Journal of Real-Time Image Processing manuscript No.
(will be inserted by the editor)

Fast automatic camera network calibration through human mesh recovery

Nicola Garau · Francesco G.B. De Natale · Nicola Conci

Received: date / Accepted: date

Abstract Camera calibration is a necessary preliminary step in computer vision for the estimation of the position of objects in the 3D world. Despite the intrinsic camera parameters can be easily computed offline, extrinsic parameters need to be computed each time a camera changes its position, thus not allowing for fast and dynamic network re-configuration. In this paper we present an unsupervised and automatic framework for the estimation of the extrinsic parameters of a camera network, which leverages on optimised 3D human mesh recovery from a single image, and which does not require the use of additional markers. We show how it is possible to retrieve the real-world position of the cameras in the network together with the floor plane, exploiting regular RGB images and with a weak prior knowledge of the internal parameters. Our framework can also work with a single camera and in real-time, allowing the user to add, re-position, or remove cameras from the network in a dynamic fashion.

Keywords camera calibration · pose estimation · human mesh recovery · 3D matching

1 Introduction

In computer vision and 3D reconstruction, many works over the years have tried to automate the process of camera resectioning and calibration. Having the possibility to minimise the manual intervention within the calibration pipeline could simplify its deployment in many contexts and in a significant way. However, there is still a lack for fully unsupervised and markerless approaches for camera calibration in literature. The manifoldness of camera sensors and lenses present in the market hinders any generalization attempt. Another aspect that plays an important role in increasing the difficulty of automatic calibration is the dynamic nature of the environments, in which camera networks are generally being installed. For example, in many scenarios, including video surveillance, Ambient Assisted Living (AAL) and environmental monitoring, the reconfiguration and consequent re-calibration of the camera network is a common process, often due to the repositioning or addition of pieces of furniture, or, more in general, the presence of obstacles that can partially or fully limit the visibility of the observed environment. In addition, cameras with pan-tilt-zoom (PTZ) capabilities are often used. A big issue linked to the usage of PTZ cameras is that they are capable of changing their internal configuration, making it necessary to recalibrate the whole network whenever these changes occur. In addition, wind or other weather conditions may also further complicate the scenario, introducing noise, and making it difficult to accomplish even the simplest vision tasks, such as keypoints extraction, motion detection and tracking.

Generally speaking, the internal configuration of a camera usually remains fixed, unless when zooming, refocusing or changing the lens parameters. Many good solutions to estimate the intrinsic parameters of a camera have been provided in the literature, and they usually require the usage of a checkerboard or other calibration tools.

Nicola Garau (✉)
University of Trento, Via Sommarive, 9, 38123 Povo,Trento TN (Italy) E-mail: nicola.garau@unitn.it

Francesco G.B. De Natale
University of Trento, Via Sommarive, 9, 38123 Povo,Trento TN (Italy) E-mail: francesco.denatale@unitn.it

Nicola Conci
University of Trento, Via Sommarive, 9, 38123 Povo,Trento TN (Italy) E-mail: nicola.conci@unitn.it

On the other hand, extrinsic parameters model the relation between the camera coordinates and the real-world coordinates. Ideally, extrinsic parameters remain unaltered if both the camera and world long-term steadiness can be guaranteed. For this reason, even a slight movement of the camera can cause a loss in calibration precision, leading to the need of re-calibrating the whole system. This is problematic because the standard calibration procedures, though not complex, are rather time consuming and require the usage of third party calibration instruments by an expert technician who needs to be on the spot to perform the task. Another issue is that whenever a calibration is in progress, the camera network remains busy and inoperable.

A few approaches in literature [8] try to simplify the calibration process by increasing the accuracy of the calibration pattern detectors, thus reducing the number of required checkerboard images. However, despite being fast, they still require manual intervention. For example, although in a different application scenarios, the adoption of markerless solutions has been explored in the autonomous driving context, exploiting visual odometry [21], SLAM [6], and optical flow [19] for feature tracking; however, they are often not suitable for surveillance scenes, since such methods require a fixed camera configuration with respect to the vehicle, in order to exploit the car movement information for calibration. Other methods use SIFT/SURF feature matching between camera views to estimate the extrinsic camera parameters [11], but they usually require additional data from other sensors, such as active range sensors.

A recent trend in computer vision is pedestrian-based camera calibration, which focuses on finding how to estimate both intrinsic and extrinsic camera parameters by exploiting the cues provided by walking humans. In particular, these approaches are usually based on:

- Manhattan World Assumption
- Planar trajectories
- Skeleton data from 3D sensors

The approaches based on the Manhattan World Assumption [4] are usually adopted in city-like environments due to their geometric homogeneity, but may fail in other scenarios, when no such geometric cues are being found. Human detection and tracking have been explored in literature as a support for vanishing point estimation and to estimate the ground plane from multiple camera views [30]. However, these methods often require a prior knowledge of the cameras' vertical position or of the people height, they are not robust to occlusions, noise, and can be fooled by unconventional human poses. Recently, RGBD sensors such as the Microsoft Kinect V2[1] and the Intel RealSense[2] allowed obtaining a better understanding of the scene through depth and 3D skeleton pose estimation [26][27][23]. However, there are many issues linked to the usage of RGBD sensors such as Kinect and RealSense to calibrate a camera network from the skeleton information. Among them, the most relevant ones are:

- Small range (usually $4m$) of the depth sensor; this constraint is not suitable for large environments.
- Low precision; occlusions, ambiguities and reflections in the scene are an important factor for the skeleton extraction precision.
- High infrastructural and processing cost; multiple computers and GPUs are usually required to process data coming from a network of RGBD sensors in real-time.

A recent trend in computer vision concerns the area of human pose estimation from monocular images. There have been many successful examples, such as [24], [33] and [2]. Many of the good results have been made possible thanks to the availability of very large datasets, in particular CMU's Panoptic Studio [12], which contributed to speed up the development of many popular and open source frameworks, such as OpenPose [3].

Amongst the different kinds of 3D monocular human pose estimation techniques used in literature, we can distinguish between:

- two-stage approaches
- end-to-end approaches

Two-stage approaches, such as [31], first estimate 2D joints and then recover the depth component. On the other hand, end-to-end approaches try to recover the 3D skeleton or mesh in one shot. Kanazawa et al.'s work [13] is one of the most recent ones, which takes as input an image, encoding it into body pose, shape and weak camera pose via a CNN encoder; then, a discriminator is used as supervisor to encourage a better loss, by comparing the produced 3D model with a pool of real scanned 3D human poses (Fig. 2). Despite it being a very good approach to estimate the 3D mesh of a person, it may still fail, especially when dealing with unusual viewpoints and in time-constrained scenarios. Kolotouros et al. in SPIN (SMPL oPtimization IN the loop) [16] provide a fix to these issues by employing an hybrid top-down and bottom-up approach that aims at optimising the human mesh recovery (HMR) phase.

[1] https://developer.microsoft.com/en-us/windows/kinect

[2] https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html

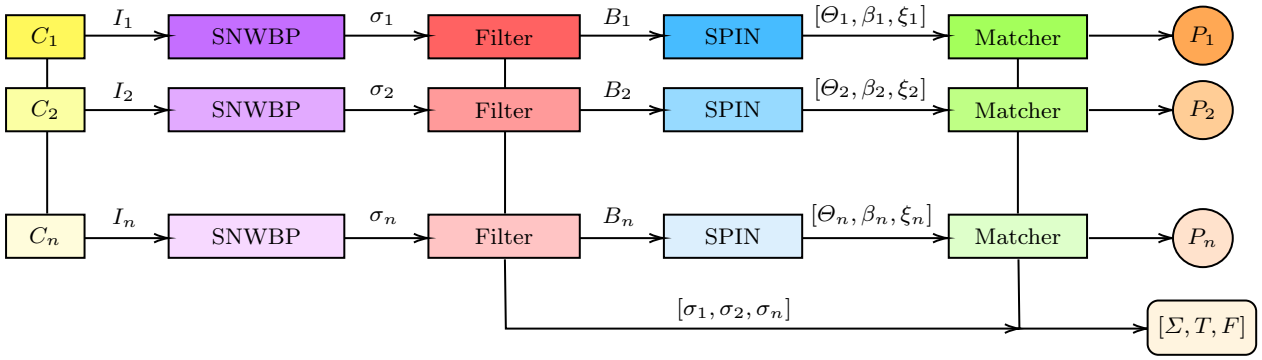

Fig. 1 An overview of our camera network calibration pipeline. In the figure, SNWBP refers to a Single-Network Whole-Body Pose Estimation, the filters prepare the inputs for the SPIN human mesh recovery module. The matcher simply matches the obtained 3D skeletons in order to compute the output poses.

Their method is based on the iterative application of optimisation and regression-based approaches (such as [13]) to further improve human mesh recovery, by mixing the advantages of both the approaches, in particular the accuracy of the first one with the speed of the second one.

Starting from Kanazawa's work, we extend it in a similar fashion as the one described in [16], and repurpose in order to work with multiple views and with a more realistic camera model, that allows us to better estimate the extrinsic parameters for each camera. Our results show that, starting from a single frame, the retrieved human skeleton alone can provide a sufficient number of keypoints to estimate the real-world 3D position of cameras in a network, thus achieving fully unsupervised camera calibration. We show results in different scenarios and with different cameras configurations and discuss on how our method can be further extended for better accuracy. This work is an extension of our previous work [7]. The main contribution, compared to the work in [7] consists of the capability of the system to obtain real-time camera network calibration, at comparable accuracy. This is achieved thanks to the adoption of a faster SNWBP network [9] and a more precise human mesh recovery pipeline [16]. These improvements allow for an even easier deployment in real-world scenarios, and are particularly helpful when dealing with large camera networks and real-time constraints.

2 Related work

2.1 Human Pose Estimation (HPE)

Before the advent of deep learning, classical HPE approaches were based on the so-called pictorial structures framework [1]. Later on, this kind of hand-crafted features, as well as customised hardware solutions (e.g., RGBD-based sensors) became less popular, making room for HPE algorithms based on deep learning paradigms. Many human pose estimation techniques [33, 2] are based on bottom-up 2D skeleton estimation to guarantee good performances. Recent contributions [31] explore two-stage approaches, in which the 2D pose is first estimated and then used as a baseline to infer the corresponding 3D pose.

2.1.1 Bottom-up approaches

Estimating the human pose in a bottom-up fashion means first estimating all the joints in a frame and then linking them together in a meaningful, structured hierarchy. Cao et al.'s Realtime multi-person 2D pose estimation using part affinity fields [2] is one of the most popular multi-person real-time 2D pose estimation works in literature. It combines the architecture of a CNN-based variation of Pose Machines, called Convolutional Pose Machines [33], leveraging on part affinity fields. Part affinity fields can be defined as a group of oriented vectors linking different joints. In other words, part affinity fields can be seen as confidence maps identifying bones, while joint confidence maps identify joints and articulations.
The solution, presented in [2], is very robust to large scale occlusions and self occlusions. Its dual-branch architecture for CNN-based joint parts and pairs estimation is optimised to run in real-time on consumer hardware, making it suitable for many research applications, and known as *OpenPose* [3]. However, it is still not faster than many top-down approaches when dealing with low density scenarios. Recently, it has been extended with a single track architecture [9], rendering it much faster than before, also embedding the hand and face joint information.

2.1.2 End-to-end solutions for 3D human pose estimation

End-to-end recovery of human shape and pose[13] is one of the most popular works in joint 3D human shape and pose. From a single RGB image of a person, the human pose θ and body shape β are regressed, together with camera scale s, rotation R and translation T.
An issue with this approach is that it is not suitable for run-time application and it is highly affected by viewpoint changes and flickering between frames, due to the lack of temporal coherence. Other works (see [16][14][22]) inspired by [13] try to solve the flickering issue by using temporal cues or predicting future poses.

2.2 Automatic calibration

Most of the automatic extrinsic calibration works in literature leverage on the so called Manhattan World Assumption [4], which assumes that the geometry typical of urban areas makes it easier to discover vanishing points from images taken in those kind of environments. As an example, Zhang et al. in [34] propose a solution that exploits the geometry of solar panels to estimate orthogonal vanishing points. While such assumption is valid and works well in city scenarios, it does not generalize sufficiently, especially when indoor scenes are taken into consideration. Many methods in literature deal with the problem of finding the parameters of a single camera which is being plugged in to an existing and already calibrated camera network. Vasconcelos et al. in [32] exploit sets of pairwise correspondences among images in order to estimate the pose of the new camera. Despite the high deployability, this method only works when the extrinsic parameters of the other cameras are known. In literature, methods for self-calibration of pan-tilt [15] and tilt-zoom [25] cameras can also be found. Traditionally, such kind of self calibration problems are handled using geometrical constraints; however, with the growing popularity of deep learning, some approaches tried to solve the problem, as in Hold-Geoffroy et al.'s work [10], which aims at estimating pitch, roll and focal length of a single camera employing convolutional neural networks. Of particular interest from this viewpoint, some very recent works focus on embedding CNN capabilities directly into camera sensors. One of the first works to achieve such results is Bose et al.'s *A Camera That CNNs: Towards Embedded Neural Networks on Pixel Processor Arrays*, an interesting proposal that could open new possibilities for in-camera self-calibration. With the growing popularity of omnidirectional cameras, Miyata et al. in [20] show how to exploit their large field of view to anchor non-overlapping views. However, such solution are not employable in some scenarios, such as AAL, since occlusions may play an important role for the failure of the keypoints detectors. Augmented reality (AR) also played an important role for refreshing the field of camera calibration. Zhao et al. in [36] employ augmented reality markers placed directly on top of cameras in order to perform camera calibration. However, the method requires the usage of an additional dedicated camera just for the recognition of the AR markers.
Perhaps the most popular approaches for automatic camera calibration are the ones employing vanishing points estimation. Tang et al. in ESTHER [30] propose a complete pedestrian trajectory-based solution for joint intrinsic and extrinsic parameters estimation, especially focusing on intrinsic calibration for distortion correction. However, their method requires pedestrian to walk in a standard upright position, as well as a prior knowledge of the cameras vertical height.
To our best knowledge, few other works exploit human pose cues for camera calibration, and most of them exploit 3D sensors data, such as depth maps or cameras disparity information.

Desai et al. in [5] propose a skeleton-based method for semi-automatic continuous calibration of Kinect V2 sensors. In their work, they also explore some issues related to working with depth sensors, such as low range of vision, skeleton flipping and high computational costs. Among the many recent 3D human pose estimation works, some also try to jointly retrieve human pose and weak camera parameters. Kanazawa et al.'s approach [13] provides an estimation of the subject in terms of mesh, shape and pose representation, as well as some shallow cues of the camera pose.

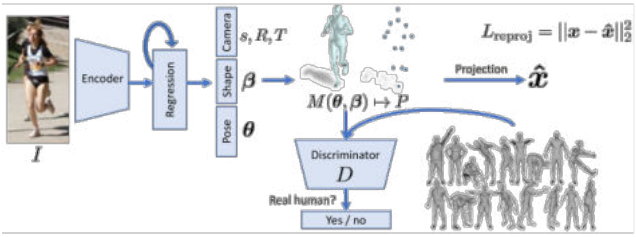

Fig. 2 Joint pose, shape and camera estimation of *End-to-end recovery of human shape and pose* [13] pipeline.

3 Method overview

In this section, we refer to Fig. 3 to provide a red thread to explain our method's pipeline. During phase A, for each camera C_i in the network we acquire a single frame in a synchronous fashion. Then, each frame

is forwarded to a Single-Network Whole-Body Pose Estimation (SNWBP) [9] network (phase B), which is a very fast convolutional network that is able to infer the 2D skeleton (σ) of multiple people inside the image in real-time. In this phase we also use the 2D skeleton information to compute a 2D bounding box B_i for each detected person in each frame. Then, during phase C, we use our joint human mesh recovery and camera pose estimation network, which is based on [16]. Starting from the monocular human mesh recovery network described in [16], we extend it by modifying the underlying camera model, providing a full perspective camera model. This addition, makes it possible, during phase D, to exploit the information acquired in the previous steps, such as the bounding boxes, 2D and 3D skeletons, body shape and pose parameters, to retrieve a good estimation of the camera pose for each camera in the network. All the four phases can run in parallel for each camera in the system, and in a continuous loop, in order to maximise both performance and precision by refining the calibration results over time.

4 The proposed model

In this section, we propose our one-shot method for fully automatic and unsupervised camera network calibration that leverages on monocular 3D human pose estimation from single images. In Figs. 3 and 1 we describe the pipeline and the different steps of our architecture. Looking at the bigger picture, our framework receives as input a single RGB frame $I_{0,\dots,n}$ from $n >= 1$ camera video streams $C_{0,\dots,n}$. We then apply fast, single network 2D pose estimation [9] for each frame, in order to filter matching subjects across frames and obtain the corresponding bounding boxes $B_{0,\dots,n}$ (Fig. 4). We then apply our custom optimised human mesh recovery method based on [13][16] to infer the 3D position of skeletal joints together with their real-world scale. Finally, when dealing with $n > 1$ cameras, we align the skeletons centroids and use a least squares approach to find a set of rigid transformations $T_{\{i\to 0|i=1,\dots,n\}}$ from each skeleton to another one in 3D world space. After minimizing the displacement error between skeletons in 3D space, we can exploit the epipolar geometry as well as the world-space and image-space position of joints to retrieve both the extrinsic parameters for rotation and translation $R \mid \mathbf{t}$ and the fundamental matrices $F_{\{i\to 0|i=1,\dots,n\}}$. In case of a single camera, the matching step and the fundamental matrix calculations are being ignored and we simply retrieve the camera matrix as well as the 3D human pose and shape.
Whenever the framework detects a difference in the detected 3D-space joints, which is bigger than a threshold, it triggers a new re-calibration cycle, in order to keep the network calibrated over time, progressively refining its accuracy. Another big advantage of our method is its flexibility. In fact, it can work even with a single camera and it allows for new cameras to be plugged into the system in a dynamic fashion.

In the next sections we refer to the camera matrix as P. The intrinsic matrix is defined by K, where $f_x = fm_x$ and $f_y = fm_y$ represent the focal length values in pixels, scaled along x and y by a scaling value m. The principal point of the camera is represented by (x_0, y_0). Extrinsic parameters are modelled by $[R \mid \mathbf{t}]$, where R is the rotation matrix and t identifies the translation vector.

$$P = \overbrace{\begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}^{\text{K}} \times \overbrace{\left(\begin{array}{c|c} R & \mathbf{t} \\ \hline 0 & 1 \end{array}\right)}^{[\text{R} \mid \mathbf{t}]} \tag{1}$$

4.1 2D pose matching

As a first step, we provide a module that handles fast multi-person 2D pose estimation and filters detected skeletons to ensure good pose-based subject matches across the views. This first part of the architecture takes as input n RGB frames and outputs a bounding box for the target person in each image in terms of 2D pose, together with the overall highest detection confidence score among all the views. To ensure real-time performances, we employ an improved version of the method described by Cao in [2] for joint parts and pairs detection, namely Single-Network Whole-Body Pose Estimation (SNWBP) [9]. Alternatively, under particular conditions such as fixed, single-person, noise-free and occlusions-free scenarios, it is possible to employ classic background subtraction methods or more advanced versions such as [29] to extract the bounding boxes.

At this point the skeleton joints information is already sufficient to calculate the fundamental matrices that link the views. However, we decided to further reinforce this estimation by providing additional points obtained from the re-projection of the 3D skeleton onto the image planes, as we will explain later on. By doing so, we observe an increment in the accuracy of the final fundamental matrices. Therefore, at this phase we only keep a reference to the displacement of the central point $[D_x^{pix}, D_y^{pix}]$ and the pixel-size of each bounding box, as well as an *unscale* factor, which serves as a parameter that can be used to reverse the scaling of the bounding boxes.
The 3D mesh recovery module based on [13] is able to retrieve an estimation of the person height, which can

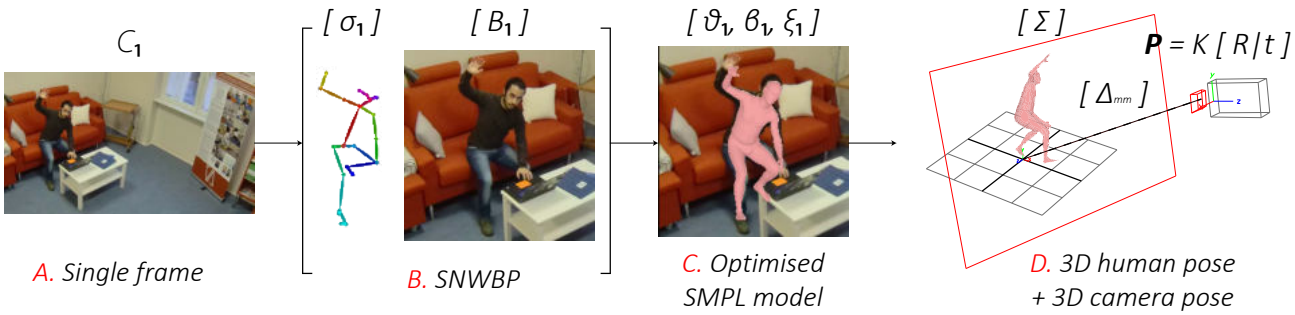

Fig. 3 Illustration of the proposed pipeline: from a single RGB image to the estimation of the extrinsic parameters of the camera. In case of a camera network, the same pipeline is applied for each camera, before a 3D matching phase, as described in Fig. 1

Fig. 4 2D pose estimation for bounding box extraction

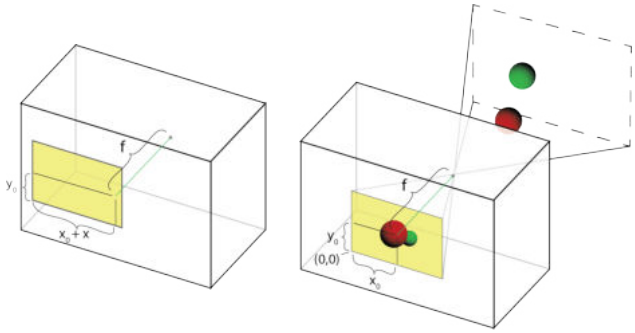

Fig. 5 Principal point offset for mesh positioning in the weak perspective camera model does not correspond to a real-world mesh translation [28].

be used as a substitute to the real height. However, we provide as an option the possibility to give as additional input the real height of the considered subject in order to maximise the accuracy of the calibration.

4.2 Mesh recovery

Once we recovered the matching bounding boxes across all the different views together with the optional joint information, we need to recover 3D joints information that we will use to calculate the extrinsic parameters. At this point, each scaled bounding box $B_{0,\dots,n}$ is configured as a crop of the frames containing the subject chosen by 2D pose-similarity, as seen from different viewpoints. We now need to retrieve the 3D skeleton joints from each viewpoint, in a monocular fashion without relying on information from the other views. To achieve this, we employ our modified version of the method described in [13] and [16] (SPIN). By feeding each bounding box B_i to the network, we obtain the vector Θ, corresponding to the SMPL (Skinned Multi-Person Linear Model) [17] human body model parameters, which is configured as follows:

$$\Theta = [\overbrace{s, t_x, t_y}^{\text{camera}}, \overbrace{\theta}^{\text{pose}}, \overbrace{\beta}^{\text{shape}}] \tag{2}$$

From each human mesh $M(\theta, \beta)_i$ it is possible to obtain a set of $J = 19$ world-scale 3D joints ξ_i (in meter coordinates). The 10 body shape parameters β encode different deformations of the mesh shape, and are used to refine the weak 2D pose matching described in 4.1 as well as removing remaining outliers. We discard the original camera parameters s, t_x, t_y since they model a weak perspective pinhole camera model with its principal point shifted by $[t_x, t_y]$ (Fig. 5). In the original model described in [13], the world translation of the mesh is computed as $z = F/s$, where s is a scaling factor.

The weak perspective model is not accurate enough for retrieving real-world mesh displacements because it does not take into account perspective transformations. In fact, in weak perspective geometry, perspective transformations are modeled via a simple scaling in the subject size, proportionally to its distance from the camera. In practice, if we take into account the manifold of commercially available sensors and lenses, employing a weak camera model is a strong generalisation, which can lead to substantial errors. For this reason, we substitute the original weak camera model with a fully-fledged perspective one, to recover the real-world mesh displacement Δ^{mm} in millimeters, as shown in Eq. 3:

$$\Delta^{mm} = \left[\overbrace{\frac{\Delta_z^{mm}\Delta_x^{pix}}{f_x^{pix}}}^{\Delta_x^{mm}} , \overbrace{\frac{\Delta_z^{mm}\Delta_y^{pix}}{f_y^{pix}}}^{\Delta_y^{mm}} , \overbrace{\frac{f^{mm} w B^{mm}}{W B^{pix}}}^{\Delta_z^{mm}} \right] \tag{3}$$

where $f^{pix} = [f_x^{pix}, f_y^{pix}]$ corresponds to the focal length values in pixels, w is the image width in pixels and W is the sensor width in millimeters. B^{pix} and B^{mm} are the image-coordinates and world-coordinates sizes of the bounding boxes retrieved from 4.1. At this point, Δ_i^{mm} contains the real-world relative translation going from the camera C_i to the 3D skeleton ξ_i.

4.3 Skeleton matching

At this stage, in presence on an arbitrary number $n > 1$ cameras, we have obtained n camera-centric systems each one referring to a 3D skeleton. The next step is setting each skeleton's centroid c_i as the pivot point for each corresponding camera C_i. Thus, we need to find the rotation matrices R_i that map each skeleton $\xi_{1,\dots,n}$ to ξ_0. We achieve this by moving towards a skeleton-centric system, in which each skeleton centroid c is positioned in the center of coordinates $(0, 0, 0)$. In this space, we can find the relative skeleton-to-skeleton transformations in terms of rotations and translations using a single value decomposition (SVD) approach, as explained in equations 4 and 5.

$$H = \xi_0 \cdot \xi_i \quad , \quad U, S, V^\top = SVD(H) \tag{4}$$

$$R = V \cdot U^\top \quad , \quad \mathbf{t} = c_i - (R \cdot c_0^\top) \tag{5}$$

More in detail, we calculate H as the dot product of a pair of 3D point sets of joints ξ_0 and ξ_i. We then apply an SVD to H to find the matrices U, S, V, as explained in Eq. 4. Finally, we find the rotation matrix R and the translation vector t as detailed in Eq. 5. A simple representation of the 3D skeleton match can be seen in Fig. 6.

Then, we move back to the camera-centric space and find the transformation that maps ξ_0 to Δ_0^{mm}. We finally find the inverse transformations Δ_i^{mm}, starting from Eq. 3.
By applying this procedure, we obtain a 3D space, in which the first camera C_0 is positioned at the center of the coordinate system, the n skeletons are in Δ_0^{mm} and the relative position of all the other virtual cameras is known. An example of the final output of the whole pipeline can be seen in Fig. 7.

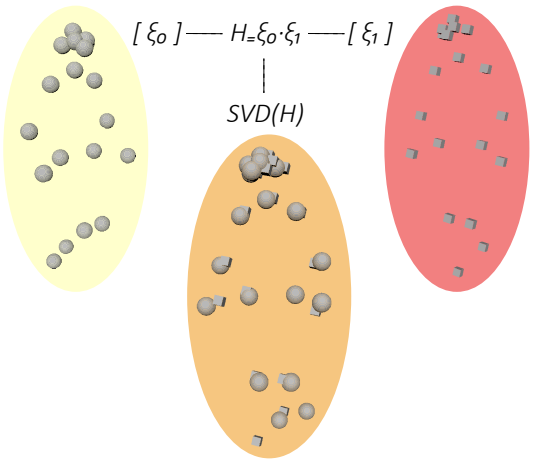

Fig. 6 Our SVD approach for 3D skeleton matching

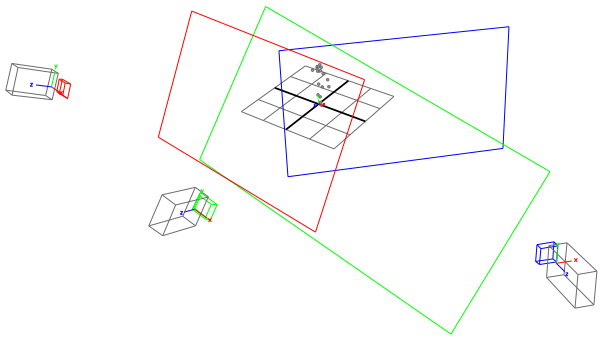

Fig. 7 Visualisation of the final result of our automatic calibration pipeline.

4.4 Fundamental matrix

With the skeletons $\xi_{0,\dots,i}$ correctly positioned in the 3D world, we calculate Σ as the merged 3D skeleton containing the mean values of all the joints coming from $\xi_{0,\dots,i}$ in world-space coordinates. Since we also know the position of each camera in the 3D world, we can project Σ to each image plane of C_i, obtaining σ_i. We then build a vector σ_i containing 2D skeleton joints values for a batch of frames coming from C_i and use it as ordered keypoints to find the fundamental matrices $F_{i\to 0}$ that match camera C_i with camera C_0.
This allows us to find the epipolar lines and corresponding matching points between pairs of camera views. Moreover, since the extrinsic matrices have been previously retrieved, it is possible to describe, how points in world coordinates map to each camera coordinate system, and viceversa.

5 Results

To test our framework, we conducted seven real-world experiments in different scenarios, as listed in Table 2.

The first four experiments were carried out in a real living lab consisting of three rooms, which is equipped with a network of identically-configured HD and FullHD cameras monitoring all the rooms. The last three experiments serve as a comparison of the proposed pipeline with our previous method, which employed video sequences instead of single frames, as well as slower and less precise human pose estimators. Our results are comparable with the ones provided by [5], both in terms of spatial configuration and precision, even if we rely on just monocular information from simple RGB cameras and not on depth or triangulation. Experiments 2,3,7 show how our method is robust against important occlusions in the scene. In experiment 5 we demonstrate how our method can also work with very distant and little overlapping cameras. In experiment 6 we employ two handheld smartphones (not stabilized) and successfully retrieve a good estimation of their pose in the 3D world.

5.1 Quantitative results

The main results of our experiments are listed in Table 1 and Table 2. As can be seen, they are comparable with the results provided by [5], particularly taking into consideration that we only employ monocular cameras and no additional depth sensors. The metrics MinSDE, ASDE and MaxSDE, describe the minimum, average, and maximum displacement of skeletal joints, respectively, after the matching in 3D space, in meters, calculated by the Euclidean distance:

$$SDE = \sqrt{\sum_{i=1}^{n} \left(\xi_0 - \xi_i\right)} \tag{6}$$

RPD, VPD (real and virtual plane displacements) are the measures of the displacement from the origin along the real world plane and the virtual world plane respectively. The RPD has been calculated starting from ground truth annotations, while the VPD can be calculated once again with an Euclidean distance from the origin, discarding the z component. The plane displacement error (PDE) is computed as $\mid RPD - VPD \mid$, once again in meters. The MRE is the mean reprojection error calculated after applying the fundamental matrix F to the set of points σ_i.
Our results for most of the scenarios are also better than the checkerboard results obtained by [5] using the method described in [35].

5.2 Reprojection error

After finding the fundamental matrices F for each scene and the corresponding epipolar lines, we assess the precision of our method by calculating the reprojection error in term of point-line-distance, as follows:

$$\frac{\mid ax_0 + bx_0 + c \mid}{\sqrt{a^2 + b^2}} \tag{7}$$

where a, b and c are the epipolar lines coefficients and $[x_0, y_0]$ are the coordinates of the projected points. In Table 1 the reprojection errors in pixels for each scenario are listed, showing that the proposed method is robust in all the four test environments considered.

	Scenarios			
	Kitchen	Gym	Laboratory	Apartment
MRE	12.77	**6.15**	12.03	8.38

Table 1 Reprojection errors in pixels for the four test scenarios

5.3 Qualitative results

In Figs. 8 and 9 we provide some qualitative results through the Autodesk Maya® 3D animation, modeling, simulation, and rendering software. In each image a reconstruction of the 3D scene is shown, including the 3D skeleton used for the matching, the virtual plane and every virtual camera with correct roll, pitch, yaw, translation, focal length and frustum size. We decided to discard the approximate differentiable render OpenDR [18] used by [17] and [13] in favour of Maya because the latter lets us configure in fine details many camera parameters including the focal length, the film gate and frustum size in millimeters. Moreover, our entire code can directly run into the Maya environment, allowing us to easily extend the scope of our work to weak monocular 3D human motion capture from video footage, also from a single camera. Our 3D reconstruction module in Maya is standalone and can receive camera and skeleton data from an external machine via a command port socket in real-time. As an alternative, we provide bindings for Open3D.
Concluding, despite the lack of proper datasets to benchmark these kind of applications, we also provide, in addition to the original experiments, some good qualitative results from the Panoptic Dataset [12] and from fully simulated scenarios. In Fig. 10 we show an example of camera pose estimation from 8 different views caught from 8 virtual cameras inside the Unity 3D environment. Similar results can be obtained for all the 480 VGA cameras in the Panoptic Dataset.

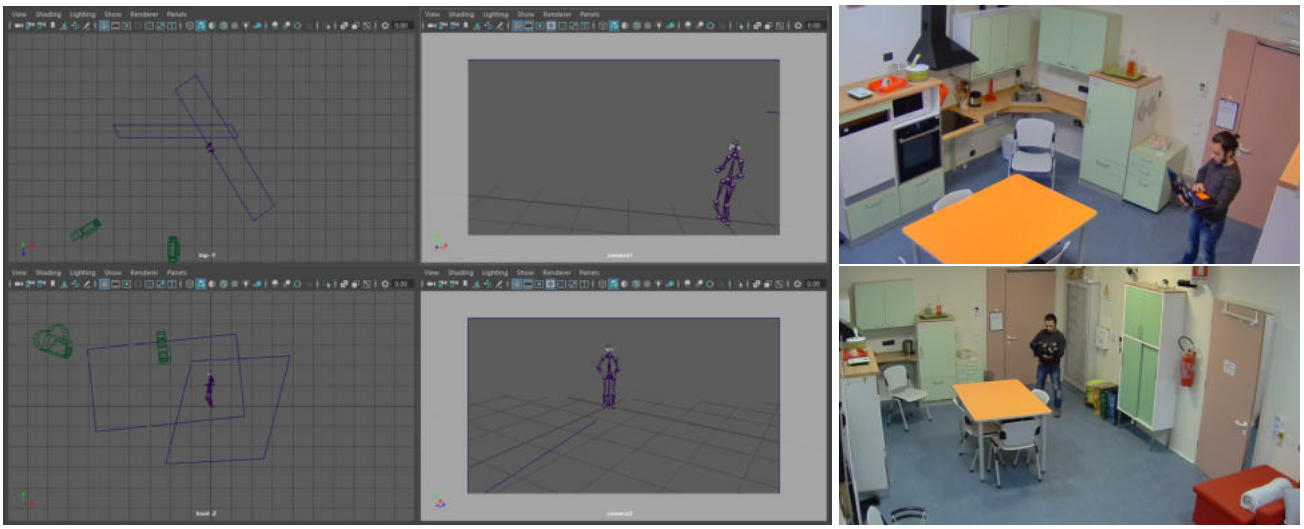

Fig. 8 Kitchen scenario

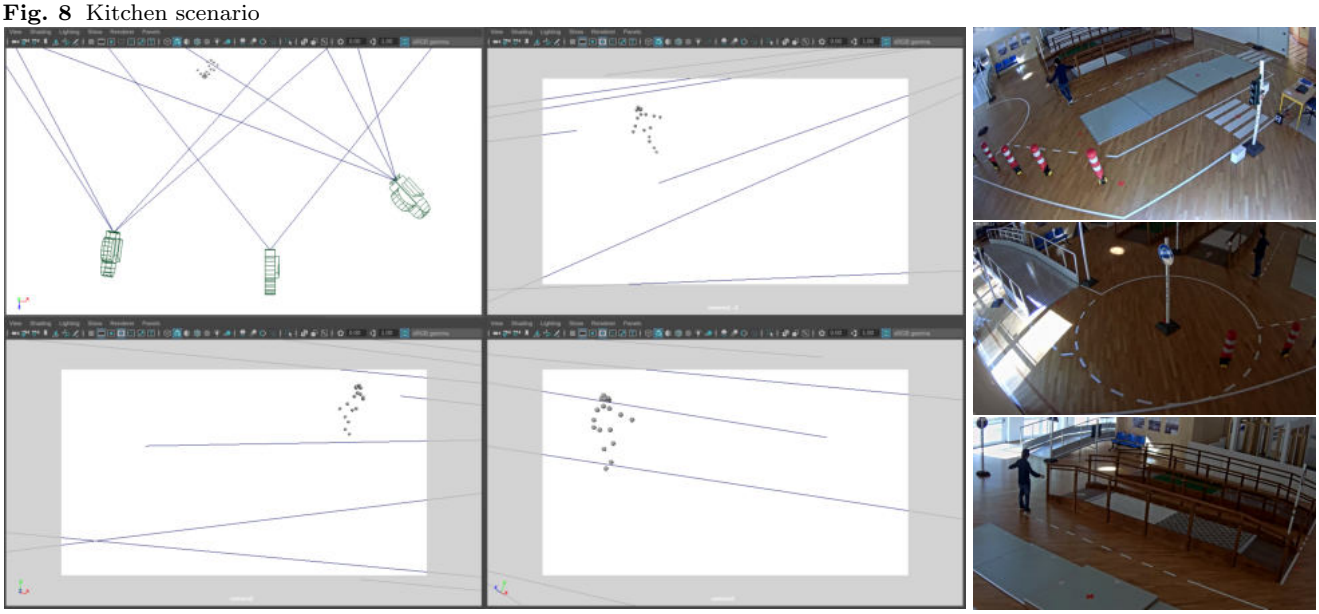

Fig. 9 Gym scenario

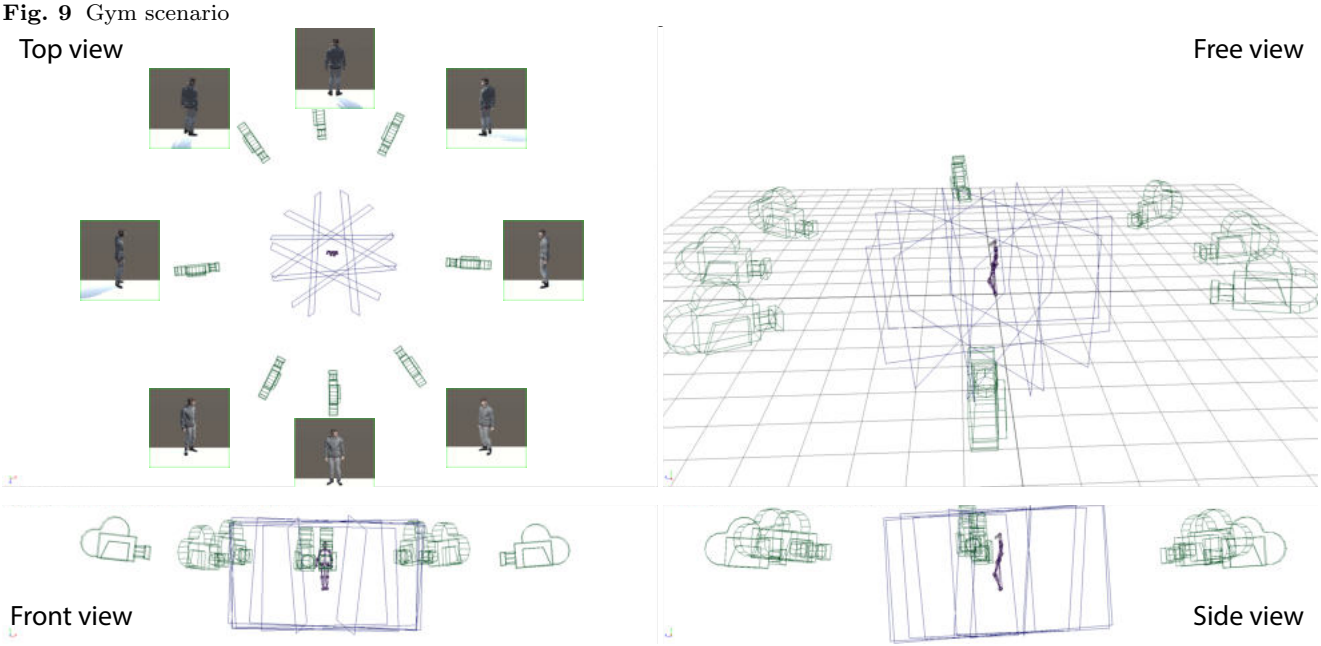

Fig. 10 Simulated scenario: estimating the pose of 8 virtual cameras inside Unity

	Experiments													
	1 Kitchen A		2 Kitchen B		3 Kitchen C		4 Living room	5 Wheelchair gym			6 Laboratory		7 Apartment	
Configuration														
Num. of cameras	2		2		2		1	3			2		2	
Num. of frames	1		1		1		1	250			250		875	
Sensor size	1/3" 6.28mm							1/3" 9.1mm			1/3.2" 5.7mm	1/2.8" 6.3mm	1/2.8" 7mm	
Focal length	4.0	5.0	4.0	5.0	4.0	5.0	5.0	3.0	3.0	5.5	4.0	4.07	4.5	6
3D matching														
SDE min	3.34e-08		3.28e-08		**2.24e-08**		n.d.	0.04			0.01		0.10	
SDE avg	**0.04**		0.07		0.04		n.d.	0.08			0.06		0.13	
SDE max	**0.06**		0.10		0.08		n.d.	0.14			0.08		0.16	
Real-world displacement														
RPD	6.09		6.97		5.86		5.45	7.73			5.66		3.26	
VPD	6.46		7.04		6.24		5.50	7.92			5.79		3.35	
PDE	0.37		0.07		0.38		**0.05**	0.21			0.13		0.09	

Table 2 Experimental results. Columns: the seven different test scenarios (the last three are results from our previous method [7]). Rows: number of cameras, number of frames, sensor sizes, focal length, 3D skeleton displacement error (min, average, max), displacement along the plane (real plane, virtual plane, plane displacement error)

6 Conclusions

We presented a completely unsupervised and one-shot camera network calibration framework capable of calibrating a single camera or a camera network only from monocular human pose estimation cues. We employ a 3-stage approach which comprises (i) fast, single network whole body pose estimation and matching among camera views, (ii) perspective corrected, optimised monocular human mesh recovery from a single frame and (iii) joint 2D and 3D skeleton matching in camera-centric and skeleton-centric coordinates. As final output we provide the extrinsic parameters for linking world space with camera space for each camera in the network, as well as their fundamental matrices, to link camera views. Compared to the other related works in literature and with our previous approach, the presented framework enables the possibility for real-time, one-shot network calibration, which is camera-independent and which requires only one frame as input. It is robust to occlusions and noise in the scene thanks to the 3D skeleton matching approach, and it is able to perform real-time re-calibration thanks to its streamlined parallel architecture.

6.1 Future work

As future work, the adoption of a capsule network model for estimating the body pose could solve many issues, particularly with respect to (i) pose flickering, (ii) extreme camera viewpoints and (iii) non-existent viewpoint-equivariance. Additional improvements could be made by reinforcing the matching algorithm with SIFT/SURF features and alike. Adding the possibility to estimate the intrinsic parameters in a robust way could greatly improve the overall deployability and accuracy of our framework.

Acknowledgements This research was developed within the framework of the project AUSILIA (2015-2020), funded by the Autonomous Province of Trento (Italy).

References

1. M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1014–1021, 2009.
2. Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310, 2017.
3. Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.
4. James M. Coughlan and Alan L Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 845–851. MIT Press, 2001.
5. Kevin Desai, Balakrishnan Prabhakaran, and Suraj Raghuraman. Skeleton-based continuous extrinsic calibration of multiple rgb-d kinect cameras. In *Proceedings of the 9th ACM Multimedia Systems Conference*, MMSys '18, page 250–257, New York, NY, USA, 2018. Association for Computing Machinery.
6. H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110, 2006.
7. Nicola Garau and Nicola Conci. Unsupervised continuous camera network pose estimation through human mesh recovery. In *Proceedings of the 13th International Conference on Distributed Smart Cameras*, ICDSC 2019, New

York, NY, USA, 2019. Association for Computing Machinery.
8. A. Geiger, F. Moosmann, Ö. Car, and B. Schuster. Automatic camera and range sensor calibration using a single shot. In *2012 IEEE International Conference on Robotics and Automation*, pages 3936–3943, 2012.
9. Gines Hidalgo, Yaadhav Raaj, Haroon Idrees, Donglai Xiang, Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Single-network whole-body pose estimation. *arXiv preprint arXiv:1909.13423*, 2019.
10. Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matthew Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A perceptual measure for deep single image camera calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2354–2363, 2018.
11. Ryo Inomata, Kenji Terabayashi, Kazunori Umeda, and Guy Godin. Registration of 3d geometric model and color images using sift and range intensity images. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Song Wang, Kim Kyungnam, Bedrich Benes, Kenneth Moreland, Christoph Borst, Stephen DiVerdi, Chiang Yi-Jen, and Jiang Ming, editors, *Advances in Visual Computing*, pages 325–336, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
12. H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3334–3342, 2015.
13. A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018.
14. Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5614–5623, 2019.
15. H. Kim and K. S. Hong. Practical self-calibration of pan-tilt cameras. *IEE Proceedings - Vision, Image and Signal Processing*, 148(5):349–355, 2001.
16. Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. *arXiv preprint arXiv:1909.12828*, 2019.
17. Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):248, 2015.
18. Matthew M. Loper and Michael J. Black. Opendr: An approximate differentiable renderer. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 154–169, Cham, 2014. Springer International Publishing.
19. Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. page 674–679, 1981.
20. S. Miyata, H. Saito, K. Takahashi, D. Mikami, M. Isogawa, and A. Kojima. Extrinsic camera calibration without visible corresponding points using omnidirectional cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(9):2210–2219, 2018.
21. David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.
22. Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. In *SIGGRAPH Asia 2018 Technical Papers*, page 178. ACM, 2018.
23. Liliana [Lo Presti] and Marco [La Cascia]. 3d skeleton-based human action classification: A survey. *Pattern Recognition*, 53:130 – 147, 2016.
24. Varun Ramakrishna, Daniel Munoz, Martial Hebert, James Andrew Bagnell, and Yaser Sheikh. Pose machines: Articulated pose estimation via inference machines. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 33–47, Cham, 2014. Springer International Publishing.
25. Y. Seo and K. S. Hong. Theory and practice on the self-calibration of a rotating and zooming camera from two views. *IEE Proceedings - Vision, Image and Signal Processing*, 148(3):166–172, 2001.
26. Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR) 2011, 2011.
27. Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.
28. Kyle Simek. Pinhole camera diagram, dissecting the camera matrix. `http://ksimek.github.io/pinhole_camera_diagram/`, 2013. Accessed: 2019-04-26.
29. Z. Tang, J. Hwang, Y. Lin, and J. Chuang. Multiple-kernel adaptive segmentation and tracking (mast) for robust object tracking. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1115–1119, 2016.
30. Z. Tang, Y. Lin, K. Lee, J. Hwang, and J. Chuang. Esther: Joint camera self-calibration and automatic radial distortion correction from tracking of walking humans. *IEEE Access*, 7:10754–10766, 2019.
31. Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2500–2509, 2017.
32. F. Vasconcelos, J. P. Barreto, and E. Boyer. Automatic camera calibration using multiple sets of pairwise correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):791–803, 2018.
33. S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732, 2016.
34. Gaopeng Zhang, Hong Zhao, Yang Hong, Yueyang Ma, Jing Li, and Huinan Guo. On-orbit space camera self-calibration based on the orthogonal vanishing points obtained from solar panels. *Measurement Science and Technology*, 29(6):065013, 2018.
35. Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.
36. Fangda Zhao, Toru Tamaki, Takio Kurita, Bisser Raytchev, and Kazufumi Kaneda. Marker-based non-overlapping camera calibration methods with additional support camera views. *Image and Vision Computing*, 70:46 – 54, 2018.