

# Institutionen för systemteknik

## Department of Electrical Engineering

**Examensarbete**

### **3D Position Estimation of a Person of Interest in Multiple Video Sequences: Person of Interest Recognition**

Examensarbete utfört i Datorseende  
vid Tekniska högskolan vid Linköpings universitet  
av

**Victor Johansson**

LiTH-ISY-EX--13/4718--SE

Linköping 2013



**Linköpings universitet**  
**TEKNISKA HÖGSKOLAN**



# 3D Position Estimation of a Person of Interest in Multiple Video Sequences: Person of Interest Recognition

Examensarbete utfört i Datorseende  
vid Tekniska högskolan i Linköping  
av

**Victor Johansson**

LITH-ISY-EX--13/4718--SE


Handledare: **Erik Ringaby**  
ISY, Linköpings universitet

**Peter Bergström**  
SKL

Examinator: **Per-Erik Forssén**  
ISY, Linköpings universitet

Linköping, 23 September, 2013



	<b>Avdelning, Institution</b> Division, Department  Computer Vision Laboratory Department of Electrical Engineering Linköpings universitet SE-581 83 Linköping, Sweden		<b>Datum</b> Date  2013-09-23
	<b>Språk</b> Language  <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English  <input type="checkbox"/> _____	<b>Rapporttyp</b> Report category  <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____	<b>ISBN</b> _____ <b>ISRN</b> LiTH-ISY-EX--13/4718--SE <b>Serietitel och serienummer ISSN</b> Title of series, numbering _____
<b>URL för elektronisk version</b> <a href="http://www.cvl.isy.liu.se/">http://www.cvl.isy.liu.se/</a> <a href="http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-97970">http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-97970</a>			
<b>Titel</b> 3D positions estimering av sökt person i multipla videosekvenser: Igenkänning av Title            sökt person  3D Position Estimation of a Person of Interest in Multiple Video Sequences: Person of Interest Recognition  <b>Författare</b> Victor Johansson Author			
<b>Sammanfattning</b> Abstract  <p>Because of the increase in the number of security cameras, there is more video footage available than a human could efficiently process. In combination with the fact that computers are getting more efficient, it is getting more and more interesting to solve the problem of detecting and recognizing people automatically.</p> <p>Therefore a method is proposed for estimating a 3D-path of a person of interest in multiple, non overlapping, monocular cameras. This project is a collaboration between two master theses. This thesis will focus on recognizing a person of interest from several possible candidates, as well as estimating the 3D-position of a person and providing a graphical user interface for the system. The recognition of the person of interest includes keeping track of said person frame by frame, and identifying said person in video sequences where the person of interest has not been seen before.</p> <p>The final product is able to both detect and recognize people in video, as well as estimating their 3D-position relative to the camera. The product is modular and any part can be improved or changed completely, without changing the rest of the product. This results in a highly versatile product which can be tailored for any given situation.</p>			
<b>Nyckelord</b> Keywords    Computer Vision, Re-identification, Pedestrian detection, 3D-position estimation			



# Abstract

Because of the increase in the number of security cameras, there is more video footage available than a human could efficiently process. In combination with the fact that computers are getting more efficient, it is getting more and more interesting to solve the problem of detecting and recognizing people automatically.

Therefore a method is proposed for estimating a 3D-path of a person of interest in multiple, non overlapping, monocular cameras. This project is a collaboration between two master theses. This thesis will focus on recognizing a person of interest from several possible candidates, as well as estimating the 3D-position of a person and providing a graphical user interface for the system. The recognition of the person of interest includes keeping track of said person frame by frame, and identifying said person in video sequences where the person of interest has not been seen before.

The final product is able to both detect and recognize people in video, as well as estimating their 3D-position relative to the camera. The product is modular and any part can be improved or changed completely, without changing the rest of the product. This results in a highly versatile product which can be tailored for any given situation.





# Acknowledgments

I would like to thank my loving wife Matilda Johansson for the endless support and understanding, without her this project would have been impossible.

To my dear friend and colleague Johannes Markström, for the constant encouragement and help during this project, a big thank you.

Great many thanks to Peter Bergström, whose support and input throughout this project has been invaluable.

I would also like to thank my family for the support provided during the entire project.

A thanks goes out to the people at the forensic document and information technology unit, for letting us use their equipment and facilities.

Finally I would like to thank Erik Ringaby for providing expertise and guidance whenever needed.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Definitions . . . . .	1
1.3	Problem Formulation . . . . .	2
1.4	Purpose and Goal . . . . .	3
1.5	Limitations . . . . .	4
1.6	Hardware . . . . .	5
1.7	Related Work . . . . .	5
1.8	Choices and Motivations . . . . .	6
1.8.1	Feature Channels . . . . .	6
1.8.2	Feature Extractors . . . . .	7
1.8.3	Classifiers . . . . .	7
1.9	Contributions . . . . .	7
1.10	Report Outline . . . . .	7
<b>2</b>	<b>Theory</b>	<b>9</b>
2.1	People Detection . . . . .	9
2.2	Person of Interest Recognition . . . . .	10
2.2.1	Feature Channels . . . . .	10
2.2.2	Feature Extractions . . . . .	12
2.2.3	Classifiers . . . . .	13
2.3	3D-Position Estimation . . . . .	15
2.3.1	Estimating the Error . . . . .	19
2.4	Kalman filtering . . . . .	20
2.5	Camera Calibration . . . . .	20
<b>3</b>	<b>Implementation</b>	<b>21</b>
3.1	Core System . . . . .	21
3.2	User Interface . . . . .	21
3.2.1	GUI . . . . .	21
3.2.2	Visualization . . . . .	21
3.2.3	Main System Interface . . . . .	22
3.3	Selecting Person of Interest . . . . .	24
3.4	Person of Interest Recognition . . . . .	27

3.4.1	Feature channels . . . . .	27
3.4.2	Feature extractions . . . . .	29
3.4.3	Classifiers . . . . .	30
3.5	Background model . . . . .	31
<b>4</b>	<b>Evaluation</b>	<b>33</b>
4.1	Ground Truth Generation . . . . .	33
4.2	Person of Interest Recognition Evaluation . . . . .	34
4.3	Complete System Evaluation . . . . .	35
<b>5</b>	<b>Results</b>	<b>39</b>
5.1	Person of Interest Recognition Results . . . . .	39
5.1.1	Feature Channels . . . . .	45
5.1.2	Feature Extractors . . . . .	52
5.1.3	Classifiers . . . . .	57
5.2	Complete System Results . . . . .	60
5.2.1	Evaluation with Ground Truth . . . . .	60
5.2.2	Evaluation with Ground Truth used for Detecting People . . . . .	63
5.2.3	Evaluation with Ground Truth used for Recognizing the PoI . . . . .	67
5.2.4	Evaluation without Ground Truth . . . . .	70
<b>6</b>	<b>Discussion</b>	<b>75</b>
6.1	Person of Interest Recognition Discussion . . . . .	75
6.2	Complete System Discussion . . . . .	76
6.3	Future Work . . . . .	78
<b>7</b>	<b>Conclusion</b>	<b>79</b>
	<b>Bibliography</b>	<b>81</b>

# Chapter 1

## Introduction

Here general information about this thesis is presented, such as background, problem formulation and definitions. The report outline is also presented here.

### 1.1 Background

The area of this project is a current one, since more and more security cameras are installed. This yields more data than a human could efficiently process, and computer aid is therefore necessary. Furthermore computer vision is generally computationally intensive, and as computers keep getting faster and faster, more and more computer vision algorithms are becoming practical to use. Another aspect of this project is that the client does not have constraints regarding speed, but rather precision. This is contrary to most current uses where speed, often real-time, is a requirement. In this project speed may therefore be traded for higher precision, instead of the other way around. Applications and further development of this project may lead to determining whether a person could have been present at a location or event. It may also help while trying to extract the true identity of the person.

### 1.2 Definitions

Following are some definitions used throughout this report and project.

#### **PoI**

Person of Interest, the selected person to track through the sequences.

#### **GUI**

Graphical User Interface, the main interface from which the user controls the program.

**Class**

A set of objects, which all share a specific feature.

**Class label or label**

Describes a class with a tag, for instance “PoI” could be used as class label for persons identified as the person of interest.

**System**

The software which is designed to accomplish the goals of this project.

**Global coordinate system**

A Cartesian 3D-coordinate system to which all positions are relative. For a more elaborate definition see Section 2.3.

**Bounding box**

An axis aligned rectangle in an image, used to mark the presence of a supposed person. The entire person should be contained inside the rectangle.

**Ground truth**

The ground truth of a video consists of a set of bounding boxes and identification numbers. Every bounding box corresponds to a person in a specific frame. Each bounding box has an identification number, which is unique for the person within the bounding box, and consistent over the entire video.

## 1.3 Problem Formulation

In this project the method for finding the 3D-position of a person of interest in multiple video sequences was divided into 5 smaller problems;

1. Sorting the video sequences and selecting the order of the frames, so that they are chronologically ordered.
2. Detecting every person in an image, or video frame.
3. Recognizing the person of interest, if present, from a set of people.
4. Estimating the 3D-position of a person in an image.
5. Filtering and predicting the estimated path of a person.

The parts 1, 2 and 5 are only briefly discussed in this report, they are covered in the thesis by Johannes Markström [13]. Parts 3 and 4 are the main focus of this thesis, but the user interface of the complete system was also a big part. A concept image of the final product may be seen in Figure 1.1.

The project was divided into several difficulty levels, the idea was to create a working base system fulfilling the lowest difficulty level. When this was achieved the system would be improved to handle a difficulty level as high as possible. A detailed explanation of the difficulty levels can be found in Section 1.5. The project was also designed to require as little human interaction as possible, which has

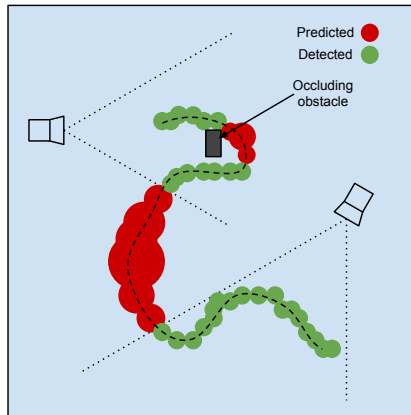


Figure 1.1: A concept of what the final product visualization could look like. The dots represent the estimated or predicted positions of the person of interest.

influenced several choices throughout the project. This includes the 3D-position estimation, which only requires the height of the PoI in addition to the camera information, as well as the selection of the PoI, in which the user only has to select the PoI in one video, rather than every video the PoI should be detected in.

## 1.4 Purpose and Goal

The purpose of this thesis was to implement and evaluate a recognizer, which could be used to separate a specified person of interest from a set of people. Functionality to estimate the 3D-position of a PoI, using an monocular camera should also be implemented. Finally a user interface from which the user could control the entire system should be created.

The goal was that the recognizer ought to be easy to extend or redesign, and individual parts of the recognizer should be interchangeable. It also ought to be able to handle different observation angles, changes in illumination and work for different resolutions.

LEVEL	Color channels	Color variance between cameras	Knowledge of global position and angle of cameras.	Intrinsic parameter	Calibration image	Height of Pol	Freedom regarding placement of cameras.	At least one video sequence where Pol is seen.
1	3	Low	Known	Known	Not Available	Available	Very limited	Available
2	3	Low	Known	Not Available	Available	Available	Very limited	Available
3	3	Low	Known	Not Available	Available	Available	Limited	Available
4	3	High	Known	Not Available	Available	Available	Limited	Available
5	3	High	Known	Not Available	Available	Not Available	Limited	Available
6	1 or 3	High	Known	Not Available	Available	Not Available	Limited	Available

Figure 1.2: The restriction levels summarized in a table.

## 1.5 Limitations

Figure 1.2 shows a visual representation of the different restrictions on the environment in which the program is supposed to be used. The design of the level is such that the higher level the more flexible the program is, and the harder the problem is to solve. Below is a more detailed explanation of the different criteria.

- **Color channels**, restrictions regarding the color information in the given videos.
  - 3, Full color image
  - 1, Grayscale image
- **Color variance between cameras**, restrictions regarding the differences in the color mappings of different cameras.
  - Low, almost no difference in how the cameras assign color values.
  - High, some or major difference in how the cameras assign color values.
- **Knowledge of global positions and angle of the cameras**, restrictions regarding the knowledge about the positioning of the cameras
  - Known, static camera with constant focal length, extrinsic parameters known.
  - Unknown, no knowledge of the placement of cameras relative to each other.
- **Intrinsic parameters**, restrictions regarding the knowledge of the intrinsic parameters of the cameras.
  - Known, the intrinsic parameters are known for every camera in the system.



- Unknown, some or all cameras in the system are uncalibrated.
- **Calibration image**, restrictions regarding if there needs to exist an image of a calibration pattern for the cameras.
  - Not available, no image containing a calibration pattern is needed.
  - Available, a sequence containing calibration patterns exists for every uncalibrated camera.
- **Height of the person of interest**, restrictions regarding the knowledge of height of the person of interest.
  - Known, the height of the person of interest is known, with some (known) uncertainty.
  - Unknown, the height of the person of interest is not known.
- **Freedom regarding placement of cameras**, restrictions regarding the cameras angle, relative to the people.
  - Very limited, all cameras observe the people from approximately the same angle, relative to the people.
  - Limited, all cameras are placed in front of, or above the people, or somewhere in between.
- **At least one sequence where the person of interest is seen**, restrictions regarding if there is a sequence where the person of interest is known to appear.
  - Available, one or more sequences in which the person of interest is known to appear does exist.
  - Not available, there is no knowledge regarding if the person of interest appears in any video sequence.

## 1.6 Hardware

The system is designed to work on a standard PC with a Windows operating system. The following operating systems are supported; Windows XP, Windows Vista and Windows 7.

## 1.7 Related Work

The problem of person recognition is commonly discussed in computer vision, and many different approaches have been presented. Following are some vastly different approaches to the problem of person recognition.

Paper [12] made an effort to try and find the best combination of feature channels to describe a given person, by comparing the variations in a set of feature channels, both in the specific person, and in other persons. The different channels were weighted depending on the result, and every person that was to be recognized had an own set of weights.

Paper [11] tried to combine high- and low-level features to recognize a person. The high level descriptions could be, for instance;

- Does the person have a backpack?
- Does the person wear shorts or trousers?
- Does the person have glasses?

Paper [4] based their recognition on the symmetry and asymmetry lines of a person. Pixels closer to these lines would affect the resulting descriptions more than pixels further away from the lines. The symmetry lines were calculated in the HSV color space, see Section 2.2.1. The calculation of the symmetry and asymmetry lines required a good background segmentation in order to get stable results.

Paper [23] took a completely different approach to solve this problem, and based their recognition on the gait of a person, instead of the visual appearance. This method was based on the silhouette of a person and how it changed with time.

## 1.8 Choices and Motivations

Since the PoI recognizer should be easy to extend, change or modify, it was decided to design a modular recognizer. The feature channels, feature extractions and classifiers should all be independent of each other, so that any one could be changed or modified, without having to redesign the entire recognizer. Following are the choices and motivations of the feature channels, feature extractors and classifiers.

### 1.8.1 Feature Channels

Based on the articles discussed in Section 1.7 the following feature channels were chosen; RGB, HSV, YCrCb and Gabor as well as Schmid. These are described in detail in Section 2.2.1. The main reason for choosing these is that they are all often used by themselves or in combination with each other. Which article is using what feature channels is described in the section for each feature channel.

## 1.8.2 Feature Extractors

The feature extraction methods chosen were; the mean, the histogram [15], and a sub grid version of both. Since most articles use histograms to describe their features in one way or another, the histogram was an obvious choice. The mean by itself may not be an exciting feature channel, but if the bounding box is divided into a grid, the mean could give more useful information, such as the color of the shirt, hair or trousers. Many of the articles also try to divide the bounding box into logical parts such as head, torso, legs etc. which motivated the use of feature extractors on separate parts of the bounding box.

## 1.8.3 Classifiers

Regarding the choice of classifiers, none of the articles discussed used classifiers which fulfilled the demand of being able to train both off-line and on-line. Instead the classifiers chosen were neural networks [20] and nearest centroid [21] classifiers. The reason for choosing a neural network classifier is that it has been around for a relatively long time and handles both off-line and on-line training. The other choice, the nearest centroid classifier, was chosen on account of its simplicity. Like the neural network classifier it is able to handle both off-line and on-line training.

# 1.9 Contributions

Together with [13] a complete system which is able to detect people, recognize a person of interest among these, and estimate the 3D-position of this person, using only a monocular camera, has been created. In this thesis a person of interest recognizer is presented, which is modular and easy to modify or extend. New results showing that the choice of classifier and feature extraction process may be as important as the choice of feature channels, and may vary from person to person, is presented. A method for estimating the 3D-position of a person in a monocular camera is also presented.

## 1.10 Report Outline

This report is divided into four sections; introduction, theory, implementation and finally evaluation and result. The introduction contains general information about the project, thesis and this report. The theory section details the different methods used in this project. The implementation part describes the implementation of the methods detailed in the theory section. The final part of this report discusses the evaluation methods used and the results from these, along with the author's thoughts about this project.

Some sections of this report are not directly part of this thesis and are instead parts of the thesis by Johannes Markström [13]. These sections are:

- People detection
- Kalman filtering of the estimated path.
- Camera calibration.

# Chapter 2

## Theory

### 2.1 People Detection

This section is a short summary of the people detector which was implemented by Johannes Markström, for a more elaborate explanation see his thesis [13].

This system uses a detector based on extracting patches and propagating them through a tree classifier consisting of a number of random trees. The result is votes in the different positions, where a centroid of a person may exist. A patch is defined differently when training a tree and when detecting people.

When training, a patch is a combination of feature channels and offset vectors as well as a class label. If the class label indicates that the patch was derived from a person, then the offset vector contains the relative position of the centroid of said person. If the label indicates that the patch is not derived from a person, the offset vector is undefined. The goal of the training is to get a high purity in class labels and a small variation of the offset vectors in every leaf. This is achieved by choosing fast binary tests, which decide into which branch the given patch should be placed. When the training is completed, every leaf will contain a number of offset vectors, pointing to the possible positions of a person, relative to the patches that ended up in the leaf.

When detecting, a patch contains feature channels and the position of the patch in the image. The patch will be passed through the forest. When the patch reaches a leaf, votes will be placed on the position pointed by the offset vectors in the leaf, translated by the position of the patch.

## 2.2 Person of Interest Recognition

The main concept of this PoI recognizer is that it should be modular. That is, it should be easy to add or remove features or classifiers. In order to achieve this goal the following design decisions were made.

- The input image should be supplied in 8-bit 3-channel BGR-format.
- The features were divided into a feature channel and a feature extractor. The feature channels and extractors are designed to be fully interchangeable. For instance could the same extractor be used on multiple channels, and multiple extractors could be used on the same channel.
- The classifiers were implemented so that the input and output format was the same for all classifiers.

The result is a system in which features and classifiers are easily added, updated or removed.

### 2.2.1 Feature Channels

The feature channels are the data that is extracted from the original image, in which the person of interest may or may not appear. Before any feature extraction is performed, the original image is converted to 32-bit float with values in the range  $[0, 1]$ . The feature channels may be individual color channels such as RGB, or results of more sophisticated filter responses. A value in a feature channel should be directly linked to the pixel with the same coordinates in the original image. Every feature channel has exactly one value per pixel in the original image. This leads to that for instance RGB will return three feature channels, one for every color. These feature channels are designed so that any feature extraction may be applied to any feature channel. All feature channel values will be described as floating point numbers in the range  $[0, 1]$ .

#### RGB

The RGB feature channels are the color values described as a combination of red, green and blue. This is the standard way of representing an image, and while it is not the most sophisticated color space, it is still widely used for re-identification by itself [2][1], or in combination with other color spaces [18][6][11]. Every color will have their own feature channel, with the amount of said color in each position.

#### HSV

The HSV [19] feature channels are the color values described as a combination of hue, saturation and value. The HSV color space is designed to represent colors

in a way, which is more intuitive than the basic RGB color space. In [19] the dimensions are described as follows “Briefly, hue is the dimension with points on it normally called red, yellow, blue-green etc. Saturation measures the departure of a hue from *achromatic*, i.e., from white or gray. Value measures the departure of a hue from black, the color of zero energy. These terms, . . . , are meant to capture the artistic ideas of *hue*, *tint*, *shade* and *tone* . . .”. The HSV color space is used in [18][6][12][11]. H, S and V will each be contained in its own feature channel.

### YCrCb

The YCrCb color space [7] describes an image by the luma, Y, and the red and blue chromacity differences, Cr and Cb respectively. The main usage of this color space lies in image and video compression. The reason for this is that the luma component can be stored with high precision, while the chromacity differences may be stored in a lower resolution. This type of compression makes sense since humans are more sensitive to grayscale than color. The YCrCb color spaced is used alongside RGB and HSV in [6][12][11]. Y, Cr and Cb will each be contained in its own feature channel.

### HSVCrCb

The HSVCrCb feature channels are a combination of the HSV and YCrCb color channels, without the luma channel of YCrCb, as proposed in [6][12] and [11].

### Schmid

The Schmid feature channels are the spatial filter responses of different Schmid filter [17] applied to the luma channel of YCrCb. Schmid filters are originally designed to model rotation invariant textures, however [6] found them useful when re-identifying people in different views and poses. In this application the filters are derived from the following formula

$$F(r, \sigma, \tau) = \cos\left(\frac{2\pi\tau r}{\sigma}\right) e^{-\frac{r^2}{2\sigma^2}} \quad (2.1)$$

This is the same formula as [6] uses.  $r$  is defined according to

$$r = \sqrt{(x - x_{center})^2 + (y - y_{center})^2} \quad (2.2)$$

$\sigma$  and  $\tau$  are parameters which determine the shape of the filter, and  $x$  and  $y$  describe the current position in the filter.  $x_{center}$  and  $y_{center}$  describe the center point of the filter. Schmid filters are used together with Gabor filters [5] in [6][12][11].

## Gabor

The Gabor feature channels are composed of spatial filter responses, namely Gabor filters [5] applied to the luma channel of YCrCb, as proposed in [17]. The Gabor filters are designed to mimic specific filtering operations performed in the pre-processing of human vision system. Here, the following formula is used to determine the shape of the schmid filters

$$F(x, y, \lambda, \theta, \psi, \sigma, \gamma) = e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}} \cos\left(\frac{2\pi x'}{\lambda} + \psi\right) \quad (2.3)$$

$\lambda, \theta, \psi, \sigma$  and  $\gamma$  are parameters which determine the shape of the filter and  $x'$  and  $y'$  are defined according to

$$\begin{aligned} x' &= x \cos(\theta) + y \sin(\theta) \\ y' &= -x \sin(\theta) + y \cos(\theta) \end{aligned} \quad (2.4)$$

In this formula  $x$  and  $y$  describe the current position in the filter. Gabor filters are used together with Schmid filters [5] in [6][12][11].

### 2.2.2 Feature Extractions

In this report *feature extraction* refers to the mapping from feature channel to feature vector. A *feature vector* is a single dimensional or multi dimensional value which is used by the classifiers to separate the PoI from other people. To qualify as a feature extractor the following criteria must be met.

- The size of the feature vector should be independent of the input feature channel.
- The value, or values, of the feature vector must be in the range  $[0, 1]$ .
- Two identical feature channels must always return identical feature vectors.

These demands are chosen to allow classifiers to be designed without the prior knowledge of the origin of the feature vector. As a result of these demands, simply passing the entire feature channel as feature vector would not classify as a feature extractor. Passing the feature channel as feature vector would change the dimension of the feature vector depending on the size of the feature channel. However, if the feature channel was resized to a fixed size and then returned, it would classify as a feature extractor.

## Mean

The “mean” feature extractor returns a one dimensional feature vector for every feature channel. The extraction formula is the sum over all values in the feature channel, divided by the area of the feature channel.



## Subgrid

The “subgrid” feature extraction is not a feature extractor in itself. Instead of applying a standard mapping, feature channel to feature vector, this feature extractor subdivides the input image and sends these patches to another feature extractor, which then returns a feature vector for every patch.

## Histogram

The “histogram” feature extractor returns the histogram [15] over a feature channel as the feature vector. In order to maintain the range  $[0, 1]$  the values of the histogram will be divided by the number of pixels in the feature channel. The result is that every value of the feature vector represents the percentage of pixels which are contained in the corresponding bin. Histograms are commonly used to extract feature vectors and are used in [4][2][18][12][1].

### 2.2.3 Classifiers

The classifiers decide whether a person is the person of interest or not by applying a function to the feature vector returned by the feature extractors. These may also return a dissimilarity value, describing how closely the person matches earlier instances of the person of interest. The classifiers may be learning the appearance of the person of interest as more and more images are analyzed, or it could just be based on an initial description of the person of interest. The only common part of these classifiers is that they all accept a feature vector and return if the person is the person of interest or not.

#### Nearest Centroid

The “Nearest centroid” classifier is a very simple classifier, based on the “minimum distance classifier” described in [21]. The initial training of the nearest centroid classifier requires a set of class labels, as well as a set of instances corresponding to those labels. For every given label a *centroid* is created, based on the instances of that label. A centroid is a vector containing the mean over every dimension in the feature vector, calculated from every prior instance of the given label.

The classification of a feature vector is based on finding the closest centroid to that feature vector. The label of the nearest centroid is chosen as the label of the feature vector, hence the name of the classifier.

When a feature vector has been classified, the corresponding centroid is updated to account for the new instance. Every detected instance is trained on, and every instance is given equal weight.

The distance used for comparing and finding the closest centroid is the  $L^2$ -norm. The dissimilarity value returned from this classifier is based on the previously mentioned distance. The actual value is calculated as  $\tanh(\text{distance})$ , this ensures that the range of the dissimilarity is  $[0, 1]$  where 0 is a perfect match, and 1 would correspond to the feature vector being infinitely far away from all classes. One advantage of this classifier is that it is very fast both when training and classifying. Another advantage is that the memory usage is very low. A disadvantage of this classifier is that it cannot handle complex relations within a class. In order for this classifier to be accurate, the classes must be linearly separable in the space spanned by the feature vector.

## Neural network

The “neural network” classifier is based on neural networks [20]. These networks represent a mapping from the input feature vectors to a voting for every known class. The class with the highest vote is assumed to be the correct class. The neural network is inspired by the biology of a brain, where every node in the network is a model of a neuron. The artificial neuron was introduced by [14]. Briefly, a neuron is a mapping from several inputs to a single output. Every input is given a weight, and the sum of the weighted inputs is transformed by an activation function to the output. The output is then used as input for the next layer in the neural network. The final layer will have the same number of neurons, and outputs, as there are classes, and the result will be scaled into the range  $[0, 1]$  where 1 is supposed to represent a perfect match, and 0 a complete mismatch, however because of the design of neural network this definition may not hold in all cases.

When using a neural network as a classifier the weights are chosen so that the input feature vector gives an output which is as close to the correct label as possible. The method used for choosing these weights is based on how the output error, of a given training data, changes when modifying the weights. If increasing a weight lowers the output error, the weight is increased until the error starts to increase again. For a more detailed description see [16].

In this system the activation function is chosen as (2.5), since this is the standard in the OpenCV [9] implementation which utilizes the training method described in [16]. The formula is

$$y_i = \tanh(w_{0,j} + \sum_j w_{i,j} \cdot x_j) \quad (2.5)$$

where  $y_i$  is the output value of the neuron,  $x_j$  is the input with index  $j$  of the neuron.  $i$  is the index of the neuron at the current layer and  $w_{n,m}$  is the weight of the input  $m$  at the neuron  $n$ .

## 2.3 3D-Position Estimation

In order to estimate the location of the PoI between cameras and video sequences, a way of representing global positions was necessary. The solution is to use the first camera as the reference system, and construct 3D-coordinates based on this. The axes are defined as follows.

- Z-axis, The z-axis is defined as parallel to the normal of the image plane, with positive values in front of the camera.
- X-axis, The x-axis is placed along the rows of an image in the ideal image plane, with positive values to the left of the image.
- Y-axis, The y-axis is chosen as the cross product between the z- and x-axes. This gives a vector which points up relative to an image in the ideal image plane.

The relative rotation of two cameras is described using the same rotation descriptions as airplanes; pitch, yaw and roll. These angles correspond to rotations around the x-, y- and z-axes respectively.

Figures 2.1 and 2.2 show this coordinate system, along with the rotations.

The 3D-position estimation of the PoI is based on transforming the bounding box into 3D-coordinates. This is based on multiple assumptions. Firstly the cameras intrinsic and extrinsic parameters has to be known. Extrinsic parameters are relative to the first camera in the system. Secondly the height of the person as well as the angle of the camera relative to the floor is required.

The basic idea of this method is to extrude the bounding box, from the image into the world, and find the distance at which the height of this bounding box is equal to the height of the PoI. A sketch of this method can be seen in Figure 2.3. The details of this implementation are described below.

The first step is to find the lines which passes through the top center and bottom center of the bounding box of the PoI. These lines can be found by using (2.13) on the top center and bottom center pixels in the bounding box. This equation transforms a point in an image to a line through the pixel in the ideal image plane and the camera center. This line is defined by two points on this line, the first one is the origin, if camera centric coordinates are used, since this corresponds to the camera center. The equation which should be solved is

$$KP_2 = Q_2 \quad (2.6)$$

where  $K$  is the intrinsic camera matrix,  $P_2$  is the second point on the line and  $Q_2$  is the second point projected into the image.  $K$  can be described as

$$K = \begin{pmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.7)$$

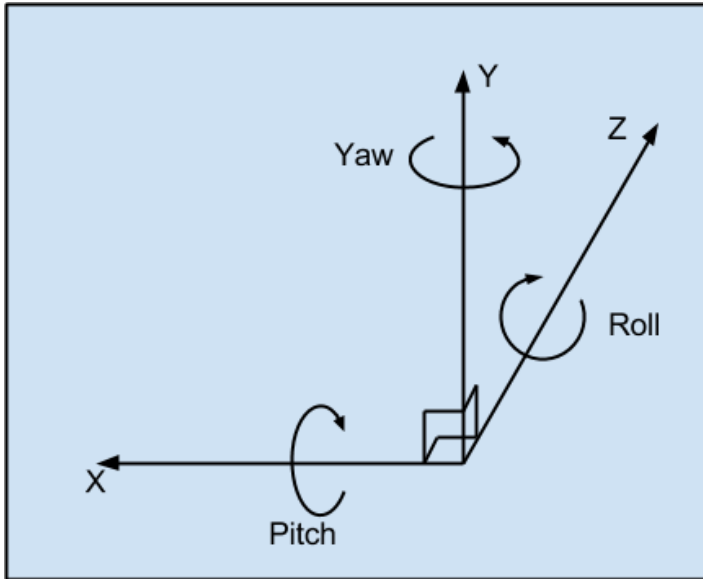


Figure 2.1: A visualization of the global coordinate system.

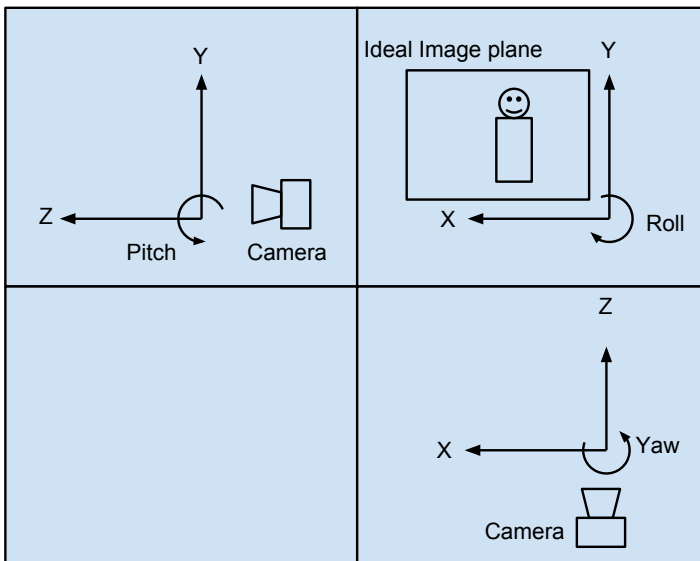


Figure 2.2: A visualization of the global coordinate system, projected along the axes.

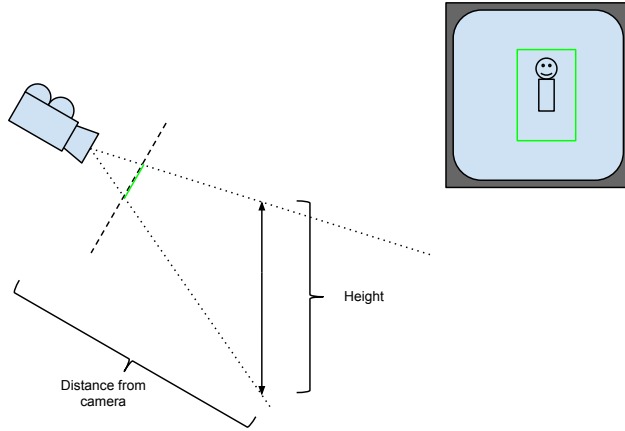


Figure 2.3: An overview of the method used to determine the 3D-position of the PoI

where  $\alpha_x$  and  $\alpha_y$  describes the focal length in pixels,  $\gamma$  corresponds to the skewing between the  $x$  and  $y$  axis and  $u_0, v_0$  is the principal point. If  $P_2$  is described as

$$P_2 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.8}$$

where  $x, y$  and  $z$  are the coordinates of the points along the x-, y- and z-axis respectively, and  $Q_2$  is described in homogenous coordinates as

$$Q_2 = \begin{bmatrix} h_2 \cdot r \\ h_2 \cdot c \\ h_2 \end{bmatrix} \tag{2.9}$$

where  $h_2$  is the homogenous coordinate of  $Q_2$  and  $r$  and  $c$  are the row and column pixel coordinates of the point. (2.6) can be expressed as

$$KP_2 = \begin{bmatrix} \alpha_x \cdot x + \gamma \cdot y + u_0 \cdot z \\ 0 & \alpha_y \cdot y + v_0 \cdot z \\ 0 & 0 & z \end{bmatrix} = \begin{bmatrix} h_2 \cdot r \\ h_2 \cdot c \\ h_2 \end{bmatrix} \tag{2.10}$$

This yields

$$\begin{aligned}
z &= h_2 \\
\Rightarrow \alpha_y \cdot y + v_0 \cdot h_2 &= h_2 \cdot c \\
\Rightarrow y &= h_2 \cdot \frac{c - v_0}{\alpha_y} \\
\Rightarrow \alpha_x \cdot x + \gamma \cdot h_2 \cdot \frac{c - v_0}{\alpha_y} + v_0 \cdot h_2 &= h_2 \cdot r \\
\Rightarrow x &= \left( r - u_0 - \frac{c - v_0}{\alpha_y} \right) \cdot \frac{h_2}{\alpha_x}
\end{aligned} \tag{2.11}$$

Which, when combined with (2.8), gives the formula for the second point

$$P_2 = \begin{bmatrix} \left( r - u_0 - \frac{c - v_0}{\alpha_y} \right) \cdot \frac{h_2}{\alpha_x} \\ h_2 \cdot \frac{c - v_0}{\alpha_y} \\ h_2 \end{bmatrix} \tag{2.12}$$

If  $h_2$  is assumed to be 1 the line can be expressed as the two points  $P_1$  and  $P_2$  according to

$$P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad P_2 = \begin{bmatrix} \left( r - u_0 - \frac{c - v_0}{\alpha_y} \right) \cdot \frac{1}{\alpha_x} \\ \frac{c - v_0}{\alpha_y} \\ 1 \end{bmatrix} \tag{2.13}$$

The next step is to find the distance from the camera at which the distance between the two lines are equal to the height of the PoI, and the vector between these points align with the direction of gravity. In order to find the 3D-position of the PoI the following relation must hold

$$\left\{ \lambda_1, \lambda_2 \left| \begin{cases} |\lambda_1 - \lambda_2| &= h \\ (\lambda_1 - \lambda_2) \bullet \hat{z} &= \sin(\varphi) \cdot h \end{cases} \right. \right\} \tag{2.14}$$

$\lambda_1$  and  $\lambda_2$  are the points on the top and bottom line, respectively. The first constraint

$$|\lambda_1 - \lambda_2| = h \tag{2.15}$$

states that the distance should be equal to the height of the PoI. The second constraint

$$(\lambda_1 - \lambda_2) \bullet \hat{z} = \sin(\varphi) \tag{2.16}$$

describes that the vector between the points should align with gravity. If  $a$  and  $b$  are points on the line through the top of the bounding box, and  $c$  and  $d$  are points on the line through the bottom of the bounding box, the directions of these lines can be found as

$$u = b - a \tag{2.17}$$

for the top line and

$$v = d - c \tag{2.18}$$

for the bottom line.  $\lambda_1$  and  $\lambda_2$  can then be described in terms of these lines as

$$\begin{aligned}\lambda_1 &= a + k \cdot u \\ \lambda_2 &= c + w \cdot v\end{aligned}\tag{2.19}$$

where  $k$  and  $w$  are scalars, which have the following relation

$$w = k \cdot \alpha + \beta\tag{2.20}$$

$\alpha$  and  $\beta$  can be expressed as

$$\begin{aligned}\alpha &= \frac{u_z}{v_z} \\ \beta &= a_z - c_z + h \cdot \sin(\varphi)\end{aligned}\tag{2.21}$$

This formulation ensures that the second constraint (2.16), is fulfilled. The next step is to determine  $k$  so that the first constraint (2.15) is fulfilled. The  $k$  which fulfills this can be described by

$$k = \frac{-\omega_2}{\omega_1} \pm \sqrt{\left(\frac{\omega_2}{\omega_1}\right)^2 + \frac{\omega_3}{\omega_1}}\tag{2.22}$$

with  $\omega_1$ ,  $\omega_2$  and  $\omega_3$  are defined as

$$\begin{aligned}\omega_1 &= \delta_x^2 + \delta_y^2 + \delta_z^2 \\ \omega_2 &= \delta_{1x} \cdot \delta_{2x} + \delta_{1y} \cdot \delta_{2y} + \delta_{1z} \cdot \delta_{2z} \\ \omega_3 &= \delta_{1x}^2 + \delta_{1y}^2 + \delta_{1z}^2 - h^2\end{aligned}\tag{2.23}$$

and with  $\delta_1$  and  $\delta_2$  are defined as

$$\begin{aligned}\delta_1 &= a - c - \beta \cdot d \\ \delta_2 &= b - \alpha \cdot d\end{aligned}\tag{2.24}$$

To get the position of the centroid of the PoI, take the mean of  $\lambda_1$  and  $\lambda_2$

$$p = \frac{\lambda_1 + \lambda_2}{2}\tag{2.25}$$

Finally the point should be transformed into the global coordinate system

$$p_g = T p\tag{2.26}$$

here  $T$  is the extrinsic camera matrix of the camera.

### 2.3.1 Estimating the Error

Since the values used when calculating the 3D-position of the PoI are estimates, small errors in the position are unavoidable. It is however possible to estimate

these errors. This is done by changing the input values to the worst case versions and estimating the position for every combination of these. The worst cases of the input variables were chosen by applying an estimated offset to them, such that  $\text{input} - \text{offset} < \text{true value} < \text{input} + \text{offset}$ . The resulting positions are compared to the original estimated position, and the maximum distance is used as error estimate.

## 2.4 Kalman filtering

To remove outliers and reduce noise in the estimated 3D-positions of the PoI, some filtering was used. It was decided to use the Kalman filter [10] as it could be used both for filtering the positions, and predicting the position, when the PoI was not detected by the system. The dimensions of the position were assumed to be independent of each other. The filter model was based on the position and speed of the PoI. For a more detailed description of the method see [13].

## 2.5 Camera Calibration

When the intrinsic parameters of the cameras in the system are unknown, camera calibration is used to estimate these parameters. The intrinsic parameters describe how the world, described in camera centric coordinates, is mapped onto the resulting image. In this system the standard implementation from OpenCV [9] was used, which is based on [24] and [3]. For a more detailed description of the method see [13].



# Chapter 3

## Implementation

### 3.1 Core System

The core system of this project was designed by Johannes Markström [13], and is only briefly discussed in this report. The system is based on an analyzer choosing which video sequence to process, based on the time of the next frame in all video sequences. When a video sequence is finished the analyzer merges the old data with the new, and when the last video sequence has been processed the data is filtered, see Section 2.4 and visualized, see Section 3.2.2.

### 3.2 User Interface

Here the different parts of the user interface are discussed.

#### 3.2.1 GUI

The graphical user interface in this project is based on the Windows specific GDI library, in collaboration with OpenGL, (loaded with GLEW [8]). GDI is used for printing text, and OpenGL displays images. The CommonControl library is used for buttons, toolbars and progress bars.

#### 3.2.2 Visualization

In order for the system to be user friendly, the results has to be presented in a clear and intuitive way. The solution was to project the estimated path along the gravity vector, this gave an approximation of the ground plane. Since the altitude of the PoI rarely changes, and when the altitude of a person changes it is for

obvious reasons such as stairs or hills, no important data is lost in this type of visualization. In some unusual cases, for instance multilevel buildings, this method is not optimal. To further help the user to understand the results, the position and field of view of the cameras are marked, in addition to the estimated path. This helps when trying to find the bearings of the path. A scale indicator is also included to enable the user to estimate distances between points.

The path is visualized by displaying each estimated position as two circles, one black and the other red. The black marked the centroid of the PoI, and the red indicated the uncertainty of the estimated position. The uncertainty is shown by letting the radius of the red circle be the estimated maximum error in cm, see Section 2.3.1, of the position. The cameras are visualized by adding white dashed lines along the edges of their field of view. An example of this visualization can be seen in Figure 3.1.

### 3.2.3 Main System Interface

The first window that is shown to the user when the system is started lets the user choose the videos that are going to be analyzed. This window can be seen in Figure 3.2. The button “Add Videos” is used to add new videos that should be analyzed. When this button is pressed a file explorer is presented to the user, which is used to select the desired videos. Every video is assumed to have a corresponding configuration file, with the same name as the video, but the extension “vsc”. In this file information about the intrinsic and extrinsic parameters of the camera, along with the starting time of the video, is placed.

If a configuration file is nonexistent, or not complete, the user may use the button “Edit Config File”. When this button is pressed the configuration file, if existent, of the selected video is read by the system, and a popup which allows the user to change the settings of the camera is presented. This popup may be seen in Figure 3.3. In this window the starting time of the video is entered by the user, along with the pitch, relative to gravity, of the camera. When the extrinsic and intrinsic parameters of the camera should be changed, the button “Edit Camera Matrices” should be pressed. This brings up a new window, see Figure 3.4. In this window the position and rotation of the camera may be entered. If the button “Select calibration images” is pressed the user is asked to choose the calibration images, from which to estimate the cameras intrinsic parameters. This window is presented in Figure 3.5.

In the main window the user is also asked to supply the estimated height of the PoI, along with an error estimate of this height. (3.1) must hold in order for the error estimation to be accurate.

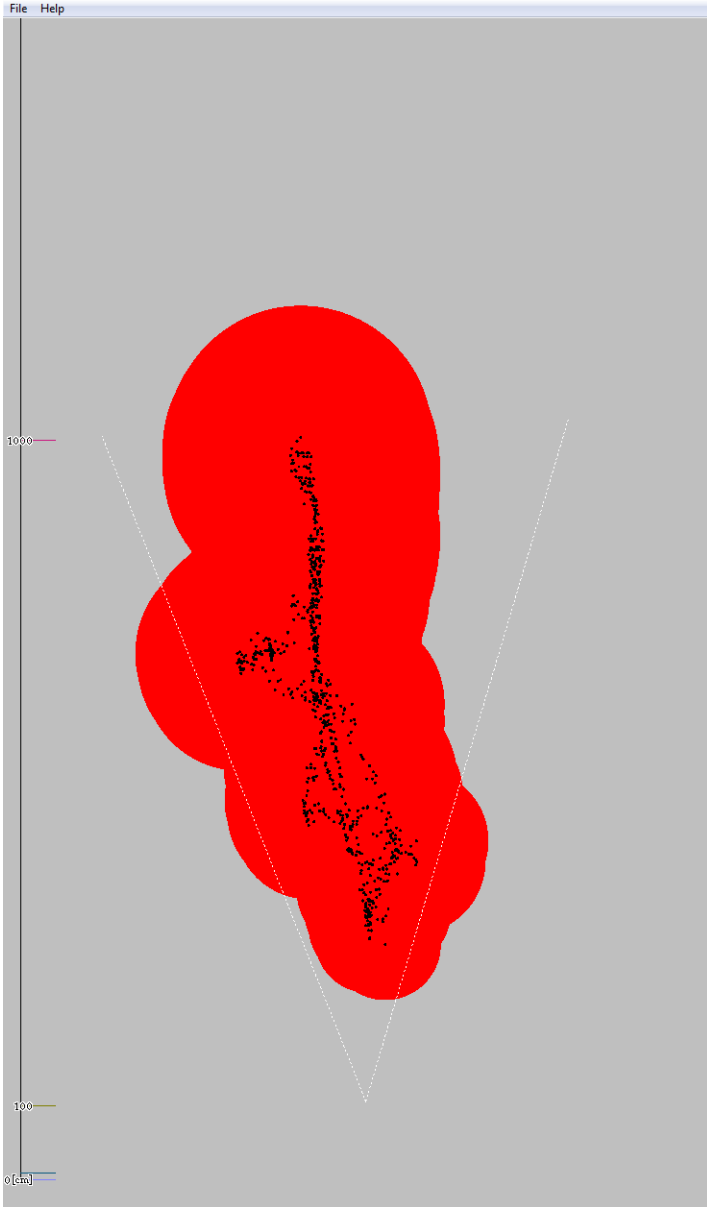


Figure 3.1: An unfiltered path estimation of the PoI, based on ground truth.

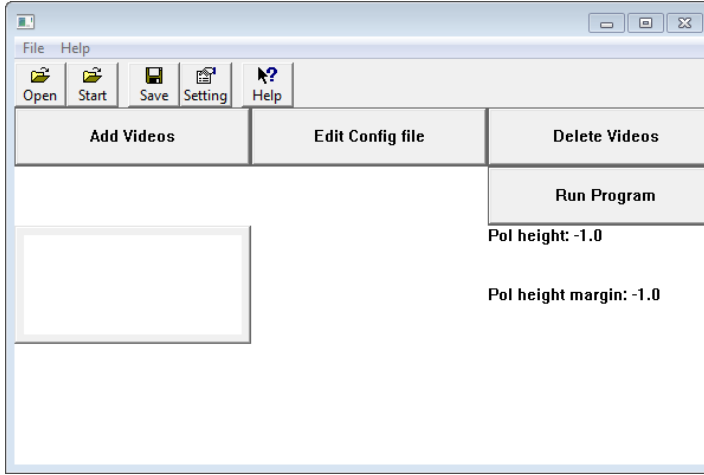


Figure 3.2: The main window of the system.

$$\begin{aligned}
 & h - \varepsilon < h_t < h + \varepsilon \\
 & h = \text{Estimated height of the PoI, in cm} \\
 & h_t = \text{True height of the PoI, in cm} \\
 & \varepsilon = \text{Error of the estimated height of the PoI, in cm}
 \end{aligned}
 \tag{3.1}$$

When all videos that should be analyzed are selected and have valid configuration files, the button “Run Program” is used to continue to the next step. When the button is pressed a popup window is displayed, see Figure 3.6. This popup window allows the user to select the people detector, and PoI recognizer which should be used when analyzing the videos. When the button “Run Program” is pressed, the system starts to analyze the supplied videos.

The toolbar located in the top of the window is not used, and the buttons doesn’t do anything, they are left there for future expansion of the system.

### 3.3 Selecting Person of Interest

In order to let the system learn the appearance of the PoI, the user is asked to supply a set of instances of the PoI. To keep the program user friendly and easy to use, a help tool for selecting the PoI was implemented. It works as follows; the system is given a video sequence, in which the PoI is known to appear. From this sequence the system stores every detected person. When finished, the PoI is selected among these detections, by the user. An example of the interface of this tool is shown in Figure 3.7

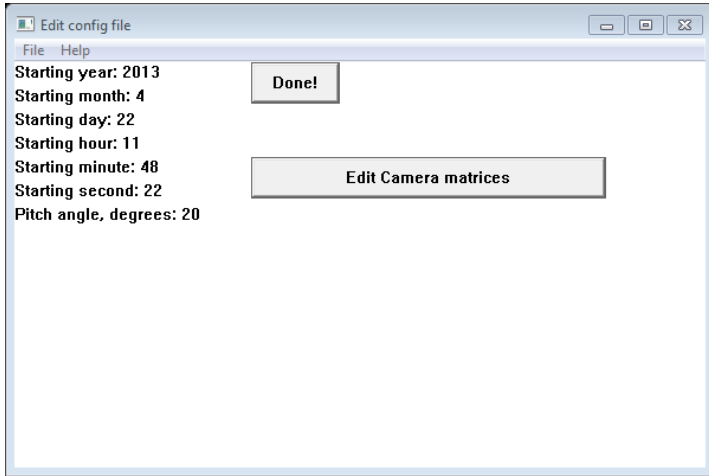


Figure 3.3: The window used for editing the configuration file of a video.

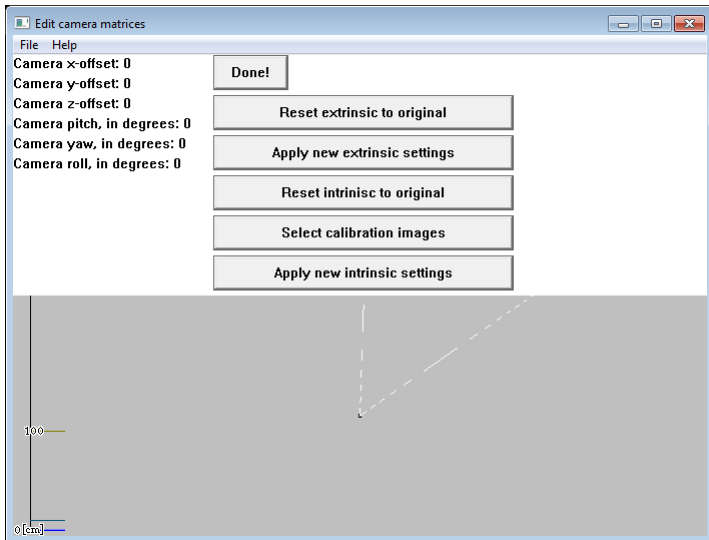


Figure 3.4: The window used when editing the extrinsic and intrinsic camera parameters.

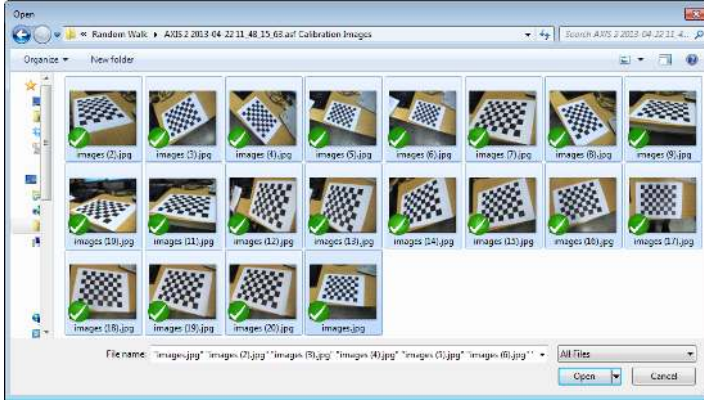


Figure 3.5: The interface when choosing the images for camera calibration.

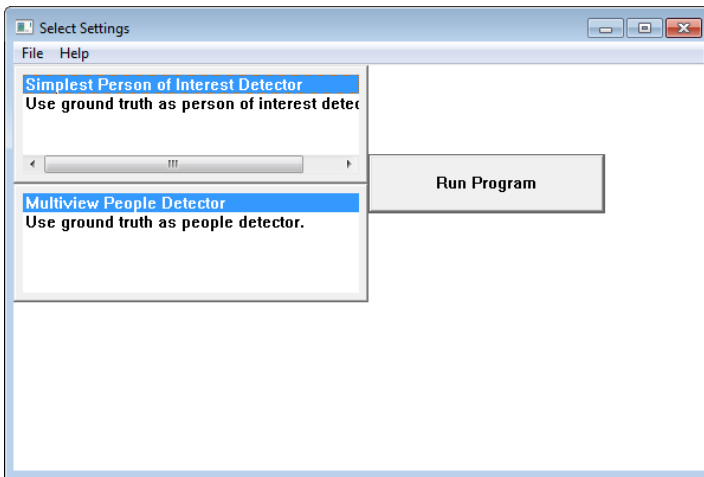


Figure 3.6: The window where the user is asked to select which people detector and PoI recognizer should be used.



(a) Label selection of a group.

(b) Selecting a group to label

Figure 3.7: These are the two main windows of the selecting PoI tool.

The detected people are grouped together by the system, in order to avoid excessive labeling from the user. Two instances are grouped together if they are close enough, both temporally and spatially. To minimize the chance of incorrect grouping the difference in time, or frame number, between two instances within the same group may not exceed  $n$ .

## 3.4 Person of Interest Recognition

The following sections discusses the implementation details of the PoI recognizer.

### 3.4.1 Feature channels

The following section describes how the different feature channels were implemented. The input image is an image described as BGR, with floating point precision and in the range  $[0, 1]$ , unless otherwise stated.

#### RGB

The RGB feature channels were implemented using OpenCV's `split` function, which takes an image and returns the individual channels of an image.

#### HSV

The HSV feature channels were implemented using OpenCV's `cvtColor` function, with the setting `CV_BGR2HSV`. The resulting image will be split in the same way as the RGB channels were extracted. The Hue component is scaled a factor  $1/360$ , in order to keep the feature channel within the range  $[0, 1]$ .

## YCrCb

The YCrCb feature channels were implemented using OpenCV's `cvtColor` function, with the setting `CV_BGR2YCrCb`. The resulting image will be split in the same way as the RGB channels were extracted.

## Schmid

In this project thirteen Schmid filters were used, as in [6]. The parameters used to create the filters are listed in (3.3). The filter responses were calculated using the OpenCV function `filter2D`, which calculates the filter response of an arbitrary filter onto an image. All filters are normalized before applied, to avoid bright areas getting higher value responses than darker areas. The normalization is performed as (3.2), this ensures that the filter response is in the range  $[-1, 1]$ . The filter response is then transformed to the feature channel with interval  $[0, 1]$  by (3.4). The images were also scaled before filtering, so that the range was  $[-1, 1]$ . After the filter responses were calculated, the result was then scaled back to the range  $[0, 1]$ . A visualization of the filters can be seen in 3.8a.

$$F_n = \frac{F - \text{mean}(F)}{\sum_{i,j} |F_{i,j}|} \quad (3.2)$$

$F_n$  is the normalized filter.

$F$  is the unnormalized filter.

$$\begin{aligned} \{\sigma, \tau\} = & \{2, 1\}, \{4, 1\}, \{4, 2\}, \\ & \{6, 1\}, \{6, 2\}, \{6, 3\}, \\ & \{8, 1\}, \{8, 2\}, \{8, 3\}, \\ & \{10, 1\}, \{10, 2\}, \{10, 3\}, \{10, 4\} \end{aligned} \quad (3.3)$$

$\{\sigma, \tau\}$  are the parameters used in equation (2.1)

$$FeatureChannel = \frac{FilterResponse}{2} + 0.5 \quad (3.4)$$

## Gabor

In this project six Gabor filters were used, as in [6]. The parameters used to create the filters are listed in (3.5), the same as used in [6]. The filter responses were calculated using the OpenCV function `filter2D`, which calculates the filter response of an arbitrary filter onto an image. Like the Schmid feature channels, the filters are normalized before the application, using (3.2). The luma channel on which the filters would be applied was transformed into the range  $[-1, 1]$ . The filter response





(a) A visualization of the Schmid filters.

(b) A visualization of the Gabor filters.

Figure 3.8: Visualization of the filters used in the system.

would then be in the range  $[-1, 1]$ , and must therefore be transformed back onto the range  $[0, 1]$  with (3.4) in order to qualify as a feature channel. A visualization of the filters can be seen in 3.8b.

$$\begin{aligned}
 \{\gamma, \theta, \lambda, \sigma^2\} = & \{0.3, 0.0, 4.0, 2.0\} \\
 & \{0.3, 0.0, 8.0, 2.0\} \\
 & \{0.4, 0.0, 4.0, 1.0\} \\
 & \{0.3, \frac{\pi}{2}, 4.0, 2.0\} \\
 & \{0.3, \frac{\pi}{2}, 8.0, 2.0\} \\
 & \{0.4, \frac{\pi}{2}, 4.0, 1.0\}
 \end{aligned} \tag{3.5}$$

$\{\gamma, \theta, \lambda, \sigma^2\}$  are the parameters used in equation (2.3)

### 3.4.2 Feature extractions

The following section describes how the different feature extractions were implemented. All features are applied to a single channel image with floating point precision in the range  $[0, 1]$ .

#### Mean

The “Mean” feature extraction was implemented using OpenCV’s mean function, which returns the mean of an image.

#### Subgrid

The “Subgrid” feature extraction was implemented by dividing the given feature channel into a rectangular grid, and applying a feature extractor on every cell in

this grid. The resulting feature vectors would then be concatenated, row-major order, into one feature vector.

## Histogram

The “Histogram” feature extraction was implemented using OpenCV’s `calcHist` function, which returns the histogram of an image. The resulting values over each bin was then divided by the number of pixels in the image region, in order to keep the feature values in the range  $[0, 1]$ . In the case of only partial overlap of the bounding box and the image, only the pixels in the intersection of the two is accounted for, and the values are divided by the intersection area.

### 3.4.3 Classifiers

The following section describes how the different classifiers were implemented.

#### Nearest centroid

The “Nearest centroid” was implemented as follows. The initial training labels were chosen as “PoI” and “Not PoI”, which corresponded to a set of feature vectors of the PoI, for the “PoI” label, and other people, for the “Not PoI” label. To avoid storing every prior instance of a class, (3.6) was used for updating the centroid of a class label. This formula only required the number of instances, and the centroid, to be stored. The distance used for finding the closest was the  $L^2$ -norm.

$$Centroid_{n+1} = \frac{(n \cdot Centroid_n + FeatureVector)}{n + 1} \quad (3.6)$$

#### Neural Networks

The “Neural networks” classifier was implemented using the OpenCV machine learning library, namely the `CvANN_MLP` class. This class handles the training, both offline and online, as well as prediction. In this implementation, the dimensionality of the input layer is equal to the number of elements in the feature vector. The output layer has two neurons. There are three hidden layers, and the number of neurons in each hidden layer was linearly interpolated from the input and output layers’. In the current version of the system, the online learning is not enabled, only offline learning is performed. The training parameters were not altered from the standard OpenCV implementation, as the training method is very robust to the choice of parameters according to [16]. The dissimilarity value is given as  $Dissimilarity = |1 - \max(output_{PoI}, output_{NotPoI})|$ .



(a) A frame before the foreground segmentation. (b) The same frame after the foreground segmentation.

Figure 3.9: An example of the background model.

## 3.5 Background model

A novel background model was implemented using Matlab. It is based on the median background model [22], with an extension of structure and convex hull to minimize holes in the foreground. Sufficient deviations in the color of the current frame and the background model will correspond to foreground. The color representation is RGB, with L1 norm as deviation measurement. The new approach is to create a convex hull over the segmented foreground objects, which will remove holes. This background model is not used in the project, but a prototype was implemented in Matlab, and a before and after example can be seen in Figures 3.9a and 3.9b.



# Chapter 4

## Evaluation

### 4.1 Ground Truth Generation

The ground truth of a video is created by the user, by a tool developed by Johannes Markström [13]. The user is presented with each frame in the video, one at a time. The user is then asked to mark every person present in the current frame with a bounding box, and then choose an identification number for that person. When every person is marked, the user continues to the next frame of the video. An example of the user interface of this tool can be seen in Figure 4.1.

When evaluating the system, the ground truth can be used as a people detector or as PoI recognizer, or both.

When the ground truth is used as people detector, a bounding box is returned for every person present in the frame, and no extra false bounding boxes are returned. This is useful for evaluating the PoI recognizer, as the input will contain minimal noise, regarding size, placement and false detections.

When the ground truth is used for PoI recognition, it chooses the bounding box that is the best match of the ground truth bounding box, corresponding to the PoI. These bounding boxes are compared by (4.1), and the highest value represents the best match. The best value has to be above a certain threshold, default is set to 0.5, in order to classify as the PoI. This is useful for evaluating the people detector, regarding the ability to place the bounding boxes around the people. If the bounding box contains too much noise regarding size or placement, the PoI will not be classified as the PoI by the ground truth.

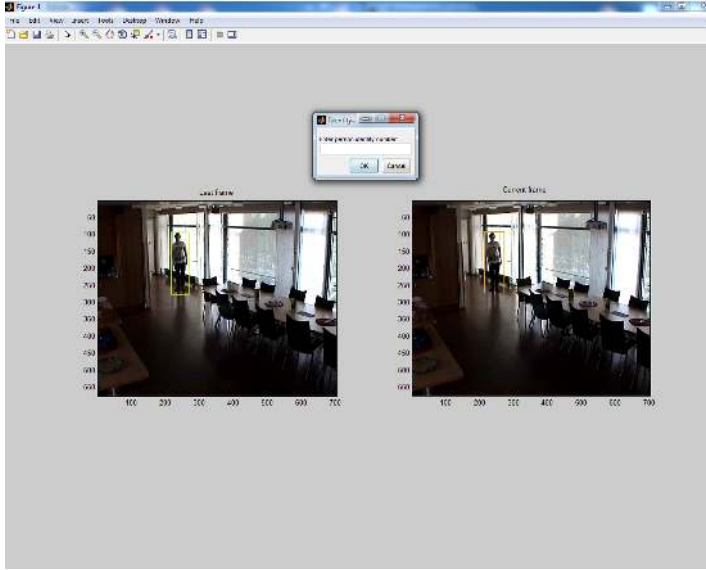


Figure 4.1: The left image shows the ground truth of the previous frame, and the right image shows the current frame, with the bounding boxes that have been marked by the user. The popup is used for entering the identification number of the last bounding box.

$$value = \frac{area(BB_{PoI} \cap BB_{Guess})}{area(BB_{PoI} \cup BB_{Guess})} \quad (4.1)$$

$BB_{PoI}$  is the ground truth bounding box corresponding to the PoI.

$BB_{Guess}$  is the bounding box which is to be tested.

## 4.2 Person of Interest Recognition Evaluation

When evaluating the PoI recognizer the performance is based on the following measurements:

### True Positive

The recognizer classifies as “PoI” and the correct label is “PoI”

### True Negative

The recognizer classifies as “Not PoI” and the correct label is “Not PoI”

### False Positive

The recognizer classifies as “PoI” and the correct label is “Not PoI”

### False Negative

The recognizer classifies as “Not PoI” and the correct label is “PoI”

**True Positive Rate, TPR**

The number of true positives divided by the number of true positives and false negatives, over a complete evaluation sequence, according to

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

$$TP = \text{Number of true positives}$$

$$FN = \text{Number of false negatives}$$
(4.2)

**False Positive Rate, FPR**

The number of false positives divided by the number of false positives and true negatives, over a complete evaluation sequence, according to

$$FPR = \frac{FP}{FP + TN}$$

$$FP = \text{Number of false positives}$$

$$TN = \text{Number of true negatives}$$
(4.3)

A common method used for visualizing these measurements is a *receiver operating characteristic*, or ROC curve. A ROC curve is generated by determining the true positive rate and false positive rate of a recognizer, using different thresholds. The TPR is then plotted against the FPR. A perfect recognizer would yield a single dot in the top left corner, where the TPR is 1.0 and FPR is 0.0. Choosing the label completely at random would yield a line through the bottom left corner,  $TPR = FPR = 0.0$ , and the top right corner,  $TPR = FPR = 1.0$  see Figure 4.2.

## 4.3 Complete System Evaluation

The evaluation of the complete system is based on the estimated 3D-positions of the PoI. The idea is to project the path into a reference camera, which is unknown to the system and observes the sequence from a different viewpoint. The projected path will then be compared to the centroids of the ground truth in the reference camera, by measuring the distance between two corresponding points. This is hereby referred to as the “backprojection error”. To calculate the backprojection error of a 3D-position, the position is projected into the reference camera. The projected point is then compared to the centroid of the corresponding bounding box, in the reference camera. The distance between these two points is the backprojection error, measured in pixels.

In order to make the measurement independent on resolution and distance to the estimated path, a normalization factor is used. The normalization factor transforms the error from pixels to world distance. The idea is to measure how far the estimated 3D-point has to be moved in order to perfectly align with the

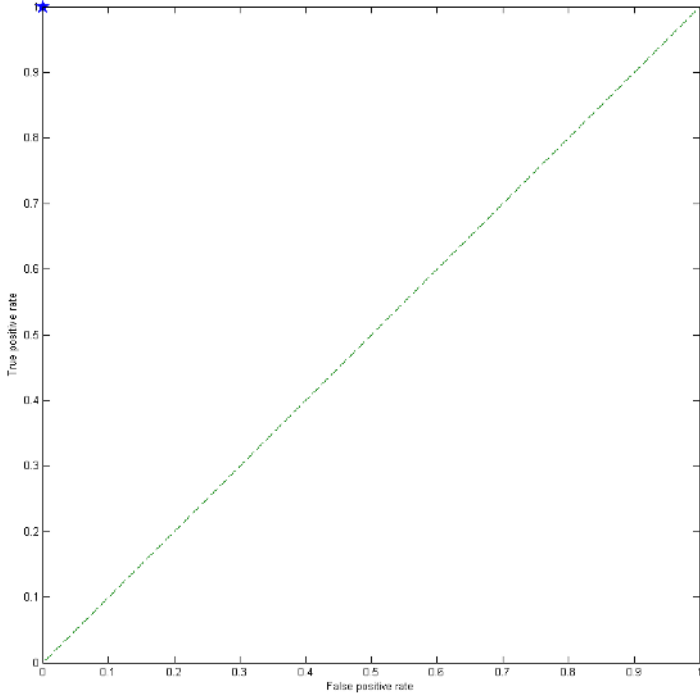


Figure 4.2: Reference results for a ROC curve, the blue star is the result of a perfect classifier, the dashed green line is the result of a completely randomized labeling.



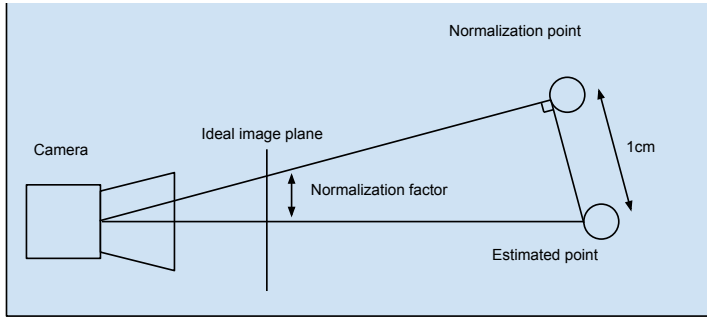


Figure 4.3: A visualization of the normalization factor calculation.

centroid of the bounding box in the reference camera. This normalization factor is determined by measuring how much a 1 cm offset of the estimated path would displace the projection, see Figure 4.3. For example, if the normalized value is 50, the closest point that would project onto the centroid of the PoI's bounding box in the reference camera, is at least 50 cm away from the estimated point.

A variation of this normalization factor is also used, which uses the estimated error of the estimated point, see Section 2.3.1, instead of 1 cm. When the error is normalized with this factor, the value describes if the estimated error was accurate or not. A value of one means that the distance that the estimated point has to be moved in order to align with the centroid of the bounding box, in the reference camera, is equal to the estimated error of that point. A value below 1 indicates that the distance is less than the estimated error, and a value above one indicates that the point has to be moved further than the estimated error. A value below or equal to one is acceptable, a lower value is better.



# Chapter 5

## Results

### 5.1 Person of Interest Recognition Results

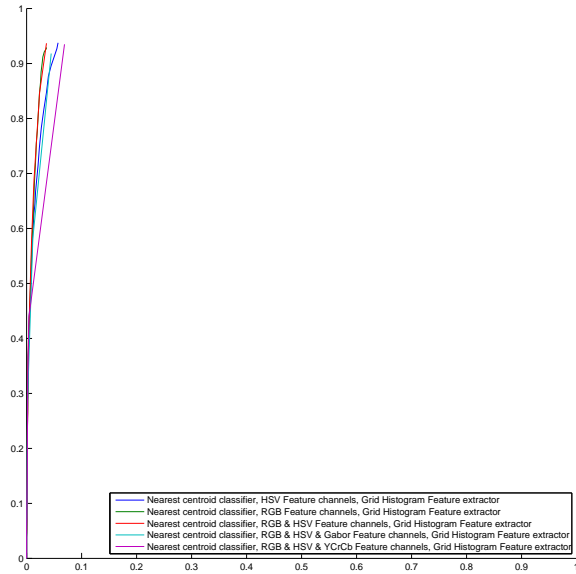
The PoI recognizer was evaluated on two videos captured at SKL, using two AXIS 214 PTZ 50HZ network cameras. The two cameras are viewing the same area, at the same time and both videos span approximately 1500 frames, at 15 frames per second. One video is used for training, and the other is used for evaluation. These videos fall into the level 2 restrictions, see Section 1.5. Both videos are in full color, there is little variation in the color representation of the two cameras, the cameras automatically compensate for different lighting and since one is facing the light, and the other is not, the mappings will differ slightly. The global positions of the cameras are known, but not the intrinsic camera parameters, however a calibration image exists. The height of the PoI is known, and the cameras observe the scene at approximately the same angle.

In the training video three persons are present, and the recognizer was evaluated three times, each time a new person was selected as the PoI. The true positives, true negatives, false positives and false negatives from the different cases were combined into one ROC curve by taking the sum for every threshold. This leads to that even if one recognizer works well for one person, it's not automatically in the top five, as it may not work well when the other persons are selected as the PoI. Because of the large number of possible combinations of classifiers, feature channels and feature extractors, the evaluation results of all combinations are not presented in this thesis. However, every combination of classifiers and channel extractors as well as feature extractors were evaluated, only one feature extractor was used per combination. In this thesis only five combinations per classifier, channel extractor and feature extractor are presented. The five combinations were chosen as the ones which came closest to a perfect classifier in the ROC curve. The distance measurement used was the Euclidian distance. The five best recognizers, according to this measurement, can be seen in Figure 5.2. The five best recognizers

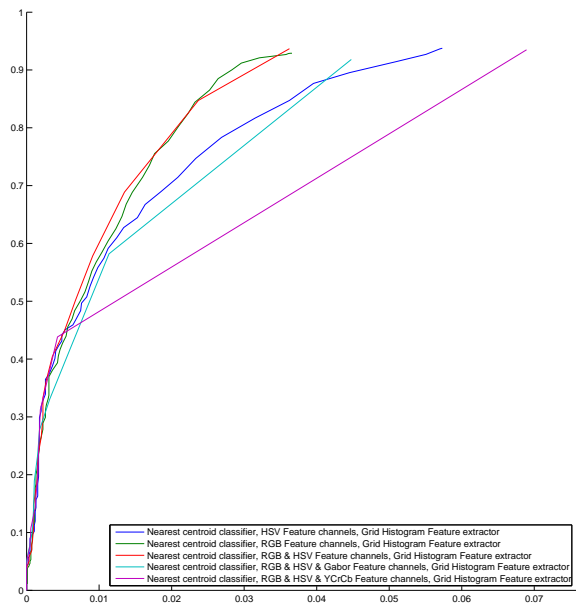
for each person can be seen in Figures 5.3, 5.4 and 5.5.



Figure 5.1: A frame from the video used for evaluation. From left to right, person one, two and three.

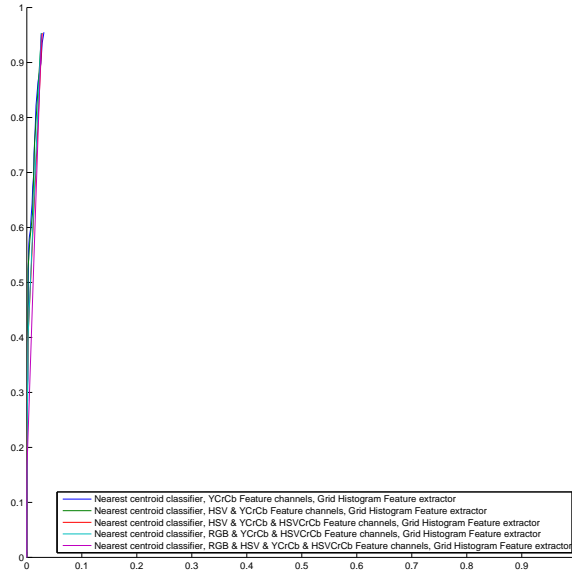


(a) Full view.

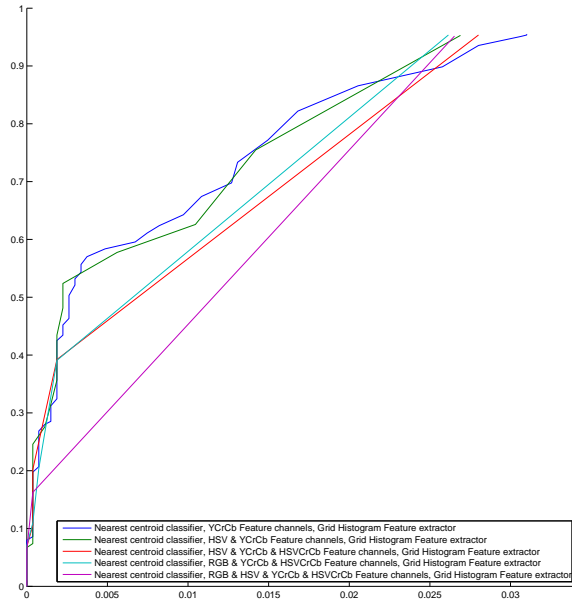


(b) Detail view, note the range.

Figure 5.2: The ROC curves of the five best PoI recognizers.

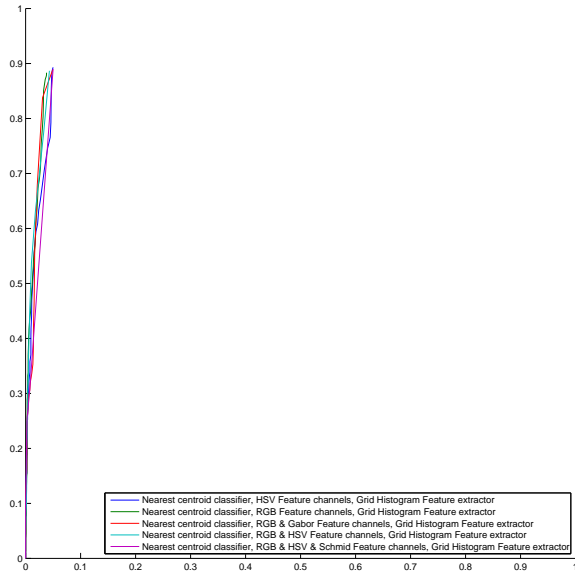


(a) Full view.

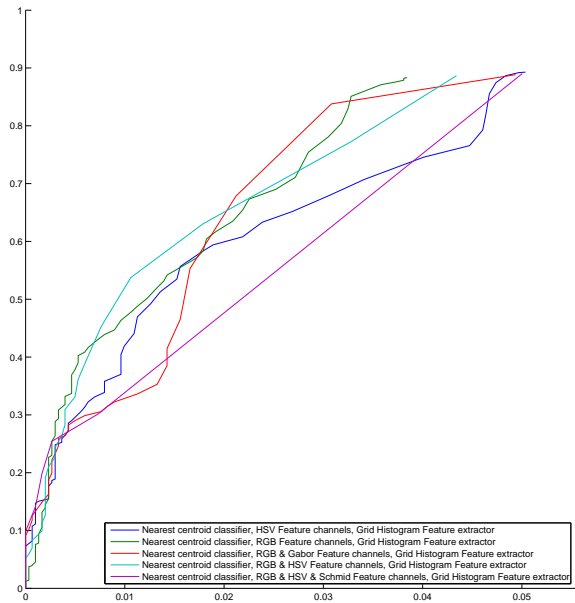


(b) Detail view, note the range.

Figure 5.3: The ROC curves of the five best PoI recognizers, when the first person was PoI.

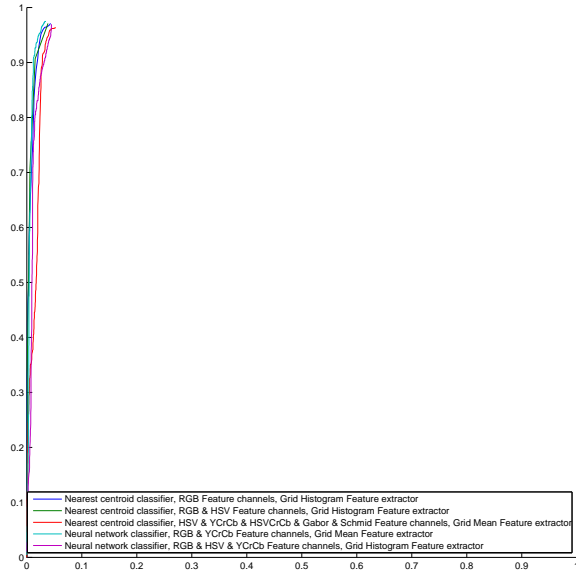


(a) Full view.

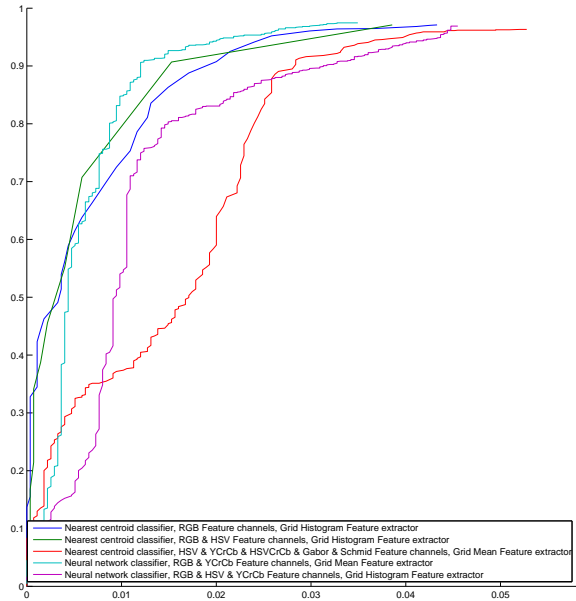


(b) Detail view, note the range.

Figure 5.4: The ROC curves of the five best PoI recognizers, when the second person was PoI.



(a) Full view.



(b) Detail view, note the range.

Figure 5.5: The ROC curves of the five best PoI recognizers, when the third person was PoI.

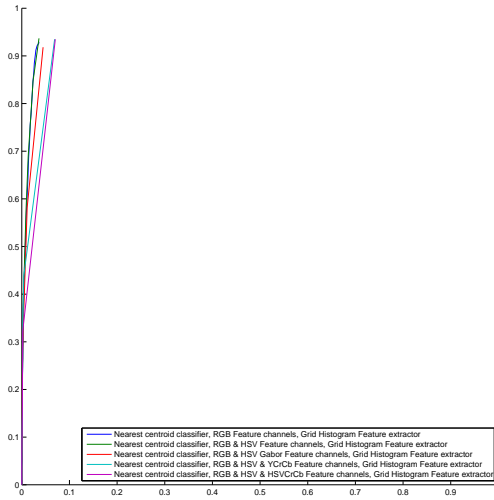


**5.1.1 Feature Channels**

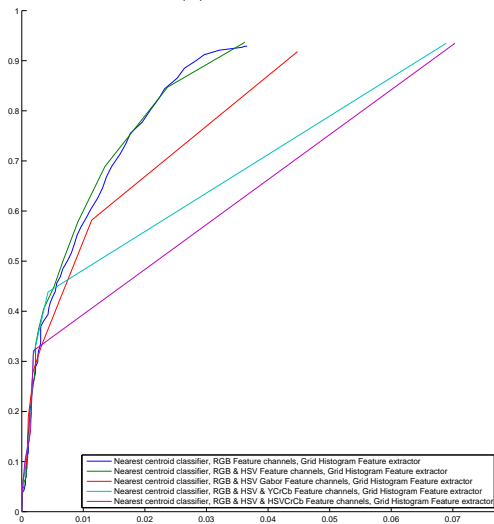
Following are the five best combinations including specific feature channels.

## RGB

The five best recognizers which contain the RGB feature channels can be seen in Figure 5.6.



(a) Full view.

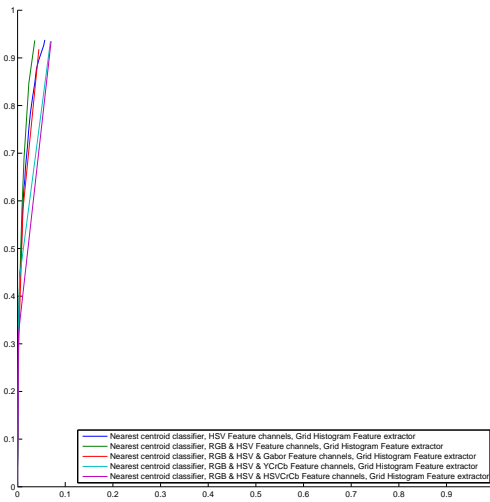


(b) Detail view, note the range.

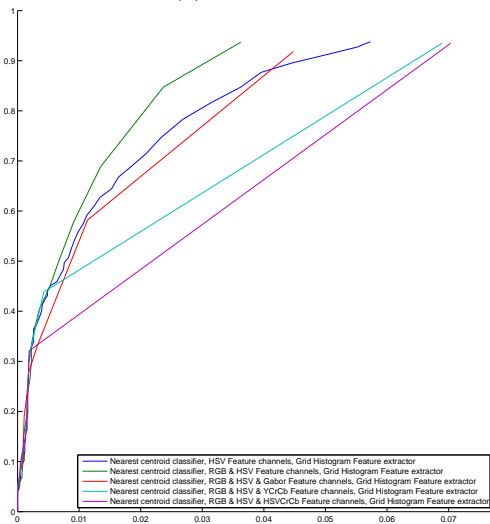
Figure 5.6: The ROC curves of the five best PoI recognizers, which contain the RGB feature channels.

### HSV

The five best recognizers which contain the HSV feature channels can be seen in Figure 5.7.



(a) Full view.

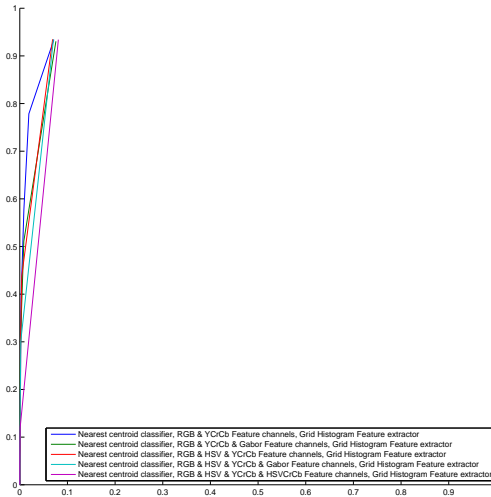


(b) Detail view, note the range.

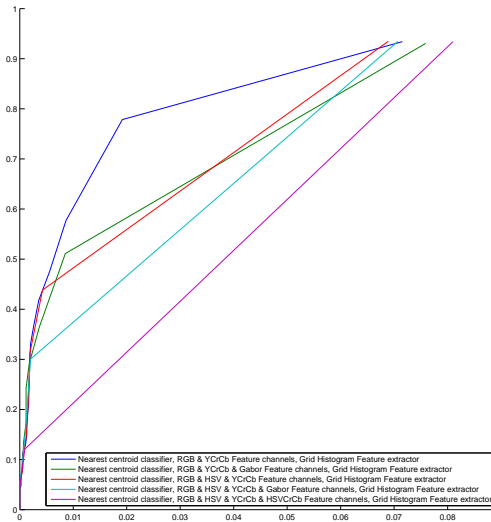
Figure 5.7: The ROC curves of the five best PoI recognizers, which contain the HSV feature channels.

### YCrCb

The Figure 5.8 shows the five best recognizers which contain the YCrCb feature channels.



(a) Full view.

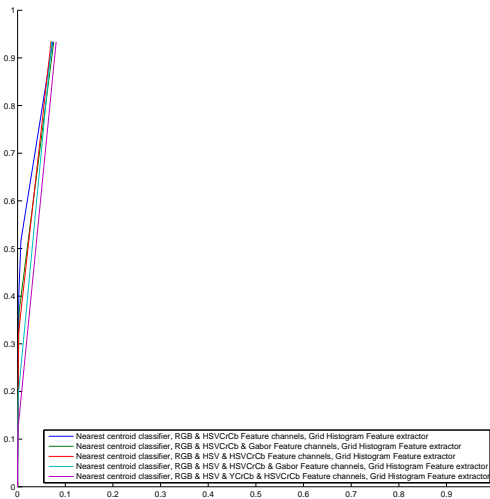


(b) Detail view, note the range.

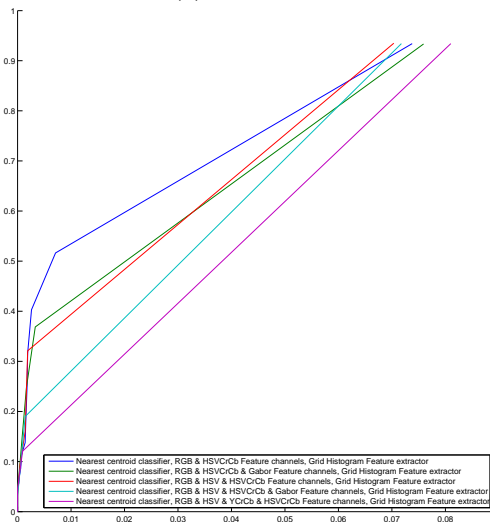
Figure 5.8: The ROC curves of the five best PoI recognizers, which contain the YCrCb feature channels.

### HSVCrCb

The Figure 5.9 shows the five best recognizers which contain the HSVCrCb feature channels.



(a) Full view.

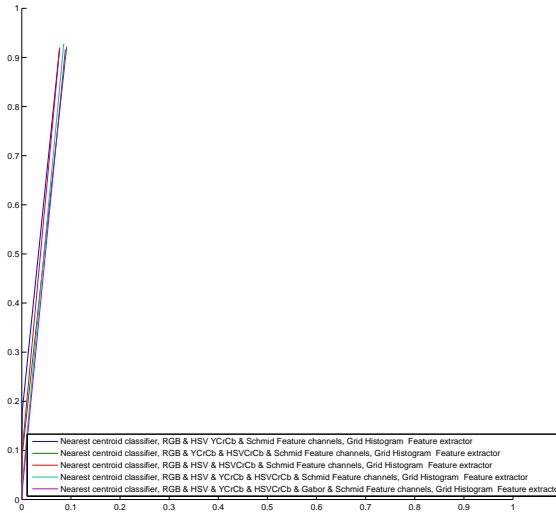


(b) Detail view, note the range.

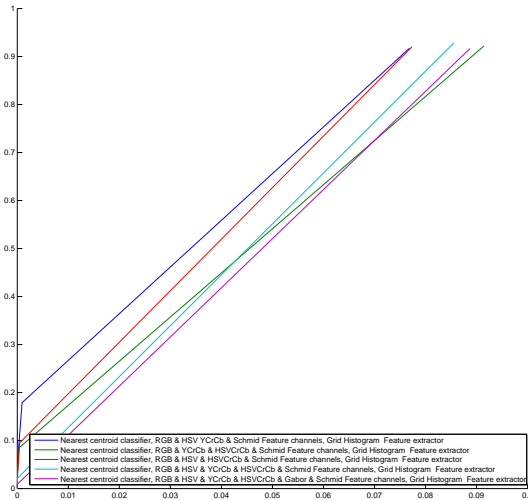
Figure 5.9: The ROC curves of the five best PoI recognizers, which contain the HSVCrCb feature channels.

### Schmid

In Figure 5.10 the five best recognizers which utilize the Schmid feature channels can be seen.



(a) Full view.

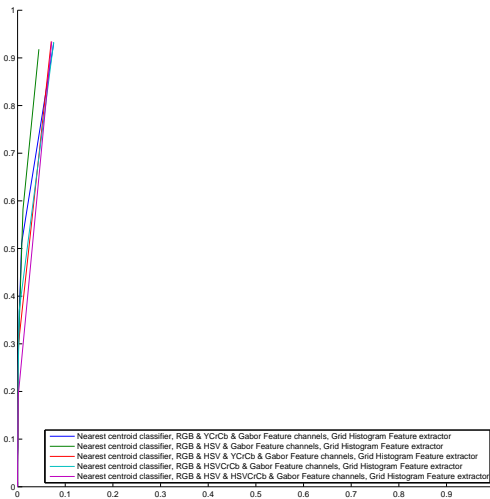


(b) Detail view, note the range.

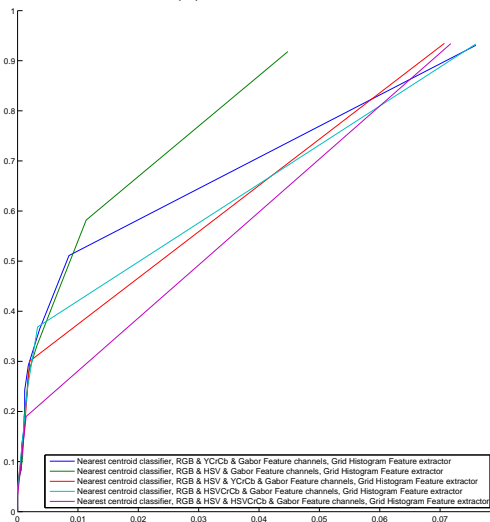
Figure 5.10: The ROC curves of the five best PoI recognizers, which contain the Schmid feature channels.

### Gabor

In Figure 5.11 the five best recognizers which utilize the Gabor feature channels can be seen.



(a) Full view.



(b) Detail view, note the range.

Figure 5.11: The ROC curves of the five best PoI recognizers, which contain the Gabor feature channels.

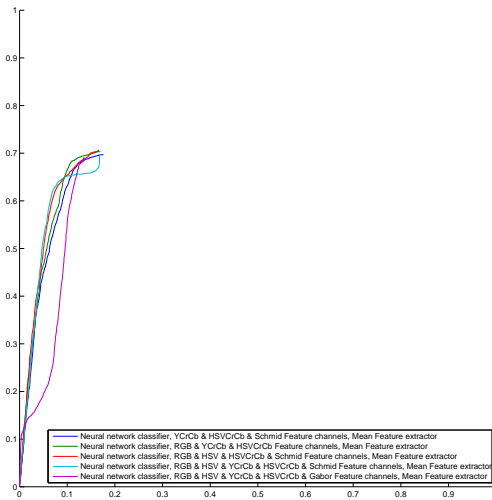
### 5.1.2 Feature Extractors

Here the five best combinations including specific feature extractors is presented. When the grid is used, it is composed of four rows and three columns.

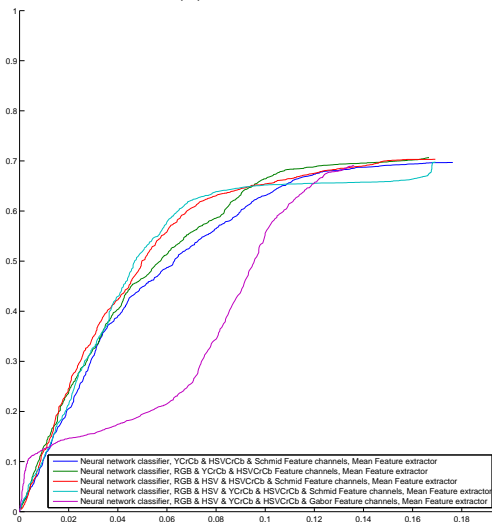


Mean

The results of the five best recognizers which uses mean as feature extractor can be seen in Figure 5.12.



(a) Full view.

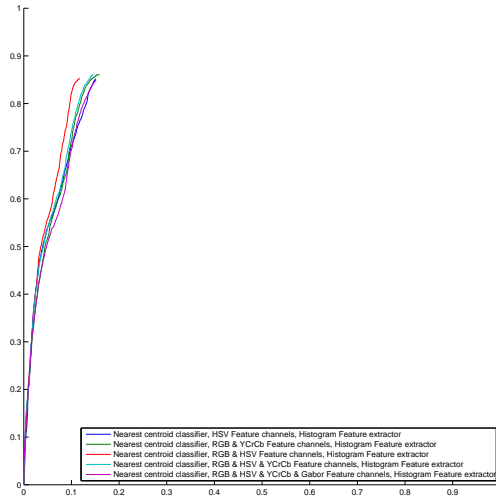


(b) Detail view, note the range.

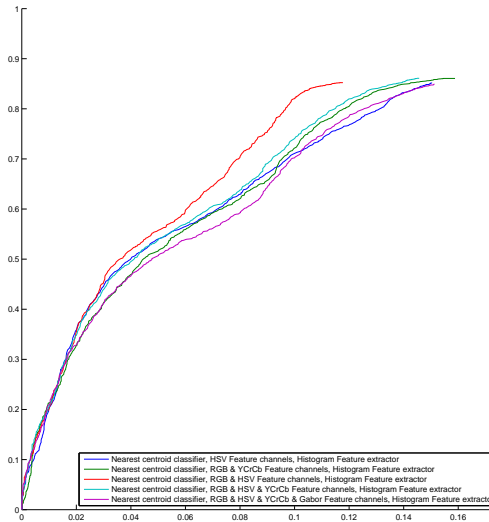
Figure 5.12: The ROC curves of the five best PoI recognizers, which contain the mean feature extractor.

## Histogram

The results of the five best recognizers which uses histogram as feature extractor can be seen in Figure 5.13.



(a) Full view.

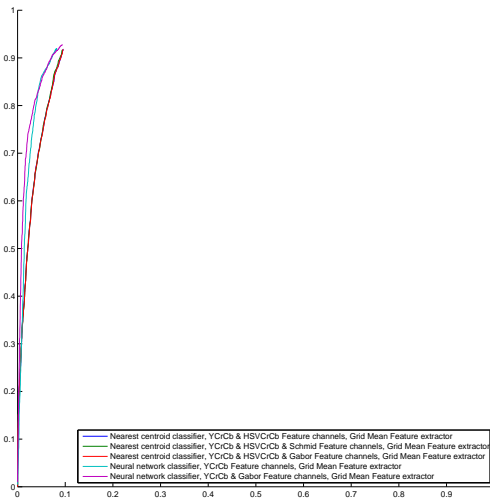


(b) Detail view, note the range.

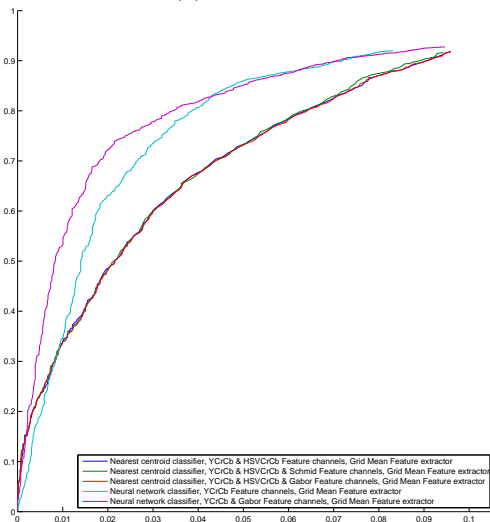
Figure 5.13: The ROC curves of the five best PoI recognizers, which contain the histogram feature extractor.

### Grid with Mean

The results of the five best recognizers which uses mean in the cells of a grid as feature extractor can be seen in figure 5.14.



(a) Full view.

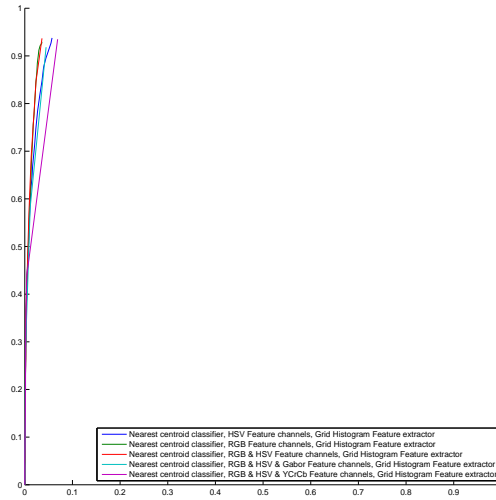


(b) Detail view, note the range.

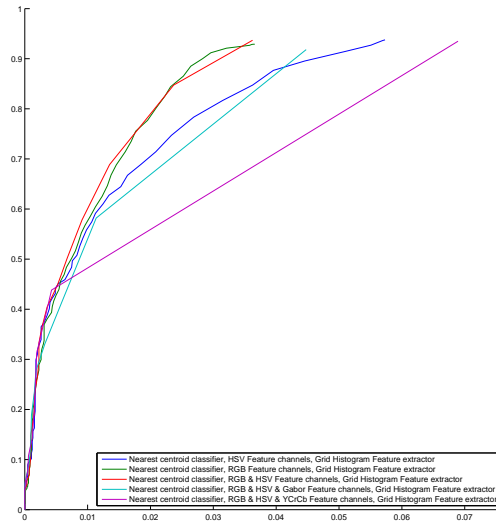
Figure 5.14: The ROC curves of the five best PoI recognizers, which contain the feature extractor grid and mean.

### Grid with Histogram

The results of the five best recognizers which uses histogram in the cells of a grid as feature extractor can be seen in figure 5.15.



(a) Full view.



(b) Detail view, note the range.

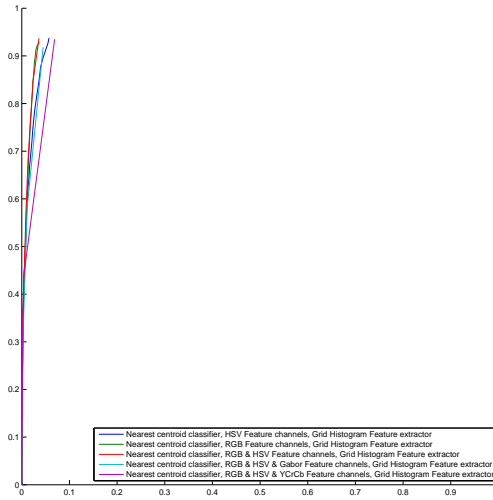
Figure 5.15: The ROC curves of the five best PoI recognizers, which contain the feature extractor grid and histogram.

**5.1.3 Classifiers**

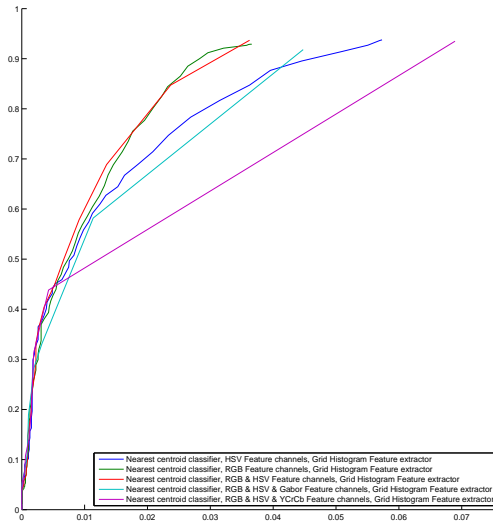
In this section, the five best combinations of each of the classifiers is presented.

### Nearest Centroid

The results of the five best recognizers which uses nearest centroid as classifier can be seen in Figure 5.16.



(a) Full view.

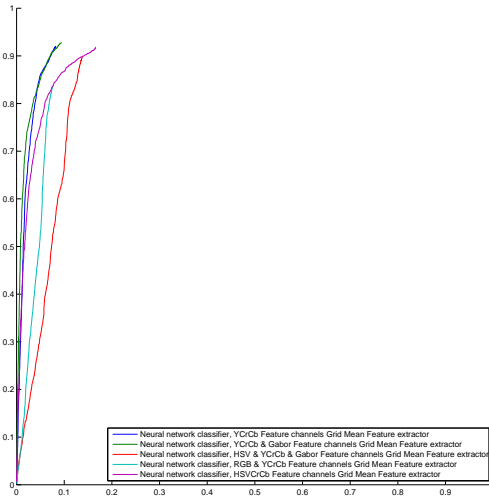


(b) Detail view, note the range.

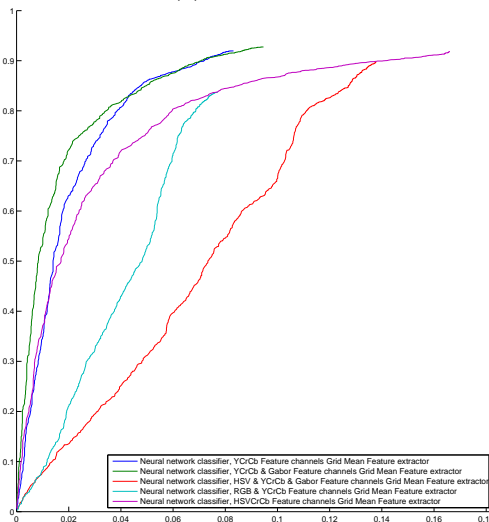
Figure 5.16: The ROC curves of the five best PoI recognizers, which uses nearest centroid as classifier.

### Neural Network

The results of the five best recognizers which uses neural network as classifier can be seen in Figure 5.17.



(a) Full view.



(b) Detail view, note the range.

Figure 5.17: The ROC curves of the five best PoI recognizers, which uses neural network as classifier.

## 5.2 Complete System Results

The system was evaluated on four different cases, listed below.

1. Ground truth is used for detecting people and recognizing PoI.
2. Ground truth is used for detecting people but is not used for recognizing PoI.
3. Ground truth is not used for detecting people but is used for recognizing PoI.
4. Ground truth is not used either for detecting people or recognizing the PoI.

The same two videos was used for all four cases. The two videos are the same as used when evaluating the PoI recognizer, see Section 5.1.

### 5.2.1 Evaluation with Ground Truth

Figures 5.18, 5.19 and 5.20 shows the back projection error of the unfiltered 3D-position of the PoI.

Figures 5.21, 5.22 and 5.23 shows the back projection error of the filtered, see Section 2.4, 3D-position of the PoI.

In Figures 5.19 and 5.22 the back projection error is normalized with the unit sphere according to Section 4.3. In the Figures 5.20 and 5.23 it is normalized in the same way, but with the estimated error sphere instead of the unit sphere.

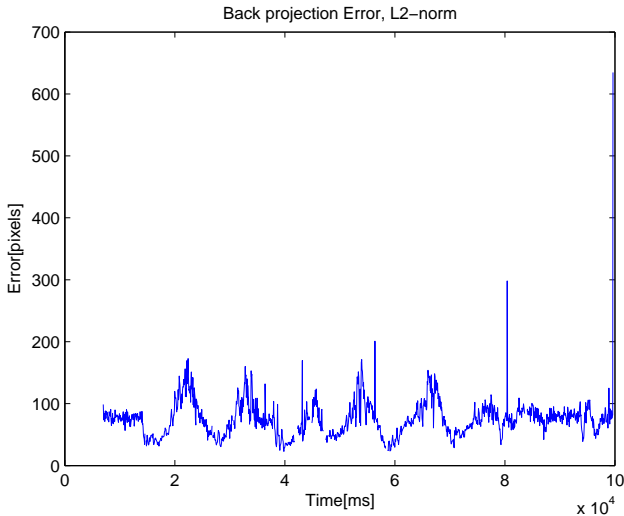


Figure 5.18: The back projection error of the unfiltered estimated 3D-position of the PoI.



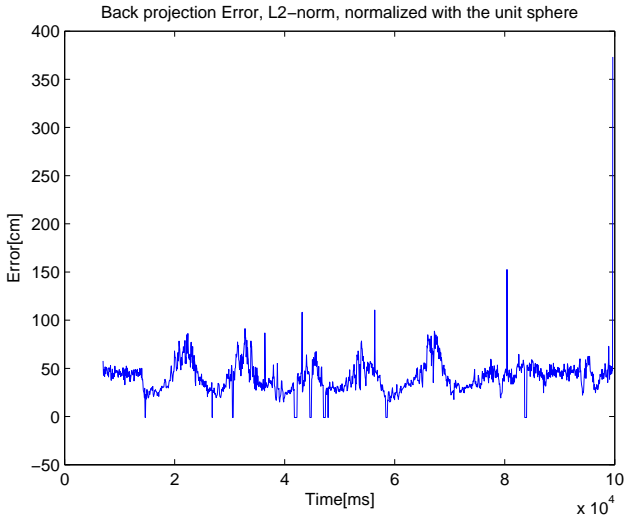


Figure 5.19: The back projection error of the unfiltered estimated 3D-position of the PoI, normalized with the unit sphere.

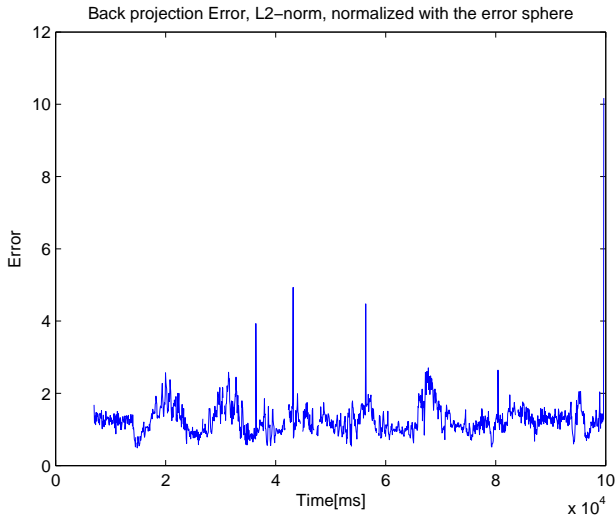


Figure 5.20: The back projection error of the unfiltered estimated 3D-position of the PoI, normalized with error sphere.

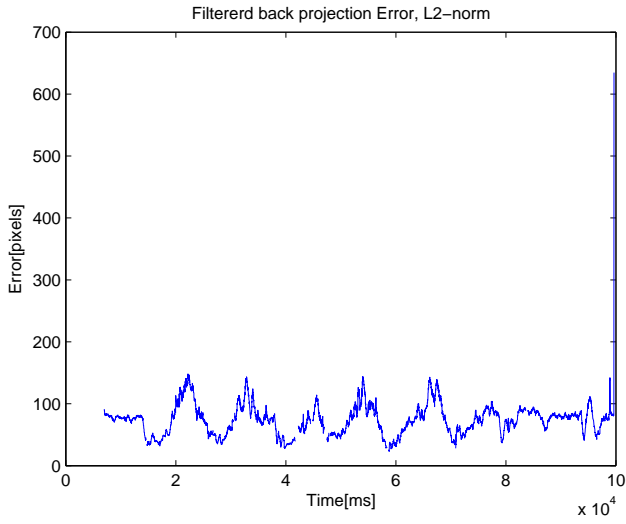


Figure 5.21: The back projection error of the filtered estimated 3D-position of the PoI.

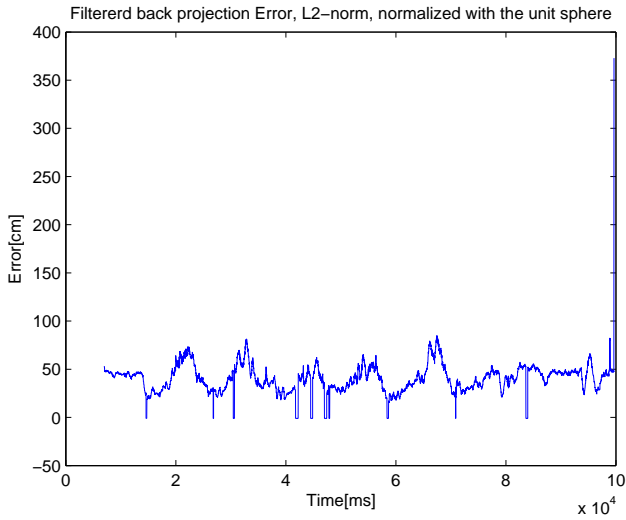


Figure 5.22: The back projection error of the filtered estimated 3D-position of the PoI, normalized with the unit sphere.

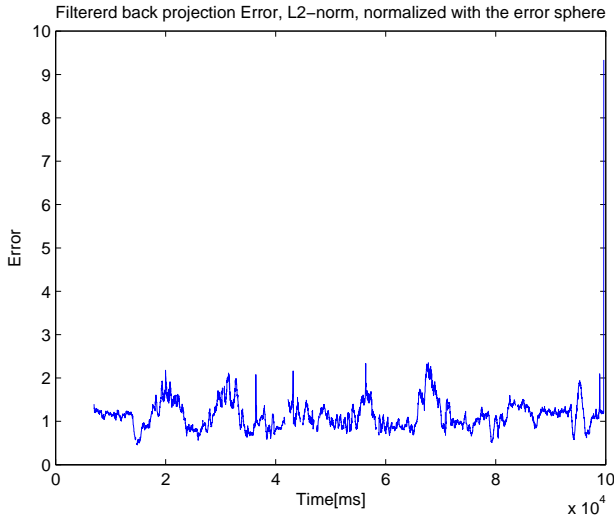


Figure 5.23: The back projection error of the filtered estimated 3D-position of the PoI, normalized with error sphere.

## 5.2.2 Evaluation with Ground Truth used for Detecting People

The PoI recognizer which yielded these results is based on a nearest centroid classifier, see section 2.2.3, RGB and HSV feature channels, see Section 2.2.1 and grid histograms, Section 2.2.2, as feature extractor. The grid had four rows and three columns, and the histogram had ten bins per feature channel. The PoI recognizer was trained on the reference video, not on the video that it was tested on.

Figures 5.24, 5.25 and 5.26 shows the back projection error of the unfiltered 3D-position of the PoI.

Figures 5.27, 5.28 and 5.29 shows the back projection error of the filtered, see Section 2.4, 3D-position of the PoI.

In Figures 5.25 and 5.28 the back projection error is normalized with the unit sphere according to Section 4.3. In Figures 5.26 and 5.29 it is normalized in the same way, but with the estimated error sphere instead of the unit sphere.

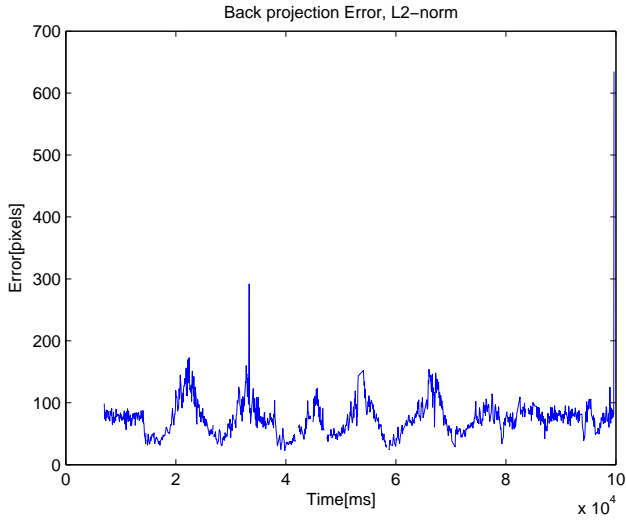


Figure 5.24: The back projection error of the unfiltered estimated 3D-position of the PoI.

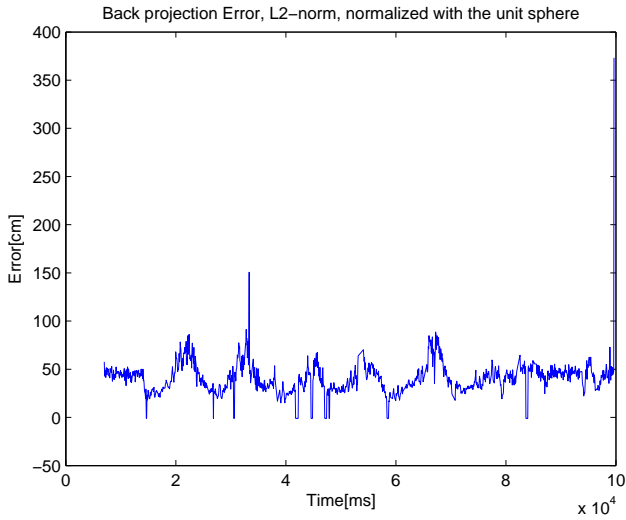


Figure 5.25: The back projection error of the unfiltered estimated 3D-position of the PoI, normalized with the unit sphere.

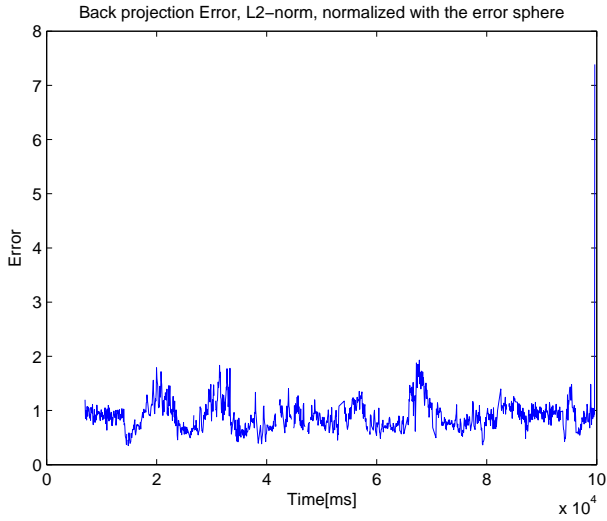


Figure 5.26: The back projection error of the unfiltered estimated 3D-position of the PoI, normalized with error sphere.

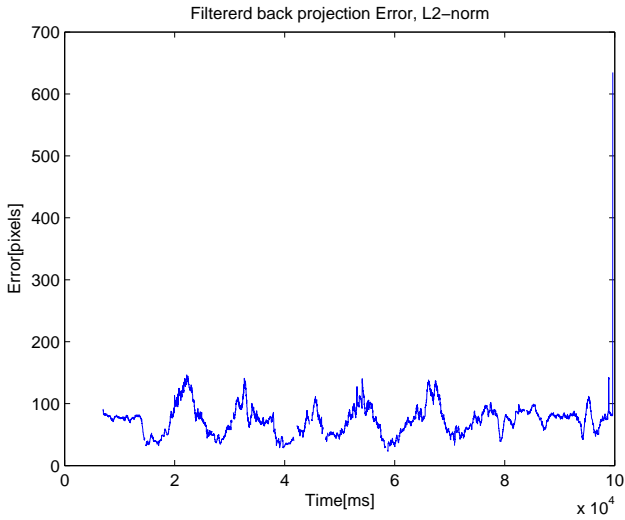


Figure 5.27: The back projection error of the filtered estimated 3D-position of the PoI.

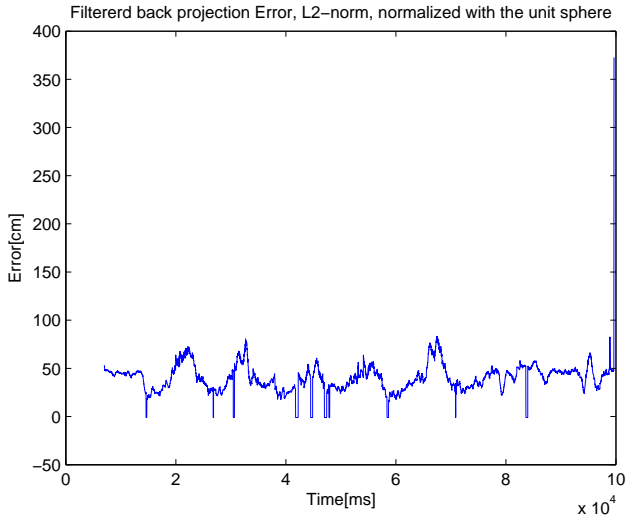


Figure 5.28: The back projection error of the filtered estimated 3D-position of the PoI, normalized with the unit sphere.

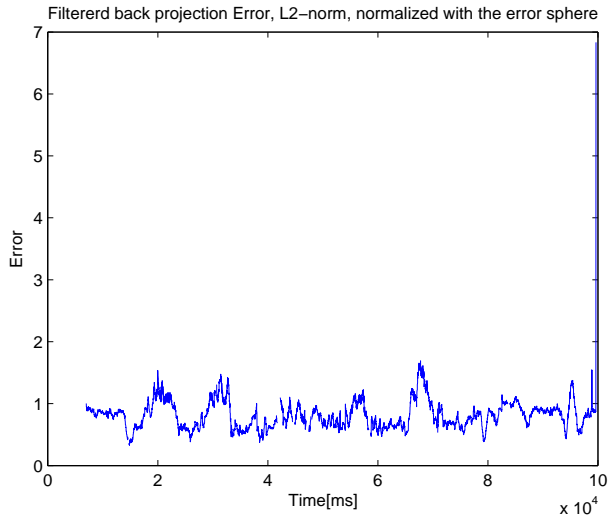


Figure 5.29: The back projection error of the filtered estimated 3D-position of the PoI, normalized with error sphere.

### 5.2.3 Evaluation with Ground Truth used for Recognizing the PoI

The following results are based on the people detector discussed in Section 2.1, implemented by Johannes Markström. For a more detailed description, see [13].

Figures 5.24, 5.31 and 5.32 shows the back projection error of the unfiltered 3D-position of the PoI.

Figures 5.33, 5.34 and 5.35 shows the back projection error of the filtered, see Section 2.4, 3D-position of the PoI.

In Figures 5.31 and 5.34 the back projection error is normalized with the unit sphere according to Section 4.3. In Figures 5.32 and 5.35 it is normalized in the same way, but with the estimated error sphere instead of the unit sphere.

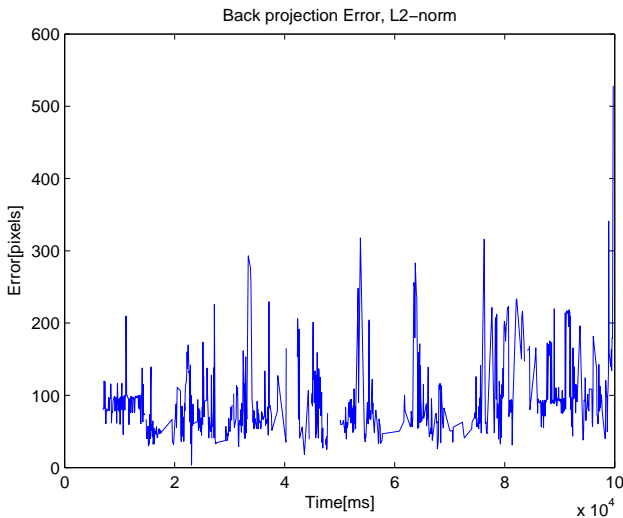


Figure 5.30: The back projection error of the unfiltered estimated 3D-position of the PoI.

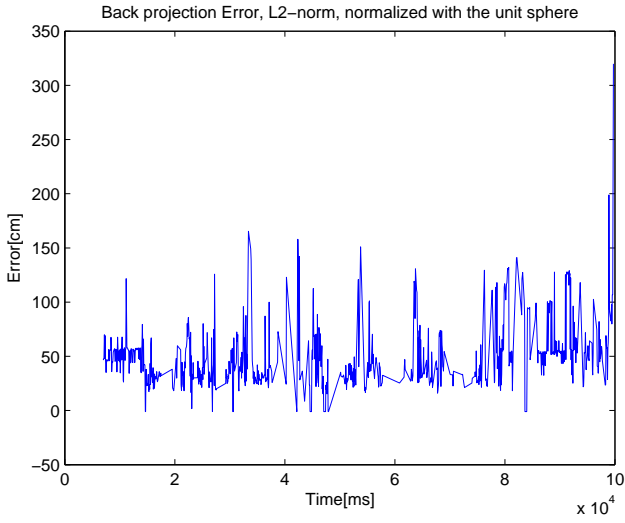


Figure 5.31: The back projection error of the unfiltered estimated 3D-position of the PoI, normalized with the unit sphere.

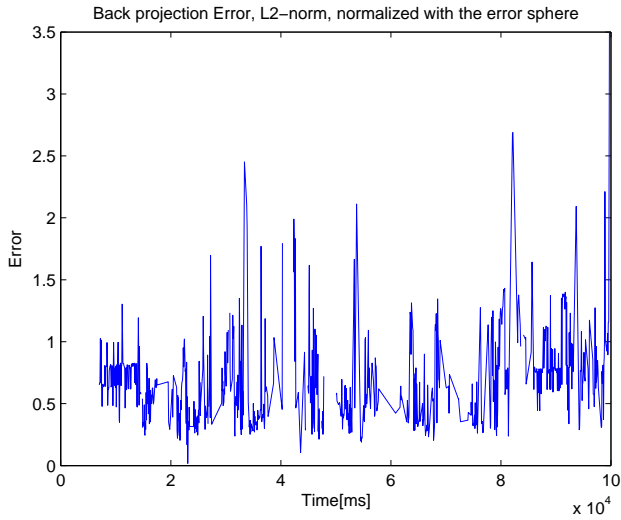


Figure 5.32: The back projection error of the unfiltered estimated 3D-position of the PoI, normalized with error sphere.



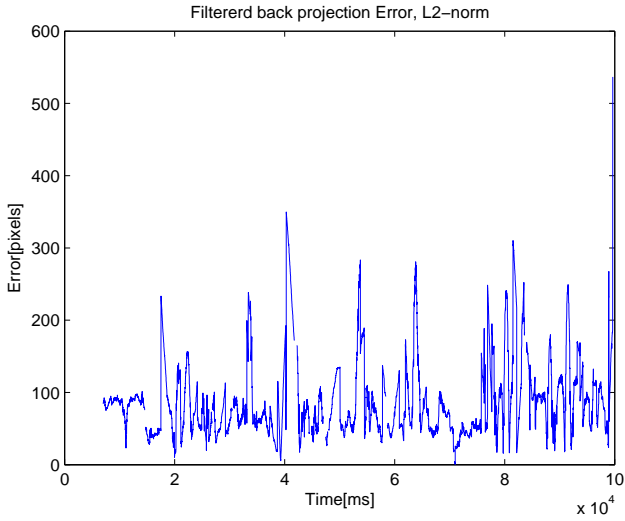


Figure 5.33: The back projection error of the filtered estimated 3D-position of the PoI.

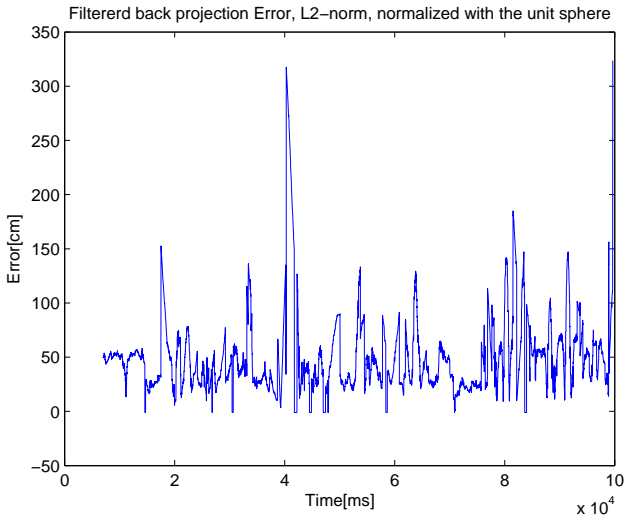


Figure 5.34: The back projection error of the filtered estimated 3D-position of the PoI, normalized with the unit sphere.

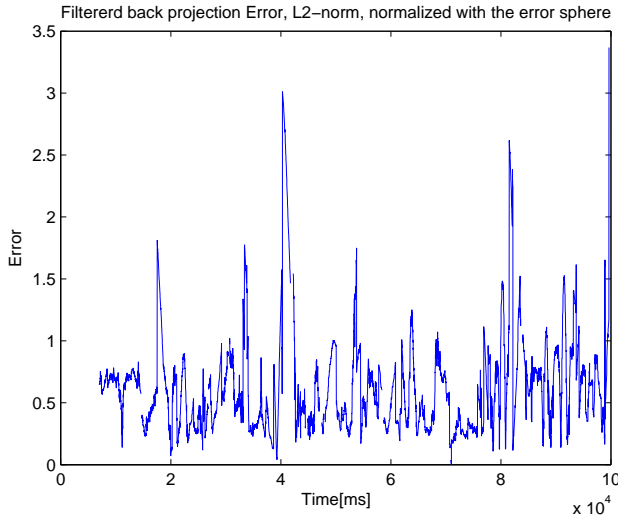


Figure 5.35: The back projection error of the filtered estimated 3D-position of the PoI, normalized with error sphere.

## 5.2.4 Evaluation without Ground Truth

The PoI recognizer which yielded these results is based on a nearest centroid classifier, see Section 2.2.3, RGB and HSV feature channels, see Section 2.2.1 and grid histograms, Section 2.2.2, as feature extractor. The grid had four rows and three columns, and the histogram had ten bins per feature channel. The PoI recognizer was trained on the reference video, not on the video that it was tested on. When training the PoI recognizer, the people detector was used to detect the people, and the ground truth was used to classify these to create the training data. The people detector used is discussed in Section 2.1, implemented by Johannes Markström. For a more detailed description, see [13].

Figures 5.36, 5.37 and 5.38 shows the back projection error of the unfiltered 3D-position of the PoI.

Figures 5.39, 5.40 and 5.41 shows the back projection error of the filtered, see Section 2.4, 3D-position of the PoI.

In Figures 5.37 and 5.40 the back projection error is normalized with the unit sphere according to Section 4.3. In Figures 5.38 and 5.41 it is normalized in the same way, but with the estimated error sphere instead of the unit sphere.

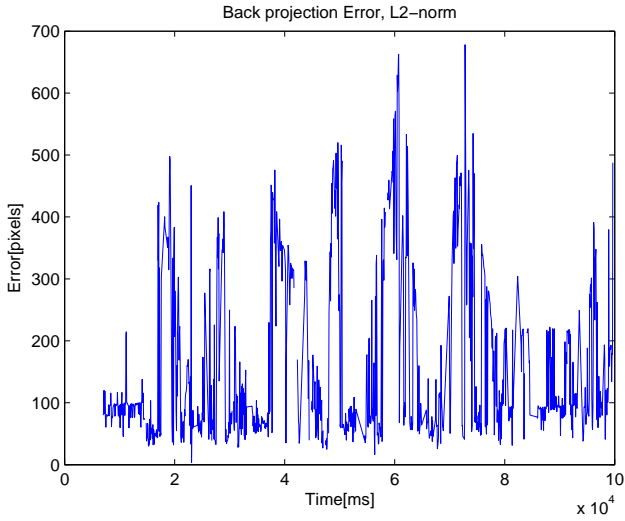


Figure 5.36: The back projection error of the unfiltered estimated 3D-position of the PoI.

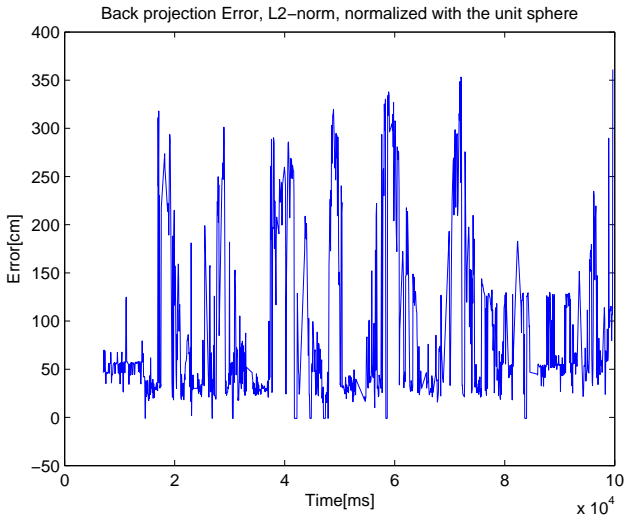


Figure 5.37: The back projection error of the unfiltered estimated 3D-position of the PoI, normalized with the unit sphere.

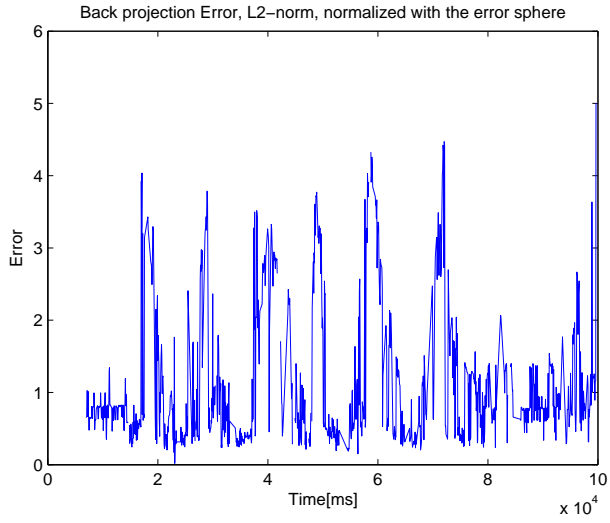


Figure 5.38: The back projection error of the unfiltered estimated 3D-position of the PoI, normalized with error sphere.

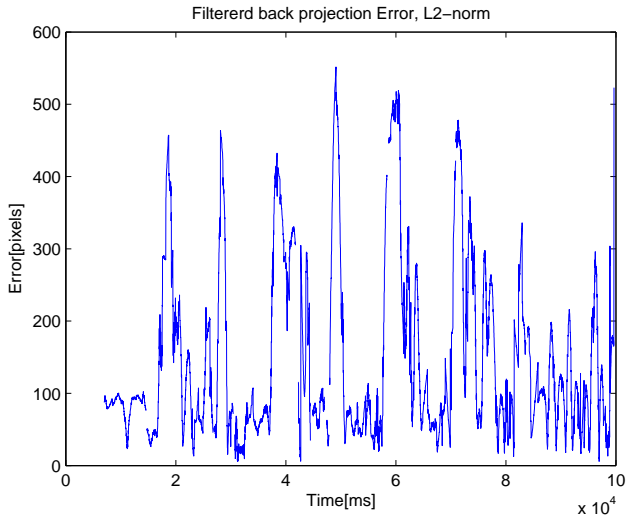


Figure 5.39: The back projection error of the filtered estimated 3D-position of the PoI.

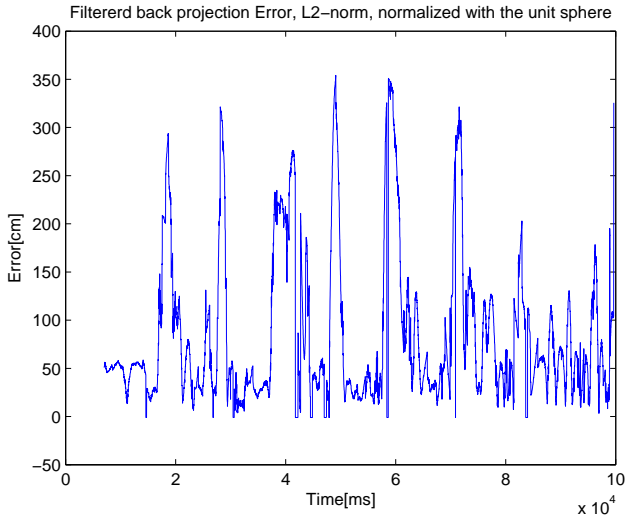


Figure 5.40: The back projection error of the filtered estimated 3D-position of the PoI, normalized with the unit sphere.

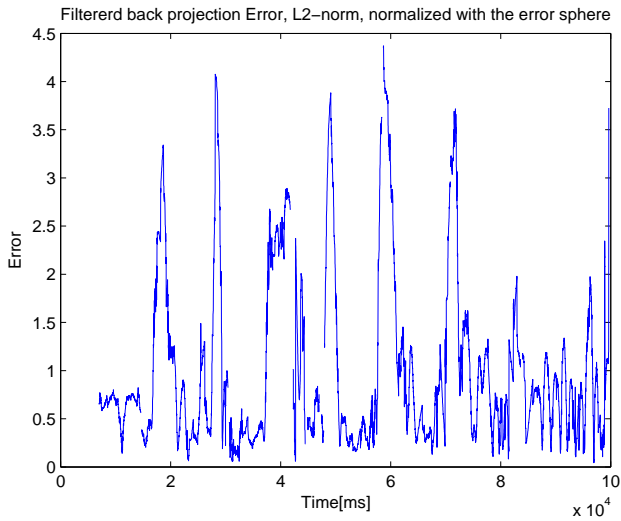


Figure 5.41: The back projection error of the filtered estimated 3D-position of the PoI, normalized with error sphere.



# Chapter 6

## Discussion

This chapter presents an interpretation of the results, and discusses the reasons behind them.

### 6.1 Person of Interest Recognition Discussion

When examining the results in Section 5.1 it is apparent that different feature channels may be of different importance for different people, just like [12] suggested. When the first person is selected as PoI the YCrCb feature channels are present in all of the five best recognizers. When the second person is selected PoI YCrCb does not appear once among the best five. What is also interesting is that when the third person is selected PoI, the nearest centroid classifier is not used in the best recognizer. This is interesting since the nearest centroid, and grid histogram, has been used in every top five recognizer when the other persons are selected as PoI. This is something that [12] does not explore; that the optimal choice of classifier and feature extractor, may vary from person to person.

From the results in Section 5.1 a couple of interesting observations can be made. Firstly the RGB and HSV feature channels are present in close to every “best five” combination. This is a strong indication that the RGB and HSV feature channels hold the most relevant information, in the setup used to create the result. One exception from this is when the feature extractor is fixed to be the grid mean feature extractor. In that case the YCrCb feature channel is overrepresented. Another interesting observation regarding the grid mean feature extractor is that it is the only one where the best five recognizers include both nearest centroid and neural network classifiers. The only other “best five” combination to include the neural network classifier is when the feature extractor is fixed to mean. One possible explanation to this is that the mean, and grid mean, feature extractors yield relatively small feature vectors, one and twelve features per channel respectively. Neural networks are not a great choice for classifier when the dimensionality of

the problem is large, but if the dimensionality is smaller it may find nonlinear correlations between the classes, which the nearest centroid cannot find. In this training data however the nearest centroid classifier is superior, and the five overall best recognizers uses the nearest centroid as classifier. The grid histogram feature extractor is also overrepresented in the results, and the only case where it is not used in the “five best” is when the classifier is fixed as neural network. Another notable occurrence is when both HSV and HSVCrCb or YCrCb and HSVCrCb are used together. This occurs for instance when grid mean, see Figure 5.14, is fixed as feature extractor, and when the recognizers containing Schmid feature channels are considered, HSV, YCrCb and HSVCrCb are used together in the best five, see Figure 5.10. A probable cause for this is that this gives these channels a higher influence, as they appear twice. This indicates that a weighting function over the feature channels may improve the results, but that is left as a future improvement.

All these results should be interpreted carefully since the testing environment is very limited, three people in the same room at the same time of day does not produce a lot of variation in the testing data. As such these results should not be considered truth, but rather an implication of what may work and what may not.

## 6.2 Complete System Discussion

The first thing to notice when examining the results of the system evaluation is that, even in the case when ground truth is used both for detecting people and recognizing the PoI, the back projection error is never zero. There are several reasons for this, firstly the ground truth is marked by people, so there are flaws, some bounding boxes may be a little bit too big and other too small, this moves the estimated position slightly. Secondly, the measurements of the testing environment were made by hand and contain errors. Thirdly, the time synchronization of the two cameras is not perfect, there is about a half second delay between them. All of these errors are unknown to the system and can therefore not be compensated for.

The results that contain the best representation of how well the system works, are the filtered and normalized results. These represent how the path is calculated when the system is actually used, and the normalization gives a measurement which is easier to interpret than the unnormalized version.

When the projection is normalized with a unit sphere, as described in Section 4.3, the value describes how far away the point that would project perfectly onto the reference camera is from the estimated point, in cm.

The other normalization is based on the estimated error, and describe how far from the worst case estimation the actual point is. This is useful for checking the error estimation, the value should always be below one if our worst case error



estimation is correct. This is not the case, as can be seen in Figures 5.23 and 5.20. The reason for this is that the error estimate does not take into account false classifications and time differences. It estimates the difference between the correct position of a person compared to the estimated position of the same person, in that exact moment.

When evaluating the complete system with an actual PoI recognizer, rather than the ground truth, it is hard to spot any increase in error. In fact the number of spikes in the error has even been reduced. This is because the ground truth, as mentioned above, contains errors. The actual recognizer does not make the same false classifications, and the spikes disappear. The remaining spikes are probably the result of errors in the ground truth of the reference video, and does not necessarily indicate erroneous classifications by the recognizer. This indicates that the chosen recognizer is very accurate when recognizing the PoI, however the video is a very simple one, with few occlusions, high resolution, and high variance between the color of the people present in the video. Nevertheless the implemented recognizer does fulfill the most basic purpose of this thesis, which is to implement a recognizer that could be used to separate the specified PoI from a set of people. If the normalized back projection error is compared between the filtered, Figure 5.28, and the unfiltered version, Figure 5.25, the importance of the filtering can be seen. The spikes are gone and the error does not vary as much.

When the PoI recognizer is paired with the people detector implemented by Johannes Markström [13] the results are less impressive. The width of the spikes in the errors indicate that the recognizer classifies the wrong person as PoI over multiple frames. This is a bigger problem than multiple single frame misclassifications, as the filtering will not remove these. The reason for the decrease in performance is probably caused by the variance of the bounding boxes provided by the detector. If the bounding box is displaced or resized enough the cells in the grid used by the feature extractor may move from one part of the body to another. For instance may the cell which usually contain the head of the person, align with the torso instead, if the head is not confined in the bounding box. Since the 3D-position estimation is highly dependent on good bounding boxes, the system fails to return relevant estimations, when that is not the case. In the error normalized results, Figure 5.28, and the unfiltered version, Figure 5.25, it is apparent that the error is much greater than anticipated. The reason for this is that when estimating the error, an assumption was made about the precision of the bounding boxes returned from the people detector, which was much too low. One way to decrease the error from this would be to try and recalculate the sizes of the bounding boxes after the people have been detected. The top of the bounding box could be moved up and down until an asymmetry line was found, which may be the top of the head. The bottom of the bounding box could also be moved this way, assuming that the feet of the person yields an asymmetry line as well.

## 6.3 Future Work

Due to the size and complexity of the system there is still a lot of improvements to be done. One thing that could improve the precision of the PoI recognition is a tracker. If every person in a video was tracked, multiple instances of the same person could be classified at the same time. One could then use the accumulated dissimilarity to decide if a person is the PoI or not. This could probably improve the recognition to be more stable to occlusions, if the tracker could keep track of the person before and after the occlusion. Another possible improvement could be the inclusion of a background model, this would minimize the noise in the training data, and possibly improve the classification. One way to reduce the human interactions of this program, which was a goal of this project, is to let the system choose what feature channels, classifier and feature extractors should be used. This could be done by training a large number of different recognizers on the training data and determine which combination is the most efficient for that specific PoI.

# Chapter 7

## Conclusion

In conclusion, we have provided a stable base which may detect people, recognize a specific person of interest, and estimate the 3D-path of that person. The task at hand was a very challenging one, and there is room for a lot of improvement in both the people detector and person of interest recognizer as well as the 3D-position estimation. However the system that we have built is made for expansion with both modifications and improvements easy to implement.

In this project we have shown the possibilities and obstacles when trying to estimate a 3D-path of a specific person using only monocular, non-overlapping cameras, similar to those used for surveillance. The possibilities increase as the technology progresses, and I suspect that these sort of systems are not too far into the future, but still in the future.

I feel that the goal of this project has been achieved, a functioning base system has been implemented, with the ability to detect people, recognizing a person of interest, and estimating the 3D-path of that person, with the limitations described in Section 1.5. A graphical user interface from which the entire project can be controlled, was also created.



# Bibliography

- [1] Tamar Avraham, Ilya Gurvich, Michael Lindenbaum, and Shaul Markovitch. Learning Implicit Transfer for Person Re-identification. In *ECCV 2012*, 2012.
- [2] Loris Bazzani, Marco Cristani, Michela Farenzena, and Vittorio Murino. Multiple-shot Person Re-identification by HPE signature. In *International Conference on Pattern Recognition*, 2010.
- [3] Jean-Yves Bouguet. MATLAB calibration tool. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/). Updated: 2010-09-07.
- [4] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person Re-Identification by Symmetry-Driven Accumulation of Local Features. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, 2010.
- [5] L. Fogel and D. Sagi. Gabor filters as texture discriminator. In *Biological Cybernetics*, volume 61, pages 103–113. Springer-Verlag, June 1989.
- [6] Douglas Gray and Hai Tao. Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features. In *ECCV 2008*, 2008.
- [7] Eric Hamilton. JPEG File Interchange Format. Technical report, C-Cube Microsystems, 1992.
- [8] M. Iktis and M. Magallon. GLEW. <http://glew.sourceforge.net>. Accessed: 2013-01-22.
- [9] itseez. OpenCV. <http://opencv.org>. Accessed: 2013-01-22.
- [10] Kalman, Rudolph Emil et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1).
- [11] Ryan Layne, Timothy M. Hospedales, and Shaogang Gong. Towards Person Identification and Re-identification with Attributes. In *ECCV 2012*, 2012.
- [12] Chunxiao Liu, Shaogang Gong, Chen Change Loy, and Xinggang Lin. Person Re-identification: What Features are Important? In *ECCV 2012*, 2012.
- [13] Johannes Markström. 3D Position Estimation of a Person of Interest in Multiple Video Sequences: People Detection, 2013.

- [14] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 4:115–133, 1943.
- [15] Karl Pearson. Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material. *Royal Society of London Philosophical Transactions Series A*, 186:343–414, 1895.
- [16] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In *Neural Networks, 1993., IEEE International Conference on*, 1993.
- [17] C. Schmid. Constructing models for content-based image retrieval. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001.
- [18] Damien Simonnet, Michal Lewandowski, Sergio A. Velastin, James Orwell, and Esin Turkbeyler. Re-identification of Pedestrians in Crowds Using Dynamic Time Warping. In *ECCV 2012*, 2012.
- [19] Alvy Ray Smith. Color Gamut Transform Pairs. In *SIGGRAPH 78, Conference Proceedings*, August 1978.
- [20] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis and Machine Vision, International Student edition*, pages 404–407. Cengage Learning, Third edition, 2008.
- [21] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis and Machine Vision, International Student edition*, pages 393–396. Cengage Learning, Third edition, 2008.
- [22] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis and Machine Vision, International Student edition*, pages 776–780. Cengage Learning, Third edition, 2008.
- [23] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1505–1518, 2003.
- [24] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. *Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.



## Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>

## Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: <http://www.ep.liu.se/>