Extrinsic Camera Calibration from A Moving Person

Sang-Eun Lee, Keisuke Shibata, Soma Nonaka, Shohei Nobuhara, and Ko Nishino

***Abstract*—We propose a novel camera calibration method for a room-scale multi-view imaging system. Our key idea is to leverage our articulated body movements as a calibration target. We show that a freely moving person provides trajectories of a set of oriented points (*e.g.*, neck joint with spine direction) from which we can estimate the locations and poses of all cameras observing them. The method only requires the cameras to be synced and that 2D human poses are estimated in each view sequence. By elevating these 2D poses to 3D which directly provides a set of oriented 3D joints, we compute the extrinsic parameters of all cameras with a linear algorithm. We also show that this enables self-supervision of the 3D joint estimator for refinement, and the iteration of the two leads to accurate camera extrinsics and 3D pose estimates up to scale. Extensive experiments using synthetic and real data demonstrate the effectiveness and flexibility of the method. The method will serve as a useful tool to expand the utility of multi-view vision systems as it eliminates the need for cumbersome on-site calibration procedures.**

***Index Terms*—Calibration and Identification, Computer Vision for Automation, Camera Calibration, Self-supervised Learning**

I. Introduction

MULTI-camera calibration is an essential requirement for 3D computer vision including geometry reconstruction, scene understanding, and motion analysis. It usually consists of two separate steps, namely estimation of the camera intrinsics consisting of the projection model parameters and camera extrinsics that represent the location and pose in the world coordinate frame. Although the intrinsic parameters can be estimated prior to deployment, the extrinsic parameters are only fixed after installation. As such, extrinsic camera calibration always needs to happen on-site. Multi-camera extrinsic calibration fundamentally relies on point correspondences between different camera viewpoints. This inevitably necessitates special calibration equipment and procedures on deployment, such as walking around with a large calibration board. The resulting calibration accuracy depends on the coverage and precision of this non-trivial manual task. This cumbersome on-site extrinsic camera calibration severely limits the adoption of computer vision systems as it requires successful completion of a meticulous process by an experienced person that cannot be fulfilled in many cases. By way of example, imagine installing a multi-view camera system at home for an elderly. Even though computer vision can greatly help in realizing ambient assisted living, they cannot be deployed because elderlies cannot easily calibrate the cameras!

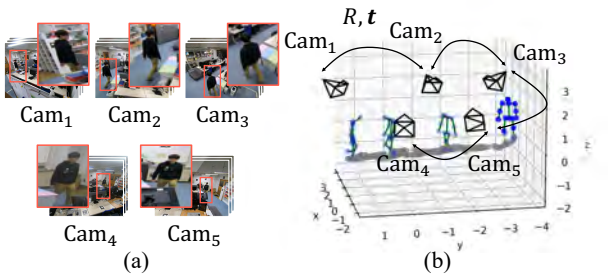

Fig. 1. We present a novel multi-camera extrinsic calibration by leveraging the human body as a calibration target. Given a set of synchronized videos capturing a moving human from different views as input (a), by using a set of oriented 3D joints of the human body, our method estimates the extrinsic parameters of each static camera (b) with a linear algorithm which can optionally be followed with non-linear bundle adjustment. The gray dots denoting the trajectory of a moving person in space and the 3D skeletons in (b) are generated to quantitatively verify the accuracy of extrinsic camera calibration.

Can we get rid of this on-site extrinsic camera calibration requirement altogether? Can we just ship cameras intrinsically calibrated at the factory and let people install them on their room ceiling and monitor their health, detect their falls, and predict their behaviors? For this we need to answer the following question. What can we use as a calibration target for multi-view cameras that we can find in daily environments and how can we use them for accurate extrinsic parameter estimation?

In this paper, we show that we can calibrate multiple cameras by just observing us, freely moving people. By simply walking around without any special intent in front of the cameras, we can collect sufficient information to accurately calibrate the extrinsics of them. The key idea is to exploit the fact that our body is articulated. Several past methods [1]–[4] have indeed shown that human joints can provide automatic correspondences between cameras, *e.g.*, left shoulder is left shoulder in all camera views, and have used them for extrinsic camera calibration. In contrast, we leverage the fact that the moving human body provides us with corresponding *oriented points* in the free space captured by the cameras and show that they lead to a much more accurate and flexible extrinsic camera calibration.

Our method assumes that the cameras are stationary, temporally in sync, and only requires 2D poses detected in each frame in each view. Off-the-shelf deep 2D human pose estimation methods suffice for obtaining these 2D poses. We first elevate these individual 2D poses to 3D with an existing 3D

Manuscript received: Feb., 24, 2022; Revised May., 25, 2022; Accepted Jun., 27, 2022.

This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments.

All authors are with Graduate School of Informatics, Kyoto University.

Digital Object Identifier (DOI): see top of this page.

pose estimator. This 3D pose needs not be accurate as we will refine the 3D pose estimator itself later. From the 3D human poses for each camera view, we can extract corresponding 3D oriented points. For instance, the neck joint with the spine direction can be used as one oriented point. Using the collection of 3D oriented points across all camera views across time, we solve for the extrinsic camera parameters via matrix factorization with RANSAC for robustness. The optimal camera poses up to scale are obtained by minimizing the reprojection errors in a bundle adjustment (BA) formulation. In addition, we show that we can use the estimated camera poses to fine-tune the 3D pose estimator in a self-supervised manner. By iterating this linear extrinsic camera calibration and fine-tuning, we arrive at accurate camera poses all just by observing a freely moving person.

We validate our method with an extensive set of experiments using synthetic and real data. We also compare with baseline methods and show that our method outperforms them in a wide variety of scenes. We also experimentally evaluate the accuracy of refining the 3D pose estimator. These results clearly demonstrate the effectiveness of our method and its flexibility as a practical extrinsic camera calibration tool that can be widely used in real-world situations. We plan to release our code with the hope that it will serve as a fundamental tool to catalyze expanded utility of multi-view vision systems.

II. Related works

a) Multi-View Camera Calibration: Multi-view camera calibration [5] is a challenging problem in particular for "in-the-wild" scenarios. Agrawal *et al*. [6] proposed a calibration and a 3D shape reconstruction pipeline from a large scale collection of uncalibrated images. To calibrate moving multi-view cameras [7]–[9], Hasler *et al*. [7] proposed an SfM-based approach. These methods, however, rely on feature points on static objects (*e.g*., background), and do not make use of dynamic foreground objects such as humans themselves as calibration targets.

b) Human As A Calibration Target: The idea of using foreground objects for both 3D shape reconstruction and calibration has been proposed by Furukawa and Ponce [10]. They assume that an initial guess of the camera parameters is available, and optimize it via 3D shape reconstruction of general objects, including human, as a collection of textured patches.

When reconstructing the geometry of a human body, the target human body itself can serve as a calibration target. Detecting persons [11] or joints in images can provide semantic correspondences between different views regardless of the baseline length, and the joints move as an articulated object due to the kinematic structure [1]–[4], [11]–[16]. Elhayek *et al*. [12] proposed simultaneous reconstruction of the human skeleton and camera poses as an energy minimization problem. Puwein *et al*. [2] obtained 2D joint correspondences to calibrate the cameras with SfM, and Takahashi *et al*. [1] combined SfM and PnP. Garau *et al*. [16] obtained 3D human poses directly from images to estimate the relative camera poses. Lv *et al*. [13] used silhouettes of a walking person to detect vanishing points for calibration. Nakano [4] proposed a closed-form solution using parallel line segments. By assuming that people in the scene always appear in upright posture, they used the line connecting the neck and the mid-hip as vertical and hence parallel segments. In contrast, we propose a new linear algorithm that uses both 2D joint detection and single-image 3D pose estimation without making any assumptions on the human pose.

c) Human Pose Estimation: Human pose estimation from a single image has been an emerging research topic in recent years. For 2D joint estimation [17]–[22], Wei *et al*. [18] proposed confidence map estimation of each joint, and Cao *et al*. [19] proposed part affinity field estimation. For 3D joint estimation [23]–[30], Mehta *et al*. [23] estimated 3D skeleton pose directly from an image, and Bogo *et al*. [24] fit a 3D template to images. Pavllo *et al*. [27] estimated 3D poses from a time series of 2D joint positions. While these 3D pose estimators work successfully on large-scale datasets, they become inaccurate on unseen views [31]. We use an existing 3D pose estimator pretrained on large-scale public datasets, and demonstrate that our method enables fine-tuning for a new scene with self-supervision.

d) Self-supervision for 3D pose estimation: Multi-view observations can provide 3D annotation of human joints automatically [32]–[34]. Simon *et al*. [33] triangulated hand keypoints from reliable 2D detections in multi-view images and reprojected them to other views to train the detector. Rhodin *et al*. [34] used estimated 3D human body for rotation estimation, and trained their 3D pose estimator to return 3D poses consistent with the rotation. We also train a pose estimator by reprojecting estimated 3D joints to the images as additional annotations by estimating both the rotation and translation of each camera.

III. Calibration from Oriented Joints of Moving Person

We fully leverage the moving human body to calibrate extrinsic camera parameters. The key idea is to use human pose as a collection of *oriented* points, instead of using it as mere 2D or 3D corresponding points.

Consider a camera c of pose R^c and $\boldsymbol{t}^c$. An oriented point $\langle \boldsymbol{x}_i, \boldsymbol{v}_i \rangle$ $(i = 1, \ldots, N)$ at position $\boldsymbol{x}_i \in \mathbb{R}^3$ and oriented to $\boldsymbol{v}_i \in \mathbb{R}^3$ in the world coordinate system appears as $\langle \boldsymbol{x}_i^c, \boldsymbol{v}_i^c \rangle$ $(i = 1, \ldots, N)$ in camera c coordinate system as

$$\boldsymbol{x}_i^c = R^c \boldsymbol{x}_i + \boldsymbol{t}^c \,, \tag{1}$$

$$\boldsymbol{v}_i^c = R^c \boldsymbol{v}_i \,, \tag{2}$$

and the point is projected to $\boldsymbol{y}_i^c \in \mathbb{R}^2$ in the camera c image as

$$\lambda_i^c \begin{bmatrix} \boldsymbol{y}_i^c \\ 1 \end{bmatrix} = K^c \boldsymbol{x}_i^c = K^c \left(R^c \boldsymbol{x}_i + \boldsymbol{t}^c \right) \,, \tag{3}$$

where K^c is the pre-calibrated intrinsic camera parameter and λ_i^c is a scaling factor corresponding to the distance between the camera and the point.

The goal of the proposed method is to estimate the extrinsic parameters $\langle R^c, \boldsymbol{t}^c \rangle \in \mathrm{SE}(3)$ of each camera c linearly from the oriented points. We assume that all the cameras are synchronized beforehand, and that the index i encodes both the

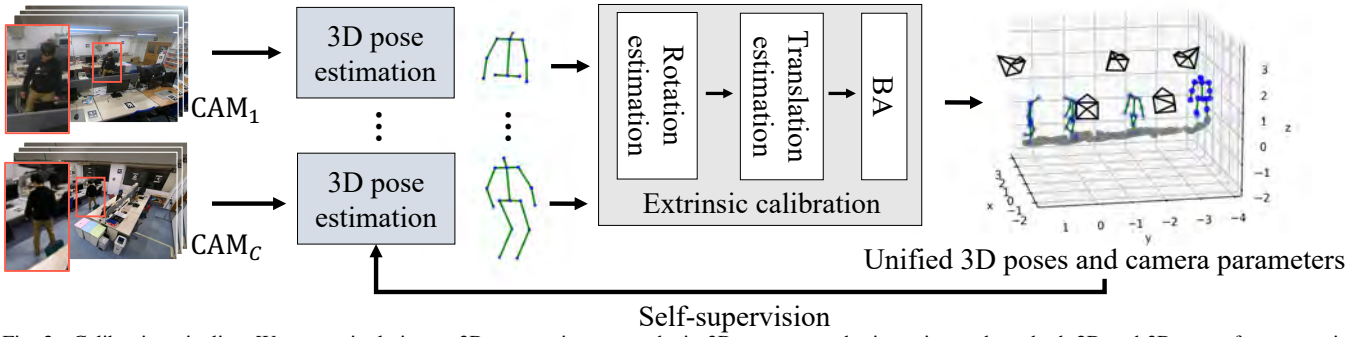

Fig. 2. Calibration pipeline. We use a single-image 3D pose estimator to obtain 3D poses at each viewpoint, and use both 2D and 3D poses for our extrinsic calibration. The calibration provides a unified 3D pose which enables fine-tuning of the 3D pose estimator to recalibrate the cameras again.

body part and the capture frame: $i = (k, f)$, where $k = 1, \ldots, K$ denotes the index of body part and $f = 1, \ldots, F$ denotes the frame.

A. Rotation Estimation

Given a set of $\boldsymbol{v}_i^c$ of C cameras, *i.e.*, the transformations of unknown 3D orientations $\boldsymbol{v}_i$ in the camera coordinate system, we first estimate the rotation matrix R^c satisfying the orthonormality using Eq. (2). For N 3D orientations estimated for a camera c, we have, by transposing both sides of Eq. (2),

$$\begin{aligned} \begin{bmatrix} \boldsymbol{v}_1^c & \ldots & \boldsymbol{v}_N^c \end{bmatrix}^\top &= \begin{bmatrix} \boldsymbol{v}_1 & \ldots & \boldsymbol{v}_N \end{bmatrix}^\top R^{c\top}, \\ \Leftrightarrow \quad V^c &= V R^{c\top}, \end{aligned} \tag{4}$$

where V^C is an $N \times 3$ observation matrix and V is an $N \times 3$ matrix of 3D unknown orientations in the world coordinate system. Observing the same set of unknown 3D orientations V by another camera c', we have

$$\begin{bmatrix} V^c & V^{c'} \end{bmatrix} = V \begin{bmatrix} R^{c\top} & R^{c'\top} \end{bmatrix}, \tag{5}$$

and for C cameras in general,

$$\begin{aligned} \begin{bmatrix} V^1 & \cdots & V^C \end{bmatrix} &= V \begin{bmatrix} R^{1\top} & \cdots & R^{C\top} \end{bmatrix}, \\ \Leftrightarrow \quad V^{1:C} &= V R^{1:C}, \end{aligned} \tag{6}$$

where $V^{1:C}$ is an $N \times 3C$ observation matrix and $R^{1:C}$ is a $3 \times 3C$ rotation matrix.

Eq. (6) indicates that the matrix $V^{1:C}$ is rank 3, and hence its SVD $V^{1:C} = YDZ^\top$ returns Y, D, and $Z^\top$ as $N \times 3$, 3×3, and $3 \times 3C$ matrices, respectively. As a result, we can factorize $V^{1:C}$ as

$$V = YDM^{-1}, \quad R^{1:C} = MZ^\top, \tag{7}$$

where M is an arbitrary invertible 3×3 matrix. Here, we choose M to make the recovered camera pose matrices become orthonormal and also to make the recovered orientations normalized by using the following proposition.

Proposition 3.1: Any rotation matrix scaled by $\sqrt{C}$ can be the matrix M in Eq. (7) that makes the recovered orientations row-normalized and also makes the recovered camera poses orthonormal.

Please refer to the appendix for a proof. In practice, we can use M given as the inverse of the left-most 3×3 submatrix in $Z^\top$ to make the rotation matrix of the first camera $R^{1\top}$ become an identity matrix.

B. Translation Estimation

Given the rotations of the cameras, we can estimate translations from collinearity and coplanarity constraints [5].

a) Collinearity Constraint: Eq. (1) suggests that $\boldsymbol{x}_i^c$ and its projection in the normalized camera coordinate system are collinear, *i.e.*, their cross product is zero:

$$\begin{aligned} \boldsymbol{n}_i^c \times \boldsymbol{x}_i^c &= [\boldsymbol{n}_i^c]_\times (R^c \boldsymbol{x}_i + \boldsymbol{t}^c) \\ &= \begin{bmatrix} [\boldsymbol{n}_i^c]_\times R^c & [\boldsymbol{n}_i^c]_\times \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{t}^c \end{bmatrix} = \boldsymbol{0}_{3\times 1}, \end{aligned} \tag{8}$$

where $[\boldsymbol{n}_i^c]_\times$ denotes the skew-symmetric matrix of $\boldsymbol{n}_i^c = [n_{i,x}^c, n_{i,y}^c, n_{i,z}^c]^\top = (K^c)^{-1}[\boldsymbol{y}_i^c, 1]^\top$.

b) Coplanarity Constraint: The rays back-projected through the corresponding points from two views form an epipolar plane, and hence $\boldsymbol{n}_i^c$, $\boldsymbol{n}_i^{c'}$, and $\boldsymbol{t}^c - \boldsymbol{t}^{c'}$ are coplanar, *i.e.*, their scalar triple product is zero:

$$\begin{aligned} &((R^{c\top}\boldsymbol{n}_i^c) \times (R^{c'\top}\boldsymbol{n}_i^{c'}))^\top (R^{c\top}\boldsymbol{t}^c - R^{c'\top}\boldsymbol{t}^{c'}) \\ &\quad = (m_i^{c,c'})^\top (R^{c\top}\boldsymbol{t}^c - R^{c'\top}\boldsymbol{t}^{c'}) = 0. \end{aligned} \tag{9}$$

With N corresponding points, we have

$$\begin{bmatrix} (m_1^{c,c'})^\top R^{c\top} & -(m_1^{c,c'})^\top R^{c'\top} \\ \vdots & \vdots \\ (m_N^{c,c'})^\top R^{c\top} & -(m_N^{c,c'})^\top R^{c'\top} \end{bmatrix} \begin{bmatrix} \boldsymbol{t}^c \\ \boldsymbol{t}^{c'} \end{bmatrix} = \boldsymbol{0}_{N\times 1}. \tag{10}$$

With N points and C cameras, Eqs. (8) and (10) form a set of linear equations with $3N + 3C$ unknowns $\begin{bmatrix} \boldsymbol{x}_1 & \ldots & \boldsymbol{x}_N & \boldsymbol{t}_1 & \ldots & \boldsymbol{t}_C \end{bmatrix}^\top$ of the form:

$$A \begin{bmatrix} \boldsymbol{x}_1 & \ldots & \boldsymbol{x}_N & \boldsymbol{t}_1 & \ldots & \boldsymbol{t}_C \end{bmatrix}^\top = \boldsymbol{0}, \tag{11}$$

where A is a sparse matrix with $[\boldsymbol{n}_i^c]_\times R^c$, $[\boldsymbol{n}_i^c]_\times$, $m_i^{c,c'} R^{c\top}$, and $-m_i^{c,c'} R^{c'\top}$ from Eqs. (8) and (10).

This system has 4 degrees of freedom, *i.e.*, 3 for global X/Y/Z-translations and 1 for global scaling. That is, the system has 4 right singular vectors corresponding to the 4 zero singular values, and we can find a set of coefficients to make the translation of the first camera be zero [35].

Both the rotation estimation and the translation estimation require 3 or more joints to be observed in each view. For noisy input with outliers, we can apply RANSAC to make the calibration procedure robust.

TABLE I

QUANTITATIVE EVALUATIONS WITH SYNTHETIC DATA FROM KIST SYNADL [36]. EACH ROW SHOWS THE RESULTS OF OUR LINEAR CALIBRATION ("L") AND THAT FOLLOWED BY THE PROPOSED BUNDLE ADJUSTMENT ("BA") ON DIFFERENT COMBINATIONS OF JOINTS. THE COLUMNS SHOW DIFFERENT SCENES FROM A1 (SMALLEST MOTION) TO A4 (LARGEST MOTION). THE RESULTS SHOW THAT OUR METHOD CAN HANDLE SCENARIOS RANGING FROM SMALL HUMAN MOVEMENTS TO LARGE HUMAN ACTIVITIES. PLEASE SEE FIG. 4 FOR VISUALIZATIONS.

	A1				A4			
	E_R	E_t	E_{3D}	E_{2D}	E_R	E_t	E_{3D}	E_{2D}
BBOX [11]	1.544	3.211	2.719	55.105	1.526	3.577	13.443	41.775
AO+MA [16], [34]	2.170	4.974	3.502	350.621	2.082	4.740	10.918	214.040
AO+MA+BA (λ_1=λ_2=0)	1.159	1.765	0.729	0.989	1.337	3.419	2.325	8.003
AO+MA+BA	1.243	2.000	0.803	2.172	1.766	4.704	3.904	42.299
SfM+PnP [1]	1.389	2.725	9.244	43.777	0.141	0.354	0.856	2.591
SfM+PnP+BA (λ_1=λ_2=0)	1.338	2.360	1.926	9.186	0.130	0.330	0.543	2.177
SfM+PnP+BA	1.312	3.050	2.244	17.613	0.116	0.251	0.159	6.541
J1 Ours (L)	0.519	0.924	1.800	61.190	0.156	0.380	0.903	13.765
J1 Ours (L+BA)	0.160	0.307	0.370	3.752	0.017	0.048	0.050	0.825
J2 Ours (L)	0.037	0.491	2.451	24.152	0.154	0.691	0.838	32.247
J2 Ours (L+BA)	**0.023**	0.052	**0.054**	0.559	**0.007**	**0.020**	**0.021**	0.339
J2 Ours (L+BA) (λ_1=λ_2=0)	**0.023**	**0.051**	0.057	**0.535**	**0.007**	**0.020**	**0.021**	**0.338**
J1 Ours (L) +J2 (BA)	0.024	0.055	0.063	0.550	**0.007**	**0.020**	**0.021**	0.339

C. Bundle Adjustment

We formulate reprojection error minimization as a maximum likelihood estimation problem. Assuming that 2D human joint detectors produce a probability distribution of each joint position that can be approximated by a normal distribution, we measure the goodness of the camera poses by evaluating it with the reprojected joints. That is, for each joint detected at $\boldsymbol{y}$, we minimize the negative log likelihood $-\log \mathcal{N}(\hat{\boldsymbol{y}}; \boldsymbol{y}, \sigma)$, where $\hat{\boldsymbol{y}}$ is the reprojected joint position and σ is the standard deviation of the distribution returned by the joint detector.

In addition to this reprojection error, we also minimize the alignment error of the oriented points modeled with their spherical variance and the variance of the bone length over the frames. In summary, our objective function to be minimized is

$$\begin{aligned}\mathcal{E}(\boldsymbol{\theta}) = &-\sum_{c=1}^{C}\sum_{i=1}^{N} \log \mathcal{N}(\hat{\boldsymbol{y}}_i^c(\boldsymbol{\theta}); \boldsymbol{y}_i^c, \sigma) \\ &+ \lambda_1 \sum_{i=1}^{N} \varsigma_1(\boldsymbol{\theta}; \{\boldsymbol{v}_i^1, \ldots, \boldsymbol{v}_i^C\}) + \lambda_2 \sum_{b\in\mathcal{B}} \varsigma_2(\boldsymbol{\theta}; \mathcal{B}_b)\,,\end{aligned} \tag{12}$$

where $\boldsymbol{\theta}$ denotes the parameters to be optimized, *i.e.*, K, R, $\boldsymbol{t}$, and $\boldsymbol{x}$. $\varsigma_1(\cdot)$ returns the spherical variance defined by

$$\varsigma_1(\boldsymbol{\theta}; \{\boldsymbol{v}_i^1, \ldots, \boldsymbol{v}_i^C\}) = 1 - \frac{1}{C}\left\| \sum_{c=1}^{C} R^{c^{-1}} \boldsymbol{v}_i^c \right\|. \tag{13}$$

$\mathcal{B}$ denotes the set of bones, and $\mathcal{B}_b$ denotes the set of 3D joint pairs corresponding to the endpoints of a bone b in different frames. $\varsigma_2(\cdot)$ returns the variance of the bone b length, *i.e.*, the variance of the distances between the 3D point pairs in $\mathcal{B}_b$. λ_1 and λ_2 control the weights of the three components and are determined experimentally. In general, if the 2D joint estimates are more reliable than the 3D pose estimates, we can make its weight smaller, and vice versa.

IV. SELF-SUPERVISED FINE-TUNING OF 3D POSE ESTIMATOR

As shown in Fig. 2, we can triangulate the 3D human pose for every frame once the cameras are calibrated with our method, and then use them as pseudo ground-truth to fine-tune the single-view 3D human pose estimator $f_{\boldsymbol{\theta}} : \mathbb{R}^{N\times J\times 2} \mapsto \mathbb{R}^{N\times J\times 3}$, where J is the number of joints. By alternating between calibrating the camera poses and fine-tuning the 3D human pose estimator with the updated 3D poses, we can simultaneously improve the camera calibration and the 3D pose estimation accuracy. Note that the proposed linear calibration only depends on the 3D rotations of the joints, not their 3D positions. As a result, $f_{\boldsymbol{\theta}}$ needs to only estimate normalized 3D poses and does not need to estimate the absolute scale.

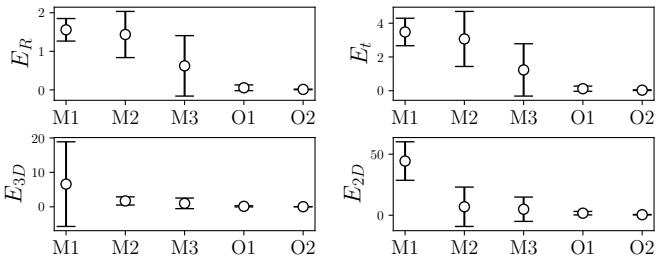

Fig. 3. Calibration errors for the various activities in KIST SynADL [36]. M1, M2, M3, O1, and O2 denote BBOX, AO+MA+BA, SfM+PnP+BA, J1 Ours (L+BA), and J2 Ours (L+BA), respectively. Each error bar represents the total mean and standard deviation of all A1 to A4 scenes. Our algorithm achieves high accuracy for a variety of scenes even when using a small number of joints.

V. EXPERIMENTS

We conducted extensive experiments on synthetic and real-world scenes. The former enables us to quantitatively validate the performance of extrinsic calibration. The latter demonstrates successful use of our method in various daily scenarios. We also show that our method can be used for self-supervised learning of 3D pose estimation. The experiments were approved by the Research Ethics Committee of the Graduate School of Informatics, Kyoto University (KUIS-EAR-2020-002).

A. Experimental Details

In order to measure the errors of the estimated poses from the ground truth, we evaluate the rotation with Riemannian distance [39] E_R and the translation with RMSE (root mean squared error) E_t. We also evaluate the reprojection error E_{2D} in pixels and the RMSE of the reconstructed 3D joint positions E_{3D}. In what follows, the units for E_t and E_{3D} are meters. We use median for E_{2D} and mean for the rest.

We use VideoPose3D [27] as the 3D human pose estimator $f_{\boldsymbol{\theta}}$ and fine-tune it in a self-supervised manner. It takes a 2D keypoint sequence as input and outputs 3D human pose in the camera coordinate system. We used Detectron2 [17] for 2D keypoint detection.

We compare our method with three algorithms BBOX [11], SfM+PnP [1] and AO+MA [16]. BBOX [11] uses the center of human bounding boxes as 2D corresponding points across views. It estimates pairwise poses with the 5-point algorithm [40], and integrates them with motion averaging [41] followed by non-linear optimization that minimizes the reprojection errors [5]. SfM+PnP uses 2D joints. It first recovers

TABLE II

QUANTITATIVE EVALUATIONS WITH REAL IMAGES. THE RESULTS CLEARLY SHOW THAT OUR METHOD OUTPERFORMS EXISTING METHODS. PLEASE SEE FIG. 4 FOR VISUALIZATIONS.

	Our dataset				Panoptic [37]				Human 3.6M [38]			
	E_R	E_t	E_{3D}	E_{2D}	E_R	E_t	E_{3D}	E_{2D}	E_R	E_t	E_{3D}	E_{2D}
BBOX [11]	2.050	7.609	21.952	141.454	1.690	4.592	11.212	406.614	1.717	6.502	14.959	96.319
AO+MA [16], [34]	1.877	7.337	16.265	747.461	1.443	5.418	2.803	388.665	1.481	5.428	16.643	455.138
AO+MA+BA	1.544	5.725	6.338	36.853	2.083	1.312	12.542	4.190	2.037	7.574	24.203	18.319
SfM+PnP [1]	0.364	1.424	1.578	194.997	1.168	4.677	3.161	96.818	0.007	0.035	0.024	2.455
SfM+PnP+BA	0.061	0.162	0.139	6.744	1.580	4.296	3.038	8.381	0.003	**0.013**	0.013	2.048
J1 Ours (L)	0.182	1.307	2.009	85.136	0.038	1.520	0.704	30.557	0.097	0.357	0.436	9.123
J1 Ours (L+BA)	0.016	0.073	0.058	3.631	0.033	**0.073**	0.126	2.088	0.006	0.032	0.024	2.796
J1 Ours (L w/ obs. mask)	0.181	1.318	2.010	85.210	0.038	1.520	0.704	30.557	0.144	0.591	0.655	7.410
J1 Ours (L+BA w/ obs. mask)	0.016	0.074	0.058	3.646	0.033	**0.073**	0.126	2.088	0.005	0.043	0.027	3.013
J1 Ours (RANSAC)	0.059	1.592	3.616	47.700	0.132	1.732	1.526	20.226	0.097	0.371	0.421	8.267
J1 Ours (RANSAC+BA)	**0.015**	0.066	0.057	3.653	0.033	0.074	0.132	2.017	0.006	0.032	0.024	2.796
J2 Ours (L)	0.052	2.154	3.108	101.275	0.535	1.539	1.770	101.690	0.057	0.146	0.302	4.850
J2 Ours (L+BA)	0.040	0.125	0.096	3.731	0.032	0.119	0.133	1.576	0.003	0.017	**0.013**	1.788
J2 Ours (L w/ obs. mask)	0.043	1.414	1.817	55.083	0.536	1.496	1.754	98.709	0.085	0.154	0.352	4.579
J2 Ours (L+BA w/ obs. mask)	0.020	**0.053**	**0.041**	**3.014**	0.032	0.118	0.133	1.569	0.003	0.016	0.013	**1.777**
J2 Ours (RANSAC)	0.243	1.426	1.731	51.672	0.212	1.004	2.008	12.316	0.057	0.184	0.232	4.599
J2 Ours (RANSAC+sBA)	0.040	0.125	0.096	3.728	**0.019**	0.083	**0.092**	**0.724**	**0.003**	0.017	**0.013**	1.788
J1+J2 Ours (L+BA)	0.040	0.124	0.096	3.731	0.030	0.107	0.115	1.612	**0.003**	0.017	**0.013**	1.788

TABLE III

QUANTITATIVE EVALUATIONS WITH OUR DATASET AFTER FINE-TUNING. E_O, E_H, AND E_P DENOTE THE 3D POSE ESTIMATION ERRORS W.R.T. THE GROUND TRUTH FOR OUR, HUMAN3.6M, AND PANOPTIC DATASET, RESPECTIVELY. THE RESULTS SHOW THAT THE REFINED 3D POSES LEAD TO MORE ACCURATE CAMERA POSES WITHOUT OVERFITTING TO OUR DATASET. NOTE THAT THE RESULTS OF "INITIAL CALIBRATION" ARE IDENTICAL TO THOSE IN TABLE II.

	Initial calibration $E_O = 0.052, E_H = 0.025, E_P = 0.104$				1st fine-turning $E_O = 0.036, E_H = 0.062, E_P = 0.096$				2nd fine-turning $E_O = 0.036, E_H = 0.063, E_P = 0.078$			
	E_R	E_t	E_{3D}	E_{2D}	E_R	E_t	E_{3D}	E_{2D}	E_R	E_t	E_{3D}	E_{2D}
J1 Ours (L)	0.182	1.307	2.009	85.136	0.029	0.077	0.107	15.805	0.031	0.090	0.140	20.764
J1 Ours (L+BA)	0.016	0.073	0.058	3.631	0.016	0.069	0.056	3.845	0.016	0.069	0.056	3.846
J1 Ours (L w/ obs. mask)	0.181	1.318	2.010	85.210	0.041	0.161	0.375	30.774	0.052	0.209	0.599	42.927
J1 Ours (L+BA w/ obs. mask)	**0.016**	0.074	0.058	3.646	0.011	0.050	0.036	3.742	0.012	0.053	0.042	3.737
J2 Ours (L)	0.052	2.154	3.108	101.275	**0.009**	0.434	0.811	73.014	**0.010**	0.384	0.663	70.968
J2 Ours (L+BA)	0.040	0.125	0.096	3.731	0.041	0.131	0.102	3.788	0.041	0.131	0.102	3.782
J2 Ours (L w/ obs. mask)	0.043	1.414	1.817	55.083	0.079	0.196	0.419	43.329	0.033	0.261	0.671	54.112
J2 Ours (L+BA w/ obs. mask)	0.020	**0.053**	**0.041**	**3.014**	0.014	**0.038**	**0.030**	**2.579**	0.013	**0.035**	**0.028**	**2.512**

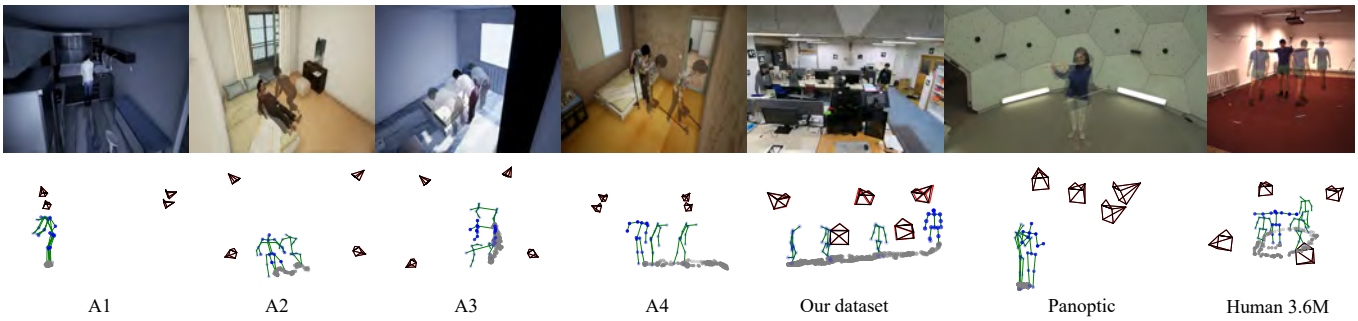

Fig. 4. Visualizations of calibration results for the real and synthetic datasets. Ground truth cameras and our estimated cameras are shown in red and black, respectively. Gray dots depict footprints of each subject. We used J2 Ours (L+BA w/ obs. mask) and J2 Ours (L+BA) for our dataset and for the others, respectively. The results show that our method successfully estimates camera poses in the scene with a wide range of activities of daily living, from small to large movements. The ground truth cameras in red overlap the estimated ones in black as shown quantitatively in Tables I, II, and III.

the relative pose of a camera pair, triangulates the 3D point, then solves PnP [42] for the rest of the cameras with non-linear minimization of the reprojection error. AO+MA uses 3D joints. It estimates a sequence of 3D human poses for each camera, recovers pairwise relative poses of all the camera pairs with absolute orientation [43], and integrates them into a single coordinate system with motion averaging. For SfM+PnP and AO+MA, we also report the results with our bundle adjustment using the 3D human structure (Eq. (12)) as SfM+PnP+BA and AO+MA+BA. Note that we used the mid-hip instead of the center of the bounding box for evaluating BBOX to ensure better accuracy. BBOX reimplements the calibration using person reidentification by Yan *et al*. [11], SfM+PnP reimplements the linear calibration part of Takahashi *et al*. [1], and AO+MA reimplements that of Rhodin *et al*. [34] and Garau *et al*. [16], since their original implementations are not publicly available.

B. Dataset

We quantitatively evaluate our method using a publicly available synthetic dataset KIST SynADL which is based on 3D motions of elderlies [36]. The dataset provides rendered image sequences with 3D and 2D joint annotations of a single person[1]. The movements of people are based on motion

[1]Note that KIST SynADL dataset does not provide the camera parameters. We calibrated the cameras by ourselves from the provided 2D and 3D joints, and confirmed that the reprojection errors with the calibrated parameters are about 1e-6 px, *i.e*., effectively zero, and can be used as the ground truth.

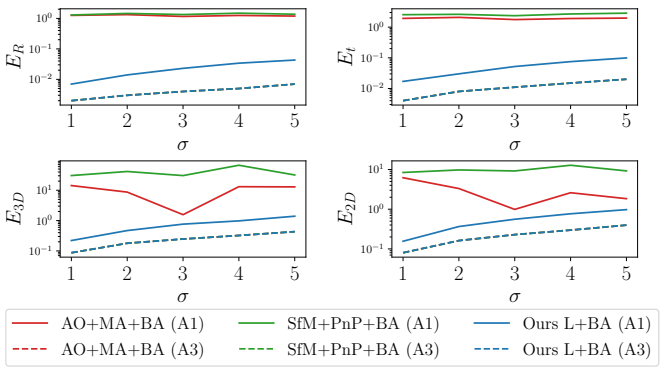

Fig. 5. Calibration errors for different noise levels and human activities. σ denotes the standard deviation of Gaussian noise injected to all available 2D joint positions (J2) in pixels. Our method achieves robust accuracy for all noise levels regardless of the person's activities and whether they have small (A1) or large motion (A3). Notice that the green and red dashed lines are almost hidden behind the blue dashed line.

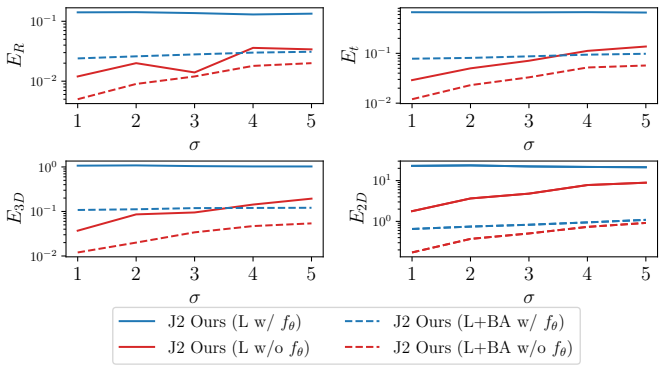

Fig. 6. Calibration accuracy degradation due to the 3D pose estimator f_θ. Each plot shows the results of J2 Ours (L) and (L+BA) for the synthetic dataset (A2) [36]. σ denotes the standard deviation of the joint position errors in pixels. "w/o f_θ" directly uses noise-injected 3D ground truth poses and "w/ f_θ" estimates the 3D pose with f_θ from noisy 2D joint positions. These plots show that the use of f_θ can degrade the accuracy in practice, but our method is not affected by those errors as shown in Table I.

capture data, and we selected four activities of daily living for evaluation.

For real-world data, we use *S11 Walking 1* sequence from Human3.6M [38] and *flute* sequence from Panoptic datasets [37]. *flute* is a challenging sequence since the player exhibits small motion and trajectory spanning a small area in the scene. In addition, we recorded a daily scene of a single person walking in a room for 13 seconds with five surveillance cameras. We subsampled 20 frames out of the 13 seconds as input images. This is a novel scene for the 3D pose estimator f_θ [27], so that we can evaluate our self-supervised refinement of 3D pose estimation. The intrinsic parameters are calibrated with Zhang's method [44] and the extrinsic parameters are calibrated using correspondences given by AR markers and also manually-annotated feature points. The mean reprojection error was 1.568 pixels and we used these parameters as the pseudo ground truth in the evaluation.

C. Extrinsic Camera Calibration

a) Synthetic dataset: This section evaluates the robustness of our method against noise and different combinations of joints. Table I shows calibration errors of our method, Ours (L) and Ours (L+BA), for different combinations of joints (J1 and J2) and scenes (A1 to A4). AO+MA and SfM+PnP use all available joints (J2) in the scenes. Ours (L+BA) is the proposed calibration in Section III, and Ours (Linear) is that without the bundle adjustment (Section III-C). We added Gaussian noise of $\sigma = 3$ pixels to 2D joints defined in 640×360 image resolution, and used them to estimate 3D joints from [27]. We ran 15 trials for each scene. J1 uses shoulders, thorax, and pelvis as oriented points with bones joining them defining their unique orientations. J2 adds elbows, hip, knees, wrists, and ankles to J1. The target person moves in approximately 0.5m×0.5m, 1.0m×1.0m, 1.5m×1.5m, and 2.0m×2.0m areas in A1 (*cutting vegetable on the cutting board*), A2 (*lying down*), A3 (*spreading bedding/folding bedding*), and A4 (*vacuumming the floor*) scenarios, respectively. We uniformly subsampled the sequence into 25 frames for A4. The processing time was approximately 0.5s and 20s for L and BA, respectively, on a PC with Intel Core i7-9700X.

As shown in Fig. 3, our method using 2D and 3D joints outperforms other methods even when using only a relatively small number of joints. In addition, for activities with small motions (A1 and A2), SfM+PnP suffers from larger errors of rotation, but our method achieves consistent accuracy in all scenarios. BA further improves the accuracy of our method.

Figs. 5 and 6 illustrate the calibration errors when using the full body with different noise levels computed as the average error of 15 trials for each noise level. Fig. 5 shows, in comparison with other methods, that our method is robust to different activities, including those consisting of small and large motions, and also severe noise that can often occur in the wild. Fig. 6 shows the performance drop caused by the 3D pose estimator f_θ for different scenes. These plots show that the accuracy of f_θ can decrease, but our method is robust to such errors as quantitatively shown in Table I.

As shown in Fig. 4, our method can calibrate the cameras reliably even from a small daily motion sequence, in contrast to conventional calibration methods that rely on a calibration target being moved across the scene, which is often challenging to realize. These results demonstrate the effectiveness of our method, especially as a convenient calibration method for elderly support applications.

b) Real-world dataset: We evaluate our method with the *walking 1* sequence of Human3.6M and the *flute* sequence of Panoptic datasets. In the *walking 1* sequence, a subject walks around in a room, and his joints move around in the image. In contrast, in the *flute* sequence, a subject playing the flute stands in one position and her joints move in a small area in the image; it is a challenging scene for our method. We also evaluate our method in a daily environment captured by five GoPro 7 which are not strictly synchronized at the frame-level. That is, the cameras share a clock up to a few frames ambiguity. Evaluations with our dataset demonstrates robustness to the time difference in practice. Notice that the 3D pose estimator is not pretrained for our dataset. In these evaluations with real images, we also report the performance of our method combined with RANSAC as Ours (RANSAC) and Ours (RANSAC+BA). In addition to

this, we employ observation masks for our linear calibration to cope with the occlusion of body parts caused by the scene. The observation masks are constructed after BA by filtering joints with reprojection errors greater than a threshold.

Table II shows calibration errors, and Fig. 4 visualizes calibrated camera poses. We can observe that our method performs robustly even for sequences with smaller motions in comparison with others. These results show that the use of both 2D and 3D joints helps improve robustness to such challenging cases. The results on our dataset demonstrate that our method generalizes well to environments beyond those used to train the 3D pose estimator.

c) Joint optimization of the intrinsic parameters: As a standard practice in camera calibration, the intrinsic parameters and the extrinsic parameters can be jointly optimized. For this, we conducted additional evaluations on our dataset (J2 Ours) with two different intrinsic parameters: the pseudo ground truth K^p calibrated with a chessboard [44] and a parameter calculated from the camera specifications K^s. The values of K^s vary by 1.6 percent on average from those of K^p. Our linear calibration with K^p and K^s yields $E_{\text{2D}} = 101.27$ and 102.74px, respectively, and bundle adjustment on both the intrinsic and extrinsic parameters results in $E_{\text{2D}} = 6.84$ and 7.32px, respectively. In contrast, bundle adjustment only on the extrinsic parameters results in $E_{\text{2D}} = 3.73$ and 4.66px, respectively. These results show that our 3D pose estimator $f_{\boldsymbol{\theta}}$ is not accurate enough for joint intrinsic and extrinsic parameter estimation, and fixing the intrinsic parameters, even if they are approximate, performs better.

D. Self-Supervised Training

Once we obtain the camera parameters and the 3D joints, we can fine-tune the 3D pose estimator $f_{\boldsymbol{\theta}}$ by using the triangulated 3D joint locations. In this fine-tuning, we jointly used Human3.6M to prevent the 3D pose estimator from overfitting to the target environment. Note that *walking 1* sequence was not included in any set of the training and test for fine-turning.

Table. III shows the calibration results using enhanced 3D joints obtained from the model after fine-turning. The observation masks help prevent training on incorrect 3D joints. As a result, the iterative fine-tuning can jointly improve the calibration and the 3D pose estimation, while maintaining the accuracy of the 3D pose estimator. These results demonstrate that the fine-tuning can let us collect pseudo ground truth 3D motion of people which are otherwise hard to directly measure, *e.g.*, elderlies.

VI. Conclusions

We showed that by treating a person as a moving 3D oriented calibration target we can achieve robust and accurate extrinsic camera calibration of a multi-view vision system. Our method fully leverages the articulated human body as a set of moving oriented points which can be robustly estimated from a video with a deep 3D human pose estimator. We introduced a factorization approach for estimating the camera poses from corresponding oriented points and a linear system for the camera positions. The linear solution is then non-linearly optimized by leveraging the articulated human body as a calibration target. Bundle adjustment minimizes the variances of 3D joint orientations and 3D bone lengths in addition to the standard 2D reprojection errors.

We also demonstrated that the deep 3D pose estimator can be further fine-tuned to the installed environment in a self-supervised manner to improve both the calibration accuracy and the 3D pose estimation. Our initial calibration with a pre-trained 3D pose estimator enables triangulation of the calibration target's 3D skeleton which in turn serves as a pseudo ground truth for fine-tuning the 3D pose estimator. The fine-tuned estimator provides improved 3D poses for our calibration and improves the calibration accuracy.

A limitation of our method is the scale ambiguity as described in Section III-B. This ambiguity, however, can be resolved by simply observing a single object of known size, *e.g.*, the person in the scene. That is, given the height of the person, we can determine the scaling factor as our method obtains the 3D skeleton of the person as well as the camera calibration parameters.

We believe that our camera calibration method can serve as a practical tool that can be widely used in real-world situations and will catalyze the deployment of multi-view vision systems for societally critical applications including autonomous robots and health care monitoring systems.

Acknowledgement

This work was in part supported by JSPS 20H05951, 21H04893, 22H05654, JST JPMJCR20G7, JPMJPR1858, and RIKEN GRP.

APPENDIX

Proof of Proposition 3.1

The SVD of the observation V^1 by a single camera $c = 1$ in Eq. (4) returns Y^1, D^1, and $Z^{1\top}$ as an $N \times 3$, 3×3, and 3×3 matrices, respectively as $V^1 = Y^1 D^1 Z^{1\top}$. Here $Z^{1\top}$ is an orthonormal matrix by definition, and hence assigning $V = Y^1 D^1$ and $R^{1\top} = Z^{1\top}$ guarantees that R^1 is a valid rotation matrix, and also that V is row-normalized because it corresponds to $V^1 Z^1$, a rotation of the originally row-normalized matrix V^1.

Consider the SVD of the observation V^c by another camera $c (c \neq 1)$. By denoting the relative rotation between the camera 1 and c by Q_c, we obtain $V^c = V^1 Q_c = Y^1 D^1 (Z^{1\top} Q_c)$. That is, because SVD is rotation invariant in the sense that right-multiplying a rotation matrix only rotates the row space while keeping the column space and the singular values unchanged, the SVD of V^c returns the singular values and the left singular vectors identical to those of V^1, and only the right singular

vectors are rotated by Q_c. This suggests that we can rewrite $V^{1:C}$ as

$$\begin{aligned} V^{1:C} &= \left[V^1 \cdots V^c \cdots V^C\right] \\ &= \left[Y^1 D^1 Z^{1\top} \cdots Y^c D^c Z^{c\top} \cdots Y^C D^C Z^{C\top}\right] \\ &= \left[Y^1 D^1 Z^{1\top} \cdots Y^1 D^1 Z^{1\top} Q_c \cdots Y^1 D^1 Z^{1\top} Q_C\right] \\ &= Y^1 D^1 Z^{1\top} \left[I \cdots Q_c \cdots Q_C\right] = Y^1 D^1 Z^{1\top} Q\,, \end{aligned} \tag{14}$$

where I is an identity matrix and $Q = \begin{bmatrix} I & \cdots & Q_c & \cdots & Q_C \end{bmatrix}$. Y^1 is an $n \times 3$ column-orthonormal matrix, and D^1 is a 3×3 diagonal matrix. $Z^{1\top} Q$ is a $3 \times 3C$ row-orthogonal matrix and

$$Z^{1\top} Q (Z^{1\top} Q)^\top = Z^{1\top} Q Q^\top Z^1 = C Z^{1\top} Z^1 = C I_{3\times 3}\,. \tag{15}$$

As a result, scaling $Z^{1\top} Q$ by $C^{\frac{1}{2}}$ makes it row-orthonormal, and hence the following decomposition of $V^{1:C}$ becomes identical to its SVD:

$$V^{1:C} = Y^1 \left(C^{\frac{1}{2}} D^1\right)\left(C^{\frac{-1}{2}} Z^{1\top} Q\right), \tag{16}$$

and factorizing the SVD of $V^{1:C} = YDZ^\top$ as Eq. (7) yields

$$\begin{aligned} V &= YDM^{-1} = C^{\frac{1}{2}} Y^1 D^1 M^{-1}, \\ R^{1:C} &= MZ^\top = C^{\frac{-1}{2}} M Z^{1\top} Q\,. \end{aligned} \tag{17}$$

Therefore, letting $M = C^{\frac{1}{2}} I$ or its rotation, *i.e.*, a rotation scaled by $C^{\frac{1}{2}}$, makes V become a row-normalized matrix, and also makes $R^{1:C}$ become a horizontal stack of rotation matrices each of which satisfies orthonormality.

References

[1] K. Takahashi, D. Mikami, M. Isogawa, and H. Kimata, "Human pose as calibration pattern; 3d human pose estimation with multiple unsynchronized and uncalibrated cameras," in *Proc. CVPRW*, 2018.

[2] J. Puwein, L. Ballan, R. Ziegler, and M. Pollefeys, "Joint camera pose estimation and 3d human pose estimation in a multi-camera setup," in *ACCV*, 2014.

[3] S. H. Olivier Moliner and K. Astrom, "Better prior knowledge improves human-pose-based extrinsic camera calibration," in *ICPR*, 2020.

[4] G. Nakano, "Camera calibration using parallel line segments," in *ICPR*, 2020.

[5] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[6] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building rome in a day," in *ICCV*, 2009.

[7] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel, "Markerless motion capture with unsynchronized moving cameras," in *CVPR*, 2009.

[8] A. Taneja, L. Ballan, and M. Pollefeys, "Modeling dynamic scenes recorded with freely moving cameras," in *ACCV*, 2011.

[9] A. Mustafa, H. Kim, J.-Y. Guillemaut, and A. Hilton, "General dynamic scene reconstruction from multiple view video," in *ICCV*, 2015.

[10] Y. Furukawa and J. Ponce, "Accurate camera calibration from multi-view stereo and bundle adjustment," in *CVPR*, 2008.

[11] Y. Xu, Y.-J. Li, X. Weng, and K. Kitani, "Wide-baseline multi-camera calibration using person re-identification," in *CVPR*, 2021.

[12] A. Elhayek, C. Stoll, K. I. Kim, and C. Theobalt, "Outdoor human motion capture by simultaneous optimization of pose and camera parameters," *CGF*, vol. 34, no. 6, pp. 86–98, 2015.

[13] F. Lv, T. Zhao, and R. Nevatia, "Camera calibration from video of a walking human." *TPAMI*, vol. 28, pp. 1513–1518, 2006.

[14] X. Zhang, Y. Zhang, X. Zhang, T. Yang, X. Tong, and H. Zhang, "A convenient multi-camera self-calibration method based on human body motion analysis," in *Int. Conf. on Image and Graphics*, 2009.

[15] P. A. Tresadern and I. D. Reid, "Camera calibration from human motion." *Image Vis. Comput.*, vol. 26, no. 6, pp. 851–862, 2008.

[16] N. Garau, F. G. B. D. Natale, and N. Conci, "Fast automatic camera network calibration through human mesh recovery," *Journal of Real-Time Image Processing*, 2020.

[17] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[18] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.

[19] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *TPAMI*, vol. 43, no. 1, pp. 172–186, 2019.

[20] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *ICCV*, 2017.

[21] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose Flow: Efficient online pose tracking," in *BMVC*, 2018.

[22] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," in *CVPR*, 2019.

[23] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," *TOG*, vol. 36, no. 4, pp. 1–14, 2017.

[24] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *ECCV*, 2016.

[25] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *CVPR*, 2018.

[26] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *TOG*, vol. 34, no. 6, pp. 1–16, 2015.

[27] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *CVPR*, 2019.

[28] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3d human dynamics from video," in *CVPR*, 2019.

[29] Z. Luo, S. A. Golestaneh, and K. M. Kitani, "3d human motion estimation via motion compression and refinement," in *ACCV*, 2020.

[30] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *ICCV*, 2017.

[31] G. Varol, I. Laptev, C. Schmid, and A. Zisserman, "Synthetic humans for action recognition from unseen viewpoints," *IJCV*, 2021.

[32] M. Kocabas, S. Karagoz, and E. Akbas, "Self-supervised learning of 3d human pose using multi-view geometry," in *CVPR*, 2019.

[33] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *CVPR*, 2017.

[34] H. Rhodin, J. Spörri, I. Katircioglu, V. Constantin, F. Meyer, E. Müller, M. Salzmann, and P. Fua, "Learning monocular 3d human pose estimation from multi-view images," in *CVPR*, 2018.

[35] N. Jiang, Z. Cui, and P. Tan, "A global linear method for camera pose registration," in *ICCV*, 2013.

[36] H. Hwang, C. Jang, G. Park, J. Cho, and I.-J. Kim, "ElderSim: A synthetic data generation platform for human action recognition in eldercare applications," *IEEE Access*, p. 1–1, 2021.

[37] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. S. Godisart, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social interaction capture," *TPAMI*, 2017.

[38] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3d human sensing in natural environments," *TPAMI*, vol. 36, no. 7, pp. 1325–1339, 2014.

[39] M. Moakher, "Means and averaging in the group of rotations," *SIAM J. on Matrix Analysis and Applications*, vol. 24, no. 1, p. 1–16, 2002.

[40] D. Nistér, "An efficient solution to the five-point relative pose problem," *TPAMI*, vol. 26, no. 6, pp. 756–770, 2004.

[41] V. M. Govindu, "Combining two-view constraints for motion estimation," in *CVPR*, 2001.

[42] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *IJCV*, vol. 81, no. 2, p. 155, 2009.

[43] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA-A*, vol. 4, no. 4, pp. 629–642, 1987.

[44] Z. Zhang, "A flexible new technique for camera calibration," *TPAMI*, vol. 22, no. 11, pp. 1330–1334, 2000.