Hes·so VALAIS WALLIS
School of Engineering

Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts Western Switzerland

Degree Programme

Systems Engineering

Major Infotronics

BACHELOR'S THESIS

DIPLOMA 2021

Steve Anchise

Client identification in ski resort

- *Professor*
  Prof. Silvan Zahno
- *Expert*
  Remo Schnyder
- *Submission date of the report*
  20.08.2021

Données du travail de diplôme
Daten der Diplomarbeit

Filière / Studiengang **SYND**	Année académique **/** *Studienjahr* **2020/21**	No TD / *Nr. DA* **IT/2021/42**
Mandant **/** *Auftraggeber* ☐ HES—SO Valais ☒ Industrie ☐ Etablissement partenaire *Partnerinstitution*	Etudiant **/** *Student* **Steve Anchise** Professeur / *Dozent* **Silvan Zahno**	Lieu d'exécution / *Ausführungsort* ☒ HES—SO Valais ☒ Industrie ☐ Etablissement partenaire *Partnerinstitution*
Travail confidentiel / *vertrauliche Arbeit* ☐ oui / ja [1] ☒ non / nein	Expert / *Experte* *(données complètes)* **Remo Schnyder** Syrto AG, Alustrasse 81, 3940 Steg	

Titre / *Titel*

Identification des clients dans le domaine skiable syBaB

Description / *Beschreibung*

La zone d'entrée de la télécabine de Hohsaas est surveillée par une caméra Vivotek. Le nombre d'invités dans cette zone doit être suivi pour le contrôle du flux d'invités et en raison du concept d'exploitation approuvé. L'objectif des travaux est d'augmenter la précision du système. La solution existante sera remplacée ou étendue. Les données vidéo de la caméra installée sont disponibles ainsi qu'une caméra Vivotek pour les tests en laboratoire.

Caractéristiques : Capteur caméra, Capteur Lidar, Programmation Python ou C++.

Le diplômant devra :

- Comprendre la configuration actuelle avec les caméras Vivotek installées
- Étudier les méthodes et systèmes de détection supplémentaires ou alternatifs pour améliorer la détection
- Etablir le principe de mesure, en déterminer le concept et les interfaces
- Etablir son budget de développement
- Mettre en place l'environnement de test de la solution sélectionnée
- Mettre en œuvre l'algorithme de détection de personnes personnalisé ou existant
- Tester la solution à la station Hohsaas (facultatif).

Objectifs / Ziele :

- Amélioration de la solution existante de mesure de flux des invités dans la zone d'entrée du télécabine

Signature ou visa / *Unterschrift oder Visum*	Délais / *Termine*
Responsable de l'orientation / filière *Leiter der Vertiefungsrichtung / Studiengang:* .. . [1] Etudiant / *Student* : ……..	Attribution du thème / *Ausgabe des Auftrags:* **10.05.2021** Présentation intermédiaire / *Zwischenpräsentation* **07 – 08.06.2021** Remise du rapport / *Abgabe des Schlussberichts:* **20.08.2021, 12:00** Exposition / *Ausstellung der Diplomarbeiten:* **25 – 27.08.2021** (si autorisé / *falls genehmigt*) Défense orale / *Mündliche Verfechtung:* **30.08 – 09.09.2021**

[1] *Par sa signature, l'étudiant-e s'engage à respecter strictement la directive DI.1.2.02.07 liée au travail de diplôme.*
Durch seine Unterschrift verpflichtet sich der/die Student/in, sich an die Richtlinie DI.1.2.02.07 der Diplomarbeit zu halten.

Rapport reçu le / *Schlussbericht erhalten am* ……..…….. Visa du secrétariat / *Visum des Sekretariats* ……..

Client identification in ski resort

Graduate Steve Anchise

Objectives

The number of clients in the entry zone to the gondola in Hohsaas must be tracked. Currently it is regulated autonomously, but the system is not accurate enough. The aim of the work is to increase the precision of the system.

Methods | Experiences | Results

This paper compares various turnkey tracking solutions to find the best, which will then be implemented. In the four solutions studied, one uses 2D video, two stereo video and one 3D LiDAR technology. A test setup emulating the real situation was used to produce comparable datasets. The best existing turnkey solution was the one already installed, which is not precise. A custom solution was developed from scratch and turnkey solutions were put aside. It uses data from a LiDAR sensor with a custom tracking algorithm and ROS as an execution environment. It is written in C++ and uses PCL and OpenCV libraries. A launch file is used to set the parameters. A box cropping is used to remove incomplete data. It is followed by a projection in the top view, required to then apply an opening. Finally, a blob detector is used to extract the blobs to then be tracked with a centroid tracking algorithm. Adjustable edges are used to resolve merge and split scenarios. This custom solution has proven to be better performing than the current solution, achieving a perfect F1 score of 4 in all the lab tests. On-site tests produced an average F1 score of 3.46 over eight extracts.

Bachelor's Thesis
| 2021 |

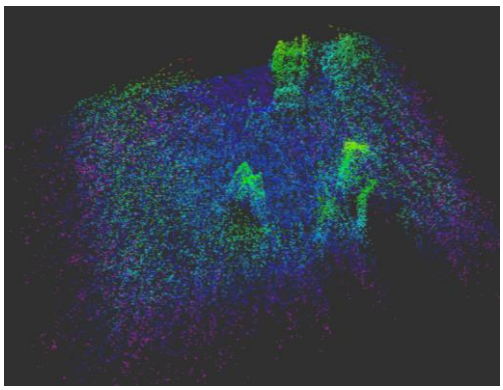
Degree programme
Systems Engineering

Field of application
Infotronics

Supervising professor
Silvan Zahno
silvan.zahno@hevs.ch

Partner
Syrto AG

The test setup in Steg. The zone of interest in green with the board holding the sensor array on top.



Point cloud as it can be seen by the LiDAR. The brighter the color the higher the point.

Information about this report

Contact information

Author: Steve Anchise
Bachelor Student
HES-SO//Valais Wallis
Switzerland
Email: steve.anchise@students.hevs.ch

Declaration of honor

I, undersigned, Steve Anchise , hereby declare that the work submitted is the result of a personal work. I certify that I have not resorted to plagiarism or other forms of fraud. All sources of information used and the author quotes were clearly mentioned.

Place, date: August 19, 2021

Signature: 

Validation

Accepted by the HES-SO//Valais Wallis (Switzerland , Sion) on a proposal from:

Prof. Silvan Zahno , Thesis project advisor

Remo Schnyder , Syrto AG , Main expert

Place, date: _____

Prof. Silvan Zahno	Prof. Pierre Pompili
Advisor	Dean, HEI-VS//Systems Engineering

Acknowledgments

I would first thank the Smart Processes lab for their welcoming work attitude. Special thanks to my mentor Silvan Zahno and my assistant Gilles Mottiez. Thanks to Syrto AG and all its collaborators for their cooperation and interest. Final thoughts go to my family and friends for their kind support and patience and to all the other people I haven't mentioned yet who helped me in their own way.

Abstract

Monitoring the number of people inside a defined area can be very important in the context of people transportation for security purposes. It would be possible to post an employee for this role, but the costs would be too high for the task. This paper compares various turnkey tracking solutions to help finding the best, which will then be implemented. In the four solutions studied, one uses 2D video, two stereo video and one 3D LiDAR technology.

A test setup trying to emulate the real situation was used to produce comparable datasets. The best existing turnkey solution was the one already installed, which is not efficient enough for its use as it is the goal of this work to improve the system. A custom solution was developed from scratch and turnkey solutions were put aside. The custom solution uses data from a LiDAR sensor with a custom tracking algorithm and ROS as an execution environment.

A ROS node, the LiDAR driver, is used for data acquisition. The algorithm is a ROS node written in C++ and uses both PCL and OpenCV libraries. A launch file is used to set the parameters of the algorithm. A box cropping is used to remove incomplete data due to the FOV of the sensor. It is followed by a projection in the top view, required to then apply an opening using the OpenCV library. Finally an OpenCV blob detector is used to extract the blobs with a size filter to then be tracked with a centroid tracking algorithm. Adjustable edges are used to resolve merge and split scenarios.

This custom solution has proven to be better performing than the current solution, achieving a perfect F_1 score in all the lab tests. On-site tests produced very encouraging results, with an average F_1 score of 3.46 over eight extracts. This performance exceeds that of the current solution in lab tests.

Key words: real-time analysis, object tracking, people counting, sensor comparison, 3D LiDAR, ROS, PCL, OpenCV

Contents

List of Figures

List of Tables

1 Introduction

1.1 Context / Problem

The number of clients in the entry zone to the gondola in Hohsaas must be tracked for security purposes. Overcrowding of the zone could cause an unwanted fall in the gondola pit. It would be possible to post an employee for this role, but the costs would be too high for the task. Currently the passage through the rotary gate is regulated autonomously. A Vivotek camera using its proprietary embedded tracking software counts the number of people in the Zone of Interest (ZOI). A Programmable Logic Controller (PLC) commanding the rotary gate uses this number to decide whether or not to block the arrival of people.

Figure 1.1 *The current setup in Hohsaas.*
The camera is on top of the ZOI. An extra sensor is on this picture for testing purposes.

However the system is not accurate enough. For example, large backpacks or skis can be mistaken for people, or people may not be detected at all. These kinds of scenarios falsify the count and hinder the regulation. The aim of the work is to increase the precision of the system. The existing solution will then be replaced or extended.

1.2 Objectives

The objectives are to understand the current setup, to investigate additional and alternative sensor methods and systems for improving detection, to setup a test environment, to implement a custom or existing person detection algorithm and finally optionally to test the solution at the station Hohsaas itself.

1.3 Structure of this report

In the section Analysis, the state of the art possibilities for sensors and algorithms will be explained, as well as typical usage scenarios of such a system. The Design section presents the current solution and other solutions to be compared and tested. The tests results will decide what solution to then develop. The elected solution will be elaborated and tested in Implementation. Final performance comparisons against the current solution and on-site test will be carried in the Validation section.

2 | Analysis

In this chapter, state of the art possibilities for hardware and software will be explained and reviewed. The usual scenarios will be described too, as the background or use case can influence the results by a lot, and the studied scenario is not a standard one.

There are a lot of possibilities along the following axis : sensor technology, number of sensors, sensor(s) point of view, algorithm. These choices can then be narrowed down, as a given scenario often calls for a specific solution.

Contents

2.1 Sensor technology

2.1.1 Image Sensor

An image sensor or imager is a sensor that can convert the light waves to make a digital image. The waves could come from light or other radiation, it depends on the sensibility of the sensor, which is determined by the type of semiconductor used.

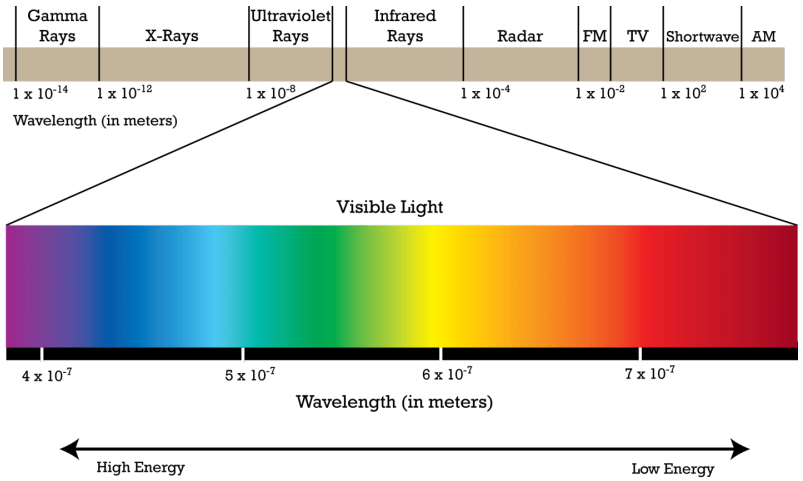


Figure 2.1 *The light spectrum.*
Source : https://www.vippng.com/preview/iiRRwTR_light-spectrum-image-electromagnetic-spectrum-chemistry/

As Cressler explains in [1], the two main types of electronic image sensors are the charge-coupled device (CCD) imagers and the complementary metal-oxide-semiconductor (CMOS) imagers. They are very closely related, but in general CMOS imagers offer higher pixel density, lower power dissipation, reduced system form factor, at the expense of image quality (particularly under low-light conditions), and system design flexibility. CMOS imagers are thus a natural fit for most consumer applications. CCDs, however, offer superior image quality and flexibility at the expense of system size. They remain the most suitable technology for high-end imaging applications, professional, industrial, scientifical and medical applications. Both of them play a significant role in this market.

To get a color image, the sensor has to separate the 3 different colors. The Bayer Mask is the cheapest and most common solution, where each square of four pixels has one filtered red, one filtered blue and two filtered green (the human eye is more sensitive to green). The resulted color resolution is then lower than the intensity resolution. Better color separation can be obtained with 3-CCD respectively 3-CMOS, where a prism splits the image into red green and blue components. Each of the three sensors is then optimized to respond to its respective color. This method reaches higher quantum efficiency, because most of the light entering the aperture is captured by a sensor, whereas a Bayer mask absorbs a high proportion (perhaps 2/3) of the light falling on each pixel. A last way would be with a Foveon x3 sensor, which uses an array of layered pixel sensors, separating light via the inherent wavelength-dependent absorption property of silicon, such that every location senses all three color channels. This method is similar to how color film for photography works. It is a good compromise between quality and price, but at the cost of a high noise in the image.

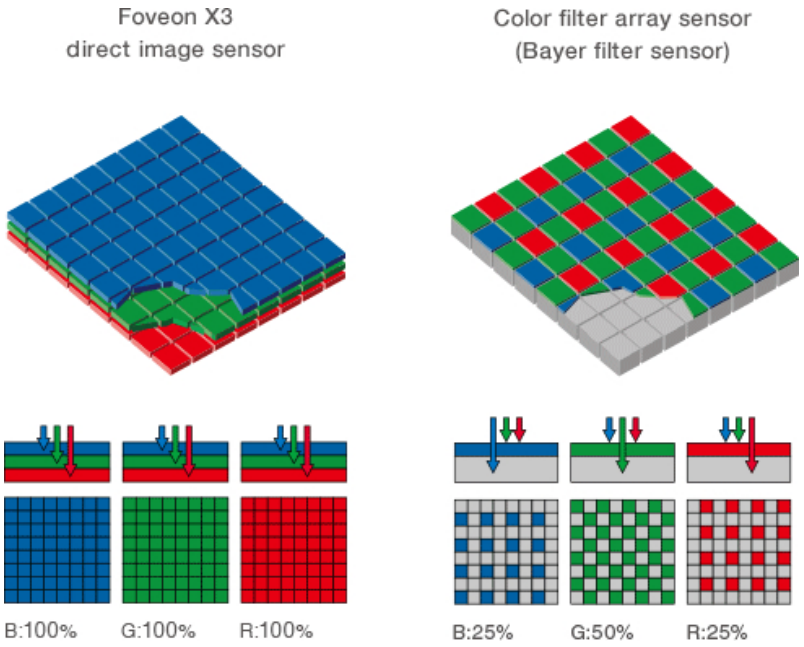


Figure 2.2 *Image sensor comparison : Foveon X3 and Bayer filter.*
Source :
https://nofilmschool.com/sites/default/files/styles/article_wide/public/1.jpg

2.1.2 Infrared

It's possible to use the same kind of technology as an image sensor to detect other wavelengths, such as infrared with infrared sensors. As explained in [2], infrared radiation, in principle, is not any different from the light that we see with our eye (visible radiation), except it has a different wavelength range. With visible light, it's possible to tell different wavelengths apart because they look like different colors to the human eyes. Red light is a longer wavelength than blue light with most other colors being somewhere in between as shown in Figure 2.1. Infrared radiation that we think of as heat has longer wavelengths than the light humans can see.

A major difference with optical cameras is that the focusing lenses cannot be made of optical glass, as glass blocks thermal infrared bands. Special expensive materials such as silicon crystal, germanium, zinc selenide or others must be used. The higher cost of these special lenses is one of the reasons why thermographic cameras are more expensive.

As thermal imaging can often detect that there is something warm behind nonsolid barriers and do not depend on ambient light, it finds many uses in military, security, firefighting, construction, industrial, spatial and medical applications.

Figure 2.3 *Comparing the same image in the visible and infrared spectrum. Looking at both images provides insights that a single image cannot.*
Source :
https://learningweather.psu.edu/sites/default/files/Lesson3/infrared-spitzer.jpg

2.1.3 LiDAR

Light Detection And Ranging (LiDAR) is a method for determining ranges (variable distance) by targeting an object with a laser and measuring the time for the reflected light to return to the receiver. LiDAR can also be used to make digital 3-D representations of areas on the earth's surface and ocean bottom, due to differences in laser return times, and by varying laser wavelengths. It has terrestrial, airborne, and mobile applications.

Time Of Flight (TOF) is one of the technologies used to measure a distance between objects. It does so by measuring the time that a sound wave, a ray, a micro wave or other wave travels to the object and returns. It has been widely used, for example with sonar, survey or motion capture. When a pulsed laser is emitted, there is phase difference between the emitted pulse and the returned pulse after reflecting at an object. The farther apart the objects are, the larger the phase difference. [3]

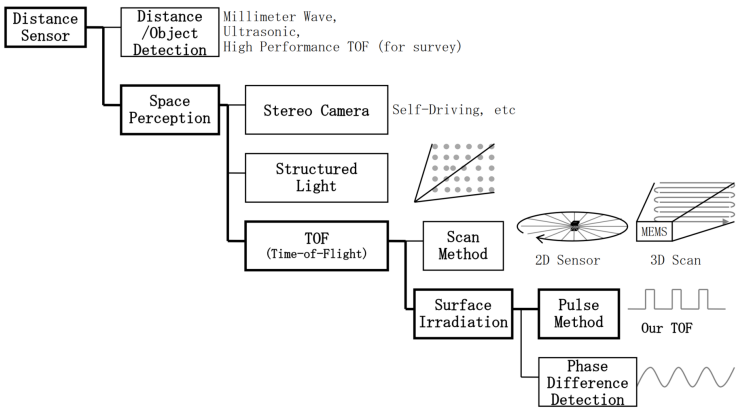


Figure 2.4 *Different distance sensors and technologies.*
Source : [3]

2.1.4 Others

There are a lot of other technologies and sensor types available, but those precited were the most used in similar state of the art works and seem the most adapted to the use case.

2.2 Number of sensors

Having a single sensor is the simplest way, as it doesn't require any calibration or background knowledge. The limit is the Field Of View (FOV) of the sensor, which can determine if a single sensor at a given distance can cover the ZOI. If it isn't enough and the ZOI must be fully covered, the situation could require more than a single sensor. Another limiting factor is the occlusion when a single sensor is not mounted directly on top of the scene. Objects can then be partially or totally hidden by other objects or scene elements, thus making reliable tracking harder.

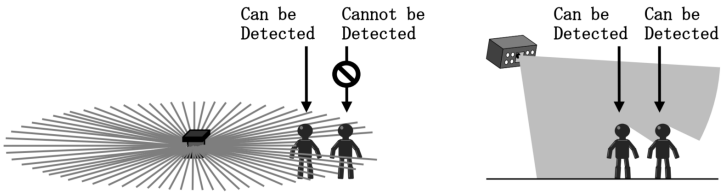


Figure 2.5 *An occlusion example.*
Source : [3]

Another factor would be the accuracy. Because a single sensor can only give so much information, particularly in special situations (poor lighting, extreme weather...), it can help to have a different sensor with its different set of advantages and drawbacks.

Good results are obtained in [4]–[9] with a single sensor.

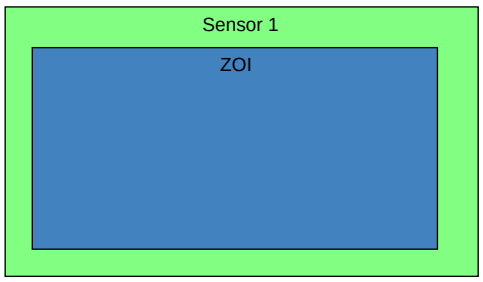


Figure 2.6 *ZOI surveilled by a single sensor.*

An array of sensors surveilling the same ZOI can be used to increase accuracy with redundancy and limit occlusion. It's possible to reconstruct 3D data if the sensors used provide 2D data only. [10] presents a tracking system in a scene using multiple overlapping cameras, limiting occlusion problems by using 3D scene reconstruction.

Figure 2.7 *ZOI surveilled by a stereo sensor.*

Having more than one sensor technology can help too, for example in a surveillance scenario to have a standard video sensor for the day and another infrared sensor for the night. It's possible to mix the informations to get better results, as different sensors can measure different values and redundancy increases the reliability and robustness. [11] uses both infrared and video imaging to count people which allows low cost, low-resolutions cameras to be used, as cross-referecing pixels with their color and temperature makes requiring less of them for equivalent accuracy. [12] uses both infrared and video imaging. [13] uses both LiDAR and video imaging.

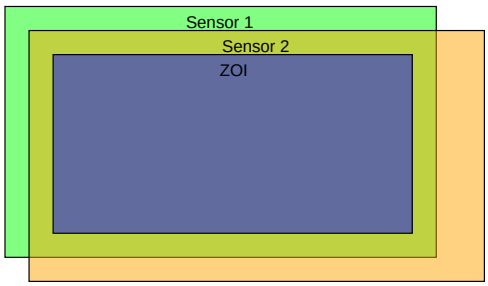


Figure 2.8 *ZOI surveilled by sensors with different technologies.*

As a single sensor has a limited area of use, it's possible to use multiple and to stitch the data together to have a bigger covered zone. Calibrating is in this case very important, more so when the scene has no particularities in the overlapping zone. [14] uses stitched depth cameras.

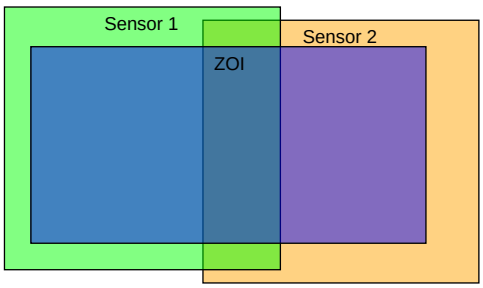


Figure 2.9 *ZOI surveilled by stitched sensors.*

Another way would be to surveil the entry, exit or key zones with a shared counter. [15] can track reliably across non-overlapping camera views.

2.3 Sensor point of view

Sensors can be mounted on top of the scene. This permits minimal overlapping of the passing objects but limits the FOV. It's also called ceiling mounted, bird's eye view or top down view. [8] use a direct downward view, and so do [11], [14].

Wall mounted sensors overlooking the scene allow for great FOV and easier installation, but making overlapping worse. It can use already installed security cameras and process that data for tracking. Occlusion problems can be helped by using multiple sensors from different angles, like in [10]. A non-direct downward is used in [7]. Similar point of view can be found in [4]–[6], [12], [15]–[18]

Sensors can also be mounted with a front view. It is often seen in the automotive car industry. This is the easier way to install sensors on moving vehicles. This presents massive overlapping problems like in Figure 2.5. [13] uses direct frontward view on a vehicle. [9], [15] use the same point of view.

2.4 Algorithm

It's difficult to know what algorithms firmware solutions use, as it can be "trade secret", so source code is rarely if at all available. So with little customizable settings, it's either the firmware solution can work well enough for the work or doesn't with very little in-between. No source was found mentioning the use of firmware, as developing the software was often the core of the work.

In the object tracking section, the same pattern can often be seen :

- Preprocessing
- Detection and Tracking
- Counting

Preprocessing can consist of reducing the data size to improve then processing speed afterwards, for example by filtering the data, changing the color to levels of gray or removing the background and the shadows.

Like previously mentioned, overlapping is a recurring problem depending on the point of view. It can be resolved with software too, by the use of a tracking algorithm which can assign an identity to an object, permitting correct re-identifying when the object is visible again. It's usually done with size, color, moving pattern or speed.

[8] uses a background subtraction algorithm and a filling algorithm as preprocessing, then two-level blob-tracking in a single topdown-view camera setup. [6] uses a background subtraction algorithm along an adaptive background model with automatic thresholding as well as shadow removal, then an automatic segmentation algorithm and finally

tracking with an adaptive Kalman filter. [7] uses Gaussian mixture model to remove the background, followed by shadow removal, then blob-tracking and multi-level reverse tracking for people counting.

3 Design

In this chapter, the current installation will be elaborated. The tested systems and the test installation will be presented. The different solutions will then be tested using a defined test procedure. The results will be used to compare the solutions and select the best one.

Contents

3.1 Current setup

In the current setup a SC8131 Vivotek stereo camera is installed and has a rule setup on its proprietary software to count the number of clients in the loading zone, which is in turn read through a WebSocket by a Raspberry PI. The Raspberry PI then transmits this number over a cable with I2C to a PLC which can block the rotary gate if the number of people exceeds the limit of 8.

A hidden button was installed and wired to the PLC in case of an employee wanting to go through in the reverse direction. It allows him to go through for 30 seconds.

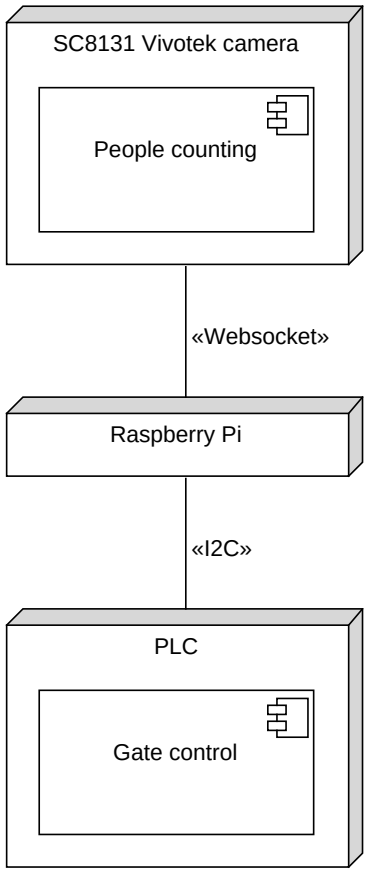


Figure 3.1 *Schema of the current setup.*

The chosen solution would replace the Vivotek camera.

3.2 Case study

The solutions to be tested are the following :

- SC8131 Vivotek stereo camera with its firmware, the currently installed one
- Hitachi 3D LiDAR with its sample firmware (available source code)
- S74 Mobotix camera with its firmware

- DVA3221 Synology Surveillance Station firmware with raw video from the Vivotek camera

These solutions were already available at Syrto AG and permitted immediate testing, that's why they were selected in this paper.

3.2.1 Test setup

A first test location was chosen in Sion, but it couldn't reproduce the real test setup well enough and fell outside the use case of the material, results were not valid and another location had to be found. A second location was found in Steg at Syrto AG directly, and the test platform is installed on a catwalk. A wooden board screwed to a pallet frame holds the sensors. The sensors are placed directly above the ZOI to better emulate the current location. The zone to control is an approximately 4x4 m^2 zone directly under the sensors.

Figure 3.2 The Zone of Interest (in green) with the wooden board holding the sensors on top.

The sensors are placed very close one to another to be tested at the same time in the same conditions.

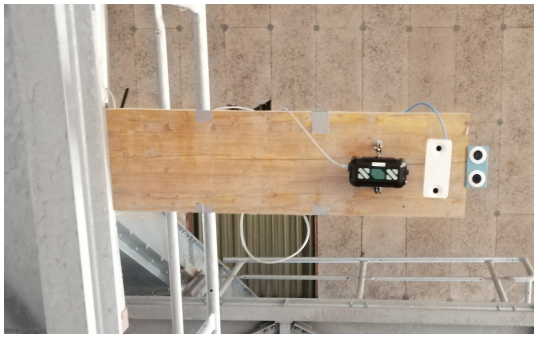


Figure 3.3 *Sensors placement on the test rig : Hitachi, Vivotek (feeding Synology too) and Mobotix (from left to right)*

3.2.2 Solutions

Vivotek stereo camera

The model is SC8131. Google Chrome on Windows was used to display the live, as Mozilla Firefox didn't allow modifications. Height is set to 400 cm and tilt angle to 0°. The ZOI is defined with adjustable points.

Figure 3.4 *Vivotek SC8131*
Source : https://www.vivotek.com/sc8131

Hitachi-LG Data Storage 3D LiDAR

The model is HLS-LFOM3. An Hitachi demo program on a Windows machine is used to configure and use the sensor. X and Y angle are set to 0°, Z angle to 90° to have the same disposition as the other solutions and height is set to 3800 mm. The ZOI is defined with an adjustable rectangle.

Figure 3.5 *Hitachi-LG Data Storage HLS-LFOM3*
Source : https://hlds.co.jp/product/details_en

Mobotix stereo camera

Model is S74, with two modules. Google Chrome on Windows was used to display the live. The Crowd/Overcrowd functionality from the AIRetail module was setup and used. The ZOI is defined with adjustable points.

Figure 3.6 *Mobotix S74*
Source : https://www.mobotix.com/en/products/mobotix-7/s74

Synology Surveillance station

Model is DVA3221. The web client is used for configuration. Google Chrome on Windows was used to display the live. The video feed comes from the Vivotek camera. The ZOI is defined with adjustable points.

Figure 3.7 *Synology DVA3221*
Source : https://www.synology.com/en-global/products/DVA3221

3.2.3 Recording setup

To record firmware solutions, OBS Studio on a Windows machine was used. 4 sources were setup on the same scene, to show the 4 solutions at the same time. The various windows were trimmed down to only show the ZOI, without unnecessary numbers if possible.

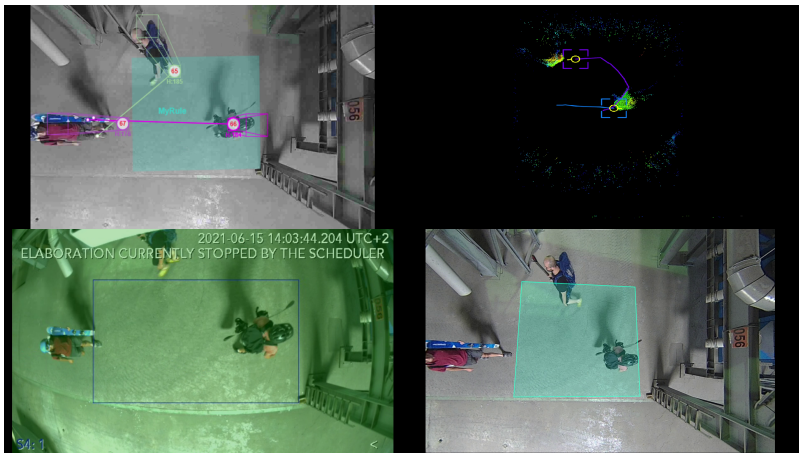


Figure 3.8 *Recording screen layout : Vivotek (topleft), Hitachi (topright), Mobotix (bottomleft) and Synology (bottomright)*

To record the raw video, the Synology Surveillance Station was used.

To record the raw LiDAR data, a custom Robot Operating System (ROS) program was used to read the data from the LiDAR and then publish it on a ROS topic, finally recorded to a ROS bag.

3.2.4 Test scenarios

3 test scenarios were used in the first test to decide. In the first scenario, the 3 actors, without accessories, enter the scene one after another then leave it without stopping. In the second scenario, the 3 actors, equipped with accessories composed of helmets, bags, skis and snowboards, enter the scene one after another then leave without stopping as in the first scenario. For the third and last scenario, the 3 equipped actors enter the scene one after another, stay in the ZOI as a group, then leave the scene one after another.

The first scenario is meant to establish a baseline. The second one tests the use of skiing and mountain accessories and the last one tests accessories in conjunction with overlapping objects.

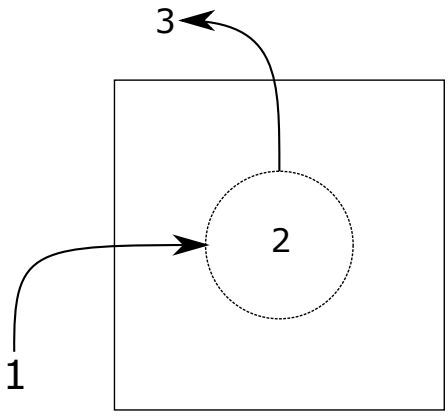


Figure 3.9 *Scenario layout*
1 : actors entering the ZOI
2 : waiting zone
3 : actors leaving the ZOI

3.2.5 Results

To quantify the results, precision and recall are used. They are calculated using true positive (TP), false positive (FP) and false negative (FN) rates with the following equations. TP means a detected object as it should be, true negative (TN) an undetected object as it should be, FP a detected object that should not be and FN an undetected object that should not be.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

As these are the only metrics used, TN rates were not calculated. Manually counted TP, TN and FP rates can be found in appendix A. The source video and the pictures used to produce these results can be found on the repository.

To then evaluate and take both precision and recall into account, the F-measure, also called balanced F-score or F_1 score, is used. It is the harmonic mean of precision and recall and can be calculated with the following formula.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

This score finally permits to rank the solutions.

	precision	recall	F_1
Vivotek	100%	100%	4.00
Hitachi	100%	84%	3.08
Mobotix	100%	68%	2.26
Synology	100%	59%	1.86

Table 3.1 Results test 1 scenario 1

In the first scenario, the Vivotek solution hits the perfect score. This is a good indication that the use made of the solution is correct, contrary to some preliminary tests which yielded very poor results. The Hitachi solution stands out from Mobotix and Synology solutions by getting a good score, while both Mobotix and Synology struggle with recall despite perfect accuracy and come lasts with a score close to 2.

	precision	recall	F_1
Vivotek	90%	96%	3.24
Hitachi	93%	87%	2.89
Mobotix	100%	81%	2.93
Synology	100%	58%	1.81

Table 3.2 Results test 1 scenario 2

In the second scenario, the average score is lower, as expected from the use of skiing accessories. The Vivotek solution dominates again with a score greater than 3, followed closely by Mobotix and Hitachi. Mobotix manages to increase its score. Synology is behind again with a score lower than 2.

	precision	recall	F_1
Vivotek	73%	100%	2.53
Hitachi	100%	79%	2.81
Mobotix	100%	76%	2.68
Synology	100%	65%	2.17

Table 3.3 Results test 1 scenario 3

In the third and final scenario, Vivotek finally falls below 3 points and is overtaken by Hitachi and Mobotix. Synology gets its best score but remains last. The average score is the lowest of all scenarios. We can see here the difficulties encountered by the Vivotek solution in reality, as this scenario attempted to reproduce real situations encountering errors.

	F_1 mean	rank
Vivotek	3.26	1
Hitachi	2.93	2
Mobotix	2.63	3
Synology	1.95	4

Table 3.4 Test 1 ranking

As maybe a mean of the score is not the best way to establish a ranking as not all situations are depicted in these scenarios, it is however a good indication. This test clearly puts the currently installed solution in front, but because the best solution is the already installed one, no solution can be selected from these.

3.3 Selected solution

A custom solution has to be chosen because no tested solution produced good enough results. The choice lies between the LiDAR and camera sensors as those are available along with the recorded data.

LiDAR seems the most robust solution, as it is less background and light dependent and the first test presents good results even though only a simple demo program is used. The sensor is however expensive, quoted at more than 1'000 CHF, but could later be replaced by a less expensive sensor in the range of 400 CHF. A standard surveillance camera is cheaper and can be found in the range of 200 CHF. Both solutions also require external processing but a simple computer in the range of 1'000 CHF should be enough.

The choice was therefore made for the most robust solution : the LiDAR sensor. The algorithm used will be introduced later.

3.4 Conclusion

Four available solutions were tested in an environment trying to emulate the real situation. The Vivotek stereo camera yielded the best results, but those are not enough for the application. As this solution is the one installed and the best performing one but not precise enough for the use case, a custom solution had to be chosen.

Custom algorithm on LiDAR data was selected, as it's the most robust choice.

4 Implementation

In this chapter the chosen solution, a custom algorithm with a LiDAR sensor, will be further analyzed, optimized and tested in the test environment.

The algorithm uses ROS as an execution environment. It is written in C++ and uses Point Cloud Library (PCL) and OpenCV libraries to function. A launch file is used to set the various parameters of the algorithm. The following operations are done on the data : box cropping, projection, opening, blob detection, tracking and split/merge handling.

A ROS bag can be used to replay the recorded data or alternatively a ROS node with its own launch file to play live data. The data played is used by the algorithm in the detector node.

Contents

4.1 Architecture

The solution uses ROS Melodic to function. The detector is setup as a C++ ROS node, launched with a launch file containing the important settable parameters. This permits changes of parameters' values without requiring a new compilation of the C++ code.

A ROS bag was used to play the recorded data which emulates the live data for testing. A ROS node can also be used to read the live data from the LiDAR. Both the ROS bag and the ROS node can publish the recorded or live data to the ROS topic "/cloud1". The detector node is subscribed to this topic. The node reading the live data was also used to make the ROS bag recordings.

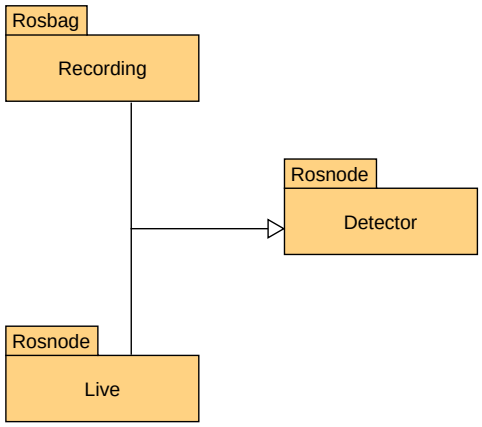


Figure 4.1 ROS nodes configuration.

The project is separated into three repositories, respectively the documents and report, the LiDAR driver and the detector. The driver for the LiDAR can be found on `https://gitlab.hevs.ch/SPL/bachelorthesis/2021-sybab/sybab-tof-driver`. The detector can be found on `https://gitlab.hevs.ch/SPL/bachelorthesis/2021-sybab/sybab-ros-pcl`. Use instructions can be found in the README files from the repositories.

4.2 Detector operations

Figure 4.2 shows how the data is processed. Only once at the beginning the detector node subscribes to the topic "/cloud1". The rest is currently set to run 10 times per second. This speed is defined by the publishing rate of the data player node. First the player node gets the data from the sensor, translates it, and publishes to the topic "/cloud1". Because the detector node is subscribed to the topic, it is notified and receives the data. The detector then processes the data.

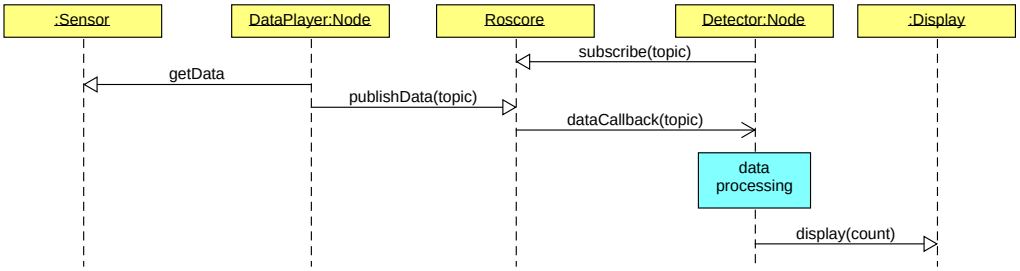


Figure 4.2 *Simplified sequence diagram.*
The data processing diagram can be found in the Figure 4.3.

The operations in Figure 4.3 are done on the data. The goal with each operation is to be as effective as possible with the least amount of data loss. The result is processed picture with the number of clients detected in the ZOI. The launch file contains the settable parameters for the operations.

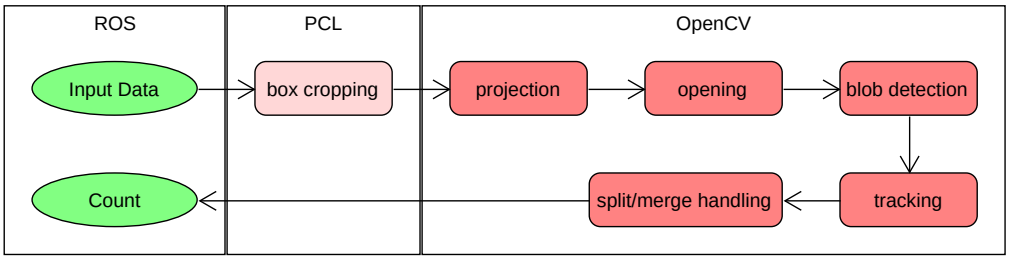


Figure 4.3 *Data processing diagram.*

Box Cropping

Filtering into a box makes sense because the data is in a conical shape, see Figure 4.4. Data on the edges cannot be high enough to be used and must be discarded. On the height axis, it filters out measurements from the ground and measurements that are too large from noise. These parameters are adjustable and can be changed if smaller or larger objects need to be detected or filtered.

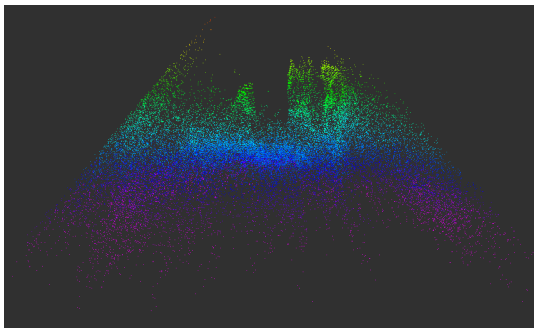


Figure 4.4 *Data comes in conical shape.*
It is a front view with the sensor on top. Ground measures and noise form the light blue and lower points.

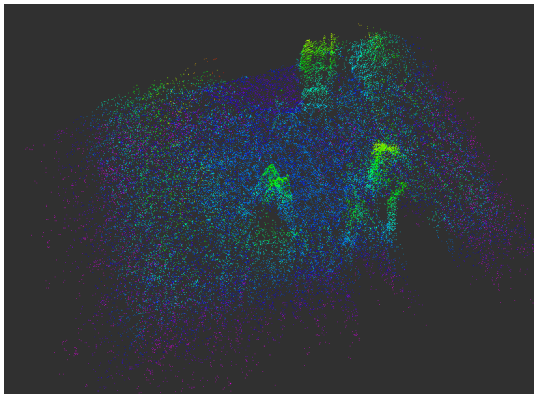


Figure 4.5 *Data from the LiDAR.*
It is a non direct downward view with the sensor on top. The brighter the color the higher the point. It's another view from the same data as in Figure 4.4.

Projection

To use OpenCV, it is necessary to have a planar image, so a projection on the xy plane is made. It is a top view and a binary mask (black or white).

Figure 4.6 *Data after box cropping and projection.*
It is a binary top down view processed from Figure 4.5.

Opening

Noise is always present in the image and objects are perforated. To remove the noise and fill the holes, an opening is used. The only parameter used is the diameter of the disk used for the morphological transformation, which is 15px. To do this transformation, the OpenCV library is used.

Figure 4.7 *Data after opening.*
Processed from Figure 4.6.

Blob detection

To find the blobs in the image, it is necessary to use a blob detector. It returns the centers of the blobs with their average size. It is then possible to draw a circle for each detected blob. The parameters are the minimum distance between the blobs, a boolean to choose if a filter by area is necessary and finally the minimum and maximum area for that size filter. The filter was set with very permissive values, to facilitate the tracking. The OpenCV library was used.

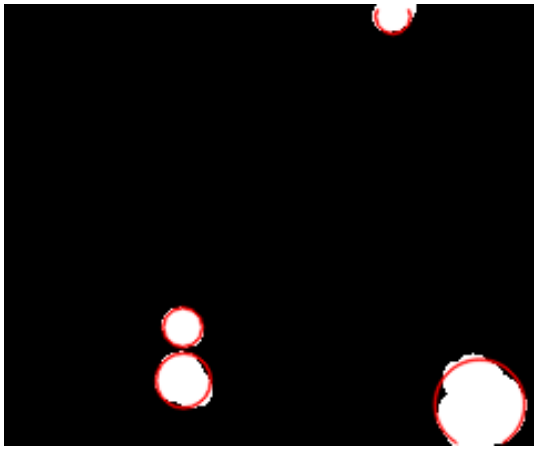


Figure 4.8 *Data after blob detection.*
Processed from Figure 4.7.

Tracking

The tracking uses a centroid tracking algorithm. New objects are given a new unique id number and can only appear or disappear on the configurable edges.

To achieve the tracking, it first calculates the square of the distance between all the objects centers from the previous frame (the old objects) and the objects detected in the current frame (the new objects) and puts them in a list. It then orders this list from smallest to largest. It then goes through this list, updating values of the old tracked object with the values from the new matched object and discards all the distances with

that old and new object, because they have already been matched. It does this until the list is empty. A filter limits the distance between two objects to avoid a mismatch and same with the size, as those cannot change too much in-between frames. The size filter must be high enough to not detect unwanted isolated objects as persons but low enough to detect children.

Finally it goes through the tracked objects to check if all of them have been matched or not. If not, it checks for merges, because two blobs in the previous frame can become one in the current frame if they touch. If still no match is found, it increases a counter to count the number of frames where the tracked object was not matched. When the counter reaches the configurable limit value, it deletes the tracked object. This permits objects to come back if they only briefly left the ZOI.

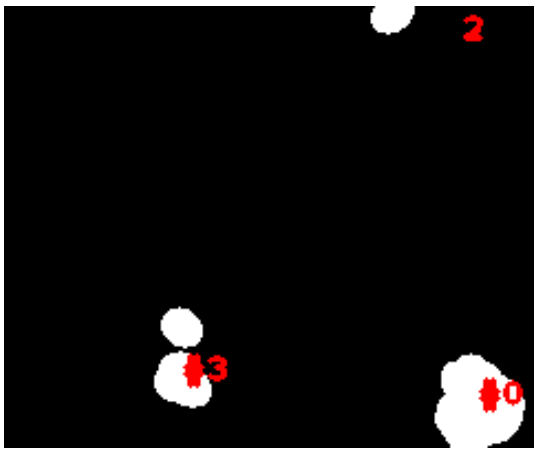
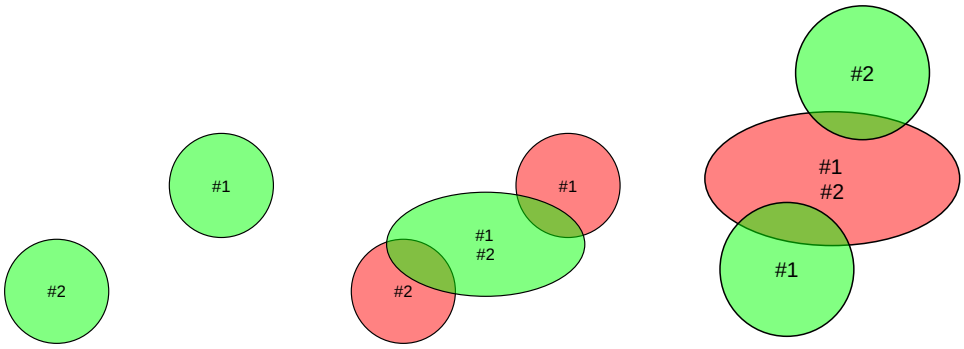


Figure 4.9 *Data after tracking.*
Processed from Figure 4.8. The number of people detected in the frame is displayed on the top right. The number on the detected persons is their attributed id number.

Split/merge handling

Some problems could appear when at least two blobs are touching, for example with two people standing very close. The blob detector finds only one blob, but there is two people forming it and so the algorithm should be able to know they are really two. The aforementioned configurable edges come back into use. People can only appear or disappear on the borders of the ZOI. They also appear separately thanks to the rotary gate. Then if a matched object suddenly disappears from the middle of the frame, it means a merge scenario happened. To resolve a merge scenario, the algorithm looks for the nearest new object and matches it with the old yet unmatched object. The blob now carries two objects at the same position. When they split back, the objects are then naturally matched in the first tracking operation. It can happen that the object ids are swapped in the process, because the algorithm finds two old objects with the same distance to a new object. It matches the object that appeared the first in the ZOI, but it is not necessarily the correct one.



(a) Two objects appear. *(b) The objects merge.* *(c) The objects split up.*

Figure 4.10 *Split and merge scenario.*
The green color stands for a new object and red for old. In the Figure 4.10b the objects are set on the same location. We can't be sure with this method if the identities are correct in Figure 4.10c. 1 could have continued going down or went back up.

5 Validation

In this chapter the selected solution will first be tested against the current solution with the same test, then tested alone more precisely. Finally it will be tested on-site at the station Hohsaas.

Contents

5.1 Lab results

Going back to the first test scenarios from the case study, the custom solution is compared to the installed solution, namely the Vivotek camera. The source code and the recorded lidar data, both on the server, can be used to recreate these results.

	precision	recall	F_1
Vivotek	100%	100%	4.00
Custom	100%	100%	4.00

Table 5.1 Comparing Vivotek and custom solution : results lab test 1 scenario 1

The custom solution matches the Vivotek camera with a perfect score in the first scenario.

	precision	recall	F_1
Vivotek	90%	96%	3.24
Custom	100%	100%	4.00

Table 5.2 Comparing Vivotek and custom solution : results lab test 1 scenario 2

In the second scenario, the custom solution surpasses the score achieved by Vivotek with a clean sweep.

	precision	recall	F_1
Vivotek	73%	100%	2.53
Custom	100%	100%	4.00

Table 5.3 Comparing Vivotek and custom solution : results lab test 1 scenario 3

In the third scenario the custom solution still gets a perfect score unlike Vivotek.

A second test was required to test merge and split situations. Two scenarios were used, in the first the objects enter the ZOI one after another then wait while in contact with each other, then leave one after another. In the second scenario, the objects enter the scene one after another, wait while in contact with each other and finally leave the zone together.

The second test was produced using only the LiDAR sensor, thus preventing comparison with the current solution. The LiDAR data can also be found on the server.

	precision	recall	F_1
Scenario 1	100%	100%	4.00
Scenario 2	100%	100%	4.00
Scenario 3	100%	100%	4.00

Table 5.4 Custom solution : results lab test 2

The second test was achieved with the perfect score too, but tracking identity was not always assured. It is however not required to ensure identity for this application with the added benefit of further ensuring anonymity.

5.2 On-site results

Since the laboratory tests were satisfactory and the planning permitted it, on-site tests could take place. For this, the LiDAR sensor was installed next to the Vivotek camera to cover an area of approximately 4.5m by 3.5m. The installation setup can be seen on the Figure 1.1. The Hitachi LiDAR is installed next to the Vivotek camera on top of the ZOI.

Since the station was open, passing customers served as test subjects, which also allowed for testing of more natural, unscripted scenarios. Three multi-minute sequences were recorded with the LiDAR to allow evaluation of the algorithm. Eight extracts were selected, i.e. all the moments when people were present in the ZOI.

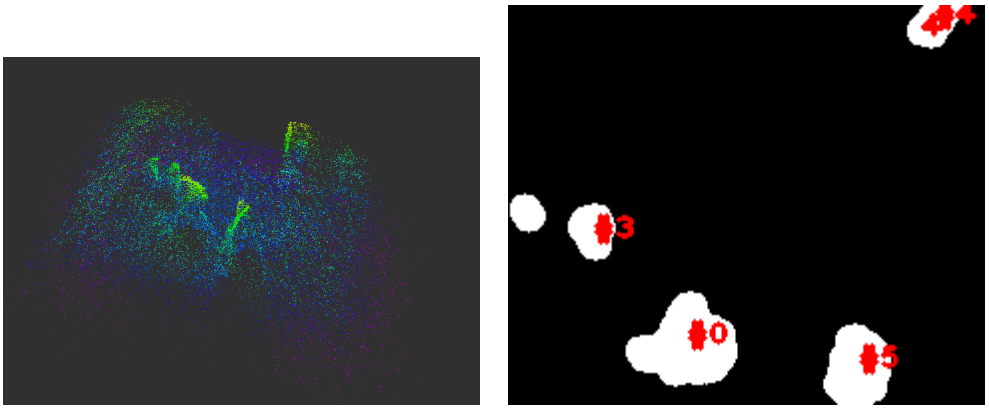
The following results were produced without changing either the parameters or the functioning of the algorithm, which could have further improved the results.

sequence number and timeframe used in [s]	precision	recall	F_1
sequence 1 : 190-290	100%	94.7%	3.69
sequence 1 : 330-370	90.7%	100%	3.46
sequence 2 : 0-70	100%	94.5%	3.67
sequence 3 : 120-210	98.4%	96.3%	3.69
sequence 3 : 210-260	92.3%	72.7%	2.22
sequence 3 : 325-375	96.4%	100%	3.78
sequence 3 : 375-435	97.1%	100%	3.83
sequence 3 : 500-545	100%	88.6%	3.34

Table 5.5 Results on-site tests

We obtain an average F1 score of 3.46, with average precision and recall of 96.9% and 93.3% respectively. These performances exceed those of all the solutions tested in the laboratory with the first test.

However, these scores could be further improved as recurring errors appear. Indeed, children under 1m are totally ignored which can lead to mistakes in the results. It is not necessarily problematic if the zone logic takes it into account, as children take less room and are under their parents responsibility. There is also some play in the settings to change the minimal detected height if wanted. Another problem is sometimes the partial detection of gondolas as objects and the presence of blobs at the edge of the zone creating problems with splits and merges. This could be resolved by having a FOV larger than the ZOI and then a definable closed polygon inside as ZOI to define where to count people exactly. This is a feature that was not implemented in this algorithm.



(a) Data creating the frame. *(b) Tracking view.*

Figure 5.1 *Most common errors with the tracking.*
A child (middle far left) is not detected because the blob is too small and the gondola (top right) is falsely detected as a person.

6 Conclusions

6.1 Project summary

To improve the currently installed system which was not performing well enough, four solutions including the current one were tested in a case study. The best one being the one already installed, a customized solution was developed. The customized solution obtained better results than the current one in the laboratory and very encouraging results in the field.

6.2 Comparison with the initial objectives

The currently installed configuration has been studied and understood. Additional or alternative detection methods and systems to improve detection have been studied. As solutions were already available from the company, the research part was somewhat reduced, but the measuring principle was established, as well as its concept and interfaces. A test environment was set up for all available solutions. Existing person detection algorithms and a customized one were used. The solution with the customized algorithm was tested at the Hohsaas station and a budget for the deployment was established.

Thus, according to the results of the laboratory tests, the objective of improving the existing guest flow measurement solution in the gondola entrance area is successful.

6.3 Encountered difficulties

The school in Sion seemed to be a good place to set up a test environment, but as the move brought its own uncertainties and difficulties, the decision was finally made to do the installation in Steg at Syrto AG. This decision turned out to be ideal, as very little hardware was needed to be operational and the test environment was very similar to the actual location, which was not so easy to establish in Sion.

Navigating through the various firmware was sometimes problematic. Each solution had its own nomenclature and algorithm activation procedure, which sometimes made things harder.

Since the test environment was set up in Steg and the station is in Saas-Grund, a lot of organization was necessary to avoid being stuck without some missing material or key element.

6.4 Future perspectives

Before the real deployment it would be wise to implement another adjustable counting zone. This would improve the reliability of the algorithm and make it easier to adapt to a different environment or sensor.

Further tests with splits and merge in the case of at least four objects could be done, to ensure correct matching and maybe improving it.

The system interface would benefit from a rework. The less systems used the better. Removing the Raspberry Pi from the system to add a Modbus link between the computer and the PLC could be a viable solution.

A separation in different interfaced blocks would allow the interchangeability of the modules, which could for example be separated into a control module, a data acquisition module, a data processing module and a display module. Changing a part would then only require the modification of the corresponding module without diving deep into the code like it is necessary now.

Adding native support for multiple overlapping sensors would make sense too, as the FOV is limited in some situations with a single sensor.

A | Tests results tables

	TP	FP	FN
Vivotek	33	0	0
Hitachi	31	0	6
Mobotix	25	0	12
Synology	27	0	19

Table A.1 Raw count test 1 scenario 1

	TP	FP	FN
Vivotek	27	3	1
Hitachi	26	2	4
Mobotix	17	0	4
Synology	23	0	17

Table A.2 Raw count test 1 scenario 2

	TP	FP	FN
Vivotek	38	14	0
Hitachi	37	0	10
Mobotix	32	0	10
Synology	36	0	19

Table A.3 Raw count test 1 scenario 3

Bibliography

[1] J. D. Cressler, "Let There Be Light: The Bright World of Photonics," in *Silicon Earth: Introduction to Microelectronics and Nanotechnology*, CRC Press, 2017, pp. 326–386.

[2] T. Scope, *About Thermal Imaging*, https://tinyurl.com/4dse4fth.

[3] I. Hitachi-LG Data Storage, *API Reference Manual*.

[4] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1820–1833, Sep. 2011. DOI: 10.1109/TPAMI.2010.232.

[5] Y. Cai, N. de Freitas, and J. J. Little, "Robust Visual Tracking for Multiple Targets," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3954, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 107–118. DOI: 10.1007/11744085_9.

[6] D. Lefloch, F. A. Cheikh, J. Y. Hardeberg, P. Gouton, and R. Picot-Clemente, "Real-time people counting system using a single video camera," in *Electronic Imaging 2008*, N. Kehtarnavaz and M. F. Carlsohn, Eds., San Jose, CA, Feb. 2008, p. 681 109. DOI: 10.1117/12.766499.

[7] J. L. Raheja, S. Kalita, P. J. Dutta, and S. Lovendra, "A robust real time people tracking and counting incorporating shadow detection and removal," *International Journal of Computer Applications*, vol. 46, no. 4, pp. 51–58, 2012.

[8] S. Velipasalar, Y.-l. Tian, and A. Hampapur, "Automatic Counting of Interacting People by using a Single Uncalibrated Camera," in *2006 IEEE International Conference on Multimedia and Expo*, Toronto, ON, Canada: IEEE, Jul. 2006, pp. 1265–1268. DOI: 10.1109/ICME.2006.262768.

[9] Z. Yan, T. Duckett, and N. Bellotto, "Online learning for 3D LiDAR-based human detection: Experimental analysis of point cloud clustering and classification methods," *Autonomous Robots*, vol. 44, no. 2, pp. 147–164, 2020.

[10] M. C. Liem and D. M. Gavrila, "Joint multi-person detection and tracking from overlapping cameras," *Computer Vision and Image Understanding*, vol. 128, pp. 36–50, Nov. 2014. DOI: 10.1016/j.cviu.2014.06.003.

[11] I. Amin, A. Taylor, F. Junejo, A. Al-Habaibeh, and R. Parkin, "Automated people-counting by using low-resolution infrared and visual cameras," *Measurement*, vol. 41, no. 6, pp. 589–599, Jul. 2008. DOI: 10.1016/j.measurement.2007.02.010.

[12] A. Torabi, G. Massé, and G.-A. Bilodeau, "An iterative integrated framework for thermal–visible image registration, sensor fusion, and people tracking for video surveillance applications," *Computer Vision and Image Understanding*, vol. 116, no. 2, pp. 210–221, 2012.

[13] S. Hwang, N. Kim, Y. Choi, S. Lee, and I. S. Kweon, "Fast multiple objects detection and tracking fusing color camera and 3D LIDAR for intelligent vehicles," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, IEEE, 2016, pp. 234–239.

[14] T.-E. Tseng, A.-S. Liu, P.-H. Hsiao, C.-M. Huang, and L.-C. Fu, "Real-time people detection and tracking for indoor surveillance using multiple top-view depth cameras," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 4077–4082.

[15] A. Wu, W.-S. Zheng, and J.-H. Lai, "Robust depth-based person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2588–2603, 2017.

[16] I. Benraya and N. Benblidia, "Comparison of Background Subtraction methods," in *2018 International Conference on Applied Smart Systems (ICASS)*, Medea, Algeria: IEEE, Nov. 2018, pp. 1–5. DOI: 10.1109/ICASS.2018.8652040.

[17] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding*, vol. 122, pp. 4–21, May 2014. DOI: 10.1016/j.cviu.2013.12.005.

[18] C. Yan-yan, C. Ning, Z. Yu-yang, W. Ke-han, and Z. Wei-wei, "Pedestrian detection and tracking for counting applications in metro station," *Discrete dynamics in nature and society*, vol. 2014, 2014.

Acronyms

CCD charge-coupled device. 4

CMOS complementary metal-oxide-semiconductor. 4

FN false negative. 17, 18, 35

FOV Field Of View. ix, 7, 9, 31, 34

FP false positive. 17, 18, 35

LiDAR Light Detection And Ranging. ix, 6, 8, 12, 14, 16, 19, 21, 22, 24, 30, 31

PCL Point Cloud Library. ix, 21

PLC Programmable Logic Controller. 1, 12, 34

ROS Robot Operating System. ix, 16, 21, 22

TN true negative. 17, 18

TOF Time Of Flight. 6

TP true positive. 17, 18, 35

ZOI Zone of Interest. 1, 7–9, 13–17, 23, 26, 30, 31

Glossary

opening An opening is a morphological transformation composed by a an erosion followed by a dilatation. ix, 21, 24, 25

precision Precision is the fraction of relevant instances among the retrieved instances. 17, 18, 30, 31

recall Recall, also called sensitivity, is the fraction of relevant instances that were retrieved. 17, 18, 30, 31