Examensarbete utfört i Datorseende vid Tekniska högskolan vid Linköpings universitet av

Peter Thulin

LiTH-isy-ex--15/4889--se

Linköping 2015
Anomaly detection for product inspection and surveillance applications

Examensarbete utfört i Datorseende vid Tekniska högskolan vid Linköpings universitet av

Peter Thulin

LiTH-ISY-EX--15/4889--SE

Handledare: Martin Danelljan
ISY, Linköpings universitet
Erik Ringaby
SICK IVP

Examinator: Per-Erik Forssén
ISY, Linköpings universitet

Linköping, 21 september 2015
Anomaly detection is a general theory of detecting unusual patterns or events in data. This master thesis investigates the subject of anomaly detection in two different applications. The first application is product inspection using a camera and the second application is surveillance using a 2D laser scanner.

The first part of the thesis presents a system for automatic visual defect inspection. The system is based on aligning the images of the product to a common template and doing pixel-wise comparisons. The system is trained using only images of products that are defined as normal, i.e. non-defective products. The visual properties of the inspected products are modelled using three different methods. The performance of the system and the different methods have been evaluated on four different datasets.

The second part of the thesis presents a surveillance system based on a single laser range scanner. The system is able to detect certain anomalous events based on the time, position and velocities of individual objects in the scene. The practical usefulness of the system is made plausible by a qualitative evaluation using unlabelled data.
Abstract

Anomaly detection is a general theory of detecting unusual patterns or events in data. This master thesis investigates the subject of anomaly detection in two different applications. The first application is product inspection using a camera and the second application is surveillance using a 2D laser scanner.

The first part of the thesis presents a system for automatic visual defect inspection. The system is based on aligning the images of the product to a common template and doing pixel-wise comparisons. The system is trained using only images of products that are defined as normal, i.e. non-defective products. The visual properties of the inspected products are modelled using three different methods. The performance of the system and the different methods have been evaluated on four different datasets.

The second part of the thesis presents a surveillance system based on a single laser range scanner. The system is able to detect certain anomalous events based on the time, position and velocities of individual objects in the scene. The practical usefulness of the system is made plausible by a qualitative evaluation using unlabelled data.
Acknowledgments

I would like to thank my supervisors, Erik Ringaby at SICK IVP and Martin Danelljan at ISY, for all the help during the thesis work. I would also like to thank Ola Friman for giving me some valued input on my work and Rasmus Lind for providing me with one of the datasets. And of course many thanks to my examiner Per-Erik Forssén.

Linköping, September 2015

Peter Thulin
# Contents

## Notation

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.1.1 Product Inspection</td>
<td>2</td>
</tr>
<tr>
<td>1.1.2 Surveillance Using a Laser Range Scanner</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Problem Formulation</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Restrictions</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1 Product Inspection Restrictions</td>
<td>3</td>
</tr>
<tr>
<td>1.3.2 Surveillance Application Restrictions</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Motivation</td>
<td>4</td>
</tr>
<tr>
<td>1.5 Method</td>
<td>5</td>
</tr>
<tr>
<td>1.6 Contributions</td>
<td>5</td>
</tr>
<tr>
<td>1.7 Outline</td>
<td>6</td>
</tr>
</tbody>
</table>

## I Product Inspection

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Product Inspection System</td>
<td>9</td>
</tr>
<tr>
<td>2.1 System Overview</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Image Preprocessing</td>
<td>10</td>
</tr>
<tr>
<td>2.2.1 Image Alignment</td>
<td>11</td>
</tr>
<tr>
<td>2.2.2 Data Augmentation</td>
<td>12</td>
</tr>
<tr>
<td>2.3 Product Modelling and Defect Extraction</td>
<td>13</td>
</tr>
<tr>
<td>2.3.1 Mean Image</td>
<td>13</td>
</tr>
<tr>
<td>2.3.2 Gaussian Mixture Models</td>
<td>14</td>
</tr>
<tr>
<td>2.3.3 Sample Consensus</td>
<td>16</td>
</tr>
<tr>
<td>2.4 Classification</td>
<td>18</td>
</tr>
<tr>
<td>2.4.1 Maximum Area</td>
<td>18</td>
</tr>
<tr>
<td>2.4.2 Visualization</td>
<td>19</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Evaluation of Product Inspection Methods</td>
<td>21</td>
</tr>
<tr>
<td>3.1 Evaluation Approach</td>
<td>21</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>3.2 Results</td>
<td>22</td>
</tr>
<tr>
<td>3.2.1 Bottlenecks</td>
<td>22</td>
</tr>
<tr>
<td>3.2.2 Bottle Caps</td>
<td>26</td>
</tr>
<tr>
<td>3.2.3 Business Cards</td>
<td>30</td>
</tr>
<tr>
<td>3.2.4 Tea Bags</td>
<td>35</td>
</tr>
</tbody>
</table>

## II Surveillance Using Laser Scanners

### 4 Surveillance System Using a Laser Scanner

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 System Overview</td>
<td>43</td>
</tr>
<tr>
<td>4.2 Data Acquisition</td>
<td>44</td>
</tr>
<tr>
<td>4.3 Object Detection</td>
<td>44</td>
</tr>
<tr>
<td>4.3.1 Background Modelling and Foreground Segmentation</td>
<td>44</td>
</tr>
<tr>
<td>4.3.2 Blob Detection</td>
<td>45</td>
</tr>
<tr>
<td>4.3.3 Tracking</td>
<td>46</td>
</tr>
<tr>
<td>4.4 Anomaly Detection</td>
<td>47</td>
</tr>
<tr>
<td>4.4.1 Occupancy Grid</td>
<td>48</td>
</tr>
<tr>
<td>4.4.2 Velocity Grid</td>
<td>49</td>
</tr>
<tr>
<td>4.4.3 Classification</td>
<td>50</td>
</tr>
<tr>
<td>4.5 Visualization</td>
<td>50</td>
</tr>
<tr>
<td>4.5.1 Visualizing Laser Range Data</td>
<td>50</td>
</tr>
<tr>
<td>4.5.2 Visualizing Blob Data</td>
<td>51</td>
</tr>
<tr>
<td>4.5.3 Visualizing Occupancy and Velocity Grids</td>
<td>51</td>
</tr>
</tbody>
</table>

## 5 Evaluation of Laser Scanner Surveillance System

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Evaluation Approach</td>
<td>53</td>
</tr>
<tr>
<td>5.1.1 Data</td>
<td>53</td>
</tr>
<tr>
<td>5.2 Results</td>
<td>54</td>
</tr>
<tr>
<td>5.3 Discussion</td>
<td>58</td>
</tr>
</tbody>
</table>

## III Conclusions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Discussion and Conclusions</td>
<td>63</td>
</tr>
<tr>
<td>6.1 Product Inspection</td>
<td>63</td>
</tr>
<tr>
<td>6.2 Laser Scanner Surveillance</td>
<td>65</td>
</tr>
<tr>
<td>6.3 Conclusions</td>
<td>66</td>
</tr>
</tbody>
</table>

Bibliography 69
# Notation

## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviations</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td>Gaussian mixture models</td>
</tr>
<tr>
<td>SACON</td>
<td>Sample consensus</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of interest</td>
</tr>
<tr>
<td>TPR</td>
<td>True positive rate</td>
</tr>
<tr>
<td>FPR</td>
<td>False positive rate</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver operating characteristic</td>
</tr>
<tr>
<td>RANSAC</td>
<td>Random sample consensus</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular value decomposition</td>
</tr>
<tr>
<td>KDE</td>
<td>Kernel density estimation</td>
</tr>
</tbody>
</table>
Anomaly detection refers to the theory of finding unusual patterns or observations which do not conform with the usual behaviour of a given dataset. Anomaly detection is useful in a wide range of applications, such as network intrusion detection, bank fraud detection or analysis of hyper-spectral satellite imagery for military target detection and so forth.

This document is a master thesis in computer vision. The master thesis has been done on behalf of SICK IVP in Linköping and the Department of Electrical Engineering (ISY) at Linköping University. The master thesis consists of two parts, relating to two different application domains of anomaly detection. The first is in a product inspection application using a camera and the second is in a surveillance application using data from laser scanners.

The two applications are presented as separate parts: part I contains the product inspection application and part II contains the surveillance application. This chapter gives an introduction and motivation for the two parts, a problem formulation with related restrictions and an outline for the document.

1.1 Background

This section gives an introduction and motivation for the two parts of the thesis. As stated, two applications are explored using an anomaly detection approach: product inspection with a camera and surveillance with a laser scanner.

For the product inspection application, the data consists of images of a product. Normal data should be images of a product that are considered to be normal, by for example the manufacturer or a consumer. Abnormal data, i.e. an anomaly,
should be an image of a product which is somehow defective. It could either be that is has been damaged at some point or due to some error in the manufacturing process. If a product inspection system can be taught how a normal product looks, it should ideally be able to detect any visual defects that can appear on the product during manufacturing.

For the surveillance application, the data is acquired from a laser scanner. In this case, normal data could be when and where movement typically occurs. If the system can learn the normal movement behaviour of the scene, it should be able to detect anomalous events. What the anomalies are depends on the context and situation of the surveyed area.

1.1.1 Product Inspection

Industrial quality inspection often requires manual visual inspection of products in order to determine if the product is damaged or contains errors. Due to the large scale and speed of modern day production factories, it is useful to automate this inspection by implementing automatic visual inspection systems. These systems can perform automatic quality inspection, defect detection or sorting by the use of cameras together with image processing logic.

Setting up an automatic visual inspection system can be time consuming and in some cases non-trivial, since it usually requires some knowledge in the subject of image processing. In other cases it is also necessary to have knowledge of the defects that occur in the manufactured product, which might become an issue because of the wide variety of probable defects. This motivates the need for an inspection system that has an intuitive set-up process and that can handle a large variety of defects without any prior knowledge of them.

The first part of this thesis presents a system for automatic visual defect inspection. This means that when a product is input to the system it will either be classified as normal or defective. The implemented system is trained using images of products that have been defined as normal, i.e. they do not contain any defects. Although a few images of defective samples are used to tune the parameters of the system. Using these images the system is able to extract certain defects in novel images of the same product. This part of the system has been implemented using three separate methods, which have been evaluated individually and compared to each other.

The literature for general visual product inspection systems is sparse, the focus is usually to solve the problem of inspecting defects in a specific product. Examples are inspection of foods [3, 8] or surface defect detection in fabric [9, 13]. This master thesis aims at taking a more general approach, modelling the visual properties of a product using background modelling techniques. A commonly used technique for background modelling is Gaussian Mixture Models [16, 17], abbreviated GMM. More recent background modelling techniques include ViBe
1.2 Problem Formulation

The goal of the first part of the thesis is to implement a system for automatic product defect inspection. The goal of the second part of the thesis is to implement a surveillance system based on laser scanners, that is able to detect certain anomalous events. The performance of the implemented systems should be evaluated.

1.3 Restrictions

This section presents the restrictions related to the problem formulation.

Real-time performance
Although real-time performance of the algorithms is desirable, it will not be the focus of this thesis work. This is a restriction on both parts of the thesis.

1.3.1 Product Inspection Restrictions

The restrictions on the product inspection part of the thesis are listed below.
Variations in the inspected product
The products that are to be inspected cannot have too large variations within the acceptance criteria, i.e. all forms of accepted variations of the product must exist in the training samples.

Product defects
Since the inspection system is based on vision, the defects in the inspected product must be visible and reasonably distinguishable in the image.

Alignment and scale
Since the alignment problem is not the main focus in this thesis, the images that are input to the inspection system are limited to:

- The same scaling of the object.
- Arbitrary translation as long as the whole object is still residing completely inside the image.
- Small rotation of the object, approximately ± 2 degrees.

Furthermore results from different methods for aligning images is not presented in this thesis.

Planar objects
The products inspected by the system must be approximately planar, since the inspection is based on pixel-wise comparison.

1.3.2 Surveillance Application Restrictions
The restrictions on the surveillance part of the thesis are listed below.

A single laser scanner
Some laser scanner surveillance systems are based on combining data from multiple laser scanners surveying the same scene. The system in this thesis will be limited to the use of a single laser scanner.

1.4 Motivation
The goal of the thesis was to implement and evaluate two different systems based on anomaly detection techniques. In case of the product defect inspection system, the task of setting up such a system can be non-trivial. Mainly since this might require knowledge in the subject of image processing and of the defects that can appear in the product. By taking the anomaly detection approach, these problems could hopefully be avoided, or at least reduced by some extent. In the case of the surveillance system, the biggest gain is reducing the need for human monitoring, as mentioned in section 1.1.2. Reducing the need for human monitoring should also decrease the amount of errors caused by humans, e.g. due to fatigue.
1.5 Method

This section presents the methodical approach that was used in this master thesis project. The thesis work was introduced with a study of the current literature and a planning phase. Following this was implementation of the selected methods for product defect inspection. A data collection was conducted consisting of images of different products, taken with a gray scale camera. Each image was labelled as either normal or defective. The four datasets were used for an evaluation of the performance of the implemented methods. The results from the evaluation was analysed, concluding the first part of the thesis. After this the laser scanner surveillance system was implemented. A data collection was held simultaneously while implementing the system, resulting in a single dataset. The system was evaluated on the recorded dataset and the results analysed.

1.6 Contributions

In this master thesis two applications for anomaly detection has been explored. The first application is product inspection. Here a visual defect detection system has been implemented and evaluated. The system extracts defects in a variety of products by training only with samples which are considered to be non-defective. Three methods for modelling the products visual properties have been implemented and evaluated. The first method is based on calculating the mean image of the training data. The second method is based on background modelling with Gaussian Mixture Models (GMM) [16], [17]. The third method, called SACION, is inspired by the background modelling techniques presented in [1] and [14]. The evaluation was performed on four dataset consisting of images of different products, which have been collected as part of the thesis work. Of the three methods tested, GMM and SACION performs evenly well, followed by the Mean image method which struggles in certain situations. This system and its evaluation is presented in part I.

The second application is surveillance, specifically using a 2D ranging laser scanner. A surveillance system has been implemented which models the typical movement in the scene. Using this model it is possible to detect certain unusual events. The system uses GMM for background modelling and foreground segmentation. It uses a simplistic approach for object detection and tracking, based on blob pairing and filtering translations for velocity estimation. The anomaly detection is basically done using filtered histogramming of the positions and velocities of objects in the scene. The system also includes a part for visualizing data from different parts of the system. The system is evaluated using unlabelled data recorded from an office corridor. The surveillance system for anomaly detection is presented in part II.
1.7 Outline

The outline for the document is the following:

**Part I** contains everything in the thesis that relates to the product inspection system.

**Chapter 2** presents a detailed description of the implemented system for product defect inspection, including theory related to the different parts of the system.

**Chapter 3** presents the evaluation of the product inspection system, the data that has been used to achieve the result together with a discussion and comparison of the different methods.

**Part II** contains the surveillance part of the thesis.

**Chapter 4** presents a detailed description of the implemented laser scanner surveillance system, including any relevant underlying theory.

**Chapter 5** presents the results and evaluation of the implemented surveillance system, followed by a discussion.

**Part III** contains a discussion and conclusions on both part I and II of the thesis.

**Chapter 6** presents a general discussion about the master thesis and conclusions related to the problem formulation.
Part I

Product Inspection
This chapter presents the implemented methods for automatic visual product defect inspection. Section 2.1 presents an overview of the system and sections 2.2, 2.3 and 2.4 describe the separate parts of the system in detail. The subsystem called product modelling and defect extraction presented in section 2.3, has been implemented using three different methods which are presented individually.

Table 2.1 presents the notation used throughout this chapter.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>Input sample image.</td>
</tr>
<tr>
<td>$B$</td>
<td>Binary image.</td>
</tr>
<tr>
<td>$I_i$</td>
<td>$i$:th training sample image.</td>
</tr>
<tr>
<td>$I(x, y)$</td>
<td>Pixel value of image $I$ at position $(x, y)$.</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of training samples.</td>
</tr>
</tbody>
</table>

2.1 System Overview

The product inspection system has been implemented in C++ together with the open source computer vision and machine learning library OpenCV\(^1\). The system can be divided into three parts: image preprocessing, product modelling and defect extraction, and classification. These parts are described in sections 2.2, 2.3 and 2.4 respectively. For the product modelling and defect extraction part, three different methods have been implemented and evaluated. The three

\(^1\)http://opencv.org/
In order to perform inspection of any given product, a training procedure is required. The training builds a model of the normal variations of the product. The input to the system during training is a given number of images of non-defective products. The images are aligned and cropped to a fixed size to fit the object at hand in the image preprocessing step. With these images a model of the product is built in the product modelling step. Figure 2.1 shows an illustration of the steps in the training process.

Once training is completed, the system can perform defect inspection. This process is illustrated in figure 2.2. The defect inspection also includes an image preprocessing part to align and crop the input image. These images are then processed by the product model which in turn outputs a binary image of possible defects on the corresponding input object. The classifier uses the binary image to label the input as normal or defective.

### 2.2 Image Preprocessing

The image preprocessing step is essential for the system to be able to perform defect inspection. As mentioned in section 2.1 this step includes aligning the
image and resizing it to fit the inspected object which is illustrated in figure 2.3. As the figure implies, the image preprocessing requires a template image which is used for aligning the input image and a region of interest (ROI) used for cropping the input image to fit the object.

### 2.2.1 Image Alignment

To align the images, two methods were implemented. The first method, template matching, only estimates translation. This method was used for the datasets that were rotation symmetric, that is the Bottlenecks and the Bottle Caps datasets presented in section 3.2.1 and 3.2.2 respectively. The second method uses template matching to estimate the translation followed by Lucas-Kanade tracking to estimate small rotations of the object.

As stated previously, the image alignment process uses a template image. This template image is used as a reference for how the input image should be aligned. This means that all datasets must contain a template image, that is manually aligned in a suited way. The image alignment method is the same during both training and defect extraction.

### Template Matching

The template matching method is based on normalized cross correlation. The match is found by maximizing \( R(x, y) \) given by (2.1) over the whole image.

\[
R(x, y) = \frac{\sum_{x',y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x',y'} T(x', y')^2 \cdot \sum_{x',y'} I(x + x', y + y')^2}} \tag{2.1}
\]
Here the sums range over all locations in the template $T$, i.e. $x' = 0...w - 1$ and $y' = 0...h - 1$ where the template size is $w \times h$. Note that the template in this case is not the full template image, instead the cropped template image using the specified ROI.

**Template Matching With LK-tracking**

The second method uses the same template matching that is described above to estimate the translation of the object, followed by Lucas-Kanade tracking (LK-tracking) to estimate minor rotations. LK-tracking is a method for estimation of optical flow. The OpenCV implementation of LK-tracking\(^2\) is based on [2]. It requires interest points in the image that are well suited for tracking, these are also extracted using an OpenCV function\(^3\) based on [11].

Given the set of tracked points $p_1, \ldots, p_K$ in the input image and corresponding points $q_1, \ldots, q_K$ in the template image, the transformation is given by rigid transform estimation according to (2.2).

$$\begin{bmatrix} A^* \mid b^* \end{bmatrix} = \arg \min_{[A\mid b]} \sum_i \|q_i - Ap_i - b\|^2 \quad (2.2)$$

Where the transformation has a form according to (2.3).

$$[A\mid b] = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ -a_{12} & a_{11} & b_2 \end{bmatrix} \quad (2.3)$$

The solution to (2.2) is estimated by using an OpenCV function\(^4\) for estimating rigid transformations. The OpenCV implementation uses RANSAC [5] to remove unwanted corresponding points, i.e. outlier removal. Using the remaining points an estimate is found by singular value decomposition (SVD) [7].

**2.2.2 Data Augmentation**

In order to increase the robustness of the methods it is possible to generate excess training samples, so called data augmentation. The image alignment process extracts the transformation from the input image to the template image used for alignment. Manipulating the translation part of the transformation, i.e. $b_1$ and $b_2$ in (2.3), will generate a slightly different training sample that can be applied to the product model.

Data augmentation has been implemented by randomly applying a translation in both x- and y-direction within a specified maximum offset, using a uniform distribution. The maximum offset should be set to around a few pixels at most.

\(^2\)calcOpticalFlowPyrLK  
\(^3\)goodFeaturesToTrack  
\(^4\)estimateRigidTransform
For each training sample, a number of such randomly translated samples are generated. For example, generating one extra sample per training sample will double the amount of training data. To summarize, data augmentation requires two parameters: the number of extra samples per training sample and a maximum random pixel offset.

2.3 Product Modelling and Defect Extraction

The product modelling and defect extraction part of the system is responsible for building a model of the normal variations of the inspected product. Given this model it should be able to extract defects in a novel input image of the same product. Three methods for performing product modelling and defect extraction has been implemented. These methods are presented in detail in sections 2.3.1, 2.3.2 and 2.3.3. All three methods are based on pixel-wise comparison to the product model.

During training the input should be a number of images containing non-defective samples of the product that is to be inspected. These samples form the model of the product, and depending on the method that is currently used, this model appears differently. Once a product model has been trained, the system can perform defect extraction. In this case a single pre-aligned and cropped image of a possibly defective product is input. Using the trained product model, probable defects are extracted and marked in a binary image, where 1 represents a defective pixel and 0 represents a normal pixel. Figure 2.4 illustrates the principle for the product modelling and defect extraction part of the system.

2.3.1 Mean Image

A first straightforward approach to product modelling is the Mean image method. Training is done by calculating the mean value image $\bar{I}$ of all the training samples according to (2.4).

$$\bar{I} = \frac{1}{N} \sum_{i=1}^{N} I_i$$  \hspace{1cm} (2.4)

Here $I_i$ is the $i$:th training sample and $N$ is the number of training samples. A binary image containing possible defects can be extracted by first calculating the Euclidean distance image between the mean value image $\bar{I}$ and an input sample image $S$ using (2.5).

$$D = |\bar{I} - S|$$  \hspace{1cm} (2.5)

Here $D$ is the resulting Euclidean distance image. Note that the $|\cdot|$ operator in (2.5) represents a pixel-wise Euclidean norm operation. The extracted defects are then given by (2.6) where $B$ is the output binary image and $T_d$ is a threshold value.
which is a parameter that needs to be set.

\[ B(x, y) = \begin{cases} 1 & \text{if } D(x, y) > T_d \\ 0 & \text{otherwise} \end{cases} \]  

\[ (2.6) \]

**The Effects of \( T_d \)**

\( T_d \) is the only parameter needed for the Mean image method. This determines the threshold value for the Euclidean distance image \( D \) calculated according to (2.5). A small value for \( T_d \) means that only small differences between the input sample image and the mean value image are accepted and otherwise are marked as defective pixels. On the contrary large values of \( T_d \) will accept large deviations from the mean image. Figure 2.5 shows results from defect extraction for different values of \( T_d \), white pixels are extracted defects.

### 2.3.2 Gaussian Mixture Models

Gaussian Mixture Models (GMM) is a commonly used method for performing background modelling and foreground segmentation, usually in the purpose of tracking objects in video sequences captured by a static camera. The GMM method models each pixel using multiple Gaussian distributions. The multiple components allows for better managing of pixels which tend to switch between different values, e.g. around sharp edges or corners.

For a product inspection application, one could view the background model as
2.3 Product Modelling and Defect Extraction

Figure 2.5: Example results from defect extraction with the Mean image method for different values of $T_d$. The results have been extracted using the Tea Bags dataset presented in section 3.2.4 using 105 training samples. The sample is normal, meaning that the binary output image should ideally contain only black pixels.

the normal variations of the inspected product. The foreground segmentation process extracts pixels that are deviating from the background, in this case becomes defective pixels. The implementation for GMM uses built in functions in OpenCV which are based on [16], [17]. This variant of GMM chooses the number of mixture components dynamically per pixel. In previous implementations the number of mixture components is a fixed parameter that must be set. The OpenCV implementation of GMM used in this thesis, only a limit on the maximum number of mixture components must be set, denoted $K_{\text{max}}$. Per default the value of $K_{\text{max}}$ is set to $K_{\text{max}} = 7$, which was used throughout the conducted experiments presented in chapter 3.

Instead only two parameters must be set: the length of the history and a threshold value $T_m$ on the squared Mahalanobis distance to decide whether the pixel is
well described by the background model or not. When training the model, the history parameter is simply set to the number of training samples. $T_m$ must be chosen more carefully depending on the dataset that is currently used, its effects are presented below.

The Effects of $T_m$

$T_m$ is a parameter for the threshold value on the maximum Mahalanobis distance to the background model. Figure 2.6 shows example results for different values of $T_m$.

2.3.3 Sample Consensus

The third implemented method is called SAmple CONsensus, abbreviated SACON. The method is inspired by the work in [1] and [14], where the name has been adopted from the latter. These methods are also used for background modelling and foreground subtraction. The SACON method does not require any training procedure, instead it uses the training samples as they are to perform the defect extraction. The principle of these methods is that, for each pixel, the training samples vote if the pixel is part of the sample consensus (training set) or not. The voting procedure is performed on a per-pixel basis. The first step is to calculate which training samples that agree with the input sample, at the corresponding pixel $(x, y)$, this is done according to (2.7).

$$
\Gamma_i(x, y) = \begin{cases} 
1 & \text{if } |I_i(x, y) - S(x, y)| \leq T_r, \\
0 & \text{otherwise.}
\end{cases}
$$

(2.7)

In this case $\Gamma_i(x, y) = 1$ means that training sample $i$ at pixel $(x, y)$ agrees that the input sample is a part of the sample consensus, i.e. it should be a non-defective pixel. $T_r$ is a threshold parameter on the Euclidean distance between the input sample pixel $S(x, y)$ and a training sample pixel $I_i(x, y)$. The defective pixels are
Figure 2.7: Principle for the SACON voting procedure for a 2D case. $u_1, \ldots, u_8$ represent pixel values from the training samples and $v$ is the pixel value of the input sample. In this case $\sum_{i=1}^{N} \Gamma_i(x, y) = 3$.

then extracted according to (2.8).

$$B(x, y) = \begin{cases} 1 & \text{if } \sum_{i=1}^{N} \Gamma_i(x, y) < T_n, \\ 0 & \text{otherwise.} \end{cases}$$ (2.8)

Here $B$ is the output binary image and $T_n$ is a threshold value defining the minimum number of samples that must agree for the input pixel to be classified as normal. Figure 2.7 shows the principle for the voting procedure.

**Choosing $T_r$ and $T_n$**

$T_r$ defines the sensitivity of the method, a small $T_r$ equals a high sensitivity to deviations and a large $T_r$ is the opposite. According to [14] the value of $T_n$ should be set according to (2.9) where $\tau$ is a constant.

$$T_n = \lceil \tau NT_r \rceil$$ (2.9)

Figure 2.8 shows example results for different choices of $T_r$ and $\tau$ with $T_n$ chosen according to (2.9). Figures 2.8a-2.8c show results when varying $T_r$ using a fixed $\tau$ and figures 2.8d-2.8f show results when varying $\tau$ using a fixed $T_r$. 
Figure 2.8: Example results from defect extraction with the SACON method for different values of $T_r$ and $\tau$ with the resulting $T_n$ according to (2.9) where $N = 105$. The results is from the same sample as in figure 2.5. Ideally the binary output images should be completely black since the sample is non-defective.

2.4 Classification

The models output a binary image of pixels which are deviating from the corresponding model. This image needs to be classified in order to mark the product as normal or defective. This section presents how the binary images are classified and how the result can be visualized.

2.4.1 Maximum Area

The method for classifying the binary images is based on the assumption that defects are confined to a certain coherent area in the image. Given this assumption, classification is done by defining a maximum limit on this area, which is denoted $A_{\text{max}}$. A blob detection is done on the binary image, i.e. all coherent areas of defective pixels in the binary image are detected, and the area of each such blob is calculated. If there exists a blob larger than $A_{\text{max}}$ the product in the image is classified as defective, otherwise it is classified as normal. The areas are calculated with 4-connectivity.
2.4 Classification

(a) Aligned and cropped sample.

(b) Binary sample extracted with GMM.

(c) Marked sample using $A_{\text{max}} = 4^2$.

(d) Marked sample using $A_{\text{max}} = 7^2$.

Figure 2.9: Illustration of how the visualization works using a sample from the Business Cards dataset. The sample contains a small stain in the top right corner which should be marked as defective.

2.4.2 Visualization

When a product has been classified as defective it is interesting to see where in the image the defect has been located. The classification procedure described in section 2.4.1 identifies if there are blobs larger than the parameter $A_{\text{max}}$. All blobs that are larger than $A_{\text{max}}$ are marked with circles that encloses the defective blob, overlaid in the original image. This process is illustrated in figure 2.9.

Figure 2.9a shows an aligned and cropped sample from the Business Cards dataset, presented in section 3.2.3. The sample contains a small stain in the top right corner of the image, which should be marked as a defect. A binary image is extracted using GMM, shown in figure 2.9b. The stain can be seen as a blob of white pixels in the top right corner of the image, but there is also some noise around the edges of the text. Figure 2.9c shows how the sample is marked using $A_{\text{max}} = 4^2$ for the classification. The stain is correctly marked with a circle and also some pixels around the logo is incorrectly marked. Figure 2.9d shows how the sample is marked using $A_{\text{max}} = 7^2$ for the classification. Here the incorrect pixels marked using $A_{\text{max}} = 4^2$ has disappeared.
This chapter presents an evaluation and results from the implemented methods described in chapter 2. Section 3.1 describes the methodological approach for the evaluation process and section 3.2 presents results from the four different datasets used. The datasets have been named as: Bottlenecks, Bottle Caps, Business Cards and Tea Bags, presented in detail in sections 3.2.1, 3.2.2, 3.2.3 and 3.2.4 respectively.

### 3.1 Evaluation Approach

The evaluation process begins with dividing the data into three parts:

- **Training data** is used to train the models, contains only normal (non-defective) samples.
- **Tuning data** is used to tune the parameters of the different models, contains both normal and defective samples.
- **Validation data** is used for evaluating the performance, contains both normal and defective samples.

Approximately a third of the defective samples in the dataset is used as tuning data and remaining two thirds as validation data. Both these parts get an amount of normal samples equal to the amount of defective samples in the respective category. The remaining normal samples are used as training data. Using the divided data, the following steps are repeated for each dataset:

1. Train the models using the training data.
2. Fixate the parameter $A_{\text{max}}$ for the classification.
3. Find the optimal parameters for each method given the classification parameters using the tuning data, where optimal parameters refer to the parameters which yield the highest accuracy (3.4).

4. Calculate the receiver operating characteristic (ROC) curve for each method by varying for the classification parameter $A_{\text{max}}$ using the validation data.

To calculate the ROC curve the confusion matrix must be extracted, given by (3.1).

$$
\begin{pmatrix}
\text{True Positives [TP]} & \text{False Positives [FP]} \\
\text{False Negatives [FN]} & \text{True Negatives [TN]}
\end{pmatrix}
$$

(3.1)

The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) for different classification parameters, in this case different $A_{\text{max}}$. These properties are calculated from the confusion matrix. TPR is calculated according to (3.2) and FPR according to (3.3).

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

(3.2)

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

(3.3)

The accuracy of the system is calculated according to (3.4). This property takes both true positives and false positives into account, which makes it useful for parameter optimization purposes.

$$
\text{Accuracy} = \frac{TP + TN}{\text{Number of Samples}}
$$

(3.4)

Further evaluation of the methods can be achieved by studying specific classification result using the visualization technique described in section 2.4.2.

3.2 Results

This section presents results from the methods presented in chapter 2. Each dataset is presented separately, showing some example images of normal and defective samples from the set. The resulting ROC curves are presented and any interesting classification results are highlighted. Each dataset is concluded with a brief discussion.

3.2.1 Bottlenecks

The Bottlenecks dataset consists of images of metallic bottles taken from above. There are 111 images in total where 100 are images of normal samples and 11 are defective. Figure 3.1 shows the template image with the specified ROI. Figure 3.2 shows examples of normal samples and figure 3.3 shows defective samples from the dataset. The defects consists of bottles with real cracks on the neck, shown in figure 3.3a. Also some bottles with artificially created defects to simulate smaller
3.2 Results

Cracks, shown in figure 3.3b-3.3c. These have been created by marking with a black pen on the top of the bottleneck.

Figure 3.1: Template image for the Bottlenecks dataset with the specified ROI and the result after cropping.

Figure 3.2: Normal samples from the Bottlenecks dataset after aligning and cropping.

(a) Sample with real crack. (b) Sample with small artificial crack. (c) Sample with even smaller artificial crack.

Figure 3.3: Defective samples from the Bottlenecks dataset after aligning and cropping.
Results

Figure 3.4 shows the resulting ROC curves for the Bottlenecks dataset. The figure has been computed using 88 samples for training, 8 samples for parameter tuning and 14 samples for validation. The parameter optimization was performed with $A_{\text{max}} = 4^2$ for the classification.

Figure 3.4a has been computed without data augmentation and the parameters were optimized to $T_d = 76$ for Mean, $T_m = 116$ for GMM and $T_r = 52, \tau = 0.00125$ for SACON. Figure 3.4b was computed with 5 generated data samples per training sample using a random offset between $\pm 2.5$ pixels in both directions. The resulting parameters after optimization was $T_d = 74$ for Mean, $T_m = 122$ for GMM and $T_r = 40, \tau = 0.00125$ for SACON.

Figure 3.5 shows classification results from sample 101 using $A_{\text{max}} = 6^2$ and the optimized parameters presented above. The sample is defective since it contains a crack located in the bottom of the image. Figures 3.5a-3.5c are marked when no data augmentation has been applied. Figures 3.5d-3.5f with data augmentation applied as described above.

Figure 3.6 shows classification results from sample 110 using $A_{\text{max}} = 6^2$ and the optimized parameters presented previously. The sample has an artificial crack on the bottleneck located in the rightmost part of the images. Results before and after data augmentation were identical.

**Figure 3.4:** ROC curves from Bottlenecks dataset. (a) has been computed using $T_d = 76$ for Mean, $T_m = 116$ for GMM and $T_r = 52, \tau = 0.00125$ for SACON. (b) has been computed using data augmentation with 5 samples per training sample and random offset of $\pm 2.5$ pixels. The resulting parameters after data augmentation were $T_d = 74$ for Mean, $T_m = 122$ for GMM and $T_r = 40, \tau = 0.00125$ for SACON.
3.2 Results

(a) Sample 101 marked by Mean.

(b) Sample 101 marked by GMM.

(c) Sample 101 marked by SACon.

(d) Sample 101 marked by Mean using data augmentation.

(e) Sample 101 marked by GMM using data augmentation.

(f) Sample 101 marked by SACon using data augmentation.

Figure 3.5: Results from sample 101 in the Bottlenecks dataset, using optimized method parameters and $A_{\text{max}} = 6^2$. Sample has a real crack in the bottleneck located in the bottom part of the image which should be marked.

(a) Mean

(b) GMM

(c) SACon

Figure 3.6: Results from sample 110 in the Bottlenecks dataset, using optimized method parameters and $A_{\text{max}} = 6^2$. Sample has an artificial crack on the bottleneck located in the leftmost part of the image.
Discussion

The ROC curves seen in figure 3.4 shows promising results. All three methods manage to achieve perfect classification for a given value of $A_{max}$.

One thing is worth mentioning about this dataset. The pixels in the cracks of the bottlenecks are very dark in contrast to the average pixel values around these areas which are usually bright. This large difference means that the methods can use extremely high parameter values, i.e. large deviations are allowed, without any false positive classifications. In a real application it might not be ideal to set the parameters to such high values.

All the methods successfully mark the intended defects, which can be seen in figure 3.5 and 3.6. However the Mean method also marks incorrect areas in some cases, as seen in figure 3.5a and 3.5d. The same is true for SACON in figure 3.5c where a small incorrect defect is marked. This leads to the conclusion that GMM is more precise for this dataset, even though the ROC curves in figure 3.4 are almost identical.

3.2.2 Bottle Caps

The Bottle Caps dataset consists of images of bottle caps for plastic bottles. There is a total of 105 images where 92 samples are normal and 13 samples have been labelled as defective. Figure 3.7 shows the template image with the specified ROI. Figure 3.8 shows examples of normal samples and figure 3.9 shows defective samples from the Bottlenecks dataset.

![Figure 3.7](image.png)

*Figure 3.7: Template image for the Bottle Caps dataset with the specified ROI and the result after cropping.*
3.2 Results

Figure 3.8: Normal samples from the Bottle Caps dataset after aligning and cropping.

(a) Sample with a missing seal.  
(b) Too elliptic sample.  
(c) Sample with a folded flap.

Figure 3.9: Defective samples from the Bottle Caps dataset after aligning and cropping.

Results

Figure 3.10 shows the resulting ROC curves for the Bottle Caps dataset. The figure has been computed using 79 samples for training, 10 samples for parameter tuning and 16 samples for validation. The parameter optimization was performed with $A_{\text{max}} = 4^2$ for the classification.

Figure 3.10a is computed without data augmentation and the parameters were optimized to $T_d = 72$ for Mean, $T_m = 112$ for GMM and $T_r = 48$, $\tau = 0.00125$ for SACON. Figure 3.10b was computed with 5 generated data samples per training sample using a random offset between ±2.5 pixels in both directions. The resulting parameters after optimization was $T_d = 72$ for Mean, $T_m = 80$ for GMM and $T_r = 50$, $\tau = 0.00125$ for SACON.

Figure 3.11 shows classification results from sample 88 for the different methods using $A_{\text{max}} = 7^2$. The sample should be classified as normal, i.e. the marked
images should not contain any circles. Figures 3.11a-3.11c are computed without data augmentation and figures 3.11d-3.11f are computed using data augmentation.

Figure 3.12 shows classification results from sample 102 for the three methods using $A_{\text{max}} = 4^2$. The sample should be classified as defective, because it is too elliptic. Figures 3.12a-3.12c are computed without data augmentation and figures 3.12d-3.12f are computed using data augmentation.

**Discussion**

Figure 3.10a shows that GMM achieves the best result overall, followed by SACON and then Mean. It does however suffer worse results when data augmentation was applied, as seen in figure 3.10b.

Sample 88 seen in figure 3.11 is incorrectly marked as defective by all three methods, even when $A_{\text{max}}$ is set to high values. This is most likely caused by the sample being far away from the center in the original image, causing a large change in perspective.

Sample 102 seen in figure 3.12 is correctly classified as defective by GMM and SACON using $A_{\text{max}} = 4^2$ but left unmarked by Mean.

**Figure 3.10:** ROC curves from the Bottle Caps dataset. (a) has been computed using $T_d = 72$ for Mean, $T_m = 112$ for GMM and $T_r = 48, \tau = 0.00125$ for SACON. (b) has been computed using data augmentation with 5 samples per training sample and random offset of $\pm 2.5$ pixels. The resulting parameters after data augmentation were $T_d = 72$ for Mean, $T_m = 80$ for GMM and $T_r = 50, \tau = 0.00125$ for SACON.
3.2 Results

(a) Sample 88 marked by Mean.
(b) Sample 88 marked by GMM.
(c) Sample 88 marked by SACON.
(d) Sample 88 marked by Mean using data augmentation.
(e) Sample 88 marked by GMM using data augmentation.
(f) Sample 88 marked by SACON using data augmentation.

Figure 3.11: Results from sample 88 in the Bottle Caps dataset, using optimized method parameters and $A_{\text{max}} = 7^2$. The sample is normal and should not contain any marked defects.
Figure 3.12: Results from sample 102 in the Bottle Caps dataset, using optimized method parameters and $A_{max} = 4^2$. The sample is defective since it is too oval. It should contain marked defects.

### 3.2.3 Business Cards

The Business Cards dataset consists of images of business cards from SICK IVP. There are 121 images in total where 100 images are of normal samples and 21 images of samples that have been labelled as defective. The defects are different variants of artificial misprints and stains. Two defective samples after aligning and cropping are shown in figure 3.15. Figure 3.13 shows the template image used for aligning together with the specified ROI and figure 3.14 show two normal samples after aligning and cropping.
3.2 Results

Figure 3.13: Template image for the Business Cards dataset with the specified ROI.

Figure 3.14: Normal samples from the Business Cards dataset after aligning and cropping.

(a) Sample with a misprinted S in the SICK logo.  
(b) Sample with a stain.

Figure 3.15: Defective samples from the Business Cards dataset after aligning and cropping.
Results

Figure 3.16 shows the resulting ROC curves for the Business Cards dataset. The figure has been computed using 79 samples for training, 14 samples for parameter tuning and 28 samples for validation. The parameter optimization was performed with $A_{max} = 3^2$ for the classification.

Figure 3.16a is computed without data augmentation and the parameters were optimized to $T_d = 30$ for Mean, $T_m = 14$ for GMM and $T_r = 4$, $\tau = 0.00125$ for SACON. Figure 3.16b was computed with 5 generated data samples per training sample using a random offset between $\pm 1.5$ pixels in both directions. The resulting parameters after optimization was $T_d = 32$ for Mean, $T_m = 16$ for GMM and $T_r = 4$, $\tau = 0.00125$ for SACON.

Figure 3.17 shows classification results from sample 102 for the different methods using $A_{max} = 4^2$. The sample should be classified as defective since the "B" in the "SICK IVP AB" text in the middle is misprinted. Figures 3.17a, 3.17c and 3.17e are computed without data augmentation and figures 3.17b, 3.17d and 3.17f are computed using data augmentation.

Figure 3.18 shows classification results from sample 11 for the different methods using $A_{max} = 3^2$. The sample should be classified as defective since it contains a stain in the upper right region of the card. Figures 3.18a, 3.18c and 3.18e are computed without data augmentation and figures 3.18b, 3.18d and 3.18f are computed using data augmentation.

![Figure 3.16: ROC curves from Business Cards dataset. (a) has been computed using $T_d = 30$ for Mean, $T_m = 14$ for GMM and $T_r = 4$, $\tau = 0.00125$ for SACON. (b) has been computed using data augmentation with 5 samples per training sample and random offset of $\pm 1.5$ pixels. The resulting parameters after data augmentation were $T_d = 32$ for Mean, $T_m = 16$ for GMM and $T_r = 4$, $\tau = 0.00125$ for SACON.](image)
3.2 Results

Figure 3.17: Classification results from sample 102 from the Business Cards dataset, using optimized method parameters and $A_{\text{max}} = 3^2$. The sample should be classified as defective since the "B" in the "SICK IVP AB" text in the middle is misprinted.
Evaluation of Product Inspection Methods

Figure 3.18: Classification results from sample 111 from the Business Cards dataset, using optimized method parameters and $A_{\text{max}} = 3^2$. The sample should be classified as defective since it contains a stain in the upper right region of the card.
Discussion

Studying the ROC curves in figure 3.16 it is clear that GMM and SACON both outperform the Mean method. The images contain many sharp edges, which cannot be correctly modelled by a mean value. In both GMM and SACON a pixel can contain multiple modes which makes it possible to properly model the behaviour of a pixel close to a sharp edge. This makes the Mean method less suited for this dataset.

Figure 3.16b shows the resulting ROC curves after applying data augmentation. Here GMM and SACON achieve worse result than in figure 3.16a, which is without using data augmentation. This behaviour can be explained to some extent by studying figure 3.17. The figure shows classification result from a defective sample using the same value of $A_{max}$. The sample is correctly classified as defective by all methods without using data augmentation, but incorrectly classified by GMM and SACON when data augmentation has been applied. The error arises from the fact that GMM and SACON can contain multiple modes of a pixel. When data augmentation is applied, the pixels inside the "B" are trained with both white and black pixels. During defect extraction the pixels inside the "B" are then no longer considered to be defective, since the defect itself is black, just like the printed text.

Figure 3.18a shows a "lucky" classification by the Mean method. The intended defect is a stain in the top right corner, which is completely missed, but the image is classified as defective. When data augmentation is applied as seen figure 3.18b the image is instead incorrectly classified as normal. Figures 3.18c-3.18f shows an example where the precision is increased for GMM and SACON when using data augmentation. Here the incorrectly identified defects around the logo are removed while the correctly marked defect remains after data augmentation has been applied.

3.2.4 Tea Bags

The Tea Bags dataset consist of 199 images where 180 are normal samples and 19 have been labelled as defective. Figure 3.19 shows the template image with the specified ROI used for aligning and cropping the input samples. Figure 3.20 show two normal samples and figure 3.21 show two defective samples.
Figure 3.19: Template image for the Tea Bags dataset with the specified ROI.

Figure 3.20: Normal samples from the Tea Bags dataset after aligning and cropping.

(a) Sample with misprinted logo, to the right in image.

(b) Sample with a rip, to the bottom left of the image.

Figure 3.21: Defective samples from the Tea Bags dataset after aligning and cropping.
Results

Figure 3.22 shows the resulting ROC curves for the Tea Bags dataset. The two curves have been computed using 161 samples for training, 14 samples for parameter tuning and 24 samples for validation. The parameter optimization was performed with $A_{max} = 4^2$ for the classification.

Figure 3.22a is computed without data augmentation and the parameters were optimized to $T_d = 50$ for Mean, $T_m = 40$ for GMM and $T_r = 22, \tau = 0.00125$ for SACON. Figure 3.22b was computed with 5 generated data samples per training sample using a random offset between ±2.5 pixels in both directions. The resulting parameters after optimization was $T_d = 50$ for Mean, $T_m = 40$ for GMM and $T_r = 20, \tau = 0.00125$ for SACON.

Figure 3.23 shows classification results from sample 102 for the different methods using $A_{max} = 4^2$. The sample should be classified as defective since there is a rip in the bottom right corner of the image. Figures 3.23a, 3.23c and 3.23e are computed without data augmentation and figures 3.23b, 3.23d and 3.23f are computed using data augmentation.

Figure 3.24 shows classification results from sample 11 for the different methods using $A_{max} = 4^2$. The sample should be classified as defective since the logo to the right in the image is stained/misprinted. Figures 3.24a, 3.24c and 3.24e are computed without data augmentation and figures 3.24b, 3.24d and 3.24f are computed using data augmentation.

**Figure 3.22:** Resulting ROC curves for the Tea Bags dataset. (a) has been computed using $T_d = 50$ for Mean, $T_m = 40$ for GMM and $T_r = 22, \tau = 0.00125$ for SACON. (b) has been computed using data augmentation with 5 samples per training sample and random offset of ±2.5 pixels. The resulting parameters after data augmentation were $T_d = 50$ for Mean, $T_m = 40$ for GMM and $T_r = 22, \tau = 0.00125$ for SACON.
Figure 3.23: Classification results from sample 184 from the Tea Bags dataset, using optimized method parameters and $A_{\text{max}} = 4^2$. The sample should be classified as defective since it contains a large rip in the tea bag, seen in the bottom right of the image.
3.2 Results

(a) Sample 187 marked by Mean.
(b) Sample 187 marked by Mean using data augmentation.
(c) Sample 187 marked by GMM.
(d) Sample 187 marked by GMM using data augmentation.
(e) Sample 187 marked by SACON.
(f) Sample 187 marked by SACON using data augmentation.

**Figure 3.24:** Classification results from sample 187 from the Tea Bags dataset, using optimized method parameters and $A_{\text{max}} = 4^2$. The sample should be classified as defective since the logo in top of the tea bag, to the right in the image, is misprinted.
Discussion

The Tea Bags dataset contains the most natural variations of all the datasets used for the evaluation. A small dent in the tea bag causes large shifts of the pixels values around the area of the dent. This means that the parameters of the models must be tuned to have large acceptance of variation in terms of pixel values. With parameters set to accept large variations, some defects become difficult to detect by the methods. One example is seen in figure 3.23 which contains a large rip in the tea bag to the bottom right in the image. The defect is missed by all methods with $A_{\text{max}} = 4^2$, with the exception of GMM which marks a small region after data augmentation has been applied. The figure also illustrates a "lucky" classification by the Mean method, since it marks incorrect areas as defective.

Another approach to cope with the natural variations of the product is to have even more training data, which can be simulated to some extent by the use of data augmentation. The ROC curves seen in figure 3.22 seem to indicate that data augmentation greatly increases the performance of the GMM based method. However studying figures 3.24c and 3.24d it becomes clear that there can be drawbacks to this approach. In figure 3.24c the defects on the logo are correctly marked, although after applying data augmentation and using the same $A_{\text{max}}$ seen in figure 3.24d the defect is no longer marked.
Part II

Surveillance Using Laser Scanners
This chapter describes the implementation of the laser scanner based surveillance system. Section 4.1 presents an overview of the implemented system and sections 4.2, 4.3, 4.4 and 4.5 presents detailed descriptions of the different parts of the system.

4.1 System Overview

The laser scanner surveillance system has been implemented using C++ together with OpenCV. The laser scanner that was used is a TiM55x provided by SICK. The TiM55x measures the radial distance within an angle of $270^\circ$ at a resolution of $1^\circ$.

The system can be divided into three parts: a data acquisition part, an object detection part and an anomaly detection part. These are described in detail in sections 4.2, 4.3 and 4.4 respectively. There is also a visualization part that is described in section 4.5. Figure 4.1 illustrates an overview of the system.

The data acquisition part extracts raw laser range data that has been recorded to a file. The range data is used by the object detection part which applies background modelling to detect which of the range measurements that are likely to be foreground. Given this information together with the raw range data, a blob detection is applied which clusters the range measurements into so called blobs. The object detection also uses consecutive frames of range data to track the movement objects, in order to get an estimation of their velocities.

The blob data that is passed to the anomaly detection part consists of the position and velocity of the objects that are present in the frame. The anomaly detection part uses this information to both train the system and simultaneously detect
anomalous behaviour.

Figure 4.1: Overview of the laser scanner surveillance system.

4.2 Data Acquisition

Data from the laser scanner consists of 271 range measurements in \( mm \) with an angular resolution of \( 1^\circ \) recorded at 10 \( Hz \). The maximum range is 10 \( m \). The range data is rounded to \( cm \) precision as a filtering step. Each measurement is associated with the time and date of its recording.

4.3 Object Detection

This section describes the object detection part of the system. The input to the object detection part is raw laser range data and the output is data containing information about the objects currently present in the scene. More precisely that is the objects position, a time stamp and its current velocity.

The object detection part consists of three separate sub parts. The first step is background modelling and foreground segmentation, which outputs a binary mask of probable foreground data, described in section 4.3.1. The second step is called blob detection, which clusters the continuous foreground measurements into blobs, described in section 4.3.2. The third and final step is tracking, which tracks consecutive frames of blob data to estimate the velocities of the objects, presented in section 4.3.3

4.3.1 Background Modelling and Foreground Segmentation

Background modelling and foreground segmentation is a crucial step for detecting objects in the scene. For a given angle in the range measurements from the laser scanner, the background value should be the largest measured value at that angle. This approach is taken in [6], although they also keep track of the mean and variance for each angle to increase robustness against noise. This approach comes close to a simplified version of GMM, using only one mixture component. Since GMM proved effective in part I and a highly optimized version exists in OpenCV, it was chosen for the background modelling step.
4.3 Object Detection

As stated in section 2.3.2 in part I, the OpenCV implementation of GMM is based on [16] and [17]. In this case the input is the raw data from the laser scanner, which is a 1D array of length 271. The output is a binary mask of the same length defining which range measurements that are probable foreground. As also mentioned in section 2.3.2, it requires two parameters: the history \( H \) and a threshold value \( T_m \). The history \( H \) determines the number of frames that have an impact on the background model. \( T_m \) is a threshold value on the squared Mahalanobis distance, to decide whether the pixel is well described by the background model or not.

4.3.2 Blob Detection

The blob detection process uses the binary mask output from the background modelling step and the raw range data, to detect where objects are present. A blob is defined as a contiguous segment of range data that has been extracted as foreground.

Assume that \( r_1, ..., r_n \) is a segment of range measurements which have all been extracted as foreground, by the foreground segmentation process described in section 4.3.1. All sub segments of \( r_1, ..., r_n \) which fulfil (4.1) for each consecutive range measurement, are said to belong to the same object, i.e. a blob.

\[
\Delta r = |r_i - r_{i-1}| \leq \Delta r_{\text{max}} \quad (4.1)
\]

Here \( \Delta r_{\text{max}} \) is a parameter which defines the maximum allowed distance between two consecutive range measurements. Since the angular difference is constant between each consecutive points in the data, only the range is taken into account when calculating the distances. Furthermore an object is restricted to consisting of at least two range measurements, i.e. a single foreground measurement will not be seen as an object. Figure 4.2 illustrates the principle of the blob detection.

The range measurements which are said to belong together are then transformed into so called blob data. The blob data consists of a 2D Cartesian position \( \bar{p} \) and a radius \( R \) of the smallest circle which encloses all the range measurements. The position \( \bar{p} \) of the blob, is calculated as the centroid of all the range measurements that corresponds to the blob.
Surveillance System Using a Laser Scanner

4.3.3 Tracking

Target tracking is a widely researched topic, especially when it comes to surveillance applications. Since the system is limited to only the momentary velocities of the objects, a simplistic approach to tracking was chosen for this thesis. The tracking is based on the following few steps:

1. Pairing corresponding blob data between consecutive frames
2. Calculating and storing translations of objects between frames
3. Filtering the stored translations

The first step of the tracking is to pair corresponding blob data between consecutive frames. A blob of range data is defined by its position $\bar{p}$ in the Cartesian 2D coordinate system and its radius $R$. In order to estimate the correspondence between frames, a distance measure has been defined according to (4.2).

$$d = (R_1 + R_2) - \|\bar{p}_1 - \bar{p}_2\| + \xi$$  \hspace{1cm} (4.2)

Here $\bar{p}_1$ and $R_1$ are the measured position and radius of the blob in the first frame, and $\bar{p}_2$ and $R_2$ correspond to measurements in the second frame. $\xi$ is a slack variable. If $\xi = 0$ then (4.2) is basically the radial overlap between the circles. This is illustrated in figure 4.3. Blob pairs which maximize (4.2) and fulfil $d \geq 0$ are said to correspond to each other. If $d < 0$ the blobs are assumed to be too far apart to correspond to the same object. Here $\xi$ becomes useful, since it allows for larger distances between blobs in consecutive frames, i.e. if $\xi > 0$. 

Figure 4.2: Illustration of the blob detection process. Consecutive range measurements within the limit of $\Delta r_{\text{max}}$ are considered to belong to the same object. Green dots depict foreground measurements and blue dots are background measurements.
4.4 Anomaly Detection

Using this blob pairing procedure, the objects in the frame are tracked. Each object’s measured positions $\tilde{p}_1, ..., \tilde{p}_n$ are stored as long as it is present in the scene. The translation of the object between two consecutive frames is calculated according to (4.3).

$$\Delta \tilde{p} = \tilde{p}_i - \tilde{p}_{i-1}$$ (4.3)

If there are $n$ stored positions, there will be a $m = n - 1$ calculated translations. To estimate the velocity of an object, the translations are filtered using an exponential moving average. The exponential moving average puts an increased weight on more recent samples than older ones. This is useful, since the recent translations are more interesting for the current velocity of the object. For the displacements $\Delta \tilde{p}_1, ..., \Delta \tilde{p}_m$, the exponential moving average $\Delta \tilde{p}_{EMA}$ is calculated according to (4.4).

$$\Delta \tilde{p}_{EMA} = \frac{\Delta \tilde{p}_m + w \cdot \Delta \tilde{p}_{m-1} + w^2 \cdot \Delta \tilde{p}_{m-2} + ... + w^{m-1} \cdot \Delta \tilde{p}_1}{1 + w + w^2 + ... + w^{m-1}}$$ (4.4)

Here $w \in [0, 1]$ is a weighting factor that must be set. Large $w$ weights all samples more equally and small $w$ puts larger weight on recent samples. Finally the velocity of the object is given by (4.5).

$$\tilde{v} = \frac{\Delta \tilde{p}_{EMA}}{\Delta t} = f_s \cdot \Delta \tilde{p}_{EMA}$$ (4.5)

Here $\Delta t$ is the sampling time and $f_s$ is the sampling frequency, which are assumed to be constant.

4.4 Anomaly Detection

The anomaly detection part uses the data of the tracked objects output from the object detection part, in order to model the normal behaviour in the scene. Given
a model of the normal behaviour, the system is able to detect anomalies. The detection of anomalies is performed by defining two grids: an occupancy grid and a velocity grid. The occupancy grid models how often a position in both time and space is occupied, presented in section 4.4.1. The velocity grid models the typical movement in the scene, presented in section 4.4.2.

Both the occupancy grid and the velocity grid are based on, or at least inspired by, kernel density estimation (KDE) [4]. KDE is a non-parametric method for estimating the probability density function of a stochastic variable. The estimate of the PDF is updated each time an observation of the stochastic variable is made, by applying a kernel to the position of the observation. The term probability is avoided in this thesis, since neither of the grids model probability distributions in a strict sense.

There are two core properties that motivate this approach for anomaly detection. First of all the data from the occupancy and velocity grid is easily visualized. This is useful since a user of the system can understand what the system has learned so far. The second property is that the classification becomes intuitive, since small values in the grids correspond to uncommon data. This means that finding an anomaly can be done by thresholding on new observations which have too small values in the grids. The classification is presented in detail in section 4.4.3.

### 4.4.1 Occupancy Grid

The occupancy grid can be viewed as a three dimensional histogram, which logs the positions of the objects in the scene. Each time an object is present at a given position in the grid, that position is incremented by one. The three dimensions consist of the x and y position in the 2D Cartesian grid and a third temporal dimension. The temporal dimension is defined as the time of day in which the object was present, divided to specific temporal intervals. An intuitive approach is to have intervals with a length of one hour. The temporal resolution is determined by a parameter denoted $N_{ot}$, which defines how many temporal intervals the day should be divided into. The spatial resolution must also be set, and is determined by the number of spatial grid points $N_{os}$, in both the x and y direction. The total number of grid points for the three dimensional grid will thus be $N_{os}^2 \times N_{ot}$.

The raw values of the occupancy grid can be difficult to interpret. Therefore an equalization and normalization step has been included. The equalization step, as expected, equalizes the values in the grid. This is done by taking the square root of all values in the grid. The equalization ensures that the values of certain positions do not grow too large. For example a person standing still a longer period of time would cause the system to overestimate the value of that position. Other options for equalizing the data are possible, e.g. taking the logarithm of all values. Testing different techniques for equalizing is excluded from the work of this thesis, and is instead left as possible future work.
The normalization transforms the data to the interval \([0, 1]\). The normalization is done for each individual 2D temporal slice, by dividing its values by the maximum value of that given 2D slice. As a result, the value of the most common position in each time interval, will always be 1. Another approach is to divide by the global maximum value, or even the sum of all values. Once again these options are left as possibilities for future work.

The occupancy grid is filtered both spatially and temporally using Gaussian filter kernels. The size of the filter kernels are determined by two parameters: \(K_{os}\) for the size spatial filter kernel and \(K_{ot}\) sets the size of the temporal filter kernel. The filtering is done since it is assumed to be an uncertainty in the observed positions. If an object is observed at a given position, it is likely that closely neighbouring positions are probable as well.

### 4.4.2 Velocity Grid

The velocity grid models the typical velocities of the objects in the scene. It works in similar way as the occupancy grid does. The velocity grid does not contain a temporal dimension. The reason for this is discussed later in section 6.2. Instead each grid position models the distribution of the velocity, which is a two-dimensional vector. This means that each grid point contains a two-dimensional histogram, corresponding to the x- and y-component of the velocity vector. When an observation is made at a given position in the grid, of an object with a certain velocity, the 2D histogram at that position will be updated with the observed velocity. Figure 4.4 attempts to illustrate the structure of the velocity grid.

There are two resolutions to consider for the velocity grid. First is the external resolutions, which is determined by the number of spatial grid points \(N_{ve}\) that the 2D scene should be divided into. This is similar to the parameter \(N_{os}\) for the occupancy grid. The second thing to consider is the internal resolution, which is how accurately the velocities of the objects in the scene must be modelled. The parameter \(N_{vi}\) determines the size of the 2D histogram modelling the velocity vectors. Since the histogram must have a fixed size, the velocity vectors must be re-sampled to be able to fit inside the histograms. This means that there must also be a maximum allowed velocity, which have been denoted \(v_{\text{max}}\). Velocity components above \(v_{\text{max}}\) will be saturated to fit inside the internal velocity histogram. \(v_{\text{max}}\) together with the size of the internal grid \(N_{vi}\) determines the internal resolution.

The internal two-dimensional histograms are filtered using 2D Gaussian filter kernels, similarly to the occupancy grids. The size of the filter kernel \(K_v\) is a parameter that must be set. Furthermore the internal grids are normalized in the same way as the occupancy grid, that is by dividing all values by the maximum value. This means that the most common velocity at a given position will always have a value of 1.0.
4.4.3 Classification

Both the occupancy and velocity grid contain values limited to \([0, 1]\), where lower values close to 0 are less common and vice versa. An object is classified as an anomaly if its current value in either one of the grids are below certain thresholds. The thresholds are denoted \(T_o\) for occupancy grid and \(T_v\) for the velocity grid. Higher threshold values will yield a more sensitive system and lower thresholds will allow larger deviations from normal data.

4.5 Visualization

In order to aid the analysis of the data in the different parts of the system, some tools for visualization has been implemented. Figure 4.5 illustrates the visualization for the different parts of the system.

4.5.1 Visualizing Laser Range Data

The data acquired from the laser scanner is a 1D array of range measurements in an angle ranging from \(0^\circ\) to \(270^\circ\), i.e. polar coordinate data. To be able to visualize, the data is first transformed to a Cartesian coordinate system according to (4.6).

\[
\begin{align*}
  x &= r \cdot \cos(\alpha) \\
  y &= r \cdot \sin(\pm \alpha)
\end{align*}
\]  

(4.6)

Here the \(\pm \alpha\) in the calculation of the \(y\) coordinate depends on the orientation of the sensor. Each range measurement in the array of range data is transformed
according to (4.6). Here the extracted foreground mask is used to differentiate background measurements from foreground. Range measurements that belong to the background are marked as blue dots and green dots are measurements which have been extracted as foreground. Figure 4.5a shows an illustration of this.

4.5.2 Visualizing Blob Data

The tracked blobs are visualized as green circles and can be seen in figure 4.5b. The arrow seen in the figure is the objects current estimated velocity. The numbers to the right of the circle is the objects current value in the occupancy grid, converted to percentages. If the object has been identified as anomalous, it will instead be drawn as a blue square.

4.5.3 Visualizing Occupancy and Velocity Grids

The occupancy grids are visualized by resizing them to the same size as the current displaying window. The warm colors, bright yellow or white, represent high probability positions and the black or dark red represent low probability positions. An example can be seen in figure 4.5c.

Velocity grids are visualized using arrowed lines. For each grid point in the velocity grid, the maximum probable velocity is extracted and drawn as an arrow at its position in the grid. An example can be seen in figure 4.5d.
(a) Visualization of the laser range data. Blue dots belong to the background and green dots have been extracted as foreground. Bottom right shows the current time and date.

(b) Visualization of a detected blob. The arrow represents its current velocity. The numbers to the right of the blob represent the value of the blob's position in the grid.

(c) Visualization of the occupancy grid. Warm colors represent more common positions in the grid.

(d) Visualization of the velocity grid. The green arrows represent the most common velocity at that grid point.

Figure 4.5: Visualization of different parts of the laser scanner surveillance system. The dark grey pixels indicate areas which are outside the laser scanner field of view.
5

Evaluation of Laser Scanner
Surveillance System

This chapter presents the results and evaluation of the laser scanner surveillance system. Section 5.1 presents the evaluation approach and the data used for evaluation, section 5.2 presents the results from the evaluation and section 5.3 presents a discussion on the results.

5.1 Evaluation Approach

This section presents the approach for evaluating the laser scanner surveillance system. Since the recorded data has not been labelled, no measurement of performance of the system can be extracted in the evaluation. Instead the system will be evaluated by studying the following key items:

- The most common anomalies identified by the system.
- Plots from the visualization.
- Number of detected anomalies as a function of time.

Results from the above items are presented in section 5.2 and further discussed in section 5.3.

5.1.1 Data

The dataset consists of approximately 18 days of recorded data from an office environment mostly the hours between 8.00 and 21.00. The laser scanner was placed in the middle of a three way corridor intersection in the proximity of a printer. This means that the data is a mix of people walking by and standing still close to the printer. The data is divided into two parts: training data and
evaluation data. About 13 days of the collected data is used to train the system, and the remaining 5 days are used for evaluation.

## 5.2 Results

This section presents the results from the laser scanner surveillance system. Table 5.1 present the parameter set-up for the system used throughout this section.

### Table 5.1: Parameters used for the evaluation of laser scanner surveillance system. The parameters are presented with their selected value and in which section they appeared in the thesis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_m$</td>
<td>100</td>
<td>4.3.1</td>
</tr>
<tr>
<td>$H$</td>
<td>18000</td>
<td>4.3.1</td>
</tr>
<tr>
<td>$\Delta r_{max}$</td>
<td>50</td>
<td>4.3.2</td>
</tr>
<tr>
<td>$\xi$</td>
<td>20</td>
<td>4.3.3</td>
</tr>
<tr>
<td>$w$</td>
<td>0.85</td>
<td>4.3.3</td>
</tr>
<tr>
<td>$N_{os}$</td>
<td>200</td>
<td>4.4.1</td>
</tr>
<tr>
<td>$N_{ot}$</td>
<td>24</td>
<td>4.4.1</td>
</tr>
<tr>
<td>$K_{os}$</td>
<td>7</td>
<td>4.4.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{ot}$</td>
<td>3</td>
<td>4.4.1</td>
</tr>
<tr>
<td>$T_o$</td>
<td>2.0%</td>
<td>4.4.3</td>
</tr>
<tr>
<td>$N_{ve}$</td>
<td>40</td>
<td>4.4.2</td>
</tr>
<tr>
<td>$N_{vi}$</td>
<td>80</td>
<td>4.4.2</td>
</tr>
<tr>
<td>$K_v$</td>
<td>9</td>
<td>4.4.2</td>
</tr>
<tr>
<td>$v_{max}$</td>
<td>4.0 m/s</td>
<td>4.4.2</td>
</tr>
</tbody>
</table>

Figure 5.1 shows the visualization of the occupancy and velocity grids after applying the 13 days of training data. Figure 5.1a-5.1c show the occupancy grid for different times of the day, and figure 5.1d shows the velocity grid.

Figure 5.2 and 5.3 present snapshots of the most common anomalies detected by the system. The following list presents the most commonly identified anomalies:

- Person walking faster than normal or with other uncommon velocity, examples in figure 5.3a-5.3c.

- Incorrect tracking causing strange velocity estimations, example in figure 5.3d.

- Person being close to a wall or in another less common area, example in figure 5.2a.

- Person present at unusual time of day, example in figure 5.2d.

- Person moving the laser scanner causing anomalies due to shift in background, example in figure 5.2b.

- Person standing too close to laser scanner causing range measurement errors, example in figure 5.2c.
Figure 5.4 shows a plot of the number of frames containing anomalies as a function of time. Figure 5.4a shows the plot for training data and figure 5.4b shows the same for evaluation data, after training data has been applied to the system.

**(a) Occupancy grid at 09.00**

**(b) Occupancy grid at 12.00**

**(c) Occupancy grid at 18.00**

**(d) Velocity grid**

**Figure 5.1:** Results from the occupancy and velocity grids after training using all the training data.
Evaluation of Laser Scanner Surveillance System

(a) Person present in uncommon area. Occupancy grid at 10.00.

(b) Laser scanner was moved recently. Occupancy grid at 08.00.

(c) Person too close to laser scanner causing erroneous range measurements. Occupancy grid at 17.00.

(d) Person present at unusual time of day. Occupancy grid at 20.00.

Figure 5.2: Examples of anomalies detected by the system based on position and time.
5.2 Results

Figure 5.3: Examples of anomalies detected by the system based on their velocities.

(a) Person walking faster than usual.  (b) Person walking strangely.

(c) Person walking strangely.  (d) Anomaly detected due to tracking error.
(a) Number of detected anomalies during training.

(b) Number of detected anomalies during evaluation.

(c) Normalized number of detected anomalies during training.

(d) Normalized number of detected anomalies during evaluation.

Figure 5.4: Number of frames containing detected anomalies per hour. Figures in the top row show the raw count of frames containing one or more anomalies. Bottom row show the same, though normalized by the number of frames with one or more object present, during the same period of time.

5.3 Discussion

This section presents a discussion of the results from the laser scanner surveillance system presented above in section 5.2.

In figure 5.4b there are two noticeable larger peaks between the hours 40 and 50. The first peak at hour 43 corresponds to an event where the laser scanner was slightly moved causing a shift in background which is detected as anomalous, also seen in figure 5.2b. It takes a certain amount of time for the system to adapt to the change causing the large peak in the number of detected anomalous frames. The second peak at hour 46 is caused by a person standing still close to the laser...
scanner for a longer period of time. This causes range measurement errors which are interpreted as anomalous, since they are uncommon. An example of such an event can also be seen in figure 5.2c.

In figure 5.4d there is an extremely large peak around hour 13. This peak appears this large since there are only a few frames in total with objects present in the scene, and some of these were detected as anomalous. This causes a large percentage of detected anomalous frames. Generally studying the plots in figure 5.4, the amount of detected anomalies seem to reduce over time. This is expected, since it means that the system adapts to the behaviour of the scene.

The anomalies detected by the system seem relevant for the situation, at least for the most part. For example people walking in unusual ways or unusually fast could be interesting in a general surveillance application. Also a person present in an uncommon area or at unusual daytimes can be of interest. It should however be emphasised that the system might be overly sensitive to uncommon events. This depends on the situation and can be handled by tweaking some of the parameters, e.g. increased filtering and lowered thresholds would decrease the amount of detected anomalies.

Figure 5.3d shows a person being detected as anomalous due to erroneous tracking. This usually occurred when there were multiple people in the scene walking closely to each other, causing incorrect pairing between consecutive frames. This causes unusually large velocities which were detected as anomalous. This could have been reduced by implementing a more robust tracking algorithm. Although since these anomalies are typically just one or two frames, one could introduce a threshold which only classifies an event as anomalous if there are multiple frames in a row which are below the given thresholds. This would most likely reduce these kinds of anomaly detections.

One weakness that the system has is the fact that only the current state is taken into account when detecting anomalies. There is no analysis of the trajectory of the objects in the scene. This means that there is no context to the detected anomalies. E.g. a person standing still for too long or making a sudden turn will not likely be detected as anomalous.
Part III

Conclusions
This is the concluding chapter of this master thesis. The chapter presents general discussion of the two parts, section 6.1 presents a discussion on the product inspection system from part I and section 6.2 presents a discussion on the surveillance system from part II. Section 6.3 presents conclusion made on the master thesis as a whole.

6.1 Product Inspection

The Evaluation Approach

The binary classification can in some cases be misleading, since defective samples can be correctly classified by chance, as demonstrated in figure 3.18a from the Business Cards dataset. This makes the chosen evaluation approach questionable. Another approach to evaluate the performance of the different methods would be to skip the classification step and evaluate only the binary images output from the defect extraction process. For example by using a relevant error measurement based on ground truth data. Still this approach would require manual ground truth labelling of all datasets, which can be both time consuming and non-trivial in some cases.

Data Augmentation

Data augmentation is a useful tool for simulating variations in the inspected product. It also increases the robustness to small offsets when an image is incorrectly aligned. In figure 3.18 from the Business Cards dataset, the precision of the defect extraction is increased for GMM and SACON when applying data augmentation. Since neither of these contain any incorrectly marked defects after data augmentation has been applied. There are however examples where data augmentation
works in an undesired way. The ROC-curves for the Business Cards dataset seen in figure 3.16, indicate an overall worse result for GMM and SACON when data augmentation has been applied. The sample in figure 3.17, also from the Business Cards dataset, is another example. The defect is originally detected by GMM and SACON, but is lost after data augmentation has been applied when using the same classification parameters.

These results indicate that data augmentation can be useful, especially when there is little training data, but must be applied with caution. If the system is expected to find defects with the size of a few pixels, then data augmentation might not be a viable option. The impact of data augmentation on the results could most likely have been improved, if it had been applied in a more careful manner. That is by reducing the amount of generated data and the random offset.

**Possible Extensions and Future Work**

The product inspection system is fundamentally based on pixel-wise comparison to some model, although there are other possibilities than just comparing pixel values. One could extract other features for example by using principal component analysis (PCA) on image segments. PCA reduces the dimensionality to a desirable extent, and a general appearance of the inspected product could instead be modelled. This could reduce the impact of noise in the pixel values and minor shifts in the image alignment process, making the system more robust. It would however most likely make the system less sensitive to smaller defects.

Real-time performance was not part of this master thesis, since it both depends on the hardware used and the quality of the implementation. In a real world application the speed of the algorithm is usually essential, especially in machine vision applications. For the three implemented methods for defect extraction, the Mean image based method was the fastest, followed by GMM and then SACON. Then again SACON could most likely have been made much faster with an optimized implementation. A quantitative comparison of the computation speed of the methods could be interesting for further evaluation of the three methods.

The defect inspection system is based on classifying a binary image where all pixels have been individually classified as either normal or defective. In the process of extracting the binary image a large amount of information is lost, e.g. in the thresholding in (2.6) for the Mean image method. By using the existing information prior to thresholding into a binary image, a more robust classification of the product could be achieved.
6.2 Laser Scanner Surveillance

The Evaluation Approach

The evaluation for the second part was done on unlabelled data, since there was not enough resources to perform labelling on all of the data, which is why this approach was chosen. One weaknesses of this approach is the fact that there is no measurement for how the system performs in terms of true positive and false positive rates. This makes the question of parameter choice increasingly difficult. Also actually labelling the data as normal or abnormal is itself a non-trivial task. For example in [12] the authors have asked three individuals from different academic backgrounds to label data from 3000 frames, in order to perform a quantitative evaluation. This was unfortunately not a possibility for this thesis work.

System Design Choices

Certain choices in the design of the laser scanner surveillance system are necessary to discuss. First of all, it should be noted that the tracking of objects was added in late stage of the thesis work. Originally the system was only to be based on the positions of the objects in the scene. Once this part of the system was completed, the tracking and velocity estimation was added as an extension. This extension was added in non-invasive manner, keeping the structure from the original system. This non-invasive implementation meant that the behaviour of the system stayed approximately the same, the only difference being that a larger amount of anomalies could be detected, i.e. anomalies based on an object’s velocity. A system design where the tracking is integrated with the detection of objects, might be preferable for future implementations.

The velocity grid does not contain a temporal dimension, which can also be discussed. This is a question of implementation time and dimensionality. The velocity grid as it is presented in this thesis, already consists of four dimensions. Increasing the dimensionality even further would require even more time on the implementation. Furthermore, for the dataset used in this thesis, it is not likely that the velocities will change that much throughout the day. However this will not be true for all cases. In these cases adding a temporal dimension to the velocity can be of interest.

The background modelling presented in section 4.3.1 could have been removed completely from the system. The occupancy grid could model the raw data from the laser scanner instead of the blob data, meaning that the blob detection part of the system could have been removed. One advantage of such a system would be if the surveyed area contains an extreme amount of movement on a regular basis, so much that background modelling becomes useless. A system without background modelling would most likely function better for such data. However there would likely other issues with this system. For example the most common values in the occupancy grid would be that of background readings. Positions
where movement sometimes occur would have very low values in the occupancy grid, making the task of thresholding for anomalies increasingly difficult.

**Possible Extensions and Future Work**

Handling occluded objects was excluded from the work of this master thesis, mainly due to the fact that only the instantaneous velocities of objects were of interest for the anomaly detection. Keeping track of an objects complete trajectory could be of interest for certain surveillance applications, and this would likely require the system to handle occluded objects. Due to the nature of the data from the laser scanner, finding if an object likely occludes another is almost trivial. Also using more advanced tracking techniques, e.g. Kalman filtering [15], the prediction is more or less included.

Given the improved tracking and occlusion handling, trajectory analysis can be a logical extension to the system. Trajectory analysis is a common topic for surveillance purposes. The article in [6] is such an example which analyses the trajectories of traffic at an intersection.

The visualization tools for the laser scanner surveillance system presented in section 4.5 can be extended to other applications. One example is customer heat maps, i.e. tracking the movement of customers in a retail store for the purpose of increasing store efficiency. Installing the implemented surveillance system in a store would thus serve multiple purposes, both detecting anomalous behaviour and identifying "hot spots" in the store.

**6.3 Conclusions**

Two anomaly detection based systems have been implemented and evaluated to a certain extent. The product inspection system from the first part can be trained using only images of non-defective products, although a few samples of defective products is most likely necessary in order to fine tune the parameters of the system. The evaluation of the product inspection system showed promising result for all of the four datasets. GMM and SACON works well and with similar performance, while the Mean image method is unable to handle certain situations as expected. A more precise evaluation of the system would have been desirable, as discussed in section 6.1, since this would eliminate the impact of lucky classifications. Data augmentation used in the product inspection system can make training with less data possible, although it should be approached with some care since it can cause the system to lose precision.

The surveillance system presented in the second part is able to detect certain anomalous events based on time, position and velocity of individual objects in the scene. The detected anomalies seem relevant for the context of the data, although more datasets would have been interesting for testing the system, e.g. a store, a museum or in traffic. The system was evaluated using unlabelled data
recorded from an office corridor. No performance measure could be extracted, but the usefulness of the system is made plausible by using the implemented visualization tools and analysis of the anomalies detected.
Bibliography


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/