 718–734. Springer, 2020.

[24] Rawal Khirodkar, Visesh Chari, Amit Agrawal, and Ambrish Tyagi. Multi-instance pose networks: Rethinking top-down pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3122–3131, 2021.

[25] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1077–1086, 2019.

[26] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11977–11986, 2019.

[27] Oh-Hun Kwon, Julian Tanke, and Juergen Gall. Recursive bayesian filtering for multiple human pose tracking from multiple cameras. In *Asian Conference on Computer Vision*, 2020.

[28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[29] Yebin Liu, Juergen Gall, Carsten Stoll, Qionghai Dai, Hans-Peter Seidel, and Christian Theobalt. Markerless motion capture of multiple characters using multiview image segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[30] Yebin Liu, Carsten Stoll, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2011.

[31] Haoyu Ma, Liangjian Chen, Deying Kong, Zhe Wang, Xingwei Liu, Hao Tang, Xiangyi Yan, Yusheng Xie, Shih-Yao Lin, and Xiaohui Xie. Transfusion: Cross-view fusion with transformer for 3d human pose estimation. *arXiv preprint arXiv:2110.09554*, 2021.

[32] 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.

[33] Tariq Masood and Johannes Egger. Augmented reality in support of industry 4.0—implementation challenges and success factors. *Robotics and Computer-Integrated Manufacturing*, 2019.

[34] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018.

[35] 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.

[36] Katta G Murty and Santosh N Kabadi. Some np-complete problems in quadratic and nonlinear programming. Technical report, 1985.

[37] Xuecheng Nie, Jiashi Feng, Junliang Xing, and Shuicheng Yan. Generative partition networks for multi-person pose estimation. *arXiv preprint arXiv:1705.07422*, 2017.

[38] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4903–4911, 2017.

[39] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019.

[40] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937, 2016.

[41] Philipp A Rauschnabel, Reto Felix, and Chris Hinsch. Augmented reality marketing: How mobile ar-apps can improve brands through inspiration. *Journal of Retailing and Consumer Services*, 2019.

[42] N Dinesh Reddy, Laurent Guigues, Leonid Pishchulin, Jayan Eledath, and Srinivasa G Narasimhan. Tessetrack: End-to-end learnable multi-person articulated 3d pose tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15190–15200, 2021.

[43] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[45] Julian Tanke and Juergen Gall. Iterative greedy matching for 3d human pose tracking from multiple views. In *German Conference on Pattern Recognition*, 2019.

[46] 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.

[47] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *European Conference on Computer Vision*, 2020.

[48] DA Turner, IJ Anderson, JC Mason, and MG Cox. An algorithm for fitting an ellipsoid to data. *National Physical Laboratory, UK*, 1999.

[49] Ben Usman, Andrea Tagliasacchi, Kate Saenko, and Avneesh Sud. Metapose: Fast 3d pose from multiple views without 3d supervision. *arXiv preprint arXiv:2108.04869*, 2021.

[50] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7782–7791, 2019.

[51] Jue Wang, Shaoli Huang, Xinchao Wang, and Dacheng Tao. Not all parts are created equal: 3d pose estimation by modeling bi-directional dependencies of body parts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7771–7780, 2019.

[52] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He, and Ling Shao. Deep 3d human pose estimation: A review. *Computer Vision and Image Understanding*, 210:103225, 2021.

[53] 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.

[54] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11802–11812, 2021.

[55] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5255–5264, 2018.

[56] Hashim Yasin, Umar Iqbal, Bjorn Kruger, Andreas Weber, and Juergen Gall. A dual-source approach for 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4948–4956, 2016.

[57] Siwei Zhang, Yan Zhang, Federica Bogo, Marc Pollefeys, and Siyu Tang. Learning motion priors for 4d human body

capture in 3d scenes. In *International Conference on Computer Vision*, 2021.

[58] Yuxiang Zhang, Liang An, Tao Yu, Xiu Li, Kun Li, and Yebin Liu. 4d association graph for realtime multi-person motion capture using multiple video cameras. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[59] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.