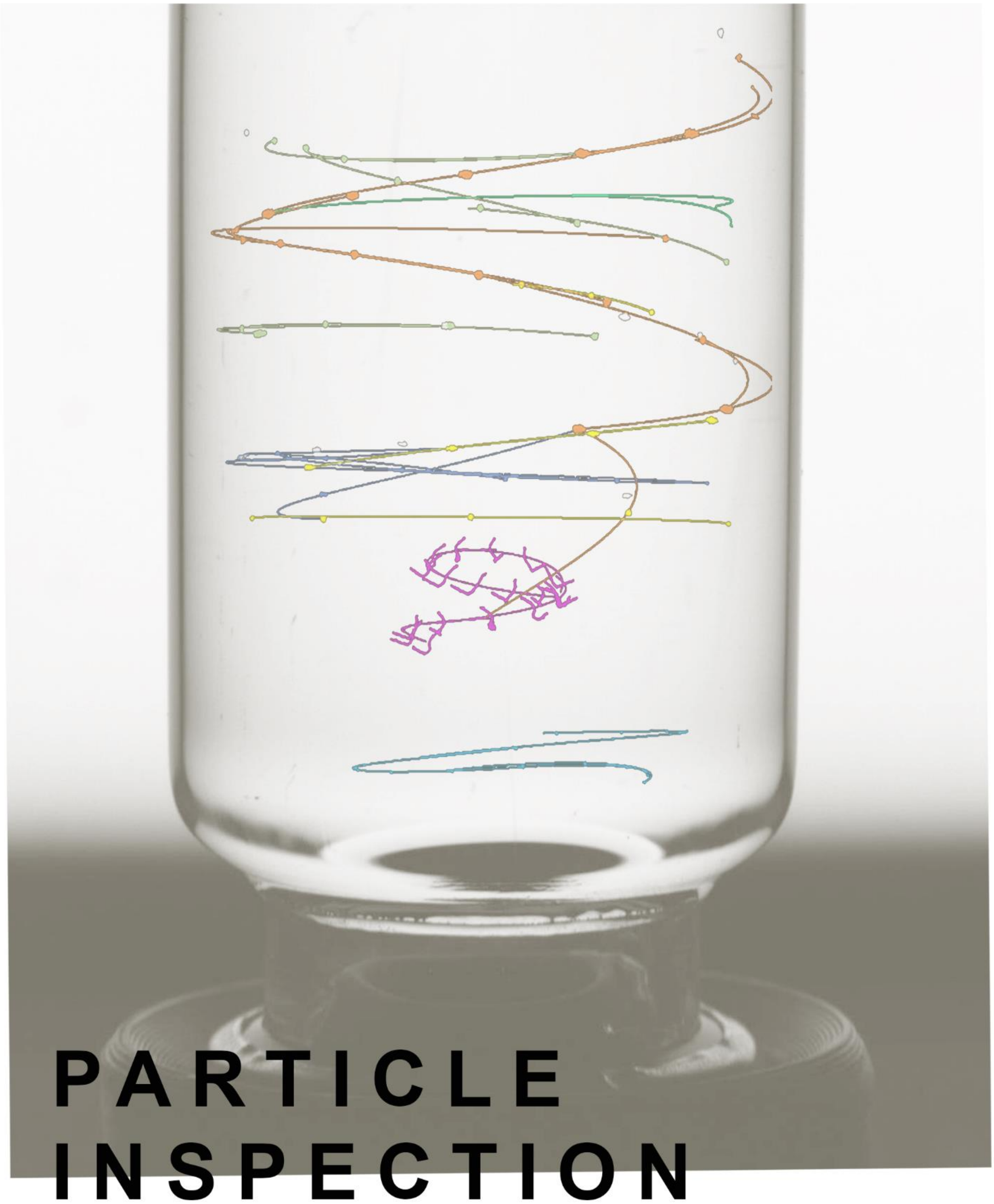
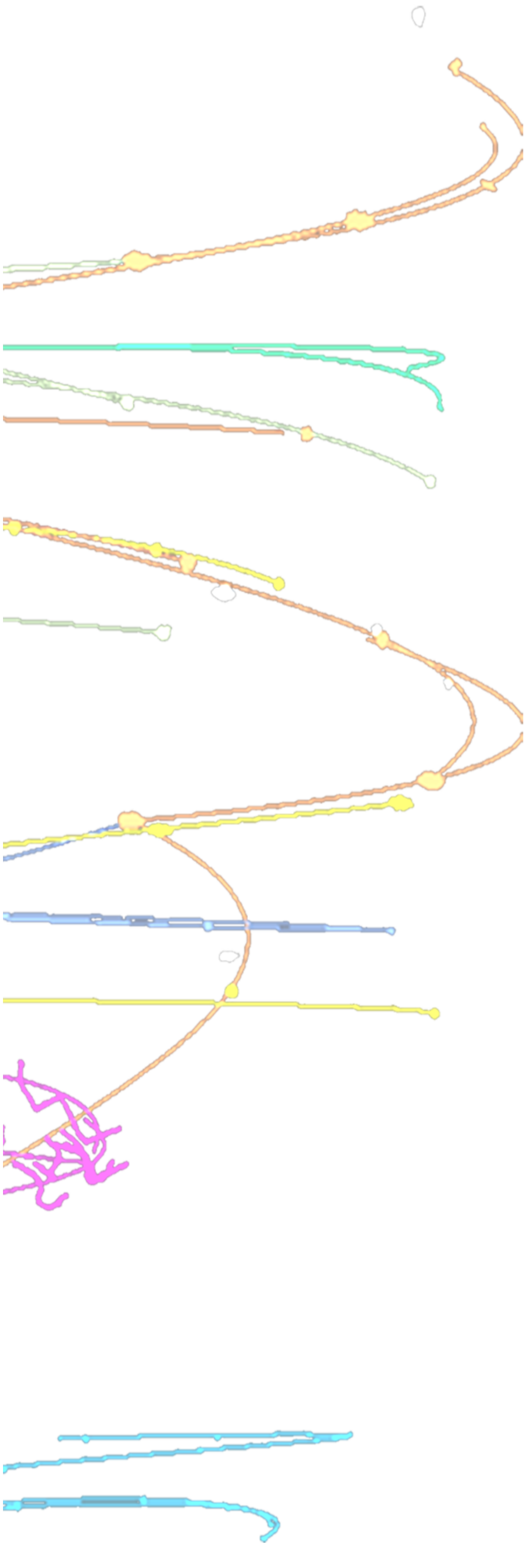


PARTICLE INSPECTION

Modules of the Visual Particle Inspection Subsystem; Detection of Particle Contamination in Medicine Containers with Novel Solutions for Background Subtraction and Segmentation, Classification, and Tracking

Master thesis
Simon van Eeden
June 2023

LUO TU Delft



Delft Technical University (Delft, Netherlands)

Faculty of Mechanical, Maritime
and Materials Engineering

Department of Cognitive Robotics

TU Delft supervisor: Prof.dr.ir. Martijn Wisse

Graduation committee: Prof.dr.ir Martijn Wisse
Dr. Holger Caesar
Ir. Ted de Vries Lentsch
Ir. Joren van der Horst

Luo Automation B.V.
Julianalaan 67a, Delft,
The Netherlands

Company mentor: Ir. Joren van der Horst

S.J.L. van Eeden (Simon)

Delft, June 2023

Preface

Throughout my academic journey as a student in Mechatronics and Robotics, I have been fascinated by Computer Vision and how it can be used to solve complex problems to improve safety and well-being. This master's thesis is the culmination of months of research, testing and writing, and I am excited to share my findings with the academic community.

The purpose of this thesis is to develop a subsystem that can detect and classify hazardous particles in liquid medication containers such as vaccines. This technology can support small-scale produces of medicine such as hospital pharmacies to compete with large-scale producers of medicine and temporarily produce medicine that are not available. Trained lab technicians that currently perform the tedious task of manual visual inspection will be able to focus on other important tasks. The benefits for patients include decreased safety risks and more affordable healthcare. The positive effects of these developments have been a great motivation for this thesis.

I would like to express my sincere gratitude to my thesis supervisor Martijn Wisse for his guidance, support, and invaluable feedback throughout this process. I would also like to thank Hans Gaiser for contributing his knowledge in the field of computer vision. I would like to thank my colleagues at Luo Automation B.V. for their discourse and creating a friendly working environment. Special thanks go out to Charlotte for her invaluable support, daily inspiration and help with the structure of my thesis.

I hope that this work will contribute to the ongoing development of automated visual inspection technology for small-scale medicine production and provide valuable insights for future researchers in the field.

Abstract

Visual inspection of liquid medicine containers for contamination and defects is mandatory and crucial to ensure their safety for injection. This document presents research and development of three modules of the Visual Particle Inspection Subsystem (VPIS), an automatic inspection subsystem with the task of detecting and classifying particle contamination in liquid medicine containers. The proposed VPIS comprises three modules for which research and development is performed separately: the backgrounds subtraction and segmentation module, the classification module, and the tracking module.

For the *background subtraction and segmentation module*, a solution is proposed with filtered temporal background modelling and locally adaptive threshold segmentation. This solution works on the assumption that moving objects such as particles and bubbles are sparse and do not occur more than twice in a pixel in twenty frames. Therefore, a representative incomplete background cluster can be obtained by removing the two lowest or highest pixel values depending on the mode of illumination. Next, the average value of the background cluster is used for background subtraction and the standard deviation of the background cluster is used to determine a locally adaptive segmentation threshold. The solution has been positively evaluated for detection of low contrast objects, insensitivity to disturbances, processing time, and parameter configuration. Additionally, an add-on solution is proposed that makes it possible to detect objects in the challenging region near the rubber stopper of a syringe.

For the *classification module*, out of four candidate classification methods a Convolutional Neural Network (CNN) is chosen for the classification of single object detections. The CNN classifier achieves an accuracy of 0.93 and is used as a baseline classifier for the rest of this research. Next, with three exploratory research questions, research is performed into opportunities and pitfalls for this classification problem summarised below.

- It is found that certain handcrafted features correlate to the classification accuracy of a detection. A method is proposed to predict the classification accuracy of detections and filter out detections that cannot be classified reliably. Compared simple filters that filter out detections based on a single feature such as area or total contrast, the proposed method should achieve a higher subsequent classification accuracy and reject less detections.
- With the simplest classification strategy that is often used for inspection, a container will be rejected if a single detection is classified as a particle. It is shown through simulation that this classification strategy is not effective for this classification problem. This is as the large number of bubble detections in a clean container would result in a high false container rejection rate.
- Two multi-detection classification strategies are proposed that use multiple detections for classification. Median voting classification uses results from the tracking module to classify multiple detections of the same object. Multi-positive classification does not use tracking results but requires multiple detections to be classified as particles. Simulation results indicate that with both strategies, the classification accuracy for containers can be drastically improved. It is theorized that the best results can be achieved with median voting classification.

For the *tracking module*, the Trajectory Driven Cluster Proposals (TDCP) algorithm is proposed and evaluated on a small dataset. The TDCP algorithm performs tracking based only on detection coordinates and works on the assumption that real objects follow a predictable spiralling trajectory. Using a motion model, each set of three detections is evaluated to find plausible partial trajectories called tracklets. Next, overlapping tracklets are merged into track candidates of varying lengths. As this method returns all possible trajectories, in general more trajectories are detected than there are objects present. TDCP has been evaluated in terms of processing time, correctly tracked objects, additional trajectories, and trajectory length. The current version of TDCP fails in terms of processing time and performs questionably in terms of additional trajectories due to the rapid growth phenomenon when multiple objects are in close proximity. Some methods are discussed to make TDCP suitable for implementation with future work.

The research and development findings presented in this thesis contribute to the advancement of automated visual inspection technology for small-scale medicine production.

Table of Contents

Glossary

Term	Definition
Visual Particle Inspection Subsystem (VPIS)	A system for detecting moving particles in liquid medicine containers.
Detection	An observation of a moving object detected by the background subtraction and segmentation module
Moving object	A moving particle or bubble
Image disturbances	Phenomena that cause a section of the image to differ from a clear background image
Meniscus	Concave surface between air and liquid
Segmentation mask	Binary image that indicates what parts of an image are part of a segmented object
Foreground	Anything that is not a part of the static background
False Negative (FN) rate	Rate at which particle detections are falsely classified as bubbles
False Positive (FP) rate	Rate at which bubble detections are falsely classified as particles
False Container Rejection (FCR) rate	Rate at which an uncontaminated container is falsely rejected as a contaminated container
False Container Acceptance (FCA) rate	Rate at which a contaminated container is falsely accepted as an uncontaminated container
Handcrafted features	Feature properties derived using various algorithms using the information present in the image
Relief feature selection	A family of feature selections that uses nearest neighbours to select features with statistical interactions with other features
One-Class Support Vector Machine (OC-SVM)	Special SVM for outlier detection based on a single class
Support Vector Machine (SVM)	A type of linear model for classification and regression problems
Neural Network (NN)	A machine learning method inspired by the human brain that can solve complex problems
Convolutional Neural Network (CNN)	A class of artificial neural network most commonly applied to analyse visual imagery
Trajectory Driven Cluster Proposals (TDCP)	Proposed solution for the tracking module that performs tracking based only on detection position using a motion model.
State-Of-The-Art (SOTA)	The best methods that have been identified from previous work

1 Introduction

This thesis is a graduation project for the master Robotics at Delft University of Technology, performed for Luo Automation B.V. in Delft, The Netherlands.

First, in section 1.1, the need for particle detection is explained as motivation for this thesis. Next, in section 1.2, the objective of this research is introduced. In section 1.3 the document structure is presented. Section 1.4 shows an overview of the image acquisition process which precedes the first module of the VPIS.

1.1 Need for particle inspection

By industry regulation, all medications that come into contact with human blood or tissue should be tested for contamination with particulate matter. Particle contamination of liquid medicine is defined by the United States Pharmacopeia (USP) as “contamination of injections and infusions consisting of extraneous, mobile, undissolved particulates other than gas bubbles, unintentionally present in the solution.” [1]. Particles such as glass chips, rubber particles, fibres, or hair can be accidentally introduced to medicine containers during the production process from various origins such as faulty machinery, workers, or an unclean production environment [2].

Small particulate matter from contamination often cannot be metabolised by the human body and can cause serious harm if injected into a patient. Possible consequences include thrombus, phlebitis, tumours, anaphylactic reactions, or even death [3] [4]. Presence of particle contamination has been one of the top ten reasons for the recall of liquid pharmaceuticals [1]. The probability that particle contamination is present in a container under normal conditions is low, less than <0,35% by estimation [1].

Visual inspection is mainly performed at pharmaceutical production facilities but also at research laboratories and hospital pharmacies. Batch sizes can range from over a million units at a production facility to a dozen units at a hospital pharmacy. Different visual inspection methods are used depending on the production scale.

At large pharmaceutical production facilities, visual inspection is mostly performed using automated inspection machines such as the Seidenader VI-S [5]. Such an automated inspection machine can process over 35.000 medicine containers per hour with a high accuracy. On the downside, these machines require a large investment with a high maintenance cost and can be more than 6 metres in length. Re-configuration of such a machine for a different container type takes some time and the required parts can be costly. These existing commercially available automatic inspection machines are therefore optimal for large-scale production but not suitable for small-scale or batch production.

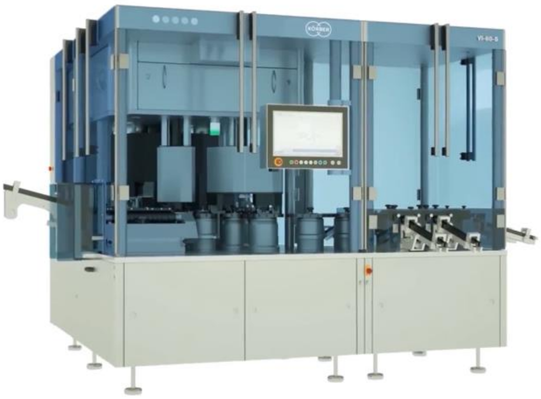


Figure 1 Seidenader VI-S automatic inspection system. Source: [5]

At small-scale production, inspection is performed manually or semi-automated. During manual inspection, a worker agitates the liquid by swirling it and visually inspects the container for particles that move in the solution. Industry regulation requires that the containers are inspected for ten seconds each under intense lighting conditions in two stages: with a bright white background to detect dark, opaque particles and with a black background with diagonal lighting to detect reflective or brightly coloured particles, this is later referred to as back

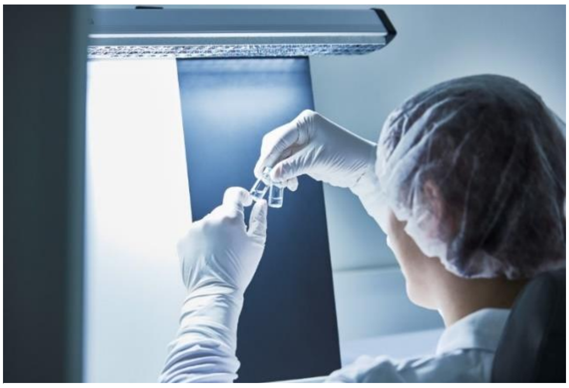


Figure 2 Manual inspection process with white and black backgrounds to detect different types of particles. Source: [38]

illumination and diagonal illumination. Workers must take regular breaks, as fatigue can cause an increased rate of false positive or false negative detections. With semi-automated inspection, the human is assisted by a machine to perform inspection. The machine agitates the container and shows it to the human operator. The human operator is assisted by a convex lens to magnify the container to spot particles quicker.

Luo Automation B.V., a start-up company based in Delft, The Netherlands, is currently working on the first automatic inspection system suitable for small-scale inspection. The development of this system is the context of this research. By enabling small scale producers of medicine to perform inspection safer and more efficiently, this system contributes to safer and more affordable healthcare and enables small-scale producers to compete with large organisations and fill-in demand for medicine that are temporarily unavailable.

1.2 Research objective

The main objective of this research is to contribute to the development of the Visual Particle Inspection Subsystem (VPIS), which is a proposed subsystem for the container inspection system by Luo Automation B.V. Figure 3 shows a functional overview of the inspection machine with the VPIS.

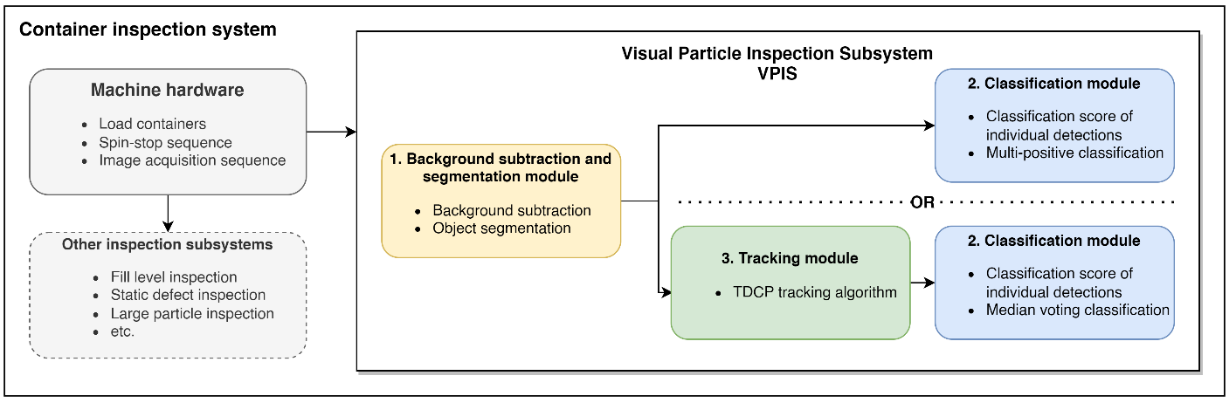

Container inspection system
Machine hardware
• Load containers
• Spin-stop sequence
• Image acquisition sequence
Other inspection subsystems
• Fill level inspection
• Static defect inspection
• Large particle inspection
• etc.
Visual Particle Inspection Subsystem
VPIS
1. Background subtraction and segmentation module
• Background subtraction
• Object segmentation
2. Classification module
• Classification score of individual detections
• Multi-positive classification
OR
3. Tracking module
• TDCP tracking algorithm
2. Classification module
• Classification score of individual detections
• Median voting classification

Figure 3 Schematic functional overview of the container inspection system with the VPIS with three modules. Research and development into the three modules of the VPIS is the topic of this thesis.

The goal of this research is not to deliver a completely functional VPIS, but to perform research and development into the three modules of the VPIS. The main task of the VPIS is to detect and classify medicine containers that are contaminated with small particles that move freely inside the container. This task has been split in three sub-tasks which are each performed by a module of the VPIS.

1. The *Background subtraction and segmentation module* performs the sub-task of detecting and segmenting any moving objects from a sequence of 20 images.
2. The *Classification module* performs the sub-task of classifying the detections from the previous module to determine whether any particles are present in the container.
3. The *Tracking module* performs the sub-task of tracking moving objects from detections by the background subtraction and segmentation module. The results of tracking indicate what detections are of the same object which is used by a proposed classification strategy called Median voting classification. The tracking module is not needed if another proposed classification strategy is used called multi-positive classification.

Research and development into each module is performed separately with separate research questions. The proposed solutions are designed to be easily reconfigured for different inspection situations with different medicine and container types, and work well with the most challenging small particles.

Previous to this research, a literature study has been performed into existing methods that are used to perform particle inspection in liquid medicine. Of the methods that can be found through academic research, no methods were found that are expected to perform optimally to detect challenging particles in a realistic situation. In this research, for each module a short summary of previous work for that module is presented, including the shortcomings of the current State-Of-The-Art (SOTA). At the end of each module chapter, the proposed solution is compared to these SOTA methods.

1.3 Document structure

Firstly, section 1.4 shortly explains the image acquisition process that precedes the task of the VPIS.

The main content of this research are the three modules that make up the VPIS. Research and development into each module has been performed separately and is presented separately in Chapters 2-4 .

Each module chapter starts with an introduction that explains the goals and challenges of that module. Next, a short summary of previous work is presented including the current State-Of-The-Art (SOTA) and its shortcomings. Next, the research questions of that module are presented. The sections after that present the main body of the research which is different for each module. For the background subtraction and segmentation module and the tracking module, observations and assumptions are made after which a solution is proposed and verified using the research sub-questions for that module. For the classification module, a dataset is created after which separate research questions are presented and concluded one by one. The first research question of the classification module selects a baseline classifier for singular detections. The other three research questions explore the possibilities and pitfalls that come with this classification problem. Afterwards, the solutions presented for each module are compared to the SOTA presented earlier. Finally, each module chapter ends with a conclusion and recommendations for future work.

1.4 Image acquisition process

This section presents an overview of the image acquisition process that collects images for the VPIS.

The automatic inspection process to detect particulate matter starts with a rapid spin-stop sequence followed by an image acquisition sequence. Medicine containers are individually accelerated to rotate up to 500-3000 rpm depending on the type of container and then quickly decelerated. As the container rotates, the liquid inside the container starts to rotate and creates a vortex. After the container is held static, the liquid continues to rotate due to inertia and lifts particles from the bottom of the container into the solution. During this time, two sequences of twenty images are collected at twenty frames per second using a high-resolution monochromatic camera.

Figure 4 shows the image acquisition process. Two illumination styles are used to detect all types of particles. An illuminating white background is used to detect light-blocking particles, such as rubber or metallic particles. The second illumination style uses a top and bottom light in front of a black background and is used to detect bright and reflective particles such as hair, fibres, and lightly coloured plastic. These are later referred to as back illumination and diagonal illumination.

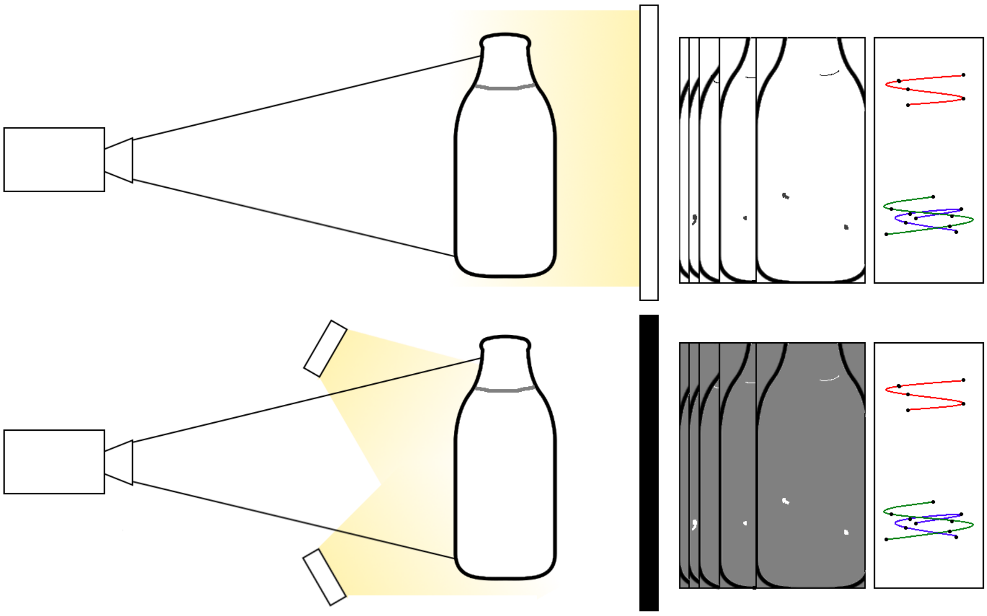


Figure 4 Image acquisition process with back illumination and diagonal illumination.

After image acquisition, it is the task of the VPIS to detect and classify any moving particles from the image sequences.

2 Background Subtraction and Segmentation module

The first module of the Visual Particle Inspection Subsystem (VPIS) is the background subtraction and segmentation module. This chapter contains research and findings regarding the background subtraction and segmentation module.

The goal of the background subtraction and segmentation module is to extract detections of moving objects from a sequence of images. These extracted detections are used as input for the classification and tracking modules. The current state of the art from previous work has shortcomings and does not perform optimally as discussed in section 2.1. The proposed solution aims to overcome these shortcomings.

Ideally, the result of background subtraction and segmentation is a complete list of detections of all objects in every frame, where each detection contour is accurately segmented. This is challenging as some objects are barely visible and can have a very low contrast to the background. It should also be avoided that small disturbances in illumination cause a false object detection as this can ultimately result in a false particle classification.

This research proposes a combined solution for background subtraction and segmentation. The proposed solution uses *filtered temporal background modelling* and *locally adaptive threshold segmentation*. First, a filtered background model for each pixel is obtained by removing the two lowest or highest pixel values in the sequence of images. Next, the average intensity and standard deviation of the background model are used to define a local threshold for each pixel. Any pixel intensity outside this threshold is assumed to be of a moving object.

The proposed solution has been developed and is shown to be effective through validation in various aspects. The solution can accurately detect the contours of small moving objects with a low contrast while detecting few false detections caused by small disturbances in illumination. The processing time less than a second for two sequences, making it fast enough for the VPIS. As the segmentation threshold is adaptive, it does not need to be carefully reconfigured for different imaging situations, making it ideal for small-scale inspection with multiple medicine container types.

Chapter structure

This chapter presents findings regarding the research and development of the background subtraction and segmentation module. First, section 2.1 presents a short summary of previous work into background subtraction and segmentation for the inspection of medicine. A selection of current state of the art methods is presented together with shortcomings and take-aways for the proposed solution. Next, in section 2.2, the research questions are presented that will be used to later evaluate the proposed solution. After that, section 2.3 presents observations and assumptions that are important to understand the problem and are used in the working principles of the proposed solution which is presented in section 2.4. Section 2.5 contains results and demonstrates the proposed solution. In section 2.6, the proposed solution is validated using the research sub-questions. In section 2.7, the proposed solution is compared directly to the current state of the art. Finally, sections 2.8 and 2.9 present the conclusion and recommendations for this module. An add-on solution is presented in Appendix A that enables the module to detect moving objects near the bottom of a syringe.

2.1 Previous work

An analysis of previous work for the problem of particle detection has been performed during a literature study [6] preceding this thesis. This section summarises the most interesting findings from the literature study relevant for the background subtraction and segmentation module. Next, current State-Of-The-Art (SOTA) methods are identified in more detail with shortcomings and take-aways. In section 2.7, the solution proposed for the background subtraction and segmentation module is compared to the SOTA methods.

Background subtraction methods

The background subtraction methods presented in previous work can be separated into four groups: image differencing [7] [8] [9] [10] [11] [12] [13] [14], background modelling [15] [16], dictionary learning [17], and subtraction after segmentation [18]. Of these methods, background modelling with Gaussian Mixture Models [16] is expected to perform the best, followed by a method referred to as simple background modelling [15]. The other methods result in more image noise, are less reliable or are overly complex.

Segmentation methods

The segmentation methods identified in previous work are: simple threshold [15] [17], local adaptive threshold [9], local line threshold [19], automatic threshold detection [7] [20], fuzzy C-means with fuzzy SVM [18], PCNN [8] [10], FCNN [12], Faster-RCNN [21] [22] [23], and Gaussian Mixture Model segmentation [16]. Of these methods, background modelling with Gaussian Mixture Models [16] is expected to perform the best. The expected second-best method is local adaptive threshold segmentation [9].

2.1.1 Current SOTA

This section explains the workings and shortcomings of the identified SOTA methods: background modelling with Gaussian Mixture Models [16], simple background modelling [15], and local adaptive threshold segmentation [9].

Background modelling with Gaussian Mixture Models

Background modelling with Gaussian Mixture Models (GMM) [16] has been identified as the SOTA for both background modelling and segmentation. After the literature study, it was hypothesized that GMM background modelling is ideal for the background subtraction and segmentation module as would be stable and not require a threshold parameter. However, the method cannot be used as described in the paper as it has shortcomings discussed below.

GMM [24] is a probabilistic clustering method that is regularly used for background subtraction outside of this specific use case. When used for background modelling, a GMM is fitted on intensity values of a pixel over time. A GMM contains gaussian components with a mean and a standard deviation. During the fitting steps, the mean and a standard deviation of gaussian components are optimised such all datapoints have a high membership with a Gaussian component. Figure 5 shows an example of a GMM with two Gaussian components fitted on a set of background values and two foreground values.

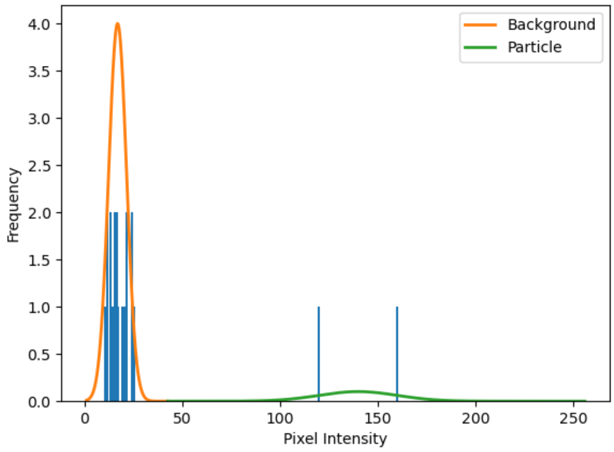


Figure 5 Example of a GMM with two components fitted on a set of background values and two particle values

There are two main shortcomings with background modelling with GMM, processing time and a low number of datapoints.

For background modelling with GMM, a separate GMM is fitted for every pixel in the image. As the fitting process is iterative this takes a significant amount of time. During initial testing, it was quickly found that GMM background modelling is too slow when used with high resolution images.

The second shortcoming is that GMM background modelling is most suited for situations with many datapoints where multiple datapoints are present per Gaussian component. For this problem, only 20 intensity values are available per pixel of which at maximum two independent values belong to the foreground cluster. As it is not possible to define a standard deviation based on one or two points, the method becomes dependent on precise parameter settings and random initialisation.

Simple background modelling

Simple background modelling [15] works on the assumption that a pixel position will never contain a pixel for two consecutive frames. To construct the model, each intensity value that is close enough to the previous intensity value is assumed to be a background value. Other intensity values can present background or a moving object. The average value of this incomplete background cluster is then used to perform image differencing.

This method is an interesting as is much simpler compared to GMM background modelling. Instead of detecting multiple clusters for background and foreground, it approximates only the background cluster to determine the average background value. A downside of this method is that it requires a good approximation of background intensity variance to determine when two intensity values are close enough to belong to the background cluster. This method will therefore become less accurate or can fail in regions where the background intensity varies more as shown in Figure 7 on page 23.

Local adaptive threshold

Local adaptive threshold segmentation [9] aims to improve on segmentation with a regular threshold by making the threshold adaptive. The proposed locally adaptive threshold is computed with a global threshold parameter combined with the mean intensity of the 3x3 neighbourhood around a pixel in the foreground image. The adaptive threshold is increased in regions with a higher foreground intensity, this is the case in regions with more illumination disturbances but also between the edges of a particle. The resulting effect is comparable to a form of edge detection and would work reasonably well on particles with a clearly visible contour. However, the method will work less well when detecting objects with a soft gradual contour. An additional downside is that the method requires two threshold parameters: the global threshold parameter and a parameter that determines the contribution of the local neighbourhood.

2.1.2 Take-aways

Although none of the SOTA are ideal, some useful take-aways have been made that have influenced the proposed solution for the background subtraction and segmentation module.

Background modelling with Gaussian Mixture Models [16] uses the background model for both background subtraction and segmentation. This is possible as any moving objects are not part of the background component. Whether an intensity value is a part of the background component or becomes a new component is dependent on the average and standard deviation of the background cluster. A practical solution that achieves the same result would not require any gaussian components other than the background component therefore the method can be simplified.

Simple background modelling [15] uses an incomplete background cluster to approximate the average background intensity. If the incomplete background cluster would be complete enough, it could also be used to determine the standard deviation and achieve a result similar to background modelling with Gaussian Mixture Models as discussed above.

Simple background modelling [15] performs background modelling not by fitting a function but by comparing intensity values over time with the assumption that a particle does not appear twice in a row. The solution proposed for the background subtraction and segmentation module uses a similar assumption to obtain a more reliable incomplete background cluster.

Local adaptive threshold segmentation [9] uses a locally adaptive threshold to improve segmentation in regions with more variation in illumination. Instead of the local neighbourhood, the solution proposed for the background subtraction and segmentation module uses a temporal locally adaptive threshold based on the values of that pixel over time.

2.2 Research questions

This section introduces the research question and sub-questions for the background subtraction and segmentation module. The research question for this module regards the effectiveness and suitability of the solution that is proposed in section 2.4.

A: *"Is the proposed solution effective and suitable to be implemented in a system for particle detection?"*

In order to answer research question A more effectively, it has been divided into the following sub-questions. Each of these sub-questions evaluates specific aspects regarding the effectiveness and suitability for implementation. Each sub-question is evaluated separately in subsections 2.6.1 - 2.6.4.

A.1: Low contrast detections
"Can the solution detect objects with a low contrast to the background?"

A.2: Insensitivity to disturbances
"Is the solution insensitive to image noise, static disturbances and soft illumination disturbances, such that approximately less than 1% of detections are caused by these image disturbances?"

A.3: Processing time
"Can background subtraction and segmentation be performed on two sequences of 20 images within one second, as not to slow down the VPIS?"

A.4: Parameter reconfiguration
"Can the method easily adapt to different container types or types of illumination?"

2.3 Observations and assumptions

This section summarises observations and following assumptions from analysing the input data for the background subtraction and segmentation module. First, an overview of the input data is presented with image specifications and some example images. Next, notable observations are shortly discussed. These observational findings serve as background information for this module and are used in the working principle for the proposed solution in section 2.4.

2.3.1 Image data

The input data for the background subtraction and segmentation module is a sequence of high-resolution greyscale images. The length of the sequence is 20 frames acquired at 20 frames per second. Images are cropped according to the size of the container resulting in the image sizes shown in Table 1. Figure 6 shows some example images of a syringe and a vial with back illumination and diagonal illumination.

Table 1 Image size per container type before and after cropping the centre region.

Container type	Size	Total area
Syringe 5ml	1600 x 500	800.000
Vial 10ml	1000 x 700	700.000
Syringe 5ml, centre region	1150 x 400	460.000
Vial 10ml, centre region	750 x 600	450.000

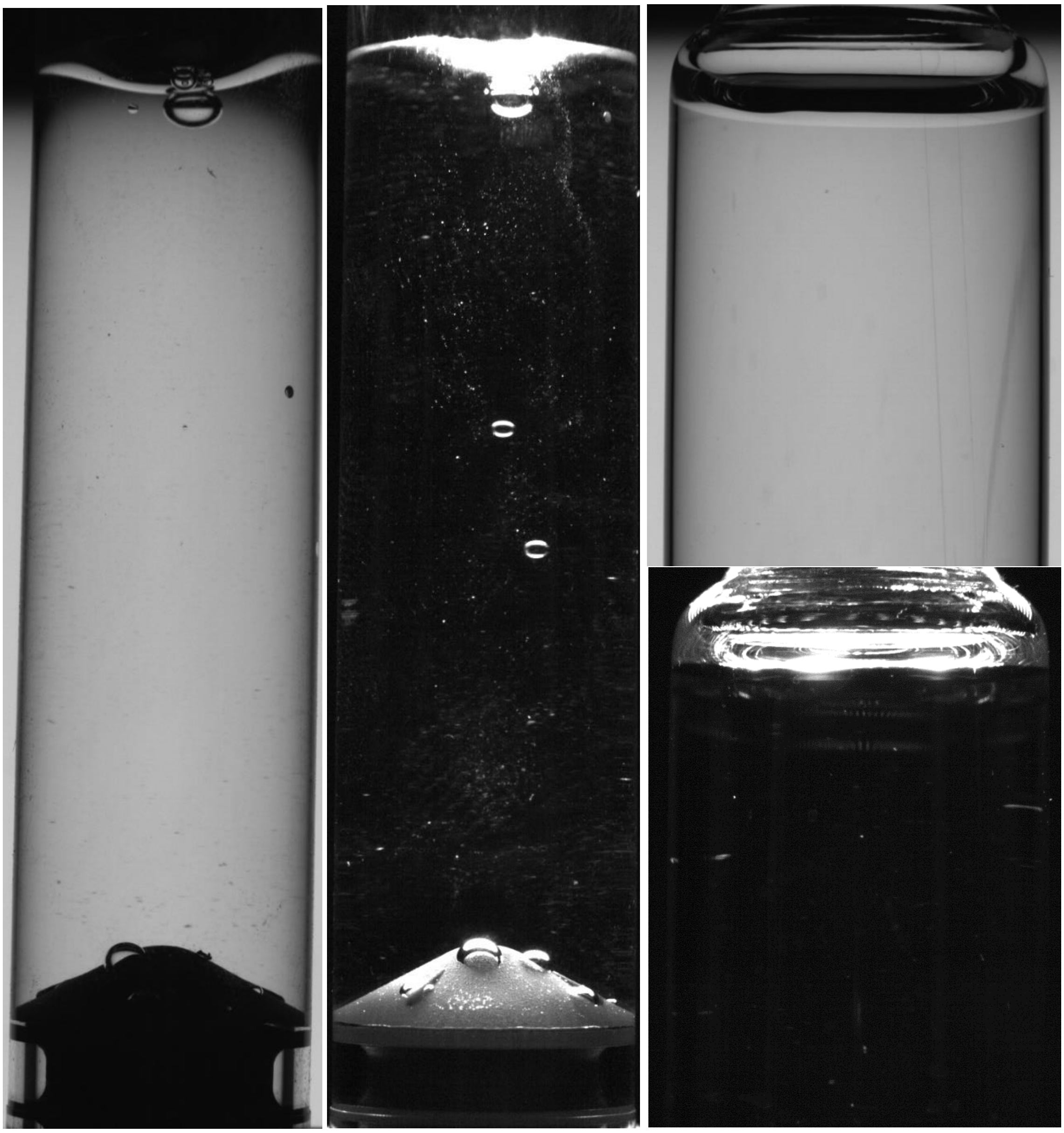


Figure 6 Example images of a 5ml syringe and 10ml vial with back illumination and diagonal illumination. The images are not on scale.

2.3.2 Notable observations

This section discusses the most notable observations and assumptions from analysing the input data for the background subtraction and segmentation module.

Image disturbances

In the ideal situation for background subtraction and segmentation, images consist of a perfectly static background image of a clean container with the addition of clearly distinguishable moving objects in the foreground. However, as can be seen in Figure 6, other disturbances are present in the image.

Image disturbances are phenomena that cause a section of the image to differ from a clear background image. Table 2 shows a list of image disturbances that have been identified. Of these image disturbances, ideally only moving objects are detected.

Table 2 Different types of image disturbances that have been identified. Of these image disturbances, only moving objects should be detected by the background subtraction and segmentation module.

Image disturbances:	Description:
Moving objects	Bubbles Particles: fibres, rubber, glass etc.
Static disturbances on the outside	Scratches, fingerprints, dirt, etc.
Semi-static bubbles	Bubbles stuck on the rubber stopper in a syringe, deforming between frames as affected by the liquid
The liquid meniscus	The meniscus with floating bubbles and reflections on the surface
Hard illumination disturbances	Reflections from the light source on the container walls and the meniscus
Soft illumination disturbances	Gradual changes to the light level of the background caused by light diffusion and the moving meniscus
Image noise	Image noise caused by the imaging sensor, this noise is normally distributed and is stronger with diagonal illumination as the analog gain is increased

Background intensity variance

In some regions of the image, the background intensity varies more between frames than in other regions. This is mainly caused by light reflecting on the moving meniscus at the top of the container and diffusing in the liquid, on the container walls, on scratches and on static debris.

The variation in background intensity can be observed by calculating the standard deviation of intensity over time. Figure 7 shows the standard deviation for a 10ml vial with diagonal illumination and back illumination, amplified for visibility. In the back illumination image, large regions can be seen that have an increased or decreased intensity variance. This is caused by reflection of the back light on the meniscus and the container walls. Thin vertical lines can also be seen, these are caused by vertical scratches on the container. In the back illumination image can be seen that dirt and static debris on the outside of the container can cause small regions with a larger intensity variation. Regions with a higher intensity variance pose a problem for global threshold segmentation methods used in previous work [15] [16] [7] [20], as these regions would lead to false object detections. If the segmentation threshold is increased to avoid these false detections, low contrast objects will not be detected.

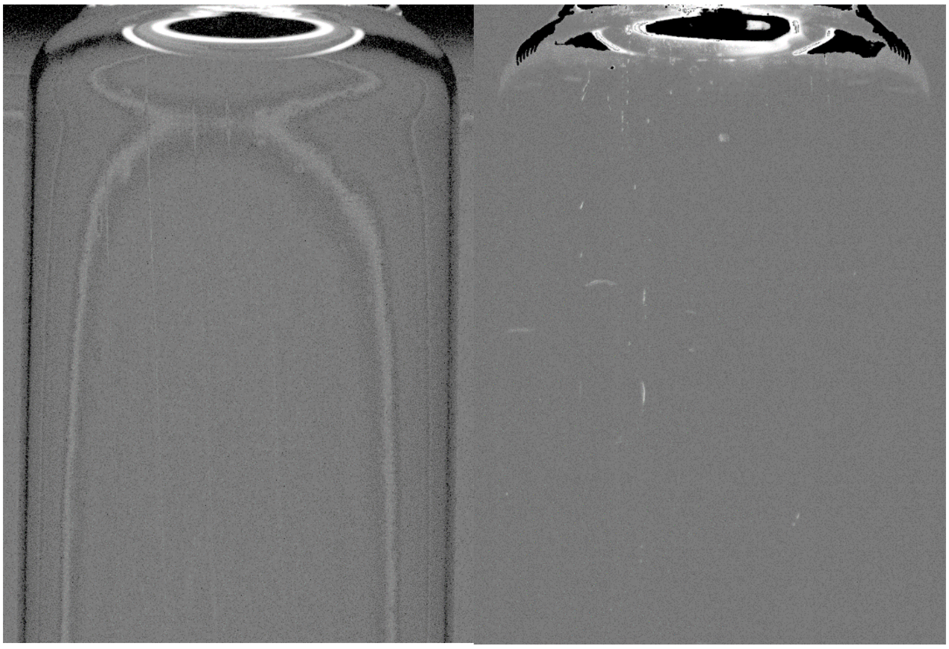


Figure 7 Standard deviation of intensity between frames, amplified for visibility. Regions that appear brighter in the image have a larger variance in pixel intensity. Left: Standard deviation of a 10ml vial with back illumination. Right: Standard deviation of the same 10ml vial with diagonal illumination. The figure shows vertical scratches and large vertical regions with increased background variance for back illumination, and static debris with increased background variance for diagonal illumination.

Object sparsity

As particles and bubbles are sparse, most pixels in an image belong to the background. In almost all cases, moving objects occupy between 0% and 0.5% of an image with an estimated average around 0.2%. Therefore, it is rare for a single pixel to contain a moving object multiple times over the full sequence of 20 frames.

Table 3 shows the probability for a single pixel to contain a moving object multiple times if moving objects are randomly distributed and cover 0.5% of the image.

Table 3 Probability that a moving object is present in a pixel for n frames out of 20.

Moving object present in n frames out of 20	Probability
0	90.4610%
1	9.0916%
2	0.4340%
3	0.0131%
4	0.0003%

The chance that a moving object is present in more than two frames is 0.0134%. From this observation the following assumption is made:

> *Assumption 1: The chance that pixel contains a moving object for more than two frames is negligible.*

Appearance of a foreground pixel

Image disturbances including moving objects appear darker or lighter than the background depending on the mode of illumination. In back illumination, disturbances appear darker than the background, whereas in diagonal illumination, disturbances appear brighter than the background.

There are rare exceptions to this rule, however this is not important as those particles can be detected better with the other illumination mode.

From this observation, the following assumption can be made:

> *Assumption 2: Compared to the background, a moving object appears darker for back illumination, and brighter for diagonal illumination.*

2.4 Proposed solution

This section presents the proposed solution for the background subtraction and segmentation module. The proposed solution first obtains a background model with a method named *Filtered temporal background modelling* (step 1 – 5). Next, background subtraction is performed using the average background intensity (step 6). Finally, object segmentation is performed with a method named *Locally adaptive threshold segmentation* and the background standard deviation (step 7 – 9).

Figure 8 shows an overview of the proposed solution. In following subsections, the workings of the methods for background modelling, background subtraction and segmentation are explained.

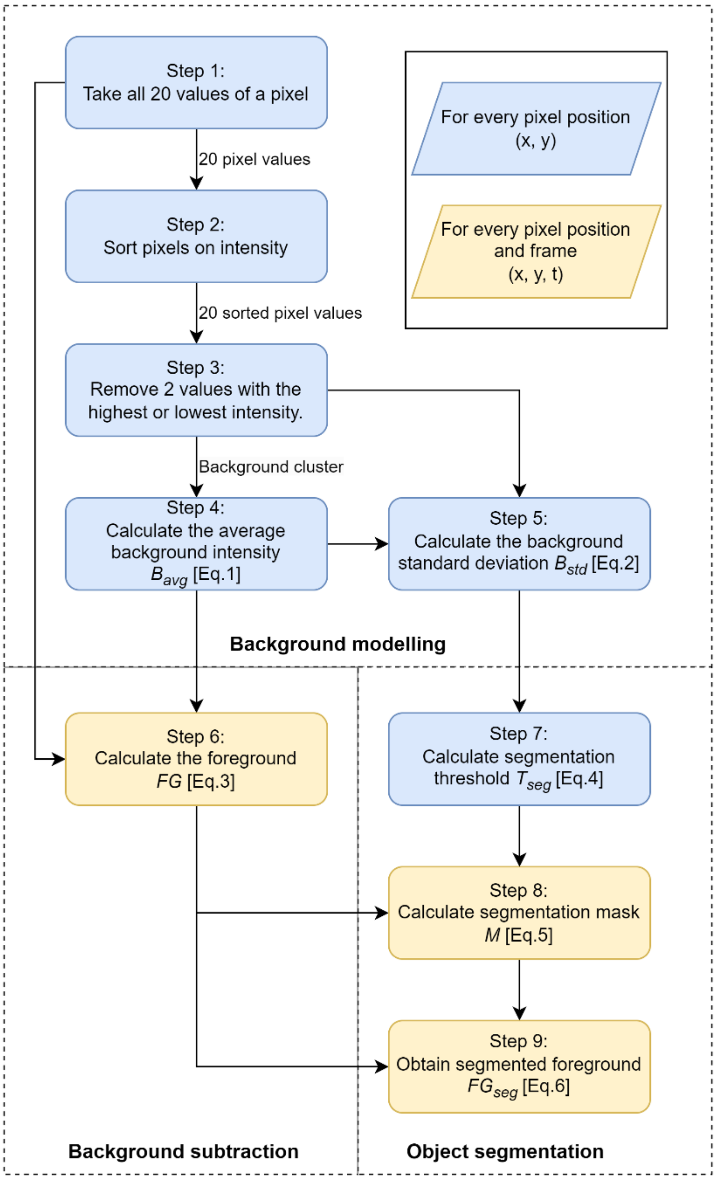

Step 1:
Take all 20 values of a pixel
20 pixel values
Step 2:
Sort pixels on intensity
20 sorted pixel values
Step 3:
Remove 2 values with the highest or lowest intensity.
Background cluster
Step 4:
Calculate the average background intensity B_{avg} [Eq.1]
Step 5:
Calculate the background standard deviation B_{std} [Eq.2]
For every pixel position (x, y)
For every pixel position and frame (x, y, t)
Background modelling
Step 6:
Calculate the foreground *FG* [Eq.3]
Step 7:
Calculate segmentation threshold T_{seg} [Eq.4]
Step 8:
Calculate segmentation mask *M* [Eq.5]
Step 9:
Obtain segmented foreground FG_{seg} [Eq.6]
Background subtraction
Object segmentation

Figure 8 Overview of the proposed solution for the background subtraction and segmentation module.

2.4.1 Filtered temporal background modelling

The proposed background modelling method is called Filtered temporal background modelling. This is shown in steps 1 – 5 in Figure 8.

Working principle

As found in assumption 1 on page 23, the chance that pixel contains a moving object for more than two frames is negligible. As found in assumption 2, pixel values from a moving object have a higher or lower intensity value depending on the illumination mode. These assumptions are combined in the working principle of filtered temporal background modelling.

> *Working principle:*
> *Removing the two lowest pixel values with back illumination, or the two highest intensity values with diagonal illumination, results in an incomplete but representative background cluster from which a background model can be computed for that pixel.*

Background model

The background model consists of the average background intensity BG_{avg} and the background standard deviation BG_{std} which are used for background subtraction and segmentation respectively. These are calculated using Equation 1 and Equation 2.

$$BG_{avg}(x,y) = \frac{\sum_{i=1}^{n} I_i(x,y)}{n} \tag{1}$$

$$BG_{std}(x,y) = \sqrt{\frac{\sum_{i=1}^{n}\left(I_i(x,y) - BG_{avg}(x,y)\right)}{n-1}} \tag{2}$$

The background cluster is incomplete as the two discarded pixel values likely also contain background values. However, this is not an issue as the remaining 18 pixel values can effectively enough represent the background in terms of mean intensity and intensity standard deviation.

Method benefit

A benefit of this method for background modelling is that the background intensity BG_{avg} which is used for background subtraction is obtained using 18 out of 20 pixel values, resulting in a temporal smoothened background image with minimal image noise. Another benefit is that this method also returns the background standard deviation BG_{std} which is used to perform segmentation.

2.4.2 Background subtraction

Background subtraction is performed by differencing each image to the average background intensity BG_{avg}. This is shown in step 6 in Figure 8. The process of background subtraction depends on the mode of illumination as this determines if moving objects appear darker or lighter than the background. The foreground image FG containing all image disturbances including moving objects is calculated using Equation 3.

$$FG(x,y) = \begin{cases} BG_{avg}(x,y) - im(x,y), & Back\ illumination \\ im(x,y) - BG_{avg}(x,y), & Diagonal\ illumination \end{cases} \tag{3}$$

2.4.3 Locally adaptive threshold segmentation

The proposed segmentation method is called Locally adaptive threshold segmentation. This is shown in steps 7 – 9 in Figure 8.

As observed in Figure 7, in some regions, the background intensity varies more than in other regions. This is problematic for global threshold segmentation methods as these regions can show up as false detections of moving objects. If the segmentation threshold is increased to avoid these false detections, segmentation becomes less sensitive and low contrast objects will not be detected.

Working principle

The working principle of locally adaptive threshold segmentation is that the background standard deviation is useful to determine a locally adaptive segmentation threshold. Such a locally adaptive segmentation threshold is low in regions where the background intensity is near constant and higher in regions where the background intensity fluctuates more.

First, the locally adaptive segmentation threshold T_{adapt} is defined for every pixel position by multiplying BG_{std} and a single sensitivity parameter value T_{param} using Equation 4. Next, the binary segmentation mask M and the segmented foreground FG_{seg} are calculated using Equation 5 and Equation 6.

$$T_{adapt}(x,y) = T_{param} * BG_{std}(x,y) \tag{4}$$

$$M(x,y) = \begin{cases} 1, & FG(x,y) > T_{adapt}(x,y) \\ 0, & FG(x,y) \leq T_{adapt}(x,y) \end{cases} \tag{5}$$

$$FG_{seg}(x,y) = FG(x,y) * M(x,y) \tag{6}$$

Method benefits

The main benefit of this method of segmentation is that it is locally sensitive or insensitive based on the temporal background model for each pixel. This should result in less false detections and more accurate low contrast detections compared to any global threshold segmentation method. Another benefit is that this method is configured using only a single parameter T_{param} that configures segmentation sensitivity. This makes it easy to reconfigure the solution if needed for other container types or liquids.

2.5 Results

This section presents a demonstration of the proposed solution for the background subtraction and segmentation module and presents the detection results on a set of cropped images.

Demonstration

Figure 9 demonstrates the proposed solution. The points in the graphs show the intensity value of a pixel position over time. Green points form the incomplete background cluster. The two extreme values excluded from the background cluster are shown in red. The graphs show how the locally adaptive segmentation threshold T_{adapt} is adjusted based on the variance of the background intensity.

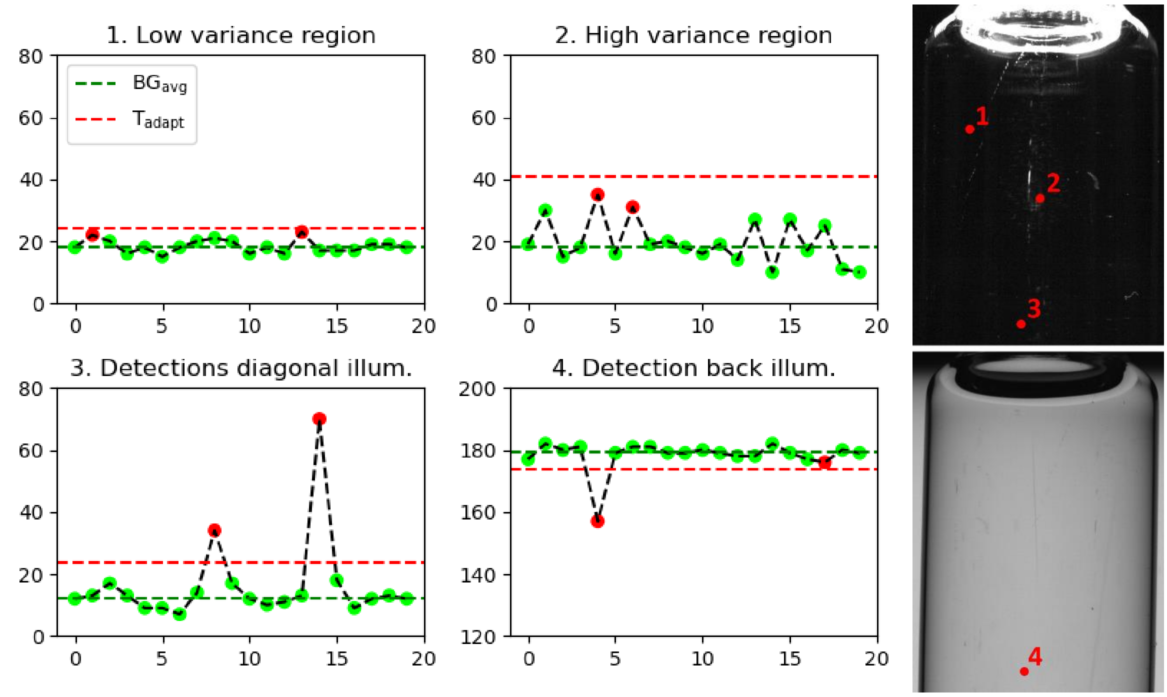
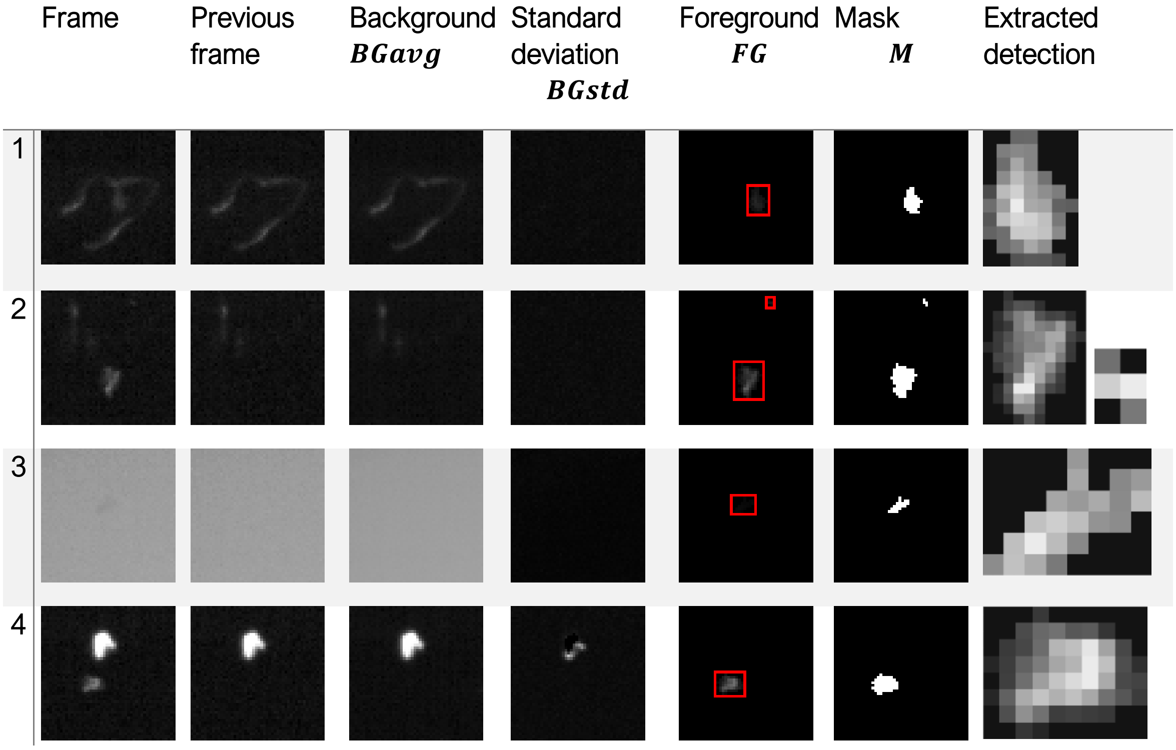


Figure 9 Demonstration of proposed solution. The graphs on the left show the image intensity over time in positions 1 – 4 in the images on the right. Red points are excluded from the incomplete background cluster shown with the green points. Points that are outside the red dotted line are detected as moving objects. Graphs 1 – 2 show how the locally adaptive segmentation threshold T_{adapt} adapts to the variance of background pixels. Graphs 3 – 4 show when a pixel value is detected as a moving object in both illumination modes.

Detection results

Table 4 shows some detection results from cropped images with a moving object. Each row shows results for a different challenging situation explained below the table. Some more results for low contrast detections are shown in subsection 2.6.1.

Table 4 Results from background subtraction and segmentation



Row 1 shows a moving particle passing in front of a large static fibre on the outside of the container. The segmentation of the moving object is accurate and is not affected by the fibre.

Row 2 shows two particles of different sizes. The smallest particle has an area of only four pixels and a low contrast to the background. In the sequence of 20 frames, the small particle is observed in 11 frames.

Row 3 shows a low contrast particle in back illumination. Even though the object is barely visible in the frame, the method detects the object with a high level of detail.

Row 4 shows a particle passing past a reflective static disturbance on the wall of the container. Over the sequence of frames, the amount of light reflecting of the lower edge of the static disturbance varies. This does not cause a false detection as the standard deviation in this region is increased.

2.6 Validation

This module will be validated by analysing the results and answering the sub-questions A.1-A.4 as presented in section 2.2. Each sub-question is validated in its own subsection.

2.6.1 A.1: Low contrast detections

This section aims to answer sub-question A.1.

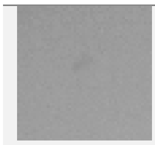
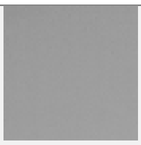
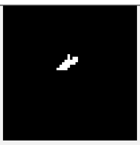
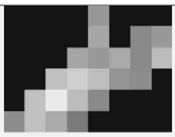
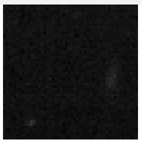
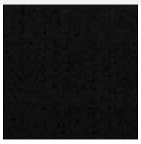
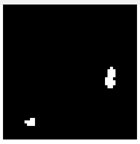
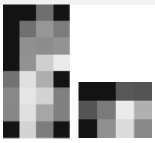
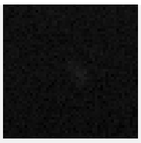
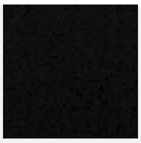
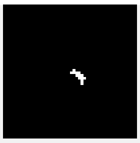
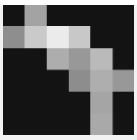
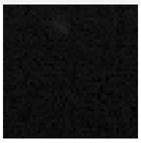
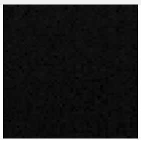
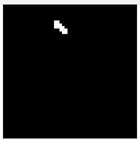
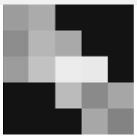
A.1: *"Can the solution detect objects with a low contrast to the background?"*

It is found that the background subtraction and segmentation module can detect objects that are smaller and have a lower contrast than what can be reliably classified as explained in section 3.6. Therefore, the method performs satisfactory on this aspect.

Detection of small low contrast objects is of importance for the limits of the VPIS in terms of the smallest or lowest contrast particles that can be classified. Ideally, the background subtraction and segmentation module should be able to detect objects that are too small or have too little contrast such that these do not contain enough information to be classified. In section 3.6 of the classification module, it is approximated that with the current method of classification, any particle detections with an area larger than 30 pixels and a total contrast higher than 700 are useful for classification and should therefore be detected.

Table 5 shows results on cropped image sections containing a small low contrast particle. Results include the calculated background, segmentation mask, extracted detection, area and total contrast of the detection. From the table can be observed that very small and low-contrast particles are accurately extracted from the images. As the extracted objects are smaller than 30 pixels and have a total contrast lower than 700, the solution exceeds the boundaries of what can be classified by classification module.

Table 5 Results from background subtraction and segmentation that show challenging low contrast objects

Frame	Background	Mask	Detection	Area	Total contrast	Mean contrast
				22	177	8.0
				23 9	389 176	16.9 19.6
				14	152	10.9
				14	186	13.3

2.6.2 A.2: Insensitivity to disturbances

This section aims to answer sub-question A.2:

A.2: *“Is the solution insensitive to image noise, static disturbances, or soft illumination disturbances, such that approximately less than 1% of detections are caused by these image disturbances?”*

On average, a container image sequence results in approximately 100 detections. Therefore, for A.2 to be true, approximately a single detection in an image sequence can be caused by image noise, static disturbances, or soft illumination disturbances. It is found that the proposed solution performs satisfactory in this aspect if detections smaller than 10 pixels are ignored.

Detections caused by image noise, static disturbances, or soft illumination disturbances can be analysed by observing an image sequence without any particles or bubbles. When no moving objects are present in the image, any detections that are still made are caused by these image disturbances.

An image sequence without any bubbles can be obtained by performing a long pre-spin sequence (see subsection 3.4.2). An image sequence without any particles can be obtained by using a clean container sample. Within the scope of this thesis, a completely clean container sample cannot be used as all container samples have been filled manually by the author without a cleanroom environment. Therefore, the image sequences used to evaluate sub-question A.2 contain some tiny particles that are detected with diagonal illumination as detections with an area between 3 - 10 pixels.

Below, it is discussed how many detections are caused by image noise, static disturbances or soft illumination disturbances for diagonal illumination and back illumination.

Diagonal illumination

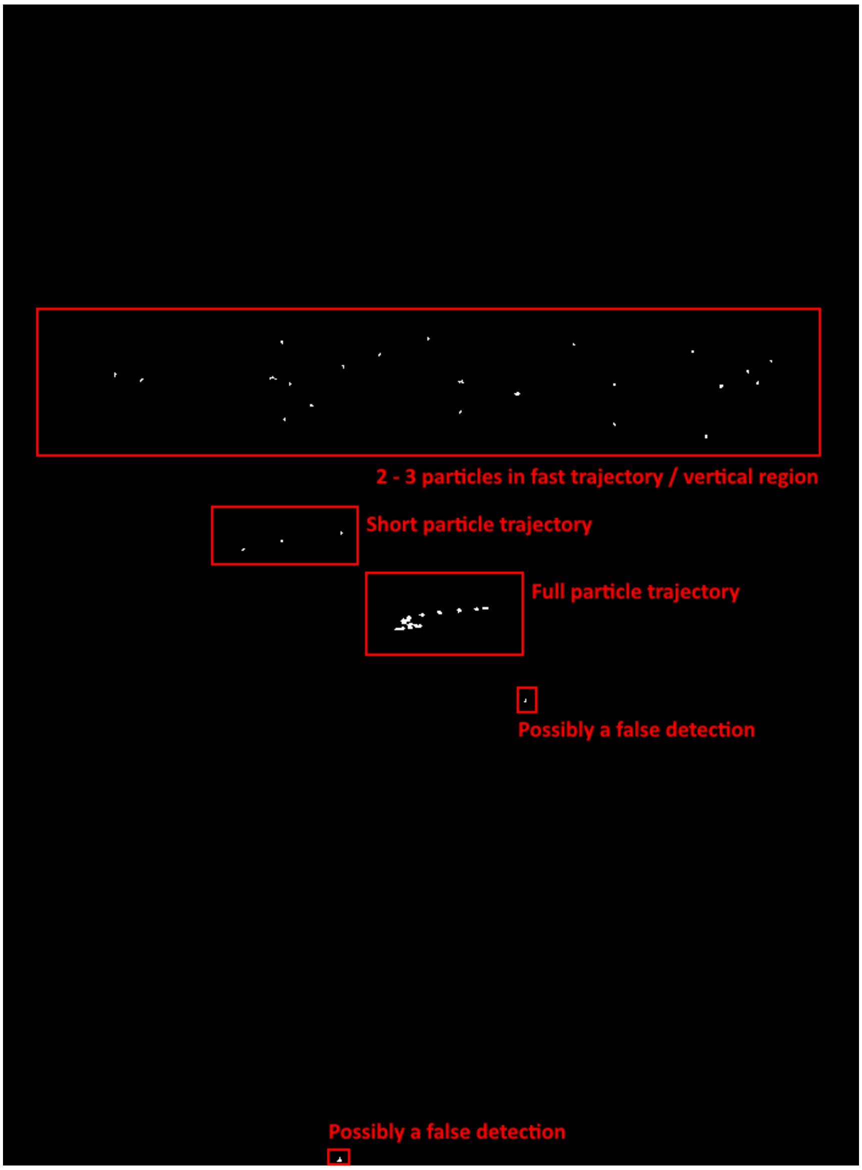

2 - 3 particles in fast trajectory / vertical region
Short particle trajectory
Full particle trajectory
Possibly a false detection
Possibly a false detection

Figure 10 Annotated combined segmentation mask of an image sequence with diagonal illumination of a mostly clean container after a pre-spin sequence to remove bubbles. The figure shows 46 small detections of which most can be contributed to small particles present in the container sample.

Figure 10 shows the combined segmentation mask of a sequence with diagonal illumination. In total, 46 detections are made between 3-10 pixels. It is likely that most of these detections are caused by tiny particles that are present in the container. This is supported by the observation that most detections are on a clearly recognisable trajectory or are clustered in a horizontal region indicating multiple fast trajectories.

Excluding detections that can be explained with the presence of small particles, very few detections are present that are possibly caused by image noise, static disturbances, or soft illumination disturbances. None of these detections are larger than 10 pixels.

Back illumination

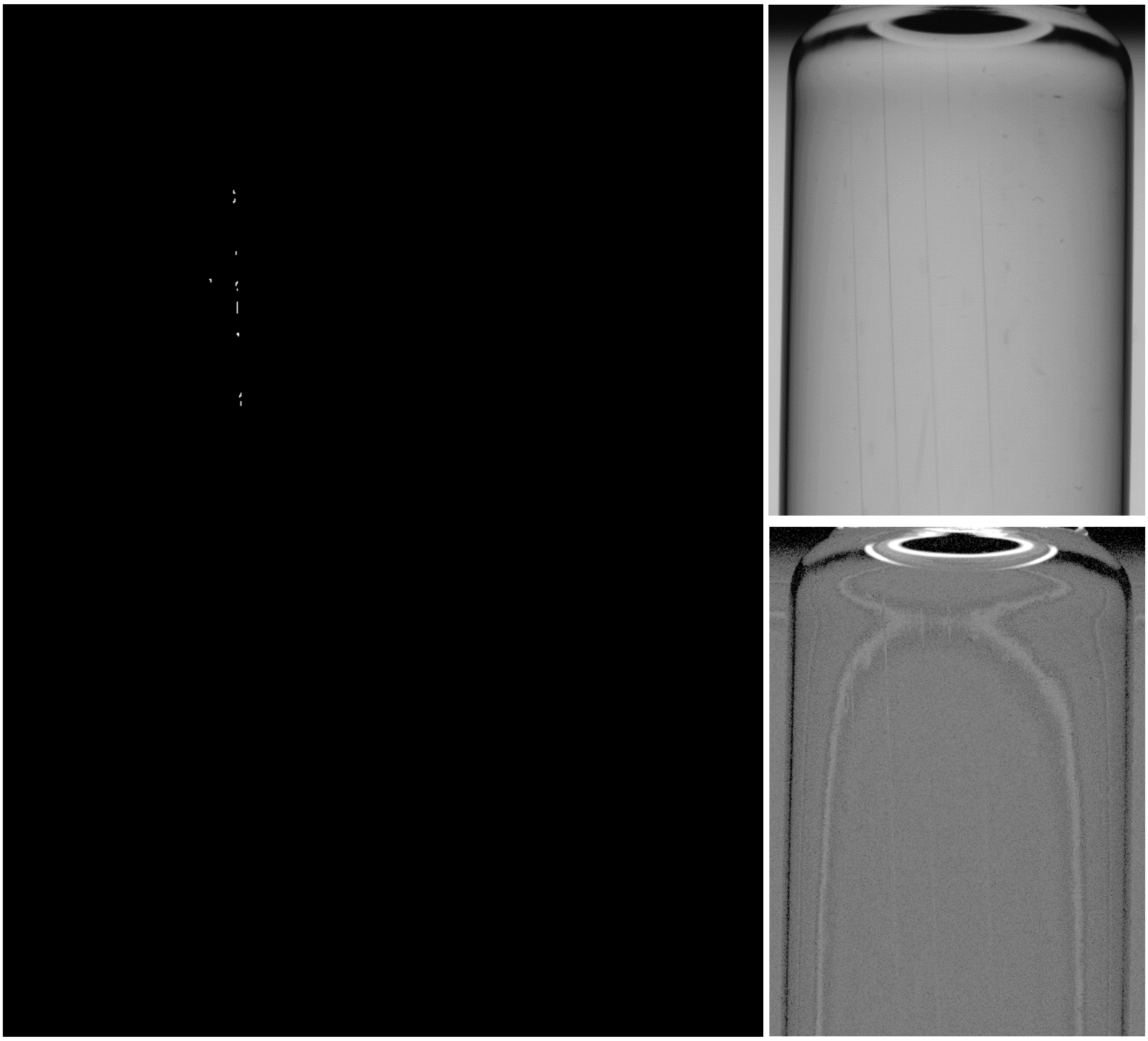


Figure 11 Left: Combined segmentation mask of an image sequence with back illumination of a mostly clean container after a pre-spin sequence to remove bubbles. Top-right: Average background BG_{avg} of the image sequence. Bottom-right: Background standard deviation BG_{std} of the image sequence, amplified for visibility. The figure shows a number of small object detections of which most are caused by a scratch, a static image disturbance.

Figure 11 shows the combined segmentation mask of a sequence with back illumination, together with the average background BG_{avg} and the background standard deviation BG_{std} amplified for visibility. In the figure it can be seen that less detections are made compared to Figure 10 with diagonal illumination, this is because dust particles present in the container are less visible with back illumination. The singular detection of 4 pixels left of this line is caused by either a tiny particle or image noise. The vertical line of detections is caused by illumination variation on one of the vertical scratches on the container.

In most cases, a vertical scratch should not result in false detections as in those regions the background standard deviation is increased. These anomalies can occur if a static disturbance reflects significantly more light for one or two frames or if a single frame is translated as a result of vibrations acting on the machine. From testing with 52 image sequences, such an anomaly has been observed in five sequences. In total, three detections larger than 10 pixels have been observed.

A.2: Conclusion

For A.2 to be true, approximately a single detection in an image sequence can be caused by image noise, static disturbances, or soft illumination disturbances.

From the results presented in this section it can be stated that a small number of detections are caused by variations on soft lighting or image noise. If detections smaller than 10 pixels are excluded, it can be stated with confidence that on average less than one detection per image sequence is caused by image noise, static disturbances, or soft illumination disturbances.

In section 3.6 of the classification module, it is found that detections smaller than approximately 30 pixels cannot be classified reliably. Therefore, it is not problematic if detections smaller than 10 pixels are ignored.

2.6.3 A.3: Processing time

This section aims to answer sub-question A.3:

A.3: *"Can background subtraction and segmentation be performed on two sequences of 20 images within one second, as not to slow down the VPIS?"*

The average processing time of a sequence of 20 frames is 0.42 ± 0.07 seconds. Therefore, the average processing time of two image sequences is below one second.

This test was performed on a laptop[1] with python with the images are already loaded into memory. Processing time was measured a total of 200 times with image sequences of 10ml vials with a resolution of 750x600 pixels. The current version of the background subtraction and segmentation module has been partially optimised to lower processing time. Section 2.9 shortly lists the optimisations that are currently implemented and discusses recommendations for how the processing time could be further decreased.

2.6.4 A.4: Parameter reconfiguration

This section aims to answer sub-question A.4:

A.4: *"Can the method easily adapt to different container types or types of illumination?"*

It is found that the proposed solution is easy to adapt to different container types and modes of illumination. This is as the only configuration parameter does not need to be reconfigured for all imaging situations observed so far.

The proposed solution contains only a single parameter T_{param} which controls segmentation sensitivity. As the proposed segmentation method is locally adaptive, changes in the imaging situation are automatically reflected in the local segmentation threshold without changing T_{param}. The proposed solution has been used effectively in the following imaging situations without changing T_{param}. For all tests performed in this research, T_{param} has been set to 4.

- Vial 10ml, back illumination
- Vial 10ml, diagonal illumination
- Syringe 5ml, back illumination
- Syringe 5ml, diagonal illumination

[1] Laptop specs: Intel i7-8750H CPU and NVIDIA Quadro P1000 GPU

2.7 Comparison to SOTA

This section compares the proposed solution to the current SOTA for this module, as described in section 2.1. Although the solution cannot be directly compared to the SOTA by comparing results, it is expected that the proposed solution outperforms the current SOTA.

Background modelling with Gaussian Mixture Models

When compared to the SOTA method background modelling with Gaussian Mixture Models (GMM) [16], it is expected that the proposed solution outperforms the SOTA in terms of processing time.

With the SOTA method, a GMM is fitted for every pixel in the image which is an iterative process with a long processing time. For this reason, most applications of GMM background modelling use low-resolution images or are used for application where processing time does not matter. In comparison, for the proposed method, the operations that take the most time are excluding the two highest or lowest values from the background cluster and determining the background standard deviation which take a fraction of the time in comparison. In initial tests it was confirmed that GMM background modelling is not suitable for this problem due to the processing time.

Simple background modelling

Compared to the SOTA method simple background modelling [15], the proposed method performs better in terms of detection of low contrast objects, insensitivity to disturbances, and ease of parameter configuration. The SOTA method performs background modelling in a similar fashion and obtains an incomplete background by comparing intensity values over time. However, instead of assuming that an object does not appear in a pixel twice in the image sequence, the SOTA assumes that an object does not appear twice in a row. As a result of this, the SOTA requires a threshold parameter to determine if two detections are similar enough which is not adaptive and will therefore fail in regions with a high background variance. In low variance regions, the method will be less successful at detecting low contrast objects. As the method requires more parameters also to perform subsequent segmentation, the method is harder to reconfigure compared to the proposed solution.

Local adaptive threshold segmentation

The SOTA method local adaptive threshold segmentation [9], proposes a locally adaptive threshold based on a global threshold and the local neighbourhood around the pixel. In comparison, the proposed method determines the locally adaptive threshold based on the values of that pixel over time and does not combine this with a global threshold parameter. The resulting threshold performs better with variance in illumination and to detect objects with a low contrast gradient contour. Additionally, the proposed solution can be reconfigured more easily as it has only a single parameter that is insensitive to most changes.

2.8 Conclusion

In this chapter, findings have been presented regarding the research and development of the background subtraction and segmentation module. The proposed solution is shown to be effective through the validation of detection of low contrast objects, insensitivity to image disturbances, processing time, and parameter reconfiguration. The proposed solution has been successfully implemented and has been used during data collection for both the classification and tracking modules. The proposed solution including a version of the add-on solution for detecting objects near the rubber stopper presented in Appendix A has already been adopted by Luo Automation B.V.

The proposed solution can detect small objects with a low contrast while detecting few false detections caused by image noise, static disturbances, or soft illumination disturbances. The current implementation of the module can process two sequences of 20 frames in less than 1 second on average, making it fast enough to be used with the VPIS without slowing down the inspection process of a container. The proposed solution has only a single parameter that controls sensitivity of the locally adaptive threshold segmentation. As the threshold is locally adaptive, has been used with both 10ml vials and 5ml syringes in both back and diagonal lighting without requiring reconfiguration of the sensitivity parameter.

2.9 Recommendations

The solution presented in this chapter for the background subtraction and segmentation module is effective according to the current requirements for the VPIS specified roughly in research sub-questions A.1 – A.4. It might however be beneficial to further improve the proposed solution to perform better in the aspects of processing time, insensitivity to disturbances, and detection of objects in challenging regions. This section discusses reasons to further improve the proposed solution and suggests strategies to achieve this.

2.9.1 Processing time

By reducing the processing time of the background subtraction and segmentation module, more time will be made available to perform the other module tasks of the VPIS. Reducing the processing time of this module also makes it possible to detect moving objects in images with a higher resolution. With the optimisations shortly discussed below, it is estimated that the processing time can be reduced by 50%

Current optimisations

The current version of the proposed solution is implemented in python and has been partially optimised to lower processing time. All large operations are performed using full images instead of singular pixels as proposed in section 2.4. The operations of calculating the average background, background standard deviation, unfiltered foreground, and performing segmentation are performed on the GPU which significantly lowers processing time for some steps.

General optimisation

As the current version of the proposed solution is implemented in python, it is expected that with general optimisation strategies the processing time can be reduced up to 50%. General optimisation strategies include implementation of the code in a compiled language such as C++, performing more tasks on GPU, implementation of multiprocessing for CPU tasks, and efficient memory management.

Unnecessary steps

The current implementation of the module performs steps that are useful during development but are not necessary during regular operation. Each detection is extracted as a class instance and saved for later usage. In addition to the extracted foreground and pixel position which are used by the classification and tracking modules, each class instance includes a segmentation mask, image metadata and a total of 40 handcrafted features listed in Appendix B. Removing the unnecessary information from the class instances, and keeping the class instances working memory, the processing time should be significantly improved.

2.9.2 Insensitivity to disturbances

In section 2.6.2 can be seen that in some cases, false object detections are made from disturbances such as scratches on the outside of the container. Although the current results are acceptable as most false detections are too small, it is recommended to further reduce the number of false detections.

Static image disturbances

Static image disturbances can result in false detections if a pixel is affected by the static disturbance for two frames or less. This can happen due to vibrations on the machine that result in translation of the disturbance or if a disturbance is only visible in the first few frames when the water surface is deformed more strongly.

Possible solution

A possible partial solution for this problem can be to use a local maximum filter on the background standard deviation BG_std. As a result of this, regions with a high background variance will result in an increased locally adaptive threshold in neighbouring pixels. This should prevent most false detections caused by vibrations. Future work is needed to test this theory.

2.9.3 Detection in challenging regions

One of the shortcomings of the proposed solution for the background subtraction and segmentation module is that the method can only detect objects in the centre region of a container. This is as the regions near the top and bottom of a container offer challenges make it difficult to perform detection.

Add-on solutions

Appendix A presents an add-on solution that enables the solution for the background subtraction and segmentation module to detect objects near the rubber stopper in the bottom region of a container.

It is recommended for future work to further improve this add-on solution and investigate methods to detect objects in other challenging regions. If the solution is able to detect objects in more challenging regions such the bottom region of a vial or in the top region of a vial or syringe near the liquid surface, this can help improve patient safety and therefore be of significant value.

3 Classification module

The second module of the VPIS is the classification module. This chapter contains research and findings regarding the classification module.

The goal of the classification module is to classify detections of moving objects that have been extracted from a sequence of images by the background subtraction and segmentation module. If a particle is classified, the container should be rejected.

Ideally, every detection is correctly classified as a particle or a bubble, such that all contaminated containers are rejected, and clean containers accepted. However, this is challenging some objects can be very small such that the detections have an area of only a few pixels with a low contrast. Additionally, many detections of bubbles can be detected from the sequence of images. If a single detection classified as a particle would result in the container being rejected, this could result many false container rejections.

The goal of this research into the classification module is not to develop a classifier that is good enough to comply with regulations and be implemented in the VPIS. But rather to select and develop a decent baseline classifier and explore the possibilities and pitfalls that come with this problem situation. Besides selecting a classification method out of four candidates and training a classifier, exploratory research is performed into three other topics:

- Correlations between handcrafted features and classification accuracy that can be used to filter out detections that cannot be classified reliably.
- The effectives of the classifier to classify a container with multiple detections.
- The theoretical effectiveness of two proposed multi-detection classification strategies.

During this early stage in the development of the classification module, the classification module will be trained and evaluated exclusively with detections from image sequences of 10ml vials with diagonal illumination. However, the proposed solution also should work with back illumination or different container types.

This research proposes a baseline classification method with a CNN classifier that achieves a classification accuracy of 93% on single detections with an area of at least 30 pixels. Besides that, a filtering method is proposed that predicts the classification accuracy for a detection to filter out detections that are too hard to classify. The proposed filtering method uses observed complex correlations between multiple handcrafted features and the classification accuracy and outperforms a set of simple threshold filters in terms of resulting classification accuracy while classifying more detections.

It is shown through simulation that the naïve classification strategy where a single detection classified as a particle results in the container being rejected is not valid. Therefore, two multi-detection classification strategies are proposed: median voting classification with tracking and multi-positive classification. Both multi-detection classification strategies achieve significant improvements in terms clean containers that are falsely rejected and contaminated containers that are falsely accepted. Median voting classification uses the results of the tracking module to classify multiple detections of a moving object and achieves the highest accuracy. Multi-positive classification requires multiple detections to be classified positive before the container is classified positive.

3.1 Chapter structure

This chapter presents findings regarding the research and development for the classification module. First, section 3.2 presents a short summary of previous work into classification for the inspection of medicine. A selection of current State-Of-The-Art (SOTA) methods is presented together with shortcomings and take-aways. Next, in section 3.3, the research questions are presented for the baseline classifier and subsequent exploratory research. After that, section 3.4 presents how the dataset was created for the classification module. Section 3.5 presents the selection of a baseline classifier and its classification accuracy on the dataset. In section 3.6, correlations between handcrafted features and the classification accuracy are analysed to define a filtering method for detections that cannot be classified reliably. Section 3.7 demonstrates through simulation that the default classification strategy is not viable. Therefore, in section 3.8 two multi-detection strategies are proposed and simulated. In section 3.9, the proposed solution is compared to existing methods. Finally, sections 3.10 and 3.11 present the conclusion and recommendations for this module.

3.2 Previous work

An analysis of previous work for the problem of particle classification has been performed during a literature study [6] preceding this thesis. This section summarises the most interesting findings from the literature study relevant for the classification module. Next, current State-Of-The-Art (SOTA) methods are identified in more detail. In section 3.9, the solution proposed for the classification module is compared to the SOTA methods.

Classification methods

The classification methods referred to in the literature study [6] can be separated into six groups: trajectory classification [18] [7] [20] [12] [8] [10] [22] [15] [23] [21], shape classification [13], wavelet classification [25] [26], handcrafted feature classification [9] [16] [26], 3D-ConvNet classification [27], and Multi-view classification [28]. Multi-view classification has not yet been used in literature for the problem of particle classification but is included as it can be useful for this purpose. Of these methods, it is expected after the literature study that Multi-view classification [28] should perform the best, followed by handcrafted feature methods [9] [16] [26].

3.2.1 Current SOTA

This section explains the workings and shortcomings of the identified SOTA methods: Multi-view classification [28] and handcrafted feature methods [9] [16] [26]. It also explains the workings and shortcomings of trajectory classification as this classification method is applied in the most papers.

Trajectory classification

Different forms of trajectory classification are applied in ten papers [18] [7] [20] [12] [8] [10] [22] [15] [23] [21]. These methods work based on the premise that the trajectory contains useful information about the object class. It is assumed that bubbles will rise to the surface, whereas particles should sink.

Although classification based on trajectory has been widely adopted in previous work, experimental observations using our own setup show that the turbulent vortex created after spinning the liquid causes both lightweight particles and bubbles to deviate randomly in vertical direction. Even after the liquid has slowed down, some lightweight particles such as polystyrene foam can rise at a similar rate as bubbles. Classification by object trajectory alone is therefore considered unreliable and inaccurate and should therefore not be used.

Handcrafted feature classification

Handcrafted feature classification methods [9] [16] [26] perform classification based on numeric features that have been extracted from object detections. Good handcrafted features are descriptive of a detection and include features such as: area, length-width ratio, mean contrast and circularity of the contour. Three papers in previous work propose a classification method based on handcrafted features. Two of these methods are considered as SOTA methods for particle classification and are described below.

Handcrafted feature classification using OS-ELM [26] extracts a total of 17 handcrafted features. Next, classification is performed using a method called OS-ELM. OS-ELM is a rather controversial algorithm [29] that behaves similar to a regular feedforward neural network (NN) but is trained in one iteration using the matrix inverse. In the paper, the classification results are significantly improved by performing tracking and including the direction of the trajectory as a handcrafted feature. The reported accuracy to distinguish between particles and bubbles is 99.83%. The paper does not state the size of objects present in the sample set.

As explained earlier for trajectory classification, early experiments using our own setup show that both bubbles and particles can deviate randomly in vertical direction. It is therefore questionable whether these results can be achieved in reality. The main benefit of the OS-ELM classification algorithm is the short duration of the training compared to a regular NN. As training should be performed beforehand, this is not of any importance.

Handcrafted feature classification using a genetically optimised neural network [16] performs classification based on a total of 6 handcrafted features. In the paper, 15 handcrafted features are extracted after which ReliefF feature selection is used to select 6 best features. Next, classification is performed using a neural network optimised with a Mind Evolutionary Algorithm. The reported False Positive and False Negative rates are 1.25% and 1.10% respectively. The paper does not state the size of objects present in the sample set. The paper uses 250ml plastic containers with a lower transparency compared to glass, therefore it can be assumed the particles are of a larger size.

A shortcoming in both papers is that these do not mention the size of particles in terms of pixels of physical dimensions, nor are close-up images of particles shown from which this can be derived. Therefore, it is hard to evaluate the effectiveness to classify small challenging particles. Another shortcoming is that the papers do not mention the effect of multiple bubbles that are detected in multiple frames. As explained in this research in section 3.7, the presence of multiple detections of bubbles can be of great consequence for the classification outcome. The exotic classification algorithms used in both papers are not often referred to in other literature as valuable improvements over a regular neural network.

Multi-view classification

Multi-view classification has not yet been used in literature for the problem of particle classification but is included as it can be useful for this purpose.

Multi-view classification with a convolutional neural network (CNN) is proposed by Seeland and Mäder [28] as a powerful extension to general CNN classification methods. The method combines multiple observations of the same object by fusing the latent representations in the CNN or by combining classification scores after individual classification. The paper shows that existing methods can be extended with the fusion strategy to outperform regular classification on existing data sets.

3.2.2 Take-aways

Although none of the SOTA in previous work are ideal, some useful take-aways can be made. The main take aways are the use of handcrafted features with feature selection and combining multiple detections to perform classification.

Different sets of handcrafted features are used for classification in previous work [9] [16] [26], a selection of the handcrafted features used by those methods Is adopted inn this research. The list of handcrafted features used in this research can be found in Appendix B. Relief [30] feature selection will be used to perform feature selection, similarly to one of the SOTA [16] which uses the multi-class variant called ReliefF [30].

At this early stage in the development of the classification module and the tracking module, it was chosen not to perform further research into a classification method with latent representation fusion as this cannot be tested or simulated effectively without a functional tracking module. However, the paper shows that for some classification problems, similar or superior results can be obtained using score fusion with addition or multiplication. As outlier detections are a significant problem for this classification problem, it was found in early experiments that median voting classification performs superior to addition and multiplication fusion methods proposed in the paper.

3.3 Research questions

This section introduces the research questions for the classification module. The research questions of this module are parts of a main research question for validation as with the background subtraction and segmentation module. The first research question B.1 results in a baseline classification method for classifying singular detections. Following research questions B.2 – B.4 are exploratory and explore opportunities and pitfalls for this classification problem. Each research question is evaluated separately in sections 3.5 – 3.8 with separate conclusions.

B.1: Classification model selection
"Which of the candidate classification methods achieves the highest classification accuracy for single detections on the dataset described in section 3.4?"

B.2: Classifiable detections
"What correlations exist between handcrafted features and the classification accuracy of a detection that can be used to filter detections that cannot be reliably classified?"

B.3: Full container classification
"What is the theoretical effectiveness of the classifier to classify a container with multiple detections, if a single positive detection results in the container being rejected?"

B.4: Multi-detection classification strategies
"What is the theoretical effectiveness of a classification strategy that uses multiple detections to classify a container sample?"

3.4 Dataset creation

This section describes how a dataset was created for the classification module. Most classification methods that are trained through the process of machine learning require a dataset with multiple instances of the objects/detections that are to be classified. With supervised learning, the different classes are labelled separately in the training data such that the classifier learns to recognise the difference most effectively. With this classification problem, the two main classes that need to be classified are bubbles and particles. In some previous work classification is also performed between different particle types or a class with false detections caused by illumination disturbances.

The goal of dataset creation is to obtain a dataset with labelled detections of particles and bubbles. The dataset should be large enough to train the classifiers described in section 3.5 and contain multiple detections of different particle types in different sizes. First, physical samples of containers with particle contaminations are created. Next, detection data is collected using the background subtraction and segmentation module and labelled such that detections of bubbles and particles are separated. In this research the classification module is only evaluated with 10ml vials with diagonal illumination, therefore the dataset is collected from that capture situation exclusively.

The resulting dataset contains a total of 142223 detections collected from 570 image sequences. A histogram with the distribution of detection area in the dataset is shown in Figure 12.

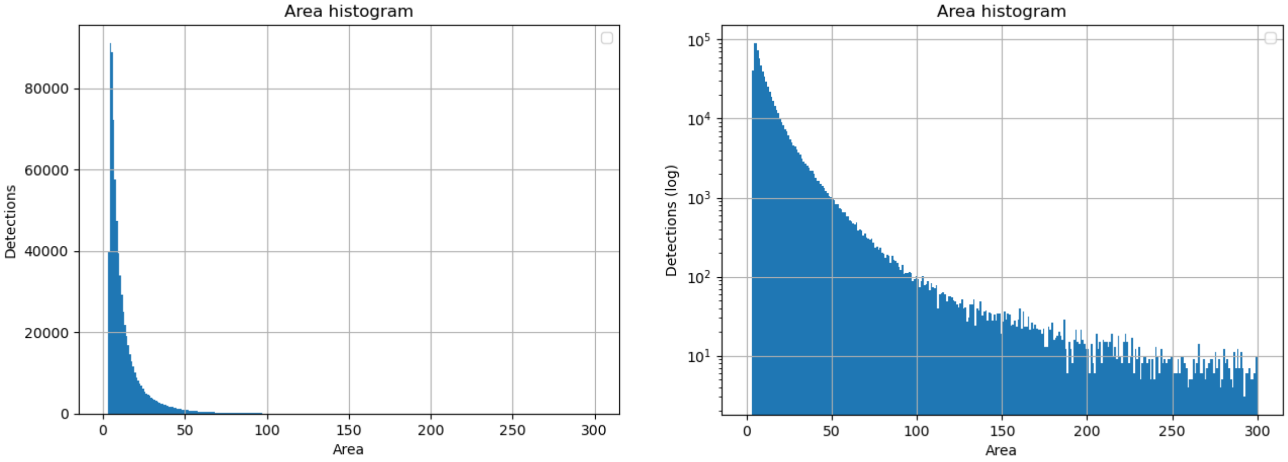


Figure 12 Histogram of detection area in the dataset in linear and logarithmic scale. The figure shows that most detections in the dataset are small.

3.4.1 Sample creation

Before a dataset can be collected, a sample set is required. In this research, a sample is defined as a physical medicine container. A contaminated sample is a medicine container that contains one or more particles. At the time of starting this research, no suitable sample set for 10ml glass vials was present at Luo Automation B.V.

Companies that perform manual visual inspection use an evaluation sample set during the training and periodic testing of their workers [31]. A good evaluation sample set contains good samples, samples with static defects, and contaminated samples and is representative of a real inspection situation. Contaminated samples contain particles of various sizes and types, generally with not more than one particle per sample as more particles make it easier to detect the contamination [2]. Every sample in a good evaluation sample set has been characterised with detail and registered.

Production of an evaluation sample set takes time and effort. Alternatively, an evaluation sample set can be purchased from a specialised vendor.

As time and resources available for this research are limited and development of the classification module is in an early stage, it has been decided to create a limited sample set of 10ml vials with the following properties:

- The sample set contains 20 sample containers, including 19 contaminated samples and 1 clean sample. The clean sample contains some tiny contaminations that are not visible to the naked eye but are detectable with the background subtraction and segmentation module. As detections of these objects have a maximum area of 3-10 pixels, smaller than the area threshold of 30 pixels approximated in section 3.6, this is not a problem. Efforts to create a clean sample without such contaminations have failed as no cleanroom was available.
- The sample set contains 1-10 particles per contaminated sample container. There can be more than one particle present as the samples are not used for validation. The number of particles is limited such that it is feasible to manually observe and annotate particle trajectories for the tracking module.
- The sample set contains mostly challenging particles with an area between 5-200 pixels. This is as larger particles can already be classified with high accuracy using an existing classification method. By limiting the range of particle size, more variation can exist in other aspects such as particle shape and material with the same number of samples. The distribution of detection sizes is shown in Figure 12.
- The sample set includes fibres, glass fragments, dust flakes and fragments of various types of plastic.

As the limited sample set contains mostly small challenging particles, the resulting classification accuracy is expected to be lower compared to when using a sample set with a wider range of particle sizes.

3.4.2 Dataset collection and labelling

A method has been found to collect detections of either bubbles or particles separately. This way, detections can be labelled directly such that detections do not need to be labelled manually afterwards. This subsection describes the method and process of collecting labelled data of bubbles and particles.

Bubble capture sequence

To obtain detections of exclusively bubbles, a clean sample container is used that contain only tiny particle contaminations smaller than 10 pixels. Any detections from this container larger than 10 pixels should be of bubbles exclusively. To collect these detections, the following steps are performed:

1. The clean container is lightly shaken to introduce bubbles and loaded into the machine.
2. The normal spin-stop capture sequence described in section 1.4 is repeated 3-5 times until most bubbles are gone.
3. Steps 1 and 2 are repeated until the desired number of image capture sequences is reached.

4. Detections are extracted from the image sequence using the background subtraction and segmentation module. Detections near the top of the container are excluded.

Particle capture sequence

To obtain detections of exclusively particles, contaminated container samples are used with a long pre-spin sequence to remove bubbles. During the long pre-spin sequence, bubbles that are stuck on the container walls are released and float to the top. Detections that are made from image sequences captured after the long pre-spin sequence are of bubbles exclusively.

1. A contaminated container sample is loaded into the machine.
2. The long pre-spin sequence is performed:
    - 500ms spin at 800 rpm
    - Waiting period of 120 seconds
    - 500ms spin at 500 rpm
    - Waiting period of 120 seconds
3. The normal spin-stop capture sequence described in section 1.4 is repeated until the desired number of image capture sequences is reached.
4. Detections are extracted from the image sequence using the background subtraction and segmentation module. Detections near the top of the container are excluded.

The effectiveness of the pre-spin sequence to remove bubbles has been confirmed with the use of the clean container sample. After the long pre-spin sequence, no moving objects larger than 10 pixels area can be observed in subsequent image sequences, meaning that all bubbles have been removed successfully.

3.5 B.1 Classification model selection

This section aims to answer research question B.1.

B.1: *“Which of the candidate classification methods achieves the highest classification accuracy for single detections on the dataset described in section 3.4.2?”*

In this section, a classification method will be chosen out of a list of four candidate classification methods to classify singular detections. The four candidate classification methods are trained and evaluated. Important outcomes are the chosen classification method and the classification accuracy on the test dataset. The trained classifier of the chosen classification method will serve only as a baseline classifier and will not be extensively optimised. Table 6 shows an overview of the four candidate classification methods. Of these classification methods, CNN classification achieves the highest classification accuracy of 0.93 or 93%.

Table 6 Overview of candidate classification methods with the type of features and training dataset

Name	Features	Training dataset
CNN	Segmented foreground resized to 32x32	Bubbles and particles with area >= 30
NN	Handcrafted features selected with Relief [30]	
SVM		
OC-SVM		Only bubbles with area >= 30

These candidate methods are presented in the next three subsections. Followed by the results in terms of classification accuracy and a conclusion in which answers research question B.1.

3.5.1 OC-SVM

OC-SVM stands for One-Class Support Vector Machine [32]. This method is a special variant of the better known SVM that can detect novelty between samples in a dataset. An OC-SVM is trained using detections from only a single class. After that it can classify whether a detection belongs to the trained class or a different class.

The great benefit that makes OC-SVM and other one-class classifiers interesting is that, if effective as a classifier, it can be trained using detections of only bubbles. As explained in subsection 3.4.2, detections of exclusively bubbles can be obtained easily using only a single clean container. This contrasts with detections of exclusively particles for which a diverse set of samples has to be created. This would make it possible to quickly and inexpensively expand the portfolio of medicine containers and that can be inspected.

A downside of OC-SVM and other one-class classifiers is that these methods usually obtain a lower classification accuracy than classification methods that use training data of all relevant classes. For some classification problems and feature spaces, OC-SVM can perform terrible compared to a regular SVM trained on all classes. This is as for some regions in a feature space, difference between two classes might be small such that the exact placing of the decision boundary is important to differentiate between the classes. Whereas in other regions, the decision boundary can be relaxed to include more diverging samples. This information is known to a regular SVM but is unknown to an OC-SVM.

3.5.1.1 OC-SVM in this research

The OC-SVM tested in this research uses handcrafted features selected with Relief [30] feature selection as explained later for the NN and SVM classification methods.

During initial testing, it has been observed that OC-SVM is not effective as a classifier for this classification problem. As the appearance of bubble detections is diverse, and there is a large overlap between detections of bubbles and particles, the method was not able to define a good decision boundary and only achieved a classification accuracy below 70%. Other one-class methods have also been tested briefly but none exceeded 70% classification accuracy.

As the performance decrease between OC-SVM and the other classification methods discussed in this chapter is too large, OC-SVM will not be evaluated further as a candidate classification method.

3.5.2 NN & SVM

This section evaluates the well-known Neural Network (NN) and Support Vector Machine (SVM) classification methods. These classification methods are evaluated together as both methods use handcrafted features as input data and the methods should achieve similar results.

Figure 13 shows an example schematic if a NN classifier, also known as Multi-Layer Perceptron (MLP). A NN model for classification is a network with one or more layers that takes numeric input features and outputs a classification label. The functionality of a NN is defined by the network structure and a large number of weight parameters that are configured during training.

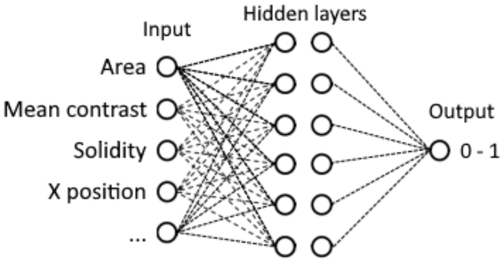


Figure 13 Example of a simple NN classifier with handcrafted features and one or more hidden layers. The output of the NN classifier is a classification score between 0 and 1.

An SVM classifier does not use layers of neurons but uses a subset of the dataset as support vectors. Between the support vectors of both classes, a decision boundary is defined that maximises the distance between detections of different classes.

Although the methods function completely different internally, both methods should perform similarly on a reasonably sized dataset. SVM has the advantage that is quicker to train and requires less data to be trained. In most cases, a NN should achieve a slightly higher classification accuracy given enough training data [33].

3.5.2.1 Input data

The NN and SVM in this research both use handcrafted features as input. Handcrafted features are numeric features that have been extracted from object detections. These features are descriptive of a detection and include features such as: area, total contrast, mean contrast, circularity of the contour, horizontal x position in the container and image moments of the contour. In total, 40 handcrafted features are extracted. A list of all handcrafted features used in this research are given in Appendix B.

Not all features however are useful to differentiate between detections of bubbles and particles as not all features can differentiate between detections. To make a selection of useful features, the Relief [30] feature selector is used. The Relief feature selector selects a set of input features for a desired number of features. The graph in Figure 14 is created by evaluating any number of features with the NN and SVM classification methods. The dip in the graph at 9 handcrafted features is caused by a false minimum within the Relief optimisation step. From this it is chosen to use a selection of 11 handcrafted features presented in Table 7.

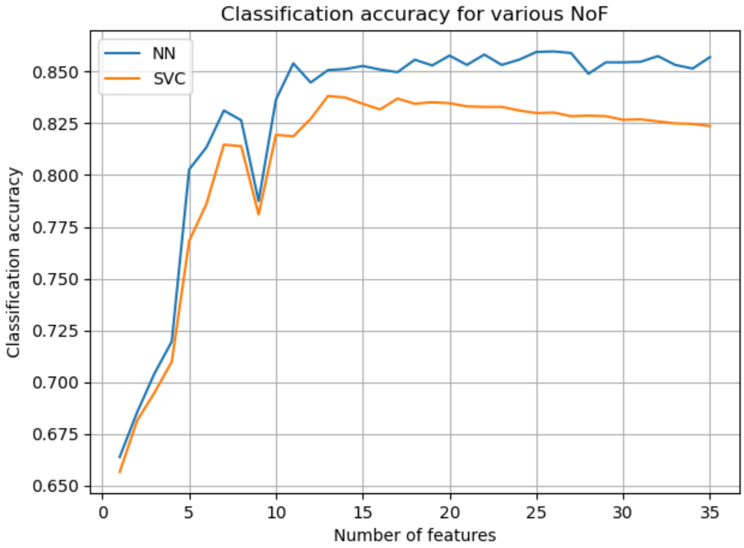


Figure 14 Classification accuracy of NN and SVC classifiers with different numbers of features (NoF) selected with Relief. The figure shows that a better classification accuracy is achieved with the NN classifier. Based on this figure it is chosen to continue with a selection of 11 handcrafted features as using more features does not result in a higher classification accuracy.

Table 7 Selection of 11 handcrafted features determined with Relief feature selection with corresponding importance metric.

	Feature	Importance	Description
1	Nu_{03}	0.604	Central normalized moment 03
2	Mean contrast	0.502	Average contrast to the background
3	Nu_{12}	0.406	Central normalized moment 12
4	Nu_{11}	0.395	Central normalized moment 11
5	Solidity	0.366	Solidity of the contour
6	X position	0.350	Horizontal position in the container
7	Nu_{21}	0.339	Central normalized moment 21
8	Nu_{02}	0.281	Central normalized moment 02
9	Width	0.252	Width of the contour
10	Nu_{30}	0.244	Central normalized moment 30
11	Ellipse mayor axis	0.238	Mayor axis of an ellipse fitted on the contour

3.5.3 CNN

Convolutional Neural Network (CNN) [34] refers to a Neural Network preceded with one or more convolutional layers. Figure 15 shows an example schematic of a CNN classifier with one convolutional layer.

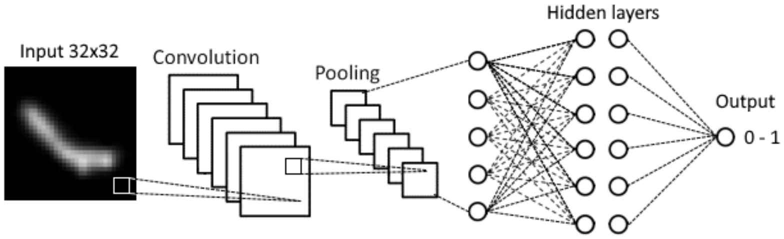


Figure 15 Example of a simple CNN classifier with a 32x32 input and a single convolution and pooling layer. The layers after the pooling layer are similar to the NN classifier in Figure 13. The output of the CNN classifier is a classification score between 0 and 1.

CNN is a popular classification method specialised for the classification of images. The convolutional layers of a CNN act as a feature extractor and recognise local patterns in images. The final layer(s) of a CNN are the same as a regular NN and predict the desired outcome, in this case the detection class.

The benefit of a CNN over a regular NN is that a CNN does not require handcrafted features as the features are extracted by the convolutional layers. In most cases, if given enough training data, a CNN model will outperform a NN that uses handcrafted features. A downside of CNN is that it generally requires more training data to be trained than a NN with handcrafted features. A NN can outperform a CNN if the handcrafted features are of a high quality and training data is limited.

3.5.3.1 Input data

A CNN uses images as input data. As detections are different sizes, these must be resized to a constant size before they can be used as input for the CNN. After some testing it was decided to resize the images to 32x32 using linear interpolation. Figure 16 shows two detections before resizing and two detections after resizing. To artificially increase the size of the dataset, all detections are mirrored around the vertical and horizontal axis, resulting in four times the number of detections. 90-degree rotations are not used to further increase the number of detections as this could make the data less representative of unaltered detections.

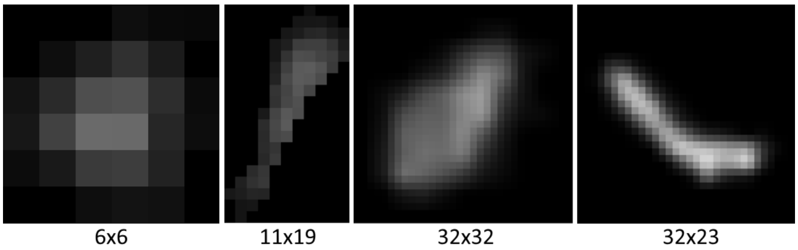


Figure 16 Examples of extracted object detections before and after resizing to 32x32 for CNN input.

3.5.4 Results B.1

To select a classification method the candidate classification methods have been trained and evaluated using the test dataset with detections larger than 30 pixels area and k-fold cross validation k=5. Table 8 shows the average classification accuracy obtained by the different classification methods.

Table 8 Classification accuracy per candidate classification method on the filtered dataset with a minimum detection area of 30 pixels

Classification method	Classification accuracy
OC-SVM	<0.70
NN	0.855
SVM	0.839
CNN	0.930

As the difference in classification accuracy is significant, it can be assumed that CNN classification method outperforms the other candidate classification methods, even if these would be further optimised.

3.5.5 Conclusion B.1

The research question to be answered in this section is:

B.1: *"Which of the candidate classification methods achieves the highest classification accuracy for single detections on the dataset described in section 3.4?"*

The candidate classification method that achieves the highest classification accuracy is the CNN classification method. The trained CNN classifier achieves a classification accuracy of 0.930 or 93% on detections with an area of at least 30 pixels. With further optimisation, it is likely that the classification accuracy of the classifier could still be increased. As the goal of this research question is only define a baseline classifier, it will not be optimised any further.

Results of the trained classifier are used throughout the rest of this research and will be used as a baseline in future work.

3.6 B.2 Classifiable detections

This section aims to answer research question B.2.

B.2: *“What correlations exist between handcrafted features and the classification accuracy of a detection that can be used to filter detections that cannot be reliably classified?”*

The background subtraction and segmentation module is able to detect objects with an area of 5 pixels or more. However, it can be assumed that detections of this small size do not contain enough information to be classified reliably. A logical step is to filter out any detections that are too small, but how small is too small? And what about detections with a very low contrast to the background? Can very small detections also not classified reliably if they have a high contrast? What other factors play a role?

The goal of this section is to find out what handcrafted features correlate to classification accuracy and can predict whether a detection can be classified reliably. It is also discussed what filter methods could be used to filter out detections most effectively and how the predicted classification accuracy can be used as a weight for multi-detection classification strategies.

An ideal filter would reject most detections that are too hard to reliably classify while retaining the most detections that can be classified.

3.6.1 Method

Correlations between handcrafted features and classification accuracy are observed from graphs in which handcrafted features are plotted against classification accuracy. The process to obtain such a plot is as follows:

1. Five CNN classifiers are trained using 80% of the dataset, such that the test datasets together form the complete dataset. Data augmentation with flipping resulting in four times the number of detections is performed before training after splitting the train/test data.
2. The resulting classification score between 0 and 1 of each detection is stored in a database together with the class and handcrafted features.
3. The database is sorted on a selected handcrafted feature.
4. For every 500 detections, the average classification accuracy is plotted in a graph.

The final two steps are repeated for every handcrafted feature. Handcrafted features with an interesting or significant correlation are described in the next subsection.

The handcrafted feature with the strongest correlation is area. To test the effect of a simple threshold filter that filters out detections smaller than a certain area and to observe more complex corelations with detection area and other handcrafted features, all steps described are repeated with different area threshold filters in place. Used area thresholds are: 6, 10, 20, and 30 pixels.

3.6.2 Observed correlations

Correlations of all handcrafted features listed in Appendix B have been tested. A selection of handcrafted features with a strong or interesting correlation to the classification accuracy are discussed below.

Area correlation

Figure 17 shows in a graph the correlation between the area of a detection and the classification accuracy. In the graph, a strong correlation can be seen between area and classification accuracy. Detections that are smaller achieve a lower classification accuracy. Above an area of 50 pixels, classification accuracy is not significantly improved. Table 9 shows the average accuracy and retained percentage for the different area thresholds. In the graph and table can be seen that a simple threshold filter for detection area would be a reasonably effective approach, as the accuracy is increased with a higher area threshold. The area threshold should not be set too high as this would result in more particles not being classified. An area threshold above 30 was not tested.

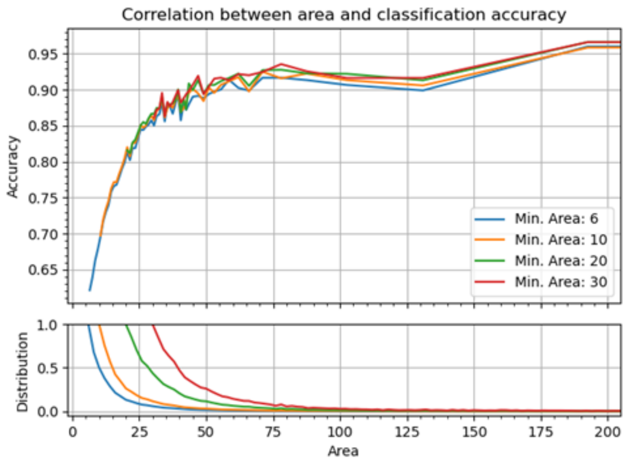


Figure 17 Correlation graph between area and classification accuracy and distribution graph of area in the filtered dataset. The figure shows that detections with a small area achieve a lower classification accuracy on average. Filtering small detections also results in a slightly higher overall classification accuracy due to efficient training.

Table 9 Percentage of detections which is retained and the average classification accuracy with different area thresholds

Min. Area	Retained %	Avg. acc.
6	100%	0.809
10	60%	0.853
20	23%	0.909
30	11%	0.930

Total and mean contrast

Figure 18 shows the correlation between total contrast and mean contrast with the classification accuracy.

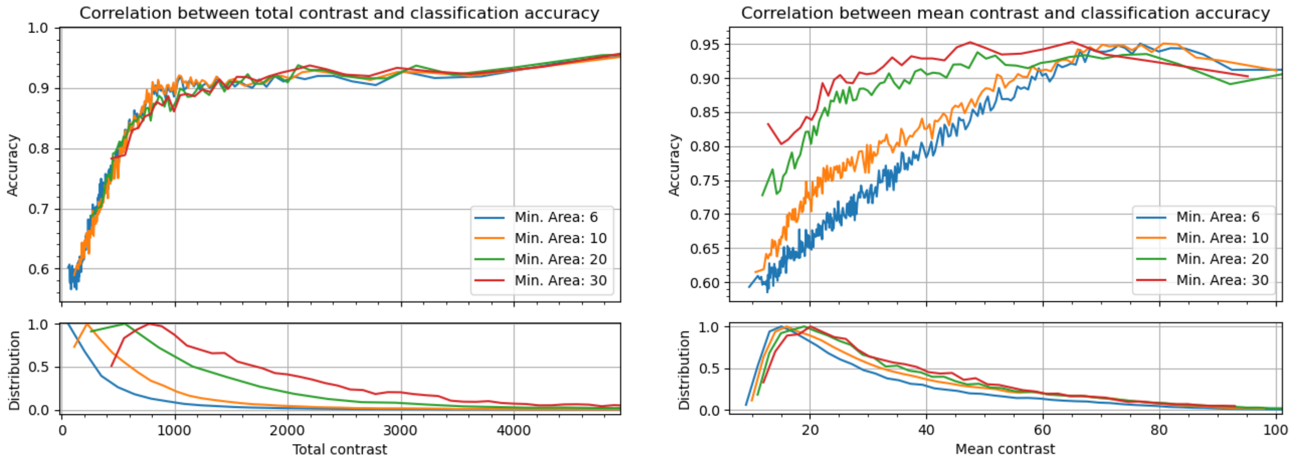


Figure 18 Correlation graphs of handcrafted features with classification accuracy, and distribution graphs of the handcrafted features in the filtered dataset. Left: Total contrast. Right: Mean contrast. The figure shows strong correlations between these features and classification accuracy and indicates complex corelations between multiple features.

The correlation between total contrast and accuracy is comparable with that of area and classification accuracy. Detections with a low total contrast achieve a lower classification accuracy on average. Interesting is that below a total contrast of 1400, a higher classification accuracy is achieved with a lower area threshold. This indicates that there are smaller detections with a high mean contrast that achieve a higher classification accuracy than larger detections with the same total contrast but a lower mean contrast. This is supported by the graph showing correlation between mean contrast and classification accuracy. In this graph can be seen that with a low area threshold, detections with a high mean contrast can achieve a high average classification accuracy.

This finding indicates that a simple threshold filter for detection area is not ideal as it reject detections that can be predicted to achieve a high classification accuracy. A more complex filter that uses correlations between multiple handcrafted features should achieve a higher average classification accuracy while retaining more detections.

Horizontal position correlation

Figure 19 shows the correlation between the horizontal x position and classification accuracy.

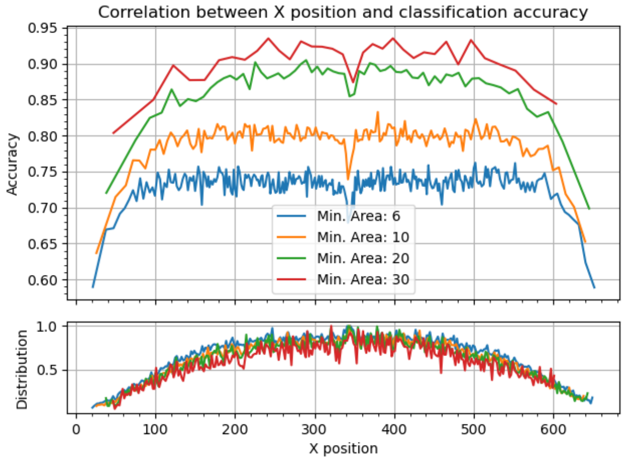


Figure 19 Correlation graph between x position and classification accuracy and distribution graph of x position in the filtered dataset. The figure shows a strong reduction in classification accuracy near the outside of the horizontal range.

In the figure can be seen that detections near the outside of the horizontal range achieve a significantly lower classification accuracy. The most probable explanation for this is that detections near the edge of the container are deformed by the container walls, making these detections less recognisable. Another possibility is that close to the outside, detections occur that are caused by another object reflecting on the container walls. Another interesting observation is that detections exactly in the centre of the container achieve a lower classification accuracy. The reason for this is unknown. In future work this can be further investigated and possibly prevented.

Features with a weaker correlation

Every handcrafted feature listed in Appendix B has been analysed for correlation with classification accuracy. Most of the handcrafted features show a correlation that is weaker and more complex. Figure 20 shows the correlation of ellipse angle, ellipse eccentricity, solidity, circularity, and aspect ratio with the classification accuracy.

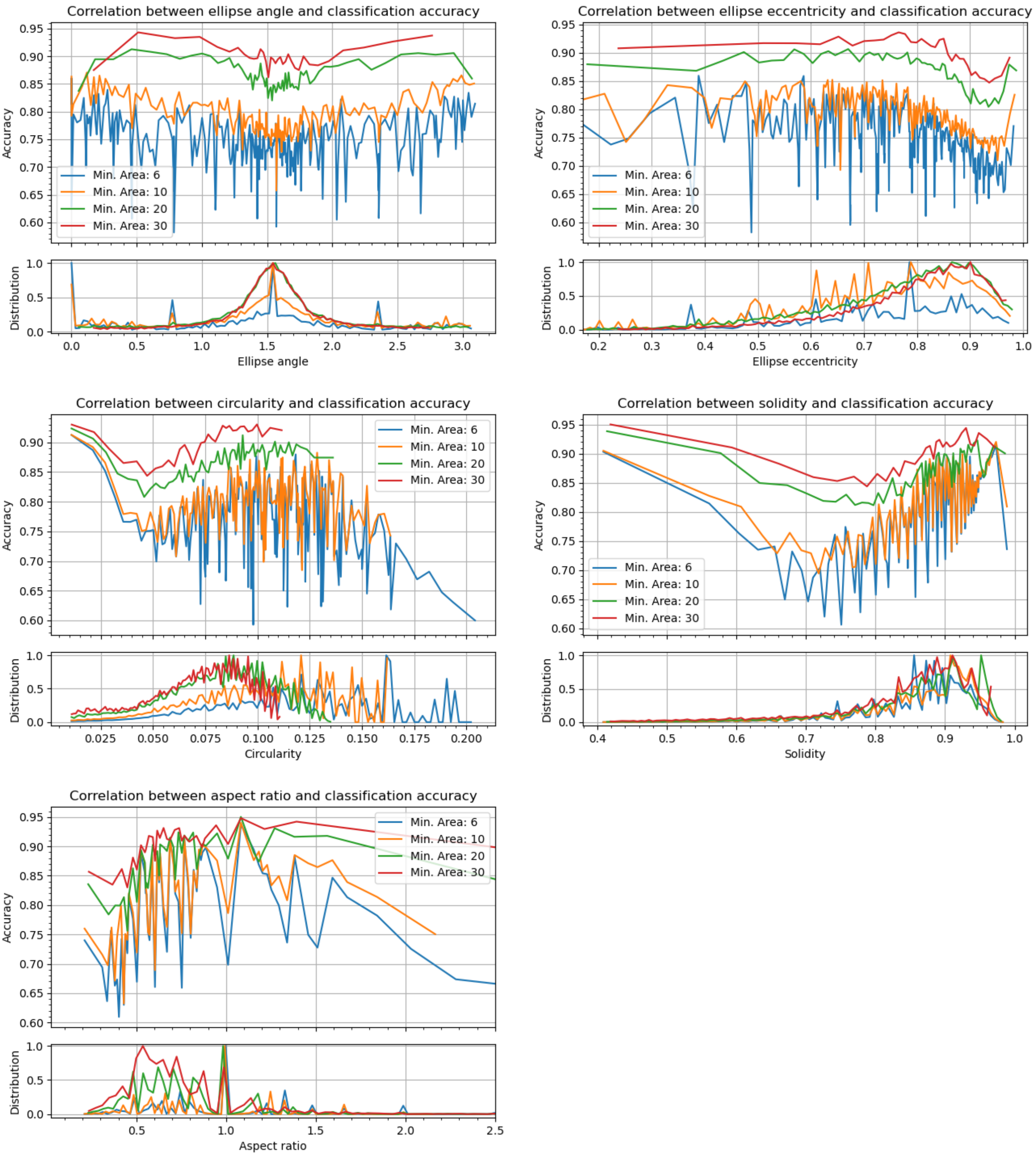


Figure 20 Correlation graphs of handcrafted features with classification accuracy with the distribution of the handcrafted features in the filtered dataset. In order respectively: Ellipse angle, Ellipse eccentricity, Solidity, Circularity, Aspect ratio. The figure shows weaker complex correlations between these handcrafted features and classification accuracy.

The correlation between these handcrafted features is weaker and more complex than other features discussed previously. As a result of this, these features are not that useful to define a simple threshold filter. For example, a simple filter that rejects any detections with an ellipse eccentricity above 0.9 would not result in a large increase in average classification accuracy but would result in a large number of classifiable detections not being classified. However, with a more complex filter, these handcrafted features could help to more accurately predict weather a detection can be classified. For example, if a detection is a bit too small and does not have a high total contrast but these weaker features indicate a high classification accuracy, the detection should not be rejected.

3.6.3 Filter methods

This subsection discusses some methods to filter out detections that are too hard to classify based on the observed correlations.

Simple threshold filter

The simplest solution to filter detections that are too hard to classify, is the introduction of one or more simple threshold filters. Such a filter rejects a detection if a certain handcrafted feature is above or below a given threshold. This threshold can be chosen arbitrarily by selecting a point in the correlation trend or selecting a minimum classification accuracy. A possible set of simple threshold filters for this situation is shown in Table 10.

Table 10 A possible set of simple threshold filters determined from the observed correlations.

Handcrafted feature	Accepted range
Area	30<=
Total contrast	700<=
Mean contrast	20<=
X position	80<=
X position	<=580

Although this filter method is simple and achieves the goal of increasing the classification accuracy, it does this by rejecting a significant of the detections including many classifiable detections. Better results can be obtained with a more complex filter as explained below.

Complex threshold filter

The findings regarding the correlations of total and mean contrast with classification accuracy imply that there are complex correlations with multiple handcrafted features that can more effectively predict whether a detection can be reliably classified.

A regression model such as an SVR or a NN for regression could be used to predict the classification accuracy of a detection based on a selection of handcrafted features. Such a model can be trained using classification results of the classifier as have been used during this research. The complex filter can be configured by simply choosing a threshold for the predicted classification accuracy. To improve prediction, learning could be performed in an iterative process as removing detections from the training data can result in a more accurate classifier.

A complex threshold filter using the selection of handcrafted features shown in Table 10, should result in a higher classification accuracy with less detections being rejected compared to a selection of multiple simple threshold filters. Additionally, multiple handcrafted features with a weaker correlation can also be included to further increase the classification accuracy.

Classification weight factor

The final usage of the correlations between handcrafted features and classification accuracy that will be discussed is to determine a weight factor during classification.

In section 3.8, two multi-detection classification strategies are discussed that perform classification based on multiple detections. By using multiple detections, it is prevented that a single bubble detection falsely classified as a particle results in a container rejection. The classification accuracy predicted by a threshold filter can be used in various ways to improve classification with a multi-detection classification strategy. Ways to do this have not been researched in this research but some possibilities are described below.

When multi-positive classification is used as described in subsection 3.8.2, the number of detections above a certain classification score is used. This can be adapted by using the sum of classification accuracies of detections above a classification score. This way, if the classifications are expected to be less reliable, more positive detections are needed. If median voting as described in subsection 3.8.1 is used, the predicted classification accuracies can be used as weights to perform weighted median voting. As this has not further been researched, it is unknown how effective these methods would be.

3.6.4 Conclusion B.2

The research question to be answered in this section is:

B.2: *"What correlations exist between handcrafted features and the classification accuracy of a detection that can be used to filter detections that cannot be reliably classified?"*

It has been found that multiple handcrafted features have a significant correlation with classification accuracy. These can be used to define simple filters that reject detections that cannot be classified reliably. A complex filter would be more effective as it can use complex correlations between multiple handcrafted features and also include handcrafted features with weaker correlations. This can be done by predicting the classification accuracy based on these features using a model.

3.7 B.3 Full container classification

This section aims to answer research question B.3.

B.3: *"What is the theoretical effectiveness of the classifier to classify a container with multiple detections, if a single positive detection results in the container being rejected?"*

In this problem situation, multiple objects are present and detected in multiple frames. Up to this point in the research for the classification module, only classification of singular detections has been considered. The goal of this research question is to analyse the effectiveness of the classifier selected in section 3.5 to classify a container sample where multiple detections are made from the capture sequences. The default classification strategy for defect inspection is to reject a container if a single detection is classified as a particle.

In addition to the well-known metrics of *False Positive* (FP) rate and *False Negative* (FN) rate, this chapter introduces the *False Container Rejection* (FCR) rate and *False Container Acceptance* (FCA) rate. It is shown that the default classification strategy is not viable as this results in high FCR and FCA rates.

Note on simulation method

For the graphs in this section and section 3.8, classification scores are sampled from all detections in the dataset with an area of at least 30 pixels. To simulate the classification of a certain number of detections, that quantity of classification scores is randomly sampled from the classification score distribution shown in Figure 21. This is not completely representative of reality, as randomly sampled scores contain detections of different objects. An easy to classify object such as a large fibre might result in mostly classifications with a high classification score close to 1, whereas a tiny dust particle can result in only difficult to classify detections with a classification score closer to 0.5.

3.7.1 False Positive rate and False Negative rate

The False Positive (FP) rate and False Negative (FN) rate are well known performance metrics for classification. For a chosen classification score threshold, the FP and FN rate specify how many bubbles will be falsely classified positive as particles or how many particles are falsely classified negative as bubbles. Depending on the application, the classification threshold can be chosen differently than the optimum where FP and FN are both the lowest.

When a detection is classified, the result of the classifier is a number between 0 and 1 called the classification score. In this research, a classification score close to 0 indicates that the detection is likely of a bubble, and a classification score close to 1 indicates that the detection is likely of a particle. The left side of Figure 21 shows the classification score distribution for bubbles and particles. The FP and FN rate are calculated by integrating the classification score distribution and is shown in the right side of Figure 21. The figure indicates that an optimal classification threshold of 0.5 would result in 7.13% of the detections of both bubbles and particles will be falsely classified.

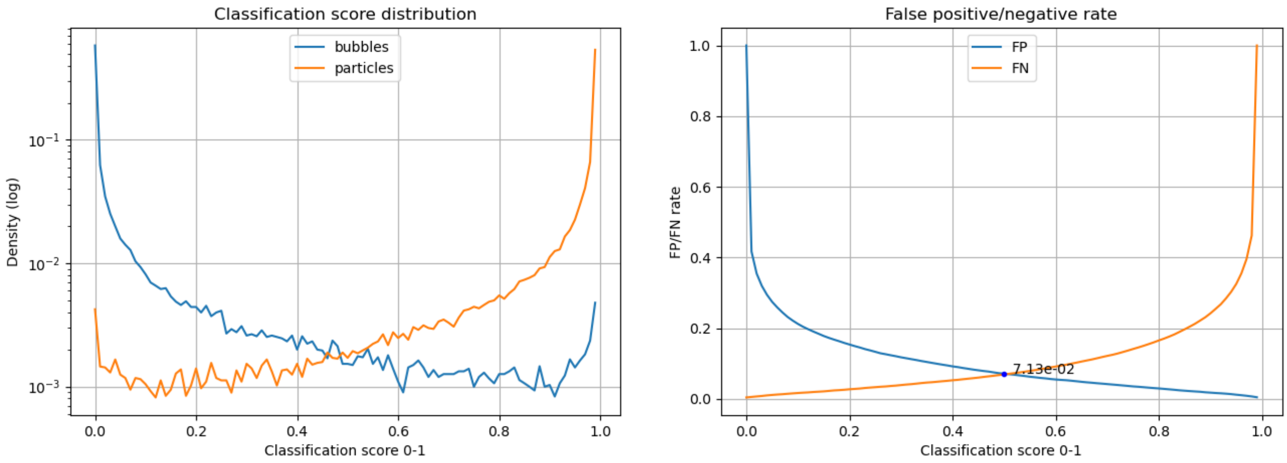


Figure 21 Left: Classification score distribution for bubbles and particles in logarithmic scale. Right: False Positive and False Negative rate obtained by integrating the classification score distribution.

3.7.2 False Container Rejection rate and False Container Acceptance rate

In this problem situation, multiple objects are present and detected in multiple frames. With the default classification strategy for inspection, if a single detection is classified positively as a particle, the entire container should be rejected. To analyse the effectiveness of classification of a complete container image sequence, the false container rejection (FCR) rate and false container acceptance (FCA) rate are introduced as more representative indicators of classification effectiveness. The FCR rate is the rate at which a clean container is falsely rejected as a contaminated container. The FCA rate is the rate at which a contaminated container is falsely accepted as a clean container.

The FCR rate and FCA rate can be calculated using Equation 7 and Equation 8. Where tr is the classification threshold, N_0 is the number of bubble detections in a clean container and N_1 is the number of particle detections in a contaminated container.

$$FCR(tr) = 1 - \left(1 - FP(tr)\right)^{N_0} \tag{7}$$

$$FCA(tr) = FN(tr)^{N_1} \tag{8}$$

Figure 22 shows the FCR rate and FCA rate if a clean container would contain 100 detections (N_0 = 100), and a particle in a contaminated container would be detected 10 times (N_1 = 10). The optimal classification threshold would be 0.998 which is extremely high. The resulting FCR and FCA rates are 23.2%. This situation is unacceptable as the FCR and FCA rates are too high, and the high classification threshold makes classification unpredictable in practice.

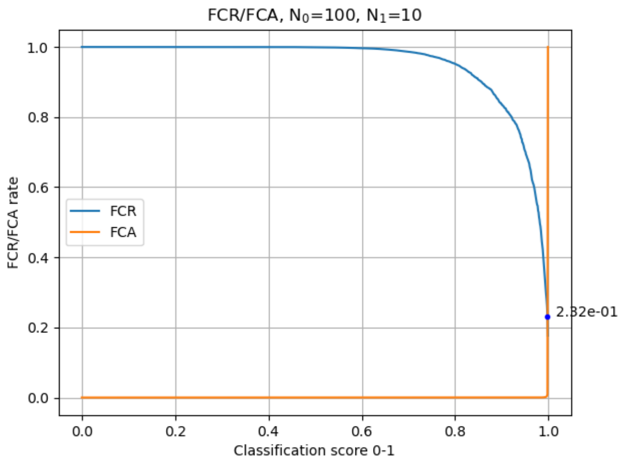


Figure 22 False Container Reject (FCR) rate and False Container Acceptance (FCA) rate with 100 detections of bubbles (N_0 = 100) and 10 detections of particles (N_1 = 10). The figure shows a resulting optimal classification threshold of 0.998 with an FCR rate and FCA rate of 23.2%.

3.7.3 Conclusion B.3

The research question to be answered in this section is:

B.3: *"What is the theoretical effectiveness of the classifier to classify a container with multiple detections, if a single positive detection results in the container being rejected?"*

It has been found that the simple classification strategy is not effective for this classification problem. This is as a clean container can contain multiple moving bubbles resulting in a large number of detections in the full capture sequence. Each of the bubble detections has the chance to be falsely classified as a particle.

The effectiveness of this strategy has been demonstrated using through simulation. The False Container Rejection (FCR) rate and False Container Acceptance (FCA) rate are introduced as useful indicators of effectiveness. Through simulation it is shown that the resulting classification threshold is 0.998, achieving an optimal FCR rate and FCA rate of 23.2%. This means that the single detection classification strategy is not feasible, even if the used classifier would be further optimised.

3.8 B.4 Multi-detection classification strategies

This section aims to answer research question B.4.

B.4: *"What is the theoretical effectiveness of a classification strategy that uses multiple detections to classify a container sample?"*

This section proposes two classification strategies to classify a container using multiple detections. The strategies that are proposed are: median voting classification with tracking and multi-positive classification.

Median voting classification uses the results from the tracking module to classify a set of detections that belong to the same object. This method can achieve a very low FCR/FCA rate if objects can be correctly tracked for multiple frames and the number of false trajectories is not too high.

Multi-positive classification does not use the results of tracking but simply requires multiple detections to be classified positive before the container is classified as contaminated. This method can also achieve a low FCR/FCA rate but requires objects to be visible for more frames compared to median voting with accurate tracking results.

In this research, a quantity of detections is simulated by randomly sampling from the score distribution shown in Figure 21. Therefore, notes in the paragraph "Note on simulation method" on page 59 also applies here. It is possible that the effect of additional trajectories on the false container rejection rate is not as intense as described in this section, as many additional trajectories contain mostly the same detections as others, while the tests in this section sample random detections for each trajectory. Future research is needed to confirm that the simulation is accurate.

3.8.1 Median voting classification

If an object can be tracked over multiple frames, classification can be performed by combining classifications of the different detections of that object. A relatively simple method to perform classification with multiple detections is median voting. Other voting schemes and more complex multi-detection classifiers also exist and might be more effective. However, in this research only median voting is discussed.

Median voting works by taking the median value of the classification scores of all detections of an object. As a result of this, an object is classified positive only if at least half of the detections achieve a high classification score. This makes median voting effective in dealing with data that contains outliers. As found in section 3.7, outliers of bubble detections that are falsely classified as particles are a critical problem for the classification of containers. For that reason, median voting classification is proposed in this research.

The left graph in Figure 23 shows the median classification score distribution with different lengths of trajectory. The right graph in Figure 23 shows the False Negative (FN) rate and False Positive (FP) rate of median voting classification with different lengths of trajectory. In the figure can be seen that the FP rate and FN rate are significantly lower with median voting when classifying a single detection/trajectory. Longer trajectories are less likely to contain a majority of outliers and achieve a lower FP rate and FN rate. Even when a trajectory contains only 3 detections (L = 3), the balanced optimal FP and FN rate is almost 5 times lower than for a single detection (L = 1).

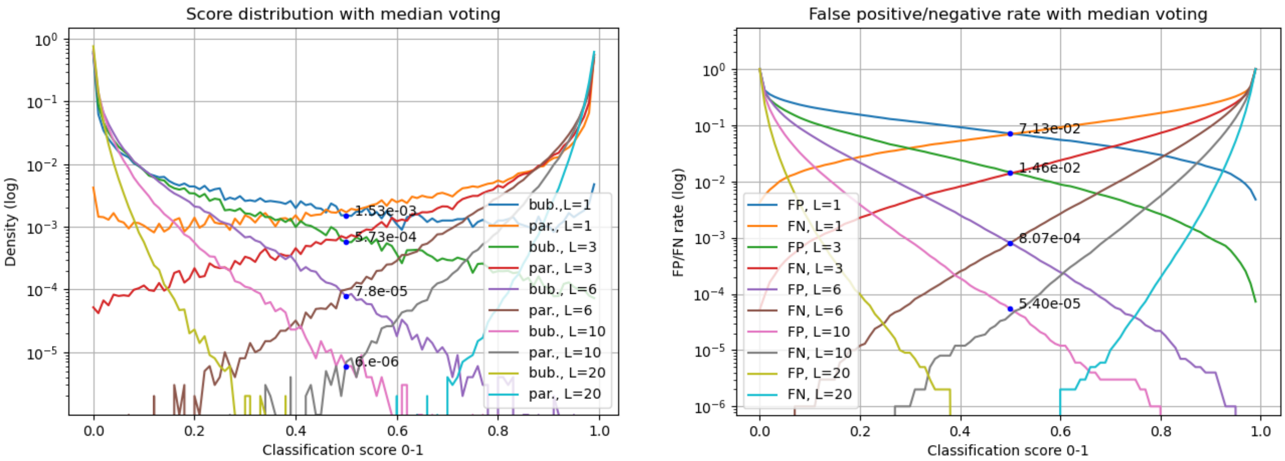


Figure 23 Result graphs of median voting with different trajectory length L on logarithmic scale. Left: Classification score distribution. Right: False Positive (FP) and False Negative (FN) rate. The figure shows significantly lower FP and FN rates with median voting compared to single detection classification (L=1). With longer trajectory lengths, FP and FN rates are further reduced.

3.8.1.1 FCR/FCA with median voting classification

The TDCP tracking algorithm described in chapter 4 for the tracking module returns a list of track candidates, these are possible trajectories of moving objects. As track candidates can overlap with other track candidates, the number of track candidates is likely higher than the number of moving objects present in a container. It is possible that more track candidates are detected than the number of objects in a sequence. Each track candidate should be classified separately and can result in the container being rejected.

The False Container Rejection (FCR) rate and False Container Acceptance (FCA) rate with median voting can be calculated with Equation 7 and Equation 8 on page 60, where N_0 and N_1 stand for bubble and particle trajectories instead of singular detections.

If no particles are present, each of the detected trajectories can cause a false reject and N_0 is equal to the number of detected trajectories. In situations where a particle is present, only most correct trajectory of that object is relevant for the classification outcome. Therefore, N_1 is equal to the number of particle objects in a container which is assumed to be one. Trajectories that are less correct contain more detections of different objects (bubbles) are not likely to affect the classification outcome if a more correct particle trajectory is present.

Figure 24 shows the FCR rate and FCA rate when all tracks are of the same length L and $N_0 = 10$, $N_1 = 1$. In the figure can be seen that the optimal classification threshold is shifted to the right compared to the right graph in Figure 23 where a single trajectory is classified. The optimal classification threshold is affected by the tracked length of a trajectory L and the number of trajectories that is detected of bubbles N_0.

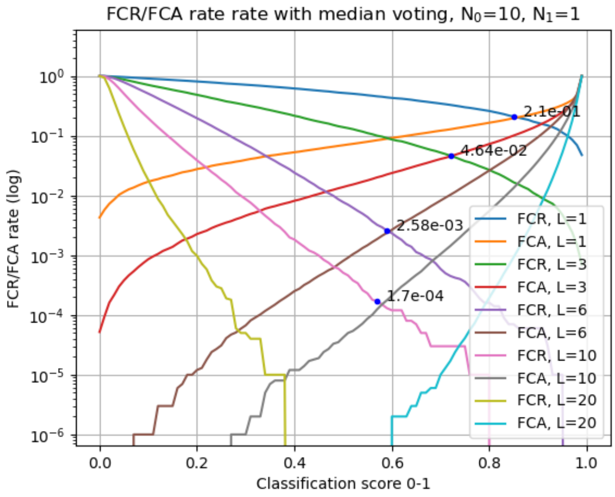


Figure 24 False Container Reject (FCR) rate and False Container Acceptance (FCA) rate in logarithmic scale with 10 detected trajectories of bubbles (N_0 = 10) and one detected trajectory of a particle (N_1 = 1). The figure shows that the optimal classification threshold and the resulting optimal FCR and FCA rates are increased.

An adaptive classification threshold can be determined for different lengths of trajectory, given an average number of detected trajectories of bubbles N_0. Figure 25 shows the adaptive classification threshold for different values of N_0. Figure 26 shows the resulting ideal FCR and FCA rates if all trajectories would be of the same length. The figure shows for example that if all objects are reliably tracked for 11 frames, an FCR and FCA rate of 0.1% could be achieved, even if 500 trajectories of bubbles are detected in clean containers ($N_0 = 500$).

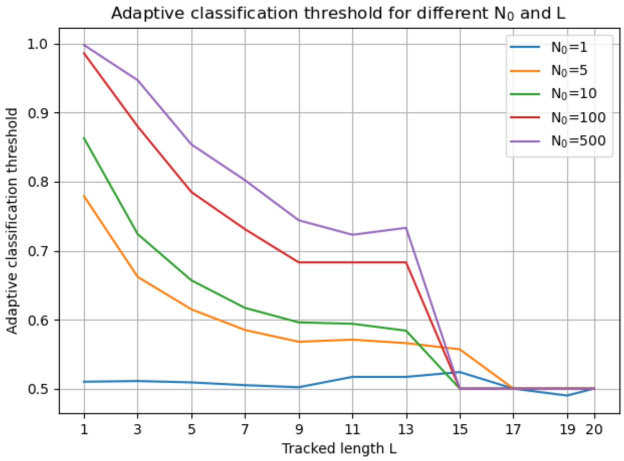


Figure 25 Adaptive classification threshold for different values of N_0 and tracked length L. The figure shows the correlation between the optimal adaptive classification threshold, the number of detected bubble trajectories, and the length of trajectories.

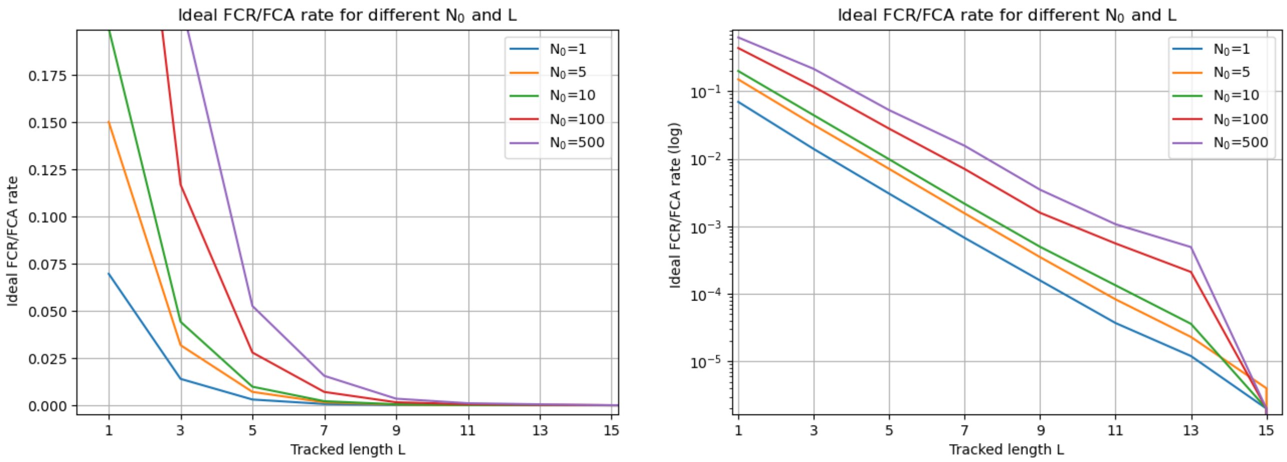


Figure 26 Optimal FCR/FCA rate for different values of N_0 and tracked length L in linear and logarithmic scale. The figure shows what FCR/FCA rate can be achieved given the number of bubble trajectories and the trajectory length. If the trajectory length is long, more bubble trajectories can be detected for acceptable results.

3.8.1.2 Acceptable number of detected bubble trajectories N_0

It has been established that the number of detected trajectories influences the FCR rate. To aid in the development and validate the tracking module, it is helpful to establish how many possibly false trajectories of bubbles can be detected while achieving an acceptable FCR rate and FCA rate.

The acceptable quantity of bubble trajectories, $N_{0\,acceptable}$ can be determined with Equation 9 based on the FP rate and FN rate, given acceptable parameters $FCR_{acceptable}$ and $FCA_{acceptable}$.

$$N_{0\,acceptable} = \frac{\log\left(1 - FCR_{acceptable}\right)}{\log\left(1 - FP(tr)\right)}, \qquad where: FN(tr) = FCA_{acceptable} \tag{9}$$

Table 11 shows values of $N_{0\,acceptable}$ for various values of $FCA_{acceptable}$ and $FCR_{acceptable}$. From the table can be seen that for example, if an FCR rate of 0.03 and FCA rate of 0.01 are accepted and every trajectory is 8 detections long, 13243 trajectories of bubbles can be detected.

Table 11 Different acceptable values of $N_{0\,acceptable}$ for different $FCR_{acceptable}$, $FCA_{acceptable}$, and tracked Length

FCA accept.	0.01			0.03		
FCR accept.	0.01	0.03	0.06	0.01	0.03	0.06
Length ↴	$N_{0\,acceptable}$ ↴					
1	0.03	0.09	0.19	0.07	0.20	0.41
3	0.54	1.62	3.30	1.49	4.53	9.20
4	6.08	18.4	37.5	19.9	60.4	123
5	10.1	30.5	62.0	35.9	109	221
6	158	478	971	609	1846	3750
7	222	674	1369	1104	3347	6799
8	4370	13243	26902	14358	43513	88393

9	7179	21757	44197	20101	60918	123751
10-20	inf	inf	inf	inf	inf	inf

3.8.2 Multi-positive classification

Multi-positive classification is a classification strategy that does not require tracking to be implemented. With multi-positive classification, a container is classified as positive if multiple detections have been classified positive. Through simulation, the FCR rate and FCA rate can be determined for every quantity of positive detections Q. The optimal classification score threshold tr is different per positive quantity Q. This is as with a larger positive quantity Q the chance that all positive detections are falsely classified outliers is lower.

Figure 27 shows the FCR and FCA rate for some positive quantity thresholds Q if a particle is visible for $N_1 = 10$ frames and $N_0 = 100$ detections of bubbles are detected. The figure shows for example that if at least $Q = 6$ detections score at least $tr = 0.85$, the resulting FCR and FCA rates are 2.75%.

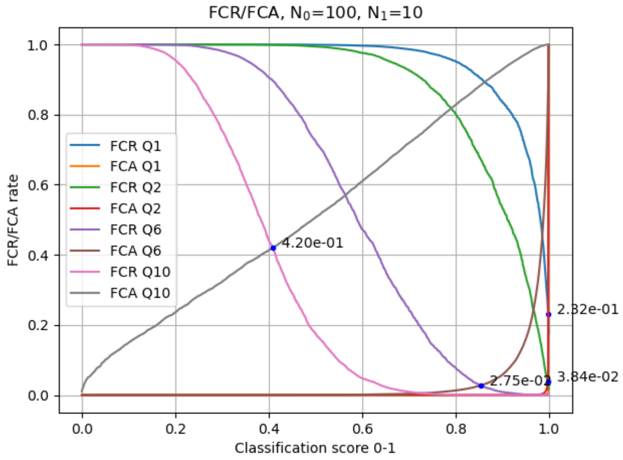


Figure 27 False Container Rejection (FCR) rate and False Container Acceptance (FCA) rate with multi-positive classification for some values of Q with 100 detections of bubbles (N_0 = 100) and 10 detections of particles (N_1 = 10). The figure shows that with Q = 2 and Q = 6, a significantly lower FCR rate and FCA rate can be achieved.

The average number of frames a particle detection is visible N_1 and the average number of bubble detections N_0 have a strong effect on the resulting FCR and FCA rates. If a particle is visible in more frames, the chance is higher that more detections achieve a high classification score. Figure 28 shows this effect. The graphs show the trends of FCR values, points show the intersections between FCR and FCA at the optimal classification threshold tr for different values of Q.

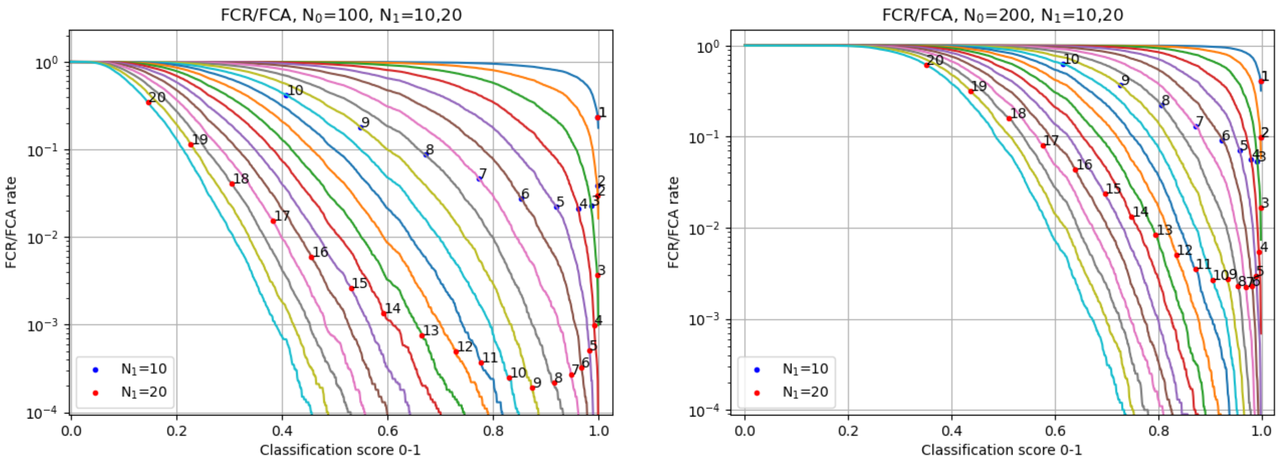


Figure 28 Optimal False Container Reject (FCR) rate and False Container Acceptance (FCA) rate in logarithmic scale for various values of N_0 and N_1. The figure shows the effect of the number of bubble detections N_0 and particle detections N_1 on the classification threshold for all values of Q and subsequent FCR and FCA rates.

Based on this simulation and a number of parameters, a list can be made with values of Q and corresponding classification thresholds. Next, a container should be classified positive if any Q detections score above the threshold tr.

Given that:

- On average, $N_0 = 100$ and $N_1 = 10$
- $FCR_{acceptable} = 0.03$
- $FCA_{acceptable} = 0.03$

A resulting Q list would be as presented in Table 12.

Table 12 Optimal classification threshold, FCR rate and FCA rate for various values of Q

Q	Threshold	FCR	FCA
10	0.64	0.03	0.653
9	0.69	0.03	0.318
8	0.74	0.03	0.137
7	0.80	0.03	0.060
6	0.85	0.03	0.026
5	0.90	0.03	0.015
4	0.95	0.03	0.013
3	0.98	0.03	0.015
2	0.996	0.03	0.049
1	-	-	-

For some values of Q in Table 12, the FCA rate is higher than the acceptable limit of 0.03. However, this might not be a problem as the FCA rate is acceptable for $Q = 3,4,5,6$. It is unclear what the resulting FCR and FCA rate are when all values of Q are used for classification and FCR is not above $FCR_{acceptable}$ for singular Q. Further research has not been performed into results of multi-positive classification with multiple values of Q.

3.8.3 Conclusion B.4

Research has been performed into two multi-detection classification strategies: median voting classification with tracking and multi-positive classification.

Median voting classification uses the results of object tracking to classify a set of detections that belong to the same object. As median voting is effective at dealing with outliers of bubbles that are detected as particles, the strategy can achieve a very low FCR and FCA rate even from short partial trajectories.

Multi-positive classification does not use results of tracking but requires multiple detections to be classified positive before the container is classified positive. Simulation shows that this method can also achieve a low FCR and FCA rate but requires particles to be visible on more frames than with median voting.

3.9 Comparison to SOTA

This section compares the proposed solution to the current SOTA for this module, as described in section 2.1. Although the solution cannot be directly compared to the SOTA by comparing results, it is expected that the proposed solution outperforms the current SOTA.

Handcrafted feature classification

Two handcrafted feature classification methods have been previously identified as SOTA [16] [26]. Both SOTA methods report a higher classification accuracy for singular detections compared to the CNN classifier selected and presented in section 3.5. However, as the papers do not clarify the size of particles present in the sample set, it is not possible to determine whether this is a result of the particles being larger and therefore easier to classify. None of the papers describe the negative effect of the large number of bubble detections on the FCR rate and FCA rate of a container described in section 3.7. Instead, the papers seem to directly relate the classification accuracy of a single detection to the classification accuracy of a container which is not realistic. The SOTA methods do not implement a multi-detection classification strategy.

When compared using challenging particles and a realistic number of bubble detections, it is expected that the proposed solution outperforms the current SOTA when using median voting classification or multi-positive classification.

3.10 Conclusion

As the development of the classification module is in an early stage, the goal of this research is not to develop a working classification module that is able to classify containers accurate enough to be implemented in the VPIS, but to select and develop a baseline classifier for singular detections and perform additional exploratory research into the opportunities and pitfalls of this classification problem.

A labelled dataset has been collected from a sample set created for this research. Labelled data of bubbles is collected from a clean container after light agitation to introduce bubbles. Labelled data of particles is collected from a newly created sample set of contaminated containers with a long pre-spin sequence to remove any bubbles from the solution.

In the first part of this research, a baseline classifier has been selected and trained. Out of four candidate classification methods that have been evaluated, the CNN classifier achieves the highest classification accuracy of 0.930 on detections with an area of at least 30 pixels. This classifier is used as a baseline throughout the rest of this research. The following research is of exploratory nature.

Next, it has been observed that there are several handcrafted features with a strong or weak correlation with the classification accuracy of a detection. Additionally, more complex correlations have been observed between multiple handcrafted features, e.g., a particle with a smaller area can achieve a high average classification accuracy if it has a high mean contrast and solidity. Simple threshold filters can be defined with some of the handcrafted features. However, this does not benefit from complex correlations and a part of the detections that will be rejected can achieve a high average classification accuracy. Therefore, this research also proposes a complex filter with a model that attempts to predict the classification accuracy of a detection based on its handcrafted features. This predicted classification accuracy could also be used in improved multi-detection classification strategies as a form of weight factor.

Next, research has been performed into the effectiveness of a single detection classification strategy. With this naïve classification strategy that is often used for classification, if a single detection is classified positive as a particle the entire container is rejected.
In addition to the False Positive (FP) rate and the False Negative (FN) rate, the False Container Rejection (FCR) rate and False Container Acceptance (FCA) rate are introduced. As many detections of bubbles can be detected from a clean container, the chance that at least one of these detections achieves a high classification score is high. Through simulation it has been found that the resulting classification threshold is 0.998, achieving an optimal FCR and FCA rate of 23.2%. This means that the single detection classification strategy is not feasible, even if the used classifier would be further optimised.

Finally, research is performed into two multi-detection classification strategies: median voting classification with tracking multi-positive quantity threshold classification.
Median voting classification uses the results of object tracking to classify a set of detections that belong to the same object. As median voting is effective at dealing with outliers of bubbles that are detected as particles, the strategy can achieve a very low FCR and FCA rate even from short partial trajectories. Multi-positive classification does not use results of tracking but requires multiple detections to be classified positive before the container is classified positive. Simulation shows that this method can also achieve a low FCR and FCA rate but requires particles to be visible on more frames than with median voting.

3.11 Recommendations

It is recommended to adopt the classifier developed in this research as a baseline to improve on in future research. Although small performance improvements can be made by further optimising the baseline classifier, it is recommended to first focus on other routes of improving the classification module as described below.

It is recommended to further research and implement a complex filter as discussed in section 3.6 to remove detections that are too difficult to classify. The classification accuracy can be predicted using a model such as a neural network for regression, and used to filter out detections if the predicted accuracy is below a certain threshold.

It is also recommended to implement a multi-detection classification strategy as proposed in section 3.8. If the tracking module can be implemented effectively, it is recommended to implement median voting classification as this can achieve the lowest FRC and FCA rate even if a particle is visible only for a few frames. Otherwise, it is recommended to implement the multi-positive classification strategy. Once a multi-detection classification strategy has been implemented and tested, this method can be adapted to use the predicted classification accuracy as a form of weight factor as shortly described in subsection 3.6.3.

4 Tracking module

The third and final module of the Visual Particle Inspection Subsystem (VPIS) is the tracking module. This chapter contains research and findings regarding this module and presents the proposed TDCP algorithm.

The goal of the tracking module is to track objects and determine what object detections are observations of the same object. This information can then be used by a multi-detection classification strategy such as median voting classification proposed in subsection 3.8.1. By classifying a container based on objects with a number of detections, the False Container Rejection (FCR) rate and False Container Acceptance (FCA) rate can be significantly increased.

Ideally, the result of tracking is a list of tracked objects where a single trajectory is found per object spanning the full trackable length during which the object is visible. If an object is not correctly tracked, the object cannot be classified accurately which would result in an increased FCA rate and risk to patient safety. If too many trajectories are detected, the FCR rate would be increased as explained in subsection 3.8.1 which is acceptable to an extent.

Object tracking is challenging as objects are small and traverse quickly through the container. As objects move a long distance between frames, it is not feasible to track objects based on overlapping sections or proximity of detections alone. As objects are small and the appearance of an object varies as it is viewed from different perspectives, it is difficult to perform tracking by recognising object appearance.

This research proposes the Trajectory Driven Cluster Proposals (TDCP) algorithm as a solution for the tracking module. This method performs object tracking based only on the positions of detections. It has been observed that trajectories of objects follow a spiralling pattern around the centre of the container. Based on this information, a motion model has been defined that can describe trajectories of real objects within a range of feasible parameters. The proposed TDCP algorithm uses this motion model to evaluate whether a set of three detection positions can be a part of the same object trajectory. Next, these short tracklets are merged into larger track candidates that describe possible trajectories.

The TDCP algorithm was first theorised in earlier work [6] where it was proposed in combination with a Siamese neural network that validates or invalidates proposed trajectories based on the similarity between detections of that trajectory. In this research, TDCP is proposed without similarity evaluation, the reasons for this are shortly discussed in subsection 4.2.1.

The proposed TDCP tracking algorithm shows potential in the quality of the tracking results as most objects are correctly tracked and correct trajectories are long. However, the current version of TDCP is not suitable to be implemented due to a phenomenon referred to as the rapid growth phenomenon explained in subsection 4.7.3. If an image sequence contains multiple objects in close proximity, the TDCP algorithm can detect a large number of possible tracklets and track candidates resulting in an unacceptably long processing time and an increased FCR rate.

A few optimisation strategies are shortly discussed. If the shortcomings of TDCP can be successfully solved with future work, TDCP could be implemented into the VPIS with great success.

4.1 Chapter structure

This chapter presents findings regarding the research and development for the tracking module. First, section 4.2 presents a short summary of previous work into particle tracking for the inspection of medicine. Next, in section 4.3, the research questions are presented that will later be used to evaluate the proposed TDCP algorithm. After that, section 4.4 presents observations and assumptions regarding the trajectory of a real object. Based on these findings, in section 4.5, a motion model is presented that can describe realistic trajectories within a feasible range of parameter values. In section 4.6, the proposed TDCP algorithm is presented. Section 4.7 presents the results of TDCP and identifies the rapid growth phenomenon. Next, in section 4.8 the proposed solution is validated by answering the main research question through four sub-questions. In section 4.9, possible optimisation strategies are discussed that could make TDCP suitable to be implemented. Section 4.10 presents a short comparison between the proposed solution and the previously identified state-of-the-art methods. Finally, sections 4.11 and 4.12 present the conclusion and recommendations for this module.

4.2 Previous work

An analysis of previous work for the problem of particle tracking has been performed during a literature study [6] preceding this thesis. This section summarises the most interesting findings from the literature study relevant for the classification module. Next, current State-Of-The-Art (SOTA) methods are identified in more detail. In section 4.10, the TDCP algorithm proposed for the tracking module is compared to the SOTA methods.

Tracking methods

The tracking methods referred to in the literature study [6] can be separated into ten groups: proximity tracking [18] [7], proximity clustering [21] [23], corner feature tracking [20], handcrafted feature matching [8] [10] [15], handcrafted feature bottom-up clustering [15], classification tracking [26], ALW-CSM [12], CSR-DCF [22], Siamese similarity tracking [35], and Trajectory Driven Cluster Proposals (TDCP). Siamese similarity tracking and TDCP have not been used in previous work for the problem of particle tracking. Siamese similarity tracking is included as in the previous years, tracking methods that use a Siamese similarity metric have been applied for general tracking problems with much success. TDCP was first proposed during the literature study as a novel solution for tracking.

4.2.1 Current SOTA

This section explains the workings and shortcomings of the SOTA methods that have been identified in previous work. During the literature study it was theorised that of the presented methods, the combination of TDCP with a Siamese similarity metric would perform the best. Although this method is not used in previous work, it is included as SOTA method as it can serve as a useful comparison for the proposed solution which does not include a similarity metric. The best SOTA method that is actually used in previous work is handcrafted feature bottom-up clustering [15].

TDCP with Siamese similarity metric

During the literature study, the TDCP algorithm proposed in section 4.6 was first theorised. The workings of TDCP were not yet fully defined but it was hypothesized that a position-based tracking algorithm can be used to detect what detections could possibly be detections of the same object.

Next, a Siamese neural network would be used to validate the proposed trajectories. A Siamese neural network is a special type of neural network that can determine the similarity between two instances of objects. Such a network can be trained to recognise objects from different perspectives [36].

By validating the proposed trajectories, the number of additional false trajectories can be lowered which should lower the False Container Rejection (FCR) rate as explained in subsection 3.8.1. However, if a correct particle trajectory is falsely invalidated, this particle would not be classified which has a significant negative result on the False Container Acceptance (FCA) rate.

A shortcoming of this method is that, although a Siamese neural network can be trained to recognise objects from different perspectives, it is expected that it is difficult for the network to differentiate detections from different bubbles. This is as bubbles can only be different in terms of object size which can be very similar. As a result of this, many of the additional trajectories with only bubbles would not be invalidated, not lowering the FCR significantly. The method would be able to differentiate between detections of particles and bubbles and therefore invalidate false trajectories containing detections with both object types. However, this is not useful to lower the FCR as the container should be rejected if a particle is present. Another shortcoming of this method is that a Siamese neural network requires a significant amount of training data where detections are labelled per object. This requires manual labelling of a large number of object trajectories.

For these reasons, it was decided not to develop a similarity metric to validate the results of TDCP, but to develop TDCP such that its results can be used for classification directly.

Handcrafted feature bottom-up clustering

The method handcrafted feature bottom-up clustering [15] is the best method used in previous work for particle tracking. The method extracts 10 different handcrafted features similar to the handcrafted feature classification method presented in subsection 3.5.2 as a candidate for the classification module. The handcrafted features used by the paper are mean brightness, area, 7 invariant Hu moments [37], and the Euclidean distance between detections. Based on these features and a series of equations, the method compares sets of two detections to determine a similarity metric similarly to a Siamese similarity metric.

To perform bottom-up clustering, each detection is first assigned to its own cluster. Next, iteratively the two clusters with the highest similarity between its members are merged into a larger cluster. This process continues until the distance between clusters is above a specified threshold, after which clustering stops. Incomplete clusters that do not contain a detection in every frame are discarded.

This method has a number of shortcomings not addressed in the paper. The first shortcoming of this method is that bottom-up clustering is not ideal for tracking. This is as the method does not account for in what frame a detection occurs. Therefore, many of the detected clusters will have multiple detections which are observed from the same frame which is not possible in reality. If this would be accounted for by only merging clusters that do not have any overlapping frames, more clusters would be incomplete or negatively affected by local minima. The biggest shortcoming is that the method discards every incomplete cluster as image noise. In reality it can happen often that an object is not visible in every frame or moves out of view. Finally, it is expected that the similarity metric determined by this method is not very effective to differentiate between detections of different bubbles.

4.3 Research questions

This section introduces the research question and sub-questions for the tracking module. The main research question for this module regards the effectiveness and suitability of the TDCP algorithm proposed in this research as a solution for the tracking module.

C: TDCP algorithm effectiveness and suitability
"Is the proposed TDCP algorithm effective and suitable to be implemented in a tracking module for the VPIS?"

In order to answer research question C more effectively, it has been divided into the following sub-questions. Sub-question C.1 is required for the algorithm to be suitable for implementation in the VPIS. Sub-questions C.2 – C.4 regard the quality of the tracking results and therefore the effectiveness of the multi-detection classification method that would use the tracking results. For validation, it is assumed that the tracking results will be used for median voting classification. Each sub-question is further detailed and evaluated separately in subsections 4.8.1 – 4.8.4.

C.1: Processing time
"Can tracking be performed on two sequences with 100 and 200 detections within one second, as not to slow down the VPIS?"

C.2: Correctly tracked objects
"What portion of the moving objects is correctly tracked at least once for a significant length of the visible object trajectory?"

C.3: Additional trajectories
"How many additional trajectories are detected besides the most correct trajectories of objects?"

C.4: Trajectory length
"What portion of the full trackable length is tracked for the most correct trajectories of objects?"

4.4 Trajectory observations and assumptions

This section presents observations and following assumptions from real object trajectories. The goal has been to observe characteristics of real object trajectories and determine if these trajectory characteristics can be recognised to distinguish between real object trajectories and false trajectories. First, observations are made from a small dataset with annotated trajectories. Next, assumptions are made about how real trajectories can be differentiated from false trajectories. The findings from this section are used to define a motion model in section 4.5.

4.4.1 Trajectory observations

A small trajectory dataset has been annotated using a newly developed annotation tool. The annotation tool shows images with the segmentation masks of all objects from the background subtraction and segmentation module. The user can navigate through the sequence using the keyboard buttons and mouse and select what detections belong to the same objects.

The annotated tracking dataset consists of three image sequences, containing a total of 13 moving objects with a total of 211 detections over 19 frames. The graphs in Figure 29 show the annotated trajectories of 8 different moving objects.

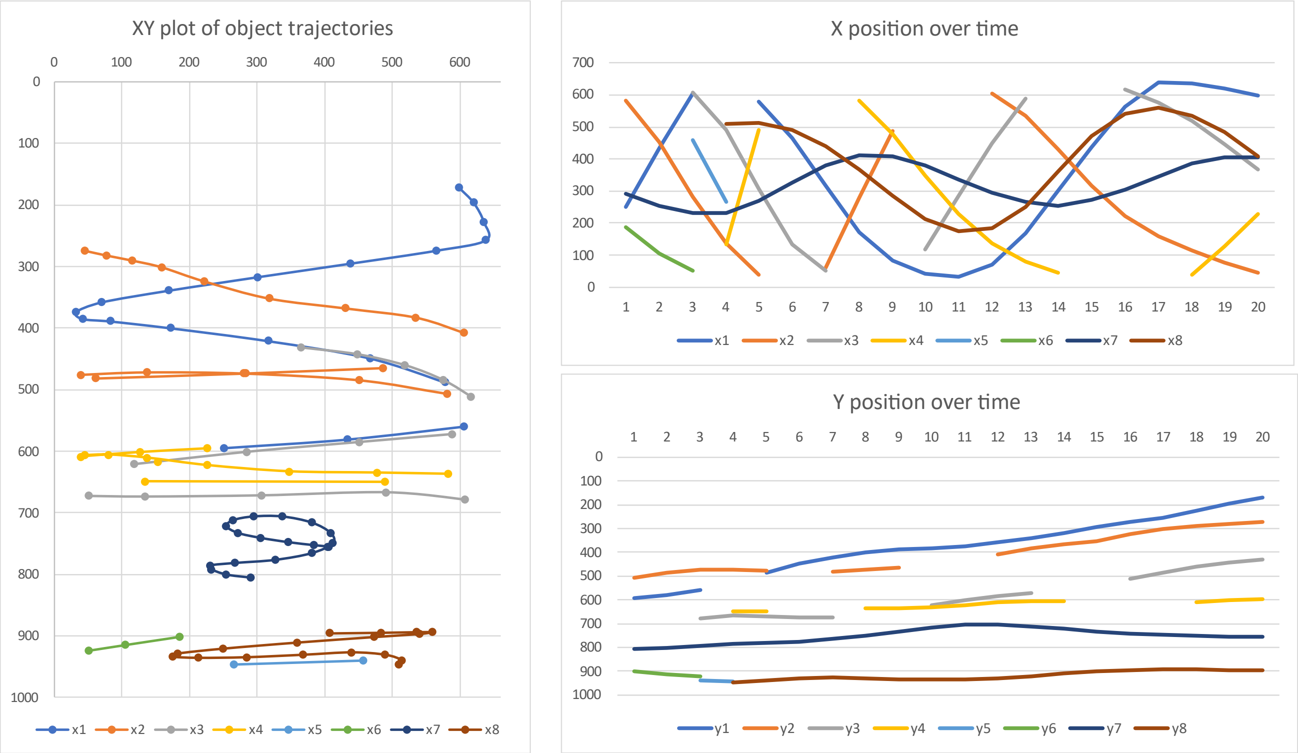


Figure 29 Left: XY plot of observed trajectories. Top-right: Horizontal object position over time. Bottom-right: Vertical object position over time. The figure shows that most objects move in a spiralling pattern around the vertical axis of the container.

From the figures can be seen that objects do not move in random directions but mostly follow a smooth spiralling path around the vertical axis of the container. A false trajectory with detections of different objects is more likely to contain discontinuities, resulting in a non-spiralling or non-smooth trajectory. Most objects are not visible in every frame. This is as some objects in the dataset are very small and objects are less visible near the sides of the container. Objects can also go out of view near the top and bottom regions of the container.

Observations from the horizontal and vertical motion in tracking data are summarised in the following lists.

Horizontal movement:

- Most observed objects spin around the vertical axis of the container. As a result of this, the horizontal position over time is approximately sinusoidal.
- The radius of rotation at which objects spin around the vertical axis is different per object. This radius of rotation can slowly change over time.
- The frequency of rotation is similar between objects with minor differences. Similar to the radius of rotation, the frequency of rotation can slowly change over time.

Vertical movement:

- At the start of the image sequence objects can be anywhere in the vertical range of the container.
- Objects move up or down with limited vertical velocity and acceleration.
- The direction of movement for particles and bubbles can be either up or down as the object is affected by the turbulent flow. Based on the observations in Figure 29 alone, it can be argued that some objects follow an upwards trajectory which could indicate that these objects are bubbles. However, in numerous occasions it has been observed that bubbles move down during the image sequence, only to drift to the surface after the liquid has slowed down.

4.4.2 Trajectory assumptions

The following assumptions are made based on the observations on object trajectories.

> *Assumption 1: Most if not all trajectories of real objects behave as described in subsection 4.4.1.*
>
> *Assumption 2: Most false trajectories with detections of multiple objects do not behave as described in subsection 4.4.1.*
>
> *Assumption 3: A motion model that can exclusively describe a trajectory that behaves as described in subsection 4.4.1 can be used to differentiate between trajectories of real objects and false trajectories.*

Assumption 3 is made based on assumptions 1 and 2 and is used as a working principle for the TDCP algorithm. Based on these assumptions, it was decided to develop the motion model and later the TDCP algorithm.

Later during development, it was found that assumption 1 is partially false as a portion of the trajectories show unexpected trajectory behaviour. This is explained in the following subsection.

4.4.3 Unexpected trajectory behaviour

This subsection describes an observation made in a later stage of the development of the tracking module. This describes a phenomenon that is problematic for the effectiveness of the tracking module.

Although most objects act according to the expected trajectory behaviour described above, some objects some objects behave unexpectedly. In Figure 29 can be seen that object 6 in green suddenly slows down during frames 17 – 20. In the dataset that was used for tracking captured in an early stage of the research, this unexpected trajectory behaviour is rare, occurring in roughly 5% of the trajectories for a part of the trajectory. However, with newer datasets, this unexpected trajectory behaviour occurs in roughly 20% of the trajectories for a significant part of the trajectory. This problem was only realised after most of the tracking module had been developed.

Through limited testing, it has been observed that small changes in the spin-stop sequence and image capture sequence have an effect on the resulting trajectory behaviour. It is not known what exact configuration was used during the collection of the first dataset or how the unexpected trajectory behaviour can be minimised. After this observation was made, development of the tracking module was put on hold. Future work is required to learn how to minimise this phenomenon.

4.5 Motion model

This section presents the motion for the tracking module. First, it is discussed what the ideal characteristics are for a motion to be effective with the TDCP algorithm. Next, motion model is presented. The proposed motion model consists of two equations that describe the horizontal and vertical position of an object over time. The motion model also defines a feasible parameter range. If a trajectory can be described by the motion model within the feasible parameter range, the trajectory is possibly of a real moving object.

4.5.1 Ideal motion model characteristics

This subsection discusses what characteristics are desired for a motion model to be suitable and effective to be used with TDCP. Every possible trajectory can bed described using a complex polynomial trajectory. However, such a set of polynomial equations does not make a good motion model as it can describe any trajectory, including unrealistic false trajectories. The ideal motion model can describe any real trajectory while omitting false trajectories.

The following list contains some desired characteristics for the motion model. The ideal motion model:

1. Can describe a realistic object trajectory with a minimum number of parameters, such that a possible trajectory can be fitted based on a minimum number of points. This way, smaller trajectories can be validated of objects that are visible for only a few frames. This will also reduce computation time as less possible point combinations need to be evaluated.
2. Contains independent parameters of which at least some are limited to a feasible range, such that trajectories that can only be described with parameters outside of this feasible range can be omitted as false trajectories.
3. Is a function of time, such that the model can predict the position of an object over time and can be fitted on trajectories with gaps.
4. Contains parameters that are easily understandable and can be configured based on observations of real trajectories.

4.5.2 Proposed motion model

This subsection presents the proposed motion. The proposed motion model consists of two equations describing the horizontal and vertical position of an object over time, together with the feasible ranges for the motion model parameters.

Horizontal motion

As described in section 4.4, most observed objects do not move in random directions but follow a smooth spiralling path around the vertical axis of the container. In the horizontal axis, this motion can be described locally with a sine wave. The horizontal position of an object is described with Equation 10. Where $x(t)$ is the horizontal position of an object at time t, $t = 0$ at the first detection in the set, θ_0 is the phase of rotation at $t = 0$, f is the frequency of rotation, and r is the radius of rotation. Figure 30 visualises the principle of the horizontal motion equation.

$$x(t) = \sin(\theta_0 + f * 2\pi * t) * r \qquad (10)$$

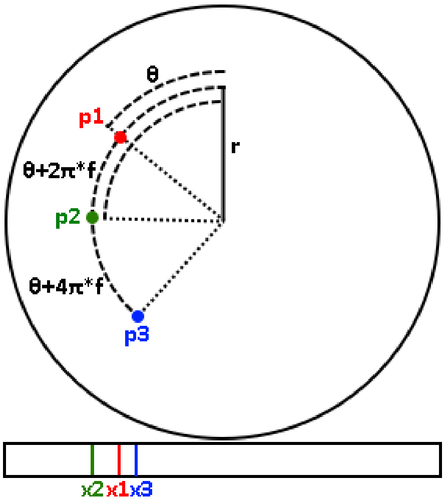


Figure 30 Visualisation of the horizontal motion equation. The figure shows how r, θ_0, f are used to determine horizontal x position.

As the equation for horizontal motion is harmonic, there is an infinite amount of parameter combinations that can describe a trajectory through any set of points by making multiple full rotations between detections. By limiting the frequency of rotation f to realistic observed values, and the radius of rotation r to the radius of the container, only a single valid solution should be feasible.

Vertical motion:

The vertical motion observed from real trajectories is random with a limited velocity and acceleration. In the vertical axis, this motion can be described locally with a second degree polynomial equation.

The vertical position of an object is described with Equation 11. Where $y(t)$ is the vertical position of an object at time t, $t = 0$ at the first detection in the set, y_0 is the vertical position at $t = 0$, and $\dot{y}$ and $\ddot{y}$ describe the vertical velocity and acceleration respectively. By limiting the vertical velocity and acceleration, false trajectories with an unrealistic vertical motion can be omitted.

$$y(t) = y_0 + \dot{y} * t + \ddot{y} * t^2 \qquad (11)$$

Feasible ranges

The feasible ranges for the motion model parameters have been observed from annotated trajectory data or from fitting and have been confirmed with testing. The feasible parameter ranges are:

Parameter	Description	Range	Source
$\boldsymbol{\theta_0}$	Phase at $t = 0$	$(0{,}2\pi)$	Full rotation
$\boldsymbol{f}$	Frequency of rotation	$\left(\frac{1}{15},\frac{1}{10}\right)$	Observed from data and tuned with fitting. See paragraph below
$\boldsymbol{r}$	Radius of rotation	$\left(0,\frac{container\ width}{2}\right)$	Container width
$\boldsymbol{y_0}$	Vertical position at $t = 0$	$(0, container\ hight)$	Container height

$\dot{\boldsymbol{y}}$	Vertical velocity in pixels/frame	$(-50{,}50)$	Observed from data
$\ddot{\boldsymbol{y}}$	Vertical acceleration in pixels/frame^2	$(-6{,}6)$	Observed from data

As the image is deformed by the shape of the container, the horizontal position at which an object is observed, can deviate slightly from where the object was at that moment. As a result of this, objects are observed to be moving slightly faster when moving through a region at the back of the container. To accommodate for this, the feasible parameter range for the frequency of rotation f is slightly wider than can be expected when closely inspecting Figure 29. It is not easily possible to compensate for this by transforming the horizontal position of each detection as the deformation is affected by the depth of the moving object which is unknown. This issue could possibly be negated without widening the parameter range for f by utilising a more complex motion model. However, this has not been researched any further as this motion model is effective.

Motion model characteristics

The proposed motion model has three independent parameters per axis of motion. Therefore, the motion model can be fitted with only three points of detection coordinates such that short sections of trajectories can be modelled, and processing time is minimised. Two out of three parameters for each equation are limited to a feasible range, making it possible differentiate false trajectories. As the equations are functions of time, it is possible to fit trajectories with missing detections and to predict object position in missing, previous, or later frames. The feasible parameter range can be determined easily from observations of real trajectories, this makes the model easy to understand and reconfigure for different medicine containers.

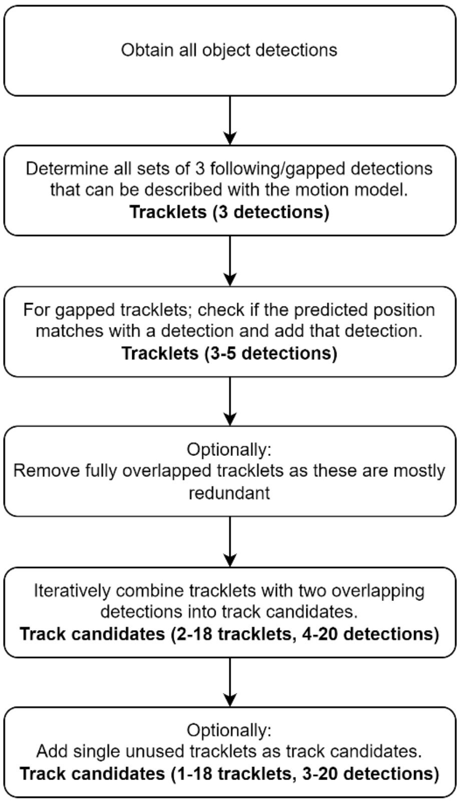
4.6 Proposed TDCP algorithm

This section gives an overview of the TDCP algorithm process.

Before TDCP is performed, a complete list of object detections with positions is provided by the background subtraction and segmentation module.

First, from the list of object detections, it is determined with the motion model what sets of three detections can be a part of a real object trajectory. Such a set is called a **Tracklet**. Every possible combination of three detections from subsequent frames or with one or two single frame gaps is checked, such that the method can track tracklets with missing frames. The number of calculations for this step has been minimised which is important as many possible combinations have to be checked.

Secondly, for each **tracklet with a gap**, the object position in the missing frame is predicted using Equation 10 and Equation 11). If the predicted position matches closely to an object detection, this detection is added into the tracklet. This results in tracklets that contain 3-5 detections. After filling gapped tracklets, some tracklets become mostly redundant as they describe detections that are also present in another larger tracklet. Removing these tracklets results in a few less trajectory candidates but can significantly decrease the processing time in subsequent steps.



Obtain all object detections
Determine all sets of 3 following/gapped detections that can be described with the motion model.
Tracklets (3 detections)
For gapped tracklets; check if the predicted position matches with a detection and add that detection.
Tracklets (3-5 detections)
Optionally:
Remove fully overlapped tracklets as these are mostly redundant
Iteratively combine tracklets with two overlapping detections into track candidates.
Track candidates (2-18 tracklets, 4-20 detections)
Optionally:
Add single unused tracklets as track candidates.
Track candidates (1-18 tracklets, 3-20 detections)

Figure 31 Steps overview of the proposed TDCP algorithm.

Next, **Track candidates** are formed by combining tracklets with two or more overlapping detections. Track candidates are grown progressively by adding more tracklets in a way that minimises the computational cost. This process returns every possible track candidate based on the detected tracklets. Optionally, the motion model parameters of the tracklets can be compared to ensure that overlapping tracklets describe a similar trajectory.

Optionally, **singular tracklets** that are not present in any track candidates can be added as track candidates, such that short trajectories of 3-5 detections are also included in the result.

4.7 Results

This section presents the results of the TDCP algorithm. The results are shown from two sequences from the small, annotated tracking dataset. These two image sequences contain six objects each and are referred to as sequence 1 and sequence 2.

The results are presented in two parts. First, a table is presented with numerical results including processing time. Next, the detected track candidates are visualised, and the detected track candidates are presented in a table showing the quality of the tracking results. Finally, an observed phenomenon referred to as the rapid growth phenomenon is explained.

4.7.1 Numerical results

Table 13 presents numerical results from TDCP for sequence 1, sequence 2 and both sequences combined. The table shows data from tracklet formation and subsequent track candidate formation, with and without the recommended filtering step that removes overlapping tracklets. The processing time is given for each step.

Table 13 Numeric results for sequence 1, 2 and both sequences combined.

	Seq. 1	Seq. 2	Seq. 1+2
Number of detections	110	111	221
Number of objects	6	6	12
Tracklet formation:			
Tracklets unfiltered (incl. overlapping)	191	350	857
Tracklets filtered (excl. overlapping)	59	126	417
Processing time	320 ms	585 ms	1611 ms
Track candidate formation from filtered tracklets			
Nr. of tracklets	59	126	417
Track candidates unfiltered	10	85	396
Track candidates	8	49	332
Processing time	18 ms	92 ms	1034 ms
Total processing time	338 ms	677 ms	2645 ms
Track candidate formation from unfiltered tracklets			
Nr. of tracklets	191	350	857
Track candidates unfiltered	50	142	886
Track candidates	9	52	346
Processing time	85 ms	823 ms	7503 ms
Total processing time	405 ms	1262 ms	9114 ms

From the figure can be seen that the number of tracklets, track candidates and the processing time varies strongly between sequence 1, sequence 2 and both sequences combined. This is caused by the rapid growth phenomenon explained in subsection 4.7.3.

4.7.2 Tracking quality results

Figure 32 – Figure 35 visualise the tracking results for sequences 1 and 2. Figure 32 and Figure 34 visualise the trajectories of the detected track candidates. Figure 33 and Figure 35 present the detected track candidates in the form of a table from which the quality of the tracking results can be seen.

Viewing guide for track candidate tables

Figure 33 and Figure 35 show the detected track candidates in the form of a table. This method of visualisation most effectively shows the performance of TDCP to track moving objects.

The left section of the table shows the ground-truth presence of annotated moving objects. It shows for each annotated object in what frames the object is visible and has been annotated.

The centre section shows the resulting trajectory candidates as determined by TDCP. When TDCP returns the list of trajectory candidates, it is not known what trajectory candidates are correct or to what real object each of the detections correspond. For the purpose of evaluation, each detection in the trajectory candidates has been linked to the ground truth object that it belongs to. This is shown with the corresponding number and colour.

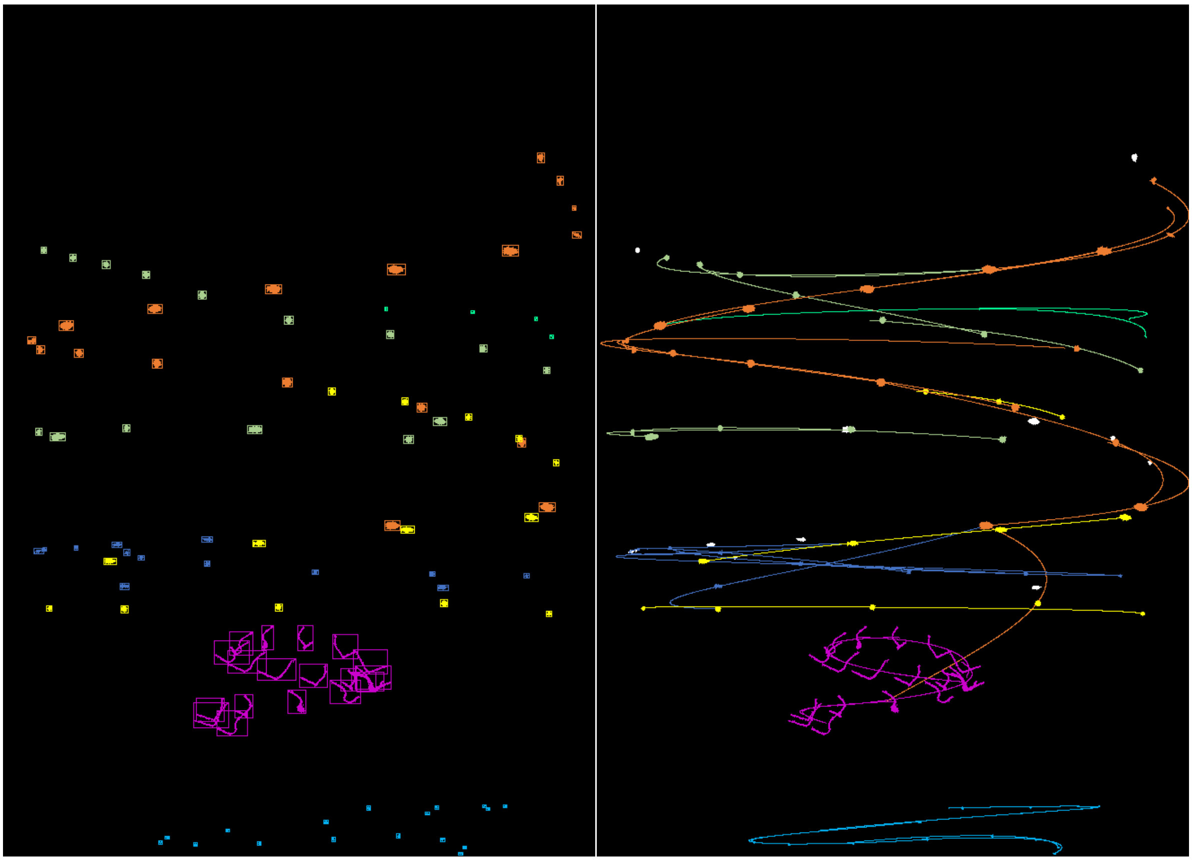
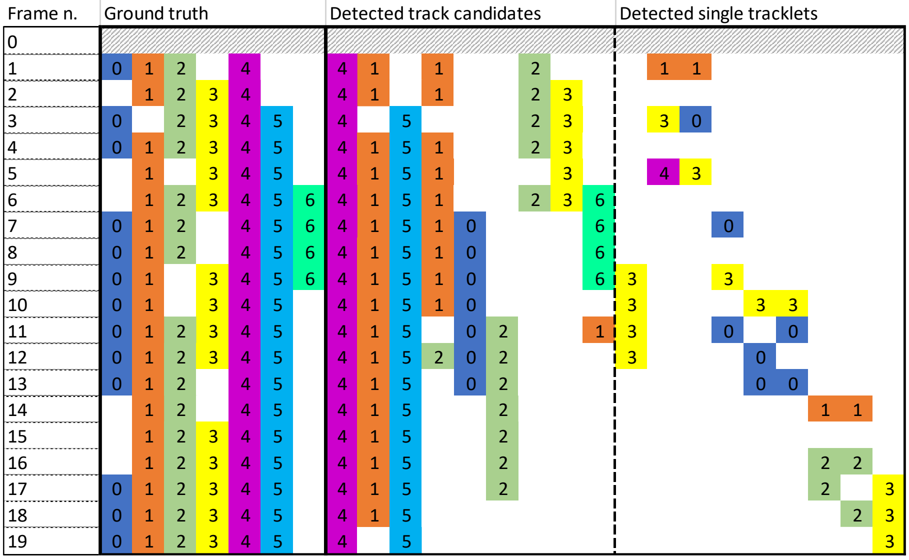


Figure 32 Trajectory visualisation of sequence 1, Left: Ground truth detections with colours from Figure 32. Right: Detected trajectory candidates including single tracklets, the track colour is determined by the colour of the dominant ground truth object in each track candidate.



Frame n.	Ground truth							Detected track candidates									Detected single tracklets								
0																									
1	0	1	2		4			4	1		1			2				1	1						
2		1	2	3	4	5		4	1		1			2	3										
3	0		2	3	4	5		4		5				2	3			3	0						
4	0	1	2	3	4	5		4	1	5	1			2	3										
5		1		3	4	5		4	1	5	1				3			4	3						
6		1	2	3	4	5	6	4	1	5	1			2	3	6									
7	0	1	2		4	5	6	4	1	5	1	0				6				0					
8	0	1	2		4	5	6	4	1	5	1	0				6									
9	0	1		3	4	5	6	4	1	5	1	0				6	3			3					
10	0	1		3	4	5		4	1	5	1	0					3				3	3			
11	0	1	2	3	4	5		4	1	5		0	2			1	3			0		0			
12	0	1	2	3	4	5		4	1	5	2	0	2				3				0				
13	0	1	2		4	5		4	1	5		0	2								0	0			
14		1	2		4	5		4	1	5			2										1	1	
15		1	2	3	4	5		4	1	5			2												
16		1	2	3	4	5		4	1	5			2										2	2	
17	0	1	2	3	4	5		4	1	5			2										2		3
18	0	1	2	3	4	5		4	1	5														2	3
19	0	1	2	3	4	5		4		5															3

Figure 33 Tracking results for sequence 1 presented as a table.
***Left**: Object annotations per frame, each column represents an annotated object and shows in what frames (rows) the object was visible. E.g., object 6 was visible in frames 6-9.*
***Centre**: Detected track candidates; each column represents a detected track candidate and shows what detections are present in each candidate, the number for each detection corresponds to the ground truth label for that detection and has been assigned afterwards for evaluation.*
***Right**: Single tracklets that are not present in a track candidate consisting of multiple tracklets, these can be included as track candidates but are generally less accurate.*

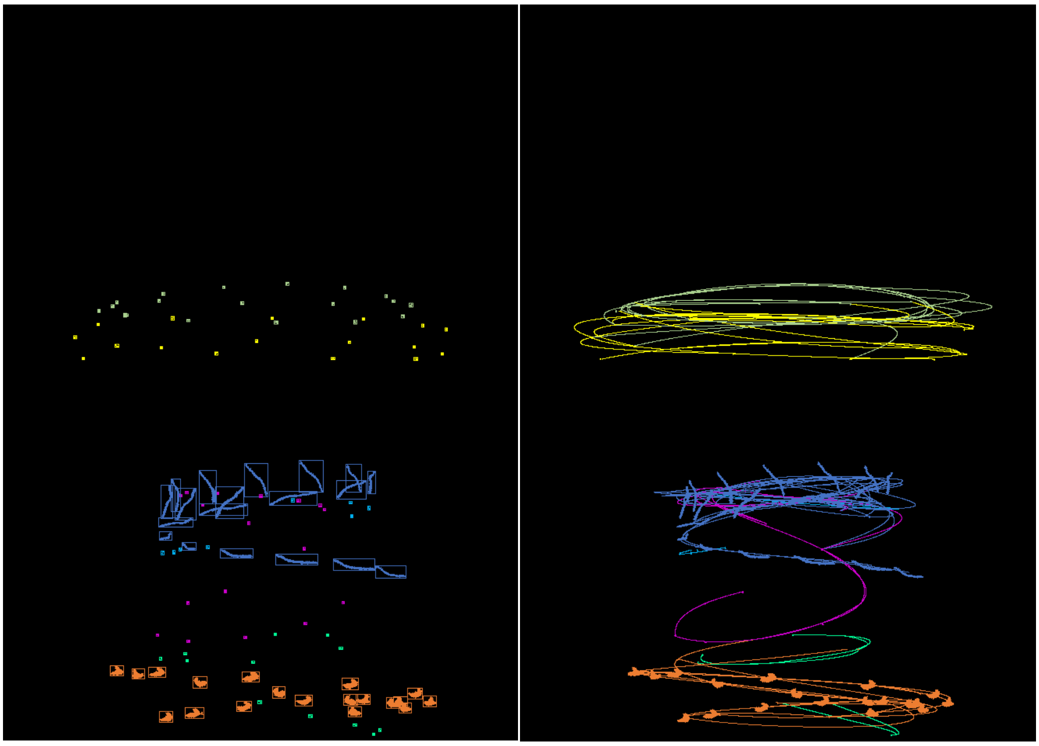


Figure 34 Trajectory visualisation of sequence 2. See caption below Figure 32.

Frame n.	Ground truth							Detected track candidates part 1																										
0																																		
1	0	1	2		4		6	2	0	0	0	0	0	1	1	2	2	2	2	2	2	0	0	0	2		0	0				0	0	1
2	0	1	2	3	4		6	2	0	0	0	0	0	1	1	3	3	2	2	2	2	0	0	0	2	2	0	0				0	0	1
3	0	1	2	3			6	2	0	0	0	0	0	1	1	3	3	3	2	2	2	0	0	0	3	3	0	0				0	0	1
4	0	1	2	3	4		6	2	0	0	0	0	0	1	6	3	3	3	2	2	2	0	0	0	3	3	0	0	4	4		0	0	1
5	0	1	2	3	4		6	2	0	0	0	0	0	1	1	3	3	3	2	2	2	0	0	0	3	3	0	0	4	4	6	0	0	1
6	0	1	2	3	4			2	0	0	0	0	0	1	1	3	3	3	3	3	3	0	0	0	3	3	0	0	4	4	1	0	0	1
7	0	1	2	3	4			2	0	0	0	0	0	1	1	3	3	3	3	3	3	0	0	0	3	3	0	0	4	4	1	0	0	1
8	0	1	2	3	4		6	2	0	0	0	0	0	1	1	3	3	3	3	3	3	0	0	0	3	3	0	0	4	4	1	0	0	1
9	0	1	2	3				2	0	0	0	0	0	1	1	3	3	3	3	3	3				3	3	0	0			1	0		1
10	0	1	2	3	4	5		2	0	0	0	0	0	1	1	3	3	3	3	3	3	5	5	5	3	3	0	0	4	4	1	0	5	1
11	0	1	2	3	4	5	6	2	0	0	0	5	5	1	1	3	3	3	3	3	3	5	0	0	3	3	0	0	4	4	1	5	0	1
12	0	1	2	3	4	5	6	2	0	0	0	0	5	1	1	3	3	3	3	3	3	0	0	0	3	3	0	0	4	4	1		0	1
13	0	1	2	3	4	5		2	0	0	4	0	5	1	1	3	3	3	3	3	3	0	0	0	3	3	0	4	4	4	1	4		
14	0	1	2	3	4		6	2	0	0	4	0	4	1	1	3	3	3	3	3	3	0	0	0	3	3	4	4	4	4	1		4	6
15	0	1	2	3	4			2	0	4	4	0	4	1	1	3	3	3	3	3	3	0	0	4	3	3			4	0	1	0		
16	0	1	2	3	4	5	6	2	0	0	4	0	0	1	1	3	3	3	3	3	3	0	0	0	3	3	0	0	4	4	1		0	
17	0	1	2	3	4	5		2	0	0	4	0	0	1	1	3	2	3	3	3	2	0	0	0		2			4	4	1			
18	0	1	2	3	4	5	6	2	0	0	4	0	0	1	1	3	2	3	3	2	2	0	0	0	2	2			4	4	1			
19	0	1	2		4	5	6	2	0	0	4	0	0	1	1							0	0	0					4	4	1			

Frame n.	Detected track candidates part 2																										Detected single tracklets							
0																																		
1	1							2									1	2	2				6				6	1	4					
2	1							2									1			3	3		6				6		4	3				
3	1							2									1	3	3	3	3		6				6	6		3				
4	6							2								4	6	3	3	3	3		6				1		4					
5	1							2	6							4	1	3	3	3	3		6					1		2				
6	1							2	1							4	1	2	2	2	2										4			
7	1	3	2	2	2			2	1							4			3		3													
8	1	2	3	3	3	3		2	1							4	6	3		3											6			
9	1	2	3	3	3	2	3	2	1																									
10	1	2	3	3	3	2	2		1	0	0	0	5	5	5	4								4	5						1	4	4	5
11	1	2	3	3	3	2	2	3	1	5	4	4	5	4	4	4						6												
12	1	2	3	3	3	2	2		1	5	4	4	5	4	4	4						6		5	5							5	5	4
13		2	3	3	3	2	2		1	5	4	4	5	4	4																	0		
14	6	2	3	3	3	2	2		6	0	4	4	0	4	4	0						6		4	4								0	0
15		2	3	3	3	2	2			4	4	0	4	4	0																			
16		2	3	3	3	2	2			0	4	4	0	4	4							6		0	0	5								
17		2	3	3	2	2	2			0	4	4	0	4	4											5								
18		2	3	2	2	2	2			0	4	4	0	4	4							6				5								
19		2				2	2			0	4	4	0	4	4							6				5								

Figure 35 Tracking results for sequence 2 as a table. See caption below Figure 33.

4.7.3 Rapid growth phenomenon

This section describes an observed phenomenon referred to as the *rapid growth phenomenon*. This phenomenon is problematic for the current version of TDCP as it can result in a long processing time and a large quantity of additional detections.

Sequence 1 and 2 both contain six objects resulting in a similar number of detections. However, in Table 13 can be seen that for sequence 2 compared to sequence 1, about two times more tracklets are detected after filtering, resulting in seven times more track candidates and twice as much processing time. With both detections of sequences combined in a single sequence, the resulting processing time is four times more than for sequence 2.

This difference between sequence 1 and 2 is explained by the fact that in sequence 2 more objects are close to other objects during the same frames. If more objects are close together, more feasible but false tracklets can exist between detections of other objects, resulting in an increase in possible track candidates and processing time. An example of this can be seen in Figure 35, where 19 track different candidates are detected that contain detections of objects 2 and 3 combined. In Figure 34 can be seen that these objects are positioned close together.

As demonstrated with both sequences combined, the presence of more objects in a sequence results in a further rapid growth of detected tracklets, track candidates and subsequent processing time to the point where it is no longer practical.

The rapid growth of processing time cannot be characterised as exponential or quadratic as it depends on the proximity of objects during the image capture sequence. In the worst-case scenario, if two objects are close together during the entire sequence, this can cause up to 544 tracklets to be detected including overlapping tracklets. This can in turn lead to a theoretical maximum of 1,048,576 track candidates which would take an unacceptable amount of processing time.

In this thesis, no solution is developed that deals with the rapid growth phenomenon. In section 4.9, some methods are shortly discussed that could possibly solve this issue. It is recommended to perform future research into dealing with the rapid growth phenomenon.

4.8 Validation

The main research question which will be answered through validation is:

C: *“Is the proposed TDCP algorithm effective and suitable to be implemented in a tracking module for the VPIS?”*

The TDCP algorithm will be validated by analysing the results and answering the sub-questions C.1 – C.4 as presented in section 4.3. Each sub-question is validated in its own subsection. Afterwards, the main research question C for the tracking module will be answered.

Sub-questions C.2 – C.4 regard the quality of the tracking results and therefore the effectiveness of the multi-detection classification method that would use the tracking results. For validation, it is assumed that the tracking results will be used for median voting classification.

4.8.1 C.1 Processing time

This section aims to answer sub-question C.1.

C.1: *“Can tracking be performed on two sequences with 100 and 200 detections within one second, as not to slow down the VPIS?”*

In order for the tracking module to be implemented in the VPIS without slowing down the total processing time of a container, processing of two sequences with 100 and 200 detections should take no longer than 1 second. This dictates whether the method is suitable for implementation.

From the numerical results in Table 13 can be seen that, with the current version of TDCP, the processing time varies strongly between the sequences depending on the number of tracklets and subsequent number of track candidates. This variance is caused by the rapid growth phenomenon explained in subsection 4.7.3. When processing two sequences of 100 and 200 detections, it is likely that the resulting processing time is higher than 1 second.

In its current form, TDCP algorithm does not satisfy the processing time requirement and is therefore not suitable to be implemented in the VPIS. However, with optimisation methods as shortly discussed in section 4.9 it might be possible to decrease the processing time and pass this requirement.

4.8.2 C.2 Correctly tracked objects

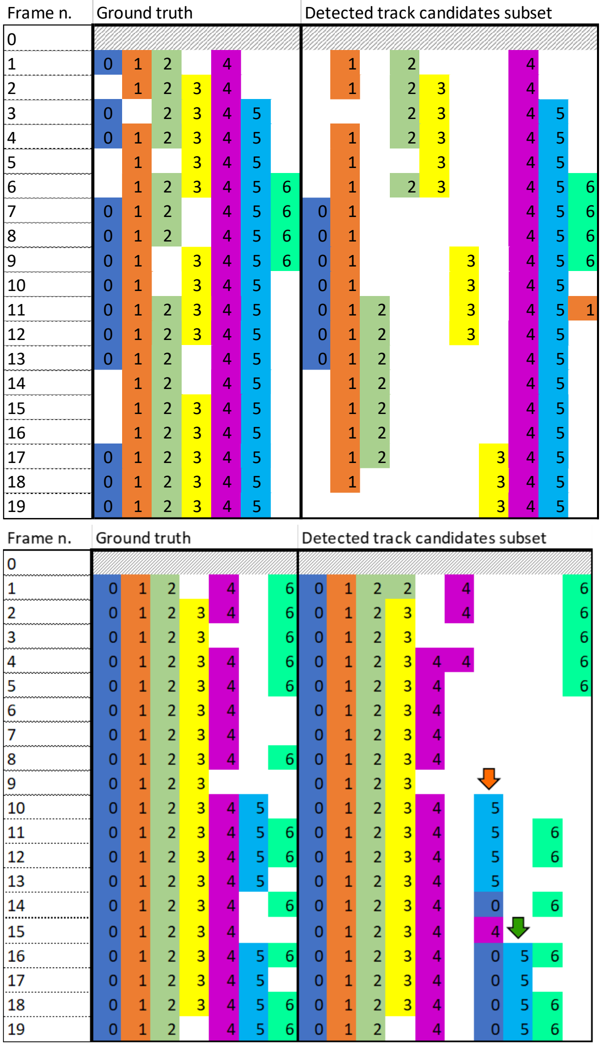
This section aims to answer sub-question C.2:

C.2: *“What portion of the moving objects is correctly tracked at least once for a significant length of the visible object trajectory?”*

What portion of the moving objects is correctly tracked at least once is arguably the most important aspect of tracking quality. If a particle object is present in a sequence but the detections of the object are not tracked or only tracked in a false trajectory, the particle will not be classified, and the container will be falsely accepted. Therefore, the portion of moving objects that is not correctly tracked strongly relates to the False Container Acceptance (FCA) rate for median voting classification presented in subsection 3.8.1. It is important that each object is correctly tracked, even if the object is only visible for a few frames.

As it is assumed that the results of tracking will be used with median voting classification, a trajectory can contain some detections of other objects without strongly affecting the classification result. Therefore, a trajectory is considered correct if at least 75% of the detections are from the same object.

Figure 36 shows a subset of the tracking results from Figure 33 and Figure 35 with only the most correct trajectories. In the figures can be seen that every moving object present in the sequences is tracked correctly for a significant length at least once.



Frame n.	Ground truth							Detected track candidates subset									
0																	
1	0	1	2		4				1		2				4		
2		1	2	3	4				1		2	3			4		
3	0		2	3	4	5					2	3			4	5	
4	0	1	2	3	4	5			1		2	3			4	5	
5		1		3	4	5			1			3			4	5	
6		1	2	3	4	5	6		1		2	3			4	5	6
7	0	1	2		4	5	6	0	1						4	5	6
8	0	1	2		4	5	6	0	1						4	5	6
9	0	1		3	4	5	6	0	1				3		4	5	6
10	0	1		3	4	5		0	1				3		4	5	
11	0	1	2	3	4	5		0	1	2			3		4	5	1
12	0	1	2	3	4	5		0	1	2			3		4	5	
13	0	1	2		4	5		0	1	2					4	5	
14		1	2		4	5			1	2					4	5	
15		1	2	3	4	5			1	2					4	5	
16		1	2	3	4	5			1	2					4	5	
17	0	1	2	3	4	5			1	2				3	4	5	
18	0	1	2	3	4	5			1					3	4	5	
19	0	1	2	3	4	5								3	4	5	

Frame n.	Ground truth							Detected track candidates subset									
0																	
1	0	1	2		4		6	0	1	2	2		4				6
2	0	1	2	3	4		6	0	1	2	3		4				6
3	0	1	2	3			6	0	1	2	3						6
4	0	1	2	3	4		6	0	1	2	3	4	4				6
5	0	1	2	3	4		6	0	1	2	3	4					6
6	0	1	2	3	4			0	1	2	3	4					
7	0	1	2	3	4			0	1	2	3	4					
8	0	1	2	3	4		6	0	1	2	3	4					
9	0	1	2	3				0	1	2	3						
10	0	1	2	3	4	5		0	1	2	3	4		5			
11	0	1	2	3	4	5	6	0	1	2	3	4		5		6	
12	0	1	2	3	4	5	6	0	1	2	3	4		5		6	
13	0	1	2	3	4	5		0	1	2	3	4		5			
14	0	1	2	3	4		6	0	1	2	3	4		0		6	
15	0	1	2	3	4			0	1	2	3	4		4			
16	0	1	2	3	4	5	6	0	1	2	3	4		0	5	6	
17	0	1	2	3	4	5		0	1	2	3	4		0	5		
18	0	1	2	3	4	5	6	0	1	2	3	4		0	5	6	
19	0	1	2		4	5	6	0	1	2		4		0	5	6	

Figure 36 Subset of the tracking results shown in Figure 33 and Figure 35 with only the most correct trajectories.

Although the results are satisfactory, some imperfections can be observed. The seventh track candidate of sequence 2 (highlighted with an orange arrow) is a false trajectory candidate as it contains more than 25% detections from different objects. If object 5 would only be visible during frames 10-13, the object would not have been tracked correctly and therefore not be correctly classified with median voting classification. However, as object 5 is also visible and correctly tracked during frames 16-19 (highlighted with orange arrow), it can still be correctly classified.

The portion of correctly tracked objects in the tracking results is considered impressive and satisfactory to effectively use with median voting classification.

4.8.3 C.3 Additional trajectories

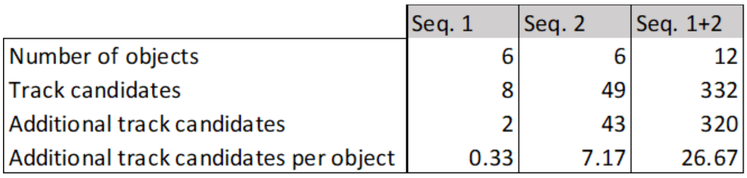
This section aims to answer sub-question C.3:

C.3: *"How many additional trajectories are detected besides the most correct trajectories of objects?"*

The presence of a false trajectory does not negatively affect the False Container Acceptance (FCA) rate as long as there is at least a single trajectory that correctly tracks each object for a significant length. However, additional false trajectories of bubbles increase the total number of bubble trajectories and are therefore problematic. Each additional trajectory containing bubbles has a small chance to contain a majority of outlier detections with positive classifications and can therefore cause a false rejection, increasing the False Container Rejection (FCR) rate. For additional trajectories of bubbles, it does not matter whether the trajectories are false or correspond to a real bubble trajectory. The effect of additional trajectories on the FCR rate can be seen in Figure 26 on page 65. For this reason, the portion of additional false trajectories should be limited.

Table 14 shows an extract of the numerical results from Table 13 with additional track candidates. A large difference can be seen in the number of additional track candidates between the different sequences. For sequence 1, only a few additional track candidates are detected. For both sequences combined, more than 26 additional track candidates are detected for each tracked object. Although some additional trajectories are correct trajectories of which the real trajectories have been split into sections, most additional trajectories are caused by the rapid growth phenomenon explained in subsection 4.7.3.

Table 14 Numerical results with additional track candidates



	Seq. 1	Seq. 2	Seq. 1+2
Number of objects	6	6	12
Track candidates	8	49	332
Additional track candidates	2	43	320
Additional track candidates per object	0.33	7.17	26.67

As shown in Table 11 on page 65, the results of median voting classification can be satisfactory, even with 332 track candidates as with both sequences combined. However, in a sequence with more objects in close proximity even more track candidates would be detected, possibly the results unacceptable. With a more accurate classifier the acceptable limit of additional trajectories will be higher.

It is possible that the effect of additional trajectories on the false container rejection rate is not as intense as described in subsection 3.8.1, as many additional trajectories contain mostly the same detections as others, while during simulation detections are randomly sampled separately for each trajectory. Future research is needed to confirm that the simulation in subsection 3.8.1 is accurate.

4.8.4 C.4 Trajectory length

This section aims to answer sub-question C.4:

C.4: *"What portion of the full trackable length is tracked for the most correct trajectories of objects?"*

Multiple figures in subsection 3.8.1 show the effect of trajectory length on the classification results with median voting classification. From this can be observed that FCR and FCA rate is significantly lower with longer trajectories. Therefore, ideally, every object should be tracked over its full trackable range. However, it should be noted that even with shorter object trajectories the FCR and FCA rates can be significantly improved compared to classifying single detections. This is as outliers have significantly less effect on the median voting classification result.

Figure 36 shows only the most correct and valuable track candidates from the tracking results. It can be observed that most of the correct track candidates are of close to the full trackable length. However not all objects are tracked in a single long track candidate. Object 3 in sequence 1 is an example of an object that is tracked in multiple shorter sections. This is as object 3 was not visible for some frames and the TDCP tracking algorithm can only track objects with singular missing detections. Object 2 in sequence 1 is also not tracked for its full trackable length. In Figure 32 can be seen that object 2 does not behave as expected as described in section 4.4. In the final two frames, when the object is close to the top of the container, the object does not spiral towards the centre but continues in a straight line.

Overall, the observed resulting trajectory length is considered satisfactory.

4.8.5 C. TDCP algorithm effectiveness and suitability

This section aims to answer the main research question for the tracking module:

C: *"Is the proposed TDCP algorithm effective and suitable to be implemented in a tracking module for the VPIS?"*

Of the sub-questions C.1 – C.4 that together answer this main research question, C.1 is answered unsatisfactory, C.1 and C.4 are answered with great satisfaction, and C.3 is answered with uncertain limited satisfaction. Based on these answers, it can be concluded that although the TDCP algorithm shows potential, it cannot be implemented in the VPIS in its current form. The TDCP algorithm fails the processing time criteria to be implemented in the VPIS without slowing down the inspection process. Additionally, number of additional track candidates is not ideal and detrimental to the resulting FCR rate from subsequent classification. Both issues are caused by a phenomenon referred to as the rapid growth phenomenon described in subsection 4.7.3 of the results. The proposed TDCP tracking algorithm shows potential in the quality of the tracking results as most objects are correctly tracked and correct trajectories are long.

4.9 Optimisation strategies

This section shortly discusses some possible optimisation strategies which could result in a version of the TDCP algorithm that is effective and suitable to be implemented into the VPIS. The first optimisations strategy is to simply optimise the processing time of TDCP by optimising calculations. The second and third optimalisation strategies sacrifice some thoroughness and classification accuracy to improve the processing time significantly.

4.9.1 Processing optimisation

The current version of TDCP has been partially optimised to reduce the number of possible combinations that need to be evaluated, which has already significantly reduced the processing time. It is expected that significant performance gains can be made by further optimisation of the code.

With the current version of TDCP, based on the position of an initial detection, it is calculated roughly in what range of positions a second detection can be that could fit in a plausible trajectory. Next, with each plausible second detection, it is calculated in what range a third detection can be. If there is a hit, these three detections form a tracklet. The process of tracklet formation could be significantly sped up by using a precomputed look-up tables for the motion model. By precomputing where a second and third detection can be on the horizontal range based on the first detection position, tracklet formation could be performed in a single fast operation.

The current version of TDCP has been fully implemented in python, implementing parts of the code in a compiled language such as C++ would significantly improve the speed of some repetitive steps during tracklet formation and track candidate formation.

Currently, the class instance structure of TDCP is not optimised for speed but convenience of development. Each track candidate instance includes a list of all contained tracklet instances. Each tracklet instance includes a list of all contained detection instances. Each detection instance contains all information from that detection, including the foreground image, segmentation mask and a total of handcrafted features. From the detection instances, only the unique identifier, position and frame number are required. By removing irrelevant information and making use of pointers, it is expected that the processing time can be lowered significantly.

It is hard to predict the resulting processing time of TDCP if the three processing optimisations described above are implemented. It is approximated that the resulting processing time will be 2-10 times lower than the current version of TDCP.

4.9.2 Rule relaxation

The rapid growth phenomenon is unavoidable if every possible track candidate should be classified separately. An approach that relaxes this rule could possibly prevent the rapid growth phenomenon.

Such a method could work as follows; if a situation is recognised where two objects are close together, detections of both objects could be treated as one detection. For these detections the detection with the highest classification score would be used for tracking. As a result, the number of tracklets, track candidates and processing time would be lower. But as the highest scoring detections are used, the resulting False Container Reject (FCR) rate would also be increased.

It is likely that there are other methods possible that include a form of rule relaxation. Further research is required into this topic.

4.9.3 Hybrid classification

Another method to increase the average processing time of TDCP such that it can be implemented in the VPIS is with a form of hybrid classification.

One method of hybrid classification would be to first perform classification on the individual detections and afterwards only perform tracking on a subset of detections that can possibly result in a positive classification. For example, trajectory of seven detections can only be positive if at least four out of seven detections achieve a classification score above a certain threshold. If four detections classify as such within a span of 12 frames to include missing detections, targeted TDCP tracking can be used to see if these detections can be a part of the same seven detection trajectory.

Another method of hybrid classification could include another multi-detection classification strategy such as multi-positive classification. If it is found through multi-positive classification that a number of positive detections indicate the presence of a particle, TDCP tracking could be used to validate that these detections can indeed be of the same object.

4.10 Comparison to SOTA

This section compares the proposed solution to the current state-of-the-art (SOTA) for this module, as described in subsection 4.2.1. It is found that the proposed solution is superior to the current SOTA.

The proposed method is fundamentally different compared to the SOTA method handcrafted feature bottom-up clustering [15]. Handcrafted feature bottom-up clustering recognises detections of moving objects based on object appearance with a number of handcrafted features. Although handcrafted features can be used for classification, it is expected that this method would not be effective to differentiate detections of different bubbles of a similar size. In contrast, the proposed TDCP algorithm performs tracking based only on the positions of detected objects without using object appearance. This is better as it is reliable for any type of particle. Another downside of the method is that bottom-up clustering can detect clusters with multiple detections per frame and can detect trajectories which are not continuous. This is not the case with TDCP.

If TDCP would be implemented with a Siamese similarity metric [35] as previously theorised [6], it is expected that this would result in a longer processing time, a higher FCA rate and an only slightly lower FCR rate. Additionally, this would require a large, annotated tracking dataset with labelled trajectories. It is expected that the downsides of implementing a Siamese similarity metric outweigh the benefit of a slightly lower FCR rate.

4.11 Conclusion

In this chapter, findings have been presented regarding the research and development of the proposed TDCP algorithm for the tracking module.

It has been observed that trajectories of real objects behave predictably and follow a spiralling pattern around the centre of the container. Based on this information, a motion model has been defined that describes the horizontal and vertical the motion of a moving object with a set of parameters limited to a feasible range. By checking if a trajectory through a subset of three detections can be described using this motion model, it is determined if the trajectory can possibly be of a real object or not. The proposed TDCP algorithm uses the motion model to detect short tracklets of three detections and merges these into larger track candidates that describe possible trajectories.

The proposed TDCP tracking algorithm shows potential in the quality of the tracking results as most objects are correctly tracked and correct trajectories are long. However, the current version of TDCP is not suitable to be implemented due to a phenomenon referred to as the rapid growth phenomenon explained in subsection 4.7.3. If an image sequence contains multiple objects in close proximity, the TDCP algorithm can detect a large number of possible tracklets and track candidates which results in an unacceptably long processing time. The larger number of track candidates detected in this situation has a negative effect on the effectiveness of median voting classification, which can be unacceptable if even more trajectories are detected in different image sequences.

A few optimisation strategies are shortly discussed. If the shortcomings of TDCP can be successfully solved with future work, TDCP could be implemented into the VPIS with great success.

4.12 Recommendations

The current version of the TDCP algorithm shows great potential but is not yet ready to be implemented in the VPIS. If the shortcomings of TDCP can be successfully solved with future work, TDCP could be implemented into the VPIS with great success. Therefore, it is recommended to continue with future research for TDCP in the following topics:

Unexpected object trajectory behaviour

Subsection 4.4.3 describes that unexpected trajectory behaviour has been observed in newer datasets. This new trajectory behaviour is problematic as it does not comply with the motion model developed in section 4.5 and therefore objects that show this behaviour cannot be tracked correctly. As explained in subsection 4.4.3, it is known that small changes to the spin-stop sequence and image capture sequence have an effect on the resulting trajectory behaviour. Future research is required to learn how the results of the first dataset can be reproduced or how the unexpected trajectory behaviour can be minimised.

Optimisation

The current version of TDCP cannot be implemented as processing time is too long as a result of the rapid growth phenomenon. In section 4.9, a number of optimisation strategies are shortly discussed that can reduce the processing time and make TDCP suitable to be implemented. It is recommended to implement one or more of these optimisation strategies.

Effect of additional trajectories

As explained in subsection 3.8.1, additional track candidates can increase the False Container Reject (FCR) rate as each detected track candidate with bubbles has the chance to be falsely classified as a particle. However, it is possible that the effect of additional trajectories on the false container rejection rate is not as intense as described in subsection 3.8.1, as many additional trajectories contain mostly the same detections as others, while during simulation detections are randomly sampled separately for each trajectory. Future research is needed to confirm that the simulation in subsection 3.8.1 is accurate.

5 Conclusion

Research and development has been performed into solutions for the three modules of the Visual Particle Inspection Subsystem (VPIS): background subtraction and segmentation module, classification module, and the tracking module. Research into each of the modules has been performed separately with separate research questions and conclusions. This section summarises the conclusion for each module.

Background subtraction and segmentation module

For the background subtraction and segmentation module, a solution has been proposed that performs *filtered temporal background modelling* and *locally adaptive threshold segmentation*. The proposed solution is shown to be effective through validation in four performance aspects: detection of low contrast objects, insensitivity to image disturbances, processing time, and parameter reconfiguration. Validation results show that the solution can accurately detect small objects with low contrast while detecting a minimal number of false detections. Processing time is below one second for two image sequences such that the solution can be implemented in the VPIS and can be further reduced through optimisation. The proposed solution has only a single parameter for segmentation sensitivity which does not need to be reconfigured for all tested imaging situations.

In addition to the base solution for the background subtraction and segmentation module, an add-on solution is proposed in Appendix A that makes it possible to detect particles near the rubber stopper of a syringe. The proposed solution including a version of the add-on solution has been adopted into usage by Luo Automation B.V.

Classification module

For the classification module the goal was not to develop a working classifier that is good enough to be implemented into the VPIS, but to develop a baseline classifier and perform exploratory research into the possibilities and pitfalls that come with this problem situation. Out of four candidate classification methods that have been tested, the best classification accuracy is achieved with a CNN. The baseline classifier achieves a classification accuracy of 0.930 and is used throughout the rest of this research. The following findings are made from exploratory research:

It is shown that handcrafted features can be used to predict the classification accuracy of a detection, this predicted classification accuracy can then be used to filter out small detections that are too hard to classify.

It is shown through simulation that if a single particle classification results in a container rejection, the resulting chance that a container is falsely rejected is extremely high. This is as a capture sequence of a container often results in many detections of bubbles and each bubble detection has a chance to be falsely classified as a particle.

Two multi-detection classification strategies are proposed that perform classification based on multiple detections. Median voting classification uses the results of the tracking module to classify objects using all detections of that object. Multi-positive classification does not use tracking but requires multiple detections to be classified positive. Of these classification strategies, median voting classification achieves the best results but both methods can significantly improve the classification performance.

Tracking module

The TDCP tracing algorithm is proposed as a solution for the tracking module. The TDCP algorithm is able to track small objects that move quickly through a medicine container based only on the positions of object detections.

It has been observed that trajectories of real objects behave predictably and follow a spiralling pattern around the centre of the container. Based on this information, a motion model has been defined that describes the horizontal and vertical the motion of a moving object with a set of parameters limited to a feasible range. By checking if a trajectory through a subset of three detections can be described using this motion model, it is determined if the trajectory can possibly be of a real object or not. The proposed TDCP algorithm uses the motion model to detect short tracklets of three detections and merges these into larger track candidates that describe possible trajectories.

The proposed TDCP tracking algorithm shows potential in the quality of the tracking results as most objects are correctly tracked and correct trajectories are long. However, the current version of TDCP is not suitable to be implemented due to a phenomenon referred to as the rapid growth phenomenon explained in subsection 4.7.3. If an image sequence contains multiple objects in close proximity, the TDCP algorithm can detect a large number of possible tracklets and track candidates which results in an unacceptably long processing time. The larger number of track candidates detected in this situation has a negative effect on the effectiveness of median voting classification, which can be unacceptable if even more trajectories are detected in different image sequences.

6 References

[1] L. Doessegger, H. C. Mahler, P. Szczesny, H. Rockstroh, G. Kallmeyer, A. Langenkamp, J. Herrmann and J. Famulare, “The Potential Clinical Relevance of Visible Particles in Parenteral Drugs,” *Journal of Pharmaceutical Sciences,* vol. 101, p. 2635–2644, August 2012.

[2] J. A. Melchore and D. Berdovich, *Considerations for design and use of container challenge sets for qualification and validation of visible particulate inspection,* vol. 66, 2012, pp. 273-284.

[3] J. H. Bowen, B. H. Woodard, T. K. Barton, P. Ingram and J. D. Shelburne, “Infantile Pulmonary Hypertension Associated with Foreign Body Vasculitis,” *American Journal of Clinical Pathology,* vol. 75, p. 609–614, April 1981.

[4] P. Dewan, H. H, G., D. Middleton and J. Terlet, “Plastic particle migration during intravenous infusion assisted by a peristaltic finger pump in an animal model,” *Pediatric Surgery International,* vol. 18, p. 310–314, July 2002.

[5] *Seidenader VI-S.* [Film]. Germany: Körber Pharma, 2022.

[6] S. v. Eeden, “Literature study, Methods for detection and classification of,” 2022.

[7] J. Lu, Y. N. Wang, J. Zhang and B. W. Zhou, “On-line detection of foreign substances in glass bottles filled with transfusion solution through computer vision,” *Proceedings of the 2008 IEEE International Conference on Information and Automation, ICIA 2008,* p. 424–429, 2008.

[8] J. Ge, Y. N. Wang, B. W. Zhou and H. Zhang, “Intelligent foreign particle inspection machine for injection liquid examination based on modified pulse-coupled neural networks,” *Sensors,* vol. 9, p. 3386–3404, April 2009.

[9] J. Chen, Y. Wang, C. Wu, H. Zhang, J. Ge, B. Zhou and J. Mao, “Research on real-time vibration-insensitive inspection and classification algorithms for automatic online vision-based inspector,” *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering, TMEE 2011,* p. 1737–1742, 2011.

[10] Y. Wang, J. Ge, H. Zhang and B. Zhou, “Intelligent injection liquid particle inspection machine based on two-dimensional Tsallis Entropy with modified pulse-coupled neural networks,” *Engineering Applications of Artificial Intelligence,* vol. 24, p. 625–637, June 2011.

[11] J. Fang, Y. Wang and C. Wu, Binocular automatic particle inspection machine for bottled medical liquid examination, IEEE, 2013, p. 397–402.

[12] H. Zhang, X. Li, H. Zhong, Y. Yang, Q. M. J. Wu, J. Ge and Y. Wang, “Automated machine vision system for liquid particle inspection of pharmaceutical injection,” *IEEE Transactions on Instrumentation and Measurement,* vol. 67, p. 1278–1297, June 2018.

[13] Z. Bowen, W. Yaonan, G. Ji and Z. Hui, “A machine vision intelligent inspector for injection,” in *Proceedings - 2008 Pacific-Asia Workshop on Computational Intelligence and Industrial Application, PACIIA 2008*, 2008.

[14] Y. Qin and B. Wang, “Study of on-line inspection technique for foreign substance in ampoule,” *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce, AIMSEC 2011 - Proceedings,* p. 4342–4345, 2011.

[15] G. Lu, Y. Zhou, Y. Yu and S. Du, “A novel approach for foreign substances detection in injection using clustering and frame difference,” *Sensors,* vol. 11, p. 9121–9135, October 2011.

[16] F. Zhou, Z. Su, X. Chai and L. Chen, “Detection of foreign matter in transfusion solution based on gaussian background modeling and an optimized BP neural network,” *Sensors (Switzerland),* vol. 14, p. 19945–19962, October 2014.

[17] M. Feng, Y. Wang and C. Wu, “Foreign particle inspection for infusion fluids via robust dictionary learning,” in *Proceedings - The 2015 10th International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2015*, 2016.

[18] H. J. Liu, “Development of intelligent liquid inspection system based on machine vision,” *Lecture Notes in Electrical Engineering,* vol. 176 LNEE, p. 209–215, 2012.

[19] J. Zhang, Z. Lu, Z. Wang, Q. Yuan and G. Wang, “Research on intelligent inspection machine based on linear CCD,” in *Advanced Materials Research*, 2012.

[20] S. Wang, Q. Zhuo and J. Xia, “Detection of Glass Chips in Liquid Injection Based on Computer Vision,” *Proceedings - 2015 International Conference on Computational Intelligence and Communication Networks, CICN 2015,* p. 329–331, August 2016.

[21] M. Zhao, H. Zhang, L. Liu, Z. Liang and G. Deng, “Joint deep learning and clustering algorithm for liquid particle detection of pharmaceutical injection,” in *2018 8th International Conference on Image Processing Theory, Tools and Applications, IPTA 2018 - Proceedings*, 2019.

[22] Q. Wang, H. Zhang, L. Liu, H. Zhong, Y. Wang and Q. M. J. Wu, “The Automatic Liquid Particles Detection for Injection,” in *Proceedings - 2020 Chinese Automation Congress, CAC 2020*, 2020.

[23] H. Zhang, M. Zhao, L. Liu, H. Zhong, Z. Liang, Y. Yang, X. Zhou, Q. M. J. Wu and Y. Wang, “Deep Multimodel Cascade Method Based on CNN and Random Forest for Pharmaceutical Particle Detection,” *IEEE Transactions on Instrumentation and Measurement,* vol. 69, p. 7028–7042, September 2020.

[24] Dar-Shyang Lee, “Effective Gaussian mixture learning for video background subtraction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, p. 827–832, May 2005.

[25] S. Jiedi and W. Jiangtao, “Recognition of impurity in ampoules based on wavelet packet decomposition energy distribution and SVM,” *Proceedings - 2009 2nd IEEE International*

Conference on Computer Science and Information Technology, ICCSIT 2009, p. 162–166, 2009.

[26] J. Ge, S. Xie, Y. Wang, J. Liu, H. Zhang, B. Zhou, F. Weng, C. Ru, C. Zhou, M. Tan and Y. Sun, “A System for Automated Detection of Ampoule Injection Impurities,” *IEEE Transactions on Automation Science and Engineering,* vol. 14, p. 1119–1128, April 2017.

[27] G. Palmer, B. Schnieders, R. Savani, K. Tuyls, J. Fossel and H. Flore, “The automated inspection of opaque liquid vaccines,” 2020.

[28] M. Seeland and P. Mäder, “Multi-view classification with convolutional neural networks,” *PLOS ONE,* vol. 16, p. e0245230, January 2021.

[29] L. P. Wang and C. R. Wan, “Comments on "The extreme learning machine",” *IEEE Transactions on Neural Networks,* vol. 19, p. 1494–1495, 2008.

[30] I. Kononenko, “Estimating attributes: Analysis and extensions of RELIEF,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 784 LNCS, p. 171–182, 1994.

[31] J. A. Melchore, *Sound practices for consistent human visual inspection,* vol. 12, 2011, pp. 215-221.

[32] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola and R. C. Williamson, “Estimating the Support of a High-Dimensional Distribution,” *Neural Computation,* vol. 13, no. 7, pp. 1443 - 1471, 2001.

[33] G. D. Luca, *SVM Vs Neural Network | Baeldung on Computer Science,* 2022.

[34] Y. Lecun, Y. Bengio and R. 4g332, “Convolutional Networks for Images, Speech, and Time-Series”.

[35] M. Ondrasovic and P. Tarabek, “Siamese Visual Object Tracking: A Survey,” *IEEE Access,* vol. 9, p. 110149–110172, 2021.

[36] L. Chen, F. Rottensteiner and C. Heipke, “Invariant Descriptor Learning using a Siamese Convolutional Neural Network,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Vols. III-3, p. 11–18, June 2016.

[37] Ming-Kuei Hu, “Visual pattern recognition by moment invariants,” *IEEE Transactions on Information Theory,* vol. 8, p. 179–187, February 1962.

[38] Api First, “Lighting requirement for Manual Visual Inspection of Pharmaceuticals,” Api First, 13 October 2020. [Online]. Available: https://www.apifirst.in/2020/10/13/lighting-requirement-for-manual-visual-inspection-of-pharmaceuticals/. [Accessed 1 June 2023].

Appendix A: Detection near rubber stopper

One of the shortcomings of the proposed solution for the background subtraction and segmentation module is that the method can only detect objects in the centre region of a container. This appendix shortly explains why other regions are challenging and presents an add-on solution to the background subtraction and segmentation module that makes it possible to detect moving objects near the bottom region of a syringe with deforming static bubbles.

Challenging regions

It is difficult to perform inspection in the entire region of a container image. This is as the top and bottom regions of a container offer challenging situations depending on the type of container. In the top region of a container, the moving liquid surface, large floating bubbles, and reflections on the liquid surface make it difficult to detect moving objects. Near the bottom of a container, other challenging situations occur such as reflections or bubbles that are stuck to a stopper. Figure 37 shows challenging regions for 30ml vials, 10ml vials and 5ml syringes.

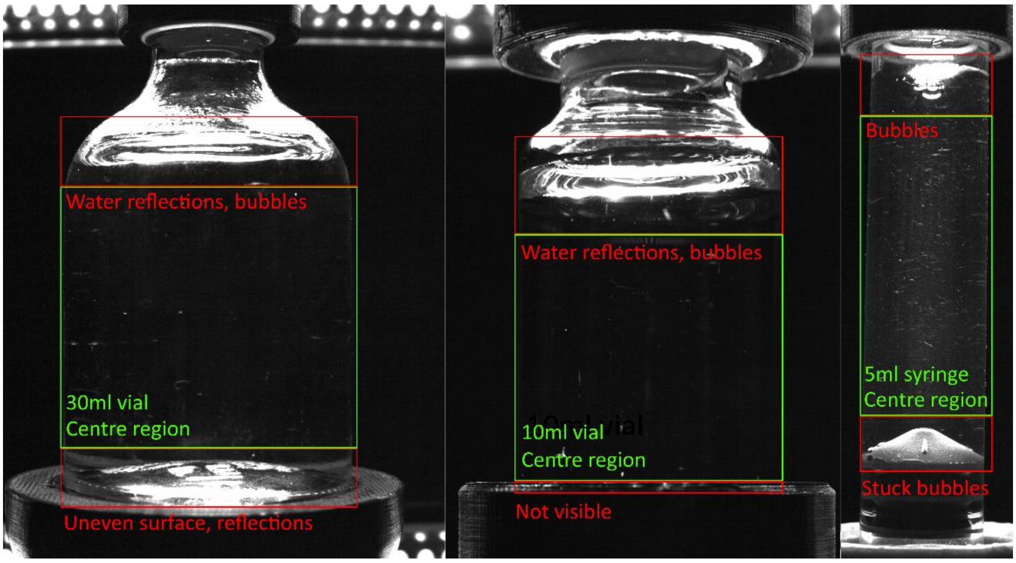


Figure 37 Challenging regions with 30ml vials, 10ml vials, and 5ml syringes. Each container type presents different challenges.

Existing solutions

Most solutions that are commercially available for large scale automatic visual inspection detect moving objects only in the centre region of a container. For some systems, optional upgrades are available in the form of additional sensors or specialised image capture sequences that enable detection of objects in challenging regions.

Minimal requirement

It is possible for automatic inspection systems to pass certification requirements while only detecting objects in the centre region of a container as most moving objects pass through this region. By performing inspection in more of the challenging regions, more particles can be detected and safety of injections can be improved.

Add-on solution

The goal is to develop an add-on solution to be able to detect moving objects in the bottom region of a 5ml syringe without detecting deforming bubbles. It was chosen to develop an add-on solution for this specific region as it is relatively common for moving objects to occur in this region. Additionally, the add-on solution developed for 5ml syringes can be adapted for different syringe sizes and types.

5ml Syringe bottom region challenge

Figure 38 shows the bottom region of a syringe where bubbles are present on the rubber stopper. Bubbles that are stuck to the stopper do not release during a regular spin-stop sequence but deform between frames as a result of the moving liquid. The right side of Figure 38 shows the absolute difference between subsequent frames, the visible region is caused by a deforming static bubble and would likely be falsely detected as a moving object by the solution proposed in section 2.4. The goal of the add-on solution in this challenging region is to determine whether a detection in this region is caused by a deforming static bubble or a real moving object.

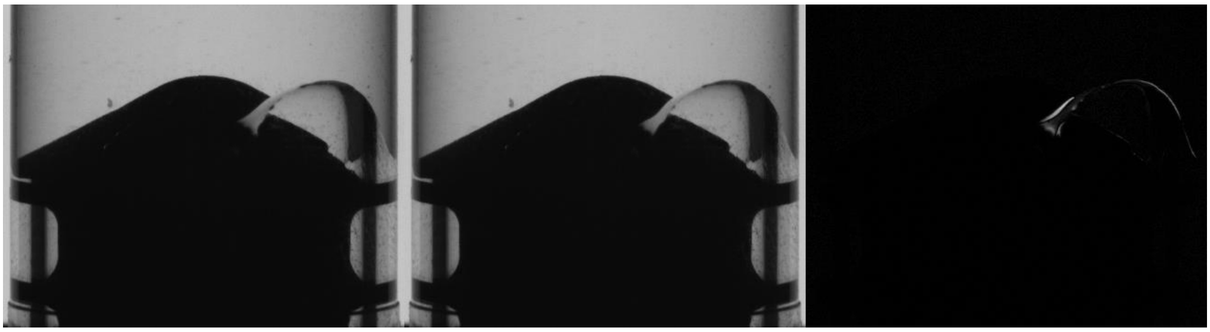


Figure 38 Left and Centre: Cropped section of a syringe with a bubble on the rubber stopper on two subsequent frames. Right: Absolute difference between the two frames.

Observations

The following observations are made to differentiate between deforming static bubbles and moving objects.

- Areas in which a static bubble is deforming can be recognised by a significant increase of the background standard deviation BG_{std}, as long as the deformation occurs for more than two frames.
- Between frames, the edges of a deforming static bubble move a limited distance in pixels.
- Between frames, the intensity of a deforming static bubble changes with an intensity.

Static bubble region

First, regions are selected where the background standard deviation BG_{std}.is above a certain threshold. Next, these regions are dilated to account for movement that occurs in less than three frames, this is shown in Figure 39. Outside the resulting static bubble region, moving objects can be detected as normal. Inside the resulting static bubble region, a new detection method is used based on a local minimum and local maximum threshold.

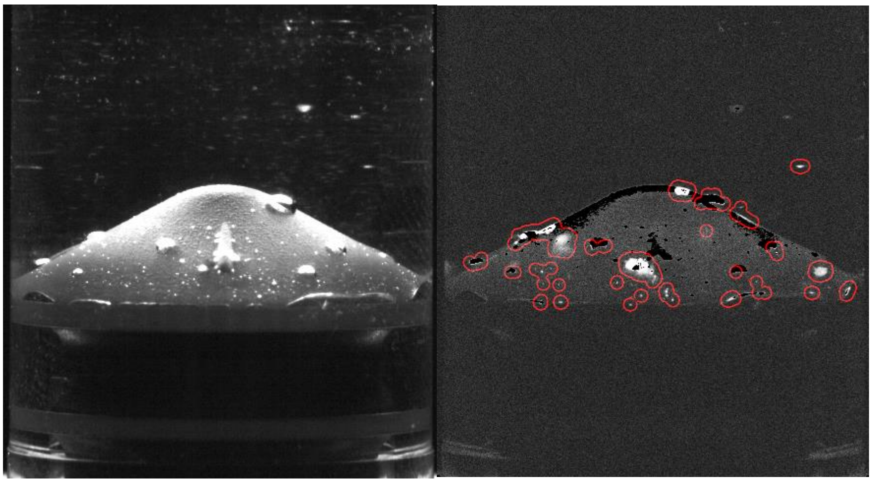


Figure 39 Left: The bottom region of a syringe with a rubber stopper and deforming static bubbles. Right: Background standard deviation BG_{std} with static bubble regions outlined in red.

Local minimum and local maximum bubble intensity

It has been observed that edges of static deforming bubbles move only a limited distance in pixels between frames and the intensity of a bubble changes between frames with a limited intensity difference. These observations will be used to detect moving objects in static bubble regions.

Based on the previous image in the image sequence, a local minimum threshold image and local maximum threshold image are computed as shown in Figure 40. The local minimum threshold image is darkened by 10 points and the local maximum is brightened by 10 points to account for the limited change in pixel values. Within the static bubble region shown in Figure 39, if a pixel intensity value is between the of the local minimum and local maximum of the previous frame, it is assumed that this pixel contains a static bubble. If the pixel intensity value is outside the local minimum and local maximum of the previous frame, this is detected as a moving object.

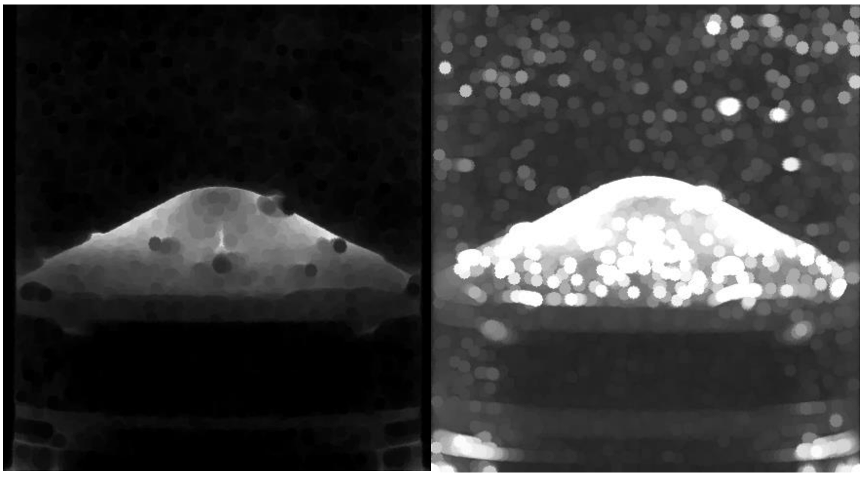


Figure 40 Left: Local minimum threshold image. Right: Local maximum threshold image. Intensity values within this range are assumed to be of deforming bubbles.

Results

Figure 41 shows a result of the add-on solution. In the figure can be seen that two moving objects are detected. The moving object near the top of the image is located outside of the static bubble regions and is therefore detected using the regular solution proposed in section 2.4. The other moving object is located in a static bubble region and is detected by the add-on solution as the object intensity is below the local minimum.

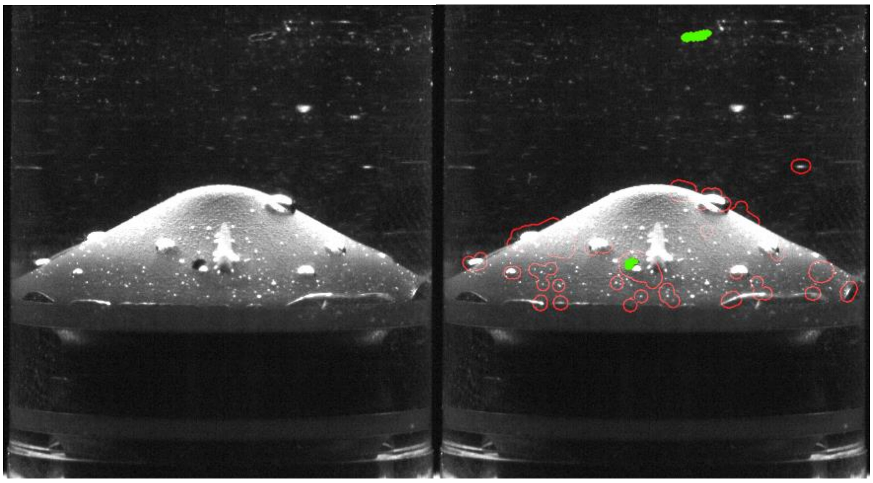


Figure 41 Left: image containing two dark moving objects. Right: The same image where static bubble regions are outlined in red and detected moving objects are outlined in green. The bottom object is detected using the local minimum and local maximum threshold.

Shortcomings

Although the add-on solution is currently functional, it is not very reliable in its current form and can be further improved.

A shortcoming with the current add-on solution is that *filtered temporal background modelling* proposed in subsection 2.4.1 does not work correctly in front of the rubber stopper. With filtered temporal background modelling, it is assumed that moving objects are always brighter than the background with diagonal lighting. In front of the illuminated stopper, moving objects can be either brighter or darker than the background. As a result of this, dark moving objects are included in the background cluster, resulting in an increased background standard deviation and inaccurate static bubble regions. Although some tests have been performed where both low and high values are excluded from the background cluster, further optimisation of the add-on solution is outside of the scope of this research.

Conclusion

An add-on solution has been developed for the solution proposed in section 2.4. With the add-on solution it is possible to detect moving objects in the bottom region of a syringe near the rubber stopper without detecting deforming bubbles. As the current add-on solution has shortcomings, it is recommended to improve invest future work into improving the method. The add-on solution shows enough potential that Luo Automation B.V. has adopted a version of it with the background subtraction and segmentation module.

Appendix B: Handcrafted features

The list below contains an overview of all handcrafted features that have been used in this thesis during testing. The handcrafted features 18 – 40 are contour image moments.

1. X position
2. Y position
3. Width
4. Height
5. Aspect ratio L/W
6. Aspect ratio smallest/largest
7. Area
8. Solidity
9. Ellipse mayor axis
10. Ellipse minor axis
11. Ellipse angle
12. Ellipse eccentricity
13. Ellipse verticality
14. Total contrast
15. Mean contrast
16. Perimeter
17. Circularity
18. M_{10}
19. M_{01}
20. M_{20}
21. M_{11}
22. M_{02}
23. M_{30}
24. M_{21}
25. M_{12}
26. M_{03}
27. Mu_{20}
28. Mu_{11}
29. Mu_{02}
30. Mu_{30}
31. Mu_{21}
32. Mu_{12}
33. Mu_{03}
34. Nu_{20}
35. Nu_{11}
36. Nu_{02}
37. Nu_{30}
38. Nu_{21}
39. Nu_{12}
40. Nu_{03}