Single View Physical Distance Estimation using Human Pose

Xiaohan Fei Henry Wang Lin Lee Cheong Xiangyu Zeng* Meng Wang Joseph Tighe
Amazon Web Services
{xiaohfei, yuanhenw, lcheong, xianzeng, mengw, tighej}@amazon.com

Abstract

We propose a fully automated system that simultaneously estimates the camera intrinsics, the ground plane, and physical distances between people from a single RGB image or video captured by a camera viewing a 3-D scene from a fixed vantage point. To automate camera calibration and distance estimation, we leverage priors about human pose and develop a novel direct formulation for pose-based auto-calibration and distance estimation, which shows state-of-the-art performance on publicly available datasets. The proposed approach enables existing camera systems to measure physical distances without needing a dedicated calibration process or range sensors, and is applicable to a broad range of use cases such as social distancing and workplace safety. Furthermore, to enable evaluation and drive research in this area, we contribute to the publicly available MEVA dataset with additional distance annotations, resulting in "MEVADA" – an evaluation benchmark for the pose-based auto-calibration and distance estimation problem.

Figure 1: *Sample output* (best viewed at 5× and in color) of our fully automated system (Sect. 5) with potential applications in social distancing. **Left**: The plot is generated from the Oxford Town Center dataset [20], where the grids in cyan represent the estimated ground plane (each cell is 6ft. × 6ft.), red indicates people within 6 feet from others, *i.e.*, possibly unsafe regarding social distancing guidelines [13], green means safe, the links show each person's nearest neighbor in 3-D with estimated distance (in feet) superimposed in pink. **Top right**: A top-down view of the scene. **Bottom right**: A heat map of individuals considered unsafe aggregated over time, which may guide safety measures, *e.g.*, workplace re-arrangement, to be taken.

1. Introduction

Estimating physical distances between objects from a single view is an emerging and challenging problem in computer vision. This task has wide applicability to many real-world situations such as deciding on appropriate movements in autonomous vehicles and robots, determining distances between players in sports, and estimating safe distances between people or dangerous objects in open spaces.

Standard image-based approaches to this task typically require complex factory or on-site camera calibration procedures [19, 30, 54, 57, 11] (*e.g.*, using a checkerboard) or specialized hardware. The former measures distances within an Euclidean reconstruction of a 3-D scene observed from *multiple vantage points* by at least one moving or multiple static *pre-calibrated* cameras. The latter utilizes RGB-D cameras or range sensors where metric distance measures are directly available. These cameras are expensive, not widely deployed and are limited in range and operating conditions.

Three challenges limit widespread adoption of objects distance estimation: (i) the majority of available video cameras output only RGB images, (ii) applying standard checkerboard-based calibration on-site for the vast majority of *uncalibrated, already-installed* cameras is prohibitively *expensive*, and (iii) in most security installations scenes are only observable from one camera view. We present a distance estimation method that can be applied to single-view RGB images that utilizes reference objects of roughly-known dimension present in scenes, such as cars, furniture, windows, *etc.*, to "auto-calibrate" a

*Work was done when Xiangyu was at Amazon.

fixed camera.

In particular, when the presence of people in imagery is ubiquitous, we use the people as a stand in for a calibration pattern to estimate the camera intrinsics and scene geometry. We follow the assumptions commonly used in human pose-based auto-calibration [33, 34, 6, 28, 26, 35, 29, 23, 50]: people are standing upright on a common ground plane and can be approximated as vertical line segments of known constant height. Current published methods require sequential steps of intersecting and fitting lines first to calculate the vertical vanishing point and horizon line [18], followed by extraction of camera parameters. We derive and present a simpler and more accurate approach to solve camera parameters *directly* from keypoint measurements in just three linear equations. Additionally, we jointly estimate the ground plane and the 3-D keypoints *from a single view*. Thus, our system is able to address all three aforementioned challenges. In this paper, we demonstrate distance estimation between people, but the formulation can be trivially generalized to other object classes with (roughly) known dimensions.

Summary of Contributions (i) We derived a direct formulation that simultaneously estimates the camera intrinsics, the ground plane, and reconstructs 3-D points from 2-D keypoints by solving three linear equations (Sect. 3). (ii) As an application of the formulation, we developed a fully automated system (Sect. 5) capable of estimating physical distances between people from one RGB image or video without manual calibration. (iii) As there are no proper datasets for the distance estimation task (Sect. 2), we built MEVADA on top of the publicly available Multiview Extended Video with Activities (MEVA) [27] dataset to drive research in this area (Sect. 6). The MEVADA dataset can be found at https://feixh.github.io/projects/physical_distance/.

2. Related Work

Single Image 3-D Networks predict depth maps [14, 16, 17], reconstruct 3-D shapes [46, 55, 51], or localize 3-D human skeletons [42, 56, 49, 38, 5] from single images that can be used in physical distance estimation. Yet, these methods require *known camera parameters* and are sensitive to scene geometry. Instead, we adopt a hybrid approach where a 2-D pose detector [52] provides keypoint measurements to a least-square estimator that estimates both the camera parameters and physical distances. Pose detectors, compared to single image 3-D networks, are less sensitive to camera and scene variability, and *require no 3-D supervision in training.*

Single View Metrology Using human poses to auto-calibrate a camera is not new. [33, 34, 6] use head and foot keypoints detected on human bodies to calibrate a camera, a setting similar to ours where temporal information is not used. In contrast, [28, 26, 35, 29, 23, 50] track walking people, and use tracks of head and foot keypoints to calibrate a camera. Regardless of the approach, all pose-based auto-calibration methods assume some prior about the human height – a constant in [6, 28, 26, 35, 29, 23] and a known distribution in [33, 34, 50] – and can find roots in early works on single view metrology [44, 9, 18]. Standard assumptions include (which we also adopt): a dominant ground plane that defines the horizon, people standing upright that can be approximated as vertical line segments and prior about the height.

With these assumptions, the vertical line segments are parallel to each other, and, in perspective geometry, intersect at the vertical vanishing point at infinity. The two planes spanned by the top and bottom ends of the vertical line segments are also parallel, and the image of their intersection at infinity, in perspective geometry, is the horizon line. As a result, existing methods on pose-based auto-calibration use standard techniques [1] to find the vertical vanishing point and horizon line by intersecting and fitting lines, and only then, the camera parameters are extracted.

Unlike existing methods, we forgo the error-prone line intersection and fitting steps altogether. Instead, we start from the basics and formulate the problem in three linear equations (Eq. (1), (2), and (3)), and directly compute the camera parameters by solving the linear equations. The proposed formulation is both simpler and more accurate, as shown in both simulation (Sect. 4) and real-world experiments (Sect. 6).

2-D Human Pose Detection is the task of localizing a set of human joints (keypoints) in an image [53, 25]. Two paradigms exist in pose detectors: top-down [25, 53, 40, 56, 48] and bottom-up [39, 8, 7, 41]. In the former, keypoints are predicted within bounding boxes produced by person detectors [15, 45, 21, 31]. In the latter, keypoints are predicted and grouped into full-body poses. We adopt HRNet [48] – a top-performing top-down model – in our end-to-end system (Sect. 5).

Human Pose Datasets from activity analysis to body reconstruction and in both 2-D [12, 43, 3, 32, 2] and 3-D [24, 47, 4, 37] are gaining popularity due to the increased research interest. Yet, none of the

[1] Such techniques can be found in textbooks, for instance p. 215 of [18] on computing vanishing points and p. 218 of [18] on computing vanishing lines.

existing datasets meet our needs for evaluating distance estimation: The 2-D human datasets are typically collected in natural environments but lack 3-D ground truth. The 3-D human datasets that have 3-D ground truth are typically collected in controlled environments with either one person in the the images [47, 24] or a few people but at *non-standing* positions [37] – insufficient to auto-calibrate the camera. To this end, we augment MEVA with extra distance annotation, resulting in the new MEVADA dataset (Sect. 6.1).

3. Methodology

Given an RGB image $I : \mathbb{R}^2 \supset \Omega \mapsto \mathbb{R}^3_+$ – captured by an *uncalibrated* camera that contains people, we want to estimate the *physical* distances between them. We first perform joint camera auto-calibration and metric reconstruction by solving three linear equations with measured 2-D human joint keypoints. Estimating pairwise distances is then trivial given the reconstruction. A 2-D keypoint $\mathbf{x} \in \Omega$ is the projection of a 3-D joint keypoint $\mathbf{X} \in \mathbb{R}^3$ defined on human body. Such keypoints can be easily detected by modern pose detectors, *e.g.*, HRNet. Our system follows a common assumption made by pose-based calibration (Sect. 2), where measurements are taken from any pair of keypoints that are expected to be vertically aligned with each other. In our method, we use the ankle and shoulder center points, *i.e.*, the middle point of each person's two ankle and shoulder keypoints, and represent them with $\mathbf{X}_{B,i}$ and $\mathbf{X}_{T,i}, i = 1 \cdots N$ where N is the number of people present in the image. We further assume a pinhole camera model where the principal point of the camera coincides with the image center – an assumption commonly made [28, 6, 33, 34]. Without loss of generality, we shift the 2-D keypoints such that the projection matrix takes the form $\mathbf{K} = \mathbf{diag}(f_x, f_y, 1)$.

3.1. Problem Formulation

Unlike previous works that align the ground plane to the $x-y$ plane of the reference frame within which the camera pose $g \in \mathrm{SE}(3)$ is estimated, we adopt a camera-centric formulation that eases derivation and results in a simpler and more accurate numeric solution. Specifically, we parameterize – in the camera frame – the ground plane with its normal $\mathbf{N} \in \mathbb{R}^3, \|\mathbf{N}\| = 1$, and distance $\rho > 0$ to the optical center. We then approximate the ground by the plane spanned by ankle center points $\mathbf{X}_{B,i} \in \mathbb{R}^3$ as $\mathbf{N}^\top \mathbf{X}_{B,i} + \rho = 0$ [36] in the camera frame. The 3-D coordinates of a person's shoulder center point $\mathbf{X}_{T,i} \in \mathbb{R}^3$ is then approximated by shifting the corresponding ankle center $\mathbf{X}_{B,i}$ along the normal $\mathbf{N}$ by the constant height $h > 0$ as $\mathbf{X}_{T,i} = \mathbf{X}_{B,i} + h \cdot \mathbf{N}$ following the constant height and upright standing assumptions in Sect. 2. The distance between people indexed by i and j is then $d_{i,j} = \|\mathbf{X}_{B,i} - \mathbf{X}_{B,j}\|$.

3.2. Calibration

Let $\mathbf{x}_{T,i}, \mathbf{x}_{B,i} \in \mathbb{R}^2$ be the projection of the i-th person's shoulder and ankle center point respectively, then we have the image formation equation: $\lambda_{T,i}\bar{\mathbf{x}}_{T,i} = \mathbf{K}\mathbf{X}_{T,i}$ and $\lambda_{B,i}\bar{\mathbf{x}}_{B,i} = \mathbf{K}\mathbf{X}_{B,i}$, where $\lambda > 0$ is the unknown depth, and $\bar{\mathbf{x}} \triangleq [\mathbf{x}^\top, 1]^\top$ is the homogeneous representation. Take the difference of the two projections and substitute $\mathbf{X}_{T,i} = \mathbf{X}_{B,i} + h \cdot \mathbf{N}$, we have

$$\lambda_{T,i}\bar{\mathbf{x}}_{T,i} - \lambda_{B,i}\bar{\mathbf{x}}_{B,i} = h \cdot \mathbf{K}\mathbf{N}. \tag{1}$$

To eliminate the unknown depth λ, we left multiply both sides by $\bar{\mathbf{x}}_{T,i} \times \bar{\mathbf{x}}_{B,i}$ resulting in

$$(\bar{\mathbf{x}}_{T,i} \times \bar{\mathbf{x}}_{B,i})^\top \mathbf{K}\mathbf{N} = 0 \tag{2}$$

which is a scale-invariant constraint on $\mathbf{v} \triangleq \mathbf{K}\mathbf{N}$ – the vertical vanishing point. Collecting constraints from all the people visible in the image, we have a linear system $\mathbf{A}\mathbf{v} = \mathbf{0}$ where the i-th row of $\mathbf{A} \in \mathbb{R}^{N \times 3}$ is $(\bar{\mathbf{x}}_{T,i} \times \bar{\mathbf{x}}_{B,i})^\top$. To solve it, we seek the optimal solution of the least-square problem $\min_{\mathbf{v}} \|\mathbf{A}\mathbf{v}\|_2$ subject to $\|\mathbf{v}\| = 1$ – avoiding trivial solution $\mathbf{v} = \mathbf{0}$, which can be solved via singular value decomposition (SVD) [18]. Unique solutions exist when *at least two people* in general configurations are visible.

As the linear system above is scale-invariant, we can only solve $\mathbf{v}$ up to scale, *i.e.*, we have $\tilde{\mathbf{v}} = \mu\mathbf{v}$ as our solution where $\mu \in \mathbb{R}$ is an arbitrary scaling factor. Substitute $\tilde{\mathbf{v}}$ into Eq. (1), we solve the depth $\tilde{\lambda}_{T,i}, \tilde{\lambda}_{B,i}$ of each individual up to the same scaling factor μ which we recover next.

For a pair of people indexed by $(i, j), i \neq j$ and $i, j \in \{1 \cdots N\}$, we derive the following constraint [2]:

$$\tilde{\mathbf{v}}^\top \mathbf{W}(\tilde{\lambda}_{B,i}\bar{\mathbf{x}}_{B,i} - \tilde{\lambda}_{B,j}\bar{\mathbf{x}}_{B,j}) = 0 \tag{3}$$

which is linear in $\mathbf{W} \triangleq \mathbf{diag}(1/f_x^2, 1/f_y^2, 1)$. What this constraint means is that the plane spanned by the ankle centers is indeed the ground plane, and thus orthogonal to the ground plane normal. Collecting all the pairwise constraints, we construct and solve $\mathbf{B}[1/f_x^2, 1/f_y^2]^\top = \mathbf{y}$, where $\mathbf{B} \in \mathbb{R}^{\frac{N(N-1)}{2} \times 2}$

[2] A similar constraint can be derived for shoulder centers, which, however, is *not* linearly independent from Eq. (3) and not used in our estimator.

and $\mathbf{y} \in \mathbb{R}^{\frac{N(N-1)}{2} \times 1}$. Specifically, $[\mathbf{B}|\mathbf{y}] =$

$$\begin{bmatrix} \tilde{\mathbf{v}}^\top \odot (\tilde{\lambda}_{B,1}\bar{\mathbf{x}}_{B,1} - \tilde{\lambda}_{B,2}\bar{\mathbf{x}}_{B,2})^\top \\ \vdots \\ \tilde{\mathbf{v}}^\top \odot (\tilde{\lambda}_{B,i}\bar{\mathbf{x}}_{B,i} - \tilde{\lambda}_{B,j}\bar{\mathbf{x}}_{B,j})^\top \\ \vdots \\ \tilde{\mathbf{v}}^\top \odot (\tilde{\lambda}_{B,N-1}\bar{\mathbf{x}}_{B,N-1} - \tilde{\lambda}_{B,N}\bar{\mathbf{x}}_{B,N})^\top \end{bmatrix} \tag{4}$$

where $\odot$ is component-wise product. To solve the linear system, at least two people have to be observed if we know as a priori that $f_x = f_y$, and three in the general case of $f_x \neq f_y$. Note, the two terms $1/f_x^2$ and $1/f_y^2$ impose positivity on the solution, which may not exist for noisy measurements. More details can be found in Sect. 1.1, 1.2 of Supp. Mat.

Factor Out Scale Ambiguity As the ground plane normal $\mathbf{N} = \frac{1}{\mu}\mathbf{K}^{-1}\tilde{\mathbf{v}}$ is unitary, we recover the scaling factor $\mu = \pm\tilde{\mathbf{v}}\mathbf{W}\tilde{\mathbf{v}}$ where the sign of μ is determined by ensuring the cheirality of the depth $\lambda_{B,i}, \lambda_{T,i}, i = 1 \cdots N$.

3.3. Reconstruction

The 3-D coordinates of each person's ankle and shoulder center are obtained by back-projection: $\mathbf{X}_{B,i} = \lambda_{B,i}\mathbf{K}^{-1}\bar{\mathbf{x}}_{B,i}$, $\mathbf{X}_{T,i} = \lambda_{T,i}\mathbf{K}^{-1}\bar{\mathbf{x}}_{T,i}$. We then compute the pairwise distances between people using 3-D keypoints. The ground plane offset is $\rho = \frac{1}{2}h - \mathbf{N}^\top\frac{1}{2}(\bar{\mathbf{X}}_B + \bar{\mathbf{X}}_T)$ where $\bar{\mathbf{X}}_B = \frac{1}{N}\sum_{i=1}^{N}\mathbf{X}_{B,i}$ is the centroid of the ankle centers and $\bar{\mathbf{X}}_T$ the centroid of shoulder centers.

3.4. Discussion

Comparison to Intersection & Fitting Methods In previous methods [28, 6, 33, 50], the vertical vanishing point and horizon line are first explicitly found by line intersection and fitting as described in Sect. 2. The camera parameters are only then recovered assuming known constant height of people. In contrast, the proposed formulation calibrates the camera and reconstructs the 3-D keypoints from just the 2-D keypoints by solving three linear equations, *i.e.*, Eq. (1), (2), and (3). By eliminating the error-prone line intersection and fitting step, our approach is more accurate, as shown in both simulation (Sect. 4) and real-world dataset (Sect. 6).

RANSAC The proposed method can be used in both batch mode and as a minimal solver in a RANSAC loop to further robustify the estimation. When used in RANSAC, two people (three in the case of $f_x \neq f_y$) visible in the image are needed to compute the model parameters, and the inliers are found among the rest of the data points. The result is further refined by applying the proposed method in batch mode on the maximum inlier set. In simulation (Sect. 4), we focus on analyzing the numerical behavior of the solver in outlier-free and batch mode. In real-world experiments (Sect. 6), RANSAC + batch refinement is used.

Time Complexity When RANSAC is used, a few thousands iterations are needed to achieve 0.99 confidence level where each iteration solves a small linear system in constant time. The complexity of the batch mode is $O(N^2)$ where N is the number of people in the scene. Typical runtime is a few tens of milliseconds with up to a few hundred people with our Python implementation using the standard `numpy` library on a commodity laptop with a 2.8 GHz Intel Core i7 CPU and 16 GB memory.

Nonlinear Refinement To further refine the results, we experimented with nonlinear optimization using Ceres-Solver [1] where the reprojection error of the ankle and shoulder centers are minimized with the proposed solution as initialization. However, we only observe negligible gain, and as such we *do not* apply nonlinear refinement in our method.

Lens Distortion is modeled with the 1-parameter division model of Fitzgibbon [11] and experimented in simulation. However, we found experimentally that the distortion modeling is relatively sensitive to measurement noise and leads to a method not as robust as a simple pinhole model, and as such we leave it to Sect. 1.5 of Supp. Mat.

4. Simulation

We first describe the evaluation metrics, and then analyze the sensitivity of the method to different noise sources at varying levels. In simulation, we randomly generate keypoint measurements following the height and orthogonality assumptions (Sect. 2), and pass them forward to our direct linear solver.

To demonstrate the numeric advantage of our method, we use the same simulated measurements in a baseline solver that uses line intersection and fitting. With same input, we rule out the possible influence of measurement quality on the estimation accuracy. As the prior art based on line intersection and fitting isn't publicly available, we implemented a baseline solver following [6] using the line intersection & fitting techniques in [18].

4.1. Evaluation Metrics

Focal Length Error f_{err} is defined as the normalized deviation of the estimated focal length from the true focal length, *i.e.*, $f_{err} = \frac{|\hat{f}-f|}{f} \times 100\%$, where $\hat{f}$ and f are the estimated and ground-truth focal length respectively.

Ground Plane Error consists of two terms: Normal error $\mathbf{N}_{err} = \arccos \hat{\mathbf{N}}^\top \mathbf{N}$ in degrees, *i.e.*, the angle spanned by the estimated and true ground plane normal, and distance error $\rho_{err} = \frac{|\hat{\rho}-\rho|}{\rho} \times 100\%$, where $(\hat{\mathbf{N}}, \hat{\rho})$ and $(\mathbf{N}, \rho)$ are the estimated and ground-truth ground plane parameters respectively.
Reconstruction Error of the i-th point is defined as the absolute estimation error $\|\hat{\mathbf{X}}_i - \mathbf{X}_i\|$ normalized by its distance to the camera $\|\mathbf{X}_i\|$, *i.e.*, $\mathbf{X}_{i,err} = \frac{\|\hat{\mathbf{X}}_i - \mathbf{X}_i\|}{\|\mathbf{X}_i\|} \times 100\%$, where $\hat{\mathbf{X}}_i$ and $\mathbf{X}_i$ are the estimated and ground-truth 3-D points in the camera frame. Normalization makes 3-D points at different distances to the camera contribute equally to the error metric as otherwise the un-normalized absolute error of a 3-D point grows with its distance to the camera. When multiple points exist, the average error is used as the overall quality measure.

4.2. Sensitivity Analysis and Comparison

Noise-Free Simulation In this experiment, we test our approach with images of 3 resolutions, *i.e.*, 640×480, 1280×720, and 1920×1080 – common for video cameras. For each resolution, we test 4 Field-of-Views (FOV), *i.e.*, 45, 60, 90, and 120 degrees. Let the vertical FOV be θ, the ground-truth focal length along y-axis f_y is determined from the relation $\tan(\theta/2) = \frac{H/2}{f_y}$ where H is the height of the image. The ground-truth f_x is then generated as $f_x = \frac{W}{H} \cdot f_y$ where W is the width of the image. We conduct Monte Carlo experiments of 5,000 trials for each resolution-FOV pair. In each trial, we randomly sample 3 pairs (minimal number of measurements required, see Sect. 3.2) of 3-D ankle and shoulder center points that satisfy the model assumptions as described in Sect. 2. We then project the 3-D points onto the image plane, and pass the resulting 2-D projections to both our solver and the baseline solver. We observe that, in both solvers, the estimation errors averaged over 5,000 trials are close to 0 (to machine precision) in all camera configurations – confirming the validity of both methods in noise-free scenarios. See Supp. Mat. Sect. 2 for itemized estimation errors.
Sensitivity to Measurement Noise We add zero-mean Gaussian noise of varying standard deviation to the measurements – possibly leading to unsolvable problems (Sect. 3.2), and thus we introduce *failure rate* to reflect the percentage of estimation failures. Table 1 summarizes the results where the growth of the estimation error is empirically linear relative to the noise std. To achieve a desired level of accuracy, more measurements can be used as shown in Table 3. As a comparison, the estimation error in the baseline algorithm scales similarly as the measurement noise varies, but is consistently worse than the proposed method.

		Measurement noise std. in pixels					
Error	Method	0.1	0.2	0.5	1.0	2.0	5.0
f_x(%)	**Ours**	**0.65**	**1.66**	**3.11**	**6.04**	**11.93**	**28.03**
	Baseline	1.20	2.00	4.58	8.43	14.90	32.69
f_y(%)	**Ours**	**0.73**	**1.67**	**2.99**	**5.51**	**10.52**	**24.97**
	Baseline	1.27	1.95	4.33	7.71	13.97	27.44
$\mathbf{N}(^\circ)$	**Ours**	**0.09**	**0.18**	**0.45**	**0.90**	**1.84**	**4.60**
	Baseline	0.20	0.33	0.83	1.71	3.72	9.70
ρ(%)	**Ours**	**0.24**	**0.50**	**1.23**	**2.38**	**4.86**	**12.41**
	Baseline	0.51	0.88	2.24	5.15	13.18	39.08
$\mathbf{X}$(%)	**Ours**	**0.67**	**1.36**	**2.88**	**5.33**	**10.71**	**24.70**
	Baseline	2.16	3.27	6.16	10.56	17.31	31.57
fail (%)	**Ours**	**0.08**	**0.32**	**1.06**	**1.52**	**4.10**	**13.72**
	Baseline	0.32	0.40	1.76	3.84	8.24	27.28

Table 1: *Estimation error as measurement noise varies.* We fix image resolution to 1920×1080, FOV to $90°$, and conduct Monte Carlo experiments of 5,000 trials. Zero-mean Gaussian noise of varying standard deviation is added to the measurements.

Sensitivity to Height One of the assumptions widely used in pose-based calibration, which we also use, is that the people in the scene are of a *known constant height*. This is not true in reality. We test the sensitivity of our system to height variation by perturbing the height of the people in the scene so that its distribution follows a truncated Gaussian, then generate and pass noisy measurements to our solver. Results in Table 2 show that our solver is quite robust to height variation. Regardless of the variation, our proposed method has consistently smaller estimation error than the baseline.
Number of Measurements While our method works with a minimal number of measurements in a single image, we found that both the estimation error and failure rate are reduced given more measurements. The practical meaning of the finding is that overall system performance can be improved by aggregating measurements from multiple video frames. Table 3 summarizes the results. The estimation error in the baseline does not reduce as fast as that in our method, and is consistently larger.

		Std. of height in meters				
Error	Alg.	0.05	0.1	0.15	0.2	0.25
f_x(%)	**Ours**	**4.66**	**8.18**	**11.60**	**13.10**	**13.87**
	Baseline	7.39	11.77	14.14	15.73	16.14
f_y(%)	**Ours**	**5.20**	**7.52**	**11.17**	**11.74**	**12.25**
	Baseline	6.80	10.39	12.39	13.97	14.78
$\mathbf{N}(^\circ)$	**Ours**	**0.79**	**1.35**	**1.87**	**2.20**	**2.40**
	Baseline	1.41	2.42	3.02	3.64	3.77
ρ(%)	**Ours**	**2.06**	**3.52**	**4.88**	**5.77**	**6.45**
	Baseline	3.71	6.27	7.83	9.19	9.78
$\mathbf{X}$(%)	**Ours**	**4.91**	**8.03**	**10.52**	**12.36**	**13.81**
	Baseline	8.56	14.38	15.80	20.41	20.37
fail (%)	**Ours**	**1.44**	**2.30**	**3.48**	**4.46**	**5.36**
	Baseline	2.74	4.66	6.12	6.80	7.68

Table 2: *Estimation error as height varies.* The setup is the same as in Table 1 except that we (i) fix the measurement noise std to 0.5 pixels, and (ii) sample height of people from a truncated Gaussian ranging from $1.5m$ to $1.9m$, with a mean of $1.7m$, and a varying standard deviation.

		Number of people				
Error	Alg.	5	10	20	50	100
f_x(%)	Ours	**21.03**	**14.15**	**8.17**	**6.36**	**5.27**
	Baseline	21.53	15.62	12.40	11.59	12.39
f_y(%)	Ours	20.38	**11.47**	**8.24**	**5.44**	**4.78**
	Baseline	**17.53**	14.11	13.05	12.79	12.55
$\mathbf{N}(^\circ)$	Ours	**3.87**	**2.21**	**1.39**	**0.92**	**0.76**
	Baseline	4.87	3.62	2.93	2.70	2.61
ρ(%)	Ours	**10.24**	**5.79**	**3.87**	**2.66**	**2.19**
	Baseline	13.71	9.33	7.60	6.77	6.19
$\mathbf{X}$(%)	Ours	**16.31**	**12.73**	**10.39**	**9.50**	**9.31**
	Baseline	20.55	19.94	18.00	18.57	17.91
fail (%)	Ours	**9.88**	**4.70**	**2.28**	**1.84**	**1.52**
	Baseline	13.78	7.20	5.1	4.2	4.0

Table 3: *Estimation error as the number of people varies.* The setup is the same as in in Table 2 except that we (i) set height std to $0.1m$, and (ii) vary the number of measurements.

5. System Overview

By combining our calibration and reconstruction algorithm with an off-the-shelf pose detection model, we build a fully automated distance estimation system that runs end-to-end: It takes as input either a single RGB image or a video captured by the same static camera, and outputs the camera calibration and 3-D keypoints on human bodies. See Fig. 1 for a sample output on the Oxford Town Center dataset [20]; a demo video on the same dataset is included in Supp. Mat.

We adopt HRNet [48] – a top-performing pose detector. As HRNet is a top-down method that requires image crops of people as input, we first run just the person detector from Mask R-CNN [21] and then pass the bounding boxes to HRNet. Both HRNet and Mask R-CNN are trained on COCO dataset [32]. Fig. 2 shows a flowchart of the proposed system where the pose detection pipeline takes a single image and outputs a list of 2-D keypoints, and then the estimator takes the keypoints as the input and estimates both the calibration parameters $(\mathbf{K}, \mathbf{N}, \rho)$ and the 3-D keypoints $(\mathbf{X})$. For a video

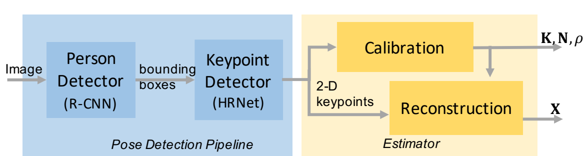


Figure 2: *System flowchart* of the proposed human pose-based auto-calibration and distance estimation system.

recorded at a fixed viewpoint, we iterate the video frames through the pose detection pipeline. The per frame keypoints are then concatenated into one batch and passed to the estimator at once. In the estimator, RANSAC followed by batch refinement on the maximum inlier set is performed. This process is used on video data in real-world experiments (Sect. 6.2).

6. Experiments

The proposed system has two functionalities, *i.e.*, camera auto-calibration and distance estimation, and we evaluate both in this section. The calibration functionality is evaluated on three publicly available datasets: vPTZ [43], POM [12], and MEVA [27]. All the datasets contain videos of both indoor and outdoor human activities and are captured by *calibrated* cameras. Subsets of vPTZ and POM have been used by prior art [33, 34, 6, 50] to evaluate pose-based auto-calibration, which we also use and compare in Sect. 6.2. Yet, neither of the three datasets mentioned above nor any existing 2-D & 3-D human pose datasets are suitable to evaluate the *distance estimation task* as discussed in Sect. 2. We address this problem by augmenting MEVA in Sect. 6.1. We then evaluate our distance estimation method along with a learning-based baseline in Sect. 6.3.

6.1. MEVADA: MEVA with Distance Annotation

MEVA [27] consists of a few thousand videos of human activities captured by *calibrated cameras* but no ground-truth 3-D position annotation. We augment it with distance annotation resulting in the new "MEVA with Distance Annotation" dataset or MEVADA for short. To construct MEVADA, we first sample videos captured by each camera in MEVA that contains ground-truth calibration. We then randomly sample pairs of people bounding boxes in each video frame predicted by Mask R-CNN [21]. Multiple annotators choose one of the 4 distance categories for the sampled pairs: (a) `0 - 1 meters`, (b) `1 - 2 meters`, (c) `2 - 4 meters`, and (d) `greater than 4 meters`, and the majority vote is used as the ground-truth label. We also asked the annotators to sort out erroneous bounding boxes predicted by Mask R-CNN. Discrete labels instead of continuous ones are used in annotation because it's difficult for humans to accurately perceive 3-D distances from 2-D images, and as such the annotators often feel more comfortable and do better with ranges. To collect true distances, careful setup with multiple cameras and/or range sensors would be required, which we leave as future work.

In summary, MEVADA contains 4,667 frames annotated with ground-truth metric distances which we split into a test set of 746 images and a training set of 3,921 image. We train a baseline model for distance estimation on the training set, and then evaluate both the baseline and the proposed distance estimator on the test set. Out of the 28 calibrated cameras in MEVA, 2 cameras have videos that contain *no people at all* – infeasible to auto-calibrate, and thus we ignore them in evaluation.

6.2. Evaluate Camera Calibration

Each video clip in MEVA is 5 minutes long and does not contain people most of the time. As such, for each of the 26 test cameras in MEVA, we randomly sample up to 5 video clips to ensure that HRNet detects a reasonable number of human keypoints. We then concatenate the per frame keypoints into a batch and perform RANSAC + batch refinement as described in Sect. 5. As the ground truth in MEVA does not contain ground plane parameters, we report only the focal length error in Table 4 and show ground plane estimation qualitatively in Fig. 4. Itemized estimation error for each camera can be found in Supp. Mat. Sect. 4. The results show that our pose-based auto-calibration method captures the focal length reasonably well (up to $\sim 45\%$ error) 90% of the time on challenging real-world data. Some of the test cameras only contain videos where all the people are very far way from the camera or distributed quite unevenly in space which challenge our calibration algorithm – the former contains significantly more measurement noise and the latter biases our solver. These challenges cause the relatively high estimation error as shown in Table 4. Better calibration can be expected if the camera observes the scene for a sufficiently long time period such that people will appear in the close proximity of the camera and more evenly in space.

Error (unit)	min	max	P_{90}	mean	std
$f_x(\%)$	2.57	70.27	45.36	23.21	16.75
$f_y(\%)$	0.44	66.12	46.71	25.49	17.46

Table 4: *Camera calibration* on MEVA. P_{90} indicates the 90-th percentile of the error distribution.

We compare our focal length estimation on vPTZ [43] and POM [12] datasets against the prior art [33, 34, 6, 50] as shown in Table 5. Our new formulation shows a significant reduction in estimation error on most test sequences compared to the prior art as we avoid the error-prone line intersection and fitting procedures.

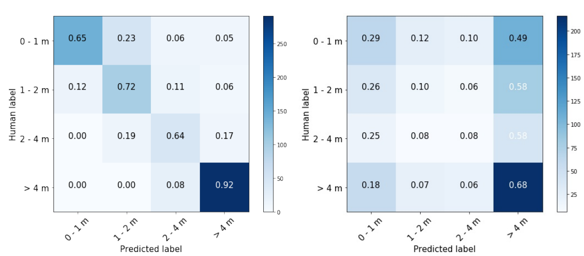
We additionally compare our approach to ESTHER [50] – a state-of-the-art calibration method based on *pose tracking*. We note that the typical runtime of [50] is about 50 seconds due to their evolution optimization scheme[3], whereas our linear solver achieves better or comparable results as shown in Table 5 at a fraction of the cost (typically a few milliseconds, see Sect. 3.2 Time Complexity).

Seq.	$f_{gt}(pix.)$	method	$f_{est}(pix.)$	error
#1 vPTZ [43] set1-cam-131	1056.81	[33]	**1044**	**1%**
		[34]	1034	2%
		[6]	N/A	
		[50]	N/A	
		Ours	1106.37	4.7 %
#2 vPTZ [43] set1-cam-132	1197.80	[33]	1545	29%
		[34]	1427	19%
		[6]	1019	15%
		[50]	N/A	10.14%
		Ours	**1201.94**	**0.35 %**
#3 vPTZ [43] set2-cam132	1048.15	[33]	N/A	
		[34]	N/A	
		[6]	787	24.92%
		[50]	N/A	12.07%
		Ours	**1160.76**	**10.74 %**
#4 POM [12] terrace1-cam0	807	[33]	N/A	
		[34]	N/A	
		[6]	850	5.33%
		[50]	N/A	**1.43%**
		Ours	786.77	2.51%

Table 5: *Comparison of focal length estimation.* We report our result along with results taken from Table 2 & 4 of [6], and Table 2 of [50] – the best performing pose-based auto-calibration method. f_{gt} and f_{est} are the ground-truth and estimated focal length. N/A means the results are not reported by the cited paper. The best results are shown in **bold**.

6.3. Evaluate Distance Estimation

We reconstruct the 3-D points using cameras calibrated in Sect. 6.2, and then compute the pairwise distances between people. Unlike in simulation where we have 3-D ground truth, we only have *discrete* distance labels and thus treat the evaluation of distance estimation as a classification problem: We quantize the estimated distances into 4 categories and compare them against the ground-truth labels in the test set resulting in the confusion matrix as shown in Fig. 3a. The results show that our method



(a) *Ours* (b) *Baseline*

Figure 3: *Confusion matrix of distance estimation* (best viewed at 5× in color) on the MEVADA test set.

captures the correct distances majority of the times. The model achieves high precision, especially at both ends of the spectrum - where people are either between 0 - 1 meters or greater than 4 meters apart. Model performance drops for the two distance classes in the middle. Intuitively, these cases are challenging

[3][50] claims $O(N^2)$ complexity of sub-systems of their method. However, the convergence rate of their evolution algorithm is not addressed, and only empirical runtime is reported.

even for human annotators, *e.g.*, it's possible for human to mis-classify people that are 2.2 meters away into `1 - 2` meters class. While this is a limitation of using coarse labels for the evaluation task, we find that our model is fairly robust as most mis-classified cases fall into adjacent distance ranges. Table 6 shows the precision-recall and F1 score for each category as well as the overall accuracy, Fig. 4 shows visual results, and with more shown in Supp. Mat. Sect. 4.

Label	method	precision	recall	F1-score
`0 - 1 meters`	Ours	**0.90**	**0.65**	**0.76**
	Baseline	0.37	0.29	0.32
`1 - 2 meters`	Ours	**0.61**	**0.72**	**0.66**
	Baseline	0.20	0.10	0.13
`2 - 4 meters`	Ours	**0.46**	**0.64**	**0.54**
	Baseline	0.11	0.08	0.09
`> 4 meters`	Ours	**0.90**	**0.92**	**0.91**
	Baseline	0.49	0.68	0.57
Accuracy	Ours	**0.78**		
	Baseline	0.40		

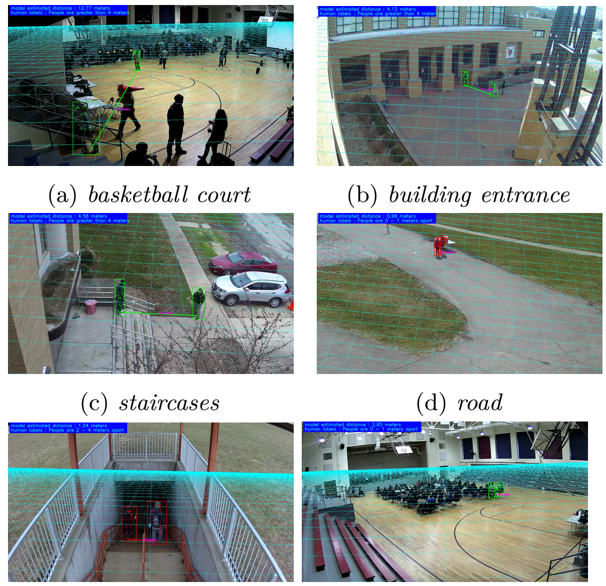
Table 6: *Distance estimation* on the MEVADA test set.

Learning-based Baseline We additionally compare our distance estimator against a learning-based baseline, where a ResNet [22]-backed classifier pretrained on ImageNet [10] is modified to perform 4-way classification and is fine-tuned on the MEVADA training set. The baseline model takes as input a blank image with crops around the selected pair of people superimposed, and predicts the distance label. We benchmark the baseline on the MEVADA test set and show the results in Fig. 3b and Table 6. It's not hard to see that the baseline mis-classifies a majority of cases to either the `0 - 1 meters` or the `greater than 4 meters` class which is expected as the learning model is both sensitive to scene geometry and data hungry. On the other hand, our pose-based distance estimator is *free of distance supervision*, and yet significantly outperforms the baseline.

6.4. Qualitative Analysis

We include visual results in Fig. 4 to better demonstrate both the strengths and limitations of the proposed method. Fig. 4a, 4b, 4c, and 4d are examples where (i) the estimated ground planes accurately represent the planes in physical world, and (ii) the estimated distances are consistent with the ground truth. Furthermore, by comparing against reference objects in the scene, we confirm both the estimated and ground-truth distances are reasonable. For instance, in Fig. 4a, one person is standing near the 3-point line whereas the other person is right outside of the baseline, and our distance estimate is 12.8 meters – consistent with the length of a half court that is 14.23 meters. We also find that our method is relatively robust to moderate lens distortion (Fig. 4b) and small elevation above the ground (Fig. 4c).

On the other hand, several scenarios pose challenges for our method. The situation where people walk on staircases as shown in Fig. 4e violates the *common ground plane* assumption and cause inaccuracies in calibration. The crowded scene makes human keypoints hardly visible leading to unreliable distance estimation as shown in Fig. 4f.



(a) *basketball court* (b) *building entrance* (c) *staircases* (d) *road* (e) *stairwell* (f) *gym*

Figure 4: *Sample results on MEVADA* (best viewed at 5× and in color). The regular grids in cyan represent the ground plane. Each grid cell is $2m \times 2m$. Distance estimates (in meters) of selected people (in colored bounding boxes) and ground-truth labels are shown on top-left corner. See Sect. 6.4 for more details.

7. Conclusion

We find the system effective when the basic assumptions are satisfied: average height, ground plane and linearity in camera intrinsics. Common settings satisfying these assumptions include pedestrian at traffic locations, factory floors, and public areas such as malls and walkways. Similar to prior methods, the system performance suffers when these assumptions are violated, *e.g.*, uneven ground plane with staircases and sloped ramps, human with non-standing positions such as sitting, and highly distorted camera lens, *etc.* Additional filters can be added in scenes containing sitting people or people whose average heights are disparate, such as the difference between children, women and men, at the cost of increased system complexity. If needed, lens distortion can be corrected using the plumb-line constraint or learning-based methods.

References

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org. 4

[2] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5167–5176, 2018. 2

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 2

[4] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005. 2

[5] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. Monoloco: Monocular 3d pedestrian localization and uncertainty estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6861–6871, 2019. 2

[6] Guido MYE Brouwers, Matthijs H Zwemer, Rob GJ Wijnhoven, et al. Automatic calibration of stationary surveillance cameras in the wild. In *European Conference on Computer Vision*, pages 743–759. Springer, 2016. 2, 3, 4, 6, 7

[7] 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. 2

[8] 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. 2

[9] Antonio Criminisi, Ian Reid, and Andrew Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000. 2

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 8

[11] Andrew W Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001. 1, 4

[12] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):267–282, 2007. 2, 6, 7

[13] Centers for Disease Control and Prevention. Social distancing – keep a safe distance to slow the spread. "https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/social-distancing.html". 1

[14] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*. Springer, 2016. 2

[15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[16] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 2

[17] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8977–8986, 2019. 2

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

[19] Richard I Hartley. An algorithm for self calibration from several views. In *Cvpr*, volume 94, pages 908–912. Citeseer, 1994. 1

[20] Jules. Harvey, Adam. LaPlace. Megapixels: Origins and endpoints of datasets created “in the wild”, 2019-2020. 1, 6

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 6

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[23] Shiyao Huang, Xianghua Ying, Jiangpeng Rong, Zeyu Shang, and Hongbin Zha. Camera calibration from periodic motion of a pedestrian. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3025–3033, 2016. 2

[24] 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. 2, 3

[25] Arjun Jain, Jonathan Tompson, Mykhaylo Andriluka, Graham W Taylor, and Christoph Bregler. Learning human pose estimation features with convolutional networks. *arXiv preprint arXiv:1312.7302*, 2013. 2

[26] Imran N Junejo and Hassan Foroosh. Trajectory rectification and path modeling for video surveillance. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–7. IEEE, 2007. 2

[27] Kitware. Meva: Multiview extended video with activities. "https://mevadata.org/". 2, 6

[28] Nils Krahnstoever and Paulo RS Mendonca. Bayesian autocalibration for surveillance. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1858–1865. IEEE, 2005. 2, 3, 4

[29] Worapan Kusakunniran, Hongdong Li, and Jian Zhang. A direct method to self-calibrate a surveillance camera by observing a walking pedestrian. In *2009 Digital Image Computing: Techniques and Applications*, pages 250–255. IEEE, 2009. 2

[30] David Liebowitz and Andrew Zisserman. Metric rectification for perspective images of planes. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pages 482–488. IEEE, 1998. 1

[31] 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. 2

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 6

[33] Jingchen Liu, Robert T Collins, and Yanxi Liu. Surveillance camera autocalibration based on pedestrian height distributions. In *Proceedings of the British Machine Vision Conference*, page 144, 2011. 2, 3, 4, 6, 7

[34] Jingchen Liu, Robert T Collins, and Yanxi Liu. Robust autocalibration for a surveillance camera network. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 433–440. IEEE, 2013. 2, 3, 6, 7

[35] Fengjun Lv, Tao Zhao, and Ramakant Nevatia. Camera calibration from video of a walking human. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1513–1518, 2006. 2

[36] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2012. 3

[37] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *2018 International Conference on 3D Vision (3DV)*, pages 120–130. IEEE, 2018. 2, 3

[38] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10133–10142, 2019. 2

[39] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in neural information processing systems*, pages 2277–2287, 2017. 2

[40] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 2

[41] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–286, 2018. 2

[42] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017. 2

[43] Horst Possegger, Matthias Rüther, Sabine Sternig, Thomas Mauthner, Manfred Klopschitz, Peter M. Roth, and Horst Bischof. Unsupervised Calibration of Camera Networks and Virtual PTZ Cameras. In *Proc. Computer Vision Winter Workshop (CVWW)*, 2012. 2, 6, 7

[44] Ian Reid and Andrew Zisserman. Goal-directed video metrology. In *European Conference on Computer Vision*, pages 647–658. Springer, 1996. 2

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[46] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *Advances in neural information processing systems*, pages 4996–5004, 2016. 2

[47] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1-2):4, 2010. 2, 3

[48] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 2, 6

[49] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018. 2

[50] Zheng Tang, Yen-Shuo Lin, Kuan-Hui Lee, Jenq-Neng Hwang, and Jen-Hui Chuang. Esther: Joint

camera self-calibration and automatic radial distortion correction from tracking of walking humans. *IEEE Access*, 7:10754–10766, 2019. 2, 4, 6, 7

[51] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019. 2

[52] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems*, pages 1799–1807, 2014. 2

[53] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 2

[54] Bill Triggs. Autocalibration from planar scenes. In *European conference on computer vision*, pages 89–105. Springer, 1998. 1

[55] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2626–2634, 2017. 2

[56] 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. 2

[57] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000. 1