Master Thesis

Joint human detection from on-board and off-board cameras

Author(s):
Mišeikis, Justinas

Publication Date:
2012

Permanent Link:
https://doi.org/10.3929/ethz-a-010075304

Rights / License:
In Copyright - Non-Commercial Use Permitted
Joint Human Detection from On-Board and Off-Board Cameras

Autumn Term 2012
# Contents

Abstract iii  
Symbols v  
1 Introduction 1  
2 Related Work 3  
3 Method 5  
3.1 Off-board Cameras 6  
3.1.1 Ground Homography 6  
3.1.2 Background Segmentation 6  
3.1.3 Noise Reduction and Filtering 7  
3.1.4 Blob Detection, Confidence Calculation, and Filtering 8  
3.1.5 Point Conversion to the World Coordinate Frame 9  
3.1.6 HMC Position Estimation and On-board Camera Position Estimation 10  
3.1.7 Finding Inlier Objects For Each On-board Camera 10  
3.2 On-board Cameras 11  
3.2.1 Camera Calibration and Image Undistortion 11  
3.2.2 People Detector - HOG 12  
3.2.3 Targeted Analysis 13  
3.2.4 Full Image Analysis 14  
3.3 Data Fusion 14  
3.3.1 Kalman Filter 14  
3.3.2 Object Tracking 15  
3.3.3 Hungarian Algorithm 15  
3.3.4 Off-board Camera Position Fusion 16  
3.3.5 Data fusion from off-board and on-board cameras 16  
4 Implementation 19  
4.1 Hardware 19  
4.1.1 Hot Metal Carrier - HMC 19  
4.1.2 Off-board Cameras 20  
4.1.3 On-board Cameras 20  
4.1.4 Computers 20  
4.2 Software and Frameworks Used 21  
4.3 ROS Nodes and Software Structure 21  
4.3.1 On-board Processing 22  
4.3.2 Off-board Processing 24  
4.4 ROS Messages 27  
4.4.1 Default Messages 27
Abstract

Pedestrian detection is one of the key functionalities of video-based safety systems. Research on this topic usually considers the detection from a mobile camera, on-board a vehicle, or from a static camera (off-board) mounted on infrastructure. In contrast, we propose a joint on-board and off-board detection approach to identify pedestrians. Assuming that the considered vehicle contains some sort of localization capability, we combine detection information from both sources. These sources are semi-independent, with substantially different illumination and view angle perspectives. This brings more statistical diversity than a multi-camera network observing an area of interest, for example. Therefore, combining off-board and on-board detection significantly increases the system reliability. The proposed method finds applicability in industrial environments, where industrial vehicle localization is becoming increasingly popular and there are areas of particular safety concern. Experiments illustrate the improved results of the joint multi-camera detection approach compared to methods which perform off-board and on-board solutions separately. Furthermore, a conference paper about this project was accepted and published in the ROSIN 2012 workshop taking place at the IROS 2012 conference.
Symbols

\[ a_{\text{cam}} \] Camera’s field of view
\[ A_{\text{blob}} \] Area of the blob
\[ AB \] Area Bottom
\[ AT \] Area Top
\[ A, B, C \] Coordinates of the triangle corners
\[ C_r \] Confidence level
\[ c_x, c_y \] Camera optical centre
\[ d_{\text{obj}} \] Distance between two objects
\[ d_p \] Distance to the person
\[ d_{\text{sensor}} \] Camera sensor size
\[ d_x, d_y \] Padding in horizontal and vertical directions
\[ D_{\text{conf}} \] Distance confidence level
\[ D_c = \left[ k_1 \ k_2 \ p_1 \ p_2 \ k_3 \right] \] Camera distortion parameter vector
\[ f_{\text{lens}} \] Focal length of the lens
\[ f_x, f_y \] Focal length in horizontal and vertical directions
\[ F_t, D_t \] Control inputs (Kalman filter)
\[ G \] Transition matrix
\[ h \] Height
\[ H \] Homography matrix
\[ H_{1,1} \] Element of the homography matrix
\[ p_h \] Person height
\[ P \] Coordinates of the point
\[ s_r \] Resize ratio
\[ t_{\text{lost}} \] Elapsed time when tracking is lost
\[ T_{\text{conf}} \] Distance confidence threshold
\[ T_{\text{ka}} \] Timeout threshold
\[ u_t \] Observation noise
\[ u, v \] Vertices in the barycentric coordinate system
\[ w \] Width
\[ w_t \] Process noise
\[ x_{c_r, y_c} \] Coordinates in the Camera Coordinate System
\[ (x, y, \theta) \] HMC pose
\[ x_{t+1} \] Current time step (Kalman filter)
\[ x_t \] Previous time step (Kalman filter)
$x_{tc}, y_{tc}$  
Tangential distortion correction

$x_{rc}, y_{rc}$  
Radial distortion correction

$x_w, y_w$  
Coordinates in the World Coordinate System

$\sigma$  
Measurement accuracy

Indices

$x$  
$x$ axis

$y$  
$y$ axis

Acronyms and Abbreviations

EKF  
Extended Kalman Filter

ETH  
Eidgenössische Technische Hochschule

CCD  
Charge-Coupled Device

CPU  
Central Processing Unit

CSIRO  
Commonwealth Scientific and Industrial Research Organisation

DDX  
Dynamic Data eXchange

FOV  
Field Of View

FPS  
Frames Per Second

GB  
Gigabyte

GHz  
Gigahertz

GPS  
Global Positioning System

GPU  
Graphics Processing Unit

HMC  
Hot Metal Carrier

HOG  
Histograms of Oriented Gradients

IDL  
Interface Description Language

IP  
Internet Protocol

MOG  
Mixture Of Gaussians

OpenCV  
Open Source Computer Vision Library

RAM  
Random Access Memory

ROI  
Region Of Interest

ROS  
Robot Operating System

SSD  
Solid-State Drive

SVM  
Support Vector Machine

URI  
Uniform Resource Identifier
Chapter 1

Introduction

Research in video-based pedestrian detection has seen a dramatic increase in the last decade. Applications include home care, patient monitoring, pedestrian flow counting, security and industrial safety. In industrial environments, pedestrian detection can monitor the presence of people in restricted or hazardous areas. It can also alert machinery or vehicle operators about any pedestrians in the vicinity. Detection can be performed from mobile cameras, on-board vehicles, as well as from static cameras mounted on infrastructure, which we refer to as off-board cameras in this thesis.

Focusing on industrial environments, we propose a novel system which integrates the detection from these two different and independent sources: on-board and off-board cameras. With increasing accuracy and dropping price of sensors, self-driving autonomous vehicles are becoming more popular, especially in industrial environments. Even if the vehicle can perform its task perfectly well, one concern still stands, should people be allowed to work in the same environment where these vehicles operate and how to guarantee safety. Many current methods have an emergency stop procedure upon the detection of an object in the path of the vehicle using range sensors. Although it often performs satisfactory, no information is provided on the type of the object, whether it is a human or some other unexpected static or moving object. This additional information could prove to be very beneficial in the artificial intelligence of the vehicle and for the calculation of an alternative route. This combination can be achieved whenever vehicles have a reliable localization system, such that the vehicle pose \((x, y, \theta)\) in the environment can be estimated. Although this assumption can restrict the system, vehicle localization is becoming increasingly common in industrial environments, performed through laser, GPS, radio-frequency identification and radar solutions [11], motivating this research.

Off-board cameras provide traditional pedestrian detection as often applied in surveillance applications [31, 32]. Generally, background segmentation is used to extract moving objects and the position in real world coordinates is estimated with knowledge of the homography. By estimating the current pose of the vehicle and a relative on-board camera placement with respect to the vehicle localizer reference, it is possible to identify the region (in real-world coordinates) which falls into the camera’s field of view (FOV). When a pedestrian is detected by the off-board detection system, a message is sent to the vehicle indicating that a moving object is within the FOV of the on-board camera. On-board cameras extract the indicated area of interest and analyze it in detail while searching (in parallel) the remaining part of the image for any pedestrians that were missed by the off-board cameras. Upon the detection of a person, his/her position in the workspace is calculated by combining estimations of both on and off-board cameras as well as estimating the level of confidence. Camera placement and detection accuracy is taken into consideration.
during the data fusion process for each of the sensor as well as the capability of
the system functioning correctly when the data from one or more sensors is not
available or is incorrect.
The main advantage of having cameras placed both on-board and off-board is that it
provides different viewpoints and increases the detection accuracy, especially under
full or partial occlusion of the subject, which can become a very significant problem
in the environments with many buildings and no complex areas. People can be
blocked by the corner (blind spot) of the building and can be standing on the
planned route of the vehicle. With our proposed method this problem can be averted
provided off-board cameras have sufficient coverage. Furthermore, the detection is
based on a combination of monocular cameras which are often likely to be already
present both on buildings or structures and the vehicle.
We present experimental results from a real industrial environment in various con-
ditions. We compare the joint detection with the traditional independent detection
approaches (i.e. separately for the off-board and on-board cameras), illustrating
and quantifying the improved performance. We also compared against the off-board
multi-camera setup, in order to evaluate against other multi-source methods.
This thesis is organized as follows. In Chapter 2 we discuss related work. We present
a detailed description of the method in Chapter 3 and discuss the implementation
details in Chapter 4. Experimental results are presented in Chapter 5, followed by
the relevant summary and future work in Chapter 6.
Chapter 2

Related Work

Research on pedestrian detection from moving platforms usually focuses on cameras mounted on vehicles, targeting urban environments [20] [21]. In industrial scenarios, driver support systems, which provide a warning signal if a person enters a blind area have been developed [22]. Alternatively, warning signals can be sent to vehicles if there is a collision potential between the vehicle and pedestrians in its path [23]. Tracking dense crowds introduces a challenge because of obstructions and the fact that people can move close to each other, often changing direction and velocity. The problem was addressed by using static synchronized multi-camera systems observing the same area and using homography constraints calibrated for different heights from the ground plane [24] [27].

Research has also been done on people and car detection from a moving vehicle through an urban area, as seen in Figure 2.1, using a stereo camera configuration as well as from a fixed camera observing areas like a pedestrian crossing [25]. Given the depth information, many false positives can be filtered out [35], however, additional computational power is needed to obtain a depth map and stereo camera hardware costs significantly more compared to monocular cameras.

Figure 2.1: Pedestrian detection as a driver assistance system. Image taken from [35].

An extensive study was completed on pedestrian detection from moving vehicles using monocular cameras by evaluating sixteen state-of-the-art algorithms on the six popular datasets [28]. Results have shown that occlusions are common with
70% of pedestrians being occluded in at least one frame and a solution using motion estimation of the person is not effective when the camera is mounted on the moving vehicle. Detection rates varied from 70% when the pedestrian is in close proximity to the vehicle to only 20% success at a medium distance, pointing out a major drawback even with the most advanced people detection algorithms currently available.

Some research has shown a substantial improvement in the accuracy of the pedestrian detection over the standard HOG algorithm by introducing self-similarity features and color information as well as encoding image motion [34].

![Figure 2.2: People detection and tracking at the pedestrian crossing using a static camera. Image taken from [25].](image)

An interesting approach combining imagery from fixed and moving cameras is proposed by A. Alahi et al. [26] performing a rapid search of objects using a master-slave approach with fixed and mobile cameras. The concept is to create a descriptor for the region of interest using an off-board camera image and identify the same object in the mobile camera image. The problem they approach is similar to ours, however, mobile cameras are not localized, resulting in the continuous correspondence search in the whole image and object classification is not performed.

In contrast to the works above, we perform joint detection, combining information from both off-board and on-board cameras, exploiting the knowledge of the vehicle position.
Chapter 3

Method

The basis of the proposed method lies in the combination of off-board and on-board cameras to improve the detection of people. The process is illustrated in the diagram in Figure 3.1. The system consists of two main processes performed in parallel. The first stage, marked by the top bounding box in the figure, considers the off-board camera image and the pose of the vehicle (or other moving platform). Using background segmentation it identifies moving objects in the environment. The position of moving objects is then sent to the on-board computer, where the second process uses a descriptor based method to detect people in the on-board camera image. The full on-board image is analyzed. In addition, a more detailed analysis of the areas of interest indicated by the off-board process is performed. Finally, positions of detected people in the world coordinate frame are obtained by combining detector outputs from both cameras. This chapter explains the method in detail.

Figure 3.1: Diagram illustrating the processing steps in the methods. The green box corresponds to the off-board processing, whereas the red box corresponds to the on-board processing. The inputs to the system are represented by the blue arrows, and the final output is given by the red arrow.
3.1 **Off-board Cameras**

The purpose of cameras placed on buildings is to monitor the workspace and in our work to provide information to the on-board camera mounted on the vehicle. The steps listed next are performed to obtain the position of human-like objects in the workspace.

Although the description below focuses on a single off-board camera, in our setup two off-board cameras are used observing the same area, which increases the robustness to occlusions by providing more than one off-board viewpoint. The process on both cameras is identical, although different camera-specific parameters are used.

### 3.1.1 Ground Homography

Ground homography is needed to convert points from image pixels to physical coordinates of the workspace. It is a type of projective transformation which describes the change in positions of objects when the point of view of the observer (camera) changes, in our case, we get a top-down view. The resulting plane is the ground plane of the workspace assuming it is approximately flat. For uneven surface, there would be deviations in accuracy depending on the position, and more sophisticated assumptions are necessary.

In order to estimate the homography, four distinct points must be identified with a known physical position, as shown in Figure 3.2 and their corresponding pixel coordinates must be found in the camera image [5]. A $3 \times 3$ homography matrix $H$ is calculated to be used for point coordinates conversions between two frames:

$$
H = \begin{bmatrix}
H_{1,1} & H_{1,2} & H_{1,3} \\
H_{2,1} & H_{2,2} & H_{2,3} \\
H_{3,1} & H_{3,2} & H_{3,3}
\end{bmatrix}
$$

![Figure 3.2](image.png)

Figure 3.2: Four points identified in our workspace with precise image pixel positions in red, and real world coordinates (in meters) in green for the ground homography calculation.

### 3.1.2 Background Segmentation

The most efficient method of extracting moving objects in continuous image stream from a static camera is background segmentation. The basic idea is based on statistically modeling the background and subtracting it from the current camera image to identify any objects not belonging to the background. However, a constant update
of the background model is needed by adding new samples and removing old ones to accommodate for new static objects (like a parked car) or old ones disappearing or being relocated.

A number of different background segmentation methods exist, however, evaluations have shown that a Mixture of Gaussians (MOG) algorithm performed among the best in terms of accuracy and processing speed [4] [3]. The idea behind the MOG method is to use a mixture of an adaptive number of Gaussians for each channel of the pixel. It is a process of high complexity, so an efficient implementation is reused in our application. The original description can be found in [16].

![Figure 3.3: Standard background segmentation process workflow. Graph from [3].](image)

### 3.1.3 Noise Reduction and Filtering

Post-processing foreground images can significantly improve the background segmentation where noise and dynamic background objects like tree leaves or safety lines are being moved by the wind and can introduce false positive detections [3]. Simple post-processing techniques are used to remove the unwanted false objects from the foreground image. In our implementation, a median filter is used for an efficient noise removal. The technique is based on a sliding window scanning the whole image and the value of the pixel in the middle of the window is replaced by the median of neighbouring pixel values. It is commonly used to filter ‘salt and pepper noise’.

The second part of the filtering is morphological closing. Remaining small objects and small holes inside objects are removed during this process. It is a combination of two methods: first erosion, followed by dilation of the resulting image. In simple terms, a structuring element is used to probe and shrink the shapes in the input image during the erosion process, while in dilation, shapes are expanded by the radius of the element. An example of the results can be seen in Figure 3.4.

![Figure 3.4: Results of dilating or eroding a simple binary image containing the letter ‘j’ (middle). Letter ‘j’ images from [2]](image)
3.1.4 Blob Detection, Confidence Calculation, and Filtering

Distinct object identification in a foreground binary image can be performed using a blob detector. Each separate object is labeled and the following information extracted: position, area size, contour and a bounding box. An algorithm proposed by Fu Chang et al. [1] was chosen for its reported efficiency. The method is based on the principle that any object is determined by its contours and the detection is completed in a single pass over the image, however, some contour points are revisited more than once. The analysis is done by scanning the image from top to bottom and left to right in each line and as soon as white (object) pixels are encountered, a contour tracing technique is used, both for internal and external contours and assigning labels. The principal functionality of the blob detector is shown in Figure 3.5.

![Figure 3.5: Blob detection based on contour-tracing procedure. Image from [1]](image)

Considering an object to be an upright standing person, a simple measure of confidence is implemented using a ratio between width and height of the blob. Popular people detection algorithms utilize the search window of the aspect ratio - height is twice the width, to detect a person [29]. Based on the same assumption, the calculation \( C_r = 1 - |1 - 2 \times \frac{w}{h}| \), where \( h \) and \( w \) are height and width, respectively, and \( C_r \) is the confidence level, returns 1 as a perfect match, while deviations from the previously mentioned ratio reduce the confidence level, which can also be negative in cases when the width of an object is larger than the height.

Detected blobs are filtered out by using the calculated confidence as well as the area size, \( A_{blob} \) of the object, allowing an easy distinction between people and large moving objects, like vehicles. However, given the possible variability in person’s size because of different height and body build, carrying various objects and even two people walking close together which can be merged and identified as one blob, appropriate margins have to be used. Furthermore, the size of the person in the image can significantly differ depending on their distance from the camera. Instead of fully adaptive area size estimation, a simplified solution of splitting the image into two areas according to the \( y \) coordinate, as shown in Figure 3.6. Then separate minimum and maximum object size thresholds for each of the splits ‘Area Top’ and ‘Area Bottom’, \( AT_{min}, AT_{max} \) and \( AB_{min}, AB_{max} \) respectively, are applied. The filtering criteria can be seen below.

\[
\text{blob}(y, A_{blob}) = \begin{cases} 
\text{True} & \text{if } y < I_{height} \\
\text{False} & \text{otherwise} \\
\text{True} & \begin{cases} 
\text{if } AT_{min} < A_{blob} < AT_{max} \\
\text{False} & \text{otherwise} 
\end{cases} \\
\text{True} & \begin{cases} 
\text{if } AB_{min} < A_{blob} < AB_{max} \\
\text{False} & \text{otherwise} 
\end{cases} 
\end{cases}
\]
3.1.5 Point Conversion to the World Coordinate Frame

Camera specific ground homography matrix $H_{\text{cam}}$ is used to convert the position of the person from pixel positions, $x_c$ and $y_c$, in the camera image to the world coordinate system, $x_w$ and $y_w$, defined the workspace, measured in meters. The bottom center point of the bounding box is used as the approximation of the feet position, which ideally corresponds to the average of the feet positions [33]. Then a simple conversion is done, where $Z$ is a scale factor:

$$Z = \frac{1}{H_{3,1} \times x_c + H_{3,2} \times y_c + H_{3,3}}$$

$$x_w = (H_{1,1} \times x_c + H_{1,2} \times y_c + H_{1,3}) \times Z$$

$$y_w = (H_{2,1} \times x_c + H_{2,2} \times y_c + H_{2,3}) \times Z$$

(3.1)

The whole image conversion would result in a bird’s-eye view, as if cameras were overlooking the workspace directly down.
3.1.6 HMC Position Estimation and On-board Camera Position Estimation

HMC pose \((x, y, \theta)\), at a few centimeter accuracy \([11]\), is provided by the on-board localizer and can be used as the real position of the vehicle. Four on-board cameras are placed at the following known poses, in the format \((x, y, \theta)\) and measured in meters, on HMC in relation to the vehicle’s coordinate system located in the center of the rear wheel axis:

- Front far proximity camera: \((0, 4.2, 0^\circ)\)
- Front close proximity camera: \((0, 4.2, 0^\circ)\)
- Left camera: \((-0.98, 3.14, 90^\circ)\)
- Right camera: \((0.98, 3.14, -90^\circ)\)

Simple transformation matrices including rotation and translation are used for recalculating points from the HMC coordinate frame to the camera coordinate frame. When a point is specified in the world coordinate system, the following transformation could be used to get it in camera’s coordinates: world coordinate frame \(\rightarrow\) HMC coordinate frame \(\rightarrow\) camera coordinate frame. For the opposite transformation, an inverse matrix can be used.

3.1.7 Finding Inlier Objects For Each On-board Camera

Having on-board camera positions and transformations allow to determine which known foreground objects are in each camera’s field of view (FOV) at the time instance. FOV \(a_{\text{cam}}\) depends on the sensor size \(d_{\text{sensor}}\) and lens focal length \(f_{\text{lens}}\). It can be calculated (in radians) using the following equation:

\[
a_{\text{cam}} = 2 \times \arctan\left(d_{\text{sensor}} \div (2 \times f_{\text{lens}})\right) \quad (3.2)
\]

Using on-board camera positions and FOVs, an observed area can be defined using multiple triangles in the world coordinate frame, as shown in Figure 3.8. An efficient way to determine inliers within the camera’s observed area defined as triangles is to use a barycentric coordinate system \([30]\).

This can be determined with the following equation:

\[
P = A + u \times (C - A) + v \times (B - A) \quad (3.3)
\]

where \(P\) represents point coordinates, \(A\), \(B\) and \(C\) represent triangle corner coordinates, and \(u\) and \(v\) are vertices to be found. The following conditions must be met for \(P\) to be inside the triangle:

\[
(u \geq 0) \quad \text{and} \quad (v \geq 0) \quad \text{and} \quad (u + v < 1). \quad (3.4)
\]

In our implementation all points (objects) can be classified into three main groups: ‘unseen’, ‘seen’ and ‘danger’. The latter is defined as being within the camera’s observed area, but excessively close to the vehicle for the whole object to fit into the on-board camera frame.
3.2 On-board Cameras

The final people detection process is performed on the on-board camera images with the support of data provided by the off-board computer. Two main approaches are being used and eventually combined - the analysis of the whole image as well as a detailed analysis of the areas indicated by the off-board cameras. This section describes all the preparation steps and the main computation to finalize the people detection.

3.2.1 Camera Calibration and Image Undistortion

Lens distortion is a common issue, especially with the wide angle lenses, which can cause precision issues when a pixel-accuracy is needed. However, these undesired effects can be minimized in software by obtaining an accurate camera model \([12]\). Five distortion parameters \(D_c\) are obtained during the calibration process:

\[
D_c = [k_1 \ k_2 \ p_1 \ p_2 \ k_3]
\]  

(3.5)

and can be used to correct the radial distortion \((x_{rc}, y_{rc})\):

\[
x_{rc} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)
\]

\[
y_{rc} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)
\]

(3.6)

as well as tangential distortion \((x_{tc}, y_{tc})\):

\[
x_{tc} = x + [2p_1 xy + p_2(r^2 + 2x^2)]
\]

\[
y_{tc} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]
\]

(3.7)
where \( x \) and \( y \) are the original input coordinates. In order to achieve better accuracy and possibly estimate object sizes in physical units, a camera matrix, containing focal lengths \( f_x, f_y \) and optical centres \( c_x, c_y \), is calculated. For the unit conversion, a simple matrix multiplication is needed:

\[
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix} =
\begin{bmatrix}
  f_x & 0 & c_x \\
  0 & f_y & c_y \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  X' \\
  Y' \\
  Z
\end{bmatrix}
\]  

(3.8)

The actual calibration process can be done using a known classical black-white chessboard pattern presented to the camera at varying angles, translations and distances [13].

### 3.2.2 People Detector - HOG

Following an extensive research on the existing pedestrian detection methods using just monocular cameras [20] we employ a Histogram of Oriented Gradients (HOG) algorithm [29]. The HOG method utilizes the idea that local object appearance and shape can be characterized by edge directions, even without a precise knowledge of the corresponding gradient or edge positions. It uses contrast-normalized local histograms of image gradient orientations in a dense grid. Normalization is done by accumulating local histogram ‘energy’ over spatial regions, defined as blocks. Results are used to normalize all of the sub-regions (cells) in the block, thus providing better invariance to illumination. Collected information in the detection window is then combined in a feature vector and used as an input in a linear SVM for classification. The whole HOG process is outlined in Figure 3.9 and intermediate descriptor images presented in Figure 3.10.

![Figure 3.9: Summary of the HOG process. The image is color and gamma normalized, then tiled into the detector windows and Histogram of Oriented Gradients computed. Combined HOG vectors are used in a linear SVM classifier to identify human and non-human objects. Figure taken from [29].](image)

![Figure 3.10: (a) The average gradient image. (b) The maximum positive SVM weight in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) Input image. (e) Computed R-HOG descriptor. (f,g) The R-HOG descriptor weighted by respectively the positive and the negative SVM weights. Figure taken from [29].](image)

The effect of translations and rotations is not significant if they are much smaller than the local spatial or orientation bin size. This property is utilized in human
3.2. On-board Cameras

detection, because it permits limbs and body segments to change appearance and move around a pivot point as long as a coarse spatial sampling, fine orientation sampling and strong local photometric normalization is used.

The HOG method uses a Latent SVM classifier [17] to perform the matching and it also returns a score. Score is the sum of all the features matching between the extracted region of an input image and a trained model, minus a deformation cost allowing the deviation of the part from its ideal location. The score is used as a confidence measure in HOG matching.

3.2.3 Targeted Analysis

Points of interest indicated by the off-board camera detector are in the camera coordinate system and can be directly converted to the pixel position by using the inverse of the ground homography matrix for the given camera. According to the distance from the camera, the height $h_p$ of the person in pixels can be estimated and the region of interest (ROI) containing the whole person is extracted. In order to compensate for the position inaccuracies, paddings $d_x$ and $d_y$, in our case both set to 100% of the original dimension, are used. The region is adaptively resized by a ratio $s_r = d_s / h_p$ according to the estimated height $h_p$ to match the descriptor sample size $d_s$, (in our case 128 pixels), before the detection is performed. Upon the successful HOG detection, the same method of taking the bottom center point of the bounding box approximating feet position is converted to the camera coordinate frame which can be converted to physical units in the world coordinate frame later on. The concept of the targeted analysis is shown in Figure 3.11.

![Targeted analysis method](image.png)

Figure 3.11: Targeted analysis method. The red circle is the position of the person defined by the off-board processing. Green rectangle defines the estimated ROI which is extracted and resized. Red rectangle is the actual detection of the person. In this case, the off-board indication has a false offset caused by a shadow, which is understood as an elongation of an object by the background segmentation process, however given margins for the search window still allow the successful detection.
3.2.4 Full Image Analysis

Given the ground homography and the fact that majority of the people are 1.50 m to 2.00 m tall, the image could be split into several areas according to the object distance from the camera. This allows the adjustment of the partial image size so human-like object height would be as close as possible to the training samples’ height. Figure 3.12 shows a division into three regions according to the distance from the camera marked on the ground plane. Each of the regions is then analyzed using the HOG detector and any resulting detections fused back into one system based on the camera coordinate frame. The goal of this stage is to detect people who could have been missed by the off-board cameras.

![Figure 3.12: The camera view is split according to the distance from the camera: blue (3-7 meters), green (7-12 meters), yellow (12-20 meters).](image)

3.3 Data Fusion

Having a number of sensors, especially visual ones, performing simultaneous detection can be complicated due to noise, imperfections and inconsistent error rates. The HOG descriptor, for example, performs satisfactorily in classifying the object, but suffers from location precision, especially when the distance from the camera is estimated based on the ground homography. Sensor fusion is essential to get more accurate results, especially given a precise modeling of the measurement noise. Given our own experience and other projects, an Extended Kalman Filter (EKF) is a proven robust way to perform a multi-sensor fusion [14].

Two main stages of the sensor fusion were identified, with the first one combining the data from both off-board detectors and the second stage incorporating HOG results from the on-board detector as well as confidence data, as discussed in Section 3.2.2.

3.3.1 Kalman Filter

The Kalman filter is based on a system’s dynamic model using a multiple sequential sensor measurements to estimate the system’s state. This state estimation has higher accuracy compared to one sensor measurement by adding an unreliable sensor
measurement as long as its model is defined correctly. It is a recursive estimator, which could be split into two main steps: prediction and update. In simplified terms, a prediction step uses the accumulated state estimate from previous timesteps to estimate the state at the current time:

\[ x_{t+1} = G_{t+1} \times x_t + F_t \times u_t + w_t \]  
(3.9)

and in the update the output of the predict step is taken as a priori and combined with current observations (measurements) to get the state estimate, a posteriori:

\[ y_t = H_t \times x_t + D_t \times u_t + v_t \]  
(3.10)

where \( x \) is the unobservable state of the system, \( y \) is an observable output, \( G \) is a known transition matrix, \( w_t \) and \( v_t \) are variables with defined zero mean white noise, \( F_t \times u_t \) and \( D_t \times u_t \) are inputs.

The benefit of a Kalman filter is low memory usage, because only the accumulated model from the last timestep is stored instead of the whole measurement history. The Extended Kalman Filter works by the same principle, however it is able to model non-linear systems by linearizing about the estimate of the current mean and covariance [15].

### 3.3.2 Object Tracking

Given the situation of multiple people present in the workspace at the same time instance, short-term tracking has proven to be beneficial for the fusion process and increasing robustness against occlusions or lost off-board detection. Short-term means that a person does not need to be tracked throughout the whole lifetime in the workspace and tracks of tens of seconds are sufficient for the system to function correctly as long as the re-detection occurs within few frames after the track is lost and a new ID provided. Tracked objects are defined as active objects.

### 3.3.3 Hungarian Algorithm

At each time instance, a number of detected objects can be returned by the off-board camera processing and given multiple active objects. The challenge is to identify wether the detection is a position update of already tracked object (and which one) or a new object entered the workspace. Hungarian algorithm [9] can be used to solve this assignment problem. All the data is represented by the cost matrix where rows are active objects and columns are new object position measurements with distances, \( d_{obj} \), between them populating the matrix, as seen in Figure 3.13. Hungarian algorithm is used to find the optimum assignment between the two with the minimal cost, in this example, marked in green.

<table>
<thead>
<tr>
<th></th>
<th>New 1</th>
<th>New 2</th>
<th>New 3</th>
<th>New 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actv 1</strong></td>
<td>4.582</td>
<td>1.32</td>
<td>10.84</td>
<td>7.23</td>
</tr>
<tr>
<td><strong>Actv 2</strong></td>
<td>0.71</td>
<td>2.65</td>
<td>4.68</td>
<td>8.872</td>
</tr>
<tr>
<td><strong>Actv 3</strong></td>
<td>3.45</td>
<td>5.97</td>
<td>2.87</td>
<td>6.16</td>
</tr>
</tbody>
</table>

Figure 3.13: Hungarian algorithm cost matrix. Rows, in blue, represent active objects, columns, in yellow, represent new object position measurements and matrix is populated by the cost, in this case, the distance between points. The optimum solution, with the minimal cost, is marked in green.
Additionally, a thresholding by the distance $D_{conf}$ and sensor measurement accuracy $\sigma$ was added to prevent the assignment of two far away points even when it is the most optimum solution using the equation:

$$D_{conf} = \exp\left(\frac{-(d_{obj})^2}{\sigma^2}\right)$$  \hspace{1cm} (3.11)

In this case, no assignment is done if $D_{conf}$ is larger than a defined threshold $T_{dist}$.

3.3.4 Off-board Camera Position Fusion

Off-board camera fusion consists of four stages with a Hungarian algorithm used to obtain the best assignment. The main goal is to identify which newly detected points in both off-board cameras correspond to which active objects and incorporate new measurements into the tracker. The process is split into four stages, as seen in Figure 3.14. The first step incorporates detections from camera A to the tracked objects using the previously mentioned filtering by the distance $D_{conf}$. The second step performs the same action with the detections from the camera B. Once the points are matched, they are removed from the new point list to avoid ambiguous matching. The last two steps are used to initialize new points, with the third one trying to find correspondences between points in both cameras (ignoring any previously detected active objects) and then initializing a new active object, while in the fourth step, new active objects are initialized for each remaining point left in the list of previous active points. This approach ensures that all the possible matches are considered and new active objects were created for any unmatched points.

A timing criteria is continuously used to identify ‘dead’ objects by using a simple thresholding:

$$t_{lost} < T_{ka} \begin{cases} 
\text{True} & \text{- keep} \\
\text{False} & \text{- discard}
\end{cases}$$

where $T_{ka}$ is defined timeout threshold and $t_{lost}$ is the time with lost tracking.

3.3.5 Data fusion from off-board and on-board cameras

The second fusion part is the addition of on-board detection data to active objects using the same method as described in the previous section. However, an additional measure provided in this case is the confidence provided by the HOG detector. The position is also updated, however, the main benefit of this fusion step is to classify current active objects as people and list the other ones as unknowns. Higher importance is given to points identified as people by increasing the $T_{ka}$ threshold, meaning that in case of a failed detection, the tracker will keep it longer in its active object list compared to unknown objects in the workspace.
3.3. Data Fusion

Figure 3.14: Off-board camera detection sensor fusion steps.
Step 1: Fuse the data from Cam A with Active Objects.
Step 2: Fuse the data from Cam B with Active Objects.
Step 3: Fuse the remaining data from both cams and initialize new active objects.
Step 4: Create new active objects from the remaining points in cameras.
Chapter 4

Implementation

This chapter describes implementation details of the system. First, the hardware used is briefly introduced, followed by the list of software frameworks and packages used, with more detailed description of each of the modules (nodes) at the end of the chapter.

4.1 Hardware

4.1.1 Hot Metal Carrier - HMC

The HMC, a fully autonomous forklift-type truck, designed to carry molten metal in aluminum smelters and automated at Commonwealth Scientific and Industrial Research Organisation (CSIRO), served as a localized vehicle for this project. With over 100 hours of autonomous operations conducted and no incidents reported, it is a very reliable platform for our project development.

![Figure 4.1: Hot Metal Carrier laser rangefinder coverage.](image)

The precise localization is provided by four laser rangefinders positioned around the vehicle with 360 degree coverage of up to 30 meters, with just a few close
proximity blind spots, as indicated in Figure 4.1. Based on long duration operational experiments, the average lateral deviation of the localizer was approximately 10 cm, which is enough precision to consider it as a ground truth of the HMC position [11].

4.1.2 Off-board Cameras

Two off-board cameras, placed on the buildings, overlooking the workspace, were used in the project. Models used were AXIS Q6034-E PTZ Dome Network Camera providing high resolution $1280 \times 720$ color images with up to 18x optical zoom. Integration was done using Axis camera drivers developed and open sourced by Clearpath Robotics.

4.1.3 On-board Cameras

The total of four on-board cameras were used for the project on the HMC. Side cameras:

- Basler scA1400-17gm model with Kowa LM5JC1M, 2/3", 5mm, F2.8 lenses

Front cameras:

- Basler scA780-54fc with different lenses:
  - Far proximity camera: Kowa TV Lens F1.4 3.5mm II
  - Close proximity camera: Fujinon FishEye 1:1.4/1.8mm FE185C057HA-1

4.1.4 Computers

In total, three different computers were used throughout the project, however, only up to two at one time instance. Below, hardware specifications are given for all the machines.

**HMC on-board computer**

- CPU: Intel Core 2 Duo T7500 2.20 GHz
- RAM: 2 GB
- SSD Hard drive

**Laptop (off-board camera processing)**

- CPU: Intel Core 2 Duo T9500 2.60 GHz
- RAM: 4 GB

**Lab desktop (GPU processing)**

- CPU: Intel Core 2 Duo E8400 3.00 GHz
- RAM: 6 GB
- GPU: nVidia GeForce GT 640, 128-bit, 2 GB
4.2 Software and Frameworks Used

The following software packages and frameworks were used for efficient algorithm implementations and to make the system more stable.

- **Ubuntu 10.04 and 11.10**

- **C++ and Python programming languages**

- **Robot Operating System - ROS [7]**

  An open source software framework for robot software development providing an operating system-like functionality. It has implementations of commonly used functionality for robotic applications with all algorithms split into small packages called nodes which intercommunicate using a messaging system. Nodes can be reused and all dependencies are defined making the deployment easy. This system also allows easy scalability as well as processing distribution over multiple machines.

- **OpenCV [8]**

  Open Source Computer Vision Library is a free-to-use library aimed at real-time computer vision applications. It contains highly optimized versions of the most of the commonly used computer vision algorithms with GPU implementation of some. Being free-to-use and cross platform compatible it gained popularity among researchers and companies in recent years.

- **cvBlob [10]**

  cvBlob is a library for computer vision to detect connected regions in binary digital images. cvBlob performs connected component analysis (also known as labeling) and features extraction.

- **Hungarian Algorithm [9]**

  The algorithm is described in Section 3.3.3.

- **Dynamic Data eXchange (DDX) [6]**

  Robotics software framework for building distributed robot controllers consisting reusable packages of commonly used algorithms. It provides a low-overhead, but safe, mechanism to share data between processes. Different processes can run on the same or different computers as required for load balancing or reconfiguration to cope with hardware failure.

4.3 ROS Nodes and Software Structure

The whole system is split into a number of ROS nodes, each performing a specific task, with real-time communication between them. Benefits of this configuration includes easier scalability, splitting tasks between multiple machines without any additional modifications and making adjustments to small parts of the algorithm without affecting the rest of the system, resulting in less complicated debugging. Internode communication is based on ROS messages over a standard network using a method similar to web services defined by URIs [7].

Figure 4.2 represents the node structure of the system where red arrows are inputs, blue nodes show on-board processing, yellow nodes - off-board processing and a green node performs the final position fusion and generates output.
4.3.1 On-board Processing

transmitHMCpos

HMC localization and autonomous navigation algorithm is based on DDX and all available information including HMC position and observed beacons coordinates are only given as DDX messages. This node provides continuous real-time conversion of previously mentioned DDX messages to ROS messages with instantaneous publishing.

Figure 4.3: ‘transmitHMCpos’ node republishes HMC pose from DDX format to ROS format.

Inputs:

- HMC position in DDX format
- Beacons positions in DDX format

Outputs:

- /hmcPos (geometry_msgs/Pose2D)
- /beaconsPos (geometry_msgs/PoseArray)
camera1394

Images received from firewire on-board cameras are converted and published as ROS image messages.

![Diagram](camera1394.png)

Figure 4.4: ‘camera1394’ node undistorts images received from firewire cameras and publishes them to ROS image messages.

**Inputs:**
- Front on-board firewire camera images

**Outputs:**
- /baslerCloseProx/camera/image_rect (sensor_msgs/Image)
- /baslerFarProx/camera/image_rect (sensor_msgs/Image)

analyseROI

The area of interest, indicated by off-board processing (fuseCams node), is analyzed in detail to identify people. The following workflow is used:

1. Received position of a person in the camera frame is converted to the position in image pixels using the inverse of the ground homography matrix.
2. Person’s height is estimated as a function of his distance from the camera.
3. Region of interest (ROI) is extracted.
4. ROI is resized to closely match the standard training sample size of 128 px height.
5. ROI is analyzed using HOG detector to detect people.
6. Coordinates of all people detected in ROI, if any, are converted to the camera’s coordinate frame using the ground homography matrix and published as ROS messages.

![Diagram](analyseROI.png)

Figure 4.5: ‘analyseROI’ extracts areas of interest indicated by fuseCams and analyzes them using HOG people detector.

**Inputs:**
- inputCameraTopic (sensor_msgs/Image)
- peoplePositionForInputCameraTopic (customMsgs/DetPeopleArray)

**Outputs:**
- accordingCamTopicImgOutput (sensor_msgs/Image)
- accordingCamTopicPeoplePosOutput_ROI (customMsgs/DetPeopleArray)
analyseWhole

The whole on-board camera image is continuously analyzed in parallel to the targeted analysis (analyseROI). It is used as a backup solution in case a person was not detected in the off-board camera image, however, this method is less accurate compared to the targeted analysis. The following workflow is used:

1. The on-board camera image is split into two or three areas.
2. A larger region is included in the first image (close proximity), consequently, it is scaled down to match the height of the detector training sample - 128 px, improving the detection accuracy.
3. The second region is scaled up to match the height of the detector training sample.
4. Both regions are analyzed in parallel using HOG detector.
5. Positions of successful detections of people are converted to the camera’s coordinate frame using an inverse of the ground homography matrix and published as ROS messages.

Figure 4.6: ‘analyseWhole’ node runs in parallel to analyzeROI and analyzes the whole image using HOG detector as a backup solution in case a person was not detected in off-board camera images and correct position indicated to the on-board computer.

Inputs:

inputCameraTopic (sensor_msgs/Image)

Outputs:

accordingCamTopicImgOutput (sensor_msgs/Image)
accordingCamTopicPeoplePosOutput_Whole (customMsgs/DetPeopleArray)

4.3.2 Off-board Processing

axis_camera

Images received from off-board IP cameras are converted and published as ROS image messages. Additionally, an ability to control camera position based on pan, tilt and zoom could be included.

Figure 4.7: ‘axis_camera’ node converts images received from off-board IP cameras to ROS image messages
Inputs:
IP axis camera images

Outputs:
/axis_camera1/compressed (sensor_msgs/CompressedImage)
/axis_camera2/compressed (sensor_msgs/CompressedImage)

HMCsim

Background segmentation is performed on the off-board camera images in order to detect moving objects in the environment. OpenCV implementation of MOG2 background segmentation is used together with blob detection and filtering using cvBlob library. After blobs are detected and filtered, the confidence is calculated for each, based on the blob’s width and height ratio followed by the calculation of the bottom centre point of the bounding box representing feet position of the person, in image pixels, and published to the on-board computer. Processing flow of the node is defined below:

1. Background segmentation based on MOG2 algorithm, presented in Section 3.1.2.
2. Filtering: median blur, morphological closing, dilation
3. Blob detection
4. Blob filtering according to their size
5. Blob confidence calculation
6. Blob filtering by the confidence
7. Bottom centre point calculation
8. Publishing people positions and foreground images

Figure 4.8: ‘HMCsim’ node performs background segmentation and identifies people-like objects based on their dimensions.

Inputs:
/axis_camera1/compressed (sensor_msgs/CompressedImage)
/axis_camera2/compressed (sensor_msgs/CompressedImage)

Outputs:
/offBoardPeoplePos (customMsgs/OffBoardCamPeople)
/VblockFGImg (sensor_msgs/Image)
/UblockFGImg (sensor_msgs/Image)
**fuseCams**

Position fusion and short-term tracking of moving objects, provided by the HMCsim node, is performed by the node. The fusion is based on the position change over time and the same object likelihood is calculated between existing active objects and newly received points. Fusion and tracking is performed using an Extended Kalman Filter. Each measurement error is estimated by determining the homography accuracy depending on the object's distance from the camera. If a newly received point is not assigned to any active object, a new active object is created using the data from both cameras images. A short time window is used to discard active points with lost tracking based on the timestamp of the last update. When available, information from both off-board camera images is used for tracking, however one source is also sufficient, but the accuracy is likely to be reduced. Eventually, positions of currently tracked active objects are transmitted to the on-board computer.

Step by step process description:

1. Received points are converted to the world coordinate system using the ground homography.
2. Inactive points, if they have not been updated in a specified threshold time, are removed.
3. Overlapping active points are combined.
4. Positions of active points are updated using newly received object coordinates, which are then removed from the newly received points vector.
5. New active points are initialized for the remaining received object coordinates.
6. Kalman filter prediction is performed for all the active points.
7. Top down visualizer is updated:
   (a) Kalman filter trajectories
   (b) Latest active objects’ positions
   (c) Newly received points’ positions
   (d) Latest HMC position
8. All areas observed by on-board cameras are recalculated based on the latest HMC position.
9. Inlier points, observed by the camera, are found for each of the on-board camera and objects’ coordinates are converted to the camera coordinate system.
10. Positions of inlier points are published to the specific camera topic.

![Figure 4.9: 'fuseCams' node maps objects onto workspace, performs off-board camera position data fusion and object tracking and indicates newly calculated positions to a vehicle with on-board cameras for further analysis.](image)
4.4 ROS Messages

ROS messages are used to carry the information between ROS nodes as well as computers over the network. In order to use the multi-computer configuration, the main machine has to be defined as Master, running the 'roscore' process, and others connect to it. Three ROS parameters need to be set for a successful communication between the machines:

- export ROS_IP:={IP of the current machine}
- export ROS_HOSTNAME:={IP of the current machine}
- export ROS_MASTER_URI:=http://{IP of the master machine}:11311

4.4.1 Default Messages

This subsection lists contents of used ROS messages, in interface definition language (IDL) [7], that come as standard with full desktop ROS Fuerte installation.

**geometry_msgs/Point**
- float64 x
- float64 y
- float64 z

**geometry_msgs/Pose2D**
- float64 x
- float64 y
- float64 theta

**geometry_msgs/PoseArray**
- Header header
- Pose[] poses

**sensor_msgs/Image**
- Header header
- uint32 height
- uint32 width
- string encoding
- uint8 is_bigendian
- uint32 step
- uint8[] data
sensor_msgs/CompressedImage

Header header
string format
uint8[] data

4.4.2 Custom Messages

Some information required custom messages to be created in order to carry the necessary information without a significant overhead.

DetPeopleArray

std_msgs/Header header
float64[] conf
geometry_msgs/Point[] pos

OffBoardCamPeople

std_msgs/Header header
float64[] confVblock
float64[] confUblock
geometry_msgs/Point[] posVblock
geometry_msgs/Point[] posUblock
Chapter 5

Experiments and Results

In this chapter the results of multiple experiments are presented. First, the testing environment and conditions are discussed. Then, separate parts of the algorithm are analyzed as stand alone modules and finally the full-system test is presented.

5.1 Testing Environment

Our testing environment was an industrial area, approximately $30 \times 30$ meters, with the plan shown in Figure 5.1. Two out of three off-board cameras overlooking the area were used in our tests, with views shown in Figure 5.2.

![Figure 5.1: Our testing environment: 30 m by 30 m yard with three building cameras overlooking it. However, only cameras A and B could be used for the experiment.](image)

5.2 Datasets

Despite the ability to run the algorithm in real-time on-board the HMC, for the parameter tuning as well as evaluation purposes, all the necessary sensor data and
Figure 5.2: Off-board camera views overlooking the testing environment. Camera A is placed on the V-block building and Camera B is placed on the U-block building.
Camera images were recorded into a rosbag creating a number of datasets. For the processing, real-time playback of the data was performed. Table 5.1 provides the information about all used datasets.

Datasets with all on-board cameras recorded have lower frame rates because the simultaneous recording of high resolution images from all four cameras caused buffer overflow and frames were constantly dropped. The difference was varying from 1-5 FPS for all four cameras to 15-20 FPS for the side cameras only.

## 5.3 Parameters

### 5.3.1 Field of Views of On-board Cameras

Because of different cameras and lenses used (as described in Section 4.1.3), FOVs of the HMC on-board cameras are not the same:

- Front far proximity camera: 1.438 rad = 82.4 degrees
- Front close proximity camera: 2.687 rad = 154 degrees
- Left and right cameras: 1.443 rad = 82.7 degrees

### 5.3.2 Homography Accuracy Estimation

The accuracy of the ground homography conversion is not uniform for the whole workspace. The error increases with the increasing distance of the point from the camera. For the sensor fusion to perform accurately, a good error model, a second order polynomial equation with two constants, is needed. Constants were calculated using a graph fit method on a number of points selected using step by step increments of the distance from the camera within the workspace limits.

Camera A (V-block) homography error estimation, shown in Figure 5.3, resulted in the following equation:

\[ y = x^2 \times 0.000240 + x \times 0.015092 + 0.007745 \]  

(5.1)
Chapter 5. Experiments and Results

5.3.3 Person Height Estimation

For localized detection, an estimation of person height depending on his distance from the camera is needed. Actual heights from a number of samples of a person standing at different distances were manually analyzed and results plotted. Considering fixed resolutions for cameras and in order to avoid any additional conversions...
during the process, image pixel values were used directly to define the distance and person height. Figures 5.5 and 5.6 show the linear relationship between the distance $d_p$ and height of the person $h_p$ for the front and side cameras accordingly.

An equation for the person height estimation in the front far proximity camera image:

$$h_{p,\text{front}} = d_{p,\text{front}} \times 1.5856 - 327.8592 \quad (5.3)$$

An equation for the person height estimation in the side cameras:

$$h_{p,\text{sides}} = d_{p,\text{sides}} \times 1.2068 - 312.6361 \quad (5.4)$$

Figure 5.5: Person height estimation according to his distance from the camera, in pixels, for the front far proximity camera.
Figure 5.7: Background segmentation process. The area of a moving object, likely to be a person in our case, is extracted and then a bounding box is placed around it, middle point of which represents where feet are in contact with the ground.

Figure 5.6: Person height estimation according to his distance from the camera, in pixels, for the left and right cameras.

5.4 Off-board Detection

The detection of objects in a workspace based on the background segmentation using off-board camera images is crucial for the rest of the system. After the moving objects are extracted from the input image, a bounding box is placed around them with the middle bottom point being considered as a contact point between feet and ground. The whole process is shown in Figure 5.7.

Datasets 1 and 2 were obtained from each of the two static cameras and analyzed in detail to determine the performance. Foreground image post-processing ensured that unexpected noise is fully filtered out, with only 25 frames in total containing false positive detections, corresponding to 0.368%. Analysis was based on three criteria: number of frames where the object was missed, number of false positive detections and when people are not detected due to an obstacle.

People blocked by obstacles, in majority of cases by the moving HMC itself, was the main problem. By looking at both camera images separately, in the dataset where HMC was static, a blob of a person was not detected in 72 frames, or 3.34%, and blocked by an obstacle in 58 frames, or 2.7%. However, when both, V-block and U-block, camera images were combined people were detected correctly 100% of the time.

In the dataset 2 where HMC was driving around the workspace and analyzing off-
5.5 On-board Detection

The on-board computer analyzes the whole image while placing particular emphasis on the areas with the high likelihood to contain a person, as indicated by the off-board cameras. Both processes are based on the HOG algorithm, however they contain substantially different aspects, so the results are discussed separately.

5.5.1 Localized Analysis

Targeted analysis uses the data provided from off-board cameras to locate, crop and resize the areas containing human-size moving objects in the environment; in the majority of cases it is a person. Given similar conditions and normalized target size, it can be used to evaluate the performance of HOG algorithm under different poses and distances of people as

<table>
<thead>
<tr>
<th></th>
<th>HMC Static Set</th>
<th>HMC Driving Set</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separate</td>
<td>93.96%</td>
<td>77%</td>
<td>82.38%</td>
</tr>
<tr>
<td>Combined</td>
<td>100%</td>
<td>98.64%</td>
<td>99.07%</td>
</tr>
</tbody>
</table>

Table 5.2: Detection rates of two datasets. Separate - both off-board cameras are analyzed separately. Combined - both camera viewpoints are used together resulting in significantly better detection rates.

board cameras separately, one or more people were behind an obstacle in 1066 frames, or 23%. However, when both cameras were combined, a person was blocked only in 63 frames, or 1.36%, showing a significant improvement. A typical missed detection is when blobs of a person and the HMC merge instead of the person being fully occluded by some obstacle, as seen in Figure 5.8. Detection rates (positive detections) are summarized in Table 5.2.

The average error of 0.15 meters was observed in the detection of the outline of the person in good conditions. When combined with the conversion using the ground homography, the average error increased to 0.27 meters. Difficult conditions like hard shadows proved to have a dramatic effect on the accuracy with the error increasing to 1-1.5 meter when the shadow is falling directly towards the camera.

Figure 5.8: Blobs of the person and HMC merge causing failed detection of the person. However, having more than one camera from a different viewpoint averts this problem in the majority of cases.
Figure 5.9: Person is seen by the on-board camera from the a) front, b) back and c) side, giving different scores by the HOG detector.

well as the cameras, because the image quality and resolution of side cameras is significantly better than the front cameras. Dataset 1 with HMC parked was used to avoid any additional disturbances like blur caused when the vehicle is turning and HOG detection score recorded as an accuracy measure.

**Different People Poses**

The first test was to record the detection score under three poses of a person: when a person is seen from the front, back or the side; example pictures are shown in Figure 5.9. As can be observed from the Table 5.3, the highest score was achieved with the person facing the camera with the average score being 8.567, followed by the back pose with the average score 6.205 and the most difficult detection was when a person is seen from the side with the average score of 4.811.

<table>
<thead>
<tr>
<th></th>
<th>Front</th>
<th>Back</th>
<th>Side</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average</strong></td>
<td>8.567</td>
<td>6.205</td>
<td>4.811</td>
</tr>
<tr>
<td><strong>Std. Deviation</strong></td>
<td>4.863</td>
<td>3.35</td>
<td>2.104</td>
</tr>
<tr>
<td><strong>Min Score</strong></td>
<td>2.03</td>
<td>2.03</td>
<td>2.02</td>
</tr>
<tr>
<td><strong>Max Score</strong></td>
<td>29.63</td>
<td>19.21</td>
<td>11.69</td>
</tr>
</tbody>
</table>

Table 5.3: HOG detection score variance depending on the pose the detected person is seen by the camera.

**Different Leg Placement**

Despite the largest deviation in HOG scores occurring for the front pose, a significant effect of successful and unsuccessful detection of the same person at the same distance was noticed in the side pose depending on the position of legs while walking or running. Four cases were separated: legs close, legs spread - wide, legs spread - medium and one leg lifted, visualized in Figure 5.10. During this evaluation, in the cases when detection failed, as score of 0 was given.

Table 5.4 summarizes the classifier performance where cases with legs spread wide and medium clearly stand out with very poor performance, 23.21% and 56.86% accordingly. The phenomenon can be explained by the concept of HOG where the model of a person is learned with defined margins for deformations of the model, mainly limbs. However, while walking fast, the legs are spread beyond the margin,
5.5. On-board Detection

Figure 5.10: Four different cases of leg placement while the person is walking. a) Legs close b) Legs spread wide c) Legs medium spread d) One leg is lifted resulting in an incorrect matching and failed detection. This issue is not likely to occur when the camera is facing the front or the back of the person.

<table>
<thead>
<tr>
<th></th>
<th>Close</th>
<th>Wide</th>
<th>Medium Spread</th>
<th>One Lifted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>2.998</td>
<td>0.3883</td>
<td>2.351</td>
<td>4.04</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>1.477</td>
<td>1.031</td>
<td>2.305</td>
<td>1.42</td>
</tr>
<tr>
<td>Min Score</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max Score</td>
<td>5.81</td>
<td>3.63</td>
<td>6.27</td>
<td>7.2</td>
</tr>
<tr>
<td>Successful Detections</td>
<td>85.19%</td>
<td>23.21%</td>
<td>56.86%</td>
<td>95.83%</td>
</tr>
</tbody>
</table>

Table 5.4: HOG detection score variance depending on the leg placement when a person is seen from the side.

Different People

Three people of different heights and clothing, seen in Figure 5.11, were participating in the majority of the experiments. Dataset 1 was used to evaluate if there were any accuracy differences depending on the person and clothing it is detecting. From Table 5.5 it is clear that with grayscale images as an input, the high visibility jacket does not increase the detection accuracy, and person C wearing a gray shirt had the highest detection rate. In general there are very insignificant variations between the average scores of people wearing different clothing.

<table>
<thead>
<tr>
<th></th>
<th>Person A</th>
<th>Person B</th>
<th>Person C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>4.995</td>
<td>5.327</td>
<td>5.836</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>2.21</td>
<td>3.01</td>
<td>3.75</td>
</tr>
<tr>
<td>Min Score</td>
<td>2.1</td>
<td>2.15</td>
<td>2.03</td>
</tr>
<tr>
<td>Max Score</td>
<td>11.79</td>
<td>14.57</td>
<td>18.34</td>
</tr>
</tbody>
</table>

Table 5.5: HOG detection score variance depending on the person.

Varying Distance From The Camera

The final classifier evaluation is done according to the distance from the camera to where the person is standing. Three distance categories were defined: up to 5...
meters, 5 to 7 meters and more than 7 meters with occurrences distributed 39%, 30% and 31% accordingly in our datasets. Despite the search window being normalized and resized, the difference in resolutions is still present, with people located close to the camera will be imaged in high resolution, while the ones standing far away will have to be upscaled, resulting in a lower resolution image and lost details. However, accuracy differences are minor at all distances proving the effect of the adaptive algorithm, details are shown in Table 5.6.

<table>
<thead>
<tr>
<th></th>
<th>Up to 5 m</th>
<th>5 m to 7 m</th>
<th>More than 7 m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>6.138</td>
<td>5.244</td>
<td>6.374</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>3.29</td>
<td>2.23</td>
<td>4.01</td>
</tr>
<tr>
<td>Min Score</td>
<td>2.01</td>
<td>2.04</td>
<td>2.04</td>
</tr>
<tr>
<td>Max Score</td>
<td>16.90</td>
<td>12.04</td>
<td>23.41</td>
</tr>
</tbody>
</table>

Table 5.6: HOG detection score variance depending distance from the camera the person is located.

5.5.2 The Full Image Analysis

For the continuous analysis of the whole image, a comparison was made between running the detector on the whole image and splitting it into areas depending on the distance from the camera as previously seen in Figure 3.12. The processing time was very similar between these two approaches. It should be noted that the area split method of each of the extracted areas was done in series during this test. Making use of the parallel processing would be beneficial. The detection rate was approximately the same for people in the close proximity of up to 6 meters away from the camera. However, with the increasing distance, the area split method showed significantly better results. Figure 5.12 shows an example of the two approaches trying to detect a person standing 11 meters away from the camera. The full image analysis method is only a backup solution in cases when off-board detection fails or is inaccurate causing a failure in the localized detection. Using the area split method, the accuracy is over 96%, however the false positive rate, even after filtering by the detection confidence, is high at 47.3%. Rates were calculated
5.5. On-board Detection

Figure 5.12: Person located 11 meters away from the camera. A - HOG is performed on the whole image, but a person far away not detected. B - The area split method - indicated area is cropped, normalized and HOG performed. The person successfully detected.

Figure 5.13: Distortion caused by the fisheye lens. When the person is on the side of the image, the distortion is very significant compared to the example where a person is located in the centre of the image.

by analyzing detections and re-detections in every single frame explaining such a significant amount of false positives in the full image analysis. The main objects identified as people were posts, edges of barrels and corners of scaffolds. 

An important role was the full image detection in using a close proximity camera image taken with a wide angle fisheye lens. It covered the area 30 cm to 3 meters in front of the HMC and could initiate an emergency stop if a person is identified in that area while the vehicle is moving forward. Almost no other objects are encountered in such a close proximity meaning that the rate of false positives was below 5%. The correct detection rate, true positives, were 72% with 96% when the person is located in the centre of the image. A drop in true positives when the person is seen on the side of the image is caused by the visual distortion of the fisheye lens, this effect is visualized in Figure 5.13. Distortion correction techniques described in Section 3.2.1 were applied, however, the result was still unsatisfactory and it significantly reduced the view angle. One possible solution would be to train the HOG classifier with sample pictures of people taken with the fisheye lens and use that model only for the close proximity camera.
Figure 5.14: Three people being tracked in the workspace. Blue circles indicate the current fused position of people, red and green circles are detections in off-board cameras A and B accordingly. Red and green triangles are field of views of on-board cameras of the HMC. Small black dot is the current position of the HMC.

5.6 Data Fusion

EKF is used to perform the fusion of the people position data from both off-board cameras. Addition of the estimated position from the on-board camera upon the detection of a person using the HOG classifier was tested, however, because of significantly lower accuracy it was excluded from the final version. Instead, just confidence levels were added to objects to verify the ones that are classified as people. Additionally, the lifetime threshold of these objects are increased by 50% compared to ‘unknown’ moving objects in the environment making their tracking of higher importance.

Short term tracking was sufficient giving smaller margins for the position change in the tracker for the accurate position indication to on-board cameras. The average length of the track was 25.71 seconds with some cases of a person being tracked all throughout the experiment. With the establishment of new tracks once the previous one is lost happening within 2-3 frames, it was proven to be more accurate to drop an old track and create a new one when a significant position change occurs in the off-board detection. Tracks of three people in the workspace are shown in Figure 5.14.

For evaluation purposes, the true position of the person was identified by hand in the input images and then converted to the world coordinate frame using the homography. It cannot be considered as a fully accurate ground truth, however with known homography error this indication is accurate enough to evaluate the performance of the fusion algorithm. Datasets 1, 2 and 3 were used resulting in the average error of 0.127 meters after the homography conversion in good conditions.
With hard shadows facing towards the camera and elongating the object in the foreground image, the average error rate increased to 0.75 meters. The impact of different weather conditions to the accuracy is explained further in Section 5.9.

5.7 The Full System: Joint Detection

The full algorithm was tested running two separate machines for off-board and on-board camera processing with real-time communication over the network. All data exchange was kept to a minimum, and only the pose of the vehicle and positions of detected points in the vicinity of the areas observed by the vehicle cameras were transmitted.

The experiments were carried out with varying weather conditions, the absence of shadows during cloudy days provided the best detection while sunny and rainy days resulted in hard shadows and reflections in puddles. Tests were not run during the rain, but right after, because some of on-board electronics were not waterproof.

The analysis was done by considering all the frames when a person is within the field of view of at least one of the HMC’s on-board cameras and full body can be seen in the on-board image. Results are shown in Table 5.7 with the overall detection rate exceeding 94%.

<table>
<thead>
<tr>
<th></th>
<th>Overall Accuracy</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full System</td>
<td>94.47%</td>
<td>5.53%</td>
<td>4.13%</td>
</tr>
</tbody>
</table>

Table 5.7: Detection rates of the full system where FN are false negatives and FP are false positives.

From the perspective of the on-board camera, the average position correction of the detected person based on the data from on-board and off-board cameras was 0.18 meters in the $x$ (left to right) axis and 1.2 meters in the $z$ (object’s distance from the camera) axis. It is important to note that shadows reduced the accuracy of off-board cameras based on the background segmentation.

Figure 5.15 shows intermediate results from the algorithm and could be directly referred to the steps in the diagram shown in the Figure 3.1.

5.8 Processing Speed

Configuring the algorithm to work in real-time was a significant challenge. The most processor intensive method was HOG detector to identify people in on-board camera images. Two computers were used simultaneously with the algorithm split into off-board and on-board processing. By utilizing GPU for the full image analysis, the system was working in real time using two on board cameras: two side cameras or both front cameras were tested. The frame rate of 4-7 FPS was achieved depending on the search window size (related to how close a person is to the camera) and a number of people detected in the on-board camera image. However, using all four on-board cameras was not possible in real-time, but adding a second on-board computer with a powerful GPU and two cameras directly connected to it would allow the real-time processing of the system in full configuration.

Utilizing GPU for HOG detector had a significant increase in the processing speed as well as freeing up CPU for other actions like data fusion. A speedup of up to 30 times was observed during tests of full image analysis. The effect was not so significant with the localized analysis where speedup was 5-7 times because of the smaller search window and pre-processing, like cropping. However, the accuracy
Figure 5.15: Intermediate results of the detection process. A - an object position conversion from the off-board camera frame to the world coordinate frame, B - identification of points within on-board camera observed area, C - conversion to the on-board camera image and people detection using HOG. The green rectangle defines the search area, the red rectangle indicates the area where a person is found.

of CPU implementation of HOG algorithm was better compared to the GPU implementation, meaning we could not use GPU version for localized analysis where having the highest possible accuracy was crucial.

5.9 Practical Issues

The majority of failed detections and inaccuracies were caused by unfavorable weather conditions which affect the image quality or give a false indication of an object size or position to a part of the algorithm, the background segmentation part. Two common issues were hard shadows in sunny conditions, mainly from the morning to noon. When a moving object is extracted, the shadow is considered to be a part of it, resulting in a larger bounding box and the middle-bottom point is located at the wrong position. After the conversion to the world coordinate frame using the ground homography, a false offset up to 1 meter can occur depending on the position of the sun, as seen in Figure 3.11. When the sun is located behind the off-board camera or straight above, the disturbance is minimal, however if the camera is facing the sun, shadows have a significant effect with the possibility of lens flare as well. Figure 5.16 shows a similar issue is caused by reflections in puddles after the rain. Limited visibility during the rain or storm was not tested because of some of the on-board electronics were not waterproof.

Glares can affect not only off-board cameras, but also on-board ones. When the camera is facing the sun at a particular angle, a bright vertical line can appear in the picture, caused by CCD smear, failing the detection, like in Figure 5.17. Another hard case is when a person is running fast and close to the HMC. Off-board image processing and tracking is able to keep up and indicate correct positions, however, due to wide leg placement while running, HOG detector often fails.
Figure 5.16: Reflection in a puddle after the rain causes incorrect segmentation of a moving person.

Figure 5.17: Bright vertical line appeared in the picture of an on-board camera failing the detection of a person. It was caused by a CCD smear when facing the sun.
Chapter 6

Summary and Contributions

We have proposed a system which performs joint detection of pedestrians in real-time, by combining the information from off-board static cameras with the detection from cameras on-board localized moving vehicles. Experiments have shown that the 94% detection rate can be achieved considering a variety of weather conditions proving a potential for real-life applications.

The main advantage of our approach is the usage of a combination of monocular cameras as sensors on-board of the vehicle and off-board on static structures overlooking the workspace. Such cameras are likely both to be already present. Furthermore, the price of high resolution digital cameras is continuously decreasing allowing approaches like ours to be incorporated in cheaper systems.

The implementation is based on a number of intercommunicating modules, each performing its own task. This has the advantage of easy deployment and expansion of the system. Two main procedures are needed when the system is installed in the new location: calibration of cameras to match the new world coordinate system with the new ground homography estimation for each camera and encoding the new camera placement on the vehicle in order to estimate the field of view. A separate software module is processing each camera input. If an extra sensor has been added, it’s enough to launch an additional instance and add extra input to the fusion algorithm instead of modifying the whole system. Furthermore, most of the processing is performed simultaneously which allows an easy parallelization of the processing by making the use of new multi-core designs and the increasing popularity of GPUs.

The localized moving platform in our application was a large truck operating in the industrial environment. Despite the automation and high demand of such safety systems from industrial environments, it can be also applied to smaller scale like home robots, warehouse robots [18], or even self-driving cars [19], for example to guide cars in busy locations, like pedestrian crossings, intersections or roundabouts.

6.1 Future Work

With the achieved performance and potential applications, the method should be made more robust and capable of running in real-time with the full system configuration.

In order to increase the processing speed to allowing the analysis four (or more) on-board cameras, the first step would be to further optimize the algorithm and test three computers working simultaneously by splitting the on-board cameras’ HOG processing between two machines. Furthermore, more parts of off-board processing could be adapted to run on GPU as well as another implementation of GPU
implementation of the HOG algorithm could be found which does not reduce the detection accuracy.
The operating area, workspace, can be easily expanded provided the same coordinate system is being used for the localization of the vehicle or a moving platform and sufficient off-board camera coverage is present. Off-board processing could be set up to analyze area only within a defined range from the position of the vehicle in order to make the system more efficient.
Even though the detection accuracy is over 94%, the majority of failures occurred in specific conditions, which could be identified. Shadows and reflections in wet conditions caused the misalignment of the bounding boxes during the off-board detection. By comparing the mirror effect and knowing or estimating the position of the sun, these object elongations could be discarded. Glares, mainly caused when the camera is facing the sun, affected both on-board and off-board detection. This effect is difficult to avoid and in most cases no data can be recovered, however, identifying the glare affected area would be a helpful way to avoid false positives.
With the changing lighting conditions and the vehicle getting closer and further from objects, cameras were automatically adjusting the aperture and focus. These adjustments had a dramatic effect on the foreground image with the whole model failing for few frames until the background model adapted to changes. The algorithm could instantaneously re-establish the background model upon the identification of the aperture or focus adjustment.
The HOG detector has a tolerance to a body deformations, however within certain limits, failure occasionally occurred in cases when a person is seen from the side walking fast and legs are widely spread. A possible workaround would be to train the classifier with the higher number of such samples or reduce the penalty for body transformations. However, the latter might lead to the higher number of false positive detections.
The possible expansion of the project is to implement a short term trajectory planning, both for the localized vehicle and people in the vicinity based on previous and current movement velocity and direction. By estimating the positions of objects in the workspace, any collisions could be identified and the warning signal sent to the vehicle to take the appropriate action to avoid it.
Further increase in the processing speed and easier system deployment could be made by creating an embedded system running the algorithm. Currently, not every system can support carrying one or two laptops and by designing hardware to run a specific method it can be made more efficient.

6.2 Acknowledgements

The author would like to thank Paulo Borges for the project supervision, support during the experiments and contributions throughout the project and on the final thesis. Furthermore, David Haddon and Fred Pauling for the technical support as well as Ash Tews, Vaidotas Miseikis, and Edmundas Mišeikis for valuable suggestions on the conference paper and the final report. The author also thanks the CSIRO Minerals Down Under Flagship for the financial support.
Appendix A

Camera Homography Data

Ground homography matrix had to be calculated for each of the off-board and on-board cameras. The process included manually identifying distinct points in the images, getting their coordinates both in pixels and in physical units of the world coordinate system.

A.1 On-board Cameras

In order to obtain physical units of known positions, HMC was parked next to the grid of known dimensions drawn on the ground.

A.1.1 Front Far Proximity Camera

Figure A.1: Marked point coordinates in pixels to estimate the ground homography for the front far proximity camera.
### Appendix A. Camera Homography Data

#### Point ID | Real World (meters) | Image (640 × 480)
--- | --- | ---
1. | (-1.45; 3.5) | 215 × 317
2. | (1.65; 3.5) | 505 × 322
3. | (1.65; 1.5) | 620 × 403
4. | (-1.45; 1.5) | 111 × 393

Table A.1: Point correspondences between the image and the camera coordinate system for the front far proximity camera.

**Resulting Homography Matrix:**

\[
H_{FarProx} = \begin{bmatrix}
-5.27175e^{-03} & -1.95212e^{-04} & 1.90746e^{+00} \\
-1.47156e^{-04} & 5.46264e^{-03} & -3.41902e^{+00} \\
6.67475e^{-05} & -4.74919e^{-03} & 1
\end{bmatrix}
\]

#### A.1.2 Left Camera

#### Point ID | Real World (meters) | Image (696 × 520)
--- | --- | ---
1. | (-3; 6.62) | 199 × 341
2. | (3; 6.62) | 545 × 341
3. | (3; 3.62) | 692 × 413
4. | (-3; 3.62) | 62 × 408

Table A.2: Point correspondences between the image and the camera coordinate system for the left proximity camera.

**Resulting Homography Matrix:**

\[
H_{LeftCam} = \begin{bmatrix}
-5.64862e^{-03} & -2.36919e^{-04} & 2.16276e^{+00} \\
-2.46387e^{-04} & 1.41815e^{-04} & -2.11308e^{+00} \\
-3.72186e^{-05} & -3.84719e^{-03} & 1
\end{bmatrix}
\]
A.1.3 Right Camera

![Image of marked point coordinates in pixels to estimate the ground homography for the right camera.]

Table A.3: Point correspondences between the image and the camera coordinate system for the right camera.

<table>
<thead>
<tr>
<th>Point ID</th>
<th>Real World (meters)</th>
<th>Image (696 x 520)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(-3; 7.07)</td>
<td>162 x 337</td>
</tr>
<tr>
<td>2.</td>
<td>(3; 7.07)</td>
<td>488 x 341</td>
</tr>
<tr>
<td>3.</td>
<td>(3; 4.07)</td>
<td>595 x 397</td>
</tr>
<tr>
<td>4.</td>
<td>(-3; 4.07)</td>
<td>30 x 398</td>
</tr>
</tbody>
</table>

Resulting Homography Matrix:

\[
H_{\text{RightCam}} = \begin{bmatrix}
-5.95117e^{-03} & 2.28642e^{-05} & 1.96434e^{+00} \\
5.46330e^{-04} & 2.03984e^{-04} & -2.53266e^{+00} \\
1.27019e^{-04} & -4.02540e^{-03} & 1
\end{bmatrix}
\]
A.2 Off-board Cameras

A.2.1 V-block Camera

Figure A.4: Four points identified in our workspace with precise image pixel positions in red, and real world coordinates (in meters) in green for the V-block camera ground homography calculation.

Camera position was set to:

- pan = -167.5737
- tilt = -25.2183
- zoom = 1

<table>
<thead>
<tr>
<th>Point ID</th>
<th>Real World (meters)</th>
<th>Image (800 x 450)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(9.84; 21.22)</td>
<td>348 x 107</td>
</tr>
<tr>
<td>2.</td>
<td>(25.1; 22.22)</td>
<td>675 x 183</td>
</tr>
<tr>
<td>3.</td>
<td>(26.06; 6.05)</td>
<td>491 x 422</td>
</tr>
<tr>
<td>4.</td>
<td>(11.19; 4.98)</td>
<td>52 x 245</td>
</tr>
</tbody>
</table>

Table A.4: Point correspondences between image and real world coordinates for the V-block camera.

Resulting Homography Matrix:

\[
H_{V\text{block}} = \begin{bmatrix}
5.15155e^{-02} & 1.89505e^{-01} & -2.34862e^{+01} \\
4.11261e^{-02} & -5.91601e^{-02} & 2.37582e^{+01} \\
-2.07467e^{-04} & 5.30802e^{-03} & 1
\end{bmatrix}
\]
A.2.2 U-block Camera

Camera position was set to:
pan=17.6033
Tilt=-14.9557
Zoom=1

![Image showing four points identified in workspace with precise image pixel positions and real-world coordinates for the U-block camera ground homography calculation.]

Table A.5: Point correspondences between image and real-world coordinates for the U-block camera.

<table>
<thead>
<tr>
<th>Point ID</th>
<th>Real World (meters)</th>
<th>Image (800 × 450)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(9.84; 21.22)</td>
<td>231 × 177</td>
</tr>
<tr>
<td>2</td>
<td>(25.1; 22.22)</td>
<td>471 × 135</td>
</tr>
<tr>
<td>3</td>
<td>(26.06; 6.05)</td>
<td>747 × 184</td>
</tr>
<tr>
<td>4</td>
<td>(11.19; 4.98)</td>
<td>524 × 267</td>
</tr>
</tbody>
</table>

Resulting Homography Matrix:

\[
\begin{bmatrix}
8.82355 e^{-01} & -8.77956 e^{-01} & 2.11638 e^{+02} \\
-7.45678 e^{-01} & -1.63017 e^{+00} & 1.02162 e^{+03} \\
-4.34677 e^{-04} & 1.4423 e^{-01} & 1
\end{bmatrix}
\]

Figure A.5: Four points identified in our workspace with precise image pixel positions in red, and real-world coordinates (in meters) in green for the U-block camera ground homography calculation.
Bibliography


camera_calibration.html


