CasCalib: Cascaded Calibration for Motion Capture from Sparse Unsynchronized Cameras

by

James Tang

B.Sc The University of British Columbia 2020

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University of British Columbia
(Vancouver)

November 2023

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

CasCalib: Cascaded Calibration for Motion Capture from Sparse Unsynchronized Cameras

submitted by **James Tang** in partial fulfillment of the requirements for the degree of **Master of Science** in **Computer Science**.

Examining Committee:

Helge Rhodin, Assistant Professor, Department of Computer Science, UBC
Supervisor

Bastian Wandt, Assistant Professor, Department of Electrical Engineering, LIU
Co-Supervisor

Kwang Moo Yi, Assistant Professor, Department of Computer Science, UBC
Supervisory Committee Member

Abstract

It is now possible to estimate 3D human pose from monocular images with off-the-shelf 3D pose estimators. However, many practical applications require fine-grained absolute pose information for which multi-view cues and camera calibration are necessary. Such multi-view recordings are laborious because they require manual calibration, and are expensive when using dedicated hardware. Our goal is full automation, which includes temporal synchronization, as well as intrinsic and extrinsic camera calibration. This is done by using persons in the scene as the calibration objects. We attain this generality by partitioning the high-dimensional time and calibration space into a cascade of subspaces, and introduce tailored algorithms to optimize each efficiently and robustly. The outcome is an easy-to-use, flexible, and robust motion capture toolbox that we release to enable scientific applications, which we demonstrate on diverse multi-view benchmarks.

Lay Summary

Camera calibration is the problem of finding out the camera setup. This entails the camera's location, the camera's orientation, and the field of view of the camera. This is important because it gives us information on how the points in the image are related to points in the real world. Usually, this multi-camera calibration is done manually using a known reference object in the scene. However, this requires a dedicated preprocessing step. Since humans are present in many scenes, we seek to determine these parameters using humans as reference objects. Synchronizing multiple cameras such that the frames between different views match is another problem that requires complicated hardware or manual labor. We seek to use the motion of humans that all cameras can see to simultaneously match frames between camera views and to calibrate the cameras.

Preface

The written contents of this thesis are an original work by James Tang under the supervision of Bastian Wandt and Helge Rhodin. All code, figures, and experiments used in this thesis were originally created by James Tang except if stated otherwise. The original camera calibration algorithm for single-view cameras based on human height was originally developed by Shashwat Suri in a directed study and expanded further upon in this thesis. The chapter on statistical analysis was originally done as a course project for Stat 547C in the spring semester of 2023. Hardware resources were provided by the UBC Department of Computer Science.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BFM Brute-Force Matchers

DLT Direct Linear Transform

HRnet High-Resolution Net

ICP Iterative Closest Point

RANSAC Random sample consensus

SIFT Scale-Invariant Feature Transform

Dedication

Figure 1: My Family's chickens: Legume, Ina, and Bones

Chapter 1

Introduction

Computer vision based 3D reconstruction has now reached the mainstream, enabling detailed 3D reconstructions from handheld video recordings with ubiquitous mobile phones [49]. However, when aiming for the reconstruction of dynamic human performances, it still demands multiple cameras as single-view methods suffer from occlusions and depth ambiguities [10]. Additionally, the cameras require calibration [20], which is the process of estimating the camera's intrinsic parameters, which include focal length, focal center, and the camera's extrinsic parameters, which include camera orientation and position. Together these parameters form the camera matrix, which gives us a relationship between 3D objects in the scene, and their 2D representation in an image. Traditionally, this is determined manually before filming is done. Hence, applications in visual effects [38] and medical studies, such as those in neuroscience studying motion deficits [59], still rely on dedicated motion capture studios or fairly technical camera setups. However, their manual calibration is cumbersome and error-prone, and their high cost renders them entirely inaccessible to smaller companies and labs with a non-technical background.

Another problem that occurs with multiple cameras is synchronization [37], which is the problem of having multiple cameras take frames at the same point in time. Traditionally, this is done in the hardware, where one camera is known as the master camera, which triggers all the other cameras in the system. This however also requires some technical background to wire cameras before filming starts, which may be difficult for people without this knowledge using commodity cameras that do not provide a hardware link. We believe huge opportunities are missed in life and social sciences, where highly technical experiments are traditionally less common than in neuroscience and medicine. For example, 3D reconstruction could have a wide range of applications, from studying athlete's motion on a sports field to detecting falls and accidents in an elderly care home.

Many approaches towards automating 3D capture exist, but only for specialized settings. Structure-from-motion techniques require either a continuous video stream of a static [44] or slowly deforming object [4], or a dense array of cameras that have largely overlapping fields of view. These are popular for settings where there is only one slowly moving camera or dozens of static cameras, but fail when only a few views are available. We refer to the setting with a few static cameras as the sparse camera case and show that classical multi-view geometry approaches fail. For sparse camera setups, manual calibration with a checkerboard or other markers arranged in a predetermined two-dimensional pattern is the most common approach [8]. However, the calibration object needs to be carefully placed and re-positioned by trained operators to ensure that the entire capture volume is covered, and multiple cameras see the calibration object. Moreover, re-calibration is required when cameras move ever so slightly, and fabricating calibration objects at very small or big scales, for example, meter scale for sports events, is challenging.

To achieve fully automated calibration, a promising fully automatic direction is to use humans [15, 30, 47, 54, 58], which are usually present in the scene, as calibration objects. Fei et al. [15] calibrate the focal length and ground plane under the assumption that all people have roughly the same height and the ground is flat, but do not address synchronization and the multi-view case. Takahashi et al. [47] and Liu et al. [30] subsequently use 2D keypoint detections in individual views to calibrate the extrinsic parameters of two or more cameras using classical fundamental matrix estimation, which requires seven or more correspondences across views. However, to establish the correspondence, they assume that cameras are synchronized, and only a single person is in view and visible from all cameras. In turn, Zhang et al. [58] establish temporal synchronization but assume intrinsic and extrinsic calibrated cameras, and still use a single person. Xu et al. [54] rely on reidentification, but synchronization is required and appearance matching is chal-

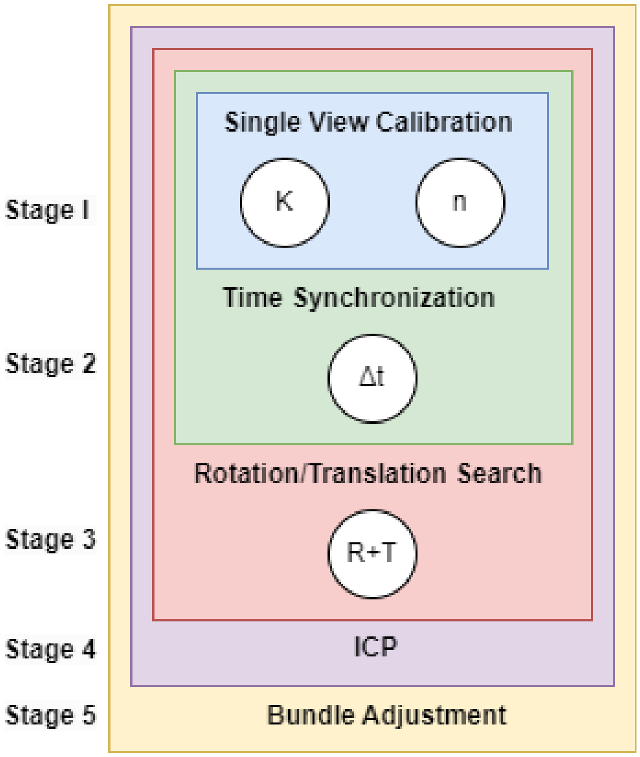


Figure 1.1: Cascaded Calibration Overview From top to bottom, we show how we break up the optimization problem into smaller subproblems by solving for a subset of the parameters at a time, with subsequent steps refining the earlier ones. The first step is the Single View Calibration step where we estimate the normal vector $\mathbf{n}$ and the intrinsics $\mathbf{K}$. Then, we estimate the time synchronization offset Δt. Finally, with the last three steps, we estimate and refine the rotation matrix $\mathbf{R}$ and the translation $\mathbf{T}$.

lenging when humans look alike, such as when a sports team dresses in the same uniform. To the best of our knowledge, there is no solution for unsynchronized sparse camera setups with a variable number of persons that are only partially visible.

We propose a cascaded calibration algorithm, shown in Figure 1.1, that breaks down the calibration of high-dimensional parameter space into subspaces that can be searched or optimized efficiently. For N cameras, we solve for $N(4\times 6+1)$

parameters which correspond to 4 intrinsics, 6 extrinsics, and 1 temporal shift parameters. The outcome is a sequential process with steps having the cascading dependencies visualized in Figure 1.1, with subsequent steps starting from preceding estimates that are further refined. In the first stage, camera focal length (intrinsic) and their orientation with respect to the ground (extrinsic) are estimated similarly to [15, 48], independently in each view so that we don't need temporal synchronization across cameras. In the second stage, we estimate the temporal offset by taking the scalar distances of the ankles from the center of the ground plane in order to reduce the dimensions to one. Then we align the sequence temporally by searching for the frame offset that results in the minimum distance between the sequences. In the third stage, we use these to reduce the 6D extrinsics problem to solving for 2D rotation and translation in the estimated ground planes, a 3D space optimized by least squares, and a greedy yet efficient search of the rotation on the ground plane. In the fourth stage, we refine the initialized extrinsic parameters alongside the additional temporal offset using ICP ([2] and [57]) in a 5D space. In the final stage, the entire 11D space is optimized using bundle adjustment [50].

This thesis is structured as follows: First, we review the existing literature on camera calibration and temporal synchronization. Second, we comprehensively explain our methodology, focusing on our cascaded pipeline. Third, we derive theoretical results from our single view step. Fourth, we conduct trials on synthetic data to gain insights into the impact of noise on our single view step. Fifth, we present Empirical results from various datasets. Sixth, we discuss various limitations within our pipeline. Lastly, we conclude this thesis with a summary and a discussion of potential future research directions as well as the societal impact of our work.

Chapter 2

Related Work

We categorize methods for multi-view calibration based on human poses by their reconstruction methodology, such as optimization or deep learning. We also consider whether they can handle a single person or multiple persons, the required prior knowledge, and the outputs, such as intrinsics, extrinsics, and temporal synchronization. Table 2.1 compares the capabilities and requirements of existing methods to ours.

Method	Framework	Finds Sync.	Finds Intrins.	Finds Extrins.	w/o GT Intrins.	w/o GT Extrins.	w/o GT Sync.	Multi-Person	Multi-View
Lee[26]	Deep	No	No	Yes	No	Yes	No	No	Yes
Zhang[56]	Deep	Yes	No	No	No	No	Yes	Yes	Yes
Zhang[55]	Deep	No	Yes	Yes	No	No	No	N/A	Yes
Grabner[18]	Deep	No	Yes	Yes	No	No	No	N/A	No
Ling[33]	Deep	Yes	No	No	No	No	No	N/A	No
Jarved[24]	Optim.	No	No	No	Yes	Yes	No	Yes	Yes
Xu[54]	Optim.	No	No	Yes	No	Yes	No	Yes	Yes
Troung[51]	Optim.	No	No	Yes	No	Yes	No	Yes	Yes
Liu[30]	Optim.	No	Yes	Yes	Yes	Yes	No	Yes	No
Fei[15]	Optim.	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Takahashi[47]	Optim.	Yes	No	Yes	No	Yes	Yes	Yes	Yes
Zhang[58]	Optim.	Yes	No	No	No	No	Yes	No	No
COLMAP[43]	Optim.	No	Yes	Yes	Yes	Yes	No	N/A	Yes
Ours	Optim.	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 2.1: Comparison of related methods. We summarize the differences, including which parameters they estimate and whether they require ground truth input. Only ours is able to calibrate and synchronize multi-person sequences.

2.1 Optimization Based Camera Calibration

These methods approach the calibration problem by obtaining keypoints in the scene from each camera, and then geometrically solving for the camera poses. We primarily focus on methods that use human poses as the keypoints, however, we start with more general methods to provide context.

Structure from motion (SfM) methods seek to estimate the 3D structure of a scene from a set of 2D images. There are many publicly available methods and software packages to perform this task. One of the most widely used examples is COLMAP [43]. COLMAP extracts features from the 2D images, matches the features, and then geometrically reconstructs the scene by triangulating between the views, using RANSAC [16] to remove outliers and bundle adjustment to refine the reconstruction. Although this method is easy to use and effective, it requires a large amount of images with a high degree of overlap. In particular, Brachmann et al [3] report that they need to store hundreds of thousands of feature vectors for matching as well as requiring several hours for the reconstruction. Additionally, the method is less effective in scenes that are featureless like a plain white background, or if they have repeating textures which can cause errors in the multi-view matching stage. Thus, we focus on using human poses as calibration objects since humans are fairly distinct objects in the scene and do not have repeating patterns of keypoints since we do the calibration and matching using the joint keypoints of the person rather than the appearance of the person, which could be unreliable if people are wearing the same clothes.

Fei et al. [15] performs intrinsic and extrinsic calibration on single view cameras using humans in the scene to solve direct linear transform (DLT) equations [46], using RANSAC [16] to remove outliers. This method is most similar to the first stage of our pipeline since it uses persons in the scene to estimate the focal length and ground plane position and orientation. However, this method only considers a single camera, which means they also don't need to consider temporal synchronization. Our method alleviates these limitations by subsequent steps that consider multiple cameras in the scene.

Liu et al. [30] performs intrinsic and extrinsic calibration on videos containing a single person. Because the videos contain a single person and are temporally synchronized, they do not need to find multi-view correspondences between the cameras since the person in one camera will always correspond to the person in the other camera. First, they obtain 2D keypoints using OpenPose [7], then they use multi-view correspondences of a single person to triangulate the 3D keypoints. Finally, they reproject the 3D keypoints back to 2D image coordinates and they optimize the reprojection error between the cameras. Similar to our last processing step, the intrinsic and extrinsic parameters are further optimized using a RANSAC loop [16], followed by bundle adjustment to optimize the camera poses. Takahashi et al. [47] calibrate extrinsic and intrinsic parameters in a similar way except they extend the method by using prior information about human poses such as bone length constraint and smooth motion constraints. However, neither Liu et al. [30] nor Takahashi et al. [47] solve the multi-person case, which requires obtaining multi-view correspondences. In addition to the requirement of multi-view correspondences, they also depend on temporal synchronization. By contrast, we also solve the multi-person case.

Many methods try to solve the multi-person case using person re-identification. Xu et al. [54] use matches bounding boxes between views. This method performs relatively well on datasets where people wear distinct clothing, as such on the Terrace and Basketball sequences [17], but performs comparatively worse on their ConstructSite dataset where everyone is wearing the same uniform. Another method of matching is to take reprojections between each camera and match the closest pose in 3D to each other. This is used in several methods [24, 51, 58] and has the advantage that they do not depend on visual features on the persons such as clothing since those could have high variation even on the same person, although combinatorically optimizing the matches can be expensive even with efficient algorithms like the Hungarian algorithm [25].

2.2 Deep Learning Based Camera Calibration

These approaches solve the calibration problem by fully regressing the camera parameters or predicting intermediate estimates through training on a labeled dataset. Lee et al. [26] use a 3D pose estimator and then use the predicted 3D poses as a 3D calibration object to optimize the camera poses such that 3D poses match and re-project to 2D pose estimates. This process alternates between optimizing the camera pose and intrinsics and optimizing the human pose. Zhang et al. [55] perform camera calibration on pan tilt zoom (PTZ) cameras that rotate and zoom but do not translate. Their application requires an online estimation. They take in as input two images from two views and use a three-part pipeline: a feature extractor based on a Siamese Network architecture [5], feature matching using correlation, and a regression network, to automatically estimate the intrinsic and extrinsic parameters, as well as the distortion parameters. Although this method can solve a more challenging problem, in the case where the cameras are not static, they require a large training set of over 100,000 image pairs. In addition, they do not handle the case where cameras are unsynchronized. Grabner et al. [18] approach the problem by extending the Faster/Mask R-CNN framework [21], by augmenting it with a focal length predictor and refining it using the reprojection error. However, despite the fact that their method gets 10 percent better focal lengths than the baseline, they use a very specific dataset of common objects as references such as chairs, sofas, and cars without occlusion. In general, these objects may not be present. Additionally, all deep learning methods introduce the need for a training set and a training procedure, which may not generalize to real-world scenes.

2.3 Synchronization

The previously described methods rely on the fact that the cameras are temporally synchronized with each other, which is an assumption that is often violated in practice. Besides methods such as hardware synchronization, which requires a pre-recording step to do, human pose can be used for temporal synchronization. In [58], Zhang et al. find the temporal offset by minimizing error in epipolar lines from reprojecting 2D pose detections. However, this method is in turn limited by the need for the cameras to be calibrated. In [13], Eichler et al. perform both camera synchronization and camera calibration using distances between joints from 3D pose detectors for synchronization, followed by a multi-view reconstruction for the extrinsic parameters. However, this method requires using a 3D pose detector, as well as assuming intrinsics are known. Zhang et al. [56] perform camera synchronization using a neural network architecture by warping one view to another, which does not assume a constant frame rate but intrinsic and extrinsic calibrations. Mei et al. [33] solves the problem using a two-stage weakly-supervised deep learning pipeline. The first stage tracks and estimates the trajectories of objects on a multi-view data set, and the second stage determines similarities between the two views and estimates an offset between them. Although this method can obtain finer precision, it is trained on a fairly narrow range of datasets that may not generalize to the real world and also requires synchronized data for training.

2.4 Summary

Although these methods have been shown to be effective, as shown in Table 2.1, they all require some sort of ground truth such as temporal synchronization, intrinsic, and extrinsic which they use to solve for the unknowns. We seek to use only 2D keypoint information without assuming calibration, synchronization, a training dataset, or person re-identification.

Chapter 3

Method

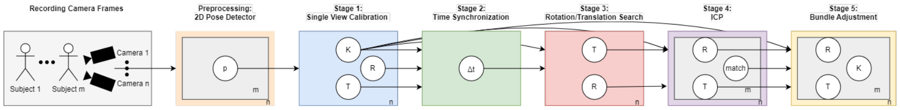


Figure 3.1: System Overview. A fine-grained view of the five stages in Figure 1.1, including how detections of single persons in single views are treated independently in Stage I and jointly subsequently. Variables are in the plate notation, with *n* the number of cameras and *m* the number of people in the scene.

To solve the camera calibration problem from multiple uncalibrated and unsynchronized views, we propose to break down the problem into several lower-dimensional problems. In a cascaded fashion, we start with a few variables that are solved globally and subsequently add details while reducing the range of the search space to stay practical.

We take in as input the 2D key point detections $\mathbf{p}^{\text{img}} \in \mathbb{R}^2$ of the major human joints, such as the head, neck, and ankles. Figure 3.1 shows how the variables are passed and refined between modules.

3.1 Single View Geometric Calibration

In the single view calibration case, our goal is to find the intrinsic camera parameters $\mathbf{K}$ of the projection transformation

$$\mathbf{p}^{\text{img}} = \mathbf{K}\mathbf{p}^{\text{cam}}, \text{ where } \mathbf{K} = \begin{pmatrix} f_1 a & s & o_1 \\ 0 & f_2 & o_2 \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.1}$$

mapping from 3D camera coordinates $\mathbf{p}^{\text{cam}}$ to 2D image coordinates $\mathbf{p}^{\text{img}}$. For simplicity, we assume that there is no skew or distortion so $a = 1$ and $s = 0$. We estimate $\mathbf{K}$, the ground plane position $\mathbf{g}$, and orientation $\mathbf{n}$ relative to the camera origin using a direct linear transform (DLT) [46]. To make the estimation feasible with only $\mathbf{p}^{\text{img}}$ as input, we assume that persons are standing up-right in some of the frames which makes them parallel to the ground plane normal vector and have a constant metric height h. Furthermore, following the cascaded, coarse-to-fine principle, we fix the principal point (o_1, o_2) to the image center and do not consider any relations across cameras as synchronization is missing. In the next section, we show the derivation of these DLT equations. Note that these strong assumptions are lifted in later refinement stages.

Direct Linear Transform

Using homogeneous coordinates, the ankle and shoulder positions of three or more people on a common ground plane are related by a linear system of equations that, when solved for its null space, reveal the sought-after camera parameters in closed form. The derivation of the direct linear transform equations is analogous to that in [15], However, we show our own derivation of it below.

We follow the well-established direct linear transform (DLT) [46] method to solve projective relations. We first write our constraints as a linear system of equations that are solved using Singular Value Decomposition (SVD) by finding the singular vector that corresponds to the smallest singular value, up to the unknown scale factor arising from the projection. To reach the form $\mathbf{M}\mathbf{x} = 0$, we take the cross product of Eq. 3.1 and

$$\mathbf{p}^{\text{cam}}_{\text{shoulder}} = \mathbf{p}^{\text{cam}}_{\text{ankle}} + h\mathbf{n} \tag{3.2}$$

and we take the cross product of Eq. 3.1 and $\mathbf{p}_{\text{ankle}}$. Then we subtract the two cross products to derive

$$\mathbf{p}^{\text{cam}}_{\text{shoulder}} \times \mathbf{K}(\mathbf{p}^{\text{cam}}_{\text{ankle}} + \mathbf{n} \cdot h) - \mathbf{p}^{\text{cam}}_{\text{ankle}} \times \mathbf{K}(\mathbf{p}^{\text{cam}}_{\text{ankle}}) = 0 \tag{3.3}$$

with h the person height, $\mathbf{n}$ the normal direction and $\mathbf{p}_{\text{shoulder}}$ and $\mathbf{p}_{\text{ankle}}$ the shoulder and ankle positions. In the following we subscript variables with an x, y, z to indicate the x,y,z-coordinates and with a number $1, 2...$ to refer to different person locations.

In matrix form, using $\Delta\mathbf{p}_x = \mathbf{p}^{\text{img},x}_{\text{shoulder}} - \mathbf{p}^{\text{img},x}_{\text{ankle}}$, $\Delta\mathbf{p}_y = \mathbf{p}^{\text{img},y}_{\text{shoulder}} - \mathbf{p}^{\text{img},y}_{\text{ankle}}$, and z to represent the unknown depth of the ankle, Eq. 3.3 can be expressed as

$$\begin{pmatrix} 0 & -1 & \mathbf{p}^{\text{img},y}_{\text{shoulder}} & 0 & -1 & \Delta\mathbf{p}_y \\ 1 & 0 & -\mathbf{p}^{\text{img},x}_{\text{shoulder}} & 1 & 0 & -\Delta\mathbf{p}_x \end{pmatrix} \begin{pmatrix} f_1\mathbf{n}_x \\ f_2\mathbf{n}_y \\ \mathbf{n}_z \\ \mathbf{n}_z\mathbf{o}_x \\ \mathbf{n}_z\mathbf{o}_y \\ z/h \end{pmatrix} = 0, \tag{3.4}$$

where f is the focal length and $\mathbf{o}$ the principal point of the camera intrinsics $\mathbf{K}$. By using at least three 2D shoulder $\mathbf{p}_{\text{shoulder}}$ and ankle $\mathbf{p}_{\text{ankle}}$ detections, we form the constraint matrix

$$\mathbf{D} = \begin{pmatrix} 0 & -1 & \mathbf{p}^{\text{img},y_1}_{\text{shoulder}} & \Delta\mathbf{p}_{y1} & 0 & 0 \\ 1 & 0 & -\mathbf{p}^{\text{img},x_1}_{\text{shoulder}} & -\Delta\mathbf{p}_{x1} & 0 & 0 \\ 0 & -1 & \mathbf{p}^{\text{img},y_2}_{\text{shoulder}} & 0 & \Delta\mathbf{p}_{y2} & 0 \\ 1 & 0 & -\mathbf{p}^{\text{img},x_2}_{\text{shoulder}} & 0 & -\Delta\mathbf{p}_{x2} & 0 \\ 0 & -1 & \mathbf{p}^{\text{img},y_3}_{\text{shoulder}} & 0 & 0 & \Delta\mathbf{p}_{y3} \\ 1 & 0 & -\mathbf{p}^{\text{img},x_3}_{\text{shoulder}} & 0 & 0 & -\Delta\mathbf{p}_{x3} \end{pmatrix} \tag{3.5}$$

that gives the system of equations

$$\mathbf{D}\begin{pmatrix} f_1\mathbf{n}_x+\mathbf{n}_z\mathbf{o}_x \\ f_2\mathbf{n}_y+\mathbf{n}_z\mathbf{o}_y \\ \mathbf{n}_z \\ z_1/h \\ z_2/h \\ z_3/h \end{pmatrix}=0. \tag{3.6}$$

We solve Eq. 3.6 using SVD by finding the singular vector that corresponds to the smallest singular value. Having more than three ankles and shoulders results in an over-determined system, for which we can find a least-squares solution.

Ground normal extraction. Since Eq. 3.6 is a 6 × 6 system with rank five, any solution we find is unique up to a scalar. In order to determine **n** from the SVD or least-squares solution, we use the fact that the normal vector is perpendicular to any vector formed by a pair of ankles. Using

$$\begin{bmatrix} \bar{\mathbf{n}}_x \\ \bar{\mathbf{n}}_y \\ \bar{\mathbf{n}}_z \\ \bar{z}_1 \\ \bar{z}_2 \\ \bar{z}_3 \end{bmatrix}=\lambda\begin{bmatrix} \mathbf{n}_x+\mathbf{n}_z\mathbf{o}_x/f \\ \mathbf{n}_y+\mathbf{n}_z\mathbf{o}_y/f \\ \mathbf{n}_z/f \\ z_1/(hf) \\ z_2/(hf) \\ z_3/(hf) \end{bmatrix}, \tag{3.7}$$

If we do not have a given focal length, we can derive the equation for the focal length. Using the assumption that f_1 equals f_2, we will simply write f to represent both focal lengths. First, we take the cross product between the normal vector and a vector in the ground plane consisting of two person's ankles, which we write as

$$\mathbf{n}_x(\mathbf{p}^{\text{cam},x_1}_{\text{shoulder}}-\mathbf{p}^{\text{cam},x_2}_{\text{shoulder}})+\mathbf{n}_y(\mathbf{p}^{\text{cam},y_1}_{\text{shoulder}}-\mathbf{p}^{\text{cam},y_2}_{\text{shoulder}})+\mathbf{n}_z(\mathbf{p}^{\text{cam},z_1}_{\text{shoulder}}-\mathbf{p}^{\text{cam},z_2}_{\text{shoulder}})=0 \tag{3.8}$$

Then, by substituting Eq. 3.7 with Eq.3.8, we get

$$f = \sqrt{\frac{(-(\bar{\mathbf{n}}_x - \bar{\mathbf{n}}_z \mathbf{o}_x)\bar{\mathbf{p}}_\mathrm{x} - (\bar{\mathbf{n}}_y - \bar{n}_z \mathbf{o}_y)\bar{\mathbf{p}}_\mathrm{y})}{(\bar{\mathbf{n}}_z(\bar{z}_1 - \bar{z}_2))}} \tag{3.9}$$

with

$$\bar{\mathbf{p}}_\mathrm{x} = ((\mathbf{p}_\mathrm{ankle}^{\mathrm{img},x_1} - \mathbf{o}_x)\bar{z}_1 - (\mathbf{p}_\mathrm{ankle}^{\mathrm{img},x_2} - \mathbf{o}_x)\bar{z}_2) \tag{3.10}$$

and

$$\bar{\mathbf{p}}_\mathrm{y} = ((\mathbf{p}_\mathrm{ankle}^{\mathrm{img},y_1} - \mathbf{o}_y)\bar{z}_1 - (\mathbf{p}_\mathrm{ankle}^{\mathrm{img},y_2} - \mathbf{o}_y)\bar{z}_2). \tag{3.11}$$

If we assume that f_1 is not equal to f_2, we can a second pair of ankles, take the cross product of it with the normal vector to get

$$\mathbf{n}_x(\mathbf{p}_\mathrm{shoulder}^{\mathrm{cam},x_1} - \mathbf{p}_\mathrm{shoulder}^{\mathrm{cam},x_3}) + \mathbf{n}_y(\mathbf{p}_\mathrm{shoulder}^{\mathrm{cam},y_1} - \mathbf{p}_\mathrm{shoulder}^{\mathrm{cam},y_3}) + \mathbf{n}_z(\mathbf{p}_\mathrm{shoulder}^{\mathrm{cam},z_1} - \mathbf{p}_\mathrm{shoulder}^{\mathrm{cam},z_3}) = 0. \tag{3.12}$$

Then we can perform the same substitution Eq. 3.7 with Eq.3.8 and Eq. 3.7 with Eq.3.12 and solve the system of equations using least squares. Either the estimated focal lengths or a given focal length enables us to recover $\lambda\mathbf{n}$ and $\lambda(z_1, z_2, z_3)$. We normalize $\lambda\mathbf{n}$ and $\lambda(z_1, z_2, z_3)$ by dividing them by the L_2 norm of $\lambda\mathbf{n}$. This results in a unique $\mathbf{n}$ of length one and ankle depths z_1, z_2, and z_3 since we assume that $\mathbf{n}$ is a normal vector with length one, this means that the norm of $\lambda\mathbf{n}$ is equal to λ. Using the normal vector $\mathbf{n}$ and the known depths z_1, z_2, and z_3, we recover the orientation and position of the ground plane.

RANSAC

In practice, we apply RANSAC [16] to filter out outliers and we select the largest inlier set. In order to determine which detections are inliers, we first reproject the ankle coordinates $\mathbf{p}_\mathrm{ankle}^\mathrm{img}$ to 3D $\mathbf{p}_\mathrm{ankle}^\mathrm{cam}$. Second, we add $h\mathbf{n}$ to $\mathbf{p}_\mathrm{ankle}^\mathrm{cam}$ to get the predicted $\mathbf{p}_\mathrm{shoulder}^\mathrm{cam} = \mathbf{p}_\mathrm{ankle} + h\mathbf{n}$. Finally, we reproject this point back to image coordinates using Eq. 3.1. We use two metrics, shoulder pixel error,

$$\text{Pixel Error} = \frac{||\mathbf{p}_\mathrm{shoulder}^\mathrm{img} - \mathbf{p}_\mathrm{pred\ shoulder}^\mathrm{img}||}{||\mathbf{p}_\mathrm{pred\ shoulder}^\mathrm{img} - \mathbf{p}_\mathrm{ankle}^\mathrm{img}||}. \tag{3.13}$$

and the angle error,

$$\text{Angle Error} = \arccos\left(\frac{(\mathbf{p}^{\text{img}}_{\text{pred shoulder}} - \mathbf{p}^{\text{img}}_{\text{ankle}}) \cdot (\mathbf{p}^{\text{img}}_{\text{shoulder}} - \mathbf{p}^{\text{img}}_{\text{ankle}})}{||\mathbf{p}^{\text{img}}_{\text{pred shoulder}} - \mathbf{p}^{\text{img}}_{\text{ankle}}||\,||\mathbf{p}^{\text{img}}_{\text{shoulder}} - \mathbf{p}^{\text{img}}_{\text{ankle}}||}\right), \tag{3.14}$$

to determine if it is an inlier. The pixel error is computed as the pixel Euclidean distance between the shoulder detection and the predicted shoulder. We normalize this by the 2D pixel height of the person. The angle error is computed by computing the angle between the vector from the $\mathbf{p}^{\text{img}}_{\text{ankle}}$ to the $\mathbf{p}^{\text{img}}_{\text{shoulder}}$ and the vector from $\mathbf{p}^{\text{img}}_{\text{ankle}}$ to the predicted $\mathbf{p}^{\text{img}}_{\text{shoulder}}$. An ankle-shoulder pair is considered an inlier if is less than both the angle and pixel threshold.

For our experiments, we use an angle threshold of 2.86 degrees and a pixel threshold of 5 percent of the pixel height, which was determined on a validation set. Once we have the largest inlier set, we run our DLT method on the entire inlier set to get the final focal length and ground plane position and orientation. We show an example of our single view calibration algorithm outcome in Figure 3.2.

Filtering Since we have an assumption that the persons are standing straight up, we must filter out the non-standing poses in the scene. To determine this, for every 2D pose **p**, we measure the 2D angles between the vectors from shoulder to hip, hip to knee, and knee to ankle. We use equation

$$\text{Filter}(\mathbf{p}^{\text{img}}) = \min(L_{\text{right}}, L_{\text{left}}) \tag{3.15}$$

where

$$L_{\text{right}} = |\frac{\hat{\mathbf{p}}_{\text{right}_1} \cdot \hat{\mathbf{p}}_{\text{right}_2}}{||\hat{\mathbf{p}}_{\text{right}_1}||\,||\hat{\mathbf{p}}_{\text{right}_2}||} - \pi| + |\frac{\hat{\mathbf{p}}_{\text{right}_2} \cdot \hat{\mathbf{p}}_{\text{right}_3}}{||\hat{\mathbf{p}}_{\text{right}_2}||\,||\hat{\mathbf{p}}_{\text{right}_3}||} - \pi|, \tag{3.16}$$

$$L_{\text{left}} = |\frac{\hat{\mathbf{p}}_{\text{left}_1} \cdot \hat{\mathbf{p}}_{\text{left}_2}}{||\hat{\mathbf{p}}_{\text{left}_1}||\,||\hat{\mathbf{p}}_{\text{left}_2}||} - \pi| + |\frac{\hat{\mathbf{p}}_{\text{left}_2} \cdot \hat{\mathbf{p}}_{\text{left}_3}}{||\hat{\mathbf{p}}_{\text{left}_2}||\,||\hat{\mathbf{p}}_{\text{left}_3}||} - \pi|, \tag{3.17}$$

$$\hat{\mathbf{p}}_{\text{right}_1} = \mathbf{p}^{\text{img}}_{\text{right ankle}} - \mathbf{p}^{\text{img}}_{\text{right knee}}, \tag{3.18}$$

$$\hat{\mathbf{p}}_{\text{right}_2} = \mathbf{p}^{\text{img}}_{\text{right hip}} - \mathbf{p}^{\text{img}}_{\text{right knee}}, \tag{3.19}$$

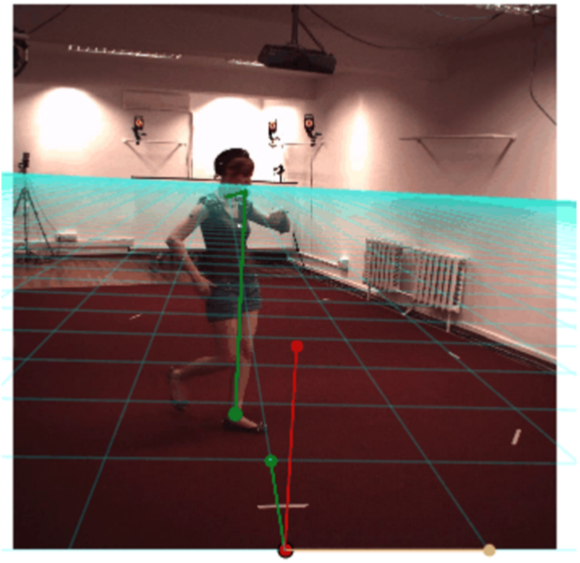


Figure 3.2: 2D Reconstruction. Visual results for the single view calibration for Human3.6M Subject 1. The blue grid represents the ground plane predicted by our method with a coordinate axis defined at the bottom of the image. The green line from the ankle to the shoulders represents the ankle to shoulder keypoints.

$$\hat{\mathbf{p}}_{\text{right}_3} = \mathbf{p}^{\text{img}}_{\text{right shoulder}} - \mathbf{p}^{\text{img}}_{\text{right knee}}, \tag{3.20}$$

$$\hat{\mathbf{p}}_{\text{left}_1} = \mathbf{p}^{\text{img}}_{\text{left ankle}} - \mathbf{p}^{\text{img}}_{\text{left knee}}, \tag{3.21}$$

$$\hat{\mathbf{p}}_{\text{left}_2} = \mathbf{p}^{\text{img}}_{\text{left hip}} - \mathbf{p}^{\text{img}}_{\text{left knee}}, \tag{3.22}$$

$$\hat{\mathbf{p}}_{\text{left}_3} = \mathbf{p}^{\text{img}}_{\text{left shoulder}} - \mathbf{p}^{\text{img}}_{\text{left knee}}, \tag{3.23}$$

to determine if a pose is standing straight up by measuring the angle of the knee keypoint and the angle of the hip keypoint.

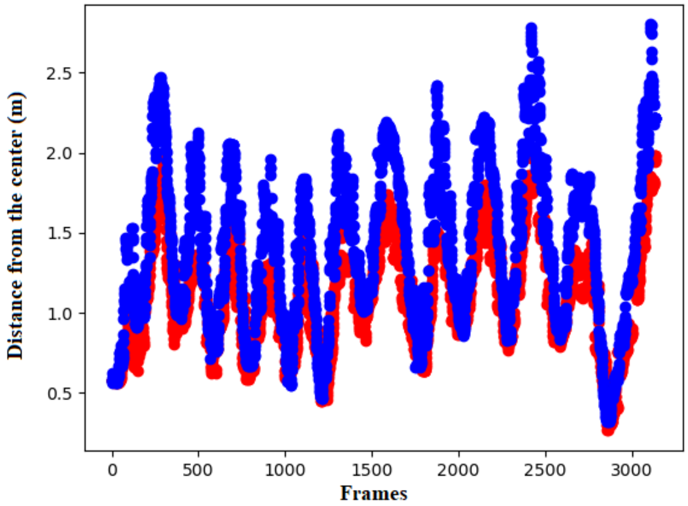


Figure 3.3: Time Synchronization. Time synchronization results between ref (red) and sync (blue) sequences for subject 1 walking sequence in Human3.6M.

Relating multiple cameras through their ground planes The relation between individually calibrated cameras is unknown, but the estimated ground plane is shared. We select one camera to be the reference camera, and use its plane coordinates as the world coordinate system. To simplify subsequent steps, we compute for each other camera the homography transformation from image coordinates $\mathbf{p}^{\text{img}}$ to the estimated ground plane,

$$\mathbf{p}^{\text{plane}} = [\mathbf{R}^{\text{cam}\to\text{plane}} | \boldsymbol{\tau}^{\text{plane}}]\mathbf{K}^{-1}\mathbf{p}^{\text{img}}. \tag{3.24}$$

Figure 3.4 shows the resulting birds-eye view of the ground plane with estimated person positions. The plane normal vector $\mathbf{n}$ is shared between all cameras and defines one column of $\mathbf{R}^{\text{cam}\to\text{plane}}$. For each camera, we derive the other 2 basis vectors by the back-projection of the 2D horizontal line to 3D as the new x-axis, and finally the cross product of the normal vector with the x-axis as the z-axis. The position $\boldsymbol{\tau}^{\text{plane}}$ is the back projection of the image center to the ground plane. This construction is intermittent. It remains to align the 2D position and orientation

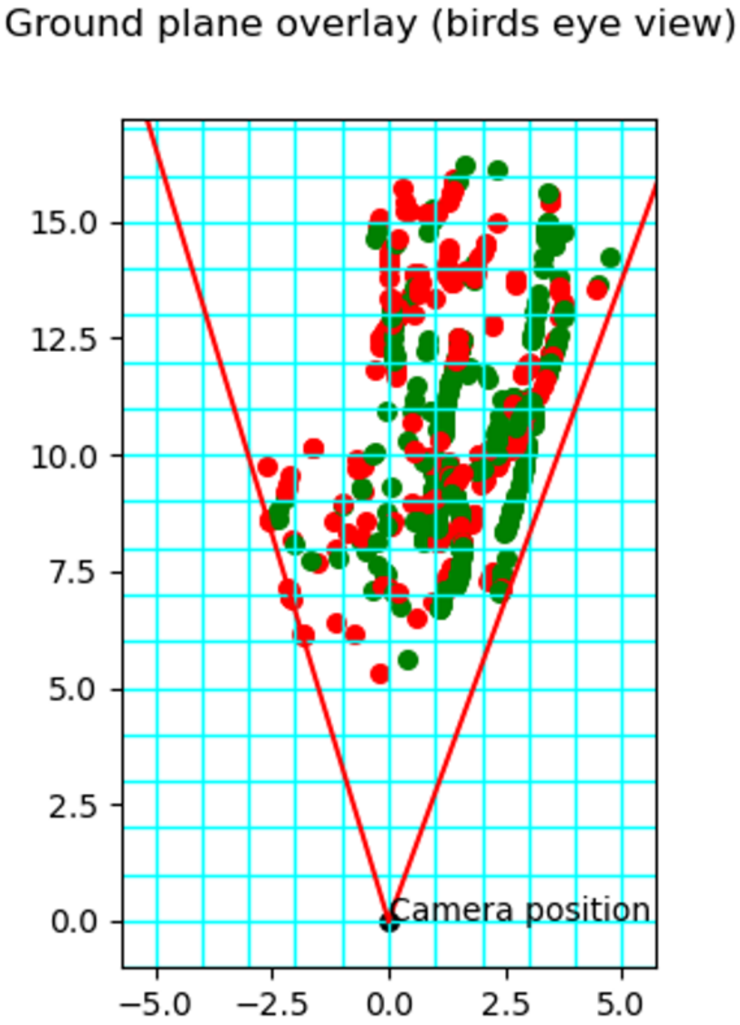


Figure 3.4: Ground Plane View Bird's eye view of the ankles on the Terrace sequence, with inliers in green and outliers in red.

within the ground plane to fully determine the camera extrinsics, as well as to estimate the time shift.

3.2 1D Temporal Search

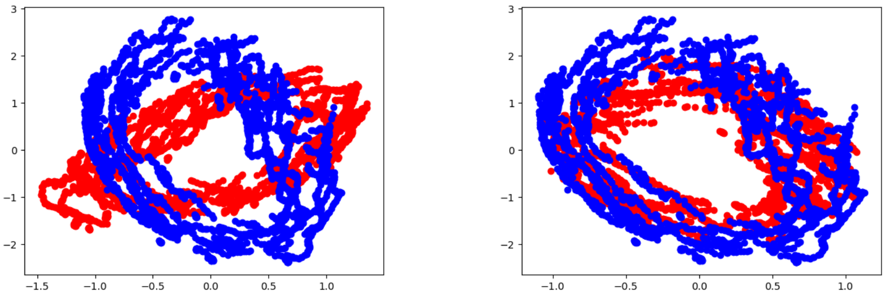
To support cameras starting or ending at different times, we model the time relationship pairwise between cameras as $t_{\text{ref}} = t_{\text{sync}} + \Delta t_{\text{sync}}$, a linear relationship between reference camera sequence t_{ref} and t_{sync} of the target camera. To find the translation Δt_{sync}, first, we project the detected ankle points onto the ground plane using $\mathbf{R}^{\text{cam}\to\text{plane}}$ and shift them such that the mean of the reference set is the same as the mean of sync set. In order to get a signal that is time-sensitive but does not depend on the unknown camera extrinsics, we compute the distance d from the center for each point. Since the search space is 1D, we can afford a brute-force search, with candidate offsets ranging from 0 to one-third of the length of the sync sequence. Note that if there is more than one person in the frame, then that time step has more than one distance associated with it, which we handle with an optimal assignment step. We show an example of the temporal alignment in Figure 3.3.

Search criteria The alignment is scored by the absolute difference of d_{sync} and d_{ref} within the same time step. If there is more than one person in the frame, we compute an optimal matching using the Hungarian algorithm [25]. While shifting the curves temporally, we continue the endpoints of the curves by repeating the endpoint values. This helps prevent the curve from shifting too much since larger shifts lead to smaller overlap and hence larger uncertainty since we are not matching large amounts of the curve.

Filtering Since noisy detections could cause outlier points to appear on the ground plane, we remove outlier points on the ground plane using a density-based spatial clustering of applications with noise (DBSCAN) [14] to find the largest cluster of points on the ground plane, and then remove all the outlier points.

3.3 2D Rotation Search

Once we match the videos temporally, we complete the extrinsic calibration between the cameras. First, we shift the means of the ankle positions in 2D plane coordinates from the *sync* camera sequence to align with the reference camera's



(a) Initial orientation

(b) Best rotation

Figure 3.5: 2D rotation search. Visual results for 2D rotation search on Human3.6M subject 1 with 2 cameras. Axes are in meters.

sequence. Then we search rotation angles from 0 to 360 degrees and apply the rotation to each sequence. For each camera i, we compute our error augmenting our detections $\mathbf{p}_i^{\text{plane}}$ with the time step, $\hat{\mathbf{p}}_i^{\text{plane}} = (x, y, t)$, and computing the distance between the closest points in the point cloud. Note that no closed-form solution is possible, since correspondences between the two point clouds are unknown.

We compute our error using Equation 3.25, where we augment our detections $\mathbf{p}_i^{\text{plane}}$ with the time step, which we notate as $\hat{\mathbf{p}}_i^{\text{plane}} = (x, y, t)$ we define c_1 and c_2 as camera 1 and camera 2 or the ref camera and the sync camera. We also define $\hat{F}$, $\hat{P}_{c,j}$, and $\hat{K}_{c,j,p}$ to represent the sets of indices of the frames, poses for view c and frame i, and keypoints for view c, frame i, and pose p.

$$\begin{aligned}
L_{\text{min}}(\mathrm{c}_1, \mathrm{c}_2, i, k_1) &= \min_{k_2 \in \hat{P}_{\mathrm{c}_2, i}} \|\hat{\mathbf{p}}^{\text{plane}}_{\mathrm{c}_1, i, k_1, \text{ankle}} - \hat{\mathbf{p}}^{\text{plane}}_{\mathrm{c}_2, i, k_2, \text{ankle}}\| \\
L_{\text{t}}(\mathrm{c}_1, \mathrm{c}_2) &= \sum_{i \in \hat{F}, k_1 \in \hat{P}_{\mathrm{c}_1, i}} L_{\text{min}}(\mathrm{c}_1, \mathrm{c}_2, i, k_1) \qquad (3.25) \\
L_{\text{rot}}(\mathrm{c}_1, \mathrm{c}_2) &= L_{\text{t}}(\mathrm{c}_1, \mathrm{c}_2) + L_{\text{t}}(\mathrm{c}_2, \mathrm{c}_1)
\end{aligned}$$

We show an example of the point cloud alignment process in Figure 3.5.

3.4 Iterative Closest Point

The previous section yields a first estimate of all camera extrinsics by estimating the 2D plane rotation and position of cameras relative to each other that remained unknown in Step 1. We refine that estimate using the Iterative Closest Point (ICP) [57]. We find the closest points by utilizing the previously estimated time synchronization to match the frames from the reference view to the synchronization view and also the same Hungarian matching process, when multiple persons are present. We then optimize the rotation and translation by minimizing the Euclidean distance between the 2D point clouds in the plane. Iteratively, we re-associate the points and repeat the process. Note that the initial 2D rotation search searches all angles between 0 and 360, making adjustments from the ICP small.

3.5 Joint Camera Refinement (Bundle Adjustment)

Once we get the result from our ICP step, we pick the top k poses with the highest confidence to use in the final bundle adjustment [50] step that refines all calibration parameters jointly, by using the association of keypoints from the previous timesteps. As opposed to previous steps using ankle and shoulder, we can now incorporate all body part detections. However, because the head and arm keypoints have a larger range of motion and are often self-occluded, we exclude them during the bundle adjustment. For camera pairs i and j, we can define the relationship from 2D to 3D as a line using $\ell(k) = m_j k + \boldsymbol{\tau}_j$, with $k \in \mathbb{R}$ and

$$m_j = \mathbf{R}_j^{\text{plane}\rightarrow\text{world}}(\mathbf{R}_j^{\text{cam}\rightarrow\text{plane}}\mathbf{K}_i^{-1}\mathbf{P}_j^{\text{img}} - \boldsymbol{\tau}_j^{\text{cam}\rightarrow\text{plane}}). \tag{3.26}$$

We optimize every camera pair c_1 and c_2 with gradient descent using the objective function

$$L^{\text{c}_1,\text{c}_2} = \alpha_0 L_{\text{3D}}^{\text{c}_1,\text{c}_2} + \alpha_1 L_{\text{left, right}}^{\text{c}_1,\text{c}_2} + \alpha_2(L_{\text{h}}^{\text{c}_1} + L_{\text{h}}^{\text{c}_2}) + \alpha_3(L_{\text{p}}^{\text{c}_1} + L_{\text{p}}^{\text{c}_2)}. \tag{3.27}$$

The terms include the intersection error

$$L_{\text{3D}}^{\text{c}_1,\text{c}_2} = \frac{\sum_{j\in\hat{F},p\in\hat{P}_j,k\in\hat{K}_{j,p}} ||\mathbf{p}_{\text{c}_1,j,p,k}^{\text{world}} - \mathbf{p}_{\text{c}_2,j,p,k}^{\text{world}}||_2}{|\hat{F}||\hat{P}||\hat{K}|} \tag{3.28}$$

which is the Euclidean distance between the two projections, where $\hat{F}$, $\hat{P}$, and $\hat{K}$ represents the frames, poses, and keypoints with $|\hat{F}|$, $|\hat{P}|$, and $|\hat{K}|$ representing the cardinality; left and right joint symmetry error

$$L_{\text{left,right}}^{\text{c}_1,\text{c}_2} = \frac{\sum_{j\in\hat{F},p\in\hat{P}_j,k\in\hat{K}_{j,p}} |||J_{\text{c}_1,j,p,k}^{\text{world}}||_2 - ||J_{\text{c}_2,j,p,k}^{\text{world}}||_2|}{|\hat{F}||\hat{P}||\hat{K}|}, \tag{3.29}$$

which constrains bones on the left and right body side to be equal lengths; the height error

$$L_{\text{h}}^{\text{c}} = \frac{\sum_{j\in\hat{F},p\in\hat{P}_j,k\in\hat{K}_{j,p}} ||J_{\text{c}_1,j,p,k}^{\text{world}}||_2}{|\hat{F}||\hat{P}||\hat{K}|}, \tag{3.30}$$

which constrains the sum of the lengths of the joints from ankle to shoulder to be the same; and the constraint that the ankle is on the plane L_p

$$L_{\text{p}}^{\text{cam}} = \frac{\sum_{j\in\hat{F},p\in\hat{P}_j} ||\mathbf{p}^{\text{world}}(z)_{\text{c},j,p,\text{ankle}}||}{|\hat{F}||\hat{P}||\hat{K}|}. \tag{3.31}$$

We weight these components using $\alpha_0 = 1$, $\alpha_1 = 10$, $\alpha_2 = 10$, and $\alpha_3 = 0.1$. We use the EPFL terrace2 as a validation set for our hyperparameters.

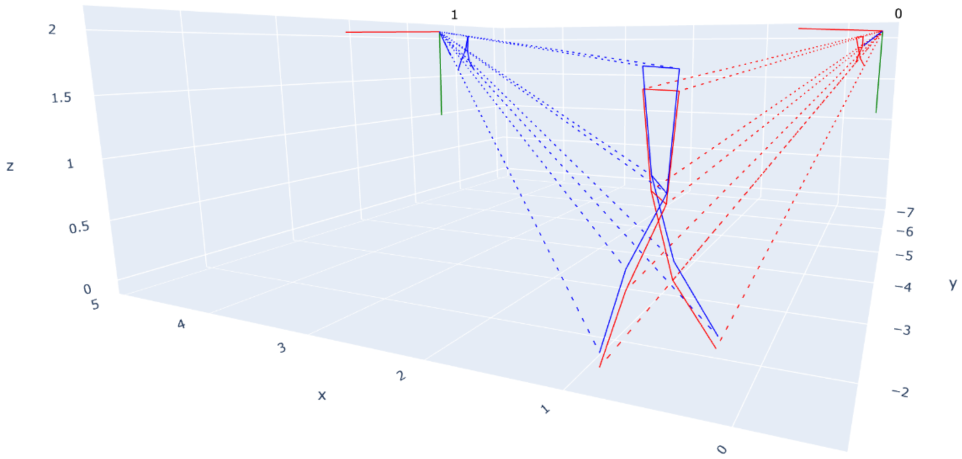


Figure 3.6: Bundle Adjustment. Visual 3D pose reconstruction for Human3.6M subject 1 with 2 cameras. The red and blue lines represent the triangulation of the pose from the two views. The dotted lines represent the reprojection lines from the camera to the person.

Chapter 4

Statistical Analysis

Throughout this chapter, we derive theoretical results for our DLT method, which we introduced in Chapter 3, in order to have a better understanding of the effects of detection noise and height variations. This chapter is divided into three sections: rank of the DLT matrix, probabilistic interpretation, and the experiments.

4.1 Rank of the DLT matrix

First, we establish a few properties of the DLT matrix,

$$\mathbf{D} = \begin{pmatrix} 0 & -1 & \mathbf{p}_{y1}^{\text{shoulder}} & \Delta\mathbf{p}_{y1} & 0 & 0 \\ 1 & 0 & -\mathbf{p}_{x1}^{\text{shoulder}} & -\Delta\mathbf{p}_{x1} & 0 & 0 \\ 0 & -1 & \mathbf{p}_{y2}^{\text{shoulder}} & 0 & \Delta\mathbf{p}_{y2} & 0 \\ 1 & 0 & -\mathbf{p}_{x2}^{\text{shoulder}} & 0 & -\Delta\mathbf{p}_{x2} & 0 \\ 0 & -1 & \mathbf{p}_{y3}^{\text{shoulder}} & 0 & 0 & \Delta\mathbf{p}_{y3} \\ 1 & 0 & -\mathbf{p}_{x3}^{\text{shoulder}} & 0 & 0 & -\Delta\mathbf{p}_{x3} \end{pmatrix}. \tag{4.1}$$

The rank of the D matrix is not entirely obvious. However, we can determine the rank by computing the determinant of D, which is given by the equation 4.2,

$$
\begin{aligned}
\det D = {} & -\mathbf{p}^{a}_{x1}\mathbf{p}^{a}_{x2}\mathbf{p}^{a}_{y3}\mathbf{p}^{s}_{y1} + \mathbf{p}^{a}_{x1}\mathbf{p}^{a}_{x2}\mathbf{p}^{a}_{y3}\mathbf{p}^{s}_{y2} + \mathbf{p}^{a}_{x1}\mathbf{p}^{a}_{x2}\mathbf{p}^{s}_{y1}\mathbf{p}^{s}_{y3} - \mathbf{p}^{a}_{x1}\mathbf{p}^{a}_{x2}\mathbf{p}^{s}_{y2}\mathbf{p}^{s}_{y3} + \\
& \mathbf{p}^{a}_{x1}\mathbf{p}^{a}_{x3}\mathbf{p}^{a}_{y2}\mathbf{p}^{s}_{y1} - \mathbf{p}^{a}_{x1}\mathbf{p}^{a}_{x3}\mathbf{p}^{a}_{y2}\mathbf{p}^{s}_{y3} - \mathbf{p}^{a}_{x1}\mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{y1}\mathbf{p}^{s}_{y2} + \mathbf{p}^{a}_{x1}\mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{y2}\mathbf{p}^{s}_{y3} - \\
& \mathbf{p}^{a}_{x1}\mathbf{p}^{s}_{x2}\mathbf{p}^{a}_{y2}\mathbf{p}^{a}_{y3} + \mathbf{p}^{a}_{x1}\mathbf{p}^{s}_{x2}\mathbf{p}^{a}_{y2}\mathbf{p}^{s}_{y3} + \mathbf{p}^{a}_{x1}\mathbf{p}^{s}_{x2}\mathbf{p}^{a}_{y3}\mathbf{p}^{s}_{y1} - \mathbf{p}^{a}_{x1}\mathbf{p}^{s}_{x2}\mathbf{p}^{s}_{y1}\mathbf{p}^{s}_{y3} + \\
& \mathbf{p}^{a}_{x1}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y2}\mathbf{p}^{a}_{y3} - \mathbf{p}^{a}_{x1}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y2}\mathbf{p}^{s}_{y1} - \mathbf{p}^{a}_{x1}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y3}\mathbf{p}^{s}_{y2} + \mathbf{p}^{a}_{x1}\mathbf{p}^{s}_{x3}\mathbf{p}^{s}_{y1}\mathbf{p}^{s}_{y2} - \\
& \mathbf{p}^{a}_{x2}\mathbf{p}^{a}_{x3}\mathbf{p}^{a}_{y1}\mathbf{p}^{s}_{y2} + \mathbf{p}^{a}_{x2}\mathbf{p}^{a}_{x3}\mathbf{p}^{a}_{y1}\mathbf{p}^{s}_{y3} + \mathbf{p}^{a}_{x2}\mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{y1}\mathbf{p}^{s}_{y2} - \mathbf{p}^{a}_{x2}\mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{y1}\mathbf{p}^{s}_{y3} + \\
& \mathbf{p}^{a}_{x2}\mathbf{p}^{s}_{x1}\mathbf{p}^{a}_{y1}\mathbf{p}^{a}_{y3} - \mathbf{p}^{a}_{x2}\mathbf{p}^{s}_{x1}\mathbf{p}^{a}_{y1}\mathbf{p}^{s}_{y3} - \mathbf{p}^{a}_{x2}\mathbf{p}^{s}_{x1}\mathbf{p}^{a}_{y3}\mathbf{p}^{s}_{y2} + \mathbf{p}^{a}_{x2}\mathbf{p}^{s}_{x1}\mathbf{p}^{s}_{y2}\mathbf{p}^{s}_{y3} - \\
& \mathbf{p}^{a}_{x2}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y1}\mathbf{p}^{a}_{y3} + \mathbf{p}^{a}_{x2}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y1}\mathbf{p}^{s}_{y2} + \mathbf{p}^{a}_{x2}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y3}\mathbf{p}^{s}_{y1} - \mathbf{p}^{a}_{x2}\mathbf{p}^{s}_{x3}\mathbf{p}^{s}_{y1}\mathbf{p}^{s}_{y2} - \\
& \mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{x1}\mathbf{p}^{a}_{y1}\mathbf{p}^{a}_{y2} + \mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{x1}\mathbf{p}^{a}_{y1}\mathbf{p}^{s}_{y2} + \mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{x1}\mathbf{p}^{a}_{y2}\mathbf{p}^{s}_{y3} - \mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{x1}\mathbf{p}^{s}_{y2}\mathbf{p}^{s}_{y3} + \\
& \mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{x2}\mathbf{p}^{a}_{y1}\mathbf{p}^{a}_{y2} - \mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{x2}\mathbf{p}^{a}_{y1}\mathbf{p}^{s}_{y3} - \mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{x2}\mathbf{p}^{a}_{y2}\mathbf{p}^{s}_{y1} + \mathbf{p}^{a}_{x3}\mathbf{p}^{s}_{x2}\mathbf{p}^{s}_{y1}\mathbf{p}^{s}_{y3} - \\
& \mathbf{p}^{s}_{x1}\mathbf{p}^{s}_{x2}\mathbf{p}^{a}_{y1}\mathbf{p}^{a}_{y3} + \mathbf{p}^{s}_{x1}\mathbf{p}^{s}_{x2}\mathbf{p}^{a}_{y1}\mathbf{p}^{s}_{y3} + \mathbf{p}^{s}_{x1}\mathbf{p}^{s}_{x2}\mathbf{p}^{a}_{y2}\mathbf{p}^{a}_{y3} - \mathbf{p}^{s}_{x1}\mathbf{p}^{s}_{x2}\mathbf{p}^{a}_{y2}\mathbf{p}^{s}_{y3} + \\
& \mathbf{p}^{s}_{x1}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y1}\mathbf{p}^{a}_{y2} - \mathbf{p}^{s}_{x1}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y1}\mathbf{p}^{s}_{y2} - \mathbf{p}^{s}_{x1}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y2}\mathbf{p}^{a}_{y3} + \mathbf{p}^{s}_{x1}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y3}\mathbf{p}^{s}_{y2} - \\
& \mathbf{p}^{s}_{x2}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y1}\mathbf{p}^{a}_{y2} + \mathbf{p}^{s}_{x2}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y1}\mathbf{p}^{a}_{y3} + \mathbf{p}^{s}_{x2}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y2}\mathbf{p}^{s}_{y1} - \mathbf{p}^{s}_{x2}\mathbf{p}^{s}_{x3}\mathbf{p}^{a}_{y3}\mathbf{p}^{s}_{y1}.
\end{aligned}
\tag{4.2}
$$

What equation 4.2 tells us is that in general, we cannot be certain if matrix 4.1 is singular or not. This means that for detections **p** in general, the nullspace has dimension 0, so this problem only has a trivial solution. However, detections **p** are not arbitrary since if our assumptions hold, detections **p** follow Equation 4.3,

$$
\begin{aligned}
\mathbf{p}_x^{\text{ankle}} &= (fx + z\mathbf{o}_x)/z, \\
\mathbf{p}_y^{\text{ankle}} &= (fy + z\mathbf{o}_y)/z, \\
\mathbf{p}_x^{\text{shoulder}} &= (f(x + \mathbf{n}_1 h) + (z + \mathbf{n}_3 h)\mathbf{o}_x)/(z + \mathbf{n}_3 h), \\
\mathbf{p}_y^{\text{shoulder}} &= (f(y + \mathbf{n}_2 h) + (z + \mathbf{n}_3 h)\mathbf{o}_y)/(z + \mathbf{n}_3 h).
\end{aligned}
\tag{4.3}
$$

We show in the next section that if detections **p** satisfy Equation 4.3, which corresponds to the projection from 3D camera coordinates to 2D image coordinates, then the rank is 5.

4.1.1 Showing that the DLT matrix has rank 5

If detections **p** satisfy 4.3, we can show by substitution in 4.2 that the determinant is 0. This means that the rank of Matrix 4.1 is less than 6. In order to determine that the rank is 5, we must find a submatrix of size 5 by 5 with a nonzero determinant. We define the submatrix D_{sub} as Equation 4.4,

$$D_{\text{sub}} = \begin{pmatrix} 0 & -1 & (\mathbf{p}_{y1}^{\text{shoulder}} - \mathbf{p}_{y1}^{\text{ankle}}) & 0 & 0 \\ 1 & 0 & (\mathbf{p}_{x_1}^{\text{ankle}} - \mathbf{p}_{x_1}^{\text{shoulder}}) & 0 & 0 \\ 1 & 0 & 0 & (\mathbf{p}_{x_2}^{\text{ankle}} - \mathbf{p}_{x_2}^{\text{shoulder}}) & 0 \\ 0 & -1 & 0 & 0 & (\mathbf{p}_{y3}^{\text{shoulder}} - \mathbf{p}_{y3}^{\text{ankle}}) \\ 1 & 0 & 0 & 0 & (\mathbf{p}_{x3}^{\text{ankle}} - \mathbf{p}_{x3}^{\text{shoulder}}) \end{pmatrix}. \tag{4.4}$$

We compute the determinant of D_{sub}, which is given in Equation 4.5,

$$\begin{aligned} \det D_{\text{sub}} = -(&f^3h^3\mathbf{n}_1^2\mathbf{n}_3y_1z_2z_3 - f^3h^3\mathbf{n}_1^2\mathbf{n}_3y_3z_1z_2 - f^3h^3\mathbf{n}_1\mathbf{n}_2\mathbf{n}_3x_1z_2z_3 + \\ &f^3h^3\mathbf{n}_1\mathbf{n}_2\mathbf{n}_3x_3z_1z_2 + f^3h^3\mathbf{n}_1\mathbf{n}_3^2x_1y_3z_2 - f^3h^3\mathbf{n}_1\mathbf{n}_3^2x_2y_1z_3 + \\ &f^3h^3\mathbf{n}_1\mathbf{n}_3^2x_2y_3z_1 - f^3h^3\mathbf{n}_1\mathbf{n}_3^2x_3y_1z_2 + f^3h^3\mathbf{n}_2\mathbf{n}_3^2x_1x_2z_3 - \\ &f^3h^3\mathbf{n}_2\mathbf{n}_3^2x_2x_3z_1 - f^3h^3\mathbf{n}_3^3x_1x_2y_3 + f^3h^3\mathbf{n}_3^3x_2x_3y_1)/ \\ &(h^3\mathbf{n}_3^3z_1z_2z_3 + h^2\mathbf{n}_3^2z_1^2z_2z_3 + h^2\mathbf{n}_3^2z_1z_2^2z_3 + \\ &h^2\mathbf{n}_3^2z_1z_2z_3^2 + h\mathbf{n}_3z_1^2z_2^2z_3 + h\mathbf{n}_3z_1^2z_2z_3^2 + \\ &h\mathbf{n}_3z_1z_2^2z_3^2 + z_1^2z_2^2z_3^2). \end{aligned} \tag{4.5}$$

It is not immediately obvious if 4.5 is nonzero, however, by using the SymPy symbolic algebra package [34], we can row reduce it to the identity matrix, which shows that the matrix has a rank of 5. Since we found a rank 5 submatrix in matrix 4.1, and we know that it has a rank less than 6, the rank of the matrix in Eq. 4.1 is 5. Since the rank is 5, this means that the nullspace is 1 dimensional, and the solution vector in unique up to a scalar.

4.2 Closed form solution

Assuming our assumptions hold, we can derive closed forms that are dependent on the variable Z_3 and by a scaling factor λ.

The full equations are given in the Appendix as equations A.1 and A.4. For the sake of brevity, we rewrite this as

$$
\begin{aligned}
f\mathbf{n}_1 &= \lambda z_3 \frac{\bar{c_1}}{h_3}, \\
f\mathbf{n}_2 &= \lambda z_3 \frac{\overline{c_2}}{h_3}, \\
\mathbf{n}_3 &= \lambda z_3 \frac{\bar{c_3}}{h_3}, \\
z_1 &= \lambda z_3 \bar{c_4} \frac{h_1}{h_3}, \\
z_2 &= \lambda z_3 \bar{c_5} \frac{h_2}{h_3}.
\end{aligned}
\tag{4.6}
$$

We denote the coefficients by

$$
\begin{aligned}
c_1 &= \frac{\bar{c_1}}{h_3}, \\
c_2 &= \frac{\bar{c_2}}{h_3}, \\
c_3 &= \frac{\bar{c_3}}{h_3}, \\
c_4 &= \bar{c_4} \frac{h_1}{h_3}, \\
c_5 &= \bar{c_5} \frac{h_2}{h_3}.
\end{aligned}
\tag{4.7}
$$

From these coefficients, we can derive a formula for the focal length, given by

$$f^2 = \frac{-c_1(c_4(\mathbf{p}^{\text{ankle}}_{x_1} - \mathbf{o}_x) - c_5(\mathbf{p}^{\text{ankle}}_{x_2} - \mathbf{o}_x)) - c_2(c_4(\mathbf{p}^{\text{ankle}}_{y1} - \mathbf{o}_y) - c_5(\mathbf{p}^{\text{ankle}}_{y_2} - \mathbf{o}_y))}{c_3(c_4 - c_5)}. \tag{4.8}$$

4.3 Probabilistic Interpretation

In practice, the assumption that all persons are of the same constant height is easily violated. We seek to model the effects of height variation on the focal length probabilistically. For the set Ω, with sigma algebra $\mathscr{H}$, we define a probability space $(\Omega, \mathscr{H}, \mathbb{P})$. We define random variables to represent the uncertainties. $\theta_k : \Omega \to \mathbb{R}^2$ represents the 2D detections. $\theta_h : \Omega \to \mathbb{R}$ represents the difference in the actual height and the predicted height. $\theta_f : \Omega \to \mathbb{R}^+$ represents the focal length.

4.3.1 Probabilistic Focal Length

We assume that θ_h is distributed normally with mean μ and variance σ^2. Since we need 3 people to solve the DLT matrix, we need 3 i.i.d normal random variables, θ_{h_1}, θ_{h_2}, θ_{h_3}. However, upon rewriting equation 4.8,

$$f^2 = \frac{-\bar{c_1}(\bar{c_4}h_1(\mathbf{p}^{\text{ankle}}_{x_1} - \mathbf{o}_x) - \bar{c_5}h_2(\mathbf{p}^{\text{ankle}}_{x_2} - \mathbf{o}_x)) - \bar{c_2}h_1(\bar{c_4}(\mathbf{p}^{\text{ankle}}_{y1} - \mathbf{o}_y) - \bar{c_5}h_2(\mathbf{p}^{\text{ankle}}_{y_2} - \mathbf{o}_y))}{\bar{c_3}(\bar{c_4}h_1 - \bar{c_5}h_2)}, \tag{4.9}$$

we notice that θ_{h_3} cancels out so we only need two random variables: θ_{h_1}, θ_{h_2}. Each θ_h variable has an induced probability space, namely $(\mathbb{R}, \mathscr{B}(\mathbb{R}), U(E))$, where U is defined by the integral of the pdf of the normal distribution

$$U(E, \mu, \sigma) = \int_E \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx. \tag{4.10}$$

Thus, we can consider equation 4.9 as a function $f^2(\theta_{h_1}, \theta_{h_2}) : \mathbb{R}^2 \to \mathbb{R}$ that induces an image probability space: $(\mathbb{R}, \mathscr{B}(\mathbb{R}), \bar{U}(E))$.

The goal of the next part is to show what the image measure $\bar{U}(E)$ is.

4.3.2 Deriving the Image Measure

Replacing h_1 with θ_{h_1} with h_2 with θ_{h_2}, we get,

$$f^2 = \frac{(-\bar{c}_1\bar{c}_4(\mathbf{p}_{x_1}^{\text{ankle}} - \mathbf{o}_x) - \bar{c}_2\bar{c}_4(\mathbf{p}_{y1}^{\text{ankle}} - \mathbf{o}_y))\theta_{h_1} + (\bar{c}_1\bar{c}_5(\mathbf{p}_{x_2}^{\text{ankle}} - \mathbf{o}_x) + \bar{c}_2\bar{c}_5(\mathbf{p}_{y_2}^{\text{ankle}} - \mathbf{o}_y))\theta_{h_2}}{\bar{c}_3\bar{c}_4\theta_{h_1} - \bar{c}_3\bar{c}_5\theta_{h_2}}. \tag{4.11}$$

Suppose that θ_{h_1} and θ_{h_2} are normal distributions with mean μ and σ^2. We can rewrite Equation 4.11 using multiplication properties of Gaussians to derive

$$f^2 = \frac{\hat{\theta}_{h_1} + \hat{\theta}_{h_2}}{\bar{\theta}_{h_1} + \bar{\theta}_{h_2}}. \tag{4.12}$$

From equation 4.12, we know that $\hat{\theta}_{h_1}$ is a normal distribution with mean $(-\bar{c}_1\bar{c}_4(\mathbf{p}_{x_1}^{\text{ankle}} - \mathbf{o}_x) - \bar{c}_2\bar{c}_4(\mathbf{p}_{y1}^{\text{ankle}} - \mathbf{o}_y))\mu$ and variance $(-\bar{c}_1\bar{c}_4(\mathbf{p}_{x_1}^{\text{ankle}} - \mathbf{o}_x) - \bar{c}_2\bar{c}_4(\mathbf{p}_{y1}^{\text{ankle}} - \mathbf{o}_y))^2\sigma^2$ and $\hat{\theta}_{h_2}$ is a normal distribution with mean $(\bar{c}_1\bar{c}_5(\mathbf{p}_{x_2}^{\text{ankle}} - \mathbf{o}_x) + \bar{c}_2\bar{c}_5(\mathbf{p}_{y_2}^{\text{ankle}} - \mathbf{o}_y))\mu$ and variance $(\bar{c}_1\bar{c}_5(\mathbf{p}_{x_2}^{\text{ankle}} - \mathbf{o}_x) + \bar{c}_2\bar{c}_5(\mathbf{p}_{y_2}^{\text{ankle}} - \mathbf{o}_y))^2\sigma^2$.

We also know that $\bar{\theta}_{h_1}$ is a normal distribution with mean $\bar{c}_3\bar{c}_4\mu$ and variance $\bar{c}_3\bar{c}_4{}^2\sigma^2$. $\bar{\theta}_{h_2}$ is a normal distribution with mean $\bar{c}_3\bar{c}_5\mu$ and variance $\bar{c}_3\bar{c}_5{}^2\sigma^2$.

Using the additional properties of independent Gaussians, we can rewrite equation 4.12 as

$$f^2 = \frac{\mathbf{n}_1}{\mathbf{n}_2}. \tag{4.13}$$

where $\mathbf{n}_1$ is a normal distribution with mean $\mu_{N_1} = (-\bar{c}_1\bar{c}_4(\mathbf{p}_{x_1}^{\text{ankle}} - \mathbf{o}_x) - \bar{c}_2\bar{c}_4(\mathbf{p}_{y1}^{\text{ankle}} - \mathbf{o}_y))\mu + (\bar{c}_1\bar{c}_5(\mathbf{p}_{x_2}^{\text{ankle}} - \mathbf{o}_x) + \bar{c}_2\bar{c}_5(\mathbf{p}_{y_2}^{\text{ankle}} - \mathbf{o}_y))\mu$ and with variance $\sigma_{N_1}^2 = (-\bar{c}_1\bar{c}_4(\mathbf{p}_{x_1}^{\text{ankle}} - \mathbf{o}_x) - \bar{c}_2\bar{c}_4(\mathbf{p}_{y1}^{\text{ankle}} - \mathbf{o}_y))^2\sigma^2 + (\bar{c}_1\bar{c}_5(\mathbf{p}_{x_2}^{\text{ankle}} - \mathbf{o}_x) + \bar{c}_2\bar{c}_5(\mathbf{p}_{y_2}^{\text{ankle}} - \mathbf{o}_y))^2\sigma^2$, and $\mathbf{n}_2$ is a normal distribution with mean $\mu_{N_2} = \bar{c}_3\bar{c}_4\mu + \bar{c}_3\bar{c}_5\mu$ and variance $\sigma_{N_2}^2 = \bar{c}_3\bar{c}_4{}^2\sigma^2 + \bar{c}_3\bar{c}_5{}^2\sigma^2$

4.3.3 Ratio of Two Normal Distributions

We can consider equation 4.13 to be a ratio of two dependent normal distributions. We can compute the covariance of this using $\text{Cov}(aX + bY, cX + dY) = ac\text{Var}(X) + bd\text{Var}(Y) + (bc + ad)\text{Cov}(X, Y)$. We use this formula on equation 4.11 to compute $\text{Cov}(\mathbf{n}_1, \mathbf{n}_2)$. First, we compute the mean of the squared Gaussians. To do so, we use the fact that a squared normal distribution with mean μ and variance σ^2 are equivalent to a noncentral chi-squared distribution with 1 degree of freedom, a non-centrality of $(\frac{\mu}{\sigma})^2$, and a mean of $1+(\frac{\mu}{\sigma})^2$.

Thus, the covariance is given by

$$\mathrm{Cov}(\mathbf{n}_1,\mathbf{n}_2) = ab(1+(\frac{\mu}{\sigma})^2-\mu^2)+cd(1+(\frac{\mu}{\sigma})^2-\mu^2), \tag{4.14}$$

where

$$\begin{aligned} a &= (-\bar{c_1}\bar{c_4}(\mathbf{p}^{\text{ankle}}_{x_1}-\mathbf{o}_x)-\bar{c_2}\bar{c_4}(\mathbf{p}^{\text{ankle}}_{y1}-\mathbf{o}_y)),\\ b &= \bar{c_3}\bar{c_4},\\ c &= (\bar{c_1}\bar{c_5}(\mathbf{p}^{\text{ankle}}_{x_2}-\mathbf{o}_x)+\bar{c_2}\bar{c_5}(\mathbf{p}^{\text{ankle}}_{y_2}-\mathbf{o}_y)),\\ d &= \bar{c_3}\bar{c_5}. \end{aligned} \tag{4.15}$$

The correlation coefficient is simply $\rho = \frac{\mathrm{Cov}(\mathbf{n}_1,\mathbf{n}_2)}{\sqrt{\mathrm{Var}(\mathbf{n}_1)\mathrm{Var}(\mathbf{n}_2)}}$.

4.3.4 Closed-Form Probability Density Function

The pdf of the ratio of two dependent normals is derived by Pham-Gia et al. [39]. Thus, we can express the closed-form PDF using

$$\bar{u}(E) = K_2 \frac{2(1-\rho^2)\sigma_x^2\sigma_y^2}{\sigma_y^2 w^2 - 2\rho\sigma_x\sigma_y w + \sigma_x^2} {}_1F_1(1;\frac{1}{2};\theta_2(w)), \tag{4.16}$$

where

$$\theta_2(w) = \frac{[-\sigma_y^2\mu_x w + \rho\sigma_x\sigma_y(\mu_y w+\mu_x) - \mu_y\sigma_x^2]^2}{2\sigma_x^2\sigma_y^2(1-\rho^2)(\sigma_y^2 w^2 - 2\rho\sigma_x\sigma_y w + \sigma_x^2)}, \tag{4.17}$$

and

$$K_2 = \frac{e^{-\frac{\sigma_y^2\mu_x^2 - 2\rho\sigma_x\sigma_y\mu_x\mu_y + \mu_y^2\sigma_x^2}{2(1-\rho^2)\sigma_x^2\sigma_y^2}}}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}}. \tag{4.18}$$

Since we have a term in the denominator of the form $\sigma_y^2 w^2 - 2\rho\sigma_x\sigma_y w + \sigma_x^2$ where $w \in \mathbb{R}$, we will check to see if there are any discontinuities in the function.

We can find the roots given by

$$\begin{aligned}&\frac{p\sigma_x\sigma_y - \sqrt{-\sigma_x^2\sigma_y^2 + \rho^2\sigma_x^2\sigma_y^2}}{\sigma_y^2},\\ &\frac{p\sigma_x\sigma_y + \sqrt{-\sigma_x^2\sigma_y^2 + \rho^2\sigma_x^2\sigma_y^2}}{\sigma_y^2}.\end{aligned} \tag{4.19}$$

Since $-1 \leq \rho^2 \leq 1$,

This means that $-\sigma_x^2\sigma_y^2 + \rho^2\sigma_x^2\sigma_y^2 < 0$. Thus, the only roots of this polynomial are complex, which shows that the denominator is non-zero for all real numbers, so equation 4.16 is defined on all of $\mathbb{R}$.

4.4 Experiments

The code for the experiments is linked in this GitHub repository. For the experiments, we pick a general set of parameters to generate our scene and we report them for reproducibility. We generate a scene in Python in order to simulate a camera calibration scenario. We generate a scene with normal vector $(0.298, 0.638, -0.710)$, and plane position $(-1.729, -6.409, 1.661)$. We generate a camera with a focal length of 2345.164 pixels. To generate the head positions, we multiply the normal vector by a fixed height of 1.6 meters and add it to each ankle.

We generate three ankles with coordinates $(4.71, -8.88, 2.15)$, $(5.67, -7.29, 3.98)$, and $(6.01, -8.23, 3.27)$.

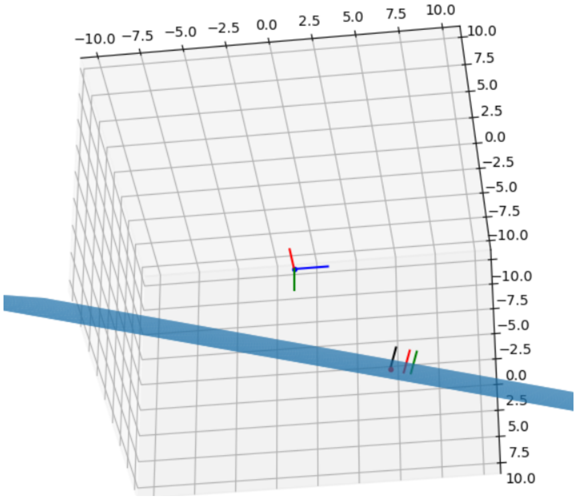


Figure 4.1: Generated scene with the blue plane representing the ground plane, the coordinate system representing the camera, and the 3 lines representing people from ankle to head.

For the experiments, we make the heights in the equation stochastic by sampling from a normal distribution with mean 1.6 and std 0.1. We plot our derived distributions for f in 2 ways. In the first method, we draw 10,000 pairs of samples from the height normal distribution and plot the values of equation 4.8. In the second method, we plot a grid of integers from 1 to 2.5×10^7 and pass them through the PDF we derived as equation 4.16. We compute the mean and standard deviation using numerical integration of our closed-form equation 4.16 with the quadrature method and compare them with the mean and standard deviation of our samples. We plot our results in Figure 4.2, which shows that the derived PDF roughly matches the histogram of random samples.

Method	Mean	Std
Closed-Form	2347.465	361.474
Samples	2365.188	484.407

Table 4.1: The mean and std of the samples compared to the mean and std derived from the closed-form PDF.

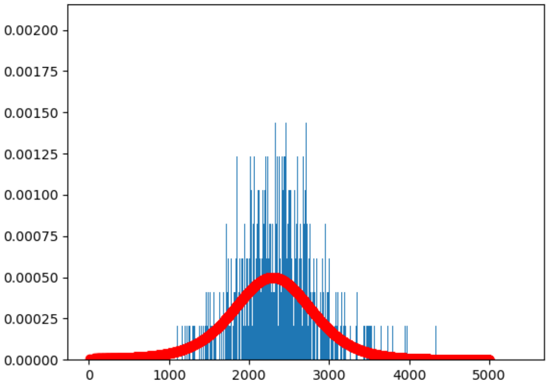


Figure 4.2: The Red curve is the PDF being graphed using the closed-form PDF, and the blue graph is graphed using samples.

In Table 4.1, we can see that the closed-form method and the sample method produce relatively similar results for the mean of the focal length. However, the result obtained through the sampling method has a 34 percent higher standard deviation than the closed-form method. This is most likely caused by the error from numerical integration since the variance is $(E(X^2) - E(X)^2)$, and as a consequence could be less numerically stable since any error in the expectation gets squared.

Chapter 5

Simulations

Following our statistical analysis, we ran trials on simulated data with various levels of noise added. We conduct simulation experiments as in [15], using the same or similar specifications that were given in their simulation study. In our simulated experiments, we seek to test the effects of measurement noise, height variations, and number of people on our single view calibration algorithm, without the effects of detection noise present from pose detectors. This also allows us to get ground truth heights of the people, which we would not have access to in any of the datasets. We generate a scene with a random ground plane and random shoulder and ankle center detections. We specify image dimensions 1920.0 by 1080.0 with focal length $f_X = 960$ and $f_y = 540$, which corresponds to a 90 degree FOV. We run each experiment for 5000 trials. We show an example of our generated data in Figure 5.1. We use the same metrics that were used in Fei et al. [15].

Metrics.

- **Focal error**: the percent focal error is simply computed by $100 \cdot \frac{|f_{\text{gt}} - f_{\text{pred}}|}{f_{\text{gt}}}$. This is represented in the tables as $f_x\%$ and $f_y\%$
- **Normal error**: The degree normal error is the angle difference between the ground truth and predicted normal vectors, represented in the table as $N()$.
- **Position error**: The position error is computed by taking the taking the predicted Euclidean distance from the camera to the ground plane ρ_{pred}, and computing $\frac{|\rho_{\text{pred}} - \rho_{\text{gt}}|}{\rho_{\text{gt}}}$. This is represented by $\rho\%$.
- **Reconstuction error**: The reconstruction error is the percent error between

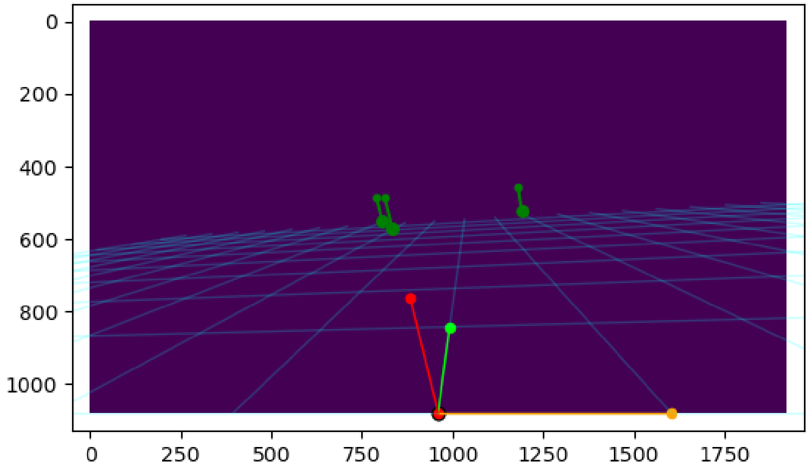


Figure 5.1: Simulation. This shows an example of a scene generated by our simulation where the green dots in the distance represent ankle and shoulder center detections, and the blue lines represent the ground plane.

Euclidean distance between the predicted 3D points and the ground truth 3D points divided by the ground truth distance to the camera. This is represented in the table by $X\%$

- **Failure rate**: Since this could lead to an unsolvable linear system, we record the failure rate which is the percentage of noisy systems that are unsolvable. This is represented as fail%

5.1 Measurement Noise Trials

For these experiments, we fix the height to be 1.7 meters and use 3 pairs of shoulder and ankle center positions. We add a zero mean Gaussian with varying standard deviations to these generated positions. We solve our DLT equations with a height of 1.7 meters. We record our results in Table 5.1. For these trials, adding zero error results in an error for all the metrics that were virtually zero, which shows that our method is implemented correctly and works for perfect data. As the noise standard deviation grows, the error for all metrics increases as expected.

5.2 Height Variation Trials

For these experiments, we fix the detection noise to be sampled from a zero mean Gaussian with a standard deviation of 0.5 and we sample heights from a Gaussian centered at 1.7 meters with varying standard deviations. In our algorithm, we solve the DLT equations with a height equal to 1.7m. We report our results in Table 5.3. As expected, as the standard deviations of the heights increase, the worse the results become since our algorithm uses a fixed height of 1.7 meters.

5.3 Number of People Trials

For these experiments, we fix the detection noise as in the height variation trails, but we sample the height from a Gaussian centered at 1.7 meters with a standard deviation of 0.1 meters. Within our DLT equations, we fix the height to be 1.7 meters like the Gaussian mean. We vary the number of pairs of shoulder and ankle center detection and record our results in Table 5.2. As the number of people increases, the system becomes more and more over determined which reduces the resulting error. However, we note that the x focal length improves more than the y focal length. This is because people are standing up and are oriented more along the y-axis than the x-axis. Therefore, adding more people adds more information along the y-axis than it does along the x-axis.

	Measurement noise std. in pixels					
Error	0.1	0.2	0.5	1.0	2.0	5.0
$f_x\%$	1.86	3.65	6.16	9.88	16.73	27.45
$f_y\%$	0.98	2.15	3.85	5.56	9.67	17.27
N (°)	0.12	0.33	0.52	1.09	1.94	4.10
$\rho\%$	0.25	1.45	1.68	10.65	5.50	14.85
$X\%$	1.00	1.68	3.23	5.78	8.65	16.97
fail%	0.66	1.00	2.4	4.78	8.56	15.94

Table 5.1: Measurement noise. We show the error from our calibration for varying measurement noise standard deviations.

	Number of people				
Error	5	10	20	50	100
$f_x\%$	27.33	27.68	26.29	18.21	21.98
$f_y\%$	32.40	25.64	25.25	14.94	14.17
N (°)	3.74	3.33	2.62	1.76	1.13
$\rho\%$	21.46	21.66	18.92	21.64	24.09
$X\%$	20.96	21.24	20.65	30.07	42.82
fail%	15.98	12.2	8.8	5.72	3.78

Table 5.2: Number of people. We show the error from our calibration for varying numbers of people

	Std. of height in meters				
Error	0.05	0.1	0.15	0.2	0.25
$f_x\%$	7.00	8.75	10.36	10.71	13.22
$f_y\%$	3.64	4.258	5.27	6.04	6.48
N (°)	0.58	0.63	0.82	0.87	1.05
$\rho\%$	6.98	10.388	13.29	16.99	21.43
$X\%$	6.69	10.85	14.60	18.048	21.54
fail%	2.62	3.5	4.18	4.7	5.98

Table 5.3: Height. We show the error from our calibration for varying random heights that are drawn from a Gaussian distribution with varying standard deviations.

Chapter 6

Experiments

For our experiments, we test different components of the pipeline on a variety of datasets including Human3.6m [22], EPFL Terrace and Laboratory [17], and vPTZ [40]. To aid comparisons with future work, we will make the code publicly available. As in Fei et al. [15], we adopt HRNet [45], as implemented by mmpose [9], to obtain 2D keypoint detections and tracking.

Metrics. We use three different metrics to evaluate different components of the camera calibration. The simplest metric is the percent focal error which is computed by taking the absolute difference between the predicted and ground-truth focal and dividing it by the ground truth focal length. Since the coordinate system for the ground truth extrinsic may be different than ours, in order to evaluate the camera pose, we compute the relative rotation and translation from the reference camera to all the other cameras for both the predicted and the ground truth camera systems. Then, we compute the angle in degree of both relative rotations and take the absolute difference to compare them. We do the same for relative translations except first we normalize both camera poses, then we scale them to the same scale as the ground truth camera pose, and finally we take the norm of the difference between the two translations. We find this scale by translating the camera pose to the mean of the camera pose points, and then taking the average distance of the points to the center. For temporal offset, we use one metric; we simply take the absolute difference between the ground truth offset and the predicted offset.

Datasets. We evaluate on the following datasets, each modeling a different setting in terms of the number of people and cameras, and the scale.

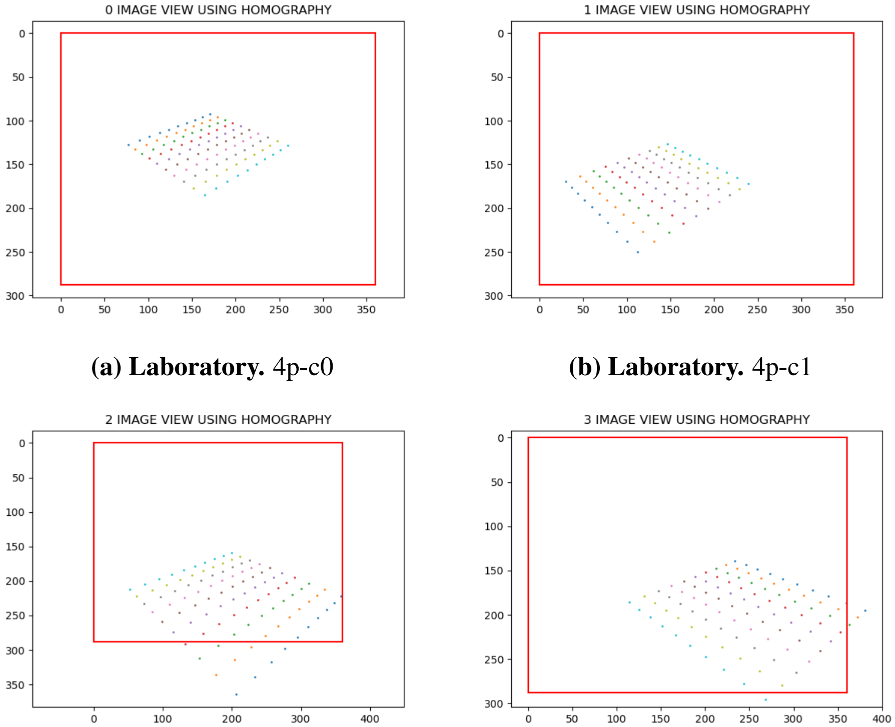
- **Human3.6M** [22] contains 4 cameras temporally synchronized and calibrated cameras that are recorded on a variety of subjects and actions. For our experiments, we use subjects 1,5,6,7,8,9,11 and use the walking action. Each video only contains one subject at a time.
- **Terrace**, **Laboratory**, **and Campus** [17] from EPFL multi-camera pedestrian videos contains 4 temporally synchronized and calibrated cameras filming an outdoor scene with up to 7 subjects (Terrace), an outdoor scene with 3 cameras with 7 subjects (Campus), and a indoor scene with up to 4 subjects (Laboratory). Although the cameras are calibrated, the dataset has two versions of the calibrations. We show how we process these files in the next section.
- **vPTZ** [40] contains 3 outdoor scenes with numerous pedestrians with 4 cameras. Each video is filmed from a fairly high view and is representative of outdoor security camera footage.

6.1 Data Preprocessing

Although the Terrace, Campus, and Laboratory cameras are calibrated, the datasets have two versions of the calibrations. In one of them, instead of the usual intrinsic and extrinsic matrices, the calibration files contain 2 homographies representing the transformation to the head plane and ankle plane. The other one contains intrinsic extrinsic calibrations using the Tsai calibration method [52], however, the calibrations are based on a different-sized image than the ones in the dataset. To rectify this, we use the homographies to create a virtual checkboard on the ground plane as well as the image dimension in the data and use OpenCV to compute the camera matrix. We show examples of the virtual checkerboards in Figure 6.1.

6.2 COLMAP

Before we run our method, we check if publicly available SfM methods can handle the datasets that we use. We use COLMAP [43] on single frames from each camera from each dataset since the method expects a static scene with moving cameras. We



(a) Laboratory. 4p-c0

(b) Laboratory. 4p-c1

(c) Laboratory. 4p-c2

(d) Laboratory. 4p-c3

Figure 6.1: Data Preproccessing. Virtual checkerboards based on the provided ground plane homographies for the EPFL Laboratory sequences.

found that COLMAP is unable to provide reconstructions from only three or four images of a scene. In addition, the backgrounds contain many repeating patterns or are featureless which results in COLMAP matching being unable to produce matches. Thus, we proceed with the experiments for our method using human keypoints instead of the image backgrounds in the next section.

6.3 Single View Experiments

To test the single view calibration stage, we test our method on vPTZ and the EPFL Terrace sequences. We compare against [15] single view calibration method, and their results. In addition, we also compare against the other methods that Fei et al. implemented for testing, including [28], [29], [6], and [48]. We report our numerical results in Table 6.1 and visual results of the ground plane and shoulder reprojections in Figure 6.2. We find that our method is comparable to the related methods in the table, getting an average error of 11 percent of the ground truth focal length. Although our method does not outperform the other methods, we deemed it to be a good enough reproduction to proceed with the main focus of this paper, as we did not have the code with additional implementation details and hyperparameter choices.

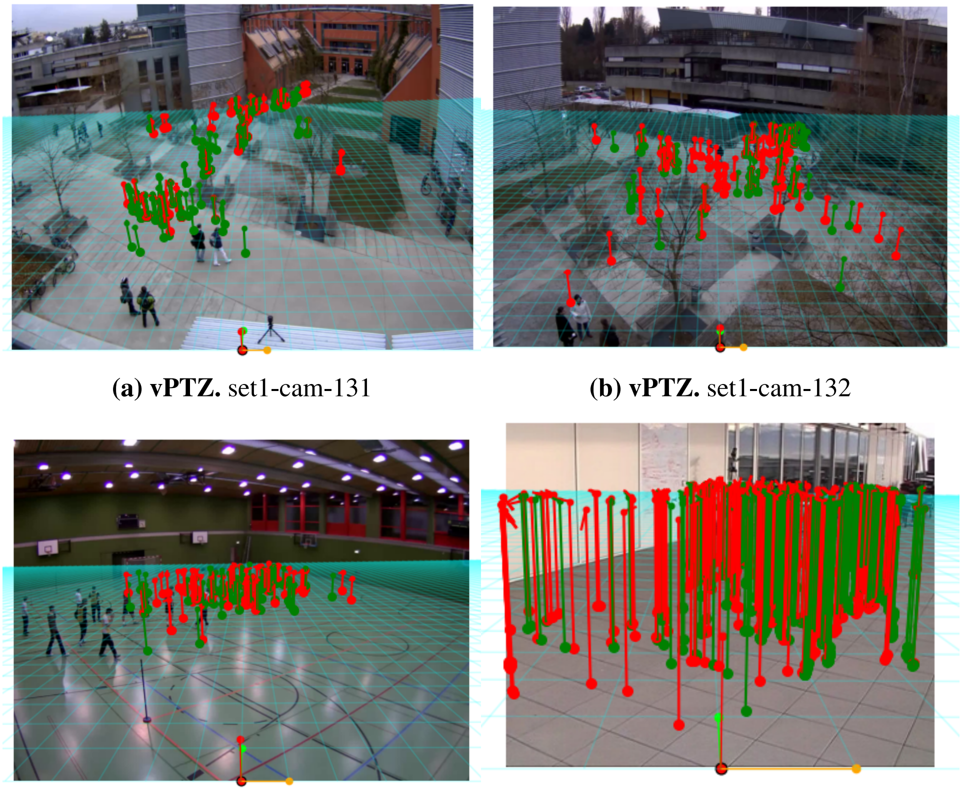
In addition to these experiments, we also evaluate our single view calibration on synthetic data, which is described in detail in the supplementary section.

Method	set1-cam-131	set1-cam-123	set2-cam-132	terrace1-cam0
[28]	1.00	29.00	N/A	N/A
[29]	2.00	19.00	N/A	N/A
[6]	N/A	15.00	24.92	5.33
[48]	N/A	10.14	12.07	1.43
[15]	4.70	0.35	10.74	2.51
Ours	11.31	0.66	18.36	13.18

Table 6.1: Single View Calibration Results. Percent focal error on sequences 1 to 4, corresponding to vPTZ set1-cam-131, vPTZ set1-cam-123, vPTZ set2-cam-132, and Terrace terrace1-cam0.

6.4 Temporal Synchronization Experiments

For comparing against existing synchronization methods, we test our method on Human3.6m. We randomly cut a section of each video that is half the length of the sequence for the Walking sequence for each subject. This is similar to the experiments that Zhang et al. [58] do except they only test their method when the true offset is zero, which is too simplistic because it doesn't test whether the method isn't biased towards zero. For our experiments, we shift the sequence with offsets 0, 50, 100, 150, and 200. We run the experiments using ground truth focal length and the focal length predicted by our method. We show that our method has a mean and median prediction that is close to the true offset, up to an error of 10 frames with a standard deviation of around 10 frames when we use the ground



(a) vPTZ. set1-cam-131

(b) vPTZ. set1-cam-132

(c) vPTZ. set2-cam-132

(d) Terrace. terrace1-cam0

Figure 6.2: Single view calibration results. Visual results of our single view calibration algorithm from table 6.1 with the predicted ground plane represented as a blue grid where every square is one meter. The green and red lines represent reprojected ankle and head detections from the entire sequence. Red represents outlier detections and green represents inlier detections.

	Pred offset (gt f)			**Pred offset (pred f)**			**[58]***
offset gt	mean	median	std	mean	median	std	mean
0	4.0	1.0	10.0	2.90	2.5	14.05	0.0
50	53.93	51	10.05	52.81	52.5	14.12	N/A
100	103.88	101	10.17	102.67	102	14.22	N/A
150	154.5	154	10.15	153.48	153	15.04	N/A
200	204.76	205	10.082	189.26	201	51.70	N/A

Table 6.2: Temporal Synchronization Experiments. This table shows the predicted offset given pairs of videos of Human3.6M with varying offsets as well as their standard deviations. (gt f) means that we run our synchronization using ground truth focal length, and (pred f) means we run our synchronization using predicted focal length. *using GT calibration and not tested on large offsets.

truth focal length. When we use our predicted focal length, we get an error of up to about 11 frames and a standard deviation of 15 frames for most of the experiments. We report our results in Table 6.2 and we also report a more detailed Table A.1 with the results for each individual subject in the appendix. We discuss one failure case happening for large shifts in the limitations section.

For the multi-person case, we perform a similar experiment as with the single-person case, except we shift the sequence with offsets 0, 25, and 50 on the EPFL Terrace and Laboratory sequences using our predicted focal lengths. We report our results in Tables 6.5 and 6.6. For these sequences obtain an error within 5 frames of the ground truth offset.

6.5 Synchronized Bundle Adjustment Experiments

To compare against existing methods requiring synchronized cameras, we test on the EPFL Terrace sequence without introducing temporal shift. We compare against Xu et al. [54] as well as the other methods that Xu et al. tested including SIFT [32], BFM [23], SuperPoint [11], SuperGlue [42], and WxBS [36]. The oracle method in Table 6.3 simply means that they used manual pose annotations. Some methods, such as Sift [32] + BFM [23], use the first method to extract the keypoints, and the second method to match the keypoints. The Sift [32] + BFM

Method	mm	degree
SIFT [32] + BFM [23]	4599	55.03
SuperPoint [11] + [23]	358	54.68
WxBS [36]	1302	54.14
[11] + SuperGlue [42]	9934	36.96
Oracle(Manual-pts)	390	1.18
[54] (Manual-bbox)	308	**0.52**
[54] (ReID-bbox)	308	**0.52**
Ours	**138**	1.82

Table 6.3: Synchronized Bundle Adjustment Experiments. Camera pose error for the Terrace sequence.

[23], Superpoint [11] + BFM [23], and Oracle methods use Xu et al. Geosolver after finding the correspondences. We find that our method gives reasonable reconstructions compared to the other methods, only being outperformed by Xu et al. and the oracle. However, only by a small margin and they both utilize ground truth intrinsics while we use our estimated intrinsics.

6.6 Multiview Offset Experiments

For multiview offset experiments, we test our method using a similar setting as in the Temporal experiments, except we also apply our multiview calibration algorithm afterward in order to analyze the effects of synchronization accuracy on the complete calibration pipeline. As a baseline, we also run our multiview calibration method on the unsynchronized sequences. For these experiments, we do not run our bundle adjustment algorithm since two views provide insufficient constraints under noisy keypoint estimates. For the single-person case, we test it on Human3.6M with 4 cameras for subjects 1,5,6,7,8,9, and 11. We report our results in Table 6.4. We show that although our synchronization method is not perfect, it performs much better than running the multiview calibration algorithm without temporal synchronization.

For the multi-person case, we utilize the EPFL sequences Terrace, Campus,

offset (gt)		0	100	150	200
offset (pred)		5.36	105.21	156.21	191.96
No Sync	°	**5.78**	36.24	42.71	73.04
	m	0.057	0.26	0.46	0.82
	%f	15.06	15.06	15.06	15.06
Sync	°	10.67	**10.58**	**10.99**	**16.89**
	m	0.066	0.065	0.068	0.21
	%f	15.06	15.06	15.06	15.06

Table 6.4: Multiview Offset Experiments for Human3.6M. Camera pose error for unsychronized sequences. No sync means that the algorithm is run without running our synchronization step. Sync means we run our synchronization step. Offset pred means the predicted offset from our synchronization method.

offset (gt)		0	25	50
offset (pred)		4	29	55
No Sync	°	2.73	6.32	13.07
	m	0.073	0.097	0.15
	%f	9.45	9.45	9.45
Sync	°	**2.14**	**2.15**	**2.18**
	m	0.070	0.069	0.069
	%f	9.45	9.45	9.45

Table 6.5: Multiview Offset Experiments for Terrace. Showing that the synchronization significantly improves camera position and angle. Focal length can be estimated from a single view and is not further refined in this experiment.

and Laboratory which contain up to 4, 4, and 7 people respectively, and 4, 3, and 4 cameras respectively. We proceed with our experiments in a similar manner to the experiments using pairwise cameras, however, we use timesteps 0, 25, and 50. We report our results in Table 6.5, Table 6.6, and Table 6.7. Like in the single-person case, we show that our synchronization method results in better performance than running the multiview calibration algorithm without temporal synchronization.

offset (gt)		0	25	50
offset (pred)		1.67	29	55.33
No Sync	°	0.81	10.02	44.10
	m	1.15	1.16	1.15
	%f	15.63	15.63	15.63
Sync	°	**0.51**	**0.61**	**0.81**
	m	1.15	1.15	1.15
	%f	15.63	15.63	15.63

Table 6.6: Multiview Offset Experiments for the Laboratory sequence. Improvements on this indoor sequence are consistent with the outdoor terrace sequence in Tab. 6.5.

offset (gt)		0	25	50
offset (pred)		8.89	26.89	41.33
No Sync	°	10.34	12.69	13.39
	m	0.095	0.12	0.13
	%f	14.55	14.55	14.55
Sync	°	**9.17**	**9.50**	**10.31**
	m	0.088	0.090	0.096
	%f	14.55	14.55	14.55

Table 6.7: Multiview Offset Experiments for the Campus sequence. Improvements on this indoor sequence are consistent with the outdoor terrace sequence in Tab. 6.5 and indoor sequence in Tab. 6.6

Chapter 7

Limitations and Discussion

Although we show that our method works well on most scenes, in certain extreme configurations, some of the steps fail and subsequent steps cannot recover due to the assumption that the initialization from the previous step is within error bounds. In future work, we plan to detect such errors and deploy learning-based solutions to overcome them.

7.1 Bundle Adjustment

For these experiments, we perform similar experiments as in section 6.6, however, we perform bundle adjustment after our initialization, and in addition to using pairs of cameras, we also use 3 or more cameras. For the case with 3 or more cameras, we fix one camera to be the reference camera, and then for the remaining sync cameras, we offset one of them, and keep the others synchronized. We report our results in Tables 7.1, 7.3, 7.2, and 7.4. Like in the pairwise case, our algorithm improves when we utilize time synchronization. With the exception of the Terrace sequence with 0 offset with 3 cameras, all the results are worse than before we run bundle adjustment, especially with the camera angle error.

For sparse views, such as our pairwise camera experiments, our bundle adjustment does not have enough constraints under noisy keypoint estimates. Triggs et al. [50] recommends taking a large range of views that are 30 to 40 degrees apart for bundle adjustment with self-calibration. However, this recommendation is used for flat compact objects, whereas humans are more complex objects. Thus, more care is needed for optimizing the pose such as utilizing bone length constraints and symmetry constraints. In addition to these limitations, we also have errors that accumulate from the 2D pose detection, as well as in our single view calibration step, since our focal length error can be as high as 18 percent. We show simulation results to investigate the results of detection error in the calibration step in the supplemental. Another major limitation is the fact that we assume a constant height among the persons in the scene. This could cause problems with the bundle adjustment since we enforce a constant height constraint in the objective function.

7.2 Failure Case: Periodic Motion

In Human36m walking sequences, the people are walking in a circle. This causes the distance curves for 2 views to have a periodic shape which means that for every n frame, the curve repeats itself. This can be problematic for the time synchronization since this would mean multiple offsets can give a similar result.

Our algorithm's time synchronization module fails on Subject 11 when we set the offset to 200 because in Figure 7.1, we note that the distance curves have a period of about 200 frames, which results in our error curve having two very similar local minimums at around 0 and 200. However, such a large misalignment paired with harmonic repetition is unusual in practice. In future work, the local motion, such as the articulation of arms could be used to further disambiguate frames. However, this is non-trivial as occlusions and different viewing angles have to be considered.

offset (gt)		0	100	150	200
offset (pred)		5.36	105.21	156.21	191.96
2 cameras (°) (m) (%f)	Pre Bundle (No Sync)	5.78 0.057 15.06	36.24 0.26 15.06	42.71 0.46 15.06	73.04 0.82 15.06
	Bundle (No Sync)	9.00 0.15 14.92	39.38 0.42 14.90	43.25 0.50 15.08	64.72 0.80 15.17
	Pre Bundle (Sync)	10.67 0.066 15.06	10.58 0.065 15.06	10.99 0.068 15.06	16.89 0.21 15.06
	Bundle (Sync)	13.09 0.17 14.93	12.95 0.18 14.93	14.77 0.16 14.90	19.45 0.18 14.97
4 cameras (°) (m) (%f)	Pre Bundle (No Sync)	3.02 0.90 15.41	15.25 3.13 15.41	17.28 4.05 15.41	27.37 5.11 15.41
	Bundle (No Sync)	19.97 1.75 15.40	19.17 3.38 15.43	27.30 4.13 15.45	34.56 5.38 15.44
	Pre Bundle (Sync)	4.70 0.97 15.47	4.68 0.97 15.42	4.84 0.99 15.37	6.92 1.31 15.36
	Bundle (Sync)	14.66 1.56 15.47	15.47 1.62 15.42	17.81 1.61 15.37	20.37 1.98 15.36

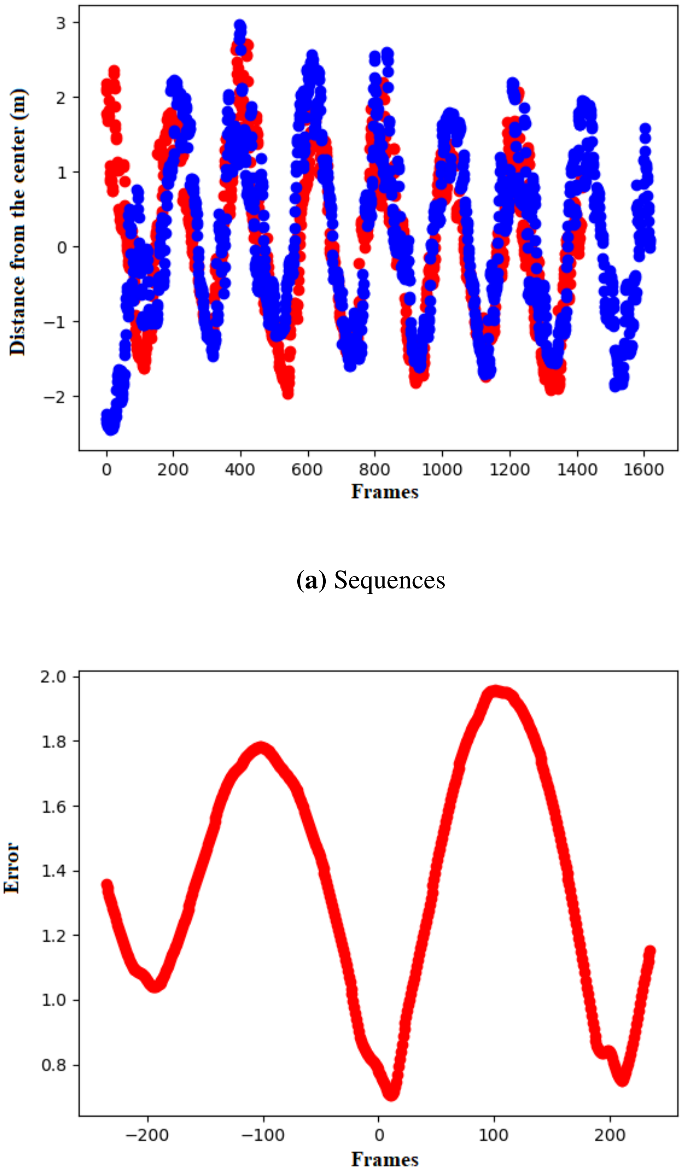
Table 7.1: Multiview Offset Bundle Adjustment Experiments for Human3.6M. Camera pose error for unsychronized sequences. No sync means that the algorithm is run without running our synchronization step. Sync means we run our synchronization step. Offset pred means the predicted offset from our synchronization method. We report the

offset (gt)		0	25	50
offset (pred)		4	29	55
2 cameras (°) (m) (%f)	Pre Bundle (No Sync)	2.73 0.073 9.45	6.32 0.097 9.45	13.07 0.15 9.45
	Bundle (No Sync)	5.90 0.12 10.27	7.53 0.14 11.52	9.08 0.11 11.08
	Pre Bundle (Sync)	2.14 0.070 9.45	2.15 0.069 9.45	2.18 0.069 9.45
	Bundle (Sync)	2.49 0.12 11.02	8.53 0.097 11.12	5.89 0.13 10.85
4 cameras (°) (m) (%f)	Pre Bundle (No Sync)	2.87 1.11 9.13	4.67 1.25 9.13	8.11 1.55 9.13
	Bundle (No Sync)	1.82 1.38 9.97	4.84 1.13 9.44	16.90 2.94 10.26
	Pre Bundle (Sync)	2.57 1.097 9.13	2.58 1.09 9.13	2.67 1.096 9.13
	Bundle (Sync)	3.37 1.62 9.77	3.61 1.52 9.98	2.91 1.55 11.10

Table 7.2: Multiview Offset Bundle Adjustment Experiments for Terrace

offset (gt)		0	25	50
offset (pred)		1.67	29	55.33
2 cameras (°) (m) (%f)	Pre Bundle (No Sync)	0.81 1.15 15.63	10.02 1.16 15.63	44.10, 1.15 15.63
	Bundle (No Sync)	10.96 1.14 16.92	5.58 1.17 16.50	32.41 1.20 16.97
	Pre Bundle (Sync)	0.51 1.15 15.63	0.61 1.15 15.63	0.81 1.15 15.63
	Bundle (Sync)	9.88 1.14 16.88	18.65 1.12 18.27	16.99 1.09 17.30
3 cameras (°) (m) (%f)	Pre Bundle (No Sync)	0.79 0.36 18.21	4.52 0.40 18.21	15.75 0.68 18.21
	Bundle (No Sync)	1.53 0.45 19.57	5.08 1.46 20.39	10.18 1.39 20.61
	Pre Bundle (Sync)	0.71 0.36 18.21	1.39 0.37 18.21	1.42 0.37 18.21
	Bundle (Sync)	1.40 0.45 19.63	3.25 1.91 19.88	3.22 1.36 20.06

Table 7.3: Multiview Offset Bundle Adjustment Experiments for the Laboratory sequence.



(a) Sequences

(b) Error curve

Figure 7.1: Periodic motion. Due to the period of 200 frames, there are 2 very similar local minimums, one at 0, and one at 200.

offset (gt)		0	25	50
offset (pred)		8.888889	26.88889	41.33333
2 cameras (°) (m) (%f)	Pre Bundle (No Sync)	10.34 0.095 14.55	12.69 0.12 14.55	13.39 0.13 14.55
	Bundle (No Sync)	12.45 0.091 14.26	16.73 0.16 14.66	15.73 0.15 14.51
	Pre Bundle (Sync)	9.17 0.088 14.55	9.50 0.090 14.55	10.31 0.096 14.55
	Bundle (Sync)	12.45 0.10 14.20	12.51 0.11 14.30	12.79 0.12 14.23
4 cameras (°) (m) (%f)	Pre Bundle (No Sync)	26.89 1.94 1.90	25.63 1.54 1.90	24.16 1.37 1.90
	Bundle (No Sync)	29.48 1.86 1.95	29.87 1.33 1.95	27.69 1.27 1.97
	Pre Bundle (Sync)	27.37 2.23 1.90	27.64 2.20 1.90	28.32 2.11 1.90
	Bundle (Sync)	32.46 2.21 1.71	30.82 2.19 1.89	29.03 2.08 1.89

Table 7.4: Multiview Offset Bundle Adjustment Experiments for the Campus sequence.

Chapter 8

Conclusion

We designed, implemented, and open-sourced a method to calibrate multiple cameras, even when videos are out of sync. It enables new application domains for 3D vision, where reconstructions require the precision of multi-view but the ease of recording with consumer-grade cameras without hardware synchronization capabilities. Our future works will focus on improving the bundle adjustment step for sparse views in addition to substituting some of our optimization-based steps with learning-based solutions.

8.1 Future Work

In order to address the limitations of our pipeline, future works could focus on improving the bundle adjustment step for sparse views, substituting some of our optimization-based steps with other learning-based solutions, and improving our single view method with theoretical and practical models.

8.1.1 Improving the bundle adjustment step

As discussed in the limitations, the bundle adjustment step needs further work. From the related works, Liu et al. [30] perform bundle adjustment on human poses from multiple viewpoints, thus in principle, it should be possible to optimize the camera pose with bundle adjustment. As stated in the limitations, further aspects that could be improved include additional constraints on the objective function, such as adding bone length constraints, and the usage of person tracking and person re-identification to ease the finding of correspondences with multi-person sequences. We could also make use of second-order optimizers if the solution is ill-conditioned [12]. Additionally, we could try optimizing the extrinsic parameters using projected gradient descent [53] to constrain the optimization so the results are always valid rotation matrices.

8.1.2 Learning Based Solutions

Multiview Reconstruction

One method that makes use of deep learning is demonstrated in Ajisafe et al. [1], which uses a single view but with a mirror in the scene to act as a rudimentary 2nd view. Then, Using the 2 views, they reconstruct the 3D pose using optimization methods. Finally, they pass the bone orientation information obtained in the optimization to a Neural Radiance Fields (NeRF) model [35] along with the image mask of the person to reconstruct a volumetric body model. They train the method in a weakly-supervised fashion using the 2D image data so they do not need any 3D ground truth. We could potentially replace the mirror with a 2nd camera view, and use our CasCalib pipeline as an initialization for the NeRF reconstruction. In fact, they already use our single-view calibration method to obtain the camera parameters in the first step..

Synchronization

Although we have shown that our temporal synchronization method can roughly align two or more video sequences, we have yet to achieve precise fine alignment. As stated in the related works, Ling et al. [33] achieves finer precision using a two-stage weakly-supervised deep learning pipeline, it requires synchronized training data, and is trained on a narrow range of datasets. It also does not take into account 3D information. Thus, one extension could be to combine our single view calibration method to obtain 3D information, and then incorporate that into Ling et al. This could improve the trajectory estimation since estimating 2D trajectory doesn't take into account the depth of tracks which could cause dissimilar tracks to appear similar in image coordinates.

Intrinsics

As stated in the related works, Grabner et al. [18] use a focal length predictor within a Faster/Mask R-CNN framework. Although we stated that the fact they train on a specific dataset with common objects like chairs and sofas with consistent dimensions, one research direction could be training a method like this on a dataset containing persons. In addition, we could impose geometric constraints on human poses to further constrain the optimization.

One major limitation of our method is that we assume that there is no lens distortion. In practice, there is often lens distortion, such as radial distortion in the camera. Li et al. [27], estimate various types of lens distortion using neural networks, however, they train the network in a supervised fashion, which results in them having to have a large labeled dataset. We would seek to estimate lens distortion through self-supervised means, since specific datasets may not generalize to real-world scenes.

8.1.3 Improving Single View Calibration

In this section, we discuss how the single-view calibration step can be further improved.

Optimal Height Assignment

One key insight that this work has shown is that even small variations in height can lead to huge variations in the focal length. In Table 4.1, the input height data for method 1 had a standard deviation of 0.1 meters, but running this method leads to a standard deviation of up to 479.773 pixels for a focal length of 2345.164 pixels. This shows that assuming that all people in the scene are the same height may not be realistic. One direction that could be explored is to assign different heights to different people in the scene and see if an optimal height assignment improves the predicted focal length. We can also use clues in the scene to aid us in assigning optimal heights; for example, when people are close to each other or standing on the same image line, we can see their relative heights to each other. If we included people tracking, and people walking around the scene, there may be sufficient information to infer each person's relative height. This application would additionally need re-identification and tracking in order to assign heights consistently to people in the scene.

Modeling other sources of uncertainty

In this work, we only analyzed the focal length. However, the depth and the normal vector also play an important role. One assumption that is often violated is the assumption that everyone is standing straight up. If we were to relax this assumption, then we would have to model each person as an independent vector in the scene. This would be an interesting problem since the random variable in this case would be in terms of angles and rotations. We could model each rotation as an element of the group of 3D rotations $SO(3)$, and define a Haar measure to compute the probabilities. Gwak et al. [19] have shown that the geometric properties of groups that describe rigid body motion can be directly applied to camera calibration.

Additionally, another direction that could be explored is to model the 2D detections stochastically. In practice, there is usually noise in the detection of 2D points. This would increase the complexity of interpreting this problem probabilistically, but this model would be closer to in-the-wild settings.

Deriving a closed form expectation

As shown in the methods section, we were able to derive a PDF for the focal length squared. However, what is yet to be determined is an analytic method to compute the expectation and variance. If a closed-form expression for the expectation exists, then it would allow for accurate computation without the need for error-prone numerical methods.

8.2 Societal Impact

Methods that automatically calibrate cameras based on human poses have a wide range of applications such as human body reconstruction. This could benefit people in other fields such as neuroscientists studying body motions of patients with neurological disorders, since they may not be experienced in calibrating cameras using traditional methods. However, one possible misuse of this technology is reconstructing humans in new undesirable poses. Furthermore, this technology can be used on cameras that are filming without their consent. We acknowledge that unethical applications should be discouraged. We also used datasets that contain people who gave explicit permission to be filmed [17, 22, 40].

Another usage of camera calibration is in distance and velocity estimation. This can have some usage with social distancing enforcement during pandemic times. This could also be used for fall and accident detection since, sudden changes in velocities could be an indication of a fall, which could be useful in care homes for the elderly. Another application is estimating the dimensions of areas that could have applications in architecture and construction. However, such technology could be employed on security cameras, which could encourage the proliferation of security cameras which could in many cases be a violation of personal privacy. Although methods such as blurring people's faces or simply not saving recorded data can be safeguards against privacy violations, in practice there is no way to guarantee that the filming party will have privacy as one of their imperatives.

Distance estimation is also used in range-finding, which is a set of techniques to estimate the distance from the observer to a target. This is often done in construction to determine the dimensions of areas. However, range finders that use lasers and cameras have many applications for precision weapons. In Liu et al. [31], they discuss combining a self-calibrating camera with a laser range finder with application to sniper rifles. An article by Horus Vision, LCC [41], gives guidelines on the estimation of range using human height for police and military use. Although we state our preference for peaceful usages, we acknowledge these uses, which often remain out of our control and at the whim of the ever-changing political landscape, to be possible. We only hope that our leaders and communities around the globe use their better judgment and choose to build a kinder and more peaceful world.

Bibliography

[1] D. Ajisafe, J. Tang, S.-Y. Su, B. Wandt, and H. Rhodin. Mirror-aware neural humans. *arXiv preprint arXiv:2309.04750*, 2023. → page 55

[2] P. Besl and N. McKay. A method for registration of 3-d shapes, ieee t. pattern anal., 14, 239–256, 1992. → page 4

[3] E. Brachmann, T. Cavallari, and V. A. Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5044–5053, June 2023. → page 6

[4] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 690–696. IEEE, 2000. → page 2

[5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993. → page 8

[6] G. Brouwers, M. Zwemer, R. Wijnhoven, and P. With. Automatic calibration of stationary surveillance cameras in the wild. In *Computer Vision – ECCV 2016 Workshops*, volume 9914, 10 2016. ISBN 978-3-319-48880-6. doi:10.1007/978-3-319-48881-3_52. → page 41

[7] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2019. → page 7

[8] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. Bagautdinov, L. Lettry, P. Fua, L. Van Gool, and F. Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In *Proceedings of the*

IEEE conference on computer vision and pattern recognition, pages 5030–5039, 2018. → page 2

[9] M. Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. → page 38

[10] H. A. Correia and J. H. Brito. 3d reconstruction of human bodies from single-view and multi-view images: A systematic review. *Computer Methods and Programs in Biomedicine*, 239:107620, 2023. ISSN 0169-2607. doi:https://doi.org/10.1016/j.cmpb.2023.107620. URL https://www.sciencedirect.com/science/article/pii/S0169260723002857. → page 1

[11] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018. → pages 43, 44

[12] N. Doikov, M. Jaggi, et al. Second-order optimization with lazy hessians. In *International Conference on Machine Learning*, pages 8138–8161. PMLR, 2023. → page 55

[13] N. Eichler, H. Hel-Or, and I. Shimshoni. Spatio-temporal calibration of multiple kinect cameras using 3d human pose. *Sensors*, 22(22), 2022. ISSN 1424-8220. doi:10.3390/s22228900. URL https://www.mdpi.com/1424-8220/22/22/8900. → page 8

[14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996. → page 19

[15] X. Fei, H. Wang, X. Zeng, L. L. Cheong, M. Wang, and J. Tighe. Single view physical distance estimation using human pose, 2021. → pages 2, 4, 5, 6, 11, 34, 38, 40, 41

[16] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. ISSN 0001-0782. doi:10.1145/358669.358692. URL https://doi.org/10.1145/358669.358692. → pages 6, 7, 14

[17] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2008. doi:10.1109/TPAMI.2007.1174. → pages 7, 38, 39, 58

[18] A. Grabner, P. M. Roth, and V. Lepetit. Gp2c: Geometric projection parameter consensus for joint 3d pose and focal length estimation in the wild, 2019. → pages 5, 8, 56

[19] S. Gwak, J. Kim, and F. Park. Numerical optimization on the euclidean group with applications to camera calibration. *IEEE Transactions on Robotics and Automation*, 19(1):65–74, 2003. doi:10.1109/TRA.2002.807530. → page 57

[20] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. → page 1

[21] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. → page 8

[22] C. Ionescu, D. Papava, V. Olaru, and C. 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, jul 2014. → pages 38, 39, 58

[23] A. Jakubovic and J. Velagic. Image feature matching and object detection using brute-force matchers. pages 83–86, 09 2018. doi:10.23919/ELMAR.2018.8534641. → pages 43, 44

[24] O. Javed, S. Khan, Z. Rasheed, and M. Shah. Camera handoff: tracking in multiple uncalibrated stationary cameras. In *Proceedings Workshop on Human Motion*, pages 113–118, 2000. doi:10.1109/HUMO.2000.897380. → pages 5, 7

[25] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. → pages 7, 19

[26] S.-E. Lee, K. Shibata, S. Nonaka, S. Nobuhara, and K. Nishino. Extrinsic camera calibration from a moving person. *IEEE Robotics and Automation Letters*, 7(4):10344–10351, 2022. doi:10.1109/LRA.2022.3192629. → pages 5, 7

[27] X. Li, B. Zhang, P. V. Sander, and J. Liao. Blind geometric distortion correction on images through deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. → page 56

[28] J. Liu, R. Collins, and Y. Liu. Surveillance camera autocalibration based on pedestrian height distributions. In *BMVC 2011 - Proceedings of the British Machine Vision Conference 2011*, pages 117.1–117.11, 01 2011. ISBN 1-901725-43-X. doi:10.5244/C.25.117. → page 41

[29] J. Liu, R. Collins, and Y. Liu. Robust autocalibration for a surveillance camera network. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, pages 433–440, 01 2013. ISBN 978-1-4673-5053-2. doi:10.1109/WACV.2013.6475051. → page 41

[30] K. Liu, L. Chen, L. Xie, J. Yin, S. Gan, Y. Yan, and E. Yin. Auto calibration of multi-camera system for human pose estimation. *IET Computer Vision*, 16(7):607–618, 2022. doi:https://doi.org/10.1049/cvi2.12130. URL https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cvi2.12130. → pages 2, 5, 6, 7, 54

[31] Z. Liu, D. Lu, W. Qian, G. Gu, J. Zhang, and X. Kong. Extrinsic calibration of a single-point laser rangefinder and single camera. *Optical and Quantum Electronics*, 51:1–13, 2019. → page 58

[32] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, nov 2004. ISSN 0920-5691. doi:10.1023/B:VISI.0000029664.99615.94. URL https://doi.org/10.1023/B:VISI.0000029664.99615.94. → pages 43, 44

[33] L. Mei, Y. He, F. J. Fishani, Y. Yu, L. Zhang, and H. Rhodin. Learning domain-adaptive landmark detection-based self-supervised video synchronization for remote sensing panorama. *Remote Sensing*, 15(4), 2023. ISSN 2072-4292. doi:10.3390/rs15040953. URL https://www.mdpi.com/2072-4292/15/4/953. → pages 5, 9, 55

[34] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, Jan. 2017. ISSN 2376-5992. doi:10.7717/peerj-cs.103. URL https://doi.org/10.7717/peerj-cs.103. → page 26

[35] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. → page 55

[36] D. Mishkin, J. Matas, M. Perdoch, and K. Lenc. Wxbs: Wide baseline stereo generalizations, 2015. → pages 43, 44

[37] P. Natarajan, P. K. Atrey, and M. Kankanhalli. Multi-camera coordination and control in surveillance systems: A survey. *ACM Trans. Multimedia Comput. Commun. Appl.*, 11(4), jun 2015. ISSN 1551-6857. doi:10.1145/2710128. URL https://doi.org/10.1145/2710128. → page 1

[38] P. Nogueira. Motion capture fundamentals. In *Doctoral Symposium in Informatics Engineering*, volume 303, 2011. → page 1

[39] T. Pham-Gia, N. Turkkan, and E. Marchand. Density of the ratio of two normal random variables and applications. *Communications in Statistics-Theory and Methods*, 35(9):1569–1591, 2006. → page 30

[40] H. Possegger, M. Rüther, S. Sternig, T. Mauthner, M. Klopschitz, P. M. Roth, and H. Bischof. Unsupervised Calibration of Camera Networks and Virtual PTZ Cameras. In *Proc. Computer Vision Winter Workshop (CVWW)*, 2012. → pages 38, 39, 58

[41] R. H. A. E. RANGES. Mission statement horus vision is dedicated to providing the rifleman the tools to yield the highest probability of a first round hit at extended ranges. 2006. → page 58

[42] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. → pages 43, 44

[43] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. → pages 5, 6, 39

[44] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. → page 2

[45] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation, 2019. → page 38

[46] I. E. Sutherland. Three-dimensional data input by tablet. *Proceedings of the IEEE*, 62(4):453–461, 1974. → pages 6, 11

[47] 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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1856–18567, 2018. doi:10.1109/CVPRW.2018.00230. → pages 2, 5, 7

[48] Z. Tang, Y.-S. Lin, K.-H. Lee, J.-N. Hwang, and J.-H. Chuang. Esther: Joint camera self-calibration and automatic radial distortion correction from tracking of walking humans. *IEEE Access*, 7:10754–10766, 2019. doi:10.1109/ACCESS.2019.2891224. → pages 4, 41

[49] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3d reconstruction on mobile phones. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013. → page 1

[50] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, page 298–372, Berlin, Heidelberg, 1999. Springer-Verlag. ISBN 3540679731. → pages 4, 21, 47

[51] A. M. Truong, W. Philips, N. Deligiannis, L. Abrahamyan, and J. Guan. Automatic multi-camera extrinsic parameter calibration based on pedestrian torsors †. *Sensors*, 19(22), 2019. ISSN 1424-8220. doi:10.3390/s19224989. URL https://www.mdpi.com/1424-8220/19/22/4989. → pages 5, 7

[52] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987. doi:10.1109/JRA.1987.1087109. → page 39

[53] T. Vu and R. Raich. On asymptotic linear convergence of projected gradient descent for constrained least squares. *IEEE Transactions on Signal Processing*, 70:4061–4076, 2022. → page 55

[54] Y. Xu, Y.-J. Li, X. Weng, and K. Kitani. Wide-baseline multi-camera calibration using person re-identification, 2021. → pages 2, 5, 7, 43, 44

[55] C. Zhang, F. Rameau, J. Kim, D. M. Argaw, J.-C. Bazin, and I. S. Kweon. Deepptz: Deep self-calibration for ptz cameras. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020. → pages 5, 8

[56] Q. Zhang and A. B. Chan. Single-frame based deep view synchronization for unsynchronized multi-camera surveillance, 2022. → pages 5, 8

[57] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152, 1994. → pages 4, 21

[58] Z. Zhang, C. Wang, and W. Qin. Semantically synchronizing multiple-camera systems with human pose estimation. *Sensors*, 21(7), 2021. ISSN 1424-8220. doi:10.3390/s21072464. URL https://www.mdpi.com/1424-8220/21/7/2464. → pages 2, 5, 7, 8, 41, 43

[59] H. Zhou and H. Hu. Human motion tracking for rehabilitation—a survey. *Biomedical signal processing and control*, 3(1):1–18, 2008. → page 1

Appendix A

Supporting Materials

A.1 Temporal Synchronization Experiments

In this section, we show our results for temporal synchronization experiments on Human3.6M for each subject. We report our results in Table A.1.

A.2 Statistical Analysis Equations

We will put long equations in this section, which we derived using the SymPy symbolic algebra package in Python.

A.2.1 Closed form solution to the DLT equation

$$f\mathbf{n}_1 = \lambda z_3 \frac{\bar{n}_{\text{num}_1}}{\bar{n}_{\text{den}_1}} \tag{A.1}$$

Subject	Shift (gt)	Shift (gt focal)	Shift (pred focal)
S1	0	11.24	11.82
S1	50	54.10	54.25
S1	100	104.10	104.25
S1	150	153.72	154.28
S1	200	204.18	204.28
S5	0	12.84	14.997
S5	50	55.49	55.68
S5	100	105.38	105.48
S5	150	155.38	155.28
S5	200	205.25	204.97
S6	0	13.94	21.75
S6	50	56.08	60.03
S6	100	106.14	110.18
S6	150	156.41	166.31
S6	200	206.51	215.91
S7	0	14.14	13.22
S7	50	56.77	54.34
S7	100	106.90	104.28
S7	150	157.24	154.8
S7	200	206.93	204.90
S8	0	15.71	14.46
S8	50	58.82	58.42
S8	100	108.58	106.79
S8	150	158.82	158.67
S8	200	208.92	208.67
S9	0	3.37	10.73
S9	50	52.63	52.15
S9	100	101.06	102.31
S9	150	150.39	152.98
S9	200	201.20	204.02
S11	0	9.22	12.71
S11	50	57.17	54.16
S11	100	107.58	104.02
S11	150	159.12	154.18
S11	200	209.28	70.31

Table A.1: Temporal Synchronization Experiments. We report our results for each subject for Human3.6M. GT shift represents the ground truth offset, while GT focal represents running our method with ground truth focal length, and Pred focal represents running our method with the focal length predicted by our method.

where,

$$\begin{aligned}\bar{n}_{\mathrm{num}_1} = (&-\mathbf{o}_x\mathbf{p}^a_{x1}\mathbf{p}^a_{y2}\mathbf{p}^a_{y3}+\mathbf{o}_x\mathbf{p}^a_{x1}\mathbf{p}^a_{y2}\mathbf{p}^s_{y3}+\mathbf{o}_x\mathbf{p}^a_{x1}\mathbf{p}^a_{y3}\mathbf{p}^s_{y2}-\mathbf{o}_x\mathbf{p}^a_{x1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}+\mathbf{o}_x\mathbf{p}^a_{x2}\mathbf{p}^a_{y1}\mathbf{p}^a_{y3}-\mathbf{o}_x\mathbf{p}^a_{x2}\mathbf{p}^a_{y1}\mathbf{p}^s_{y3}-\\
&\mathbf{o}_x\mathbf{p}^a_{x2}\mathbf{p}^a_{y3}\mathbf{p}^s_{y1}+\mathbf{o}_x\mathbf{p}^a_{x2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y3}+\mathbf{o}_x\mathbf{p}^s_{x1}\mathbf{p}^a_{y2}\mathbf{p}^a_{y3}-\mathbf{o}_x\mathbf{p}^s_{x1}\mathbf{p}^a_{y2}\mathbf{p}^s_{y3}-\mathbf{o}_x\mathbf{p}^s_{x1}\mathbf{p}^a_{y3}\mathbf{p}^s_{y2}+\mathbf{o}_x\mathbf{p}^s_{x1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}-\\
&\mathbf{o}_x\mathbf{p}^s_{x2}\mathbf{p}^a_{y1}\mathbf{p}^a_{y3}+\mathbf{o}_x\mathbf{p}^s_{x2}\mathbf{p}^a_{y1}\mathbf{p}^s_{y3}+\mathbf{o}_x\mathbf{p}^s_{x2}\mathbf{p}^a_{y3}\mathbf{p}^s_{y1}-\mathbf{o}_x\mathbf{p}^s_{x2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y3}+\mathbf{p}^a_{x1}\mathbf{p}^a_{x2}\mathbf{p}^a_{y3}\mathbf{p}^s_{y1}-\mathbf{p}^a_{x1}\mathbf{p}^a_{x2}\mathbf{p}^a_{y3}\mathbf{p}^s_{y2}-\\
&\mathbf{p}^a_{x1}\mathbf{p}^a_{x2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y3}+\mathbf{p}^a_{x1}\mathbf{p}^a_{x2}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}+\mathbf{p}^a_{x1}\mathbf{p}^s_{x2}\mathbf{p}^a_{y2}\mathbf{p}^a_{y3}-\mathbf{p}^a_{x1}\mathbf{p}^s_{x2}\mathbf{p}^a_{y2}\mathbf{p}^s_{y3}-\mathbf{p}^a_{x1}\mathbf{p}^s_{x2}\mathbf{p}^a_{y3}\mathbf{p}^s_{y1}+\mathbf{p}^a_{x1}\mathbf{p}^s_{x2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y3}-\\
&\mathbf{p}^a_{x2}\mathbf{p}^s_{x1}\mathbf{p}^a_{y1}\mathbf{p}^a_{y3}+\mathbf{p}^a_{x2}\mathbf{p}^s_{x1}\mathbf{p}^a_{y1}\mathbf{p}^s_{y3}+\mathbf{p}^a_{x2}\mathbf{p}^s_{x1}\mathbf{p}^a_{y3}\mathbf{p}^s_{y2}-\mathbf{p}^a_{x2}\mathbf{p}^s_{x1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}+\mathbf{p}^s_{x1}\mathbf{p}^s_{x2}\mathbf{p}^a_{y1}\mathbf{p}^a_{y3}-\mathbf{p}^s_{x1}\mathbf{p}^s_{x2}\mathbf{p}^a_{y1}\mathbf{p}^s_{y3}-\\
&\mathbf{p}^s_{x1}\mathbf{p}^s_{x2}\mathbf{p}^a_{y2}\mathbf{p}^a_{y3}+\mathbf{p}^s_{x1}\mathbf{p}^s_{x2}\mathbf{p}^a_{y2}\mathbf{p}^s_{y3})\end{aligned}\tag{A.2}$$

and

$$\begin{aligned}\bar{n}_{\mathrm{den}_1} = (&h\mathbf{p}^a_{x1}\mathbf{p}^a_{y2}\mathbf{p}^s_{y1}-h\mathbf{p}^a_{x1}\mathbf{p}^a_{y2}\mathbf{p}^s_{y3}-h\mathbf{p}^a_{x1}\mathbf{p}^s_{y1}\mathbf{p}^s_{y2}+h\mathbf{p}^a_{x1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}-\\
&h\mathbf{p}^a_{x2}\mathbf{p}^a_{y1}\mathbf{p}^s_{y2}+h\mathbf{p}^a_{x2}\mathbf{p}^a_{y1}\mathbf{p}^s_{y3}+h\mathbf{p}^a_{x2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y2}-h\mathbf{p}^a_{x2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y3}-\\
&h\mathbf{p}^s_{x1}\mathbf{p}^a_{y1}\mathbf{p}^a_{y2}+h\mathbf{p}^s_{x1}\mathbf{p}^a_{y1}\mathbf{p}^s_{y2}+h\mathbf{p}^s_{x1}\mathbf{p}^a_{y2}\mathbf{p}^s_{y3}-h\mathbf{p}^s_{x1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}+\\
&h\mathbf{p}^s_{x2}\mathbf{p}^a_{y1}\mathbf{p}^a_{y2}-h\mathbf{p}^s_{x2}\mathbf{p}^a_{y1}\mathbf{p}^s_{y3}-h\mathbf{p}^s_{x2}\mathbf{p}^a_{y2}\mathbf{p}^s_{y1}+h\mathbf{p}^s_{x2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y3})\end{aligned}\tag{A.3}$$

$$f\mathbf{n}_2 = \lambda z_3\frac{\bar{n}_{\mathrm{num}_2}}{\bar{n}_{\mathrm{den}_2}}\tag{A.4}$$

where,

$$\begin{aligned}\bar{n}_{\mathrm{num}_2} = \lambda z_3((&-\mathbf{o}_y\mathbf{p}^a_{x1}\mathbf{p}^a_{y2}\mathbf{p}^a_{y3}+\mathbf{o}_y\mathbf{p}^a_{x1}\mathbf{p}^a_{y2}\mathbf{p}^s_{y3}+\mathbf{o}_y\mathbf{p}^a_{x1}\mathbf{p}^a_{y3}\mathbf{p}^s_{y2}-\mathbf{o}_y\mathbf{p}^a_{x1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}+\mathbf{o}_y\mathbf{p}^a_{x2}\mathbf{p}^a_{y1}\mathbf{p}^a_{y3}-\mathbf{o}_y\mathbf{p}^a_{x2}\mathbf{p}^a_{y1}\mathbf{p}^s_{y3}-\\
&\mathbf{o}_y\mathbf{p}^a_{x2}\mathbf{p}^a_{y3}\mathbf{p}^s_{y1}+\mathbf{o}_y\mathbf{p}^a_{x2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y3}+\mathbf{o}_y\mathbf{p}^s_{x1}\mathbf{p}^a_{y2}\mathbf{p}^a_{y3}-\mathbf{o}_y\mathbf{p}^s_{x1}\mathbf{p}^a_{y2}\mathbf{p}^s_{y3}-\mathbf{o}_y\mathbf{p}^s_{x1}\mathbf{p}^a_{y3}\mathbf{p}^s_{y2}+\mathbf{o}_y\mathbf{p}^s_{x1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}-\\
&\mathbf{o}_y\mathbf{p}^s_{x2}\mathbf{p}^a_{y1}\mathbf{p}^a_{y3}+\mathbf{o}_y\mathbf{p}^s_{x2}\mathbf{p}^a_{y1}\mathbf{p}^s_{y3}+\mathbf{o}_y\mathbf{p}^s_{x2}\mathbf{p}^a_{y3}\mathbf{p}^s_{y1}-\mathbf{o}_y\mathbf{p}^s_{x2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y3}+\mathbf{p}^a_{x1}\mathbf{p}^a_{y2}\mathbf{p}^a_{y3}\mathbf{p}^s_{y1}-\mathbf{p}^a_{x1}\mathbf{p}^a_{y2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y3}-\\
&\mathbf{p}^a_{x1}\mathbf{p}^a_{y3}\mathbf{p}^s_{y1}\mathbf{p}^s_{y2}+\mathbf{p}^a_{x1}\mathbf{p}^s_{y1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}-\mathbf{p}^a_{x2}\mathbf{p}^a_{y1}\mathbf{p}^a_{y3}\mathbf{p}^s_{y2}+\mathbf{p}^a_{x2}\mathbf{p}^a_{y1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}+\mathbf{p}^a_{x2}\mathbf{p}^a_{y3}\mathbf{p}^s_{y1}\mathbf{p}^s_{y2}-\mathbf{p}^a_{x2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}-\\
&\mathbf{p}^s_{x1}\mathbf{p}^a_{y1}\mathbf{p}^a_{y2}\mathbf{p}^a_{y3}+\mathbf{p}^s_{x1}\mathbf{p}^a_{y1}\mathbf{p}^a_{y2}\mathbf{p}^s_{y3}+\mathbf{p}^s_{x1}\mathbf{p}^a_{y1}\mathbf{p}^a_{y3}\mathbf{p}^s_{y2}-\mathbf{p}^s_{x1}\mathbf{p}^a_{y1}\mathbf{p}^s_{y2}\mathbf{p}^s_{y3}+\mathbf{p}^s_{x2}\mathbf{p}^a_{y1}\mathbf{p}^a_{y2}\mathbf{p}^a_{y3}-\mathbf{p}^s_{x2}\mathbf{p}^a_{y1}\mathbf{p}^a_{y2}\mathbf{p}^s_{y3}-\\
&\mathbf{p}^s_{x2}\mathbf{p}^a_{y2}\mathbf{p}^a_{y3}\mathbf{p}^s_{y1}+\mathbf{p}^s_{x2}\mathbf{p}^a_{y2}\mathbf{p}^s_{y1}\mathbf{p}^s_{y3})\end{aligned}\tag{A.5}$$

$$\begin{aligned}\bar{n}_{\mathrm{den}_2} = (&h\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{a}\mathbf{p}_{y1}^{s} - h\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} - h\mathbf{p}_{x1}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y2}^{s} + h\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} - h\mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{s} + h\mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} + h\mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y2}^{s} -\\ &h\mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s} - h\mathbf{p}_{x1}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{a} + h\mathbf{p}_{x1}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{s} + h\mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} - h\mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} + h\mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{a} - h\mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} - h\mathbf{p}_{x2}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y1}^{s} +\\ &h\mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s}))\end{aligned} \tag{A.6}$$

$$\begin{aligned}\mathbf{n}_3 = \lambda z_3((&\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{a} - \mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x1}^{a}\mathbf{p}_{y3}^{a}\mathbf{p}_{y2}^{s} + \mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{a} + \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} + \mathbf{p}_{x2}^{a}\mathbf{p}_{y3}^{a}\mathbf{p}_{y1}^{s}\\ &-\mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{a} + \mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} +\\ &\mathbf{p}_{x1}^{s}\mathbf{p}_{y3}^{a}\mathbf{p}_{y2}^{s} - \mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} + \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{a} - \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x2}^{s}\mathbf{p}_{y3}^{a}\mathbf{p}_{y1}^{s} + \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s})/(h\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{a}\mathbf{p}_{y1}^{s} -\\ &h\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} - h\mathbf{p}_{x1}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y2}^{s} + h\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} - h\mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{s} + h\mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} + h\mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y2}^{s} -\\ &h\mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s} - h\mathbf{p}_{x1}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{a} + h\mathbf{p}_{x1}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{s} + h\mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} - h\mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} + h\mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{a} -\\ &h\mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} - h\mathbf{p}_{x2}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y1}^{s} + h\mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s}))\end{aligned} \tag{A.7}$$

$$\begin{aligned}z_1 = \lambda z_3((&-\mathbf{p}_{x2}^{a}\mathbf{p}_{y3}^{a}\mathbf{p}_{y1}^{s} + \mathbf{p}_{x2}^{a}\mathbf{p}_{y3}^{a}\mathbf{p}_{y2}^{s} + \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x2}^{a}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} + \mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{a} - \mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x1}^{s}\mathbf{p}_{y3}^{a}\mathbf{p}_{y2}^{s} +\\ &\mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x2}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{a} + \mathbf{p}_{x2}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} + \mathbf{p}_{x2}^{s}\mathbf{p}_{y3}^{a}\mathbf{p}_{y1}^{s} - \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s})/(\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{a}\mathbf{p}_{y1}^{s} - \mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x1}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y2}^{s} +\\ &\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{s} + \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} + \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y2}^{s} - \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x1}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{a} + \mathbf{p}_{x1}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{s} + \mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} -\\ &\mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} + \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{a} - \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x2}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y1}^{s} + \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s}))\end{aligned} \tag{A.8}$$

$$\begin{aligned}z_2 = \lambda z_3((&-\mathbf{p}_{x1}^{a}\mathbf{p}_{y3}^{a}\mathbf{p}_{y1}^{s} + \mathbf{p}_{x1}^{a}\mathbf{p}_{y3}^{a}\mathbf{p}_{y2}^{s} + \mathbf{p}_{x1}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} + \mathbf{p}_{x1}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{a} - \mathbf{p}_{x1}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x1}^{s}\mathbf{p}_{y3}^{a}\mathbf{p}_{y2}^{s} +\\ &\mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{a} + \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} + \mathbf{p}_{x2}^{s}\mathbf{p}_{y3}^{a}\mathbf{p}_{y1}^{s} - \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s})/(\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{a}\mathbf{p}_{y1}^{s} - \mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x1}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y2}^{s} +\\ &\mathbf{p}_{x1}^{a}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{s} + \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} + \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y2}^{s} - \mathbf{p}_{x2}^{a}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x1}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{a} + \mathbf{p}_{x1}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{s} + \mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y3}^{s} -\\ &\mathbf{p}_{x1}^{s}\mathbf{p}_{y2}^{s}\mathbf{p}_{y3}^{s} + \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y2}^{a} - \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{a}\mathbf{p}_{y3}^{s} - \mathbf{p}_{x2}^{s}\mathbf{p}_{y2}^{a}\mathbf{p}_{y1}^{s} + \mathbf{p}_{x2}^{s}\mathbf{p}_{y1}^{s}\mathbf{p}_{y3}^{s}))\end{aligned} \tag{A.9}$$