Extrinsic Camera Calibration using Human-pose Estimations and Automatic Re-identification

Willem Jan Tempelaar

MSc in Mechanical Engineering

Extrinsic Camera Calibration using Human-pose Estimations and Automatic Re-identification

by

Willem Jan Tempelaar

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday August 29, 2022 at 14:00 PM.

Student number:	4378385	
Project duration:	August 8, 2021 – August 29, 2022	
Thesis committee:	Dr. J.F.P. Kooij,	TU Delft, supervisor
	Dr. H. Caesar,	TU Delft
	MSc. M. Hennipman,	Siemens Moblity, supervisor

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

Multi-pedestrian tracking in camera networks has gained enormous interest in the industry because of its applicability in travel-flow analysis, autonomous driving, and surveillance. Essential to tracking in camera networks is camera calibration and, in particular extrinsic camera calibration. Extrinsic camera calibration incorporates the 3D position and orientation of the cameras in the camera network. Current methods for extrinsic calibration require special operators to place calibration objects in sight of all cameras in the network, which is impractical and limits the ease of tracking in camera networks.
In this thesis, an automatic extrinsic calibration model is proposed to calibrate the extrinsic camera parameters from the image data of all cameras in a network. The proposed method utilizes deep learning algorithms for feature extraction and matching. The feature extraction step is a human-pose estimator, extracting the key points of humans such as joints, eyes, and feet. The matching algorithm is a re-ID algorithm using an affinity-based feature extractor.
Two camera network datasets have been used to evaluate the proposed model. A four-camera fully overlapping dataset SALSA [1], and a more challenging seven-camera partially overlapping dataset WildTrack [6]. The calibration accuracy of the model on both datasets is calculated by comparing the ground truth value with the calibrated extrinsic camera parameters. On dataset WildTrack, the model could calibrate three of the seven cameras, whereas the model had a near-perfect calibration on dataset SALSA. Dataset SALSA achieved a root mean squared error in translation of 0.02 meters and 0.0 radians in orientation, compared to the ground truth values.
The final product of this research is an automated extrinsic camera calibration model that eases the camera calibration in camera networks. The proposed model could not calibrate all datasets, but the model provides a baseline upon which future research can be done.

Acknowledgements

Throughout my thesis, I have been guided by my supervisors, Julian Kooij and Marco Hennipman. I would like to thank them for offering me the resources necessary to complete this project and for their advice to keep an eye on the academic value of my work.
In addition, I would like to thank everyone at Siemens Mobility and, in particular, the team at Digital Lab: Emilio Tuinenburg, Florian Hutten, Danny Meringa, Alessandra Sternberg, Philline Dikker, and Daniel van Gelder; for giving me feedback during the sprint reviews.

I especially want to thank Folkert de Ronde for studying with me the entire year at the library. Our conversations during the regular walks across the campus contributed to the final result. Lastly, I want to thank the rest of my friends and family and my girlfriend, Laura de Bruijn, for offering their continued support during my thesis.

Willem Jan Tempelaar
Delft, August 2022

Nomenclature

Abbreviations

CCTV	Closed-circuit television
DLT	Direct linear transform
DOF	Degree of freedom
FOV	Field of view
fps	Frames per second
GPU	Graphical processing unit
mAP	Mean average precision
MPT	Multi-pedestrian tracking
NMS	Non-maximum suppression
OSNet	Omni-scale network
R-CNN	Region base convolutional neural network
Re-ID	Re-identification
RGB	Red-green-blue
RMPE	Regional multi-person pose estimation
RMS	Root mean squared
SALSA	Synergetic social scene analysis
SIFT	Scale-invariant feature transform
SSD	Single shot multi-box detector
SVD	Singular value decomposition
YOLO	You only look once

Symbols

$\mathbf{E}$	Essential matrix
$\mathbf{e}$	Epipole
$\mathbf{F}$	Fundamental matrix
$\mathbf{H}$	Human-pose 3D point
$\mathbf{h}$	Human-pose 2D point
$\mathbf{l}$	Epipolar line
$\mathbf{N}$	Affinity feature vector
$\mathbf{R}$	Rotation matrix

$\mathbf{S}$	Corresponding point pairs
$\mathbf{t}$	Translation vector
$\mathbf{x}$	Corresponding point
π	Epipolar plane
θ	Threshold parameter between 0 and 1
C	Camera
c	Confidence parameter
E	Reprojection error
F	Value for frame skip
s	Scale
u	Horizontal pixel coordinate
v	Vertical pixel coordinate

Contents

1 Introduction

Multi-Pedestrian Tracking (MPT) is a computer vision task that detects and tracks pedestrians over RGB video streams. It has gained interest from the industry in the last decade because it has a multitude of applications, for example, surveillance, autonomous driving algorithms, and traveler flow analysis.

This research was initiated by Siemens Mobility stating the objective: “Design a real-time multi-camera MPT algorithm used for crowd-density estimation and travel-flow analysis”. The motivation for this objective originated from current solutions in tracking. The current hardware used for tracking is top-down stereo sensors, using RGB data and depth, while in parallel to these top-down camera sensors, CCTV cameras survey the same scene. Moving this automated tracking task from stereo sensors to an already available infrastructure of CCTV sensors could drop costs by easing the implementation.
Thanks to the advancement in MPT, numerous trackers exist solving the multi-camera MPT, for example, the works of [7], [13]. These trackers use a fast object detection algorithm for detecting and tracking the pedestrians in 2D pixel locations for all cameras in the network. Tracks of these pedestrians are then transformed into real-world coordinates so that the data association step of the tracker could assign global labels to the pedestrians. Transformation of 2D pixel coordinates into real-world coordinates requires a homography matrix, and this is the camera calibration information and must be available for multi-camera multi-person tracking [7]. An example of perfect camera calibration is given in Figure 1.1.

Camera calibration information consists of the intrinsic and extrinsic camera parameters. The difference between intrinsic and extrinsic camera calibration parameters is those intrinsic camera parameters are camera specific and independent of position in the world, whereas extrinsic camera parameters are not. Intrinsic parameters are focal length, center point, and distortion parameters. Extrinsic camera parameters are the position and orientation of the camera in the real world. Calibrating the extrinsic camera parameters is, therefore, more challenging because it requires calibration on site, done by a trained operator [19]. The operator creates feature points in the camera network that are easily recognizable on the images. Later, the operator matches the pixel coordinates of all viewing angles regarding the created feature points, called correspondence points. Another camera calibration challenge occurs when the camera’s orientation or position changes. Even small changes in the extrinsic parameters require recalibration of the entire camera network, which is highly impractical [27]. These two characteristics lower the ease of adoption of MPT-tracking algorithms in CCTV camera networks.
A solution to these limitations are automated calibration models such as: [26], [19], [27]. These auto-calibration models use the intrinsic camera parameters and the RGB-video streams to determine the extrinsic camera parameters and the human-pose estimation for correspondence points. Human pose estimation determines the 2D pixel locations of human-pose key points, such as limbs, eyes, hands, and feet (Figure 1.2).
The works of [26], [19] used a single human in a controlled environment for their calibrations. Having a single human in the camera network makes the calibration significantly easier because all detected human-pose key points belong to the same person, mitigating the feature point matching step in the typical calibration techniques.

Figure 1.1: The public dataset WildTrack [6] is shown here with its calibration precision visualized. The two images on the left display the initialization points depicted in blue. These two points are projected to a single 3D point using the camera calibration information and triangulation. The four images on the right show the same object (the nose of a shoe) but from varying viewing angles. When the 3D point is back-projected to 2D pixel coordinates (using the camera calibration information of that camera angle) it shows the 2D point on the same object, only in other viewing angles. These back-projected points are shown in red.

Figure 1.2: The human-pose estimation visualized on the dataset WildTrack [6] using the algorithm: AlphaPose [9].

The authors of [27] were able to create a robust automatic calibration model in an uncontrolled environment; the authors proposed a method by finding the center line of the pedestrians from the human-pose estimation step. Matching the center lines happened using a brute-force algorithm comparing the tracks of the pedestrians. Matching these tracks by this brute-force algorithm was only possible when the tracks of the pedestrians were long. Otherwise, it could not calculate a stable extrinsic calibration.

Both the works of [26] and [19] address the extension of their method to calibration using multi-person human-pose estimations, creating more correspondence points that cover a more significant part of

Figure 1.3: Re-identification across multiple datasets using OSnet [34]

the environment, therefore, increasing the accuracy of the extrinsic calibration [26] [19].
The matching step is problematic when using multi-person human-pose estimations for the calibration. The authors of [34] proposed an Omni-scale re-identification algorithm, OSNet-AIN, which extracts distinctive re-ID features from pedestrians' bounding boxes. OSNet-AIN is trained on multiple datasets, and due to its architecture, it could do accurate re-identification on datasets it has not trained on (cross-domain re-ID). Its ability to match pedestrians could solve the feature matching step in human-pose-based automated calibration models. An example of OSNet-AIN is given in Figure 1.3.

1.1. Goal

The objective stated by Siemens was to design an MPT algorithm for camera networks. However, the introduction described the difficulty of implementing CCTV-based tracking in camera networks due to camera calibration. This research proposes an automatic extrinsic calibration model to ease the implementation of CCTV-based tracking in camera networks. This research aims to investigate the characteristics of the proposed auto-calibration model.

1.2. Research questions

The main research question is: *How accurate is an automated extrinsic camera calibration model, using the human-pose estimation as features extractor and an automatic re-identification algorithm when comparing the extrinsic camera parameters to the annotated calibration information?*

This research question can be divided into three sub-questions:

- When assuming that re-identification across frames is correct, how does a single-person human-pose estimation versus a multi-person human-pose estimation affect the calibration accuracy and is the accuracy affected by the number of human-pose key points?
- In automatic re-identification, what is the effect of evaluating each subsequent frame versus skipping frames on the calibration accuracy, and how are mismatching errors handled?
- Does automated re-identification negatively affect the calibration accuracy compared with calibration using annotated re-identification?

1.3. Contributions

This research's main contributions to the camera calibration field are summarized below.

- This research improved human-pose-based calibration by expanding the single-person calibration to multi-person calibration. Increasing the calibration accuracy of these models.
- A re-ID algorithm for a camera network is proposed, using an existing affinity-based feature extractor. This re-ID algorithm can handle mismatching errors, thanks to its temporal filter.
- The first automatic multi-person full human-pose calibration model is proposed in this research, achieving near-perfect calibration accuracy on dataset SALSA.

1.4. Outlook

This research assignment is structured as follows, Chapter 2 discusses the related work, finds the research gaps, and ends with the contributions of this research. In Chapter 3, the methodology of this research is described. This chapter provides an extensive overview of the model's architecture in the automated calibration model and the experimental design. Chapter 4 describes the results and discusses the implications and limitations of the experiments from the methodology chapter. Chapter 5 will conclude this research by answering each research question and presenting directions for future work in the recommendations.

2

Related work

In this chapter, the literature for this research is presented. This chapter elaborates on the conventional ways of camera calibration and the automatic calibration models with their sub-components. Section 2.1 the conventional method for camera calibration is described. Section 2.2 explains the automated extrinsic calibration models.
The sub-components of the proposed auto-calibration model from Section 1.1, are explained in object detection in Section 2.3, the human-pose estimator in Section 2.4, and the automatic re-ID is described in Section 2.5.
Section 2.6 examines public datasets of camera networks with (partially) overlapping fields of view.
In Section 2.7 the found research gaps in the literature are discussed.

2.1. Conventional calibration

The most commonly known method for calibrating a network's intrinsic and extrinsic camera parameters is Zhang's method [32]. This method requires skilled operators that show at least two orientations of a planar pattern to the camera's field of view. An example of a planar pattern is a checkerboard with known dimensions, shown in Figure 2.1. After this calibration procedure, both intrinsic and extrinsic camera parameters are retrieved.
One major downside to this technique is the need for a skilled operator to be on-site for the calibration or when the cameras have to be re-calibrated.

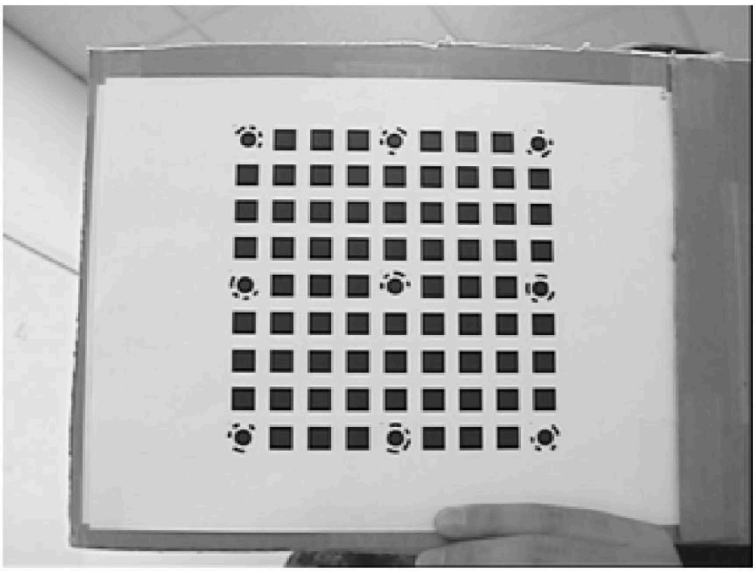

Figure 2.1: The checkered pattern used in the works of [32].

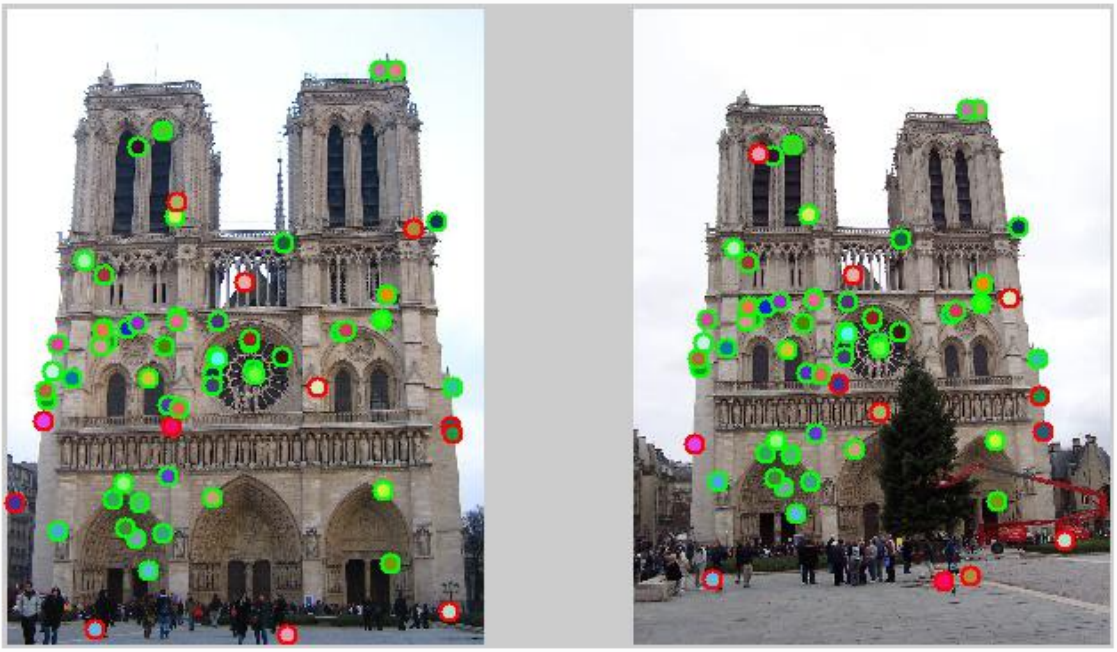

Figure 2.2: SIFT feature matching of the Notre Dame [25]

2.2. Auto-calibration models

Scale-invariant feature transform (SIFT) [18] is an algorithm capable of extracting features, with a difference-from-Gaussian function, from images and is widely used in automatic calibration. Correspondence points are created by matching features of two frames from two different camera angles. Extrinsic camera parameters are calculated using these correspondence points. A disadvantage of using SIFT feature matching is the limitation of a small baseline between two cameras. The SIFT feature matching is only possible when the baseline between the two cameras is relatively small. A small baseline between cameras creates almost the same image (Figure 2.2). With the slight variation in images, it is possible to find the same features in the image by SIFT feature matching and calculate the extrinsic camera parameters.
Other auto-calibration methods use temporal signatures to compute the extrinsic camera parameters using epipolar geometry. These temporal signatures are extracted from the motion of detected objects in the 2D pixel coordinates. The works of: [3], [4], [10] all used these temporal signatures, called motion barcodes. Motion barcodes test the epipolar geometry in the scene, and these methods omit the use of re-identification methods in their approach, which is a huge advantage. One downside of using auto-calibration models with motion barcodes is that they cannot be deployed in planar scenarios. Motion barcodes rely on the depth of moving objects to find epipolar lines. Limiting this method only to non-planar scenarios.

The last category in automated calibration is human-pose based calibration [20], [26], [19]. These methods are comparable with SIFT, whereas SIFT extracts features with the difference-of-Gaussian function, using the extracted key points from the humans-pose estimator. These key points are a human's joints, eyes, hands, and feet. The matching step of this auto-calibration model in these papers [20], [26], [19] is non-existent because they all use datasets containing a single human in a controlled environment. These methods gained interest because the accuracy of human-pose estimation has increased massively with the use of deep-learning algorithms.
The last auto-calibration category uses the human-pose estimation as the primary source for features but processes it as a vertical stick. This work [27] tracks these vertical sticks and matches them when they are long enough. The matching step uses a brute-force method, matching all tracks of vertical sticks and using them for calibration. The best calibration with the lowest relative re-projection error is chosen as the valid calibration.

2.3. Object detection

Due to the surge in deep-learning-based algorithms, object detectors have increased their performance significantly, leading to remarkable breakthroughs in object detection [35]. Faster R-CNN [23] was the first object detector that neared real-time speeds and achieved state-of-the-art accuracy. It uses two separate networks for object detection, the region proposal network suggesting a location of interest and an image classifier classifying the location of interest. Combining these two steps explains why they are called double-staged object detectors.
The other category in object detection is the single-staged object detector. Two of the most popular detectors in this category are: YOLO [22], and SSD [16]. Both algorithms skip the region proposal network and use handcrafted locations of interest, like anchors [21]. Dropping the region proposal network leads to faster than real-time detection. An example of the output of an object detector is shown in Figure 2.3.

2.4. Human-pose estimators

Human-pose estimation is extracting human key points from bounding box coordinates on the image. These bounding box coordinates come from the object detector stage. The current state-of-the-art in the field of human-pose estimation is the deep-learning algorithm AlphaPose [9]. The works of AlphaPose achieved its best mean average precision (mAP) when the object detector Faster R-CNN, based on ResNet-152 [12], was used as the human detector, in combination with the pose estimation algorithm PyraNet [30]. It achieved state-of-the-art results through its regional multi-person pose estimation framework (RMPE).
This RMPE framework is later improved by the authors of [29]. It added novel techniques such as PoseFlow builder and PoseFlow NMS to the RMPE framework. These additions made it able to track the human-pose estimation over multiple frames. The overall human-pose tracker is called PoseFlow and significantly outperforms the state-of-the-art on human-pose tracking datasets [29].

2.5. Re-identification algorithm

Person re-identification (Re-ID) is a retrieval problem, finding a person with the same features across different non-overlapping datasets. Due to the increasing demand for public safety and an increasing number of surveillance systems, the research on person re-ID has surged [31]. Unfortunately, real-person re-ID is still not solved and has its challenges, and one of the largest ones is the large domain gap between re-ID datasets. Re-identification algorithms are deep-learning algorithms, which makes the performance of these algorithms dependent on their training regime. The problem with these re-ID training datasets is that they are too varied in luminescence, background, and viewing angle, resulting in re-ID algorithms overfitting their training data [34].
OSNet-AIN is one of the first re-ID algorithms capable of learning domain-generalized features, mitigating the effect of overfitting. It learns these generalizable features by cross-dataset learning. The training regime of OSNet-AIN is pre-trained on the image classifying dataset imageNet [8], and after this, it is fine-tuned on three other source datasets. The three re-ID datasets are: Duke [24], Market1501 [33], CUHK03 [15] for training and tested on the unseen dataset MSMT17 [28] as validation [34]. OSNet-AIN reaches state-of-the-art results on cross-domain re-ID, is exceptionally lightweight, and is open-source code.

2.6. Public datasets

Public dataset SALSA [1] is a dataset created to study free-standing conversational groups, analyzing in a multi-modal approach. Four low-resolution, static surveillance cameras collect video data, and each participant wears a badge that includes an accelerometer, Bluetooth, a microphone, and infrared sensors [1]. The dataset contains two scenarios, a cocktail party and a poster session (Figure 2.4), in a room of approximately 100 m^2, with 18 participants. Both scenarios are 50 minutes long, and the intrinsic and extrinsic camera parameters are annotated.

WildTrack [6] is a dataset using seven static surveillance cameras on a public square in Zurich, Switzerland (Figure 2.5). The authors of WildTrack created this dataset because, in real-world scenarios, camera networks often have overlapping fields of view, and the authors found a lack of good quality

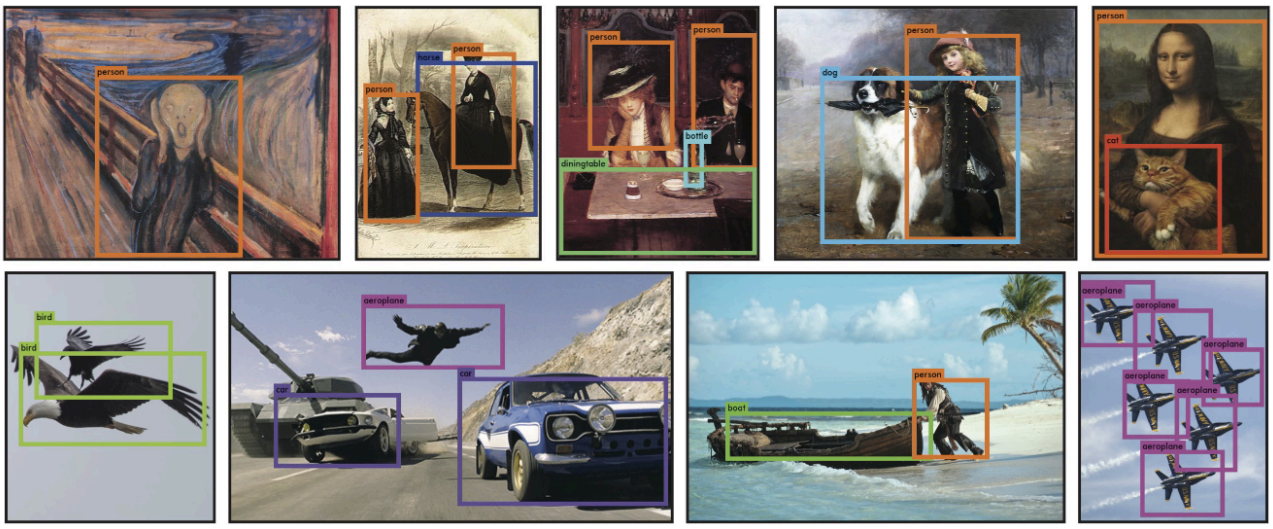

Figure 2.3: From the original YOLO paper [22]. The YOLO object detector runs on simple artwork and images from the internet. There is one error in classification, the second image from the bottom row classifies a person as an airplane.

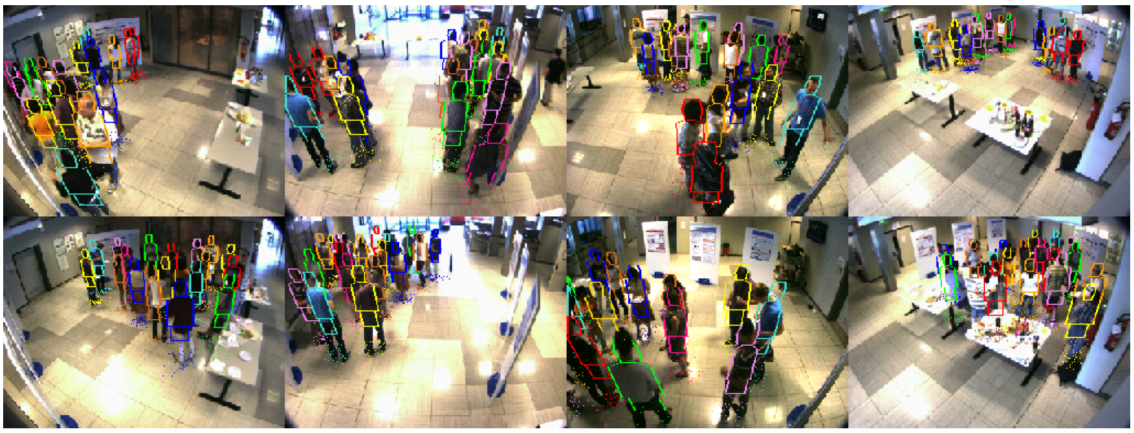

Figure 2.4: Dataset SALSA [1], displaying both the poster session (top four images) and the cocktail party (bottom four images). Around the participants are boxes drawn by HJS-PF tracking.

datasets that have this quality. Object detectors could benefit from the partially overlapping FOVs, by using the different viewing angles for tracking, for example, in heavily occluded scenarios. The authors of WildTrack suggested that prior to WildTrack, no large-scale and high-quality dataset met these requirements.
WildTrack has a high resolution of 1920x1080p and a length of over 60 minutes at 60 frames per second. Of those 60 minutes, 400 frames have annotated global labels. These 400 frames are shot at 12 Hz, spanning just over 30 seconds. Both the intrinsic and extrinsic camera parameters are available, as well as the distortion coefficients. Over the entirety of the dataset, roughly 300 individuals pass the square over an area of 500 m^2.

2.7. Research gaps

In this section, research gaps are discussed from two works: [19], [27]. The work of [19] investigated the single-person human-pose calibration method, showing promising results for further research in human-pose-based calibration. The research gap in this work is the extension to multi-person human-pose calibration, which the author mentioned in their future work section. The other recommendation comes forth out of multi-person calibration. Recommendations concerning matching the multi-person human poses for correspondence points and handling mismatching errors.
The other research gap came from the work [27]. The authors proposed an auto-calibration model

Figure 2.5: The seven views of the WildTrack dataset [6]. The top four images are shot with a Go Pro hero 3 and the bottom three are shot with a Go Pro hero 4, explaining the shift in image quality.

that does calibrate scenarios using multi-person human-pose estimation. However, instead of the total human-pose estimation, it uses a processed version of the human-pose estimation with two correspondence points. This processed version is called a vertical stick and is the maximum and minimum value in height of the pedestrian (in pixel coordinates). It has some limitations, such as the brute-force method for re-identifying pedestrian tracks. When paths are too short, this method cannot match the vertical sticks over all cameras in the network. The research gap of this method is the influence of the number of human-pose key points on the accuracy of a human-pose calibration model.

The research gaps found in the literature are summarized below and are the inspiration for the research questions from Section 1.2.

- Is there a difference in the calibration accuracy between single-person versus multi-person human-pose calibration?
- Does calibration accuracy increase if the human-pose estimation step has more key points extracted?
- When using multi-person calibration, how is the re-ID step handled, and how are mismatching errors discarded?

2.8. Summary

In this chapter, the related work gave an overview of the relevant literature of this research. The chapter elaborates on conventional calibration and automated calibration models, elaborating on their sub-components.
Section 2.1 discusses the conventional calibration method, Zhang's method [32]. Zhang's method can calibrate the intrinsic and extrinsic camera parameters.
Automated calibration in Section 2.2 explains all models that do an automatic calibration. These auto-calibration models are SIFT [18], models using motion barcodes, and human-pose-based calibration are presented.
The first Section of the sub-components of human-pose-based calibration is object detection in Section 2.3. Both double-staged and single-staged object detectors are discussed in this Section. In Section 2.4, the human-pose estimator AlphaPose and human-pose tracker PoseFlow are presented. Section 2.5 discusses the affinity-based feature extractor OSNet-AIN.
In Section 2.6, public datasets of camera networks are presented. These are datasets SALSA [1] and WildTrack [6].
Section 2.7 discusses the research gaps found in the works of [19] and [27].

3

Methodology

This chapter explains the methodological approach to answering the main research question and its sub-questions. In Section 3.1 the auto-calibration model's architecture is described on a high level. Section 3.1.1 tracks of human-pose estimations are generated. In Section 3.1.2, re-identification, two re-ID methods are proposed that assign labels to the human-pose tracks consistent over the camera network.
In Section 3.1.3, epipolar geometry, the first step in the structure-from-motion technique is set by calculating the intrinsic projection geometry. In Section 3.1.4 the first camera pair is calibrated, and Section 3.1.5 triangulates the first 3D human-pose points from this calibrated camera pair. Section 3.1.6, perspective-N-points, matches the 3D real-world points with the 2D pixel-points and calibrates the remaining camera's extrinsic camera parameters. In Section 3.1.7 the extrinsic camera calibrations from the perspective-N-points step are optimized using bundle adjustment.
Section 3.2, explains the alignment of the calibrated extrinsic parameters with the ground truth. The last three sections explain the experimental design for answering each research sub-question from the research questions in Section 1.2.

3.1. Design

Creating an algorithm capable of extracting the extrinsic camera parameters from image data is the desired output of this auto-calibration model, and this section will describe the design choices. Figure 3.1 visualizes the design on a high level.
The input of the auto-calibration model is both the intrinsic camera parameters and video data of all cameras in the network. The human-pose tracker extracts human-pose key points over time and assigns local labels to the tracked humans. The re-ID node matches pedestrians' local tracks of the human-pose tracker over all cameras in the network. The matched pedestrians' human-pose key points are then used as correspondence points and fed into the epipolar geometry node. The epipolar geometry computes the best initial camera pair to start the calibration and calculates the first 3D points of the camera network with triangulation. The Perspective-N-Points node calibrates the remainder of the cameras in the network, using 3D human-pose points and the 2D human-pose key points as input. When all cameras in the network are calibrated, the bundle adjustment step optimizes the result by minimizing the reprojection error.

3.1.1. Human-pose tracking

PoseFlow [29] is the human-pose tracker used in this research. PoseFlow uses the state-of-the-art human-pose estimator AlphaPose with its RMPE framework and improves upon it by adding tracking. The pose estimation step generates $k = 136$ different key points. The first 26 key points describe the human-pose estimation (Table 3.1), whereas the other 110 key points describe the hands and face. In the calibration step, only the 26 human pose key points are used for calibration.
Tracking of PoseFlow is per camera, meaning that it does not assign labels that are true for all cameras in the network. For the remainder of this research, tracking in a single camera is called *local tracking*, with local label l, and tracking in a camera network is called *global tracking*, with global label g.

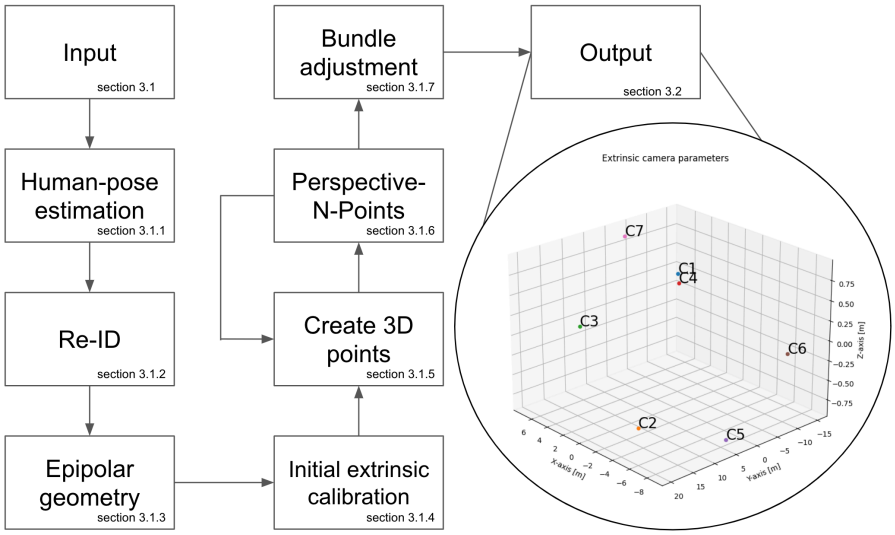

Figure 3.1: The block diagram of the proposed automated extrinsic calibration model.

The output of PoseFlow is: $\mathbf{h}_{Cfkl} = [u_{Cfkl}, v_{Cfkl}, c_{Cfkl}]$. The indices in $\mathbf{h}_{Cfkl}$ stand for camera number: C, frame length: f, key point: k, and local label: l.
Not all human-pose key points are equally hard to find. Ankles, for example, are currently the most complex key points to find for a human-pose estimator. The certainty of the human-pose estimator is quantified in the confidence parameter c_{Cfkl}. A threshold parameter detection accuracy (θ_{da}) filters out false detections ($c_{Cfkl} < \theta_{da}$). This filtration ensures that good quality human-pose estimations are used for calibration.

Index	key point	Index	key point	Index	key point
0	Nose	9	Left wrist	18	Neck
1	Left eye	10	Right wrist	19	Hip
2	Right eye	11	Left hip	20	Left big toe
3	Left ear	12	Right hip	21	Right big toe
4	Right ear	13	Left knee	22	Left small toe
5	Left shoulder	14	Right knee	23	Right small toe
6	Right shoulder	15	Left ankle	24	Left heel
7	Left elbow	16	Right ankle	25	Right heel
8	Right elbow	17	Head	-	-

Table 3.1: This table shows the human-pose key points according to PoseFlow. The tracker does detect extra key points from the hands and face, making up 136 key points in total, but these extra points are left out in this research.

3.1.2. Re-identification

Re-identification transforms the local tracks generated by PoseFlow into global tracks. Matching is done by comparing the pedestrians' affinity features with all other detected pedestrians' features in the camera network. When a match is found, the matched local tracks ($\mathbf{h}_{Cfkl}$) in two cameras get the same global label assigned ($\mathbf{h}_{Cfkg}$). The affinity-based feature extractor OSNet-AIN [34] is used for matching the pedestrians in this algorithm.
In Section 1.2, the second research question mentioned the handling of mismatching errors in automated re-ID. Mismatching errors are bound to occur due to ID switches in the tracker of PoseFlow, or a mismatch from the feature extractor OSNet-AIN. Therefore, two re-ID methods are proposed. The two methods are called *naive re-ID* and *filtered re-ID*. The difference between the two methods is a

temporal filter in the filtered re-ID method, designed to capture mismatching errors and discard them. The filtered re-ID method is an extension of the naive re-ID method, and therefore it is depicted as one flowchart in Figure 3.2. The dark tone orange blocks are the added nodes belonging to the filtered re-ID, which are ignored in the naive re-ID method. The remainder of this section will be a step-by-step explanation of both re-ID methods. First explaining the naive re-ID and later elaborate on the filtered re-ID method

The input of both re-ID methods is the human-pose tracks and the RGB-image streams of the camera network. After the first decision node, “found detections?”, in Figure 3.2, comes the feature extraction and comparing step of the re-ID models. OSNet-AIN extracts affinity-based feature vectors N from all detected bounding boxes of all cameras. When all features are extracted, the matching step of the process begins. Matching both re-ID methods is a complex step because all feature vectors from one camera viewpoint must be compared with all feature vectors of the other camera viewpoint. The reason for this inconvenience is that there is no information about the scene. Each time a new camera is added to the network, this matching problem becomes more complex.
The output of OSNet-AIN is a 512-D feature vector, and the authors of this feature extractor use the cosine distance to compare the similarity between feature vectors [34]. Cosine distance (Equation 3.1) outputs a scalar value between zero and one, and if this value is higher than the threshold parameter θ_{REID}, then these feature vectors are considered a match.

$$1 - \cos(\mathbf{N_i}, \mathbf{N'_j}) = 1 - \frac{\mathbf{N_i N'_j}}{\|\mathbf{N_i}\|\|\mathbf{N'_j}\|} \tag{3.1}$$

When a match is found between two feature vectors, the local labels l of both human-pose estimations ($\mathbf{h}_{Cfkl}$) are changed to global label g. Because the local labels are assigned to a human’s track, all local labels belonging to that track are changed to global labels. If one of the labels in the match already has a global label because, for example, it is seen by three cameras’ FOV, then only the local label changes. If there are still more frames in the RGB-image stream, this process is repeated until all frames are evaluated.

The second method adds filters to the naive re-ID method for catching mismatching errors because mismatching errors add noise to the system and negatively influence the accuracy of the calibration. There are two problems with re-ID, ID-switching and mismatching the affinity features. ID-switching is the switching of labels between two humans, and this could happen when pedestrians are occluded by walking past each other. Mismatching the affinity features happens when two pedestrians are similar in appearance in one frame but are not the same person. Catching these mismatches is done by two dark shaded orange process blocks in Figure 3.2. The first process block checks the global tracks. Each time a pair with the same global label is detected, the cosine distance is calculated again to check if the affinity features are still similar. When this value is lower than θ_{tracks}, the global labels are changed back to local labels for the remainder of both tracks. Notice that parameters θ_{tracks} and θ_{REID} are both threshold parameters for matching affinity features but will not have the same value. The threshold for matching θ_{REID} is always higher than θ_{tracks}. Therefore, easier to remain a global track and harder to match local tracks.
The second block only changes the local labels of tracks in global labels for future frames, whereas the naive re-ID method changes a track’s local label in the past and the future. Changing local labels in already evaluated frames could pose a problem. If the track switches from a pedestrian, due to an ID switch, in frames prior, then the re-ID method could not correct that anymore because it only corrects for frames in the future. Therefore, only local tracks with frames in the future will be changed to global labels.

Most video data is shot with a frame rate of 30 Hz or even 60 Hz, which means that looking for matches in each subsequent frame could be excessive since there is little change in the scenery. Therefore, the last addition is proposed; instead of reviewing every frame. Frame skip parameter F is introduced, and this parameter determines how many frames are skipped before evaluating again. By introducing frame skip, more data could be evaluated with the same number of computations, potentially increasing the model’s accuracy.

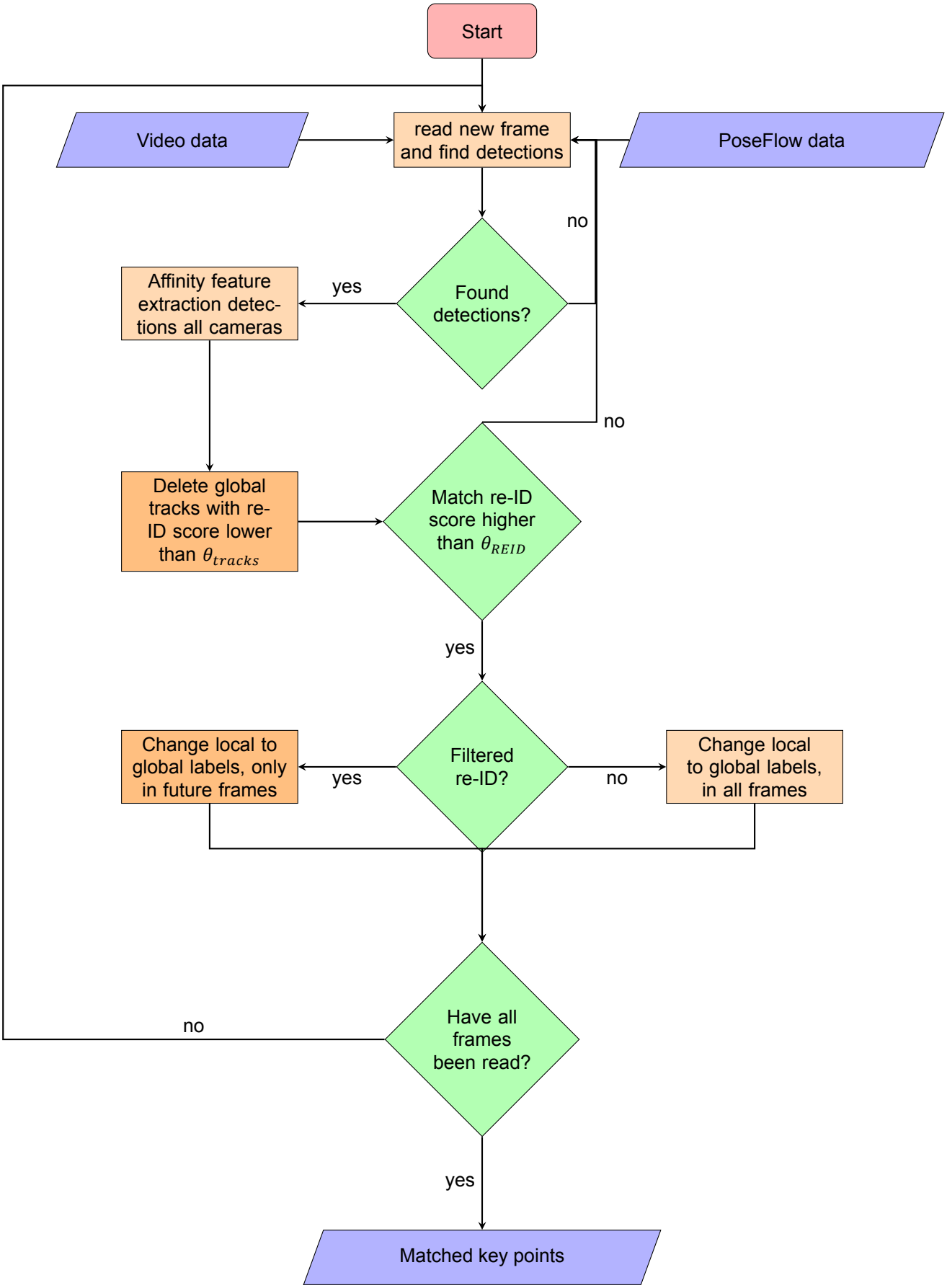

Figure 3.2: The flowchart of the filtered re-identification method. The trapezium blocks are the input and output nodes of the system. The diamond-shaped nodes are decision nodes and are always followed by an arrow with yes or no. The square nodes are process nodes divided into two tones. The lighter shade of orange is for both naive and filtered re-ID. The darker shade node is only for the filtered re-ID, and ignored by naive re-ID.

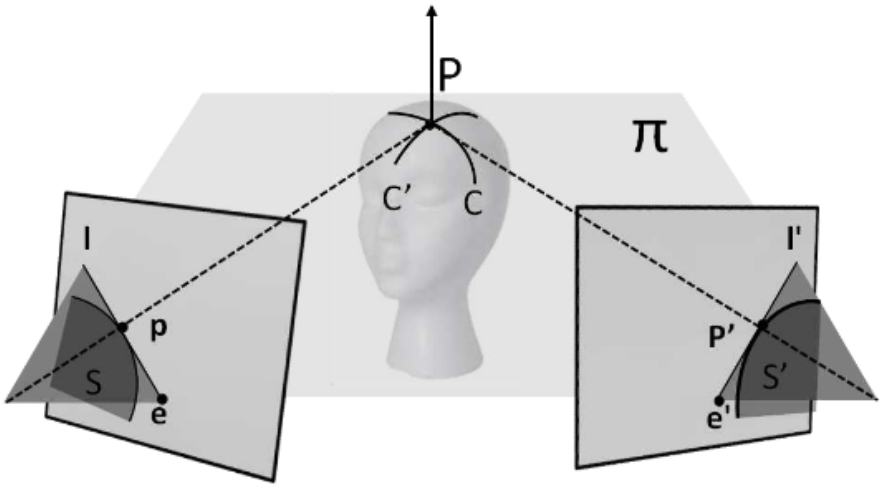

Figure 3.3: Epipolar geometry between two cameras visualized by [2]. The lines l and l' are the epipolar lines defining the epipolar plane π. Multiple epipolar lines will intersect, and this intersection is the epipole. An epipole defines the location of the other camera's center point.

3.1.3. Epipolar geometry

Following the processes of Figure 3.1, the next step is the epipolar geometry. The epipolar geometry is independent of scene structure and is the intrinsic projective geometry between two views [5]. It is called intrinsic projective geometry because it only depends on the camera's intrinsic parameters. Correspondence points are used to calculate the projective geometry between two views. These correspondence points are pixel coordinates in two views that point at the same object in 3-dimensional space. Correspondence points are depicted in Figure 1.1.

The intrinsic projective geometry is defined in three parameters: the epipoles ($\mathbf{e}$), epipolar lines ($\mathbf{l}$) and epipolar plane (π), visualized in Figure 3.3. Epipoles are the cameras' center points, an epipolar plane is a plane with one edge being the baseline of the two cameras, and the epipolar line is the intersection of the image plane with the epipolar plane. The desired output is the relative position of the cameras and one matrix that encapsulates all this information called the fundamental matrix [11]. The fundamental matrix is a matrix $\mathbf{F} \in \mathcal{R}^{3x3}$ of rank two that satisfies Equation 3.2. Here the x' and x are corresponding points in homogeneous coordinates. Correspondence points are the matched human-pose key point pixel coordinates. The normalized 8-point algorithm needs a minimum of eight corresponding points to compute the fundamental matrix and is the simplest method that performs well [11]. The simple normalization in translation and scaling is done to improve stability; the normalization is a translation so that the corresponding points' centroid is at the origin and the root mean squared (RMS) distance is equal to $\sqrt{2}$ [11]. The work of [19] used the normalized 8-point algorithm as well and found by the empirical observation that this algorithm was more robust when the observation noise was large. In human-pose-based calibration, observation noise is high because it is hard to determine a single pixel location for a human-pose key point like the ankle. Observation noise is not the only source of the noise. When the re-identification algorithm [34] is used for labeling the pedestrians, mismatch errors are bound to occur. Camera calibration methods, like SIFT [17], deal with noisy data with random sample consensus or RANSAC. The RANSAC algorithm calculates multiple fundamental matrices by forming multiple samples of correspondence points of size $n = 8$. Each subset of correspondence points is fed to the normalized 8-point algorithm, which calculates the fundamental matrix $\mathbf{F}$. All corresponding points will be tested for each $\mathbf{F}$ matrix by Equation 3.2, corresponding point pairs that satisfy $x'\mathbf{F}x < E_{rep}$ will be considered an inlier. E_{rep} is the reprojection error of the fundamental matrix, and this error term is hand-picked. The $\mathbf{F}$ matrix with the most inliers is considered the true fundamental matrix, and all its inliers are stored in $\mathbf{S}_{inliers}$.

$$x'^{T}Fx = [u_i' v_i' 1] \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = 0 \tag{3.2}$$

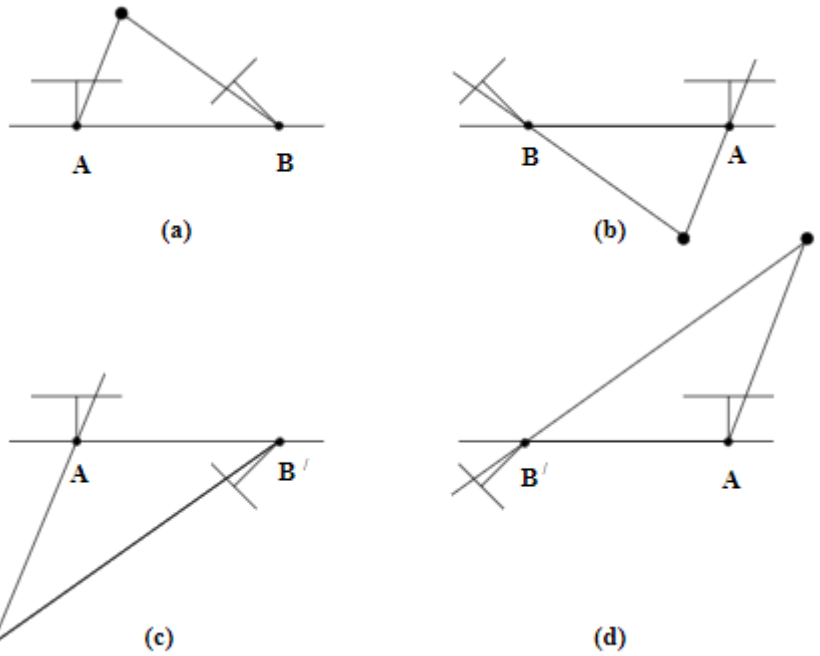

Figure 3.4: The cheirality check is visualized by [11]. **A** and **B** are the center points of the camera, the line out of the center point is the viewing angle, and the line perpendicular to that line is the image plane. The black dot represents the triangulated point from all four possibilities. When the black dot is in front of both cameras, as in scenario a, the cheirality check is over, and the correct extrinsic camera parameters are found.

3.1.4. Initial extrinsic calibration

The relative pose of the first camera pair will be determined by the decomposition of the essential matrix $\mathbf{E}$. The essential matrix between two cameras is the specialization of the fundamental matrix (Equation 3.3), requiring the initial camera calibration parameters ($\mathbf{K}_C$) of both cameras. This decomposition is only possible with the essential matrix because it has fewer degrees of freedom (DOF). The fundamental matrix has seven, and the essential matrix has five DOFs and is singular.
The camera poses can be extracted from the essential matrix, but only up to a scale and with a four-fold ambiguity [11]. The first cameras picked for the initial calibration are the camera pairs with the most inliers from the epipolar geometry block. The first extrinsic camera parameters are initialized as $\mathbf{P}[\mathbf{I}|\mathbf{0}]$ and according to the authors of [11], the singular value decomposition (Equation 3.4) decomposes the parameters needed for the second extrinsic camera parameters $\mathbf{P}'$ which is relative to $\mathbf{P}[\mathbf{I}|\mathbf{0}]$. Equation 3.5 shows the four-fold ambiguity, $\mathbf{u}_3$ is the third value of $\mathbf{U}$ and is the translation vector. This translation vector could be positive or negative. The rotation matrix is determined by the product of $\mathbf{UWV}^T$ or $\mathbf{UW}^T\mathbf{V}^T$. The only way to check which of the four configurations in Equation 3.5 is valid is by a cheirality check. A cheirality check triangulates a point with all four possible configurations. The configuration is considered the valid extrinsic camera parameters when the 3D point from the triangulation step is in front of both cameras (returns a positive depth value). This process is shown in Figure 3.4.

$$\mathbf{E}_{ij} = \mathbf{K}_j^T \mathbf{F} \mathbf{K}_i \tag{3.3}$$

$$\mathbf{E} = \mathbf{U} \begin{bmatrix} 1,0,0 \\ 0,1,0 \\ 0,0,0 \end{bmatrix} \mathbf{V}^T = \mathbf{U} \begin{bmatrix} 1,0,0 \\ 0,1,0 \\ 0,0,0 \end{bmatrix} (\mathbf{W}\mathbf{U}^T\mathbf{R}) \tag{3.4}$$

$$\mathbf{P}' = [\mathbf{UWV}^T| + \mathbf{u}_3] \text{ or } [\mathbf{UWV}^T| - \mathbf{u}_3] \text{ or } [\mathbf{UW}^T\mathbf{V}^T| + \mathbf{u}_3] \text{ or } [\mathbf{UW}^T\mathbf{V}^T| - \mathbf{u}_3] \tag{3.5}$$

3.1.5. Create 3D points

Once the first pair of cameras is found, all human-pose key points $\mathbf{h}_{Cfkg}$ could be projected onto the 3D world coordinate scene as $\mathbf{H}_{fkg}$. Projecting pixel coordinates to real-world coordinates is done via triangulation and is mostly known from stereo vision. This process is displayed in Figure 3.5. The

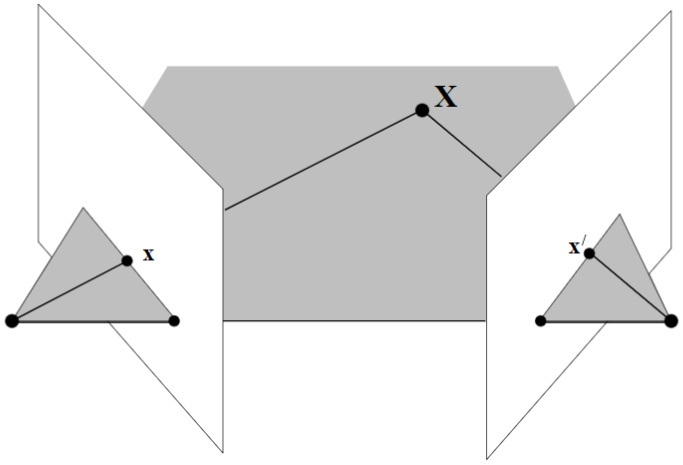

Figure 3.5: Triangulation visualized by [11]. $\mathbf{x}$ and $\mathbf{x}'$ are the corresponding pixel coordinates in two image planes. $\mathbf{X}$ is the 3D world coordinate in $[x_{wc}, y_{wc}, z_{wc}]$

homogeneous Direct Linear Transform (DLT) is used for triangulating the 3D world coordinate $\mathbf{H}_{fkg} \in \mathcal{R}^{4x1}$. Notice the drop of camera index C, which is no longer necessary because it is a real-world coordinate now, independent of camera angle. Triangulation starts by forming two equations using the corresponding human-pose key points: $\mathbf{h}$ and $\mathbf{h}'$ (all other parameters f,k,g are equal) and camera matrices $\mathbf{P}$ and $\mathbf{P}'$. The two equations are then: $\mathbf{h} = \mathbf{PH}$ and $\mathbf{h}' = \mathbf{P}'\mathbf{H}$, and when combined it is written to a form: $\mathbf{AH} = \mathbf{0}$. Solving $\mathbf{H}$ is done by a singular value decomposition (SVD) $\mathbf{A} = \mathbf{USV}^T$, finding the smallest singular vector corresponding to the smallest singular value of matrix $\mathbf{A}$ [11]. Eventually, this leads to the answer where $\mathbf{H}$ is equal to the last column of $\mathbf{V}$ from the SVD.
All 2D human-pose key points ($\mathbf{h}_{Cfkg}$) from the inliers ($\mathbf{S}_{inliers}$) are now be converted to points in 3D $\mathbf{H}_{fkg}$. It is limited to only the inliers; otherwise, the outliers that contain noise are triangulated to 3D world points $\mathbf{H}$, hurting the perspective-N-points step in the next section.

3.1.6. Perspective-N-Points

Perspective-n-points is finding the tomography between 2D pixel coordinates and 3D real-world locations. Homography is, in this research, the extrinsic camera calibration parameters. In the previous section, 3D points were triangulated from calibrated cameras. When adding a new camera to the calibrated camera network, all the human-pose key points ($\mathbf{h}_{Cfkg}$) belonging to the new camera are compared with the 3D points ($\mathbf{H}_{fkg}$). A match is found whenever parameters $[f, k, g]$ are equal. The number of matches is stored, and all calibrated cameras undergo this process. The camera angle with the most matches is chosen for the perspective-N-points step.

The algorithm used for perspective-N-point is called the EPnP algorithm [14], and it is in combination with RANSAC, known from the fundamental matrix estimation in Section 3.1.3. EPnP is a non-iterative approach that is both accurate and stable. The authors of [14] combined their proposed algorithm EPnP with RANSAC to filter out erroneous correspondence points. In this application, the erroneous correspondence points are likely to be mismatched errors from the re-identification step. RANSAC calculates the best extrinsic camera parameters with the lowest reprojection error θ_{PnP}. Setting this error term high will cause the RANSAC algorithm to converge faster to a solution, reducing calibration accuracy.

Table 3.1 shows that the node, creating new 3D points, loops back to the perspective-N-points step. When the perspective-N-points step finds new extrinsic camera parameters, it adds more 3D points by triangulation. More 3D point projections mean more chance that a newly introduced camera has 2D point projections that correspond, and so add the new camera in the same coordinate frame. This process is done when the number of cameras equals the number of intrinsic camera parameters $\mathbf{K}_C$ or if all possible camera pairs have been reviewed.

3.1.7. Bundle adjustment

The final step of the methodological framework is the optimization step of the design. Bundle adjustment refines the camera matrices' initial estimation using an objective function. In this case, it is the reprojection error. This error is an error in pixel distance between the pixel location of the human-pose key points and the reprojected 2D pixel coordinates of the 3D world coordinates. This process is visualized in Figure 3.6. The 3D points are triangulated human-pose key points from Section 3.1.5. The pixel coordinates are calculated by Equation 3.6.

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \rho & c_x \\ 0 & f_y & c_y \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.6}$$

$$L_{MSE}(y, f(x)) = \frac{1}{N}\sum_{i=1}^{N} |y - f(x)|^2 \tag{3.7}$$

$$L_{MAE}(y, f(x)) = \frac{1}{N}\sum_{i=1}^{N} |y - f(x)| \tag{3.8}$$

$$L_{huber}(y, f(x), \delta) = \begin{cases} \frac{1}{2}(y - f(x))^2 & if\ \left|(y - f(x))\right| < \delta \\ \delta((y - f(x)) - \frac{1}{2}\delta) & otherwise \end{cases} \tag{3.9}$$

Three loss functions could be used to calculate the reprojection error. These functions are the mean squared error (Equation 3.7), the mean absolute error (equation 3.8) and the Huber loss function (Equation 3.9). The main disadvantage of the mean squared error is its sensitivity to noise. Outliers are squared in the mean squared error function, increasing its influence. The mean absolute error, on the other hand, ignores these errors by dropping the quadratic term. The Huber loss is a compromise between the two loss functions, and the δ term determines the threshold of when to use which loss function. For the reason that the data could contain plenty of noisy data, the Huber loss function is used as the loss function for the reprojection error.

The optimization algorithm is the Levenberg-Marquardt least squares algorithm and is often used in bundle adjustment, like in these works [26] [19]. It finds the minimum of a function that is non-linear, in this case, the translation and orientation of the cameras within the camera matrices. The rotation matrix within the extrinsic camera parameters is written to a rotation vector and, together with the translation vector, resulting in a 6-dimensional optimization problem.

3.2. Evaluation

Each research question compares the calibrated extrinsic camera parameters with the ground-truth data from the dataset. An issue with comparing the ground truth to the calculated extrinsic camera parameters is that the output of the auto-calibration model is relative. The parameters are in relative coordinates because there is no additional information about the scene's dimensions, which means that scaling, translating, and rotating are necessary for evaluation. This section explains the data preparation needed to compare the ground truth with the calculated extrinsic camera parameters.

The first step is translation, both the ground truth and the calculated extrinsic parameters their center coordinates are translated to the origin, making it possible to align them with rotation. The next step in aligning the camera networks is by calculating the scale. Here the magnitudes of all camera translation vectors $\mathbf{t}$ are compared in Equation 3.10. For each camera C, scale s_C is calculated. The mean value of s_C is considered the overall scale of the camera network. The extracted camera network is then multiplied by the scale factor s.

$$s_i = \sqrt{|\mathbf{t}_c^{\text{ground truth}}|} / \sqrt{|\mathbf{t}_c^{\text{calibrated}}|} \tag{3.10}$$

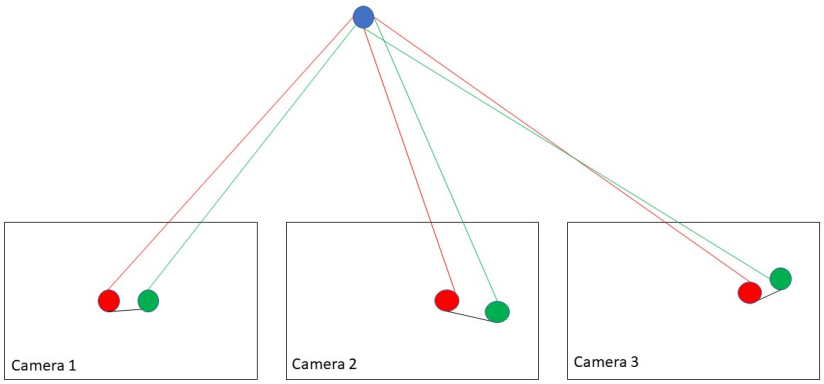

Figure 3.6: The red dots represent the human-pose key points used for triangulating the 3D real-world dot in blue. The green dot is the pixel location of the reprojected 3D world point back onto the image, and the black line between the two pixels is called the reprojection error.

After scaling, the orientation of the camera network is adjusted. The vectors of both camera networks are aligned using a Kabsch algorithm (Equation 3.11), by minimizing $L(\mathbf{R})$ using the root mean squared deviation between the two rotation matrices $\mathbf{R}$.
Multiplying this rotation matrix with the extracted camera network returns a network with almost the same scale and orientation. The difference between the ground truth extrinsic camera parameters and the calibrated camera parameters are calculated using the root mean squared (RMS) error function. This calculation is done per the camera's rotation matrix and translation vector. The RMS error values are referred to as the calibration accuracy.

$$L(\mathbf{R}) = \frac{1}{2}\sum_{i=1}^{n} ||\mathbf{t}_C^{\text{calibrated}} - \mathbf{R}\mathbf{t}_C^{\text{ground truth}}||^2 \tag{3.11}$$

3.2.1. Human-pose calibration

The first experiment concerns the first research sub-question from Section 1.2. This experiment compares single-person human-pose-based calibration with multi-person human-pose-based calibration and the influence of the number of key points used for calibration. Matching the human-pose estimations in multi-person calibration is done by the annotated re-ID, in this experiment. Annotated re-ID is done by the maker of a dataset, annotating global labels to pedestrians by hand.
Firstly, the experiment of single- versus multi-person human-pose-based calibration is executed. The single-person calibration picks the pedestrian with the longest path in the dataset. The multi-person calibration has no limit on the number of pedestrians is picks for the calibration. The comparison between the two calibrations is made by reviewing their calibration accuracy.

The second test examines whether increasing the number of human-pose key points affects the calibration accuracy. This test will examine the calibration score for four configurations, visualized in Figure 3.7. Three configurations use the human-pose key points (Full, Simple, Vertical stick), configuration full has 26 key points, configuration Simple has five key points, and Vertical stick has two key points. The fourth configuration is added to examine when zero key points are found. Instead of key points, it uses the four points from the bounding box.

3.2.2. Automatic re-ID

This experiment analyzes the automated re-ID models, naive and filtered re-ID. The re-ID methods are added to the multi-person human-pose calibration, and their calibration accuracy is compared.

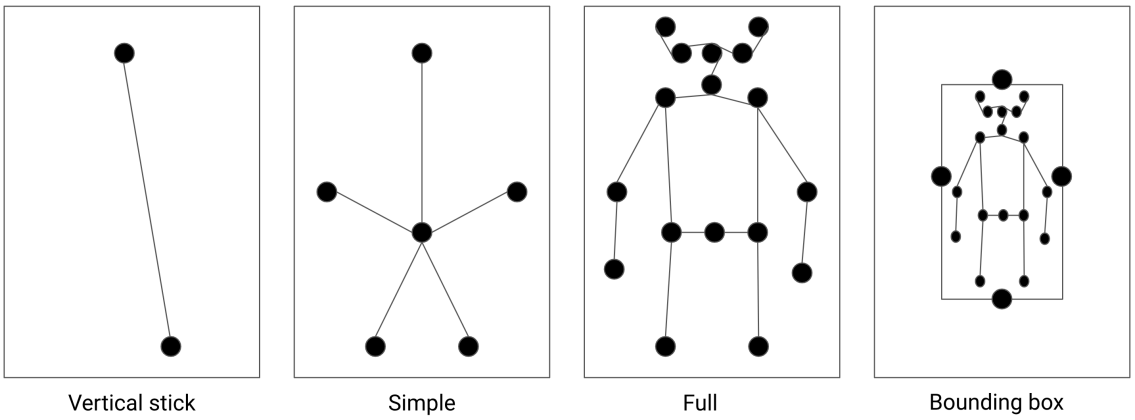

Figure 3.7: The four configurations of the correspondence points used in this experiment. The first three configurations are from the human-pose estimation, and the fourth configuration uses the bounding box coordinates confined by the human.

The first test examines if the automatic re-ID method suffers from mismatching errors by calculating the calibration accuracy over multiple frame lengths. This test could show if the model suffers from mismatching errors and if the filtered re-ID's temporal filter can discard these mismatching errors.
The second test of the experiment examines frame skip. Frame skip is added to the re-ID models because it tests the effect of evaluating each subsequent frame in re-ID versus skipping a pre-defined number of frames for each evaluation. Four values for frame skip are tested, [0,10,25,50]. All four calibrations are done with the same number of evaluations.

3.2.3. Automatic versus annotated re-ID

In this experiment, the automatic re-ID methods, including frame skip, are compared to the best annotated re-ID calibration from Section 3.2.1.
This comparison answers the last research sub-question from the research questions if there is a decrease in calibration accuracy when adding automatic re-ID.

3.3. Summary

This chapter discussed the methodological approach of designing an automated human-pose-based auto-calibration model capable of calibrating camera networks without needing a special operator on site.
Section 3.1 gives a high-level overview of the proposed model, containing the human-pose estimator, the re-ID algorithm, and the structure-from-motion technique [11].
The Section 3.1.1 discusses PoseFlow [29]. This human-pose tracker extracts the human-pose estimation and tracks humans over time. Section 3.1.2 describes the re-ID methods and how they transform tracks with local labels to tracks with global labels.
The Section 3.1.3 explains the first step in the structure-from-motion technique [11], computing the fundamental matrix with the normalized 8-point algorithm. In Section 3.1.4 the essential matrix is calculated and decomposed in the first pair of initial extrinsic camera parameters.
In Section 3.1.5 the first 3D human-pose points are triangulated for calibration.
Perspective-n-points in Section 3.1.6, adds new cameras to the calibration, using the EPnP algorithm [14] in combination with RANSAC.
Section 3.1.7 optimizes the calibration by minimizing the reprojection error using the Levenberg-Marquardt algorithm and the Huber loss function. This optimization technique is called bundle adjustment.

The evaluation Section 3.2 explains the calculation of the calibration accuracy by aligning the calibrated extrinsic camera parameters with the ground truth.
Experiment 1 in Section 3.2.1 describes the experiment of how to analyze the difference in single-person and multi-person calibration, followed by a second test analyzing the calibration accuracy with four key point configurations.
Experiment 2 in Section 3.2.2 explains the experiment where it compares the re-ID methods naive and

filtered when used in a multi-person human-pose calibration model, including the addition of frame skip. Experiment 3 in Section 3.2.3 describes the experiment automatic re-ID versus annotated re-ID.

4

Results

This chapter presents the results of the experiments proposed in the methodology in Section 3.2. Section 4.1 explains the datasets used for evaluating the proposed calibration method, and Section 4.2 describes the testing setup by describing the hyper-parameters of the model one by one. In Section 4.3 the calibration accuracy between single-human-pose estimation and multi-human-pose is compared, as is the effect of the number of human-pose key points on the calibration accuracy. In Section 4.4, the automatic re-ID methods are added to the calibration model and analyzed. In Section 4.5 the difference in calibration accuracy between automatic re-ID and annotated re-ID in human-pose-based calibration is examined.

4.1. Datasets

The datasets used for the evaluation step are publicly available datasets SALSA [1], and WildTrack [6]. The SALSA dataset consists of two social gatherings in one room of approximately 100 m^2 surveyed by four cameras $(C_1, C_2, ..., C_4)$ with a resolution of 1024x768 pixels. All four cameras overlap in their field of view and have their intrinsic and extrinsic camera parameters available. This dataset is chosen as one of the evaluation datasets because it is publicly available, has two scenarios being 25 minutes long at 15 fps, and has 18 participants entering and exiting the field of view of all cameras.

WildTrack is the second dataset and is a bit more challenging than the SALSA dataset. This dataset is filmed outside by seven cameras $(C_1, C_2, ..., C_7)$ in a scene of approximately 500 m^2, with a resolution of 1920x1080 pixels. The challenging part is that not all cameras overlap in FOV. One of the advantages over SALSA is that the annotation of global pedestrian labels is carefully done, therefore a useful dataset to compare the naive and filtered re-ID methods to. The length of the dataset is an hour long, and over this time, 300 participants cross the scene.
All information regarding the datasets is summarized in Table 4.1.

	cameras	overlap	participants	duration	resolution	area
SALSA [1]	4	full	18	50 min	1024x768	100 m^2
WildTrack [6]	7	partially	300+	60 min	1920x1080	500 m^2

Table 4.1: Properties of datasets SALSA and WildTrack.

Both datasets suffer from lens distortion, and the parameters to correct this effect are present in the known intrinsic camera parameters, called the distortion coefficients. Before calibration, all frames are corrected from this distortion by using OpenCV.

4.2. Testing setup

The experiments in this chapter were done on an Amazon EC2 G4 instance. The instance, G4dn.large, was a single GPU with 16GiB of memory.
The extrinsic auto-calibration model used specific hyper-parameters. This section explains these parameters one by one. The detection accuracy of the human-pose estimation is set to $\theta_{da} = 0.7$ (Figure A.1). The threshold for the OSNet-AIN feature extractor for matching local labels to global labels is set to $\theta_{REID} = 0.8$ (Table A.1) and the filter parameter for deleting global tracks is set to $\theta_{tracks} = 0.7$. The maximum reprojection error when estimating the fundamental matrix is $E_{SfM} = 2.0$ and the maximum reprojection error in the perspective-N-points problem is set to $E_{PnP} = 4.0$. The threshold parameter for the optimization with bundle adjustment is set to $\theta_{bundle} = 0.5$ (as in the works of [19]).
All human pose key point extractions are done with the human-pose estimator PoseFlow [9]. Other information like the evaluation dataset or the frame length is described per Section.

4.3. Experiment 1: Human-pose calibration

The first experiment, in Section 4.3, investigates the research sub-question: "*When assuming that re-identification across frames is correct, how does a single-person human-pose estimation versus a multi-person human-pose estimation affect the calibration accuracy and is the accuracy affected by the number of human-pose key points?* ". The dataset for this experiment was WildTrack, using the annotated global labels. Both tests are executed with the maximum number of frames for evaluation.

4.3.1. Single-person versus multi-person

The first test is the comparison between the single-person calibration and the multi-person calibration. The expected outcome of this experiment is a better calibration accuracy for multi-person calibration. This expected outcome is because the best result in the works of [19] was achieved when a single participant covered most of the area. Instead of one participant covering a large area, this experiment uses multiple participants for covering a large area.

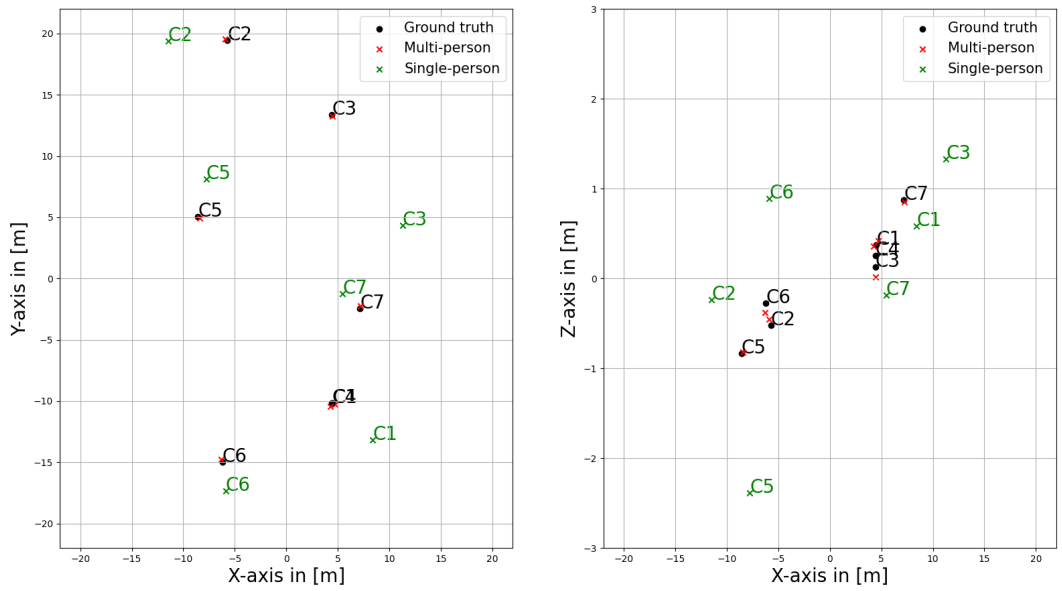

Figure 4.1: The area plot using dataset WildTrack, with frame length $f = 300$ and annotated re-ID. Single-person (in green) is compared to multi-person (in red) is compared to the ground-truth calibration of WildTrack. The XY-plot has an equal range, but the XZ-plot has not. Camera names of the multi-person plot are left out because they are close to the ground truth camera names.

	WildTrack multi-person		WildTrack single-person	
	Translation error [m]	Rotation error [rad]	Translation error [m]	Rotation error [rad]
C_1	0.019	0.0	3.928	0.005
C_2	0.019	0.0	14.983	0.018
C_3	0.010	0.0	30.964	0.244
C_4	0.032	0.0	-	-
C_5	0.012	0.0	8.492	0.028
C_6	0.025	0.0	0.632	0.015
C_7	0.020	0.0	5.216	0.003
average	0.020	0.0	10.70	0.052

Table 4.2: This table shows the RMS error for translation and rotation for the configuration $f = 300$ over all seven cameras in the network.

Figure 4.1 shows the translation vectors of the calculated extrinsic camera parameters, with $f = 300$. It has only evaluated 300 frames instead of the maximum 400 annotated frames because the GPU ran out of memory while calculating the human-pose tracks. Figure 4.1 draws three plots: the annotated values, the single-person, and the multi-person calibration. This figure shows that single-person calibration is significantly worse compared to the multi-person calibration in camera center point approximation. Table 4.2 depicts the calibration accuracy of both single- and multi-person calibration in RMS errors. The single-person calibration was not able to find all cameras in the calibration. Camera C_4 was not found for this testing setup. The calibration accuracy had an average RMS error of 10.7 meters in translation and 0.052 radians in rotation, which is poor calibration accuracy. The accuracy gained by including all people in the scene was significant, closely resembling the ground truth with an RMS error in translation of 0.02 meters and 0.0 radians. These results are in line with the expected outcome.

4.3.2. Number of key points

The second test of this experiment investigates the influence of the number of human-pose key points and the calibration accuracy. Four key point configurations are used in this experiment, with varying numbers of key points. These four configurations are visualized in Figure 3.7. Three configurations (Vertical stick, Simple, Full) use key points from the human-pose estimator, and one uses the Bounding box configuration. The frame length is again the maximum frame length from the previous Section $f = 300$.

The work of [27] used only the extremes of the human-pose (vertical stick configuration) for their robust auto-calibration model, with accurate extrinsic camera parameters as output. In the work of [19], covering more area increased calibration accuracy. Increasing the number of human key points does not increase the covered areas. Therefore, both works suggest that increasing the human-pose key points will not necessarily increase calibration accuracy.

Figure 4.2 shows the area plots of dataset WildTrack with four configurations of human-pose key points. The results are shown in Table 4.3. The bounding box coordinates do not supply good quality correspondence points for the calibration, with the calibration method finding three cameras out of seven. The three configurations using the human-pose key points do find accurate calibrations. Configuration vertical stick finds six out of seven cameras with an average RMS error of 0.109 meters in translation and 0.044 radians in rotation. Simple human finds all cameras in the network, but the calibration suffers from the poorly calibrated fourth camera. The average RMS error of this configuration is 0.652 meters in translation and 0.005 radians in rotation, meaning that the overall calibration accuracy is worse for the simple human-pose configuration than the vertical stick configuration, albeit the simple configuration could calibrate all the cameras. The best calibration is calculated when all human-pose key points that determine the entire body are used for calibration. The configuration using the full human pose has

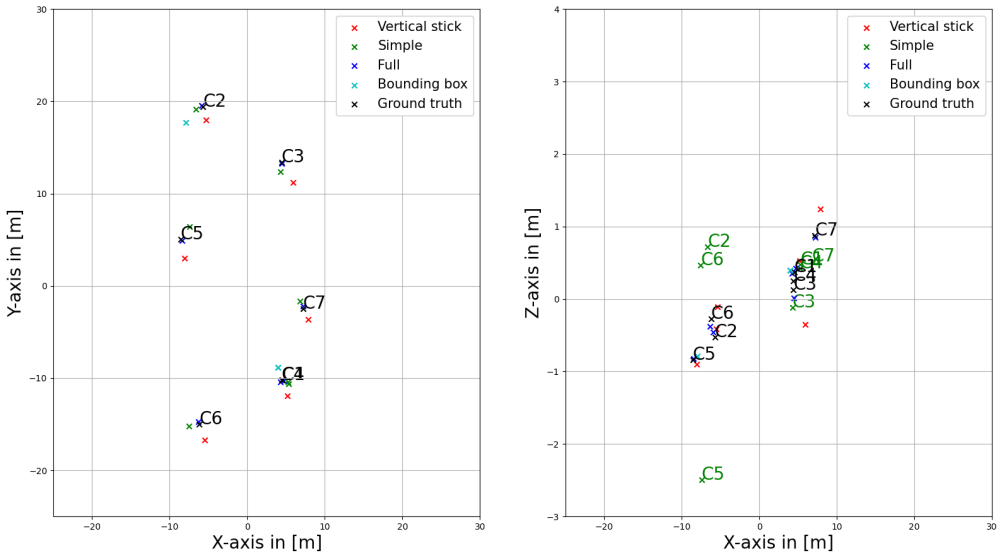

Figure 4.2: The area plot using dataset WildTrack, with frame length $f = 300$ and annotated re-ID. Four key point configurations are compared to the ground truth calibration of WildTrack. The XY-plot has an equal range, but the XZ-plot has not. The camera names of the ground truth are shown, and of the configuration Simple in the XZ-plane.

a near-perfect calibration when considering rotation and the average translation error is 0.020 meters in an area of 500 m^2.

These tests conclude the first experiment. Multi-person human-pose calibration showed an increase in calibration accuracy compared to single-person human-pose calibration. This accords with the work of [19] that more correspondence points covering a larger area lead to better calibration. When calibrating using the human-pose key points as correspondence points, it is beneficial to use more points of the human pose instead of subsets defining the longest vertical line or a more straightforward representation of the human pose. These findings could improve the work of [27], assuming that their brute-force matching algorithm could handle more human-pose key points.

There were two limitations found in this experiment. The first was that the extrinsic parameters were not calculated in a controlled environment as [19]. Single-person human-pose calibration in this test was done “in the wild” by picking the person who was recorded the longest. Therefore, it could not conclude if such a boost in accuracy is found when single-person human-pose calibration is compared to multi-person human-pose calibration in a controlled environment.
The second limitation was the GPU memory shortage. This experiment used the annotated re-ID and had 400 frames available for calibration. However, the GPU was limited to 300 frame evaluations before being killed.

	Vertical stick		Simple human-pose	
	Translation error [m]	Rotation error [rad]	Translation error [m]	Rotation error [rad]
C_1	0.007	0.0	0.281	0.003
C_2	0.107	0.0	0.832	0.004
C_3	0.338	0.07	0.376	0.002
C_4	-	-	0.261	0.003
C_5	0.058	0.001	1.965	0.022
C_6	0.012	0.0	0.791	0.001
C_7	0.130	0.001	0.318	0.003
average	0.109	0.044	0.652	0.005
	Full human-pose		Bounding box	
	Translation error [m]	Rotation error [rad]	Translation error [m]	Rotation error [rad]
C_1	0.019	0.0	5.006	0.887
C_2	0.019	0.0	0.121	0.736
C_3	0.010	0.0	-	-
C_4	0.032	0.0	-	-
C_5	0.012	0.0	-	-
C_6	0.025	0.0	-	-
C_7	0.020	0.0	6.197	0.884
average	0.020	0.0	11.32	0.836

Table 4.3: Dataset WildTrack is used with frame length $f = 300$. Root mean squared error of the four configurations in correspondence points. These configurations are Vertical stick, Simple, Full, and Bounding box.

4.4. Experiment 2: Automatic re-ID

In this experiment, both the re-ID methods are examined, naive re-ID and filtered re-ID. This experiment examines the research sub-question: "*In automatic re-identification, what is the effect of evaluating each subsequent frame versus skipping frames on the calibration accuracy, and how are mismatching errors handled?*". The naive and filtered re-ID methods are used for the re-ID step in the multi-person human-pose calibration. This analysis does not require a challenging dataset like WildTrack. This experiment examines the properties of the re-ID methods, so the smaller dataset SALSA is used.

4.4.1. Naive versus Filtered re-ID

The analysis of dataset SALSA is as follows, both re-ID methods are used for multiple frame lengths from $f = 100$ to $f = 1000$ with incremental steps of 100. For each of the steps, the calibration accuracy is calculated. The filtered re-ID method has a temporal filter against mismatching errors (Section 3.1.2). Therefore it should be more resilient against mismatching errors from the re-ID algorithm OSNet-AIN. The influence of these mismatches on calibration is shown by comparing the calibration accuracy of the naive and filtered re-ID.

Figure 4.3 plots the RMS error of the translation and rotation over the number of frames used. The RMS error remains constant over time for the filtered re-ID after $f = 300$. When using the naive re-ID method for calibration, the calibration accuracy increases RMS error, in translation, from $f = 800$.

Figure 4.4 visualizes the case of $f = 1000$ and its results are depicted in table 4.4. The difference in

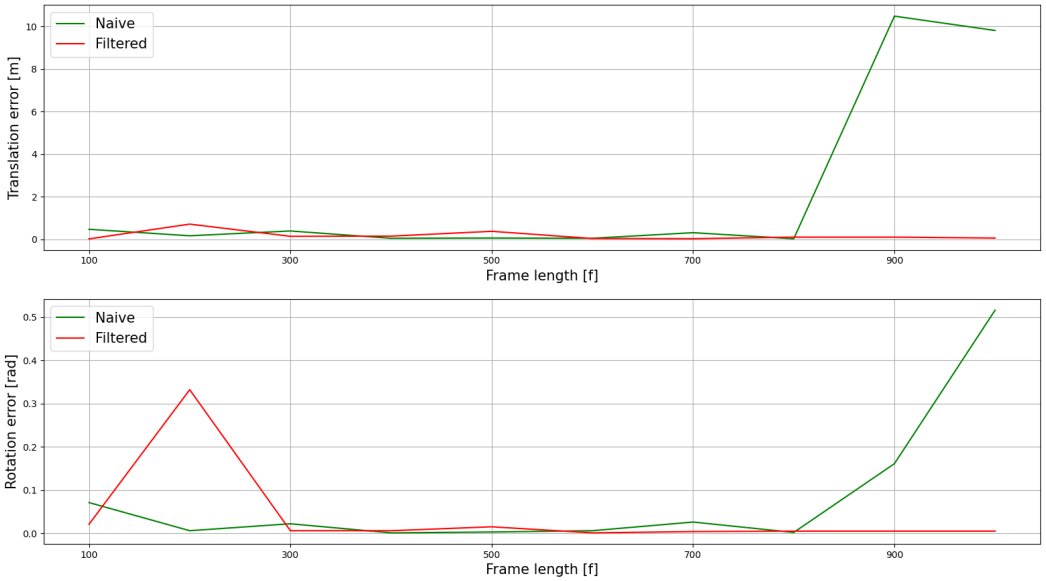

Figure 4.3: This figure displays both re-ID methods, applied to multi-person human-pose calibration, with their RMS error of the translation (top) and the rotation (bottom) plotted over their used frame lengths.

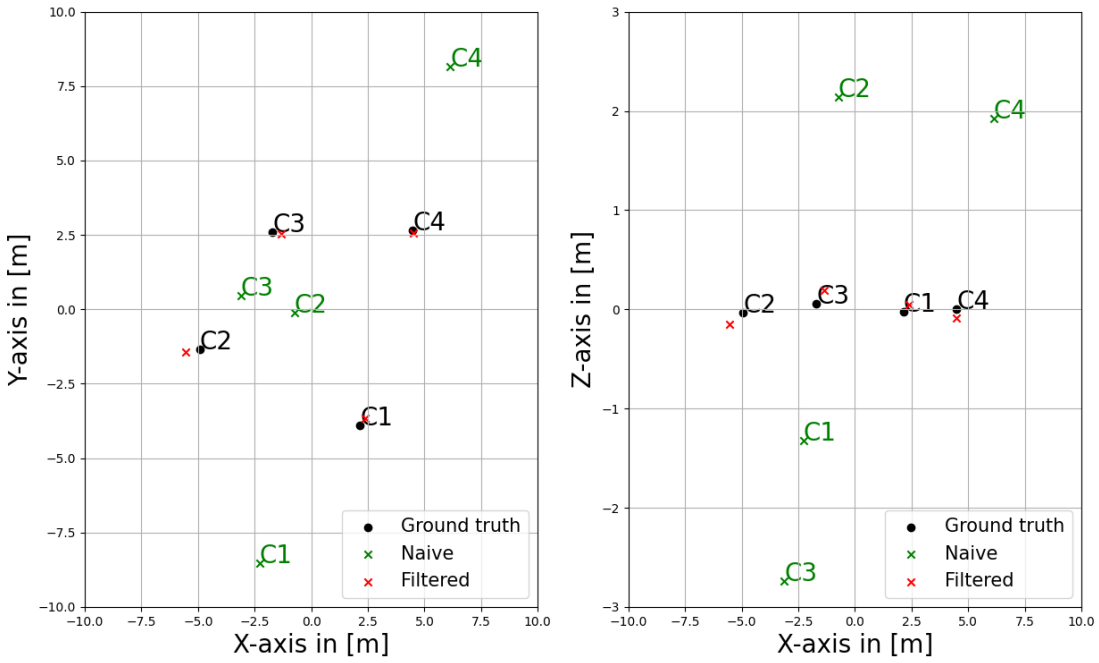

Figure 4.4: The surface plot of the dataset SALSA, using frame length $f = 1000$. The black dots represent the ground truth, and the green and red dots are the center points approximated by the auto-calibration models using the naive and filtered re-ID methods. The camera names of the filtered re-ID are not plotted because they closely resemble the ground truth. The XY-plot has an equal range, but the XZ-plot has not.

	Naive re-ID		Filtered re-ID	
	translation error [m]	rotation error [rad]	translation error [m]	rotation error [rad]
C_1	14.206	0.669	0.042	0.002
C_2	7.947	0.667	0.136	0.004
C_3	4.797	0.657	0.052	0.011
C_4	12.271	0.06	0.004	0.002
average	9.801	0.513	0.059	0.005

Table 4.4: Root mean squared error of both naive and filtered re-ID from Figure 4.4. These results were gathered with dataset SALSA for the case $f = 1000$

calibration accuracy between naive and filtered re-ID is significant. The average RMS error for naive re-ID is 9.801 meters in translation and 0.513 radians in rotation. The filtered re-ID had an RMS error of 0.059 meters in translation and 0.005 radians in rotation.

4.4.2. Frame skip

In this second part of this experiment, frame skip is added to both re-ID methods. Frame skip is set to four values: 0, 10, 25, and 50. The number of evaluations is set to 200.
Expected is that the efficiency of the model increases. The efficiency increases because frame length is enlarged, increasing the chance of correspondence points being spread out more, and the calibration is done with the same number of evaluations. Another expectation is that both re-ID methods will have similar calibration accuracies. This prediction is based on Figure 4.3, where it displays the difference in calibration accuracy. The difference in calibration accuracy is low for evaluations below 800, where this test will use 200 evaluations.

Table 4.5 shows the results of the re-ID methods with four values of frame skip. The expected increase in efficiency is only seen in the filtered re-ID, which improved the calibration accuracy. The best calibration accuracy was achieved with $F = 25$, scoring an RMS error of 0.019 meters and 0.0 radians. Another expected outcome was that both re-ID methods would not differ in calibration accuracy. However, this was only found for $F = 50$. The other values had RMS errors of multiple meters in translation, suggesting that re-ID methods when using frame skip are more susceptible to mismatching noise.

These tests conclude the second experiment. The results of this section suggest that temporal filtering in filtered re-ID mitigated mismatching noise well for up to $f = 1000$. The second test showed that to increase calibration accuracy, not all subsequent frames of a dataset need to be evaluated if a filter for handling mismatching errors is included. These findings answer the second research subquestion.

The limitation of this experiment is that comparing two re-ID methods only shows that there is mismatching noise. However, this experiment does not show the origin of the mismatching errors, like if they originate from ID switches in the PoseFlow tracker step or if the feature extraction step of OSNet-AIN creates this noise.

	Naive re-ID							
	$F = 0$		$F = 10$		$F = 25$		$F = 50$	
	t [m]	R [rad]	t [m]	R [rad]	t [m]	R [rad]	t [m]	R [rad]
C_1	0.101	0.004	1.802	0.888	2.996	0.59	0.060	0.015
C_2	0.303	0.006	1.585	0.889	3.941	0.594	0.044	0.002
C_3	0.232	0.012	2.829	0.868	11.862	0.618	0.026	0.001
C_4	0.023	0.003	3.236	0.507	4.159	0.744	0.021	0.001
average	0.165	0.006	2.363	0.788	5.740	0.637	0.038	0.005
	Filtered re-ID							
	$F = 0$		$F = 10$		$F = 25$		$F = 50$	
	t [m]	R [rad]	t [m]	R [rad]	t [m]	R [rad]	t [m]	R [rad]
C_1	0.038	0.331	0.042	0.001	0.010	0.0	0.058	0.015
C_2	1.022	0.351	0.029	0.001	0.013	0.0	0.038	0.002
C_3	0.961	0.332	0.019	0.002	0.019	0.0	0.025	0.0
C_4	0.481	0.316	0.033	0.001	0.035	0.0	0.020	0.001
average	0.626	0.333	0.031	0.001	0.019	0.0	0.035	0.005

Table 4.5: This table compares the influence of frame skip on both naive and filtered re-ID, using 200 computations on dataset SALSA. For example, $F = 50$ evaluates till frame $f = 10000$ and $F = 0$ evaluates 200 frames.

4.5. Experiment 3: Automatic versus annotated re-ID

This experiment applies the auto-calibration method to the dataset WildTrack and compares automatic versus annotated re-ID. The research sub-question for this experiment is: "Does automated re-identification negatively affect the calibration accuracy compared with calibration using annotated re-identification?". Even though the last experiment (Section 4.4) concluded that the filtered re-ID achieves better calibration accuracy compared to naive re-ID, both methods are used for the dataset WildTrack, testing if this behavior is consistent over multiple datasets. The re-ID methods will be tested with and without frame skip. In the annotated re-ID case, the best calibration result is chosen for comparison, multi-person human-pose calibration with $f = 300$ and the full human-pose configuration. This configuration had RMS errors of 0.020 meters in translation and 0.0 in rotation.

4.5.1. Full evaluation

As in the previous section (Section 4.3), the calibration was limited due to a memory shortage. For the calibration, only 100 evaluations were possible. Therefore the frame length is set to $f = 100$.
The dataset WildTrack has almost twenty times more participants than the dataset SALSA, which could potentially compensate for only having 100 evaluations to calibrate with.

Calibration on dataset WildTrack, with the naive and filtered re-ID methods, showed poor calibration accuracy. Figure 4.5 visualizes the difference in calibration between naive re-ID and annotated re-ID. With an average RMS error of 30.28 meters and 0.7 radians, only six of the seven cameras were found in the camera network. The auto-calibration model with the filtered re-ID method could not calibrate at all.

4.5.2. Evaluation with frame skip

This test investigates if adding frame skip shows a rise in efficiency on the auto-calibration using WildTrack. Both tests are done by re-ID methods naive and filtered with a frame skip of $F = 25$. The

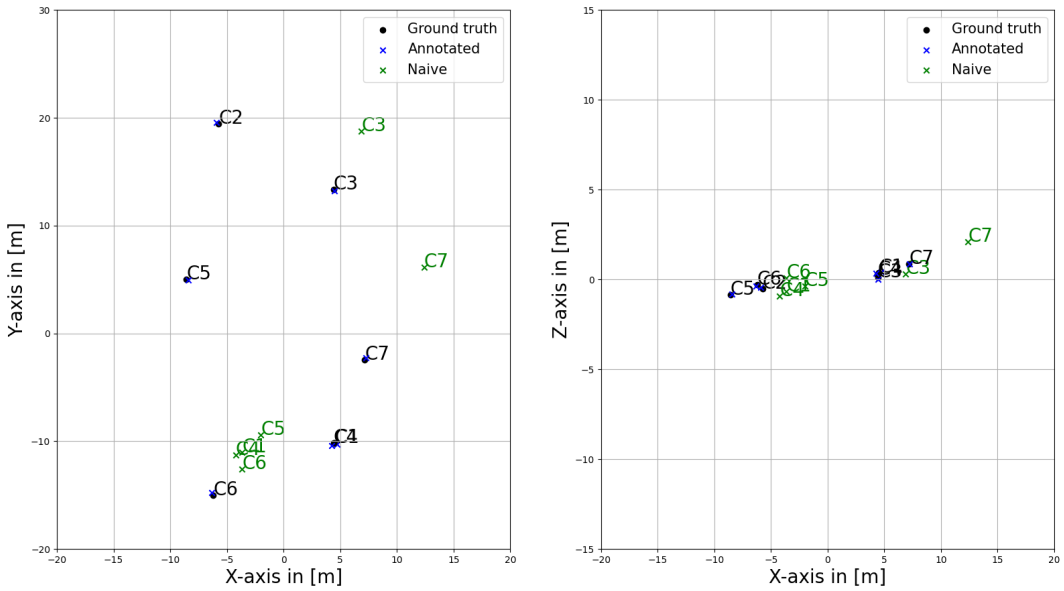

Figure 4.5: The area plot of WildTrack, using $f = 100$ and the naive re-ID method versus the annotated re-ID. The annotated re-ID plot does not have camera names plotted because it closely resembles the ground truth. Both plots do not have an equal range.

	Naive re-ID		Filtered re-ID	
	translation error [m]	rotation error [rad]	translation error [m]	rotation error [rad]
C_1	22.51	0.476	-	-
C_2	-	-	-	-
C_3	11.71	0.795	-	-
C_4	25.51	0.471	-	-
C_5	84.16	0.827	-	-
C_6	4.02	0.824	-	-
C_7	34.37	0.856	-	-
average	30.38	0.708	-	-

Table 4.6: Root mean squared error of the translation and rotation, on dataset WildTrack with frame length $f = 100$

maximum frame length for this test is set to $f = 1000$. After 1000 frames, the PoseFlow tracking algorithm led the GPU to run out of memory.
The evaluation of dataset SALSA showed an improvement when frame skip was added to the auto-calibration method with the same number of computations. However, the previous Section had 100 evaluations, and this test will have 40. Expected is that it will not significantly affect the calibration accuracy.

In Figure 4.6 and Figure 4.7 the area plots are presented and the RMS errors are displayed in Table 4.7. The naive re-ID could calibrate all cameras in the network, although it suffered an increase in average RMS error in the translation of 24 meters. The automatic calibration could output three extrinsic parameters with an RMS error of 0.022 in translation and 0.668 radians in rotation when using the filtered re-ID method.

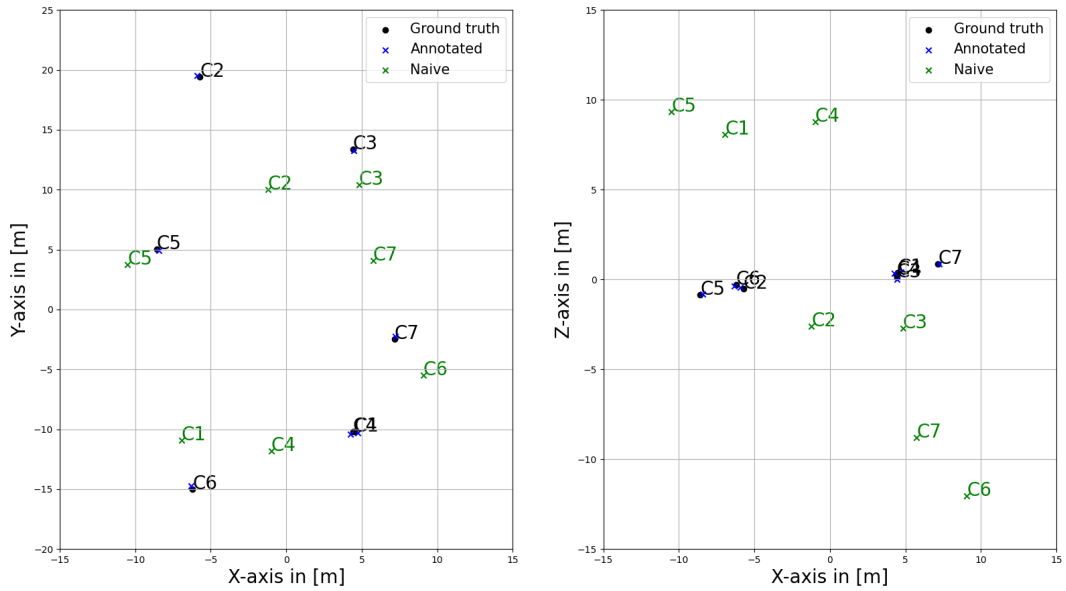

Figure 4.6: The area plot of WildTrack, using $f = 1000$ and the naive re-ID method with frame skip $F = 25$ versus the annotated re-ID. The annotated re-ID plot does not have camera names plotted because it closely resembles the ground truth. Both plots do not have an equal range.

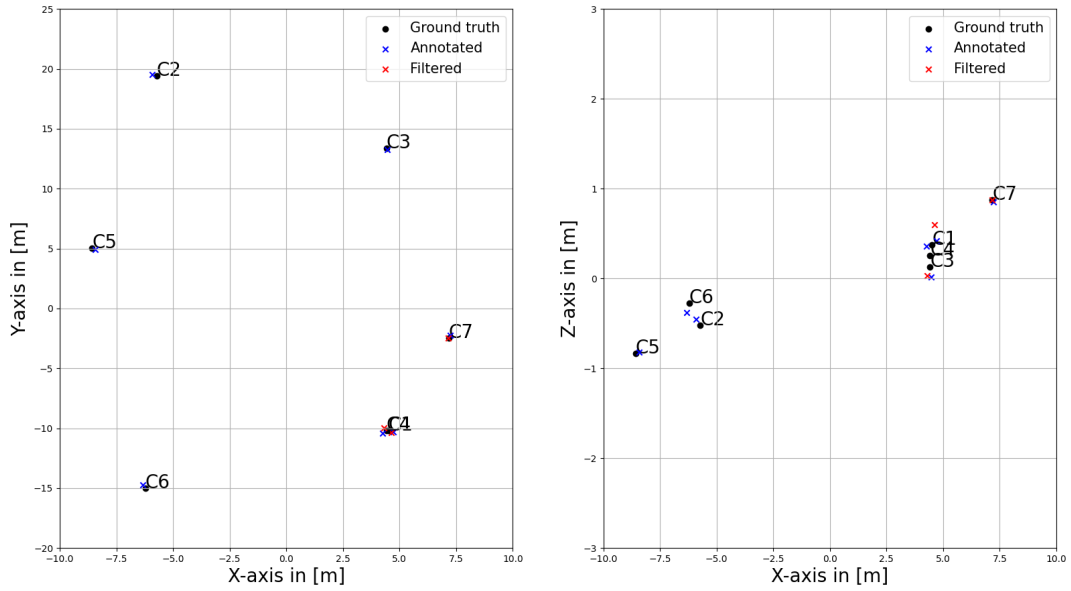

Figure 4.7: The area plot of WildTrack, using $f = 1000$ and the filtered re-ID method with frame skip $F = 25$ versus the annotated re-ID. Only the ground truth camera names are plotted because all the annotated and filtered re-ID closely resemble the ground truth.

	Naive re-ID F = 25		Filtered re-ID F = 25	
	translation error [m]	rotation error [rad]	translation error [m]	rotation error [rad]
C_1	63.55	0.695	0.033	0.589
C_2	37.79	0.758	-	-
C_3	5.643	0.751	-	-
C_4	34.73	0.591	0.034	0.569
C_5	36.25	0.881	-	-
C_6	154.1	0.740	-	-
C_7	46.10	0.701	0.000	0.845
average	54.02	0.731	0.022	0.668

Table 4.7: Root mean squared error of the translation and rotation on dataset WildTrack for $f = 1000$, comparing the naive and filtered re-ID methods when frame skip 25 is added.

These tests conclude the third experiment. In the full evaluation test of this experiment, naive and filtered re-ID could not find the extrinsic camera parameters accurately. The second test applied frame skip, but the improvement of the calibration was minor. The filtered re-ID in combination with frameskip was the only calibration with accurate position estimates. Comparing these calibration accuracies with the calibration accuracy of annotated re-ID showed that, with dataset WildTrack, this model was incapable of calibrating the extrinsic camera parameters; this answers the third research sub-question.

This experiment clearly shows the limitation of the proposed auto-calibration model. When the auto-calibration model is deployed on dataset WildTrack, it could not extract the full extrinsic calibration information. A GPU memory shortage stopped the computation of the automatic re-ID step as soon as 100 computations, potentially being too little data for accurate calibration.

4.6. Summary

This chapter presents and discusses the results of the experiments designed in the methodology chapter. The results will help to answer the research questions from Section 1.2.
Section 4.1 presented both datasets WildTrack and SALSA. Both are publicly available and are used in the experiments as input data. In section 4.2 the hyper-parameters of the auto-calibration model are described.
In section 4.3, two tests are done. The first test measures the calibration accuracy of the single-person human-pose calibration versus the multi-person human-pose estimation, re-ID is done with the annotators of the dataset. The influence, of the number of human-pose key points in four configurations, on the calibration accuracy, is analyzed. These two tests answer the first research sub-question.
Section 4.4, experiment 2, analyzes the automatic re-ID methods, naive and filtered re-ID. The first test examines their ability to discard mismatching errors by comparing the extrinsic camera parameters to the ground-truth ones. The second test adds frame skip to the re-ID methods. Four values of frame skip were used and compared calibration accuracy to each other. These two tests answer the second research sub-question.
Section 4.5 is experiment 3 and compares the automatic re-ID with annotated re-ID. Two calibrations are done, one with frame skip and one without, and their calibration accuracy is measured. These tests answer the third research sub-question.

5

Conclusions

This research presented an automatic extrinsic calibration model, using human-pose estimation as a feature extractor and an automated re-ID algorithm for matching these estimations.
With the experiments of the previous Chapter 4, the properties of this model were analyzed, and the research sub-questions were answered. This chapter gives a summary of the results and concludes the main research question. Section 5.1, discusses the human-pose-based calibration, without automatic re-ID. Section 5.2 examines the properties of the automatic re-ID methods. The comparison between automatic re-ID and annotated re-ID is made in Section 5.3. Each research sub-question is answered; thus, the following Section 5.4 concludes this research by answering the main research question. Section 5.5 gives recommendations for future work.

5.1. Human-pose calibration

The first research sub-question was: "*When assuming that re-identification across frames is correct, how does a single-person human-pose estimation versus a multi-person human-pose estimation affect the calibration accuracy and is the accuracy affected by the number of human-pose key points?*". This research sub-question led to Experiment 1: Human-pose-based calibration (Section 4.3). In the first analysis of this experiment, single-person human-pose calibration versus multi-person human-pose calibration is compared on their calibration accuracy. Multi-person human-pose calibration is an extension to the works of [26], [19], which are both single-person human-pose calibrators. This test used the person with the longest path in the dataset for the single-person case, and the multi-person case had access to all people in the dataset WildTrack. The result of this calibration was a near-perfect calibration accuracy for the multi-person case (RMS errors of 0.02 meter and 0.0 rad) and a poor calibration for the single-person case (RMS errors of 10.70 meter and 0.052 rad).

The second part of the first research sub-question analyzed the number of human-pose key points and their influence on the model's accuracy. Four configurations of key points were proposed, Full human-pose with 26 key points, Simple with five key points, Vertical Stick with two key points, and Bounding box with zero key points. The Bounding box configuration uses the line's center as correspondence points, skipping the human-pose estimation.

The first test shows that including more people in the calibration model benefits calibration. The other analysis showed that configurations including more key points contributed to better calibration accuracy, whereas the Bounding box configuration had the poorest calibration accuracy. These two results answer the first research sub-question.
The limitation of this experiment is the difference between a controlled environment and the "in the wild" environment. This experiment was an "in the wild" environment and therefore could not promise the same boost in accuracy when expanding to multi-person in a controlled environment. A second limitation was found when calibrating, the GPU ran out of memory after 300 frames during human-pose tracking. This meant that not all 400 (annotated with global labels) frames were evaluated.

5.2. Automatic re-ID

The second research sub-question was: "*In automatic re-identification, what is the effect of evaluating each subsequent frame versus skipping frames on the calibration accuracy, and how are mismatching errors handled?*". This experiment, Experiment 2: Automatic re-ID (Section 4.4), addresses this research sub-question through two tests, analyzing the naive and filtered re-ID methods. For this experiment, dataset SALSA was used.
With the first test, the performance was analyzed of both re-ID methods by comparing the calibration accuracy over varying frame lengths. Comparing calibration accuracy over varying frame lengths was done to determine how resilient both methods were to mismatching errors when more data was used for calibration. After 800 evaluations, the translation and orientation plots showed an increased RMS error.

In the second test of this experiment, frame skip was added to both re-ID methods. Frame skip was added so that not each subsequent frame was evaluated. The values for frame skip were: [0, 10, 25, 50], and the calibration was done for 200 evaluations. The calibration accuracy was compared against all values of frame skip, determining if a better calibration accuracy could be achieved with the same number of evaluations. The naive method did not improve when frame skip was added, and it was only able to get a slightly better calibration at $F = 50$. The filtered method did improve significantly, lowering its RMS errors in translation and orientation from 0.626 meters and 0.333 radians to 0.019 meters and 0.0 rad. This calibration accuracy was the best result for the filtered re-ID method with frame skip $F = 25$.

The mismatching errors were handled in the first test, showing that the influence of mismatching errors was seen in the naive re-ID method. After 800 evaluations, the calibration accuracy dropped compared to filtered re-ID. Filtered re-ID showed no drop in calibration accuracy, concluding that the temporal filter was capable of mitigating the effect of mismatching noise in this test. The second test analyzed the effect of evaluating each subsequent frame versus frame skip. It showed that when a filter for mismatches is added, frame skip positively affects the calibration accuracy. These two results answer the second research sub-question.
The limitation of this experiment is that comparing two re-ID methods only shows that there is mismatching noise. However, this experiment does not show the origin of the mismatching errors, like if they originate from ID switches in the PoseFlow tracker step or if the feature extraction step of OSNet-AIN creates this noise.

5.3. Automatic versus annotated re-ID

The third research sub-question was: "*Does automated re-identification negatively affect the calibration accuracy compared with calibration using annotated re-identification?*". This research sub-question led to this experiment, Experiment 3: Automatic versus annotated re-ID (Section 3.2.3), examining the automatic re-ID methods against annotated re-ID. This experiment used the dataset WildTrack for the annotated global labels.
The first test compared the calibration accuracy of annotated re-ID against naive and filtered re-ID without frame skip. The annotated re-ID's best calibration accuracy from Experiment 1: Human-pose calibration is used to compare the automated re-ID methods.
Annotated re-ID achieved an RMS error in translation of 0.020 meters and 0.0 radians in orientation. The automatic re-ID methods, however, do not have this high calibration accuracy. Both filtered and naive re-ID could not calibrate the entire camera network of WildTrack. The poor results were due to a shortage of GPU memory, causing the calibration to stop. The filtered re-ID could not calibrate any cameras with only 100 evaluations ($f = 100$). The naive method calibrated six of the seven cameras, albeit with an average RMS error in translation of 30.38 meters.

The second test added frame skip to the re-ID methods. The best frame skip value of the previous experiment was chosen $F = 25$ and the maximum possible frame length $f = 1000$ was used.
The improvement of the calibration accuracy was minor. The naive re-ID method could now find all cameras in the network, but its translation error worsened by 24 meters. Filtered re-ID could now accurately determine the position of three cameras in the network. The orientation RMS error, however, was still poor.

The first test showed that annotated re-ID was far superior to automatic re-ID calibration in calibration accuracy. In the second test adding frame skip to improve the calibration accuracy made minor improvements. These tests have answered this research sub-question by showing that automatic re-ID negatively influences the calibration accuracy compared to annotated re-ID.
The limitation of this experiment was a shortage of GPU memory. Experiment 1: Human-pose calibration, did have this limitation as well, but in this experiment, it killed the GPU in the automatic re-ID step after 100 evaluations. This meant that the calibration was done with 40 and 100 evaluations, which appeared to be too few. A potential reason why this shortage in memory did not happen in the calibration with dataset SALSA, is that dataset WildTrack has more people walking the scene and has a higher resolution compared to WildTrack (Table 4.1).

5.4. Conclusion

The main research question of this research stated: "*How accurate is an automated extrinsic camera calibration model, using the human-pose estimation as features extractor and an automatic re-identification algorithm when comparing the extrinsic camera parameters to the annotated calibration information?*". Two datasets have been used to calibrate the extrinsic camera parameters, SALSA and WildTrack. The best calibration accuracy of the proposed auto-calibration model, using the dataset SALSA, achieved RMS errors of 0.019 meters in translation and 0.0 radians in orientation (Table 4.5). Which is a near-perfect calibration.
When the auto-calibration model was deployed on the dataset WildTrack, the model could only calibrate three cameras in the network. Filtered re-ID combined with frame skip $F = 25$ (Table 4.7) achieved an RMS error in translation of 0.022 meters and 0.668 radians in orientation.
The reason why the calibration of WildTrack was poor is because of the limitation seen in two experiments (Experiment 1: Human-pose calibration in Section 4.3 and Experiment 3: Automatic versus annotated re-ID in Section 4.5), which was GPU memory shortage. Due to this shortage, the calibration with dataset WildTrack did not get enough data, which appeared to be too few data for accurate calibration.

The goal of this research was to ease the implementation of CCTV-based tracking by proposing an automated extrinsic calibration model and investigating its characteristics. Although the proposed model only calibrated one of the two datasets, it provides a baseline for future research in multi-person automatic extrinsic camera calibration.

5.5. Recommendations

In this section, the proposed recommendations for future work are summed up to improve this model.

- Calibration on WildTrack with this model is possible when the computational power of the hardware is increased. Future work could then conclude if the proposed method can calculate the extrinsic camera parameters, better than this research could.
- The automatic re-ID methods do not incorporate prior knowledge about their location and relation to the other cameras in the network. Therefore, all detections from all viewing angles are compared to re-ID humans. If the camera combinations that overlap are used as input in the model, then the model could exclude camera combinations that do not overlap. Adding this prior knowledge could reduce run time in the automatic re-ID step, possibly reducing the computational power of this step.
- Mismatching errors could be reduced if there is a better understanding of their origin in the proposed model. Determining if the tracks from PoseFlow create more mismatching errors than the OSNet-AIN feature extractor could lead to better filtration methods.

Appendices

A

Hyper parameters

A.1. Detection accuracy threshold

Figure A.1 shows the detection accuracies θ_{da} from 0.0 to 0.9 with incremental steps of 0.1 plotted against the calibration accuracy. The calibration setup was multi-person human-pose calibration with the filtered re-ID method at $f = 1000$.
Detection accuracy 0.7 had the best calibration accuracy.

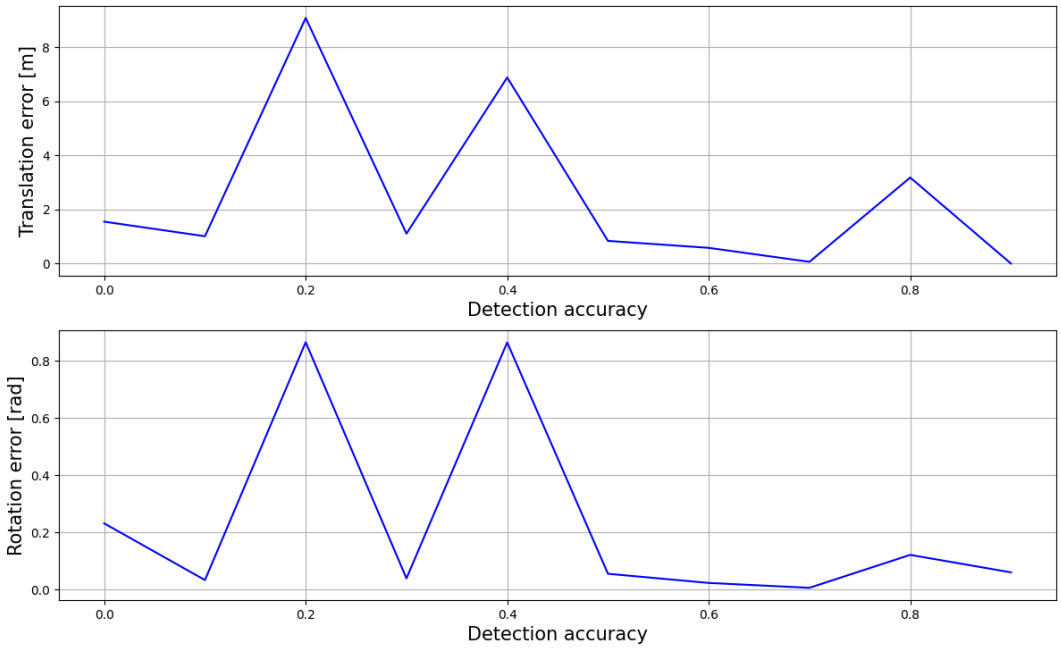

Figure A.1: The calibration accuracies are plotted against the detection accuracies.

A.2. Re-identification threshold

Table A.1 shows three values for the re-ID threshold parameter θ_{REID} 0.75, 0.8 and 0.85, with their calibration accuracy. The calibration setup was multi-person human-pose calibration with the filtered re-ID method at $f = 1000$.
Re-ID threshold parameter 0.8 had the best calibration accuracy.

	$\theta_{REID} = 0.75$		$\theta_{REID} = 0.8$		$\theta_{REID} = 0.85$	
	trans [m]	rot [rad]	trans [m]	rot[rad]	trans [m]	rot[rad]
C_1	1.182	0.012	0.048	0.003	0.252	0.152
C_2	0.855	0.072	0.144	0.005	-	-
C_3	0.359	0.051	0.056	0.011	0.331	0.476
C_4	0.672	0.017	0.008	0.009	0.38	0.788
average	0.767	0.038	0.064	0.007	0.321	0.472

Table A.1: Calibration accuracies with varying re-ID threshold parameters.

Bibliography

[1] Xavier Alameda-Pineda et al. “SALSA: A Novel Dataset for Multimodal Group Behavior Analysis”. In: *CoRR* abs/1506.06882 (2015). arXiv: 1506.06882. URL: http://arxiv.org/abs/1506.06882.

[2] Gil Ben-Artzi. “Camera Calibration by Global Constraints on the Motion of Silhouettes”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 5344–5353. DOI: 10.1109/ICCV.2017.570.

[3] Gil Ben-Artzi et al. “Camera Calibration from Dynamic Silhouettes Using Motion Barcodes”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4095–4103. DOI: 10.1109/CVPR.2016.444.

[4] Gil Ben-Artzi et al. “Epipolar geometry based on line similarity”. In: Dec. 2016, pp. 1864–1869. DOI: 10.1109/ICPR.2016.7899908.

[5] Chahat Deep Singh. “Structure from Motion”. In: *CMSC426 Computer Vision* (2021). URL: https://cmsc426.github.io/sfm/.

[6] Tatjana Chavdarova et al. “WILDTRACK: A Multi-camera HD Dataset for Dense Unscripted Pedestrian Detection”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5030–5039. DOI: 10.1109/CVPR.2018.00528.

[7] Muchun Chen et al. “Real-Time Multiple Pedestrians Tracking in Multi-camera System”. In: Jan. 2020, pp. 468–479. ISBN: 978-3-030-37730-4. DOI: 10.1007/978-3-030-37731-1_38.

[8] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[9] Haoshu Fang, Shuqin Xie, and Cewu Lu. “RMPE: Regional Multi-person Pose Estimation”. In: *CoRR* abs/1612.00137 (2016). arXiv: 1612.00137. URL: http://arxiv.org/abs/1612.00137.

[10] Tavi Halperin and Michael Werman. “An Epipolar Line from a Single Pixel”. In: (Mar. 2017).

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004.

[12] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[13] Li He et al. “Efficient Multi-View Multi-Target Tracking Using a Distributed Camera Network”. In: *IEEE Sensors Journal* 20.4 (2020), pp. 2056–2063. DOI: 10.1109/JSEN.2019.2949385.

[14] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. “EPnP: An accurate O(n) solution to the PnP problem”. In: *International Journal of Computer Vision* 81 (Feb. 2009). DOI: 10.1007/s11263-008-0152-6.

[15] Wei Li et al. “DeepReID: Deep Filter Pairing Neural Network for Person Re-identification”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 152–159. DOI: 10.1109/CVPR.2014.27.

[16] Wei Liu et al. “SSD: Single Shot MultiBox Detector”. In: *CoRR* abs/1512.02325 (2015). arXiv: 1512.02325. URL: http://arxiv.org/abs/1512.02325.

[17] D.G. Lowe. “Object recognition from local scale-invariant features”. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410.

[18] David Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60 (Nov. 2004), pp. 91–. DOI: 10.1023/B:VISI.0000029664.99615.94.

[19] Olivier Moliner, Sangxia Huang, and Åström Kalle. “Better Prior Knowledge Improves Human-Pose-Based Extrinsic Camera Calibration”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. 2021, pp. 4758–4765. DOI: 10.1109/ICPR48806.2021.9411927.

[20] Jens Puwein et al. “Joint Camera Pose Estimation and 3D Human Pose Estimation in a Multi-Camera Setup”. In: Nov. 2014. ISBN: 978-3-319-16807-4. DOI: 10.1007/978-3-319-16808-1_32.

[21] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *CoRR* abs/1804.02767 (2018). arXiv: 1804.02767. URL: http://arxiv.org/abs/1804.02767.

[22] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *CoRR* abs/1506.02640 (2015). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640.

[23] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *CoRR* abs/1506.01497 (2015). arXiv: 1506.01497. URL: http://arxiv.org/abs/1506.01497.

[24] Ergys Ristani et al. “Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking”. In: *CoRR* abs/1609.01775 (2016). arXiv: 1609.01775. URL: http://arxiv.org/abs/1609.01775.

[25] Sonia Phene. *Local Feature Matching*. URL: https://soniaphene.github.io/featurematching/.

[26] Kosuke Takahashi et al. “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)*. 2018, pp. 1856–18567. DOI: 10.1109/CVPRW.2018.00230.

[27] Anh Minh Truong et al. “Automatic Multi-Camera Extrinsic Parameter Calibration Based on Pedestrian Torsors †”. In: *Sensors* 19.22 (2019). ISSN: 1424-8220. DOI: 10.3390/s19224989. URL: https://www.mdpi.com/1424-8220/19/22/4989.

[28] Longhui Wei et al. “Person Transfer GAN to Bridge Domain Gap for Person Re-Identification”. In: *CoRR* abs/1711.08565 (2017). arXiv: 1711.08565. URL: http://arxiv.org/abs/1711.08565.

[29] Yuliang Xiu et al. “Pose Flow: Efficient Online Pose Tracking”. In: *CoRR* abs/1802.00977 (2018). arXiv: 1802.00977. URL: http://arxiv.org/abs/1802.00977.

[30] Wei Yang et al. “Learning Feature Pyramids for Human Pose Estimation”. In: *CoRR* abs/1708.01101 (2017). arXiv: 1708.01101. URL: http://arxiv.org/abs/1708.01101.

[31] Mang Ye et al. “Deep Learning for Person Re-identification: A Survey and Outlook”. In: *CoRR* abs/2001.04193 (2020). arXiv: 2001.04193. URL: https://arxiv.org/abs/2001.04193.

[32] Z. Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334. DOI: 10.1109/34.888718.

[33] Liang Zheng et al. “Scalable Person Re-identification: A Benchmark”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1116–1124. DOI: 10.1109/ICCV.2015.133.

[34] Kaiyang Zhou et al. “Learning Generalisable Omni-Scale Representations for Person Re-Identification”. In: *CoRR* abs/1910.06827 (2019). arXiv: 1910.06827. URL: http://arxiv.org/abs/1910.06827.

[35] Zhengxia Zou et al. “Object Detection in 20 Years: A Survey”. In: *CoRR* abs/1905.05055 (2019). arXiv: 1905.05055. URL: http://arxiv.org/abs/1905.05055.