PTZ Handover: Tracking an object across multiple surveillance cameras

Alexander Persson

LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6216
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2023

Abstract

Tracking objects in a scene is a crucial task in accomplishing surveillance that enhances security and provides valuable information about the events happening at the site. For this task, the PTZ (pan-tilt-zoom) cameras can be utilized to achieve fluid tracking as they provide all-around surveillance with zoom capabilities. The drawback of current tracking solutions is the lack of interoperability between cameras, e.g. to signal the position of an object so that multiple cameras can track it simultaneously. This project highlights the importance of continuously tracking an object across a site and proposes a solution on how to handover the target from one camera to another. Thus the need of performing PTZ coordinate transformation is necessary to direct multiple PTZ cameras toward the same target. For simplicity, the scope of the project was limited to a system consisting of only two cameras, with a focus on tracking one object at a time. The method consists of two steps: namely to perform a calibration procedure to determine the spatial relationship between two cameras and to then track a single object across a site. The tracking process is handled by a centralized server, which determines which objects to track, where to position the cameras and when to perform the handover.

The results show that tracking an object across two cameras, mounted at different heights and located multiple meters apart, is fully achievable. Even though the built-in tracker can be perceived as slightly delayed, the handover functionality still managed to execute as expected even with the target moving at a moderately high velocity. The output of the calibration was found to be rather satisfactory, but could however be refined to achieve even higher accuracy. In conclusion, the proposed solution works well and entails that this kind of functionality may further enhance all-around surveillance. As future work, the calibration procedure can easily be expanded to multiple cameras, but tracking multiple objects at the same time requires advanced theoretical investigation.

Acknowledgements

I would like to thank the PTZ department and ACS department at Axis Communication for their continuous support and feedback during my time at Axis. As well as all my colleagues across the whole company that have provided valuable insight for this project. Thanks to my supervisors Paul Steneram Bibby, Kenneth Ekman and Yiannis Karayiannidis for their quick feedback on my proposed solutions and this thesis. And lastly a special thanks to Per Wilhelmson for introducing this project to me.

Contents

List of Abbreviations

PTZ Pan-tilt-zoom

API Application programming interface

CGI Common gateway interface

VMS Video management system

ACS Axis Camera Station

ACAP AXIS Camera Application Platform

FOV Field of view

SIFT Scale Invariant Feature Transform

Chapter 1

Introduction

Surveillance cameras in today's society have become impressively good at detecting and tracking different forms of objects. Advanced object detection algorithms and automatic control theory enable cameras such as PTZ cameras (abbreviation for pan-tilt-zoom camera) to smoothly follow moving objects in a scene, thanks to their capability of horizontal and vertical movements.

One of the drawbacks of the tracking solution that exists today is the lack of interoperability between cameras. For the surveillance to become even better there needs to be a form of consistency when tracking objects, as to keep them under constant observation. But if an object were to disappear, e.g. behind a wall, there would be some loss of surveillance. A way to counter this is by installing more cameras to cover the whole area of interest, albeit this may not be sufficient if the cameras are not communicating with each other about the position of the object as it still may get lost at the site.

This thesis project aims to improve existing tracking methods by performing a so-called "handover" procedure, meaning that a camera is to hand over the tracking responsibility to another camera based on predefined conditions. This procedure will provide a guarantee that the object will be under constant observation across the site assuming that the cameras are well placed in the surveilled area. The conditions that the handover may depend on are object position and visibility as well as the internal state of the cameras.

1.1 Previous work

For solving the problem of constant all-around surveillance it is first necessary to examine the work previously done in the area of PTZ cameras and tracking. These

may give a hint on how to combine them into a greater control method for how the cameras are to behave and relate to each other. The following sections are the foundation of the proposed solution for this project.

1.1.1 Calibration of multiple PTZ cameras

In the paper [2] the authors present a method on how to infer the spatial relation between two cameras, mainly the relative positioning and orientation. The proposed method assumes that two papers with known side lengths are visible from multiple cameras' current field of view. By then marking the corners of the papers in the camera image, the side lengths can be measured in pixel units to get the projected length in the image plane. Comparing these projected lengths from different perspectives with the known real-world length will result in estimations of a camera's positioning and orientation, as well as their spatial relation to each other. The resulting estimation can thus be used for translating pixel coordinates between multiple cameras.

1.1.2 Real-time tracking using a PTZ camera

The authors in the paper [4] propose a tracking system that combines object detection, tracking, and PTZ control to achieve a robust and accurate tracking approach. The PTZ camera is controlled in real-time to actively adjust its position and zoom level in order to keep the target object within the camera's field of view. More specifically, the goal of the tracking is to

1. keep the target object in the center of the image by controlling the pan and tilt angles.

2. keep the projection of the target at a proper size by controlling the zoom level.

The method to achieve this is to use regulators for each of the axis (pan, tilt and zoom), which uses the focal length and the object's center position in the image as input. The controllers will then regulate the pan and tilt speeds and change the current focal length to achieve the goals. The authors found that their tracking approach worked well to form a real-time system that could track arbitrary objects in the PTZ view, with the motivation that it produced better results than current tracking methods at that time.

1.2 Problem formulation

This master's thesis project proposes a solution that allows for the tracking of one object across multiple PTZ cameras. By defining the location of the cameras, one should be able to design a system that will utilize the cameras in capturing the object as it makes its way through the site, in a smooth manner. An illustration of the problem can be seen in Figure 1.1, where camera 1 is to give over the tracking responsibility to camera 2.

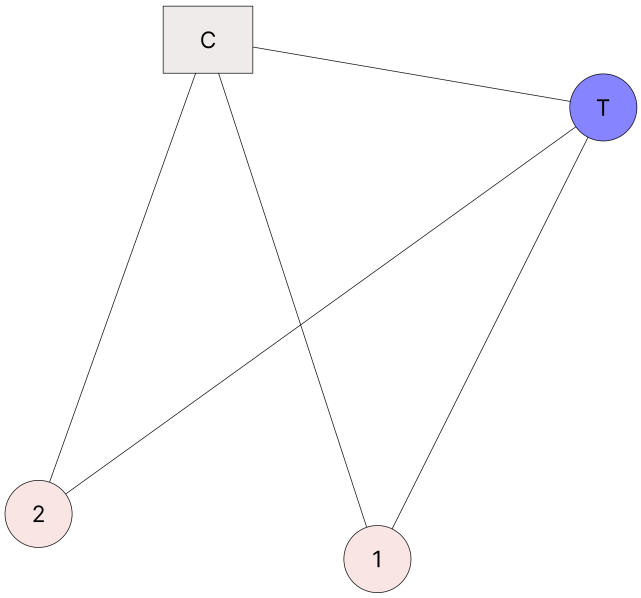


Figure 1.1: Illustration of the handover, where camera 1 is to give over the tracking responsibility for the tracked object T to camera 2 by using the common point C (known as the calibration target).

The proposal is to combine the calibration of cameras and tracking objects into one novel control method. The following goals and topics that are to be investigated in this project are:

a. to perform a calibration procedure for estimating the position and orientation of the PTZ cameras. Questions that this entail is what kind of calibration targets to use with concern to the advantages and disadvantages. This will be the foundation for calculations in the handover process.

b. to transform PTZ coordinates from one camera to another. The coordinates of an object are expressed differently for each camera, as they are dependent on the spatial position of the camera. This will allow a camera to communicate where the object is located by expressing its position in another coordinate system.

c. to perform handover based on predefined conditions. The time point to perform handover needs to be defined and the consequences for how this will affect the tracking system need to be considered.

d. to track an object. It is important to always have at least one camera that tracks the object to achieve constant surveillance.

1.3 Limitations and assumptions

Several limitations and assumptions are needed for this thesis project to facilitate the solution and reduce the scope of the project.

1.3.1 Camera position

It is assumed that the cameras in the system are mounted in such a way that their pan-planes are parallel to each other. Meaning that the only rotation that is allowed is alongside the y-axis. This is to assure that all pan angles used in the project are calculated on the same horizontal plane. Any pan error offset observed in the results may be because of faulty mounted PTZ cameras, as it is considerably hard to correctly mount the cameras.

1.3.2 Amount of objects to track

Allowing the system to track multiple objects at once would require a form of planning strategy, as illustrated in [5]. This paper formulates an approach to performing PTZ camera assignments and handoffs, based on optimal camera assignments and predefined observational goals. The ability to plan enables surveillance systems to achieve secure and continuous recordings of objects at a site, but at the cost of more complex systems. Thus a restriction for this solution proposal is that only one object will be tracked by the system at a given time. If the object leaves the area, then a new one may enter without imposing any problem.

1.3.3 Amount of cameras

To limit the development time the system will only consist of two cameras, meaning that when the handover process is called there is only one other camera to hand over the object to. This assumes that both cameras have a common scenery to perform handover in. The solution can be expandable into a system of multiple if the calibration is done pairwise, a strategy discussed more in detail in section 6.1.

1.4 Contributions

Building upon the previous work done in [2] and [4], this thesis aims at contributing with an expansion to conventional tracking. This expansion, the handover process, is meant to improve all-around surveillance by guaranteeing that an object is always under constant surveillance at a site. This entails defining an area where to hand over the tracking responsibility from one camera to another and in which circumstance to perform this action. Also, to be able to refer to the position of the tracked object between multiple cameras, calculations which may be adapted for the specifics of this project.

Chapter 2

Background

2.1 Pan-tilt-zoom camera

The PTZ camera is a type of surveillance camera that can be controlled to move horizontally and vertically, with functionality for zooming and are generally used for monitoring larger areas, as they provide all-around surveillance with zoom capabilities. Examples of larger sites to monitor are industrial sites, parking lots and people-dense areas. The PTZ camera can either be moved manually by an operator or automated to move depending on events, such as scheduled time events or visual movement events.

For the rotational position alongside the horizontal and vertical axes, hereafter titled pan and tilt respectively, the range for pan is commonly $[-180°, 180°]$ while for tilt it is $[20°, -90°]$ (where $-90°$ denotes the camera facing straight down). Other sophisticated hardware features found in PTZ cameras are infrared night vision and a built-in laser, for precise focusing and measuring the distance to objects in the center of the view.

One of the main features of the PTZ camera is its ability to adjust the motorized lens to change both the focus and the zoom of the view. This allows the camera to get closer to an object, increasing the details of the image and thus allowing for better pictures and analytics. The range for zoom is customarily divided into two ranges, [0, 9999] for optical zoom and [10000, 19999] for digital zoom, these ranges are scaled depending on the magnification of the lens for the given PTZ camera. The term 'wide' is often used in the context of zoom as this signifies that the field of view is at its widest, meaning the camera is zoomed out to its fullest.

Advanced PTZ cameras can also be equipped with software features, usually called

analytic applications, such as facial recognition, image stabilization and tracking movable objects. These applications increase the camera's surveillance ability, making them more effective for security purposes.

The PTZ camera *Q6315-LE*[1] (seen in Figure 2.1), developed by Axis Communication, is one of the newest high-end PTZ cameras used in today's market. With 31× optical zoom, a built-in laser and quick-zoom functionality, the camera allows for continuous and reliable tracking of movable objects. It offers steady 1080p streams at 60 frames per second.

Figure 2.1: AXIS Q6315-LE PTZ Network Camera.

Controlling and communication with an Axis PTZ camera can be done through its VAPIX API interface[2], which provides the user with functionality such as requesting images, controlling motors or retrieving and changing settings. Communication through the API interface is typically done through CGI (Common Gateway Interface) requests, i.e. HTTP method with a payload deciding the action the camera shall perform. The cameras can also alert on specific events, such as motion, sound or external tampering if configured beforehand.

2.2 Video Management System

A Video Management System (VMS) is a software platform used to manage security cameras by mainly handling the recording of video footage captured by the cameras. VMSs are generally used by more preponderant customers such as retail stores or

[1] AXIS Q6315-LE PTZ Network Camera `https://www.axis.com/products/axis-q6315-le`
[2] Axis VAPIX library `https://www.axis.com/vapix-library/`

manufacturing industries, where the need for a broad and covering surveillance ability is crucial. It can manage and store the recorded footage and data from network cameras, facilitating the search and retrieval of video footage for investigations or analysis. Sophisticated VMSs may also provide analytic applications for tracking objects or recognizing attributes of interest (e.g. facial recognition).

By acting as a centralized system of network cameras, the VMS provides the opportunity for communication between cameras. If configured, events generated in a camera could cause another camera to respond in a desirable way, such as capturing video footage of an object from multiple perspectives. A VMS can provide real-time alerts and notifications based on events from the camera, allowing operators to quickly respond to potential threats.

PTZ cameras can be connected to a video management system, allowing for remote access and control. The movement of the PTZ cameras can be automated based on conditions specified by the user, including preset rules, motion detection or scheduled events. This sort of automation can thus be used to capture video footage at distinct periods and areas, or on specific events.

A concrete VMS is the *AXIS Camera Station*[3] (ACS). This product is specially developed and optimized for Axis network products and comes with a range of analytic applications.

2.3 Camera tracking

One of the most used analytics applications on the PTZ cameras is object tracking. Real-time tracking of objects can be realized by adjusting the camera's orientation and zoom level, depending on the position and size of the object in the real world. The broad range of the pan and tilt axes permits the camera to follow the object's movement across large areas, such as a parking lot, warehouse and outdoor environment. By utilizing its zoom capability, the camera image can be changed to cover the entire object in its field of view, allowing for the capturing of finer details.

There are multiple methods of detecting an object in the current view of a camera, with the two prominent methods used in the industry being motion detection and object analysis. Motion detection builds on the technology of comparing sequencing frames for determining motion, and can thus be considered a fast but primitive method[4]. In contrast, object analysis is usually built on advanced AI models for

[3]AXIS Camera Station `https://www.axis.com/products/axis-camera-station`

[4]AXIS Video Motion Detection
`https://www.axis.com/products/axis-video-motion-detection`

recognizing and categorizing objects, for example into humans, bicycles or cars, giving more accurate object detection but at the cost of heavier computations[5].

A portion of the PTZ cameras developed by Axis Communications comes with a pre-installed tracking application called *Autotracker*[6]. It can either use motion detection or object analysis depending on the hardware and utilizes PD controllers for smoothly following the object in pan and tilt space. Through its API it is possible to both query status and information requests, as well as for controlling the tracker function and its settings. Section 1.1.2 describes the foundation for how tracking can be implemented in a PTZ camera to track an arbitrary object. The paper [4], described in the section, implements P-controllers with truncated negative feedback for achieving the goal of keeping the object in the center of the image with a suitable level of zoom. The *Autotracker* is implemented similarly, with the main distinction being that it instead uses PD controllers, to further improve stability and faster response that gives better disturbance rejections and enhanced error corrections.

2.4 Computer vision

Computer vision is a field of study that focuses on gaining an understanding of the visual information from images and how to manipulate them. The textbook *Computer Vision: Algorithms and Applications* [6] covers the basic of this field and explains concepts such as feature detection and image matching. Computer vision is an important aspect when relating the real world to the PTZ camera coordinate system. Using advanced algorithms it is possible to infer relations between PTZ cameras and their surrounding world. This section contains different areas within computer vision that is of importance for the solution.

The local 3-D world $[x_c, y_c, z_c]^T$ (commonly referred to as the local world coordinate system) has its origin in the center of the camera and is always relative to the position of the camera. From the view of the camera, the z-axis is the depth of the world, while the x-axis is the horizontal axis and the y-axis is the vertical axis. The image coordinate system $[x', y']^T$ has its origin in the top left corner of the image plane, with a maximum width and height of I_W respectively I_H.

2.4.1 Feature detection and matching

Feature detection and matching is a process of selecting distinctive details in an image, usually labeled as keypoint features, which can be used to describe the content

[5]AXIS Object Analytics `https://www.axis.com/products/axis-object-analytics`

[6]PTZ Autotracker API
`https://www.axis.com/vapix-library/subjects/t10175981/section/t10156132/display`

and is useful for finding locations on an image that is likely to match well with other images. The process can be divided into three stages: detection, description and matching features.

Feature detection algorithms typically operate by examining the image for regions that exhibit some unique property, such as a sudden change in color intensity or a distinctive texture pattern. These keypoint features are often described by the pixels surrounding the point of interest. In the feature description stage, each region around a keypoint feature is described in a more compact descriptor that can be matched against descriptors from other images depicting the same features. Examples of descriptors are edges, corners, contours and curves in the image. Feature matching is then to efficiently compare and analyze the descriptors in search of a match between images. Different strategies may be used depending on the application, as some may search for overlap in an image while others may be trying to find a certain object in an image.

There are many different feature detection and matching algorithms available, suitable for different applications depending on the need. Some of the most commonly used techniques include the Harris corner detector, the Speeded-up robust features (SURF) detector and Lowe's Scale Invariant Feature Transform (SIFT) [6]. These algorithms can be applied to a wide range of image data, from natural scenes and landscapes to industrial and urban scenes.

2.4.2 Contours and line segments

Contours (or curves) refer to the boundaries or outlines of objects in an image and can be defined as a group of continuous edges around an object of interest [6]. They are typically extracted using edge detection algorithms, e.g. by using the Canny edge detection algorithm that extracts especially sharp edges [6]. Using contours may help in facilitating the algorithms that are tasked with feature detection or feature matching, thus enabling more accurate algorithms for the identification of objects in an image.

A line segment is defined as a part of a line that is bounded by two points A and B and is denoted as $\overline{AB}$. It is a straight path between the points with a definite length that can be measured, unlike a line that extends infinitely in both directions. Multiple line segments can be used to define geometrical entities such as shapes and angles.

2.4.3 Homography matrix

A homography matrix, also known as a perspective transformation matrix or homography, is a mathematical matrix that can be used for 2-D transformations between two planes as described in [6]. The matrix essentially performs a mapping between the two planes, allowing for quick calculation when converting coordinates from one frame to another. This projective method also has the benefit of preserving straight lines after transformation. The equation can be written in the form

$$\tilde{x}' = H\tilde{x}, \tag{2.1}$$

where H is an arbitrary 3×3 matrix while $\tilde{x}'$ and $\tilde{x}$ are coordinates in different frames.

The algebraic operations performed by the homography matrix include scaling, translation, and rotation of a 2D image. The scaling is needed as the x and y coordinates systems between the planes are different, the translation is for adding an offset to the coordinates to shift them into the correct position while rotation is needed as the planes may be rotated with concern for each other. The homography matrix is useful for tasks that involve aligning or overlaying images, as it in practical terms is a method of mapping points from one image to another image, under the assumption that both images depict the same scene from different perspectives. It is thus used as a bijective function for mapping image coordinates across images.

2.4.4 Camera intrinsics

Camera intrinsic parameters are a set of coefficients that describe the internal properties of a camera, such as its focal length, skewness of the image and principal point [6]. These parameters are usually part of a bigger representation of a camera's properties, known as the camera matrix, which relates the camera to the 3-D world. This matrix is essential for the calibration of a camera and plays a crucial role in computer vision tasks such as object recognition, camera tracking, and 3-D reconstruction, and can be expressed as

$$\mathbf{P} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \tag{2.2}$$

where $\mathbf{P}$ is the camera matrix, consisting of the intrinsic camera matrix $\mathbf{K}$ and the extrinsic camera matrix $[\mathbf{R}\ \mathbf{t}]$. The extrinsic camera matrix $[\mathbf{R}\ \mathbf{t}]$ represents the camera's orientation in 3-D space, with $\mathbf{R}$ as the rotational matrix and $\mathbf{t}$ as the transformation matrix.

The intrinsic camera matrix **K** is given by

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

where f_x and f_y are the focal lengths for the sensor dimension x and y, s is the skew factor and the center of the image is denoted as (u_0, v_0), as seen in Figure 2.2.

The focal length f of a camera is the distance between the lens and the image sensor when the lens is focused at infinity. This parameter determines the magnification of objects in view and is typically expressed in millimeters. The skew factor s is used to denote the possible skew of the sensor axes if the sensor is not mounted perpendicular to the optical axis, and is often commonly assumed to be zero. The center of the image, or in some literature denoted as the principal point, is the point where the optical axis intersects the image place. It is normally set as $(u_0, v_0) = (\frac{I_W}{2}, \frac{I_H}{2})$ where I_W and I_H are the image width and height respectively.

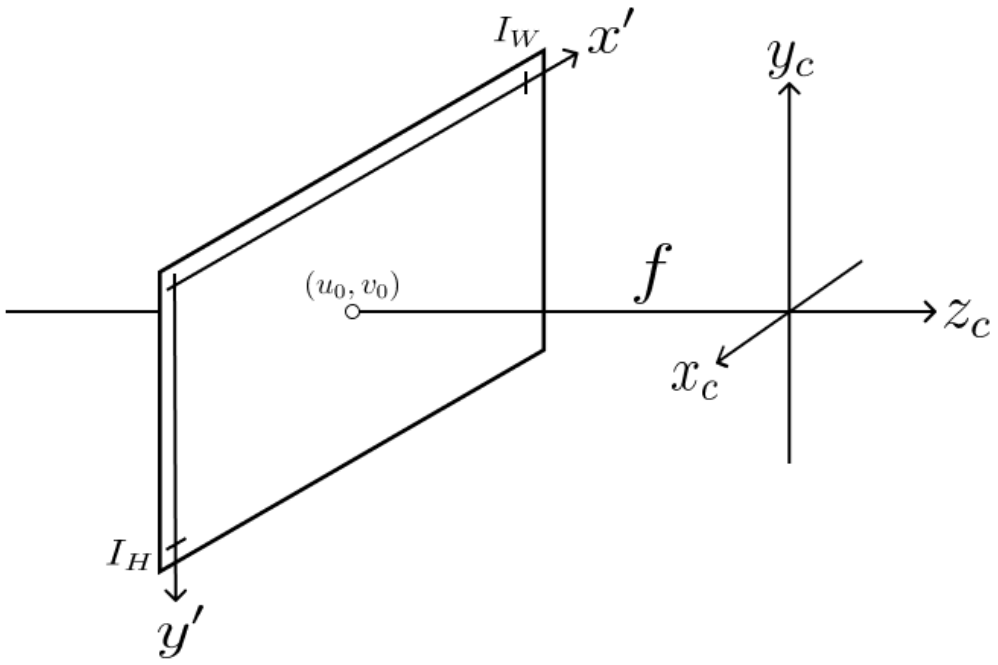


Figure 2.2: Camera intrinsics illustrating the relation between the camera image plane and the local 3-D world, showing the focal length f, the image center (u_0, v_0) and the image dimension I_W and I_H. The skew factor s is assumed to be zero.

All intrinsic parameters are usually estimated through camera calibration, a process that can be performed in different ways. An example of such a process is capturing images of a known calibration pattern, such as a chessboard, which with some computations can create a correspondence between the 2-D image and the 3-D world,

to then be used to estimate the intrinsic parameters.

After the intrinsic parameters are known, they can be used in various computer vision algorithms. For instance to compute the position and orientation of an object in 3-D space by using multiple images, known as multi-view stereo reconstruction algorithms, or to transform coordinates between different frames. To map 3-D coordinates to pixel coordinates the following equation can be used

$$\tilde{x}_s = \mathbf{P} p_w, \tag{2.4}$$

where $\tilde{x}_s$ denotes the pixel coordinates, $\mathbf{P}$ is the camera matrix and p_w is the 3-D world coordinates.

2.4.5 Focal length

The relation between f_x and f_y, i.e. the focal length for the sensor dimensions x and y, is commonly expressed as

$$f_x = \lambda f_y, \tag{2.5}$$

where λ is the aspect ratio between the sensor dimensions [6]. It is common to define f_x as f and express f_y as λf.

The relation between the focal length f_x and the horizontal field of view θ_H of the image can be expressed as

$$\tan \frac{\theta_H}{2} = \frac{W_S}{2f_x}, \tag{2.6}$$

where W_S is the physical sensor width [6]. The same equation holds for the focal length f_y and the vertical field of view θ_W by instead using the physical height of the sensor H_S.

2.4.6 OpenCV

OpenCV (Open Source Computer Vision Library)[7] is an open-source cross-platform library mainly aimed at real-time computer vision applications. It supports a wide of programming languages, such as C++, Python and Java, making it highly flexible for development. Several computer vision algorithms are supported, such as for image and video processing, feature detection, calculation of homography matrix and object recognition.

[7] OpenCV `https://opencv.org/`

2.5 Calibration of multiple PTZ cameras

An approach on how to infer the relative positioning and orientation for multiple PTZ cameras is proposed in the paper [2]. The positioning of the cameras can then be of use when performing coordinate transformation across multiple cameras mounted at different positions.

For each PTZ camera, the mounting height h is estimated by observing two papers lying on a horizontal plane, as seen in Figure 2.3. The horizontal plane on which the papers reside is defined as the xz-plane, denoted as π_0 and sometimes called the pan-plane, which is parallel to the $x_c z_c$-plane of the cameras. The height h is thus defined as the vertical length between the horizontal plane of the camera and the π_0-plane, and is estimated by relating the line segments in the camera's local 3-D world coordinate system to their projection on the image plane. These estimations are built on the knowledge that the lengths of the paper sides are known in advance and that the planes are as parallel as possible, as a slight offset in rotation may yield inaccurate estimations. The position of the camera when the target is at the center of the camera image, i.e. the pan angle ϕ and tilt angle θ, are saved for later use. With the heights of the cameras estimated, the relative angle ω between two local world coordinate systems can then be calculated by comparing the 3-D space projections of a common vector, as seen in Figure 2.4. Meaning that by identifying a shared 2-D vector in the respective images and then projecting them to 3-D space, the angle between them will be ω. The sign of ω indicates the rotational direction between the coordinate system.

Using the camera positions and the relative angle, coordinate transformations can then be performed between the coordinate systems for the two cameras. The method can also be expanded for a system of multiple cameras, by simply defining one camera as a reference and then calibrating all other cameras towards the reference. This approach can be considered both efficient and feasible as it relies on a few computations and only requires two papers of known size to be visible from the view of the cameras.

In the equations for this section, the parameters f_x, f_y, and (u_0, v_0) are the coefficients found in the camera matrix as described in previous sections, with $(u_0, v_0) = (\frac{I_W}{2}, \frac{I_H}{2})$. The skew factor s from the camera matrix is assumed to be zero for simplification. The parameter r is the distance between the rotational tilt axis of the camera and the image projection on the physical sensor. It is often assumed to be zero for simplification, as the value is in the range of millimeters and will thus not have big implications for mounting heights in the range of meters.

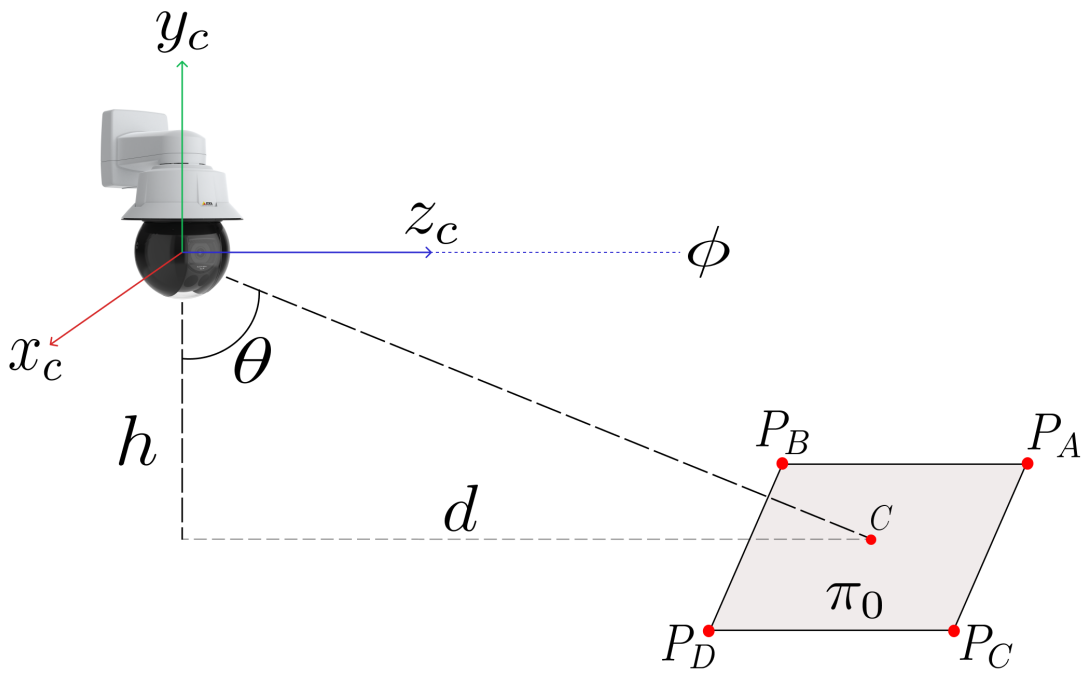


Figure 2.3: Model of the camera setup for the calibration procedure, viewed from the side. The camera, with a world coordinate system $[x_c, y_c, z_c]^T$, can be seen looking at a calibration target with a center C and corners marked as P. The target is on the π_0-plane that is parallel with the $x_c z_c$-plane. The camera is positioned at a pan angle ϕ and a tilt angle θ when having C in the center of the camera image. The camera is mounted at a vertical distance h, called the camera height, and a horizontal distance d from the target, both distances measured from the π_0-plane.

2.5.1 Coordinate mapping on a tilted camera

When the PTZ camera tilts with an angle θ, the 3-D point $[x_c, y_c, z_c]^T$ in its local world coordinate system will be projected on the point $[x', y']^T$ in the image coordinate system [2]. With a given image coordinate, the camera tilt angle θ and the height h, it is possible to back project the coordinate onto a horizontal plane with $y_c = -h$ in the local world coordinate system according to the following equation

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} \frac{(x'-u_0)f_y(r\sin\theta - h)}{f_x[(v_0-y')\cos\theta - f_y\sin\theta]} \\ -h \\ \frac{[(v_0-y')\sin\theta + f_y\cos\theta)(r\sin\theta - h]}{(v_0-y')\cos\theta - f_y\sin\theta} - r + r\cos(\theta) \end{bmatrix} \tag{2.7}$$

This equation is derived from the camera matrix, on the assumption that the camera extrinsic matrix equals [**I 0**] for a rectified world coordinate system. The derivation of equation (2.7) can be found in [3].

2.5.2 Estimation of camera position

Estimation of the camera position is done by relating the real-world length L of a line segment with its image projection l. Presuming that L is placed onto the horizontal π_0-plane.

By defining a line segment in the image as a start point (x'_A, y'_A) with the endpoint (x'_B, y'_B) the relation between L and l can be expressed as a function of h on the form

$$\begin{aligned} L = l(h) = \Bigg\{ \Bigg(& \frac{(x'_B - u_0) f_y (r \sin\theta - h)}{f_x[(v_0 - y'_B)\cos\theta - f_y \sin\theta]} \\ & - \frac{(x'_A - u_0) f_y (r \sin\theta - h)}{f_x[(v_0 - y'_A)\cos\theta - f_y \sin\theta]} \Bigg)^2 \\ & + \Bigg(\frac{[(v_0 - y'_B)\sin\theta + f_y \cos\theta)(r \sin\theta - h]}{(v_0 - y'_B)\cos\theta - f_y \sin\theta} \\ & - \frac{[(v_0 - y'_A)\sin\theta + f_y \cos\theta)(r \sin\theta - h]}{(v_0 - y'_A)\cos\theta - f_y \sin\theta} \Bigg)^2 \Bigg\}^{\frac{1}{2}} \end{aligned} \tag{2.8}$$

as described in [2]. The derivation of equation (2.8) can be found in [3].

In solving for h the Levenberg-Marquardt algorithm can be applied on the nonlinear equation (2.8), as done in [3], for data points that ranges from (x'_1, y'_1) to (x'_m, y'_m) in following manner

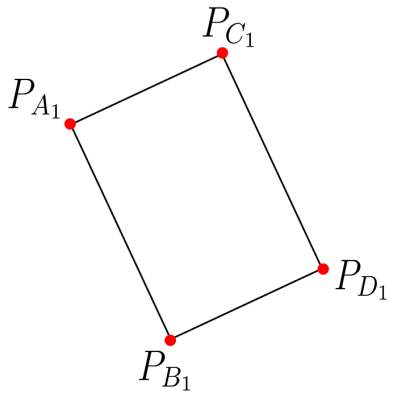
$$\begin{aligned} & F(x'_1, y'_1, x'_2, y'_2, \ldots, x'_m, y'_m, f_x, f_y, u_o, v_o, \theta, r, h) \\ & \quad = \sum_{i_1}^{m} \| l_i(x'_i, y'_i, f_x, f_y, u_o, v_o, \theta, r, h) - L_i \|^2, \end{aligned} \tag{2.9}$$

where each data point represents a corner in the image. For the estimation to be accurate and reliable, several line segments are needed in different positions on the horizontal plane, although they do not need to have the same length.

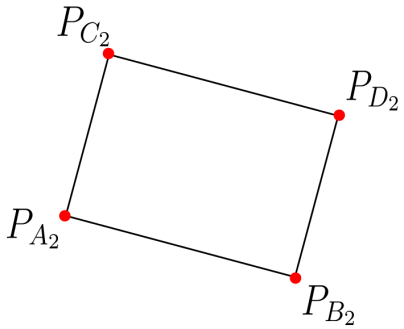
2.5.3 Calculation of relative angle

The calculation of the relative angle between two local world coordinate systems first requires finding a common vector that can be viewed from both cameras, as seen in Figure 2.4. The vector, starting on point P_A and ending on P_B, needs to be converted from its image coordinate to local world coordinates using equation (2.7) for each camera (as it is dependent on the tilt angle θ and the estimated height h). It is then possible to calculate the relative angle ω of two vectors using the inverse

(a) Target viewed from camera 1.



(b) Target viewed from camera 2.



(c) The relative angle ω, measured between the line $\overline{P_A P_{B_1}}$ (from Figure a) and $\overline{P_A P_{B_2}}$ (from Figure b). Observe that $P_A = P_{A_1} = P_{A_2}$ is chosen as the common point across the perspectives.

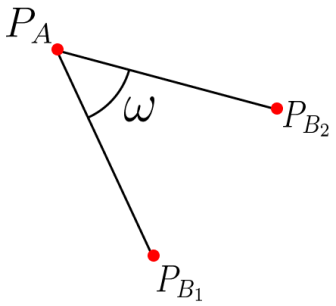


Figure 2.4: Calculation of the relative angle ω is done by relating two lines from different perspectives. All points P (and the lines thereof) are expressed as local world coordinates.

of

$$\cos(\omega) = \frac{\langle \overline{P_{A_1} P_{B_1}}, \overline{P_{A_2} P_{B_2}} \rangle}{\| P_{A_1} P_{B_1} \| \times \| P_{A_2} P_{B_2} \|}. \tag{2.10}$$

Chapter 3

Method

To achieve the goals stated for this project it is essential to develop some procedures that, when chained together, will produce the desired tasks of tracking an object across a site. This chapter contains descriptions of the necessary routines for this project; such as calibrating the cameras and performing the handover procedure, with the implementation details later outlined in the next chapter.

3.1 Calibration procedure

To produce the necessary calibration data for performing PTZ coordinate transformation it is necessary to construct a process that will output the needed data. The calibration procedure can be divided into the following steps:

1. Target detection
2. Matching procedure
3. Estimation of calibration parameters

For this process, calibration targets will be needed to calibrate against, as to find relevant features to match across perspectives. The two first steps of the calibration procedure are the base for relating two images from different perspectives, as this is later on used to map line segments across images. The estimation of the calibration parameters is built on the theory presented in section 2.5. Other approaches to estimating the height may be accomplished by using the built-in laser found in certain PTZ cameras, and may be used as a complement to evaluate the height result from the calibration procedure.

3.1.1 Target detection

Before the matching procedure can be executed it is first crucial that the calibration targets are found in the scene and zoomed in sufficiently so that the resolution of the target is of the highest quality. Target detection is an iterative process of finding the target, calculating the next camera position and then determining when to stop the search. The output of the tasks is the exact position of the target and the amount of zoom needed to cover it in the whole field of view of the camera.

The process assumes that the type of calibration targets are known in advance, to have a clearly defined target to find in the scene. For simplicity, the target is assumed to exist in the current field of view of the camera.

The process starts with performing feature detection on the current image view aiming at the target. The features are then compared to the expected calibration targets using feature matching to determine the position of the target, expressed as pan, tilt and zoom values. It is then determined if the target covers the whole field of view of the current camera view, if not then the camera may move to the new position and zoom in to get a better view of the target for the next iteration of the target detection process. If it is covered completely, the process can return the pan and tilt position of the target. The process diagram can be seen in Figure 3.1.

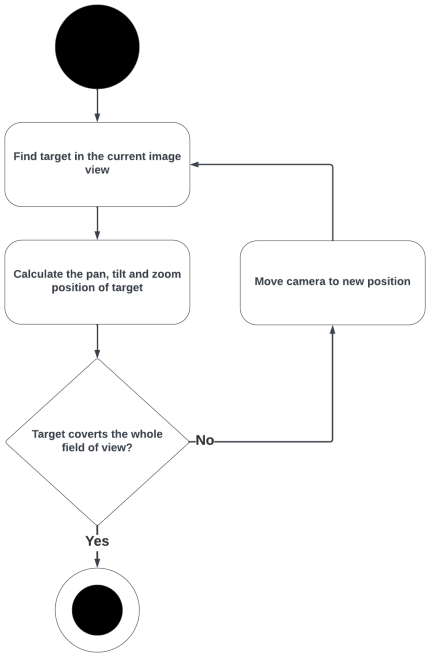


Figure 3.1: Process diagram of the target detection process. The start is represented as a black circle, and the end as a smaller black circle with a white border. For each iteration, the camera will zoom in more to cover the whole target in the current image view.

3.1.2 Matching procedure

For mapping the same object across multiple images it is required to have a homography matrix that associates the different coordinate systems, as described in section 2.4.3. After target detection has been performed on the cameras, the matching procedure may take the current images and in parallel perform feature detection, and then perform feature matching to determine if the target is the same in both images. The output of the task will be the homography matrix that associates the different coordinate systems, allowing for coordinate transformation and object mapping across different perspectives. As with target detection, it is assumed that the type of calibration targets is known in advance. The process diagram can be seen in Figure 3.2.

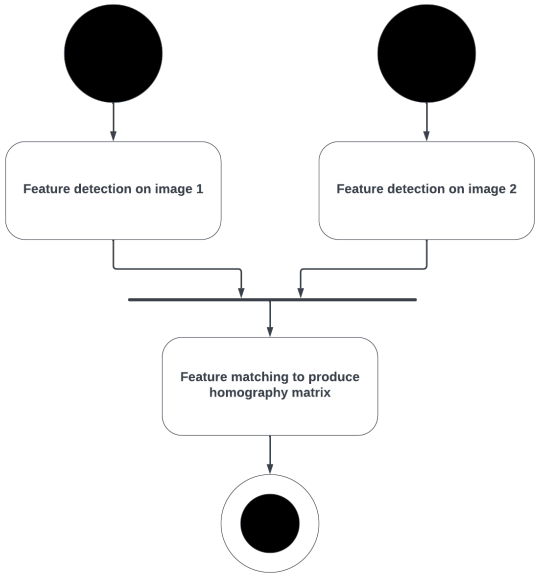


Figure 3.2: Process diagram of the matching procedure. The start is represented as a black circle, and the end as a smaller black circle with a white border. The black bar means that the process will not advance until previous states are done. The feature detection is run in parallel on two cameras, the results are then waited for so that the feature matching can be run.

3.1.3 Estimation of calibration parameters

For performing coordinate transformation it is required to obtain the necessary calibration data for the calculations. The output parameters of interest in this process are the estimation of mounting height h and the relative angle ω between two cameras. The assumption is that the line segments used for estimating parameters and relating images are on a horizontal plane that is exactly parallel to the horizontal planes of the cameras.

The estimation of calibration parameters begins with performing parallel feature

detection for line segments in the current image for each camera. Solving equation (2.9) with the line segments as input will yield an estimation of the height h for the stated camera. Applying the matching procedure on the line segments of each camera will produce the correspondence between the images. Utilizing equation (2.10) on the matched line segments will result in an estimation of the relative angle ω between two cameras' world coordinate systems. The process diagram can be seen in Figure 3.3.

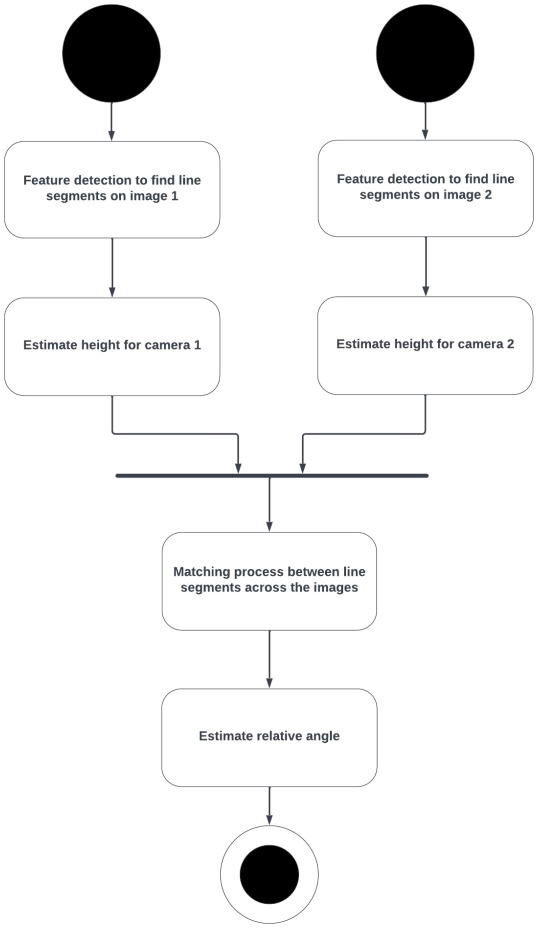


Figure 3.3: Process diagram of the estimation of camera parameters.

3.1.4 Estimation of height using the built-in laser

Alternative methods of estimating the mounting height of the camera can be performed by using the built-in laser on certain PTZ cameras. Thus two methods can be proposed:

1. Moving the PTZ camera straight down to measure the height against the ground.

2. Estimating the height by using the tilt angle to the calibration targets and

measuring the direct distance.

These methods may be used as a complement for estimating height, as this calibration parameter is used heavily in many equations. The height is easily calculated using the cosine formula, using the distance to the target and the tilt angle of the camera when aiming at the target in the center of the view. The limitation is that measuring the height by moving the camera straight down assumes that there are no objects in the way, and will measure the height from the ground and not from the π_0-plane. Meaning that the height difference between the ground and the π_0-plane must be subtracted. While for estimating the height using the tilt angle and the distance to the target assumes that the target is in the center of the image, as well as that the target is on the same height level as the π_0-plane.

3.2 PTZ coordinate transformation

The process of transforming pan and tilt angles from one camera coordinate system to another is necessary to move the camera to the desired position for the handover procedure. For the calculations below, camera 1 is denoted as the camera to perform handover from, while camera 2 is the camera to take over the tracking (i.e. the camera to transform the coordinates to). Figures 3.4 and 3.5 illustrate the top and side view of the scene from where the target is viewed from both cameras. For the pan angle calculations, it is assumed that all distances are projected onto the π_0-plane, as to preserve the relations between the distances and enable the usage of simple trigonometric calculations. This algorithm assumes the following parameters as input, with the camera index denoted as i:

- The distance d_i between the camera and the calibration targets.
- The pan angle ϕ_i of the calibration targets.
- The tilt angle θ_i of the calibration targets.
- The pan angle p_{T_1} of camera 1, aimed at the target to track.
- The tilt angle t_1 of camera 1, aimed at the target to track.
- The distance d_{T_1} measured from camera 1 to the target to track.
- The relative angle ω between the cameras.

The angles are measured in the coordinate system of the camera in that context. ω must be measured from camera 1's coordinate system, as the sign indicates the rotational direction between the cameras. A positive value means that camera 1 is

to the left of camera 2, with the calibration targets as a reference, while a negative value indicates it is to the right.

When performing PTZ transformation, there are two shared parameters between the cameras that are used to associate the coordinate systems with each other. The first is the horizontal distance d_{CT} between the calibration targets and target in the π_0-plane, and is used in the calculation for the pan angle. The second is the vertical distance h_T, also seen as the height of the target, calculated alongside the y-axis.

Pan calculation

The goal of the pan calculation is to output the pan angle p_{T_2}, which is the angle for camera 2 when it is aimed toward the target to track. The calculation can be divided into steps as follows:

The first step is to determine the angle

$$\alpha = p_{T_1} - \phi_1,$$

where the angle represents the distance, in degrees, between the calibration targets and the target. The sign of the angle indicates if the target to track is to the left or the right of the calibration targets. It is assumed that $p_{T_1} < 90°$ for these calculations.

Secondly, the distance d_{CT} between the targets

$$d_{CT} = \sqrt{d_1^2 + d_{T_1}^2 - 2d_1 d_{T_1} \cos(\alpha)},$$

which is one of the shared parameters between the cameras.

Thirdly, the angle β is computed as

$$\beta = \arccos\left(\frac{d_{T_1}^2 - d_1^2 - d_{CT}^2}{-2d_1 d_{CT}}\right).$$

For the next step the distance d_{T_2} to the target can be calculated as follows depending on the sign of α and ω

$$d_{T_2} = \begin{cases} \sqrt{d_2^2 + d_{CT}^2 - 2d_2 d_{CT} \cos(\omega + \beta)} & \text{if } (\alpha > 0 \ \& \ \omega > 0) \text{ or } (\alpha < 0 \ \& \ \omega < 0). \\ \sqrt{d_2^2 + d_{CT}^2 - 2d_2 d_{CT} \cos(\omega - \beta)} & \text{if } (\alpha > 0 \ \& \ \omega < 0) \text{ or } (\alpha < 0 \ \& \ \omega > 0). \end{cases}$$

The different cases depend on whether the target is to the left or right of the calibration targets, and if camera 1 is to the left or right of camera 2.

Then the angle γ between the calibration targets and the target is computed as

$$\gamma = \arccos\left(\frac{d_{CT}^2 - d_2^2 - d_{T_2}^2}{-2d_2 d_{T_2}}\right).$$

Ultimately, the pan angle p_{T_2} for where to move the secondary camera is

$$p_{T_2} = \begin{cases} \phi_2 + \gamma & \text{if } \alpha > 0 \\ \phi_2 - \gamma & \text{if } \alpha < 0 \end{cases}.$$

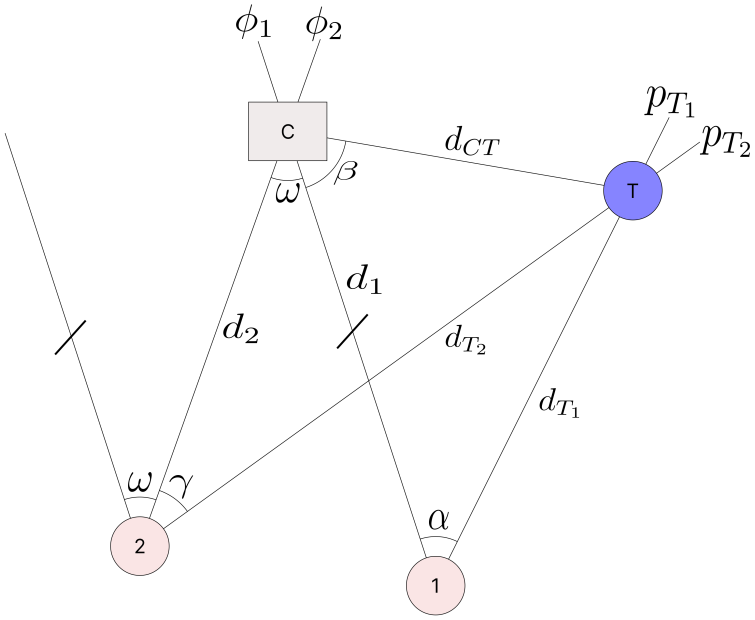


Figure 3.4: The top view of camera 1 and 2, calibration targets C and a target T. The distances, projected on the π_0-plane, are denoted as d. The pan angles to the calibration targets are denoted as ϕ_i and the angles to the target are denoted as p_{T_i}, both are indexed with concern to the corresponding camera. The bold diagonal line indicates that the lines are parallel. The parameter of interest is the distance d_{CT} between the calibration targets and the target. This specific figure illustrates the case of $\omega > 0$ and $\alpha > 0$.

Tilt calculation

Calculation of the shared parameter h_T is done from the perspective of camera 1, using the camera height h_1, the distance d_{T_1} to the target and the camera tilt angle t_1 as follows

$$h_T = h_1 - \frac{d_{T_1}}{\cos t_1}.$$

To then calculate the new angle t_2 for the secondary camera, the inverse of the following can be applied

$$\cos t_2 = \frac{h_2 - h_T}{d_{T_2}}.$$

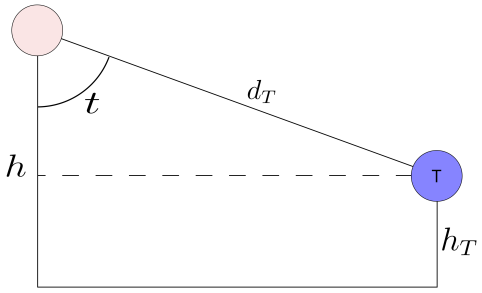


Figure 3.5: The side view of a camera tilted with the angle t, aimed at a target T. The camera height h and the target height h_T are measured from the π_0-plane. The distance d_T is measured between the camera and the target. The parameter of interest is the height h_T of the target. This specific figure illustrates the case of $h_T > 0$.

3.3 Handover state machine

The handover process is modeled as a state machine, with transitions that are only dependent on the camera events, as seen in Figure 3.6. The process starts in the idle state, where the cameras have the tracker enabled and wait for an object to enter their field of view. The cameras are positioned at their home position, where they will later return when the object disappears from the scene. On a tracking event for one of the cameras, the state changes to the tracking state.

In the tracking state, it is assumed that only one camera (expressed as the master camera) is tracking an object, and the secondary camera has the tracker disabled. The position of the master camera is continuously fetched and used for calculating if the object is in the area of handover. The area of handover may be defined differently depending on the use case and is later determined as an implementation detail. When the object enters the area, the tracker for the secondary camera is enabled and the camera is moved to the position of the object (calculated using PTZ coordinate transformation as described in section 3.2). On a tracking event for the secondary camera, the state changes to the handover state. If the tracker stops tracking, e.g. when the object disappears, the state will revert to the idle state.

For the final state, the handover state, both cameras are tracking the object and the handover process can thus be finalized. The tracking for the previous master camera will be disabled and the camera moved to its home position. On a tracking disabled event for the previous master camera, the state changes back to the tracking state and the secondary camera becomes the new master camera. As for the tracking state, if the tracker stops then the state will revert to the idle state.

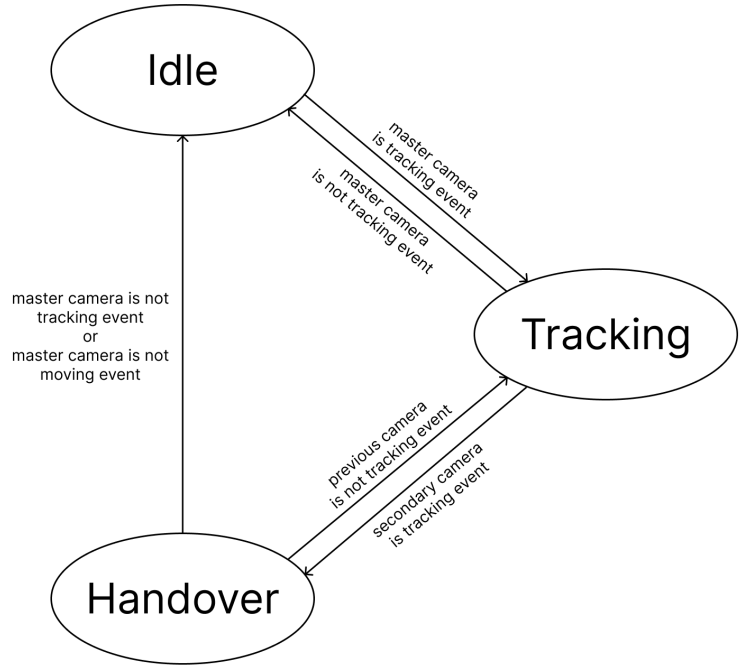


Figure 3.6: Simplified representation of the state machine used for the handover process. The states are represented as ovals and the transitions are the arrows in between. In the transition from the tracking state to the handover state, the secondary camera becomes the new master camera.

Chapter 4

Implementation

This chapter aims to concretize the proposed methods found in Chapter 3, by filling in the implementation details and taking into account the hardware and software that is to be used.

4.1 Camera APIs and events

For developing applications that perform the desired goals described in section 1.2, it is first needed to evaluate which camera APIs and events to use.

For retrieving the current PTZ position, the 'PTZ CGI' can be called which will return the current pan, tilt, zoom and FOV values. Through the same interface, it is also possible to move the camera by either specifying the absolute pan tilt position or by entering pan and tilt speeds. Although not yet implemented in common Axis PTZ cameras, it is possible to modify the firmware to enable it to send the current distance to an object, measured using the built-in laser. Through the 'Autotracking CGI' the tracker can be disabled and enabled.

The transition in the handover state machine, as outlined in section 3.3, are only dependent on the camera events. For the simple state machine (Figure 3.6) used in this project, only the "is tracking" event and "is moving" events will be used. The "is tracking" event is sent from the camera every time the tracker identifies an object to track, and contains a boolean indicating if it is tracking or not. The event will be resent with a value of false when the object disappears from the camera view. The "is moving" event is sent from the camera as soon as one of the motors starts to move, i.e. either pan or tilt changes. This event will also contain a boolean indicating whether a movement is being performed (a value of true) or not (a value of false).

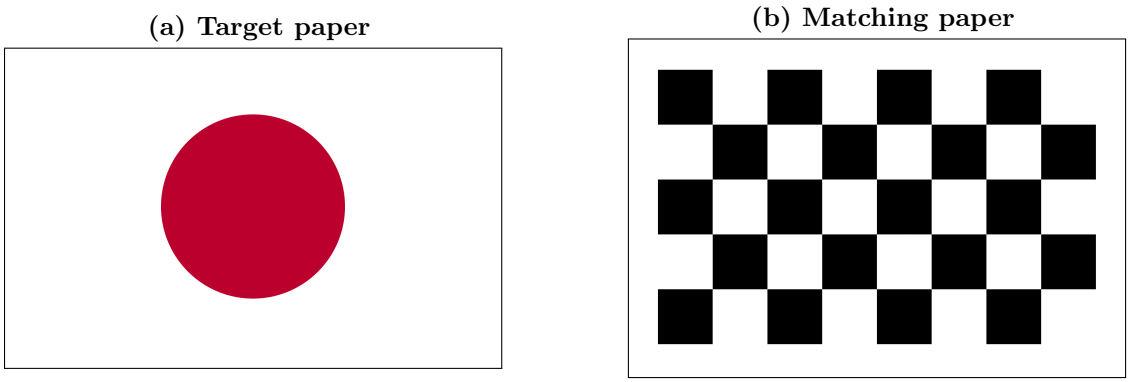


Figure 4.1: Papers used as calibration targets for the calibration procedure.

4.2 Camera calibration application

The calibration procedure is implemented as an on-edge C++ camera application, as it requires quick communication with the PTZ camera movement capabilities and certain programming libraries such as OpenCV and the internal PTZ library. It will be controllable through a CGI, which will enable the server side to both start the calibration procedure and retrieve the calibration data.

As calibration targets two A3-sized papers will be used, referred to as target paper and matching paper as seen in Figure 4.1. The first paper is a white sheet of paper with a big red circle in the middle, which will be used as the target paper and permits the PTZ camera to find the targets in a large scene, as both the red color and the circular shape can be seen as unique in the scene. The other is a paper with a black and white chessboard pattern, that will be used in the matching procedure to effectively and accurately match features across two images. Both papers are placed on a flat surface that is parallel to the horizontal plane of both cameras and should be visible from each view. The A3-sized type of paper is used as the lengths of the sides are standardized to 297×420 millimeters.

The algorithm flow of the calibration application, generally described in section 3.1.3 and Figure 3.3, can be separated into two steps:

Step 1: Start calibration

The primary function of the first step in the calibration procedure is to estimate the calibration parameters used to associate different perspectives. This is done by performing feature detection that produces the necessary features that can be sent to another camera. The benefit of separating the calibration into two steps is that this step can be run in parallel across all cameras. As input initial pan and tilt values are sent as method payload in order to roughly aim the camera at the

calibration targets. This is done to speed up the process, as opposed to searching the whole pan-tilt sphere of the camera.

After adjusting the camera position, target detection is performed as outlined in section 4.2.1 and will output the target angles ϕ and θ. The camera will be correctly oriented towards the targets, with an appropriate zoom level. The feature detection process, described in section 4.2.2, can then be performed and output the paper corners and the chessboard pattern. Finally, estimating the internal calibration parameters as described in section 4.2.4, can be performed by utilizing the newly found paper corners and will output an estimation of the height h.

The paper corners are transformed to local 3-D world coordinates using equation (2.7), which depends on the estimation of h and the θ angle. This step is essential as the later stages of the estimation of the external calibration rely on the corners being expressed in the correct coordinate system. The output of this step is the estimated height h, the PTZ position ϕ and θ, the paper corners expressed as 3-D world coordinates and the chessboard pattern from the perspective of this camera.

Step 2: Relate perspectives

For the second step, the main task is to relate the perspectives of two cameras and calculate the relative angle ω. This step can also be run in parallel on all cameras, yet it cannot be started until the first step has been performed on the other camera as it depends on its data. As input the calibration data is thus sent from another camera, through the server, as the method payload. The matching procedure can then be performed, as described in section 4.2.3, by using the calibration data from this camera and the other camera. This process will produce the corner correspondence between the perspectives. Estimating the external calibration parameters can then be done, described in section 4.2.4, and will output the relative angle ω.

4.2.1 Target detection

Target detection, as outlined in section 3.1.1, is the process of finding the calibration targets in order to increase the resolution of the target. The main target is selected to be the paper in Figure 4.1a, which with the form and color of a red circle offers unique and easy characteristics to find in a scene.

The first step of the process is to move the camera to the initial pan and tilt values that are given as input, as they roughly aim the camera toward the target paper so that the red circle is shown in the image (as seen in Figure 4.2a).

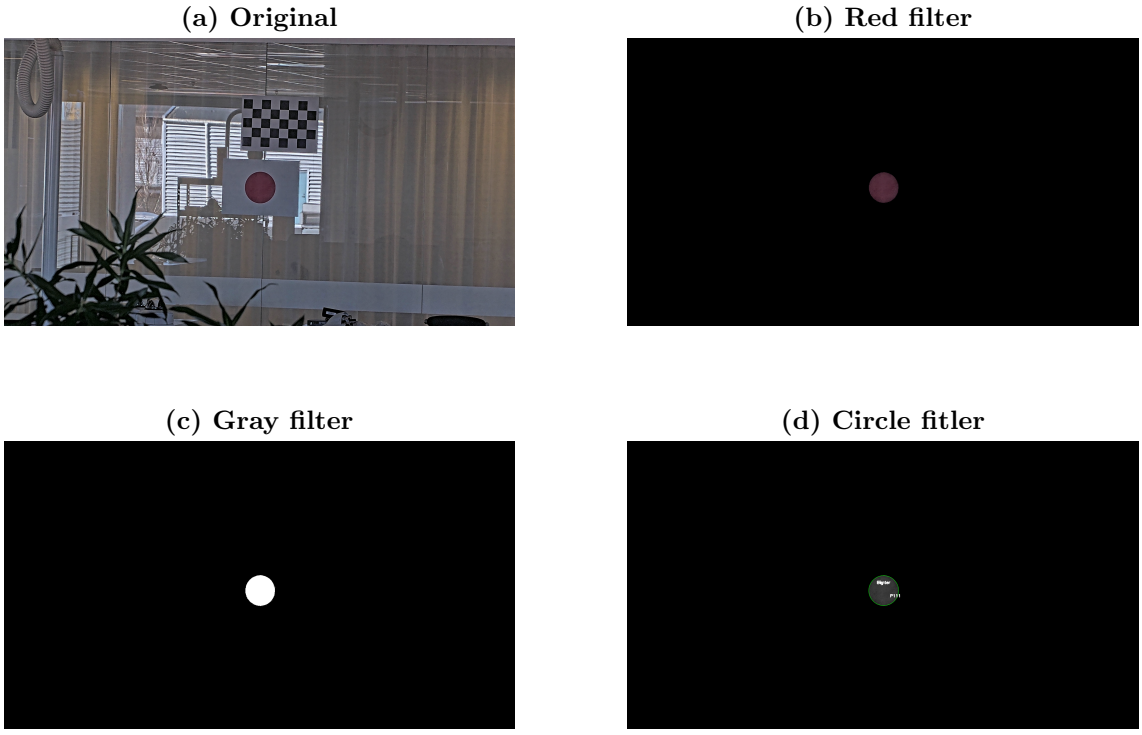


Figure 4.2: Target detection pipeline.

The process of finding the red circle is done by first filtering the image using a red filter, i.e. removing all other colors and highlighting the red (Figure 4.2b). Then the red colors can be replaced by a grayscale to facilitate contour finding and thus increase the probability of finding the circle (Figure 4.2c). The finding of the circle is accomplished by using blob detection, which searches for all circle-like objects by measuring the area, circularity and convexity of the found contours.

The blob detection will output the size and center of the circle, as seen in Figure 4.2d. Then the image coordinate of the circle center can be translated to pan and tilt coordinates by using the internal PTZ library of the camera, which are then saved as ϕ and θ. To determine whether to retry the target detection step or to continue depends on the size of the calibration targets and the current field of view. If it is guaranteed that both papers cover the whole FOV then the algorithm can proceed with the next step, otherwise target detection is retried by increasing the zoom and moving the camera to the newly calculated pan and tilt position.

4.2.2 Feature detection

For the feature detection pipeline, it is assumed that the current view is zoomed in enough so that both papers are visible and covered in the whole field of view of the camera. In this process, both the target paper and matching paper are

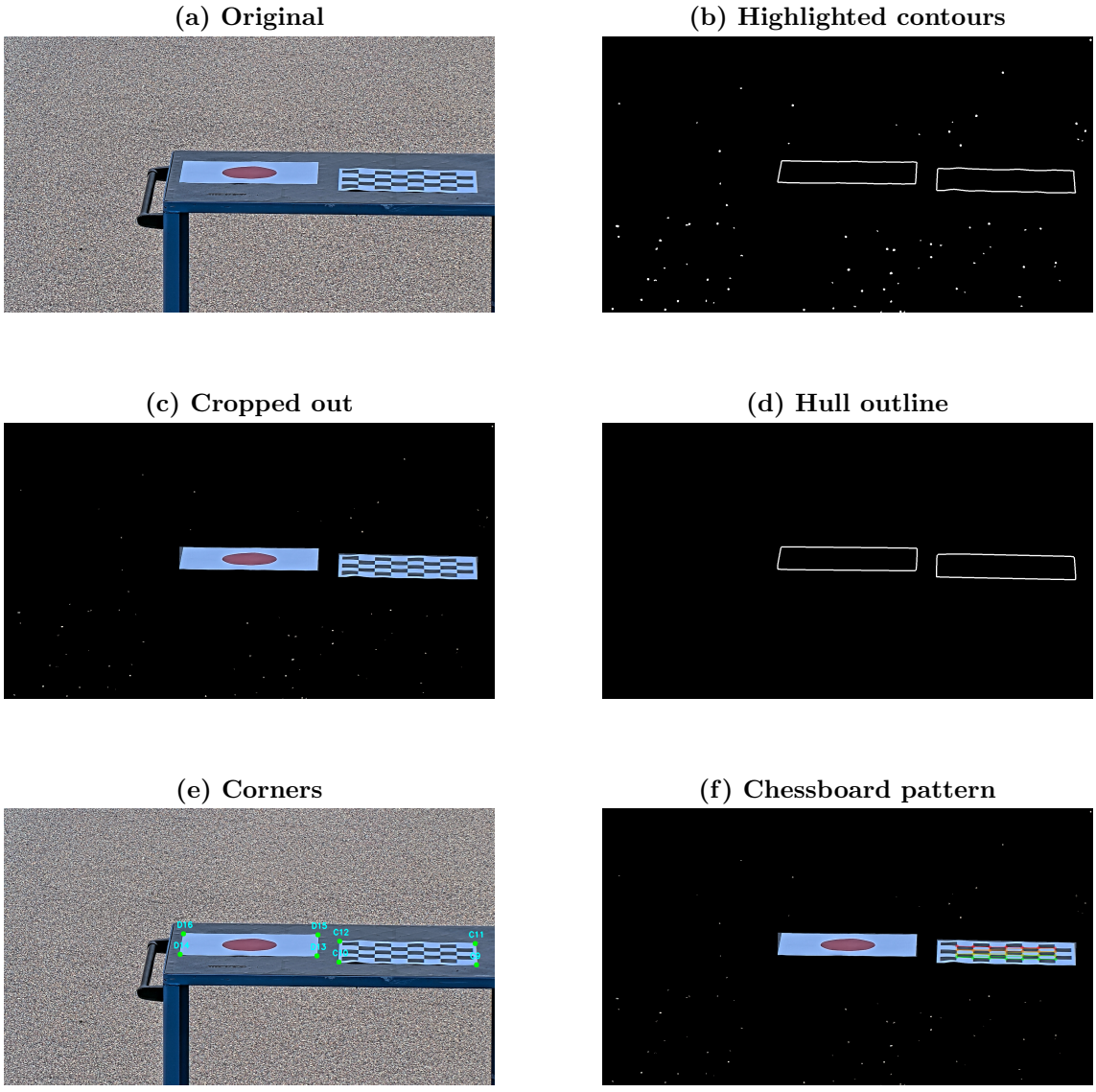


Figure 4.3: Feature detection pipeline.

used to find corner features, later used for the matching procedure. The matching paper (Figure 4.1b) contains a chessboard pattern that is also used in the matching procedure, but the features are first to be extracted in this process.

The first step is to save the current image (Figure 4.3a) as RGB JPEG and to use a gray filter, again to facilitate the contour finding algorithm. To find the contours of the papers Canny edge detection is used, as it is especially good at extracting sharp edges; in this case, the distinct boundings of the papers as seen in Figure 4.3b.

The contours are used to approximate a quadrilateral shape with four corners around each of the papers. This is to obtain a bounding shape of the papers that can then be used to crop them out of the original image and essentially remove all surrounding

noise in the image (Figure 4.3c). For finding better contours of the papers, feature detection is applied on the image consisting of only the papers, meaning once again to convert to grayscale and perform contour finding. The contours are now used to form the hull outline of each paper, which is the minimum bounding shape of the papers (Figure 4.3d).

With the hull outlines it is possible to apply a combination of the Harris corner detector and SIFT algorithm for finding the corners, in total eight corners expressed as image coordinates. From these corners, it is then elementary to form the line segments of the papers.

On the image consisting of only the calibration papers, feature detection is performed on the matching paper to find the chessboard pattern as seen in Figure 4.3f. These features are expressed as the internal corners of the chessboard pattern, here defined to be where two black squares border two white squares, which for Figure 4.1b is a total of $7 \times 4 = 28$ corners.

4.2.3 Matching procedure

The main intention of the matching procedure is to produce the homography matrix to associate two perspectives, as explained in section 3.1.2. This is accomplished by associating the chessboard feature pattern on the matching papers across images. For this, the camera performing the matching procedure needs input in the form of the chessboard features and paper corners from another camera's perspective.

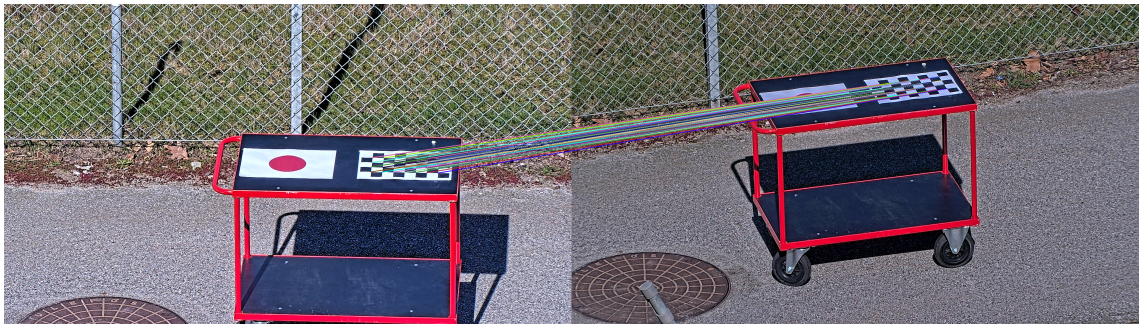
Using the chessboard features from the current camera view and associating them with the other camera's chessboard features allows for the computation of the homography matrix using built-in functions from OpenCV. This matrix performs projection from one image coordinate system to the other, and vice versa. The application of the homography matrix on the chessboard features can be viewed in Figure 4.4a.

Applying the homography matrix on the paper corners found in the feature detection process will create a bijection between the corners of one image to the other. Having the correspondence between corners also implies the correspondence between line segments, which is necessary for the estimation of calibration parameters. The paper corners and their projected counterpart are illustrated in Figure 4.4b.

4.2.4 Estimation of calibration parameters

The estimation of calibration parameters can be divided into two parts: internal and external estimation of calibration parameters.

(a) Homography matching



(b) Corner projection

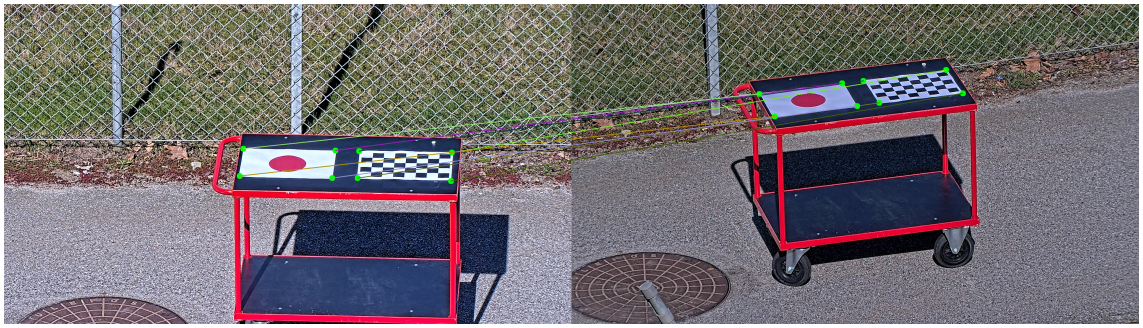


Figure 4.4: Matching procedure pipeline. The corners of the papers are represented as green circles. The corners on the left are projected to the right perspective using the homography matrix.

Internal calibration parameters

Following the description in section 2.5.2, to estimate the camera mounting height equation (2.8) and equation (2.9) is used, requiring only the paper corner coordinates as input and can thus be calculated directly after the feature detection pipeline. Solving equation (2.9) can be done by using the library ALGLIB[1], which is widely used for numerical analysis. The height can also be estimated using the laser distance to the target and the camera tilt angle, using the cosine formula.

External calibration parameters

Estimation of external calibration parameters, i.e. the relative angle ω between two cameras, is described in section 3.1.3. As the paper corners (expressed local 3-D world coordinates) are interchanged between the cameras, and then paired using the homography matrix (as seen in Figure 4.4b), it is then possible to use equation (2.10) on the line segments to calculate the relative angle between the world coordinate systems. It is of vital importance that the corresponding corners are expressed in

[1]ALGLIB `https://www.alglib.net/`

the local 3-D world coordinate of the other camera, as described in section 2.5.3.

4.3 Axis Camera Station application

The handover process is implemented as an Axis Camera Station (ACS) application. Written in C#, the application is run alongside ACS in a Windows environment. The application can either communicate directly to the ACS server, e.g. for fetching cameras in the system, or directly to the camera through a network proxy. The framework for the application provides important functionality such as asynchronously getting camera events and calling different CGI interfaces. The state machine of the application can be seen in Figure 4.5.

The process starts by fetching the cameras that are configured in the ACS system, to determine which cameras have the *Autotracker* application installed. Then a check is conducted to know if the calibration procedure has been performed, which is done by checking if a file exists with the calibration data. Either the calibration data is read, or the calibration procedure is performed by using the camera calibration application as outlined in section 4.2.

Before starting the handover state machine, an event listener process starts and will act as a transition handler for the handover state machine. Meaning that on either "is tracking" events or "is moving" events, the state machine may change state in a manner described in section 3.3.

When moving the secondary camera, the PTZ coordinate transformation is calculated as described in section 3.2. Meaning that with only the positioning of the currently tracking camera and the calibration data of both devices, the secondary camera can be moved to a position that guarantees to have the object in the field of view, which the tracker can then detect and start to track. This coordinate transformation is also complemented with a form of estimation of the object's future position, which is calculated by fetching the object's current position and velocity vector and calculating the angle difference between the current time frame and the next. This relatively small angle distance will provide the track with more time to detect the object, as the camera will be positioned in such a manner that the object will exist longer in its field of view.

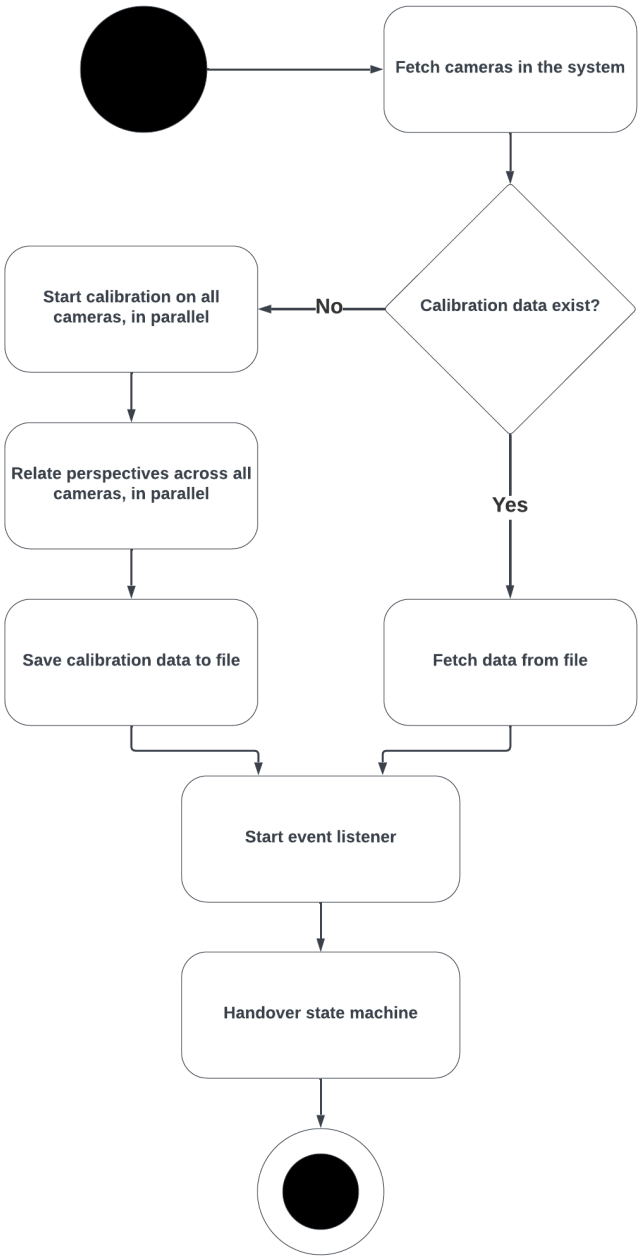


Figure 4.5: The state machine for the ACS application. The start is represented as a black circle, and the end as a smaller black circle with a white border.

Chapter 5

Results and discussion

The results of various measurements and tests are presented in this chapter, alongside a discussion on the implementation choice and limitations taken in this project. In association with this, advantages and disadvantages will be discussed for the selected approaches with short descriptions of possible future extensions.

5.1 Calibration procedure

As a whole, the calibration procedure performed its task with rather satisfying results under ideal circumstances. The process proves to be moderately quick in execution time and is uncomplicated to implement because of the wide support of libraries regarding computer vision and PTZ cameras. Both the height h and the relative angle ω could be estimated with fairly accurate values, indicating that the implementation of the method proposed in section 3.1 is suitable in the context of this specific application. A few measurement tests were performed with the setup seen in Figure 5.1, where a camera was temporarily mounted on a camera stand in front of a table with the calibration targets on top.

5.1.1 Target detection

Figure 5.2 demonstrates a couple of iterations of the target detection procedure, where the camera zooms in for each iteration and improves the pan and tilt angles to center the camera view on both papers. The process worked in the majority of the cases and served the next procedure, feature detection, with high-quality images of the calibration targets. In most scenes, the uniqueness of the red color and the matching against a circular shape provided the algorithm with enough information to quickly and efficiently search for the target paper.

Figure 5.1: Temporary setup for testing the height estimation from the calibration procedure. The two calibration papers are lying on top of the blue table with a height of 1.01m from the ground. The height of the camera stand could be changed between measurement tests.

(a) Iteration 1

(b) Iteration 2

(c) Iteration 3

Figure 5.2: Iterations of the target detection process, where new pan and tilt angles are calculated for every iteration and the camera zooms in until the papers cover the whole field of view.

However, there were a few instances where the target detection procedure failed because of the choice of the target paper. Its unique color in outdoor scenes helps differentiate the target from its surrounding, however, a problem was observed if there are other red and circular objects in the vicinity of the target paper. Figure 5.3 illustrates the problem when the target paper is too close to red objects, resulting in an erroneous adjustment of the camera position. As this issue was observed fairly early, countermeasures were implemented in the form of limiting the size of the target to search for as well as keeping the new pan and tilt angles moderately close to the initial guess supplied to the process.

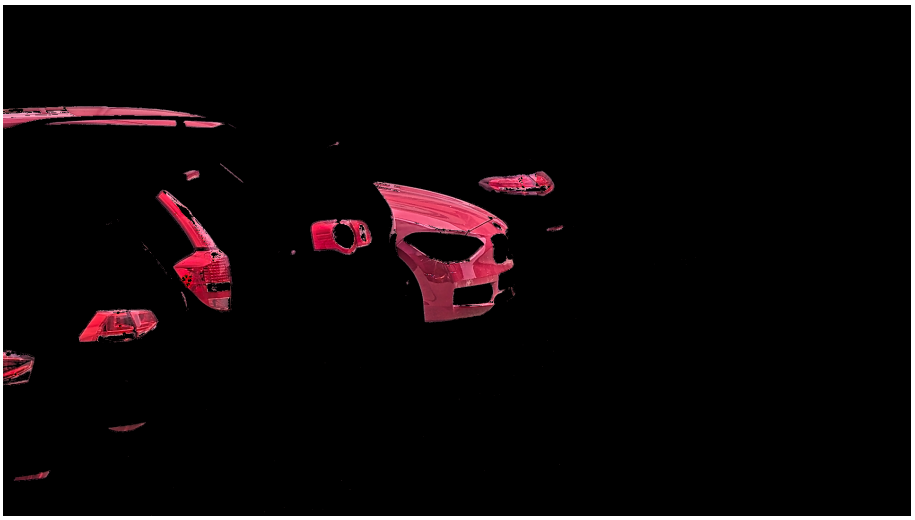


Figure 5.3: Issues with the color red, as red cars and the taillights could confuse the procedure into moving to the wrong position.

When scaling the distances even further, it proved to be hard to detect the red circle from a zoom level of 1. At long distances the target becomes too small to be seen and even warped into an oval shape, causing the algorithm to not detect it as circular. This could temporarily be solved by supplying an initial zoom value alongside the pan and tilt angles, thus enabling the camera in seeing the target. The configuration of the detection for circular shapes could be reconfigured to permit even oval shapes.

A possible solution for the issues observed at this stage would be to use a more complex target paper to further distinguish the paper from its surrounding, making the process more reliable and lower the rates of failure. This could be realized by dynamically configuring the visual characteristics of the target paper, e.g. by inputting a picture of the target to the application beforehand, allowing for the algorithm to flexibly search for distinct targets. Alternatively, an approach on how to perform dynamic calibration between multiple PTZ cameras is proposed in the paper [1]. It utilizes the mapping between a horizontal plane in 3-D space and the 2-D image plane on a panned and tilted camera to infer changes in pan and tilt

angles for each camera, by using displacement of features points. This approach still uses a pre-calibration stage, but without static calibration targets, and can adapt to changes over time making for easier calibration although a more difficult task to implement.

Another limitation of this process is to manually input guess angles for the pan and tilt, to roughly aim the camera at the correct position. This could easily be automated to instead search the whole coordinate sphere of the camera, e.g. by diving the sphere into blocks based on the camera's FOV. For each block, the camera would then perform the target detection procedure and if the target is not found next block is searched. For the issue when the target is too far away, these blocks can be further divided depending on different zoom levels. The disadvantage of this approach is the amount of time it takes for the camera to move around and check for the target. Instead, for the sake of convenience, supplying initial pan and tilt angles works without introducing any issues.

5.1.2 Feature detection

The feature detection process worked as intended, with results viewed in Figure 4.2. For the majority of the tests, expected pictures were produced with paper corners correctly marked and with correctly placed chessboard patterns.

Although a few issues were detected during testing and could prove to be a problem at certain scenes. It was observed that if the paper is not lying exactly flat on the surface, the small bumps that appear alongside the paper could be erroneously detected as a corner by the algorithm. This would further create a problem as the algorithm is designed to exclusively search for eight corners, meaning that a corner would be missed on a faulty detection. Figure 5.4 illustrates the concept of the problem, where the corners are not correctly placed, however, for this example the paper got accidentally cut out of the image because the camera was too zoomed in. The solution to this issue lies in improving the contour detection, to further improve the creation of a hull outline and enhance the approximate quadrilateral shape to reduce the number of faulty detections by estimating a perfect rectangular shape around the paper.

The algorithm assumes that the papers are the only two coherent rectangles in the camera view, as it selects the two biggest rectangles identified in the scene. If there are other rectangles, it may cause problems when masking out the papers in later stages. However, this issue has not been directly observed, but may theoretically occur in certain scenes. Multiple solutions or limitations can be implemented to countermeasure this, such as checking that the size of the rectangles is within an

expected size.

In similarity to the target detection process, an improvement would be to dynamically configure the appearance of the matching paper, by describing its shape and characteristics. This could improve the probability of successful calibrations and contribute to making the system more flexible for different scenes, as the feature detection would be more robust in detecting the target.

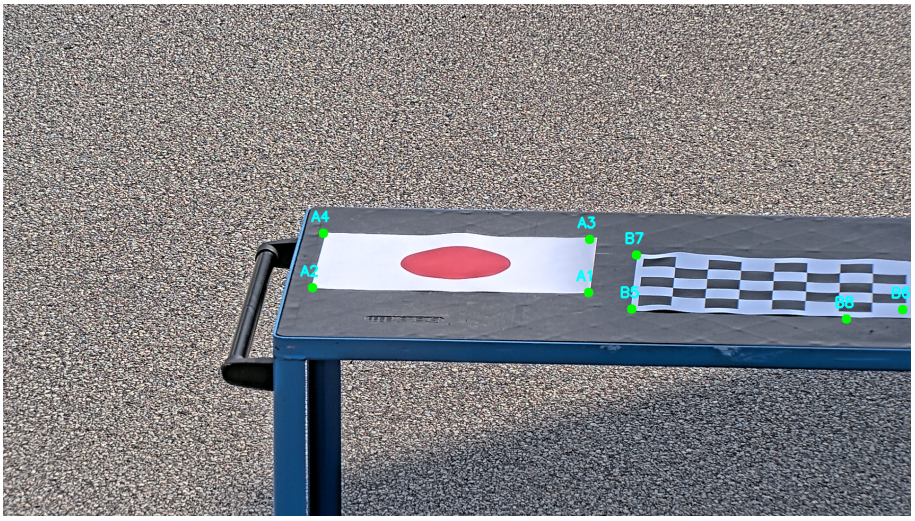


Figure 5.4: Example of failure in finding the corners of the papers because of too much zoom. Similar cases can occur if the paper is not lying flat on the surface.

5.1.3 Matching procedure

As seen in Figure 4.4 the matching procedure worked well and produced comprehensible pictures that associates two perspectives to each other. The first figure demonstrates the result of the homography matrix, where the chessboard pattern is associated across the images. Albeit slightly hard to observe, it is evident from the output that the patterns are correctly matched. This is apparent in the second figure, which illustrates the corners being projected from the left image to the right. Examining the placement of the corners shows that they are placed with good precision.

The conclusion to draw from the result is that the homography matrix worked satisfactorily in transforming pixel coordinates from one frame to another. The precision is increased by using zoomed-in images of the calibration targets, increasing the resolution of the chessboard pattern. By using reliable and well-tested functions from the OpenCV library no direct issues were observed, as the homography matrix always seemed to be calculated correctly and thus the corners were always projected to expected locations. However, it was observed that if the pattern is not centered

or zoomed-in enough, the projection of the homography matrix could be warped because of image distortion.

5.2 Estimation of calibration parameters

The estimation of the calibration parameters proved to be accurate enough for the scope of the project, although a lot of improvements could be made to further increase the accuracy. It was observed that using zoomed-in images of the calibration targets boosted the accuracy of the calibration parameters, presumably because the algorithm had more accurate lengths to estimate with. This boost proved necessary in achieving the desired precision for performing PTZ coordinate transformation. Despite this, the results may not be acceptable for other types of applications as the estimations varied by a couple of degrees or decimeters. Seeing as the tracker will correct the camera position based on the position of the object, it is not of high priority to achieve high precision.

The main issue of this approach is the assumption of having the horizontal 2-D planes parallel to each other. It proved hard during the measurement tests to mount the cameras to be as parallel to the table as possible, as the cameras are circular and have no feature to indicate their rotation in the real world. Additionally, the distortion from the camera lenses may contribute to loss of accuracy but is considered to be low and can thus be ignored. Using targets with good 3-D representation could potentially instead yield better and more accurate results if modeled correctly. However, the mathematical model required for 3-D representation would be advanced and entail a more complicated method than proposed in this project. As a result of this, a quicker approach was chosen when using 2-D space.

The estimated height could be used in many other cases, as it is almost always a requirement when dealing with the surrounding world of the PTZ camera. This included cases like calculating the depth of a scene at a certain position or possibly handling perspective distortion. While the relative angle is used in the context of relating multiple PTZ cameras at a site. This parameter is more case-specific and used for knowing where to aim the camera to achieve the best possible surveillance.

5.2.1 Estimation of camera height

Table 5.1 shows measurements tests performed with a PTZ camera mounted at different heights and a table with calibration targets at certain distances from the camera (setup can be seen in Figure 5.1). For different heights and distances, the real height was measured alongside the various form of estimating heights as described

in section 3.1.4 and section 4.2.4, i.e. estimating using the projected paper lengths and with the laser measurement device on the camera. Observe that the values in the column "Laser to target" are height values estimated by using the laser value to the target and the tilt angle of the camera using the cosine formula. Its worth noting that the heights in these tests are not against the ground, but instead the vertical distance to the π_0-plane defined in previous chapters, i.e. the surface of the blue table. Thus for some measurements the height of the blue table needed to be subtracted.

The estimated heights using the projected paper lengths proved to be moderately correct, with a maximum error of around ± 0.30m. In comparison with the result of the research in [2], which had a maximum error of ± 0.17m, the error is within an acceptable range. In the paper [2] some results demonstrated the error offset when various parameters were changed, which indicates that the method is easily influenced by various error sources. The distances used for the tests in Table 5.1 are also greater in comparison to the ones performed in the research paper, which further could prove the difference in error. In the context of PTZ coordinate transformation and tracking, this error can be viewed as acceptable, seeing as the tracker will find an object and follow it, thus correcting the position of the camera. However, for other applications that require more precise height estimation, the precision may not be good enough.

An outlier in the table is the boldly marked height on the first row. This row is an example of when the corner detection failed and a corner was misplaced, leading to a bigger error in the height estimation, as seen in Figure 5.4. Although not completely off, its error still contributes more than if the corner would have been correctly placed, as in the second row of the table.

The choice of defining a common plane such as the π_0-plane is needed as it is otherwise hard to know what the reference plane is. This is because the ground may vary at different positions, making it harder to form a common plane to calculate from. This introduces the possibility of miscalculations when the planes are assumed to be horizontal to each other. Assuring this is rather hard, as the cameras do not have any built-in gyroscopes.

The advantage of the height estimation using projected paper lengths is that it does not require special hardware, however, the assumption of parallel horizontal planes and the difficulty of finding correct corners proves to be disadvantages. While using the laser straight down may seem like the easiest method, it does not account for a common plane but measures directly from the ground, meaning that if there are objects underneath or if the ground changes elevation the reading may become

inaccurate. The last method, measuring the distance to the target, proved to be the most accurate. Both this reading and the tilt angle can be considered reliable values, making the estimated height more reliable, but with the downside of requiring special hardware.

Table 5.1: Table for heights. The left column represents the position of the target, i.e. the horizontal distance in meters and the tilt angle of the camera. All the values to the right are height values and are measured in meters. The real height is measured using a handheld laser distance meter, by placing the blue table directly under the camera. Observe that the values in the "Laser down" column have had a value of 1.01m subtracted from them, because of the height of the blue table. The boldly marked number is an example of a failed calibration.

Target distance	Tilt	Real	Estimated	Laser down	Laser to target
10.2	−16.0°	2.78	**3.59**	2.61	2.79
10.2	−16.0°	2.78	3.03	2.62	2.84
15.7	−11.9°	2.78	2.52	2.59	2.89
16.7	−10.5°	2.78	3.04	2.73	2.75
22.1	−22.0°	9.14	8.96	9.02	9.05
24.3	−24.8°	9.14	9.46	9.05	9.12

5.2.2 Estimation of relative angle

A table for the estimation of relative angles can be seen in Table 5.2. The values seem to be rather accurate, with a maximum error of ±3.2° for the wider angles. The approximation of the self-estimated relative angles is done by measuring the ϕ_1 angle of camera 1 at its position, and then mounting camera 1 at the position of camera 2. By positioning the camera towards the same point in the real world, the pan angle difference between the current position and ϕ_1 will give the angle ω. This can be deduced in Figure 3.4 by replacing camera 2 with camera 1. The reasoning behind this approach is that the pan-coordinate system for each camera may have the 0° in different directions.

Based on the results it seems plausible to conclude that the angles are less accurate the wider they are. This may be because of image distortion that affects the shape of the chessboard pattern that gets projected onto the image. For the smaller angles, the error is small enough to be insignificant for the PTZ coordinate transformation; although for the greater angle, some compensation may be needed for coordinate transformation. For other applications, the accuracy may not be enough in either case as the error is around half a degree, which could prove to be too much if the camera is zoomed-in.

Table 5.2: The self-estimated relative angles and the relative angles estimated by the calibration application. The self-estimated relative angles (denoted as "real" here) are approximated by comparing the pan angles of the two cameras when looking at a defined point in the real world when mounted at the same position.

Real relative angle	Estimated relative angle
5°	5.7°
20°	20.4°
30°	28.0°
45°	42.3°
53°	49.8°

5.3 PTZ coordinate transformation

Table 5.3 shows a few PTZ coordinate transformation calculations based on two cameras and their estimated parameters. The results indicate relatively acceptable calculations near the calibration point (ϕ, θ) and seem to worsen the further away the camera gets from the calibration position. The offset for the pan calculation is a bit higher than expected, with a maximum error of $\pm 10.5°$, and is considered unsatisfactory for the majority of the real-world cases. In contrast, the tilt angle is considered more reliable with a maximum error of $\pm 2.4°$.

This considerable big offset in the pan-angles indicates a latent error in either the theory or the implementation. Therefore further investigation is required to improve the error offset, as a lower error would benefit the handover process.

Despite this, this offset will not be noticed as much in the recordings for the cameras because of the big field of view. At zoom level 1 many PTZ cameras have a field of view of around 60°, making the error not so problematic as the object will still be in the field of view. Seeing as the object detection algorithm then will capture the object and start tracking it, the error has less importance.

Table 5.3: Table for PTZ coordinate transformation when moving camera 2 to the position of camera 1. The expected position of camera 2 is measured by hand with an accuracy of 0.1°, and the error offset is calculated from the position of camera 2 and the expected position. The parameters for these coordinate transformations are: $\phi_1 = -28.9°$, $\theta_1 = -24.9°$, $\omega = 20.4°$, $h_1 = 10.2°$, $h_2 = 9.0°$, $d_1 = 22.1$m and $d_2 = 22.2$m

Camera 1		Camera 2		Expected camera 2		Error offset	
pan	tilt	pan	tilt	pan	tilt	pan	tilt
$\phi_1 - 20°$	$\theta_1 - 20°$	$+97.7°$	$-43.8°$	$+92.9°$	$-41.4°$	$-3.8°$	$+2.4°$
$\phi_1 - 10°$	$\theta_1 - 10°$	$+102.0°$	$-32.4°$	$+91.6°$	$-31.0°$	$-10.4°$	$+1.4°$
ϕ_1	θ_1	$+87.0°$	$-22.0°$	$+93.3°$	$-21.5°$	$+6.3°$	$+0.5°$
$\phi_1 + 10°$	$\theta_1 + 10°$	$+88.2°$	$-12.8°$	$+96.3°$	$-12.4°$	$+8.1°$	$+0.4°$
$\phi_1 + 20°$	$\theta_1 + 20°$	$+90.3°$	$-4.2°$	$+100.8°$	$-2.7°$	$+10.5°$	$+1.5°$

5.4 Handover process

For the handover process, multiple videos were recorded to observe the behavior of the cameras when objects enters the site. Figure 5.5 shows screenshots from a recorded video where the handover was performed, divided into four steps. The first step shows the idle position of both cameras, oriented towards their home position. At the moment an object comes into the scene, one of the cameras will start tracking it as shown in the second step. When the object moves into the handover area, the secondary camera will adjust its orientation and be ready for handover as indicated in the third step. For the fourth and final step, the handover has been performed and the secondary camera is now the tracking camera, whiles the previous camera is moved to its home position. An example video of the process can be found on Youtube [1].

In the final implementation, i.e. after finetuning parameters based on video recordings, the handover area was defined as the camera's field of view (60.0°) divided by the arbitrarily chosen value 6. This value gave the best results as it allowed the objects to be visible long enough for the tracker to identify them and start tracking. The position of the handover area was chosen to be where the calibration targets had been placed, as this position offered the lowest error offset for PTZ coordinate transformation. Having a bigger handover area would result in the handover process being performed earlier, which is not always suitable as the handover has a higher probability of being performed successfully near the calibration targets, where the coordinate transformation is more precise.

To improve the handover process a small pan offset was added to the PTZ coordinate transformation, which depended on the velocity vector of the object. This often resulted in a pan change in the interval of $\pm 2.0°$. This allowed the tracker more time to detect the object and track it and was proven to be considerably useful. Further expansion could be to apply this for the tilt angle, although as it does not deviate as much it was not deemed necessary.

This whole solution operates under the assumption that no other objects are to enter the site during tracking, as this may confuse the tracker. To prevent this, the tracker is disabled on the non-tracking camera at certain moments. Instead, this gives the convenience of having constant surveillance of the object at the site.

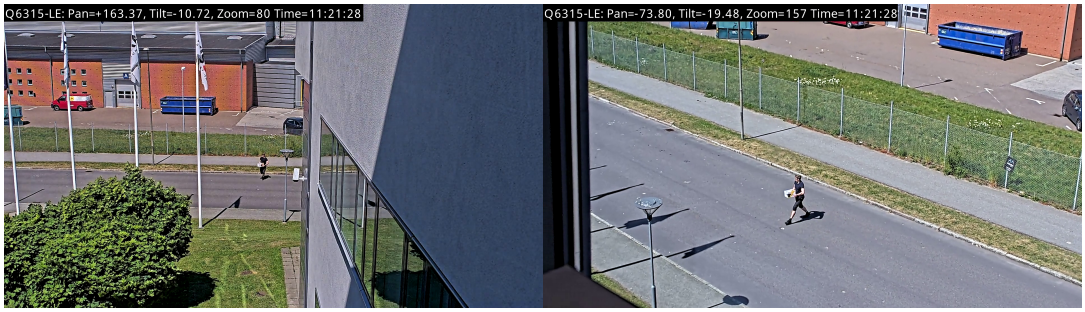
Measuring the internal state of the handover process showed that the camera position is polled roughly 10 times per second, which provided enough time for the system to decide when the camera is in the handover area. This means that the handover area

[1] Video of the Handover process `https://youtu.be/SDr-tUWDmlk`

(a) Step 1: Both cameras at home position

(b) Step 2: Camera starts tracking

(c) Step 3: Move secondary camera to handover area



(d) Step 4: Handover from one camera to another

Figure 5.5: Screenshots of a video where handover is performed. The target can be seen coming in from the right camera and moving alongside the building when handover is performed just as the target goes out of sight for the first camera.

is checked with a frequency of 10Hz. Any slower polling time proved to make the system too slow, increasing the chance of losing the object. An alternative method for this would be to perform the check of the area in the camera and send out an event to the server when the camera is in said area.

Using a centralized server to control when to perform the handover proved to not cause any issues, as no delays were observed. This approach was chosen as it provided the easiest path for implementation, as the cameras are already configured in the server and can be easily controlled through various interfaces. Despite this, camera-to-camera communication should perform quicker in theory, with the disadvantage of being harder to configure as the cameras would need to search for other cameras on the same network. The advantage of quicker communication was deemed not to be necessary for this application and thus a centralized solution was more suitable.

Another alternative tracking method that was considered during the project was not to perform handover at a pre-defined position, but to instead have constant tracking as long as the object is in the camera view. This would guarantee having the object under constant surveillance assuming the cameras can view the object, with the disadvantage of potentially missing new objects entering the site. In terms of all-around surveillance, it would be also beneficial to add 360°-cameras to the site, which would alert the system if multiple other objects were to enter the site giving the system the necessary data to divide the responsibility of tracking across multiple cameras. This was deemed to be out of scope for the project but could prove to be an interesting approach in future work.

The benefit of the handover process structure is that it can easily be expanded to a system of multiple cameras. This would in principle be to chain multiple handover areas together, that is defining which two cameras to perform handover between. The calibration procedure is also flexible enough to calibrate the cameras against each other, allowing for PTZ coordinate transformation across multiple cameras.

5.5 Camera tracking

The camera tracking application could be configured to use various object detection algorithms, but none proved to give satisfying results for quickly detecting a moving object and then maintaining it through the scene. The motion detection algorithm was regarded as relatively quick at detecting movement with the downside of giving multiple false alarms on non-interesting objects such as moving trees. The complexity of using this algorithm with a PTZ camera that moves could provide the reason

for the staggering amount of false alarms. The object analysis on the other hand results in fewer false alarms, but operates at a slower rate. It could also sometimes wrongly classify objects that are not human or vehicles, e.g. traffic cones.

The issues observed because of the object detection algorithms could be partially compensated either by moving the secondary camera a bit earlier or to calculate a position that took into consideration the velocity vector of the object. This would then give the detection algorithm more time to identify the object and start tracking it, resulting in smoother results. The preferred method is to utilize the object's position and velocity vector as this would yield good estimations of the new position for the camera. Moving a bit earlier could sometimes result in the object moving out of view.

Future application in this area is to control one PTZ camera manually and allow all other PTZ cameras in the system to follow the same target, in a pre-defined area. This type of master-slave system would result in an object being observed from multiple perspectives thus granting the observer more details of the target's actions.

Paper [5] describes PTZ camera assignment and handover as a planning problem that can achieve optimal camera assignment with concern to predefined observational goals such as distance to the target, PTZ limits and handover success probability. Expanding this project with planning theory could greatly improve the all-around surveillance offered by the handover process and decreases the probability of missing vital objects to track.

Chapter 6

Conclusion

In this master's thesis project, the handover process was examined as an expansion to conventional tracking to investigate its improvement on all-around surveillance by guaranteeing that an object entering a site would be under constant surveillance across multiple cameras. The method consisted of a pre-calibration stage for estimating the relative positioning and orientation of multiple PTZ cameras in a system and a novel control method for tracking only one object at a site. Necessary steps included the ability to perform coordinate transformation between two PTZ cameras, as well as defining the area in which to perform handover at.

The results demonstrate that the handover process could be performed with rather satisfying results, allowing for an object to be constantly tracked across a site. The optimal method of estimation for the camera mounting height was found to be using the built-in laser meter but requires a hardware feature not available on all PTZ cameras. Thus the estimation algorithm using the calibration targets is preferred in the general case if high precision is not necessary. PTZ coordinate transformation suffers from error offset and is not precise enough for applications that require high accuracy. Despite this, it demonstrated to be sufficiently adequate for the tracker to identify and track the object, with the extension of orienting the camera such that it takes into account the target's future position.

The findings of this project imply that all-around surveillance can be greatly improved by enhancing the interoperability between the cameras in a system. The recorded videos demonstrated that this new novel control method guaranteed that constant tracking of an object across a site is fully realizable. As a final statement, the method proposed in this project proved to be a good expansion of conventional tracking and could lay the foundation for future tracking and surveillance methods.

6.1 Future work

As discussed in various sections in Chapter 5, proposed expansions for this project can be investigated to either improve the accuracy of certain calculations or to expand the application of the handover process.

To improve the calibration procedure, more complex calibration targets could be used to further increase the probability of finding the targets and the features. This could be realized by dynamically configuring the targets' characteristics beforehand, adding more complexity to contrast against the surrounding scene of the targets. To make it more flexible, the camera could search its whole coordinate sphere for the target, assuring that the target will be found without the need of supplying an initial guess on the pan and tilt angles.

The current error offset found in the pan calculation for the PTZ coordinate transformation is considered to be grave and in need of improvement. This entails unraveling the cause of the miscalculation and revising the calculations, to minimize the error offset.

For the handover process, tilt compensation could be added that is based on the object's velocity vector. In terms of all-around surveillance, the project could be expanded by adding other types of cameras, such as 360°-cameras. Although the most interesting case is to expand the solution to multiple cameras, easily implemented by chaining handover areas across cameras to know which cameras to perform handover between.

For camera tracking, the most important improvement is to improve upon the object detection algorithm for faster detection of new objects. Once object detection will be improved, a possible expansion could be to implement a master-slave control method to capture an object from multiple angles simultaneously. Alternatively, to support tracking of multiple objects using planning theory as described in previous research.

Bibliography

[1] Chen, I.-H. and Wang, S.-J. "An Efficient Approach for Dynamic Calibration of Multiple Cameras". *IEEE Transactions on Automation Science and Engineering* 6.1 (Jan. 2009), pp. 187–194. DOI: 10.1109/tase.2008.918128.

[2] Chen, I.-H. and Wang, S.-J. "An Efficient Approach for the Calibration of Multiple PTZ Cameras". *IEEE Transactions on Automation Science and Engineering* 4.2 (Apr. 2007), pp. 286–293. DOI: 10.1109/tase.2006.884040.

[3] Chen, I.-H. and Wang, S.-J. "Efficient Vision-Based Calibration for Visual Surveillance Systems with Multiple PTZ Cameras". *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*. IEEE, (2006). DOI: 10.1109/icvs.2006.22. URL: https://doi.org/10.1109/icvs.2006.22.

[4] Dinh, T., Yu, Q., and Medioni, G. "Real time tracking using an active pan-tilt-zoom network camera". *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, (Oct. 2009). DOI: 10.1109/iros.2009.5353915. URL: https://doi.org/10.1109/iros.2009.5353915.

[5] Qureshi, F. Z. and Terzopoulos, D. "Planning ahead for PTZ camera assignment and handoff". *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*. IEEE, (Aug. 2009). DOI: 10.1109/icdsc.2009.5289420. URL: https://doi.org/10.1109/icdsc.2009.5289420.

[6] Szeliski, R. "Computer vision: Algorithms and applications". 2nd Edition. Springer, (2021).

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
June 2023

Document Number
TFRT-6216

Author(s)
Alexander Persson

Supervisor
Paul Steneram Bibby, Axis Communication, Sweden
Kenneth Ekman, Axis Communication, Sweden
Yiannis Karayiannidis, Dept. of Automatic Control, Lund University, Sweden
Johan Eker, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
PTZ Handover: Tracking an object across multiple surveillance cameras

Abstract

Tracking objects in a scene is a crucial task in accomplishing surveillance that enhances security and provides valuable information about the events happening at the site. For this task, the PTZ (pan-tilt-zoom) cameras can be utilized to achieve fluid tracking as they provide all-around surveillance with zoom capabilities. The drawback of current tracking solutions is the lack of interoperability between cameras, e.g. to signal the position of an object so that multiple cameras can track it simultaneously. This project highlights the importance of continuously tracking an object across a site and proposes a solution on how to handover the target from one camera to another. Thus the need of performing PTZ coordinate transformation is necessary to direct multiple PTZ cameras toward the same target. For simplicity, the scope of the project was limited to a system consisting of only two cameras, with a focus on tracking one object at a time. The method consists of two steps: namely to perform a calibration procedure to determine the spatial relationship between two cameras and to then track a single object across a site. The tracking process is handled by a centralized server, which determines which objects to track, where to position the cameras and when to perform the handover.

The results show that tracking an object across two cameras, mounted at different heights and located multiple meters apart, is fully achievable. Even though the built-in tracker can be perceived as slightly delayed, the handover functionality still managed to execute as expected even with the target moving at a moderately high velocity. The output of the calibration was found to be rather satisfactory, but could however be refined to achieve even higher accuracy. In conclusion, the proposed solution works well and entails that this kind of functionality may further enhance all-around surveillance. As future work, the calibration procedure can easily be expanded to multiple cameras, but tracking multiple objects at the same time requires advanced theoretical investigation.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-55

Recipient's notes

Security classification