Mining Business Processes from Real World Activities using Tobii Glasses

Katsikiotis Christos

Kongens Lyngby 2018
MSc-2018

Technical University of Denmark
DTU Compute
Matematiktorvet, building 303B, DK-2800 Kongens Lyngby, Denmark
Phone +45 4525 3031, Fax +45 4588 1399
compute@compute.dtu.dk
www.compute.dtu.dk MSc-2018

Abstract

Many organizations and businesses are struggling to improve the automation among its users and the machines failing to improve the efficiency. The aim of this thesis is to provide a solution to low efficiency and increase the cooperation between the human and the machine by recording the tasks he is executing and eventually using process mining techniques to generate a process model. To pursue this aim a research was conducted on how users' activity detection could be achieved pointing to the use of eye tracking solutions alongside with optical identification technologies that could lead to object identification and finally activity identification. Before the implementation of the proposal, an extended comparison among the technologies that could lead to object identification was conducted leading to the use of Barcode as the prominent object identification technology while also a strong emphasis was put on the adjustability of the algorithm. More specifically the algorithm aims to adjust to the user and the environment to achieve high accuracy rate and filter noise. Due to the nature of the proposal and the environment which is aimed to be deployed, there are some requirements that must be fulfilled.

The first requirement is the proposal to achieve a high degree of automation while requiring small or none human interaction to decrease the execution speed and limit the margin of human error. A further requirement that is needed to be fulfilled is the approach to be capable of tracking down user's activities without obstructing him from his tasks. It is vital for the user not to feel impeded by the approach so his efficiency remains at least at the same level. Finally, the approach should be able to export the activity sequence so the process model can be generated. The accuracy rate that was achieved through this approach was encouraging and even though there is a need for some improvements, it pointed to the big potential to be deployed in real organizations.

Preface

This thesis was carried out at the department of DTU Compute, at the Technical University of Denmark, in fulfillment of the requirements for acquiring an M.Sc. in Computer Science and Engineering.

The goal of this thesis was to provide an approach on how the activities of a user can be tracked using eye tracking equipment with object identification technologies, be mapped to activities and to generate a process model.

This document describes the project itself and discusses the concepts involved. It discusses in details the available technologies for each part and makes a comparison before concluding which covers most of the requirements. In the end it presents the results and more specifically the accuracy rate and the execution speed of the whole approach.

Acknowledgements

My deepest gratitude goes to my supervisor, Barbara Weber. During the last semester, the freedom and the guidance she offered me was invaluable and crucial to complete this thesis. Barbara, with her questions and her suggestions, pointed me the correct direction on how to define the problem and find the optimal solution in a field which I had limited experience before.

I would also to kindly thanks my co-supervisor, Andrea Burattin for his effort and the time he devoted to helping me. His way of thinking and the suggestions he provided, enabled me to pursue this approach from a wider perspective and increase the quality of the thesis.

Moreover, I would like to thank my family from the bottom of my heart, with their help the pursuit of an MSc and this thesis wouldn't be possible. Their patience and the support were crucial all these months.

Contents

Abstract **i**

Preface **iii**

Acknowledgements **v**

1 Introduction **1**
1.1 Motivation 1
1.2 Problem Statement 2
1.3 Goals 2
1.4 Thesis Outline 3

2 Background **5**
2.1 Eye Tracking 5
2.2 Process Mining 6
2.3 BPMN 7

3 Requirements & Goals **11**

4 Approach **13**
4.1 Overview 13
4.2 Cooking Pasta and Sausages Example 16

5 Pre-Processing Phase **19**
5.1 Preparing and Optimizing sub-phase 21
5.1.1 Introduction - Input 22
5.1.2 Procedure 22
5.1.3 Experimentation 30
5.1.4 Output 35

5.2 Processing sub-phase 35
5.2.1 Introduction - Input 35
5.2.2 Requirements & Goals 36
5.2.3 Analysis 36
5.2.4 Experimentation 39
5.2.5 Comparison 49
5.2.6 Implementation 51
5.2.7 Output 51
5.3 Filtering sub-phase 51
5.3.1 Introduction - Input 51
5.3.2 Procedure 52
5.3.3 Experimentation 52
5.3.4 Implementation 54
5.3.5 Output 54

6 Mapping Phase 55
6.1 Input 56
6.2 Analysis 56
6.3 Experimentation 60
6.4 Requirements 62
6.5 Evaluation - Comparison 62
6.6 Implementation 64
6.7 Output 64

7 Mining Phase 65
7.1 Input - Introduction 66
7.2 Process Discovery sub-phase 67
7.3 Experimentation 67
7.4 Conformance Check sub-phase 72
7.5 Output 75

8 Evaluation 77

9 Generalizability of Approach 81
9.1 Pre-Processing Phase 81
9.2 Mapping Phase 83
9.3 Mining Phase 83
9.4 Conclusion 84

10 Discussion 85

11 Related Work 87

12 Conclusion **91**
12.1 Analysis of the proposal 91
12.2 Limitations 92
12.3 Future Work 92

Bibliography **93**

Chapter 1

Introduction

1.1 Motivation

In the last century, more and more companies and businesses started to incorporate machines and automation into the production line and devoting the human resources to other aspects of businesses. This trend allowed the companies to achieve higher efficiency, lower time to produce with fewer deviations and faulty products.

This trend continued until recently when another issue has risen. A big number of businesses are struggling to introduce even more automation and take full advantage of the new technologies which leads to lower efficiency. Failing to maximize the efficiency increases the costs, decreases the quality of the product and ultimately leads into customers' dissatisfaction.

One way to combat low efficiency is to introduce even higher automation to aid workers' to complete their tasks in a shorter time with fewer mistakes. Research has emphasized the significant results of using automation and computers in improving the business efficiency [Bes16] [SWB97] [Gra95]. Another way to bring the efficiency to the maximum level is to track down the processes not only in organization level but on employee level as well as to detect any possible point in the process that can be improved.

This thesis is focused on the tracking the process of each worker by deploying Tobii Glasses, an eye tracking device that is able to record the view of the user alongside what he is observing at each moment. Following, with the combination of the observation data and Barcode tags, an optical detection technology it is possible to detect the objects the user is operating at each moment. After filtering unwanted detected objects, they are mapped to activities and finally the process model is generated. Using the process model, the business executives can extract useful information and increase the efficiency.

The results have shown a high accuracy rate of this approach, up to 80% percentage. This gives an promising evidence of using eye tracking equipment alongside optical detection technologies to perform business mining that could potentially draw a significant impact on the growth of organizations.

1.2 Problem Statement

This thesis aims to provide a conceptual design alongside with a real-life example to increase the cooperation of the machine and the worker to increase both the efficiency and the automation within an organization while reducing the costs. The efficiency is not only increased in employee-level but on organization level as well. More technically, a solution is presented to create an event log from a user's executed tasks and finally to generate a process model. Moreover, the outcome of this approach can be used for further analysis outside the scope of process mining.

1.3 Goals

The main goal of this thesis is to propose a solution that tracks down a user's activities while he is executing a process. The proposal should have a high accuracy rate and be able to create a process model for further analysis. In order for this to be achieved the propose should be able to:

- Keep the human interaction at a minimal level
- Adjust to many different settings
- Optimize for each user
- Identify objects with many variations at its shape

- Map objects to activities

1.4 Thesis Outline

Chapter 2: Background
The theory behind the approach will be presented in this chapter.

Chapter 3: Requirements & Goals
This chapter sets the requirements and the goals the solution must meet.

Chapter 4: Approach
In this chapter, a small overview of the approach is discussed while it presents an example process to be implemented throughout the thesis.

Chapter 5: Pre-Processing Phase
This is the part of the approach that identifies the objects the user operates processing the video exported from the Tobii Glasses. In addition, a filtering of the accidentally observed objects is executed.

Chapter 6: Processing Phase
In this chapter the mapping of the objects into activities is explained.

Chapter 7: Mining Phase
In the last part of the approach, the model is created from the sequence of activities. Moreover, a conformance check is presented and analyzed processing the created model and another activity sequence.

Chapter 8: Evaluation
In this chapter, a number of technical aspects of the approach are evaluated and discussed.

Chapter 9: Generalizability of Approach
In this chapter, the adjustability of the approach into different environments and users is discussed.

Chapter 10: Discussion
The degree of how much the requirements are met by the proposal is analyzed in this chapter.

Chapter 11: Related Work
All related research work is discussed into this chapter. Moreover, a short com-

parison with this proposal is presented.

Chapter 12: Conclusion
This chapter closes the document with a short overview of the approach, the limitations and what is the future work.

Chapter 2

Background

Before the analysis of the proposed solution, it is crucial to discuss the theory that this approach is based on to avoid any possible confusions and misunderstandings from the side of the reader.

2.1 Eye Tracking

Glasses and eye tracking play a vital role in this thesis. But before explaining the movements of the eyes, fixations must be defined. According to Kenneth Holmqvist in the chapter 2.5.1 of [HNA+11], fixations are the state of the eye that remains almost still for a period of time and can last from some tens of milliseconds up to several seconds. While the eye seems almost still during a fixation, there are still 3 types of some small movements that still occur:

- Tremors: small movements around 90Hz and the reason is still unknown
- Drifts: take the eye away from the fixation
- Microsaccades: bring the eye back to fixation

The movements of the eyes that occur between fixations are called saccades and they are the fastest movement of the human body. Most of the saccades do not stop at the intended target and the eye still wobbles before it is completely still. This post saccadic movement is called glissade.

On many occasions, throughout this document gazing points will be referred. During the execution of the process, at each moment, the focus of the user is tracked and recorded. This information is called gazing points.

At the table 2.1 there is a small overview of the types of eye movements.

Type of movement	Duration (ms)	Velocity (/s)
Fixation	200-300	-
Saccade	30-80	30-500 °
Glissade	10-40	20-140 °
Microsaccade	10-30	15-50 °
Tremor	-	20'
Drift	200-1000	6-25'

Table 2.1: Types of eye movements

2.2 Process Mining

The term Process mining is used in multiple times in this document so it is crucial to be defined clearly. The principle of process mining is to use event logs to extract valuable information about the process, for example, to identify bottlenecks, anticipate problems, record policy violations and recommend countermeasures and stream the processes [vdAAdM+12]. Without process mining, could lead to inefficiencies in the process model resulting in high costs to the organization. More analytically there are three types of process mining:

- **Discovery**: it takes as an input only an event log to discover the executed activities and to create a model.
- **Conformance checking**: it takes as an input a model and an event log to check if the latter conform to the former.
- **Enhancement**: it receives as an input a model and the event log and its purpose is to improve the model using the data from the event log.

The created model can be either a Petri net, BPMN, EPC or UML activity diagram.

2.3 BPMN

Throughout this thesis, the business processes are presented using the Business Process Model and Notation (BPMN) 2.0. It was created by Object Management Group as a solution to bridge the gap between business analysts and technical developers to provide an easily understandable notation according to the specifications in [OMG11]. The purpose of BPMN is to visualize the flow of execution of a process. In other words, it explains the steps of a process among the actors. In the simplest version, a BPMN model consists of nodes and lines that connect them. The lines are one-directional and point which node is going to be activated in the next step. On the other hand, nodes could either represent an activity or an operator.

XOR is one of the BPMN operators and it is divided into XOR-split and XOR-join. Both of them are represented by a diamond marked with an "X" as shown to the fig. 2.1. When an XOR-split is detected, splits the execution flow into two or more branches and only one of them can be activated. When these executions paths are finished are merged back into one using an XOR-join.

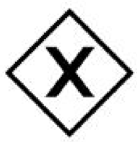


Figure 2.1: XOR gateway

AND is another important BPMN operator and it denotes parallelism. It is represented by a diamond marked with a "+" as it can be seen in the fig. 2.2. AND operator is similar to XOR-gateway as it consists of AND-split and AND-join. With AND-split the execution path is divided into two or more branches and all of them can be activated with no particular order. When all the nodes from all the paths are finished, they are merged with an AND-join.

Figure 2.2: AND gateway

OR is a combination of XOR and AND operators. It is represented by a diamond marked with an "O" as shown to the fig. 2.3. It divides the execution path into two or more branches and allows one or more to be activated. Similarly, it is divided into OR-split and OR-merge.

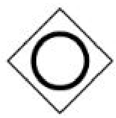


Figure 2.3: OR gateway

As previously reported, the BPMN is the visualization of the execution flow among the actors. BPMN allows defining the entities that participate in the process, for example, a bank and a custom, while allows to divide them into smaller entities like the technical support of the bank and the cashier. BPMN uses pools to define actors, like big organizations and lanes to define actors that execute a smaller and more specific set of tasks. For example in the fig. 2.4 the Bank and the Customer are represented by a pool and smaller units like an employee or the sales department within the bank are represented by lanes.

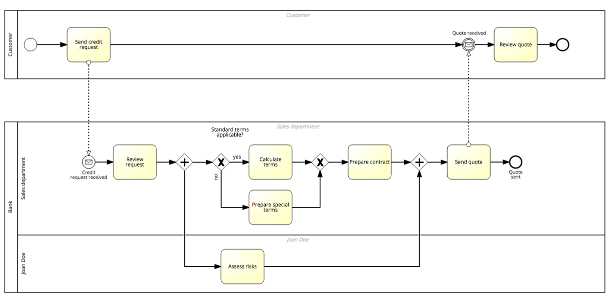


Figure 2.4: BPMN pools and lanes

BPMN also features the use of subprocesses. In some occasions, it is desired to group a sequence of activities that have the same purpose. For example activities like "go to the shop", "find the cheapest milk", "go to the cashier" and "pay" can be grouped into one subprocess with the title "buy milk". In other words, subprocesses allow simplifying the BPMN diagrams. An example of a subprocess can be seen in the fig. 2.5.

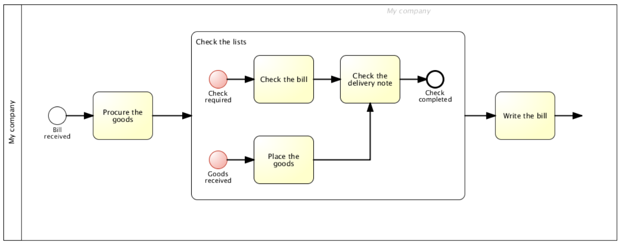


Figure 2.5: BPMN Subprocess

In many processes, there are tasks that are activated after a certain event or after a specific time period. In BPMN there are 3 types of event, the Start, the Intermediate and the End events. "Start events" are considered the ones that occur at the beginning of the process, "end events" those that occur in the end and finally, the "intermediate events" that occur in the middle of the process. Every process model has one start event and at least one finish event. An example of a start event and a finish event can be in the fig. 2.6 and fig. 2.7 respectively.

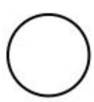


Figure 2.6: Start event

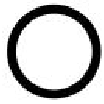


Figure 2.7: Finish event

Both start and intermediate events are divided into two categories, the Interrupting, and the non-interrupting events as can be seen in the fig. 2.8. The main difference between these two types of events is that the interrupting events stop the execution of the activity while the non-interrupting events will create a parallel flow that executes the Timeout process. The interrupting events are represented by a timer inside a single or double thin circle while the non-interrupting events with a single or double-dashed circle.

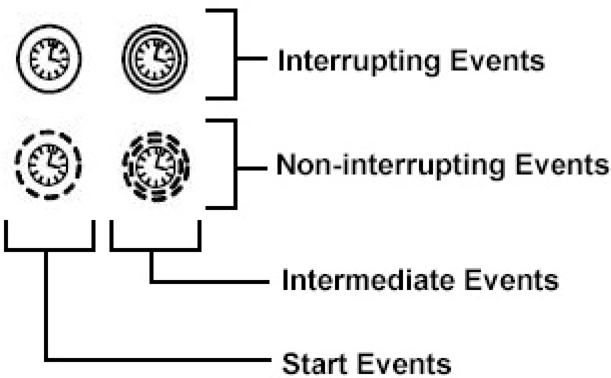


Figure 2.8: Categories of start and intermediate events in BPMN

Alongside the aforementioned operators like XOR, AND, OR there are operators that control the executions flow depending on the time events but due to their complexity and their extent, they are out of the scope of this section and won't be listed.

Chapter 3

Requirements & Goals

The proposal that aims to solve the problem mentioned in the chapter 1.2, must follow a list of requirements. These requirements are listed in the table 3.1.

Requirements
Automated procedure
Task identification
Support of various environments
Low user invasivity
Object detection position shape agnostic
Support all mapping types
Process model generation

Table 3.1: Requirements set for the approach

More analytically, the proposal should aim to increase the cooperation between the machine and the worker to increase the efficiency and the automation by identifying the worker executed activities to create the process model. Primarily, it is vital for the proposal to be an automated process capable of task detection to produce the process model.

Secondary but still important is the proposal must be able to support multiple environments without obstructing the user from his tasks and decrease his efficiency. In addition, it is important also to support as many as possible objects, to not decrease the accuracy rate of the approach while it should have to support all the mapping types to support as many processes as possible.

After the requirements the proposal is needed to meet, the goals it should try to achieve must be listed. An overview of the requirements are shown in the table 3.2.

Goals
Low cost
High Accuracy
Low preparation time
Small or no human interaction

Table 3.2: Goals of the approach

The most crucial target of the proposal should aim to achieve is to bring down the costs. In order, this to be achieved the accuracy is important to remain high and require small or no human interaction. Finally, less but still important is the preparation time of the approach to be kept as low as possible.

Chapter 4

Approach

This chapter gives a short introduction to the approach and analyzes the example process which is used during this thesis.

4.1 Overview

The main objective of this approach is to detect and mine the process that a person executes using the data fed from Tobii Glasses. This to be achieved, the proposal should be capable to detect which activity is executed at any moment and track it down.

Steven P. Tipper reported that vision is crucial when the user intends to interact with an object as it provides the required information on how to perform actions with this particular object. [Tip10] This approach takes into consideration Tipper's findings to detect user's activities by identifying the objects the user is observing at each moment, consequently this sets as a requirement the user to wear an eye tracking device. In this approach, the user is suggested to wear Tobii Glasses 2 during the execution of the tasks. Tobii Glasses 2 is a device similar to regular glasses that can record both the optical field of the user and his gazing points and can be seen in the fig. 4.1.

Figure 4.1: Tobii Glasses 2

This approach is divided into 3 sections.

The first section, which is discussed in the section 5, gives a brief overview of the Pre-Processing Phase and it is responsible to prepare the given data for the object identification and finally to omit any unwanted objects(noise).

During the second section, located in the section 6 of this thesis, the Mapping Phase, the observed objects are mapped into activities.

In the third and last section, which is located in section 7, the Mining Phase is analyzed. In this phase, the process model is created using process mining and optionally a conformance check is executed with it and a received action sequence.

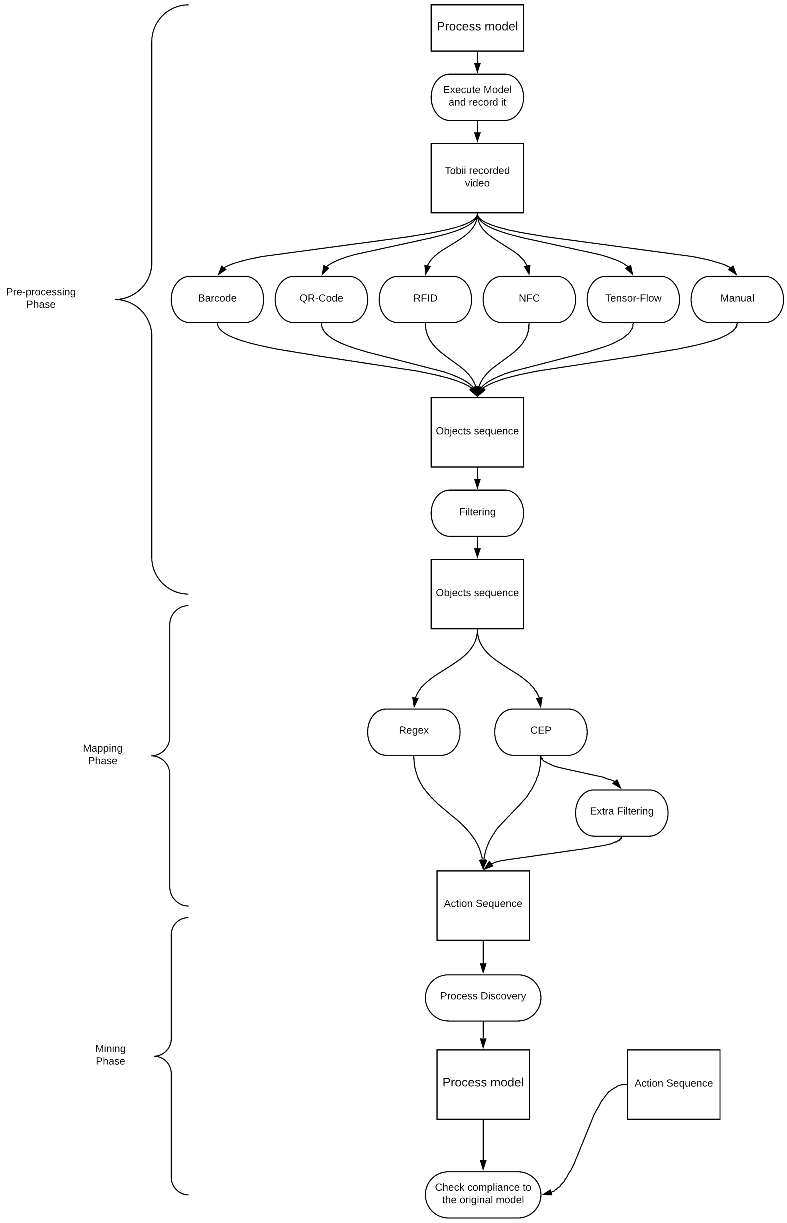


Figure 4.2: Structure of the proposed Approach

4.2 Cooking Pasta and Sausages Example

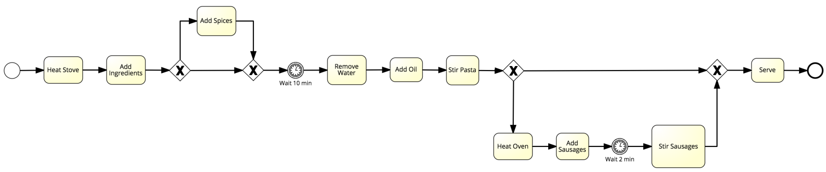


Figure 4.3: Cooking pasta and sausage process

Alongside the analysis of the approach, an example business process will be presented throughout these sections to ease the understanding of it and more importantly to give a more real-life experience to the reader.

The example business process should be a process which covers a significant part of all the cases this approach aims to solve, while at the same time is simple and easy to be understood by the reader. Taking these into consideration, cooking pasta and sausages was selected as an example process.

The selection of this example process wasn't done only because it is already known by most of the people but due to its low complexity and the fact that it satisfies the requirements set in the previous section. This example can be executed in numerous different environments and the user can move around to execute the tasks while also has different execution paths. Moreover, the small number of tasks and the easy acquisition of the objects needed for representing actions as it requires everyday objects like pot, pasta, etc. are some other elements that make this process suitable for being the example.

The flow of the example process is relatively simple with a small number of different execution paths, 4. At the beginning of the process, the user must "Heat Stove" before "Add ingredients" into the pot like pasta or water. Following, there is an XOR gate in which the user decides if he would like to "Add spices" using the pepper bottle like in the fig 4.5 or not. After 10 minutes of cooking, the user "Remove water" from the pot and "Add Oil" to the mix and next proceeds to "Stir Pasta" using the fork shown in the fig 4.4. At this point, the pasta is considered well prepared and the user selects if he would like to enrich the dish with sausages. If the user wishes to, he "Heat Oven" and "Add Sausages" inside the oven. After waiting for 2 minutes, he "Stir sausages" and finally, the user "Serve" the food. In this example, each object represents one activity. This is called mapping type 1-to-1 and will be analyzed extensively in the Mining phase, in the section 6. The objects that represent each of the aforementioned activities in the previous paragraph are listed in the table 4.1. If the mapping type was different from 1-to-1 then in this table we would expect multiple references to one or two columns of each entry.

Figure 4.4: Fork representing the "Stir Pasta" activity

Figure 4.5: Pepper bottle representing the "Add Spices" activity

Object	Activity
Oven handles	Heat Oven
Stove handles	Heat Stove
Pasta	Add Ingredients
Pepper bottle	Add Spices
Sink	Remove Water
Oil bottle	Add Oil
Fork	Stir Pasta
Sausages	Add Sausages
Ladle	Stir Sausages
Plate	Serve

Table 4.1: Objects and the representing actions using 1-to-1 mapping

CHAPTER 5

Pre-Processing Phase

This is the first part of the approach and its main responsibility is object identification from the video and record into an object sequence.

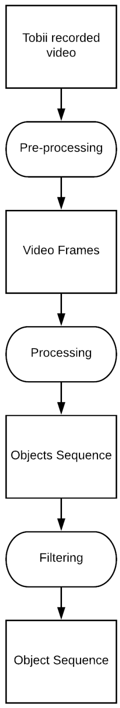


Figure 5.1: An overview of Object Detection Phase

This phase is divided into three parts(sub-phases).

In the first part, analyzed in the in the chapter 5.1, the preparing and optimizing sub-phase, the preparation of the input takes place and the unwanted objects are omitted.

The second part, the processing sub-phase, which is discussed in chapter 5.2 is responsible to identify the objects and list them to object sequence.

And finally, the filtering sub-phase, in chapter 5.3, in which the object sequence is filtered from the unwanted observed objects.

Input

This phase receives as an input three sets of data. The first, the recorded video from Tobii Glasses contains what the user has observed during the execution. This video is in 1080p resolution with the frame rate of 25 fps. [Tob17] The second set of data is the gazing positions of the user during the recording. The gazing positions are based on the recorded video and are encapsulated in a list in which each value contains a timestamp and the position of user's gazing point for both axes. It should be noted that the gazing points are referring to specific pixels of the frames and they are represented by x and y in the fig. 5.1. Finally, the last set of data is a list of objects with a unique ID, so each tag points to a unique object. More discussion on tags is going to be discussed in chapter 5.2.

5.1 Preparing and Optimizing sub-phase

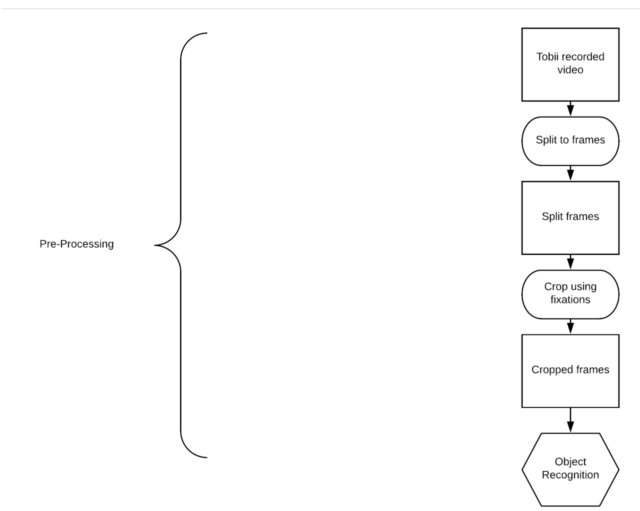


Figure 5.2: An extended overview of Preparing and Optimizing sub-phase

5.1.1 Introduction - Input

This sub-phase receives as input the recorded video from Tobii Glasses and the user's gazing points.

Listing 5.1: Gazing position sequence

```
<(timestamp, x0,y0), (timestamp, x1, y1), ...>
```

Due to the nature of the technologies that are going to be discussed in the next section, the input material cannot be processed in video format but only in still pictures. To overcome this limitation, the video must be split into frames before the processing as it is shown in the figure 5.2. In this sub-phase not only the preparation for the next steps takes place but optimizing the data for better results.

As Steven P. Tipper has suggested when a person is actively trying to execute an activity and more specifically to operate an object, he is mainly gazing at it [Tip10]. To limit any possible detection of unwanted nearby objects and make the processing faster, gaze positions from the glasses are combined with the frames to delete unneeded areas. More specifically, the gaze positions are used to crop a square area centered exactly where the user is looking at each moment.

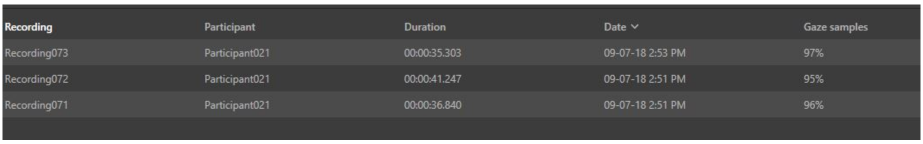
There is no a model that can 100% accurately predict the saccadic movements of user's eyes [PHL16], therefore, the threshold on how far the user can gaze from each object and still operate it cannot be defined. Research suggests it is dependant on environment and the visual cues [MB98] [JBR^{+}16] [MZHG15]. Since it cannot exist a universal threshold to apply to all environments, at this point an algorithm will be presented on how this threshold can be defined for each environment, in the section 5.1.3.

5.1.2 Procedure

To prepare the input for the main processing, the given video should be first converted into still frames, using FFMpeg [FFm18] an open source video suite.

After the conversion to still frames, the optimization takes place. According to Tobii Glasses specification, the visual field of user's view is more than 160 degrees horizontally and 70 degrees vertically. [Tob17] This area is significantly large and besides the additional data for processing, it may include objects that the user hasn't observed. After experimentation that is discussed in the next chapter, it is suggested to combine the gazing positions and the still frames, to crop the unwanted areas.

Due to Tobii Glasses API's limitations, there is no explicit way to match each frame to a gazing point entry. For instance, there is no information that each frame is matched to a specific gazing point entry. On top of that, there is not a guarantee that glasses can detect 100% of the time user's gazing points, as shown in the fig. 5.3, thus, if there are no gazing data for a frame, the frame is omitted. To achieve an accurate match between the frames and the gazing points, an extra processing must be done.



Recording	Participant	Duration	Date	Gaze samples
Recording073	Participant021	00:00:35.303	09-07-18 2:53 PM	97%
Recording072	Participant021	00:00:41.247	09-07-18 2:51 PM	95%
Recording071	Participant021	00:00:36.840	09-07-18 2:51 PM	96%

Figure 5.3: Percentage of gazing data per execution

According to the technical specifications of the glasses, the exported video is 25fps while the detection of user's gazing rate is 50Hz, which means there is a still frame every 0.04 seconds and a gazing sample for every 0.02 seconds. In other words, for every frame there are 2 gazing points entries as it can be seen in the fig. 5.4. The gazing points between the still frames can be excluded from the processing as they don't add any value to the algorithm.

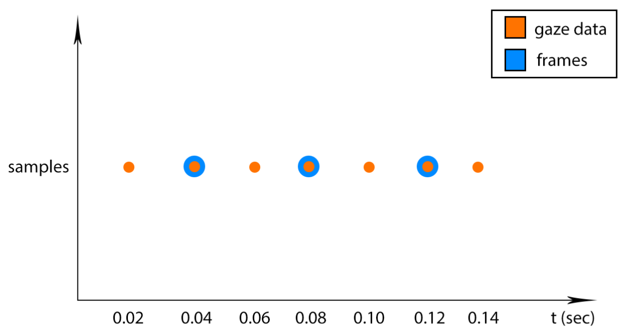


Figure 5.4: The frequency of frames and gaze data

The fact that for every frame there is an entry in gaze data, allows easy matching by calculating the timestamp of each frame and detecting the gaze entry that contains the same timestamp as it is shown in 5.2.

Listing 5.2: Matching between frames and gazing points

```
for each frame{
    frame_timestamp = frame_number * 20 // in milliseconds
    find gaze_entry where gaze_timestamp = frame_timestamp
}
```

In contradiction to the technical specifications of the glasses and experimenting with the output shown there is a mismatch between the gazing points and the frames. Trials have shown that around every 3-6 seconds, the gazing samples are getting delayed by one millisecond. An example with real data can be seen at the table 5.1 when the first delay starts in the entry with timestamp equal to 3281 ms.

Timestamp	X axis	Y axis
0	800	200
20	820	220
...		
3260	500	100
3281	550	150
...		
9882	1000	500
9902	1000	550
...		
48307	670	270

Table 5.1: An example of gazing data from Tobii Glasses 2

After communicating with the technical support of Tobii,[1] it was confirmed this is the expected behavior and in reality, the gaze detection rate is not exactly 50Hz but ∼50Hz. Even though initially, the difference doesn't seem significant, the frames and the gaze entries fail to match and after many minutes the difference can become significantly large.

To tackle this mismatch problem, an extra processing must be done. Before discussing the algorithm it should be noted that Tobii glasses exports an entry

[1] "Regarding the second point. As the glasses, 2 is a digital device the frequency may vary a bit and but overall it is 50Hz. This means that over time you might have 21ms in-between samples or 19ms. it is not a bug but expected behavior."

at ∼50Hz regardless if the gazing of the user is detected. This enables the quick matching by replacing all the real timestamps of ∼50 Hz with timestamps of 50 Hz. When it is completed, there is going to be an accurate match between the frames and the gazing entries. The algorithm is more analytically explained in the fig. 5.3.

Listing 5.3: Manual matching between frames and gazing points

```
calculate precise_50Hz_timestamps
for each precize_50Hz_timestamp{
    replace around_50Hz_timestamp with precise_50Hz_timestamp
}
```

After the match of the frames and the gazing points, there is a need to preserve only what the user is looking at each moment and crop the unwanted area from the frames that may include other unwanted objects (noise). To define what it needs to be preserved and what to be ignored, the concept of cropping area has been defined. The cropping area is a square area which is always centered to the user's gaze position. Everything that is not included in this area, it is considered unwanted and it is omitted. The size of the crop area is different depending on the environment. An example of the crop area is shown in fig. 5.5.

Figure 5.5: Cropping the still frames

The accuracy rate of this sub-phase is dependant on how effective is the cropping area on preserving only the observed objects and omitting the rest. In other words the cropping area should always span the correctly observed object. To make sure this is always true, the distance between the center of the area and its borders must be smaller than the distance to other unobserved objects. Only when it is true in all frames, there will be no conflicts.

An example is shown to the fig 5.6. As the long the distance from center of the crop area to its borders, marked with y, is smaller than the distance with another object, marked with x, there are not any conflicts.

An example of a conflict due to big size of the crop area is shown in 5.8.

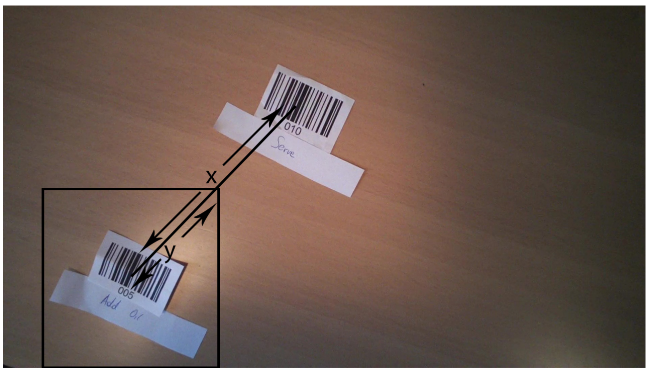


Figure 5.6: The ideal distance of the crop area

At this point it must be clarified that there are two different types of distance, the absolute and the relative distance. The absolute distance refers to the actual distance between two objects inside a room while the relative refers to the distance from user's perspective. Between the two types of distance, there can be an enormous difference, as two objects can be several meters away from another and still be perceived as if they are placed next to each other. An example is given in the fig. 5.7. In this approach, when the distance is mentioned, it is meant the relative distance.

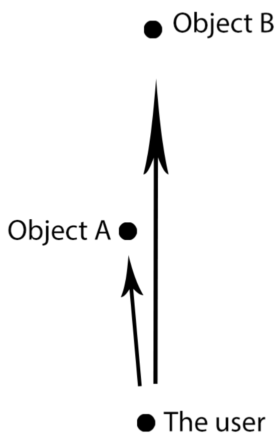


Figure 5.7: An example where two objects seem to be next to each other

Failing to adjust the size of the crop area could lead either on keeping more than one object as it is shown to the fig. 5.9, either on omitting the observed object as shown in the fig. 5.10 or preserving a part of the object for instance in the fig. 5.11.

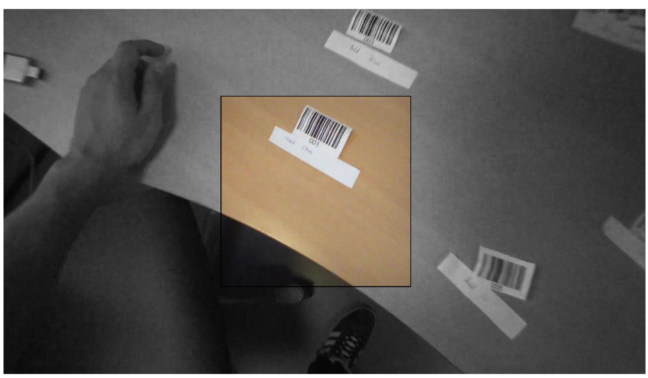


Figure 5.8: Removing unwanted objects from a frame

Figure 5.9: Too big size of the cropping area

Figure 5.10: Too small size of the cropping area

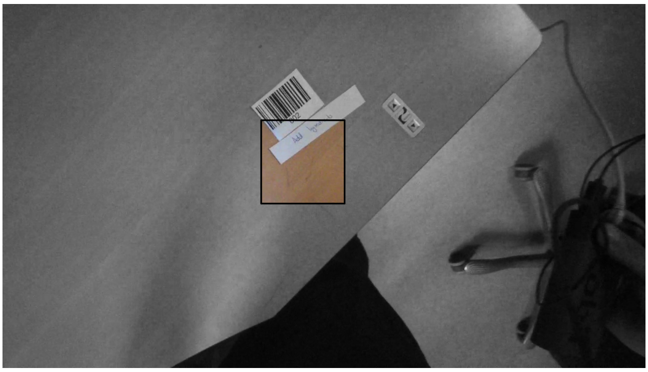


Figure 5.11: Wrong size of the cropping area

There are two parameters that affect the size of the cropping area.

The most important factor, is the relative distance between the objects. In order the crop area to be efficient the objects must be placed relatively far away from each other, so there won't be any conflicts. If the objects are placed close to each other, they crop area will fail to separate them.

Another parameter that affects the size of the cropping area is user's fixations. A number of research work has shown a user's fixation points are based on the visual cues on the scene [Hen03] [MZHG15]. This sub-phase must take into consideration these non-predicted fixations and adjust the crop area to contain the focused object while it follows the fixations.

However, since there are multiple environments with different distances among the objects and multiple users with different fixation trajectories, it is impossible to define a universal size of the cropping area that fits all the cases. As a consequence it is suggested to run a pilot run before the real-life experiments to define the optimal crop size.

After experimenting, to detect the optimal crop area, on the environment of the pasta and sausages example, it is shown that cropping a square area with the side of 350px removes the unwanted objects with the highest accuracy rate.

The cropping is beneficial as it doesn't only speed up the processing by omitting data for processing but removes any unwanted objects that may interfere with the approach and reduce its accuracy.

5.1.3 Experimentation

As it is mentioned in the previous chapter, there is a different optimal crop area for each different environment. It is suggested before deploying this approach in each environment, to run a pilot run in which the optimal area is defined.

The approach of finding the optimal size is by evaluating different square sizes and using the one that detects the most objects without containing more than one object at each frame and cause conflicts.

For better understanding, there will be an attempt to define the ideal crop space for the cooking pasta and sausages in an environment with a limited space where the objects are placed around 20cm from each other. An analysis of two executions will be done evaluating different crop area size and compare the number of identified objects. In both executions, the user is approximately 20-30cm away from each object. This distance was selected as distance increases, the accuracy rate falls sharply.

Taking a look at the first execution figure 5.12, it can be noted as the cropping area size increases the number of identified objects is increased as well. When the square size is 150, 200 and 250 pixels the number of the found objects is 5, 7 and 7 respectively. Increasing the area up to 300 and 350 pixels, the count of objects goes up to 8 and finally, when it reaches 400 pixels all the objects are identified. Even though it looks 400 pixels side is the optimal size it is not this case. In the object sequence in fig. 5.2 there is a big number of conflicts, frames that contain more than one object causing different objects to be detected alternating. An example of conflict can be seen in the fig. 5.13.

Figure 5.12: Crop area size to objects identified

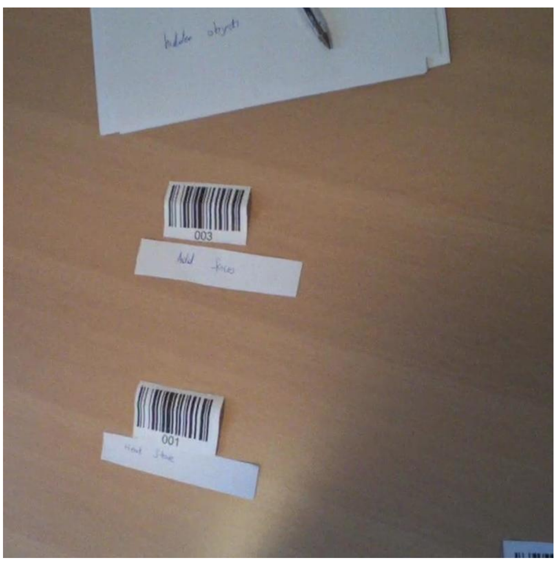


Figure 5.13: A conflict between two objects in the 1st execution

Order	Activity(from Object)
1	Oven handles
2	Pasta
3	Pepper bottle
4	Stove handles
5	Pepper bottle
6	Stove handles
7	Pepper bottle
8	Sink
9	Oil bottle
10	Fork
11	Oven handles
12	Sausages
13	Fork
14	Plate

Table 5.2: Object sequence of the 1st execution

In the second execution, the environment is almost the same except the number of identified objects is lower than the previous execution. Observing the fig. 5.14 it can be noticed the number is almost increasing linearly for every increase in the cropping area. When the side of the cropping area is 150 and 200 pixels, the unique identified objects are 3, while when the side is increased to 250, 300 and 350 the number of the identifed objects is 4, 5 and 6 respectively only to reach the maximum of 8 found objects when the side is 400 pixels.

It must be noted that even though having the side of crop area at 400 pixels yields to the maximum number of objects, it also results to conflicts as the previous experimentation. As it is seen in the object sequence in fig. 5.3 the object for "Heat Stove" and "Add Spices" are in conflict. More clearly can be seen in the fig. 5.15.

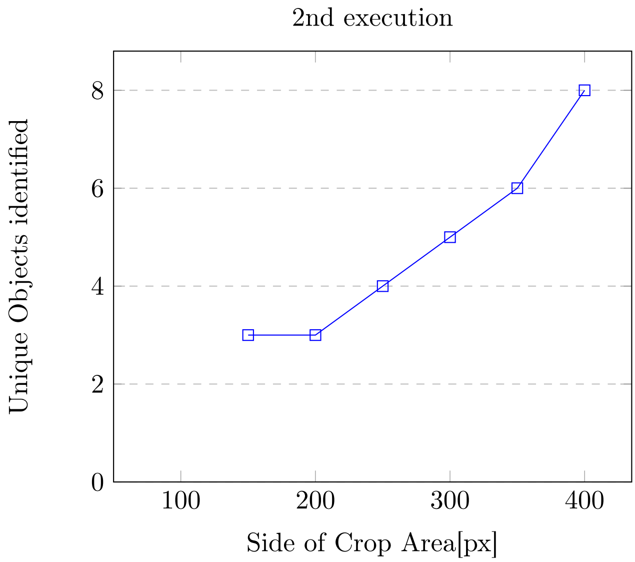


Figure 5.14: Crop area size to objects identified

Figure 5.15: A conflict between two objects in the 2nd execution

Order	Object
1	Heat Stove
2	Add Ingredients
3	Add Spices
4	Heat Stove
5	Add Spices
6	Remove Water
7	Add Oil
8	Stir Pasta
9	Stir Sausages
10	Serve

Table 5.3: Object sequence of the 1st execution

It must be noted that an unexpected behaviour using Tobii glasses was observed. A small number of executions, 2 out of 27, were detected to include strange timestamps that don't match with the video and in general with the principles that it was expected. Strangely, the timestamps seem to make a significant jump from 0ms and the following entry is at 21822ms, an example can be seen in table 5.4. Strangely, the time duration of gazing points is larger than the video but if this jump is removed, the number of the gazing entries match perfectly with the duration of the video. To not spoil the procedure, both of this executions with the unexpected behaviour were completely ignored.

Timestamp	X axis	Y axis
0		
21822		
21842	939	181
21862	938	181
21882	938	181
21902	938	181
...		

Table 5.4: Unexpected behaviour in gazing points

From the two executions, it can be inferred that having found the maximum number of objects doesn't mean that it comes without cost. Both in these executions conflicts occurred bringing the most important question, whether quantity or quality of object identification should be aimed.

To conclude, not only there is not an optimal universal size of the cropping area but also there is not an optimal size for the same environment as each run could target quantity than quality of detected objects and vice versa.

In this example, it was decided that conflicts are not accepted and the side of the crop area was set to 350px.

5.1.4 Output

The output of this phase is the sequence of the cropped still frames ready to be processed

5.2 Processing sub-phase

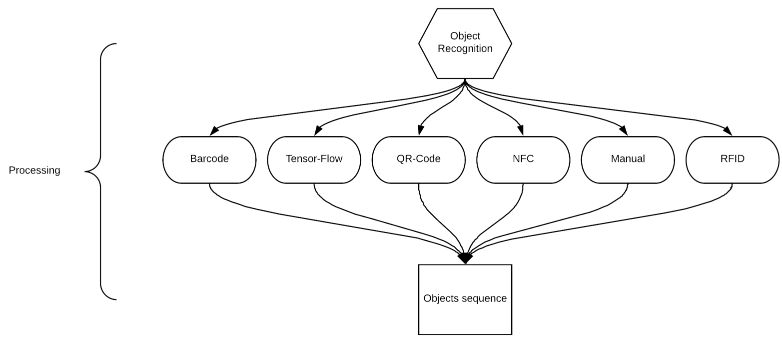


Figure 5.16: An overview of Processing sub-phase

5.2.1 Introduction - Input

Following, the utilization of one among the identification technologies takes place so the objects from the cropped frames to be detected by the machine and draw upon the information given from input. The main issue this part tries to tackle is the identification of objects to be done with the lowest margin of error. This to be achieved, a research on the currently available technologies is needed to compare to which covers most of the requirements which will be set in the following subsection.

5.2.2 Requirements & Goals

Since the proposed approach is designed to be deployed in various environments, the selected technology must meet as many of the following requirements is possible:

1. It must be an optical detection technology that can be used with Tobii Glasses 2§, and able to support various conditions and especially many lighting conditions and still be able to function correctly.
2. Require minimal or ideally, no human interaction to introduce automation to the tracking of the activities execution and decrease the overall execution time.
3. Able to identify objects even if they are partly visible. This will turn the approach suitable for many working environments in which the workers can not have a complete optical view of each object due to obstacles.

Additionally, the proposed approach should aim to cover as many of the following goals as possible:

1. Have the maximum distance detection as higher as possible to reduce the user's moves and decrease the execution time.
2. The accuracy rate of detection should be the highest.
3. The cost should be kept as minimal as possible.
4. Low preparation time to make it fast deployable.
5. Low invasivity to the user to use it without obstructing him from executing the task.
6. The tags of the selected technology should support all size variations without many physical restrictions of the objects(support foldability for example)
7. Support for multiple objects detection simultaneously.

5.2.3 Analysis

The basic approach to identify the objects is by using the frames fed by the previous sub-phase. This approach involves machine learning frameworks that are split into two categories, the category that can identify directly the object and the category that identify an object through visual cues-patterns.

The first category includes Tensorflow while the second category includes Barcode, QR-code, and others. The latter incorporates the use of visual cues, the tags, attached on each object that have a machine-readable visual structure for easy object identification.

For comparison reasons only, as an alternative way, the use of a person for manual identification of the objects was examined along with the use of wireless identification technologies like NFC and RFID.

Barcode

Barcode is one of the most famous optical, machine-readable technologies. It contains vertical lines of varying widths and spaces between them, to represent data thus it can be referred to as linear or one-dimensional (1D). Barcode is used due to its easiness of creating and reading tags, as it needs only a printer and a scanner which has low cost. In this thesis context, Tobii Glasses can function as a scanner requiring almost no human interaction and more importantly no obstruction in task execution. The scanner or the camera doesn't need to be very close the tag in order to read it. Using special equipment, the distance can be increased to 13m [Bar18]. Distance also is dependent on the size of the tags, as there is no a specific threshold on their physical size. The strong advantage of the barcode is the ability to be decoded even if it is partly hidden [unk10] introducing the possibility of having bent tags and still be readable by the reader. However, the tag can only be bent vertically, to a limited angle due to the vertical lines are distorted. There are many variations of the barcode, and each of those has a different maximum character limit. Usually, depending on the variant of the barcode the maximum it can hold is up to 20 characters. [VBHdDSL+12]. Another strong point of Barcode is that tags are material agnostic as long as it recognizable from the camera. For example plastic could be used as long the reflection doesn't interfere with the tag.

QR-code

QR-code (or Quick Response Code) is another optical, machine-readable technology and it could be referred as the 2D variation of the barcode as it follows similar principles in use; it uses low-cost printed tags with price approximately up to 0.001$ [Are18], that can be read and decoded by scanners, with higher data density and capacity [GZ11] [OHH04] up to 2000 characters. On the contrary, QR-code has a stricter limit on the angle of bent, as the utilization of 2D, distorts the grid introducing read errors and critical data loss as it has been noted by Jeng-An Lin and Chiou-Shann Fuh in [LF13]. In addition, due to data density, renders it difficult to be read from distance and effectively the distance is lower than the barcode to around 10 m [Cam18]. The distance can be increased or decreased, depending on the size of the tag. Similarly with Barcode, QR-code as well is material agnostic.

Tensor-Flow

Tensor-Flow is a machine learning framework created by Google and it is fundamentally different than the previous two technologies. Instead of identifying a tag as the two previous technologies, it uses machine learning to detect directly the objects from given images thus decreasing further down the cost as suggested by Andrea Farri in his thesis [Fer16]. Tensor-flow to be able to identify an object must be trained in advance to create the model. There is not a certain number of pictures to be fed into the model but usually it can start from 200 pictures [Edj18] to 10000 [DBLFF10]. The given pictures must be from different angles and different lightning conditions of the object and consequently any object to be detected must be similar to the given images. Consequently detecting partly visible items becomes challenging as it is shown by Andrea Ferri. All these introduces an extremely high cost and a big amount of preparation time [CSAK14] before it can be deployed. On the other hand, there are no definite limits on the optimal distance of the objects, but as the objects becomes smaller and less distinct the detection rate decreases.

RFID

Radio-frequency identification (RFID) is a technology which uses electromagnetic fields to read tags. The devices can be split into readers with antennas and tags which can hold up to 2 KB of data [Rob09]. When a reader comes close to a tag, the latter receives electromagnetic energy from the reader's antenna and uses it to transmit radio waves back to the reader containing the stored data. Due to the physics of RFID, it mandates the user to move and operate an antenna to identify objects, leading to increased execution time. Depending on the power of the antenna, the distance with the tag can be increased, for instance, there are antennas that support distance up to 100 m [RGL11]. Even though the price of an RFID tag is extremely low, at 0.10 $ [Ali18b], the acquisition of equipment has a relatively high cost, from 200 $. Utilizing electromagnetic field allows the tag to be optically hidden from the antenna and still be detectable and readable.[Are18] However, the material of the object in between is one major constraint, as it should not absorb radio-frequencies as metal [Sch04].

NFC
Near-field communication (NFC) is a subset of RFID technology and share many common elements. It is a set of protocols that enable communication between the reader and the tag. Due to physical limitations, the maximum supported distance between the devices is up to 0.1 m [VBHdDSL$^+$12], turning this technology to require human interaction to detect a tag and unable to detect multiple tags at the same time. The main advantage of RFID is the devices can be both readers and tags, and the preparation time is extremely small as it needs less than 0.1 ms to set up a tag [TPJ17]. The cost of an NFC tag is estimated around 0.1 $ and a reader around 38 $ [Ali18a].

Manual
Another alternative solution to detect objects is to use manual labor. This technology was evaluated only to be set as a benchmark and not as an alternative solution. Instead of relying upon the technology and automation, a person can be devoted to manually detect the objects that the user is focused on at each time and record it. The accuracy of using manual labor can be variant from really high to really low, depending worker's focus and fatigue while on the other hand, it has low preparation time, as a human is capable of fast learning. One of the disadvantages of this solution is the high cost as the procedure takes time, it is prone to human errors and more important it keeps an employee occupied from working on other tasks.

5.2.4 Experimentation

In order to draw a conclusion which is the most suitable among optical detection technologies, an extensive experimentation was conducted in real-life conditions. More specifically the two technologies, Barcode and QR-code were compared on the accuracy of detecting tags with various angles between the reader(camera) and the tag, having the tags skewed or even partly visible, having Gaussian and motion blur effect was applied or even different light conditions.

There were conducted two different trial runs, to cover even more real-life conditions and produce robust results. In both trials, the same API was used, Zebra Crossing or ZXing implementation in Java [ZXi18].

The test environment was a typical room with an office with both technical and natural light source and a Samsung S7 Edge was used to capture the images.

In the first trial run, in figures 5.17, 5.18, 5.19, 5.20 both tags are placed on a surface and while the camera takes photos lowering the angle. The angles that were tested are 90, 60, 45 and 30 degrees. While it was expected the results to be positive in both cases, only the Barcode tags were detected successfully.

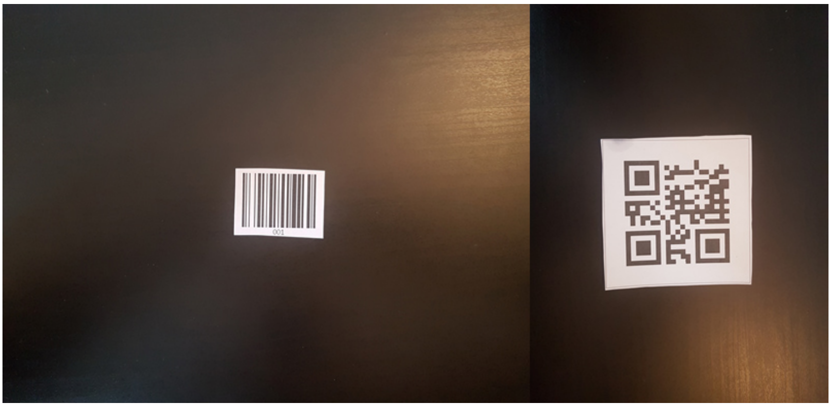


Figure 5.17: Comparison from on top of the tag

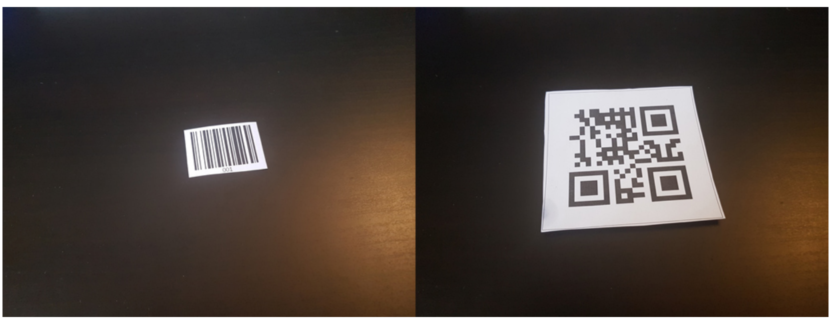


Figure 5.18: Comparison of 60 degrees

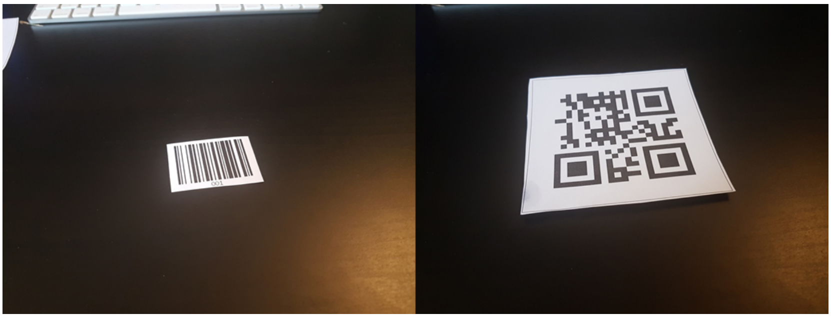


Figure 5.19: Comparison of 45 degrees

Figure 5.20: Comparison of 30 degrees

Still in the same trial run, in fig. 5.21, both technologies are tested on having a part of the tags hidden by an obstacle. Due to limitations of QR-code that they were discussed in Analysis section, it wasn't expected for the tag to be found. On the other hand, the Barcode tag was detected.

Figure 5.21: Comparison having a part of the tag hidden, first trial run

The next comparison includes the ability of detecting a tag even though it is skewed. In this case, none of the tags was identified.

Figure 5.22: Comparison on skewed tag

The last part of the first trial included the use of Gaussian and motion blur filters to simulate the shaken or blurred frames from the Tobii Glasses. The application used to apply the filters was Adobe Photoshop CC [Ado18]. Gaussian Blur filter was applied with the radius of 2 and 5 and in both cases, Barcode and QR-code tags were detected successfully. Following, motion blur effect was applied with distance set at 7, 15 and 20 pixels. Surprisingly, in both technologies, the tags were detected except in the case with the distance of 20 pixels in which Barcode was detected.

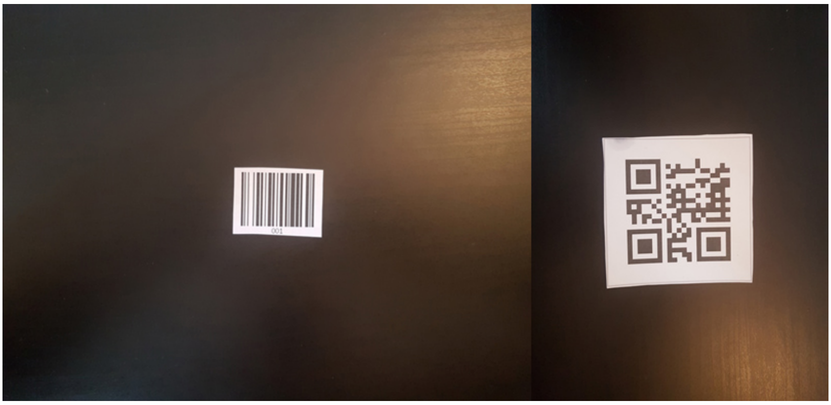


Figure 5.23: Comparison having applied Gaussian Blur with radius 2

Figure 5.24: Comparison having applied Gaussian Blur with radius 5

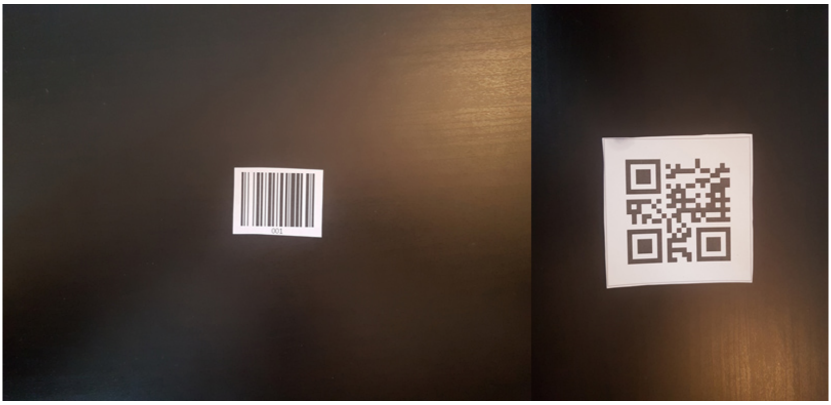


Figure 5.25: Comparison having applied Horizontal Motion Blur with distance 7 pixels

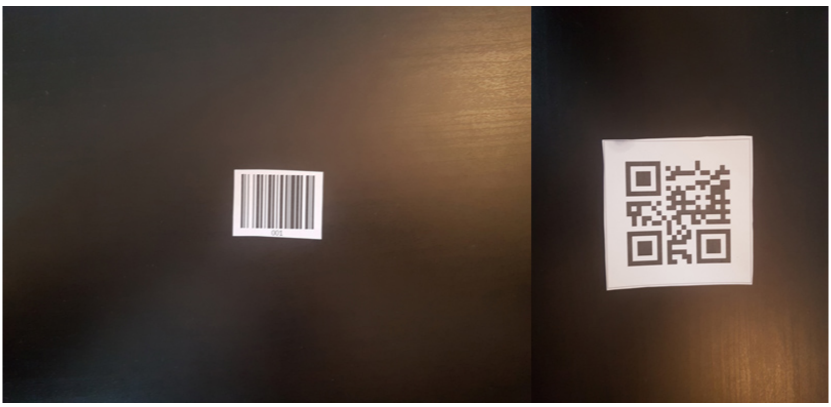


Figure 5.26: Comparison having applied Horizontal Motion Blur with distance 15 pixels

Figure 5.27: Comparison having applied Horizontal Motion Blur with distance 20 pixels

	Barcode	QR-code
From the top	yes	no
Having angle of 60 degrees	yes	no
Having angle of 45 degrees	yes	no
Having angle of 30 degrees	yes	no
Hidden part of the tags	yes	no
Skewed tags	no	no
Gaussian filter (s = 2px)	yes	yes
Gaussian filter (s = 5px)	yes	yes
Motion Blur (d = 7 px)	yes	yes
Motion Blur (d = 15 px)	yes	yes
Motion Blur (d = 20 px)	no	yes
Detection Rate	**83%**	**45%**

Table 5.5: Comparison between Barcode and QR-code in the first trial run

In the second trial run, the biggest difference is the lighting source; now it's natural and in some cases the light is limited. In addition, similar photos were captured with small variations in the location of the camera to create more robust results.

The most interesting cases, is when comparing both technologies from on top, as it is shown in the fig. 5.28, in which only Barcode manages to be detected correctly. Another case which should be mentioned is the comparison with limited brightness in which both of technologies manage to identify the tags. Finally, both technologies manage to identify tags from 1-meter distance on top.

Figure 5.28: Comparison from on top

Figure 5.29: Comparison with limited brightness

Figure 5.30: Comparison from 1m on top

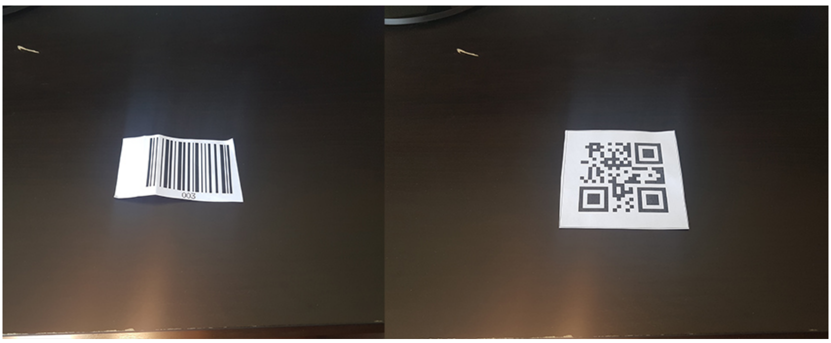


Figure 5.31: Comparison having angle around 45 degrees

Another interesting case is when tags from both technologies are hidden around 25% and 50%. Using the barcode it was successfully detected as it was expected but on the other hand, QRCode failed to detect the tags in both cases.

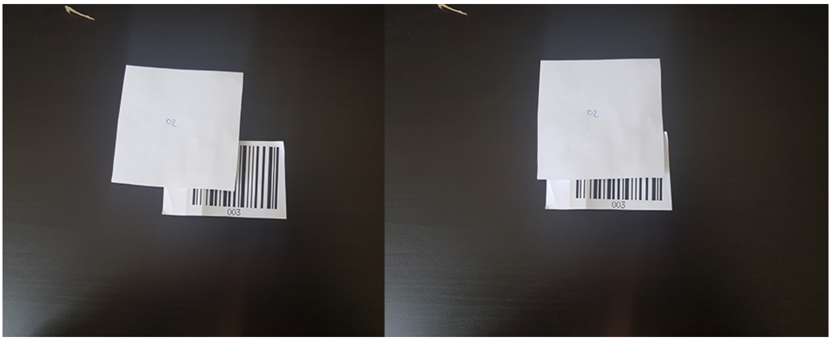


Figure 5.32: Barcode tags hidden approximately 25% and 50%

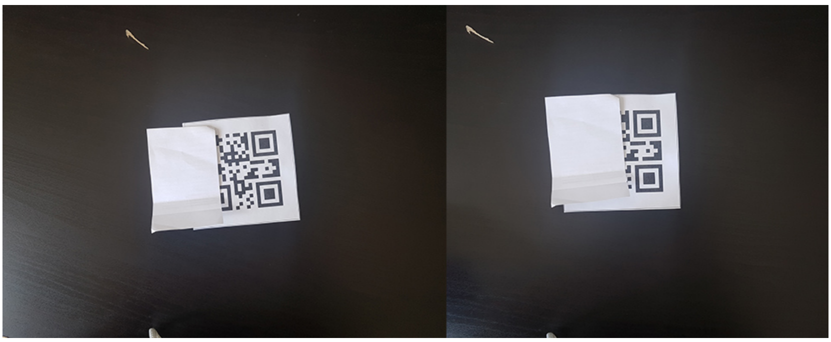


Figure 5.33: QRCode tags hidden approximately 25% and 50%

More analytically the results are listed in the table 5.6. In this execution different percentages of each technology's tag was hidden and as it was expected only Barcode's tag were identified correctly. Next, an evaluation of both technologies was done with limited brightness in which both technology were successful. Following both Barcode and QR-code were benchmarked on identifying a tag from on top with distance up to 1m and then 20cm and finally placing the camera with angle up to 45 degrees from tag. In all of these cases only Barcode managed to be successful with no error in contrast with QR-code which failed in every case.

	Barcode	QR-code
Hidden approx. 25%	yes	no
Hidden approx 50%	yes	no
Limited brightness	yes	yes
From on top (1m)	yes	yes
From on top (20 cm) Var. 1	yes	no
From on top (20 cm) Var. 2	yes	no
From on top (20 cm) Var. 3	yes	no
Having angle at around 30 degrees Var. 1	yes	no
Having angle at around 30 degrees Var. 2	yes	no
Having angle at around 30 degrees Var. 3	yes	no
Having angle at around 30 degrees Var. 4	yes	no
Having angle at around 45 degrees Var. 1	yes	no
Having angle at around 45 degrees Var. 2	yes	no
Having angle at around 45 degrees Var. 3	yes	no
Having angle at around 45 degrees Var. 4	yes	no
Detection rate	**100%**	**13.3%**

Table 5.6: Comparison between Barcode and QR-code in the second trial run

The difference between the two technologies in both trial runs is significant which doesn't leave any hesitation on which of them, should be used. QR-code tags were detected successfully in 45% and 13.3% of all cases while Barcode has almost the double detection rate at 83% and 100% respectively. If the accuracy rate is calculated for both run, the accuracy rate becomes 26% for QR-code and 92% for Barcode. Even though the number of the photos taken for comparisong of both technologies isn't large enough to reach a universal conclusion for any condition it is enough to use Barcode in the context of thesis.

5.2.5 Comparison

	Barcode	QR-code	RFID	NFC	Tensor-Flow	Manual
Maximum Distance Detection	13 m	10 m	100 m	<0.1 m	-	-
Foldable	Yes (1 dimension)	No	Yes	Yes	No	Yes
Partly Visible	Yes	Yes	Yes	Yes	No	Yes
Cost	Lowest + 10000$	Lowest + 10000$	5 + 100 $	0.09 + 40 $	0	Very High
Multiple Objects Simultaneously	No	No	Yes [War16][Siv16]	No	Yes	Yes
Detection Rate	92%	26%	99.9 % [BSRS10]	90% [BCRV16]	variant	High
Preparation	Low	Low	Low	Low	Very High	Low
Fixed Location	No	No	Yes	Yes	No	No

Table 5.7: Comparison of technologies

There is not a technology that meets all the requirements as each technology has its advantages and disadvantages. In this section, a comparison will be drawn and decide which covers most of our requirements.

The most important factor which should be taken into consideration is the cost of each technology. As the research showed using Tensor-Flow would be the cheapest solution is it doesn't involve any purchase of extra equipment or materials. Following, Barcodes and QR-codes even though they have a minimal cost for printing a tag isn't expensive, acquitting Tobii Glasses is expensive, 10000$. Similarly, RFID and NFC would require the purchase of reader and tags totaling around 105$ and 40.1$ respectively. Finally, the manual way of object detection is the most expensive as the average salary of a person starts from 18$ per hour and it is increased linearly for every hour is spent.

Another major factor in comparing the available technologies is the detection rate. The most accurate technologies are the RFID and NFC respectively with 99.9% and 90%, following by the manual way of object detection. Following are the Barcode with 92% and QR-code with 26%. (It should be noted the last two technologies are based on real-life experimentation and the detection rate depends on the APIs as well). Last, tensor-Flow cannot be measured as it mainly depends on the quality of the trained model.

There is one factor which cannot be measured quantitatively but qualitatively, the factor of intrusiveness in the workflow. There is a difference in the nature of these technologies which affect directly the person who perform the tasks. For example, RFID and NFC in order to be detected, require the reader to get close to the tag and place the reader close to it, which introduces a delay and an extra task. On the other hand, the rest of the technologies doesn't require any extra movement but only the tag or the object to be captured by the reader. It should be noted that Barcodes and QR-codes require tags to be imprinted on each object, which maybe is considered too intrusive in some real-life conditions.

Detectable distance is another basic characteristic of each technology. It limits how far an object can be placed and still be detectable. RFID with the respective equipment promises detection distance up to 100m, while Barcode and QR-code offer up to 13m and 10m respectively. It should be noted that there is not a definite limit when using Tensor-Flow and Manual, as there are other factors that affect it. (Camera, image size, model, etc.)

Another two important aspects are the preparation time and the ability to be detected partly hidden. Among the technologies, all of them require low preparation time with only exception the Tensor-Flow which requires a lot of effort to train the models for each object. Following pattern seems to occur regarding partly hidden tags-objects. All of them can identify an object even if it is partly hidden with only exception the NFC and Tensor-Flow.

Among the two optical detection technologies, Barcode managed to meet the most of the requirements set in the previous subsection. It is able to detect from a higher distance than QR-code, while the accuracy rate is more than the triple. Another two important features are the ability of a Barcode tag to be partly hidden or skewed to a certain degree and still be detectable, enabling its deployability in more conditions than QR-code.

5.2.6 Implementation

The implementation of this sub-phase was done using Java. In this phase, all it is needed is to traverse all the received cropped frames and using the Barcode API, to identify the objects by decoding the Barcode tag. Due to the data capacity of a barcode tag is limited to 20 characters, it is needed to use the decoded text and the matching data between the unique identifier and the object so the latter be saved into the object sequence list. Alongside the object's name, the timestamp when it was firstly and lastly detected is saved as well.

5.2.7 Output

The output of this phase, is a sequence of objects detected with their name and the timestamp when they were first detected and last detected.

Listing 5.4: Objects Sequence

```
<(object0, timestampBegin0, timestampEnd0), (object1, timestampBegin1,
    timestampEnd1), ...>
```

5.3 Filtering sub-phase

5.3.1 Introduction - Input

This is the last sub-phase of Pre-Processing Phase and it is purpose is to remove any unwanted detected objects.

As input, this phase receives the object sequence which includes besides the name of the object, it was firstly and lastly detected.

5.3.2 Procedure

Even though there is an optimization algorithm in the Preparing and Optimizing sub-phase which removes any unwanted objects from the background in each frame it doesn't guarantee there won't be any unwanted focused and detected objects. There is still the factor of human error which suggests users can accidentally observe objects either because they think they need them or when they are searching for an object, they shift their focus around the environment.

In order to remove any unwanted observed object, a time threshold must be defined to classify if the object is detected accidentally. This threshold is dependant on the user's experience with the tasks and his fatigue at the current moment. Also, the threshold is dependant on the number and the position of the objects the user is going to operate and in general the complexity of the environment. For instance, when there are many objects placed in an unorganized matter around the working environment, the user is expected to take more time to locate each object thus the time threshold should be increased.

As there are many different factors that affect the time threshold, it can be concluded there can not exist a universal time threshold. For this reason, it is suggested before deploying the approach in a real working environment to make a pilot run to define the threshold and apply it to the future executions.

In the experimentation subsection, the time threshold is going to be defined for the cooking pasta and sausages example.

After the experimentation, it is shown that if an object is observed less or equal to 0.16 seconds it is accidentally observed and should not be included in sequence object. It should be noted that even with filtering, there is not a guarantee that there will be not any object that is observed by mistake as the human factor can be unpredictable and can surpass the defined time threshold.

5.3.3 Experimentation

In this subsection, the approach of defining the time threshold will be discussed through an example. This threshold will be used to filter unintentionally observed objects to reduce the noise and improve the results.

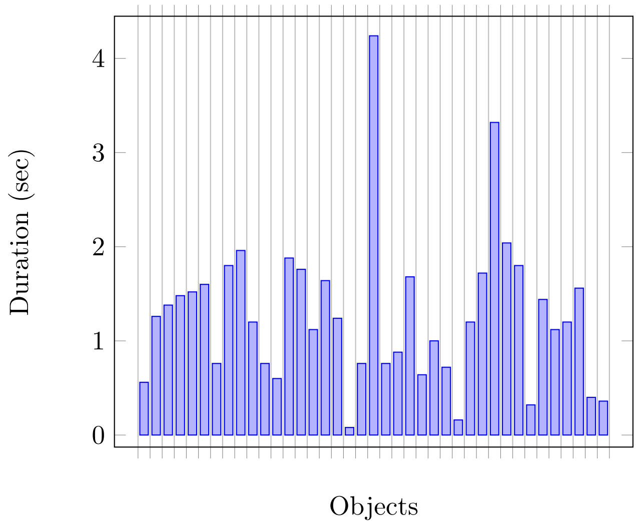


Figure 5.34: Duration of identified objects

The approach is similar to the Preparing and Optimizing sub-phase. Before running the sub-phase in real-life runs it is suggested to make a small number of pilot runs to have an overview of the duration of the objects and define the time threshold. As the number of the pilot runs increases, the robustness of the threshold is increased as well.

During the overview, the accidentally observed objects are identified and the maximum duration of each of these objects is defined. In principle, this duration is lower than all the correct observed objects.

This number is considered as the time threshold and every observed object with lower duration than this threshold is considered as unintentionally observed and should be omitted.

To define the time threshold in the cooking pasta and sausages example, four pilot runs were executed. The environment was similar to the one in pre-processing phase. As it can be observed in the fig. 5.34 the average duration of the observed object is around 1.8 seconds while there are 2 objects with duration above 3 seconds. Also, there are 5 objects with duration less than half a second.

After the analysis of all objects it is found that only two objects were identified by mistake. These objects had a duration of 0.08 and 0.16 seconds, consequently the time threshold should be defined at 0.16 seconds and each object with less or equal duration should be ignored.

5.3.4 Implementation

The implementation of this sub-phase was done in Java. The algorithm is fairly simple as it includes only the checking the duration of each identified object if is above the threshold. If the duration is under the threshold, then the object is removed from the object sequence.

5.3.5 Output

The output of this phase has the same structure as the previous phase but without any unwanted observed objects.

Listing 5.5: Filtered Objects Sequence

```
<(object0, timestampBegin0, timestampEnd0), (object1, timestampBegin1,
    timestampEnd1), ...>
```

Chapter 6

Mapping Phase

Mapping Phase is the second part of the proposed approach and its purpose is to identify activities the user has executed through the sequence of identified objects. This is achieved through the use of filtering technologies and pattern rules.

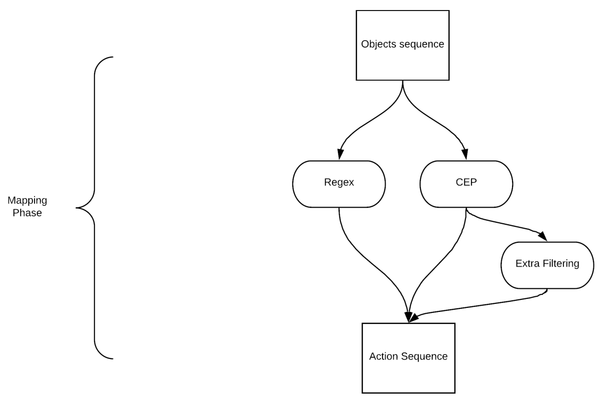


Figure 6.1: An overview of Mapping Phase

This section starts with the discussion of the mapping between the objects and the activities, listing all the types of mapping. Following the concept of user's margin of error is introduced and the procedure of defining it in each environment and each user is presented, totally technology agnostic. After the definition of the margin of error, the requirements are listed and the possible technologies are analyzed, compared and in the end, the most suitable is picked. Finally, an explanation of the implementation is given.

6.1 Input

As an input, this phase receives the object sequences created in the previous phase alongside with a set of activities and the series of the objects that are expected to be found and be mapped. For instance, in our example process, the activity "Add Oil" is matched when the user has observed the "oil", "pot" and then again "oil".

Listing 6.1: The structure of activity to objects sequence

```
<(activity0, [object0a, object0b, object0c, ...]), (activity1,
    [object1a, object1b, object1c, ...]), ...>
```

Listing 6.2: An example of activity to objects sequence

```
<(Add Oil, [oil, pot, oil, ...]), (Add Spices, [pot, pepper bottle,
    pot]), ...>
```

6.2 Analysis

The purpose of this phase is the processing the sequence of objects and generating a sequence of activities.

Mapping is the process in which a sequence of an entity A is matched to entity B. The mapping could include multiple layers of matching before the final output while it can support both directions. For instance, the sequence of entities A could be matched to entities C and finally entities C to be matched with entities B. Also, the mapping could support the match from a sequence of entities B to entities C and finally entities A.

The simplest mapping type is the 1-to-1 mapping, which means each entity A is matched to one entity B and vice versa. An example is shown in the figure 6.2. In the cooking pasta and sausages example applying 1-to-1 object to activity mapping, a match would be the "oil bottle" to be matched with "Add Oil" activity and vice versa, an "Add Oil" activity would be mapped to an "oil bottle" object.

In all the figures that are explaining the mapping type, follow the same rules. The direction of the black bold arrow shows the direction of mapping while shapes with the same pattern are matched together. The colorful rectangular shapes denote the mapping among the entities.

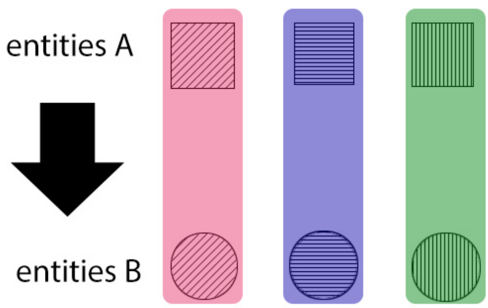


Figure 6.2: 1-to-1 Mapping

Another type of mapping is the M-to-1 mapping, which dictates more than one entities A to be matched with one entity B. A figure that visualizes this kind of mapping is 6.3. An example of M-to-1 object to activity mapping to cooking pasta and sausages example would be the sequence of objects "pan, sausages, fork, pan" be matched to "Stir Sausages".

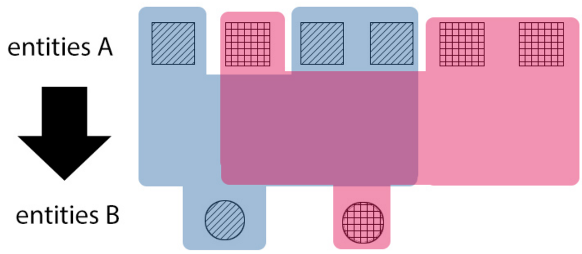


Figure 6.3: M-to-1 Mapping

Mapping type M-to-1 should not be confused with the type 1-to-M type. 1-to-M type of Mapping suggests that one entity A is matched with more than one entity B. Assuming that the object "fork" was used both in stir pasta and sausages, an example of 1-to-M object to activity mapping in the cooking pasta and sausages example would be the activities "Stir Pasta" and "Stir Sausages" to be mapped to the same object, the "Fork". An example of the mapping M-to-1 is shown to the figure 6.4.

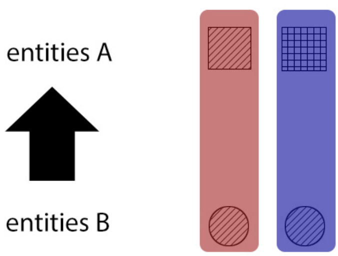


Figure 6.4: 1-to-M Mapping

Another type of mapping is M-to-N which implies more than one entities A can be mapped to more than one entities B. An example is shown to the fig 6.5 in which the same sequence of entities A, can be mapped to a different set of entities B.

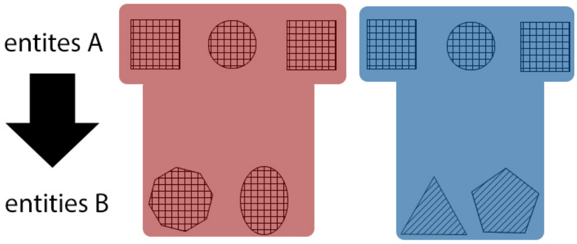


Figure 6.5: M-to-N Mapping

There are some cases in which is needed to run a mapping multiple times and map entities into more generic and broad entities. In the figure 6.6 a multiple M-to-1 mapping are shown from entities A to entities B and finally to entities C. More specifically, in the proposed example, sub-activities like "heat oven", "add sausages", "stir sausages" could be mapped into one activity called "cooking sausages".

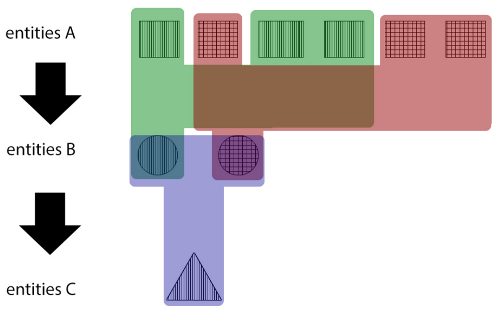


Figure 6.6: Multiple m-to-1 mappings

But before the mapping is executed, the pattern of objects for each activity must be constructed. This pattern is made out of the objects sequence for each activity. During the construction of the pattern, the factor of human behavior must be taken into consideration. The user is prone to observe objects accidentally either due to the complexity of the environment or due to excessive fatigue. Another parameter is the degree of the parallelization the user is trying to achieve. For instance, a user can execute two activities at the same time. For this reason, the construction of the pattern must include a threshold of unexpected objects observed between the expected ones.

Due to these parameters, it is concluded that there is not a universal number of unexpected observed objects and each environment and each user requires a different threshold.

In the experimentation section, 6.3, the procedure of defining the error threshold in the cooking pasta and sausages example. In this example, the chosen mapping type will be 1 activity mapped to M objects. This selection was done due to time and complexity constraints.

After experimentation it was found the threshold up to maximum two unexpected objects has an accuracy rate up to 100% in the cooking pasta and sausages example.

An example automaton of the activity "Stir Pasta" as it is shown to the table 6.1 is presented in the fig. 6.7.

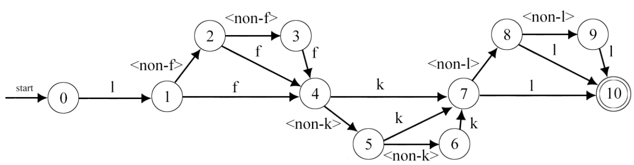


Figure 6.7: Automaton for threshold=2 of "Stir Pasta" activity

6.3 Experimentation

In this section, the process of defining the margin of human error will be defined in the context of cooking pasta and sausages example. Before explaining the procedure the mappings between the objects and the activities must be presented. Each activity is mapped to a sequence of objects which for simplicity and easiness, are represented with a single letter as it is shown in the table 6.1.

Activity	Objects Sequence
Heat Stove	a
Add Ingredients	b
Add Spices	kclk
Remove Water	d
Add Oil	e
Stir Pasta	lfkl
Heat Oven	g
Add Sausages	h
Stir Sausages	i
Serve	j

Table 6.1: Mapping between Activities and Objects

In order to define the optimal error threshold for this example, we have to evaluate different numbers of possible errors and compare the detection rate.

The environment of each execution is the same as it is presented in the section 5.1.3. In total 7 executions were recorded as they are shown to the table 6.2. In each execution run, based on the table 6.1 there will be an attempt to map activities from objects having as a maximum number of unexpected objects of one and two.

In all of the executions, the activity of "Add Spices" and "Stir Pasta" will be attempted to be detected as a benchmark.

Object Sequence	Max One Error Threshold	Max Two Errors Threshold
abde**lfakl**ghij	yes	yes
alke**lfkl**jf	yes	yes
abkcle**lfkl**j	yes	yes
ab**kclc**ke**lfaakl**j	yes (one only)	yes(both)
alke**leefk**ljf	no	yes
abkclecd**lacfkl**j	no	yes
abkclecd**lacfbdakl**j	no	no
Accuracy rate	50%	87.2%

Table 6.2: Comparison between threshold equals to one and two

As it can be seen from the table 6.2 when the threshold is maximum two unexpected objects the accuracy rate is almost 150% of the rate of when the threshold is one, at 85%. The threshold at value one fails in almost half of the cases to identify the sequences of objects leading to wrong mapping.

The significant difference between the accuracy rate of both thresholds leads us to select as a threshold of maximum two unexpected objects.

6.4 Requirements

The proposal should be able to meet all of the following requirements:

There is a need to support all three types of mapping. Not only 1-to-1 but 1-to-m, m-to-1 and m-to-n mappings while multiple mappings. Supporting all of them gives the ability to support complex business processes and make the whole methodology more generic and less restrictive.

Moreover, the proposal should be able to receive both simple and advanced queries containing a specific sequence of objects or activities. As a result, this will render the methodology to support more complex business processes and be more generic.

Last but not least, there should be support for detection of more than 1 occurrence of each query. This will guarantee, all the occurrences of the given sequence to be detected increasing the accuracy rate.

6.5 Evaluation - Comparison

There are two possible ways to map a sequence of objects to activities. One is using regular expressions and the other is CEP applications.

Regular expressions is a sequence of characters that define a pattern. They were firstly originated by Stephen Cole Kleene but they firstly used in text search in 1968. Since then, more and more programming languages started to support regular expressions. One of the most important operators is the Boolean or which is defined using the character |. Another important concept of regular expressions is the ability to define a pattern or a part of a pattern using parenthesis ( ) to be repeated from 0, to n times using operators like ? or *.

On the other hand, Complex Event Processing(CEP) applications could be used to detect a specific sequence of objects to be mapped into an activity. CEP is a method to process multiple streams of data in real time, identify events, reach conclusions after processing them and act [BK09] [ZF07]. Usages of CEP spans across different sections from the security sector to financial sector(stock market). To use CEP each an object will need to be contained in an event so it can be processed and identified. The biggest advantage of using CEP is the support for multiple streams of data and the high speed of execution.

Before the implementation, a comparison between the two aforementioned technologies should be done in order to pick the one that meets all the listed requirements.

Since Regular expressions are used in detecting a pattern inside text(string) there is a need to symbolize our objects and the activities as characters and sequence of characters. This results into limitations as the number of characters is limited. On the other hand, most of the CEP applications are used inside a programming language and they use Plain-Old Java Object(POJO) to represent an event or an object [Apa18] [Fé18]. This support of POJO allows extra filtering of events using multiple properties of each event-object.

CEP applications main purpose is to detect a sequence of events-objects from multiple streams. Their functionalities are based on grouping the events or filtering them based on the time window. Consequently, they are not focused on how to identify a complex sequence of event-objects as regular expressions can achieve.

Both solutions, Regular expressions, and CEP are able to detect multiple occurrences of a specific sequence of objects-events.

Due to the nature of CEP execution, which receives multiple streams of data in real-time and identifies events, it is not capable of matching a sequence of activities, after matching a sequence of objects. Consequently, it can not support multiple mappings. On the other hand, due to regular expressions wasn't designed to run in real time, it is able to detect and replace a specific pattern many times, allowing for multiple mappings.

Moreover, the learning curve of CEP is steep. Currently, there are many CEP applications which even though they have similar functionality, they have a different way of installation and usage. On the other hand, regular expressions is the set of the operators in all of its implementations.

Taking into consideration the comparison of the two technologies and how much they fulfill the set requirements in the previous paragraphs, regular expressions is the selected technology to be implemented in this phase of the methodology.

6.6 Implementation

The implementation of this phase was done in Java. Before any pattern matching, the sequence of objects must be concatenate into one single string containing the names of the objects sequentially, without any character between them. To make the pattern finding easier and faster, if the number of objects are less than 26, the names of the objects could be replaced by the characters of English alphabet before the concatenation. In the second step, for each of the activity, an attempt is made to match into the string. When there is a match, it is stored into the activity sequence, containing the name of the activity and the timestamp when the first object was detected and the timestamp when the last object was detected. When every activity is processed, then the activity sequence is sorted based on when each activity started being executed.

6.7 Output

The output of this section includes the sequence of the activities alongside the timestamp from when it started being executed and when it finished.

Listing 6.3: Activities sequence

```
t = <(activity0, timestampStart0, timestampEnd0), (activty1,
    timestampStart1, timestampEnd0), ...>
```

An example from the cooking pasta and sausages would be:

Listing 6.4: Cooking pasta and sausages mining phase output

```
t = <(Add Oil, 00:01:00, 00:01:20), (Stir Pasta, 01:30:00, 01:50:00),
    ...>
```

Chapter 7

Mining Phase

In the last part of the methodology, the Mining Phase, the input from the Pre-Processing Phase is received and the procession mining is executed in order to create the business model for further research.

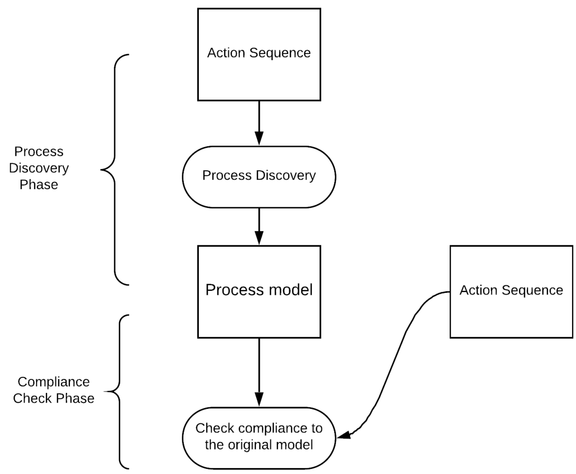


Figure 7.1: An extended overview of Mining Phase

7.1 Input - Introduction

This is the last phase of the whole methodology and its main purpose is to process the activity sequences and create the Process Model. It is divided into two sub-phases. The first phase is Process Discovery sub-phase which uses Disco [Dis18b] to create the process model and the second phase, the conformance Check sub-phase which confirms if a given activity sequence complies with the created process model from the previous phase.

Mining Phase receives as input a set of sequences of the activities from Mapping Phase which contains the user executed activities and when they started and when they finished being executed.

Listing 7.1: Activities sequence

```
t = <(activity0, timestampStart0, timestampEnd0), (activty1,
    timestampStart1, timestampEnd0), ...>
```

7.2 Process Discovery sub-phase

The input of this sub-phase is the same as the input of the Mining Phase, a set of sequences of user executed activities. Following, the sequences is imported into Disco, an application designed for process mining.

Disco is designed to create detailed visual process maps from the imported data. This alongside the filtering of the data such as defining the abstraction level or a specific case allows for easy process mining and detecting issues with the execution of the process model.

Moreover, Disco features the animation of process maps to make the inspection of the process execution even easier and simpler. This results in better detection of bottlenecks and better overall process mining.

The output of this phase is the process model. As it is discussed in the Background section 2, the model could be many types like Petri Net, BPMN, EPC or UML activity diagram.

7.3 Experimentation

In this section, all the activity sequences of the example process will be processed to create the process model. Even though any application supporting process mining is supported, in this document is suggested to use Disco.

Disco supports a plethora of file types to import the data. It supports raw data from .csv files, .txt files or even files from Microsoft Excel, .xls and .xlsx. In addition it support the import of pre-configured files in various standards file formats as .mxml, .xes, .fxl and .dsc. If .txt file is used, specific delimiting character must be used. Only comma(","),a semicolon(";"), a tab ("t"), or a pipe ("|") are supported by disco. [Dis18a]

In order Disco to process the data correctly, the user must define which column of the data refers to Case ID, Start Timestamp, End Timestamp and the name of each activity. This process is shown in the fig. 7.2.

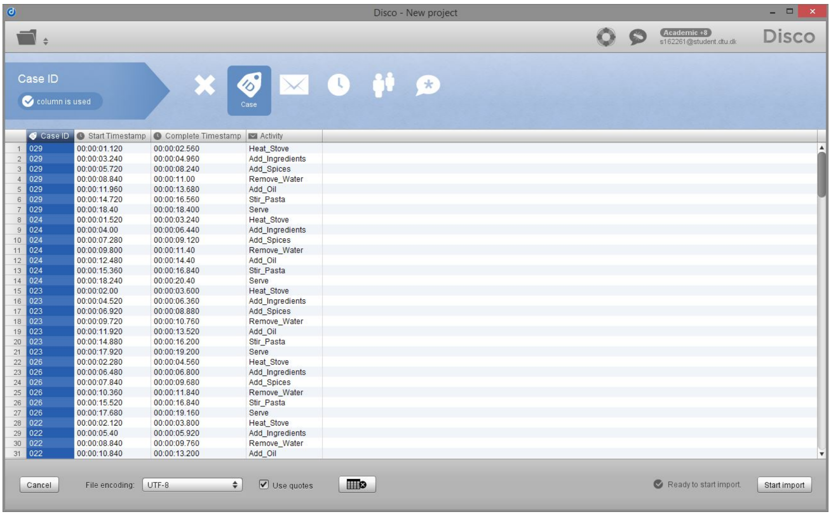


Figure 7.2: Disco import window

If the import is successful, the user is shown the created process model from the data. Disco has the feature to allow the user to define a threshold on how many executions paths or activities will be shown. To create the model in the fig. 7.6 it was selected to show 100% of the activities and the paths. Moreover, the user can view extensive statistics, as shown in the fig. 7.3, as the average duration of each activity or each execution or the meantime it takes an activity to start after the finish of the last activity and other. Another important feature of Disco is the ability to count the variants of the process as it is shown to the fig. 7.5.

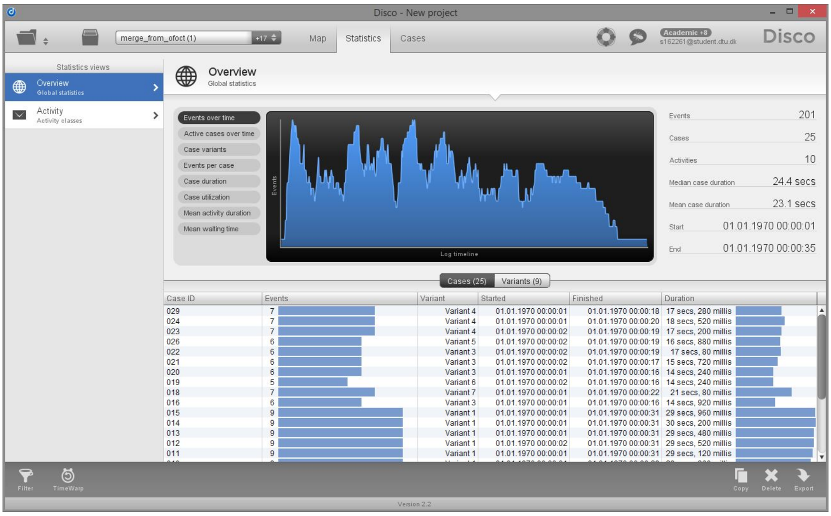


Figure 7.3: Statistic tab of Disco

Another important feature is the ability of user to see a visualization of each execution and have a better grasp, as shown to the fig. 7.4.

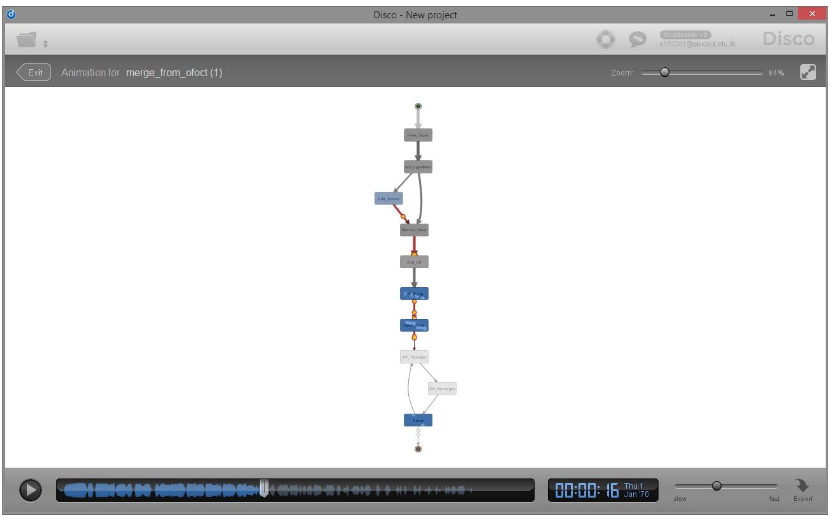


Figure 7.4: Animation feature of Disco

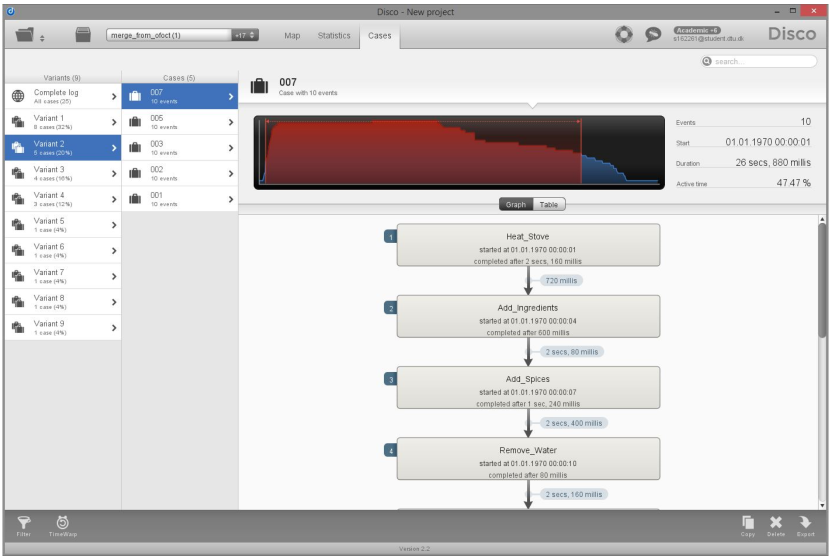


Figure 7.5: Counting variants of the imported activities sequence

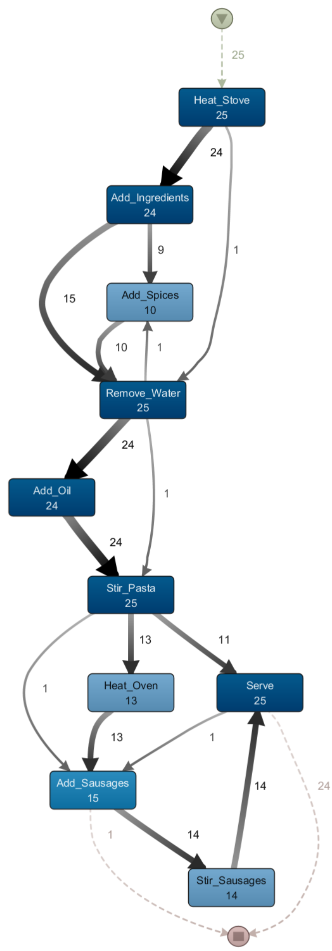


Figure 7.6: Cooking Pasta and Sausages created process model from Disco showing 100% of activities and paths

At this point it must be noted that there were around 27 executions of the process model. From these executions, the purpose was to represent equally the 4 executions paths of the process model. Due to some unexpected behaviour of Tobii Glasses, which was reported in the section of 5.1.3, two of these executions couldn't be processed as it should, resulting to 25 valid executions and leaving some execution paths less represented.

With a quick glance, the resemblance of the BPMN process from the fig. 4.3 can be seen immediately. Observing the paths with the thick line, which means the ones with the most executions, it can be noted which activities are not executed due to an XOR gate. For instance, the activity "Add Spices", wasn't executed every time, neither the "Heat Oven", "Add Sausages" and "Stir Sausages" due to XOR gateways. Some small error detections can be observed, whenever there is a path with only one execution, for example, there is a path from "Serve" to "Add Sausages", which doesn't agree with the initial BPMN model.

These mistakes can be traced to many factors. The most prominent reasons could be either the unoptimized size of the cropping area or a human mistake. If the cropping area is small enough, the whole approach is prone to detecting unwanted objects. Depending on the duration of the observance could also mean that the filtering sub-phase wasn't optimized and didn't filter an accidentally observed item.

Also, viewing the number of variants in the fig 7.5 it can be seen that the total sum of the expected variants is at 80% and only 5 variants out of 25 were with wrongly identified objects. This suggests an accuracy rate up to 80%.

All in all, the produced model has a great resemblance of the initial BPMN model. Though, it can be noted that the semantics are not that explicit; there is not an indication of any operator like XOR gate. If a BPMN model is desired to be created, it can be done using ProM Tools. [Too18]

7.4 Conformance Check sub-phase

The input of this sub-phase is the created model from the Process Discovery sub-phase and an execution log which contains the sequences of activities. It should be noted this section is purely a conceptual approach as it wasn't implemented or tested.

The conformance check is an optional sub-phase of the Mining Phase with the main purpose to check the conformance of the given execution log with the model which created from the Process Discovery sub-phase. Only if there is not a single error the execution log is considered to have complied with the model.

The main algorithm is running based on the produced model. It executes the model and whenever it detects an activity node, it checks if this activity is executed in the correct order in the given sequence of activities. As it can be deduced, this sequence of activities is chronologically ordered, and after every verification, the activity is erased.

More analytically, the possible nodes of the model are separated into two categories. The first category contains the "activities" nodes which are referred to a specific task while the second consists of "special" nodes which are which control the execution flow of the model. For instance, XOR gateway, or AND gateway belongs to the second category.

The first and the last node that can be detected in a model is the "Start" and the "End". Both of these nodes belong to the "Special" nodes category, and their purpose is to mark the beginning and the finishing of the model execution. When a "Start" node is detected, no further action is taken and the execution continues with the next node. But when an "End" is detected, the conformance check exits. In a model, there can be only one "Start" node unlike "End" where there can be multiple.

The simplest case of a node that can be found is when it belongs to the "Activities" category. Activity nodes represent activities and need to be verified they have been executed in the expected order from the action sequence.

One among the "Special" nodes is the Loop node. The loop node encloses a list of nodes and suggests these nodes can be executed multiple times in order. This list of nodes can contain all possible nodes besides a "Start" node. The pseudo-code for a "Loop" node is presented in the listing 7.2.

Listing 7.2: Loop Node

```
repeat = true
while (repeat){
    for activity in LoopActivities:
        try{
            validate(activity)
        }
        catch () {
            repeat = false
            break
        }
}
```

Another "Special" node is the parallel Gateway or AND node. AND node divides the execution flow into two paths and indicates parallelism. With parallelism, nodes from both paths will be executed with unspecified order which path should be executed. As with Loop node, nodes can contain anything beside a "Start" node. The pseudo-code for an "AND" node is given in the listing 7.3.

Listing 7.3: Parallel Gateway

```
counter0 = 0
counter1 = 0
while (true) {
    ANDmerge0 = true;
    ANDmerge1 = true;
    try{
        validate(path0.activity[counter0])
        counter0++;
        ANDmerge0 = false;
    } catch () {}
    try{
        validate(path0.activity[counter1])
        counter1++;
        ANDmerge1 = false;
    } catch () {}
    if (!ANDmerge0 && !ANDmerge1) {
        break;
    }
}
```

Moreover, there is the exclusive gateway operator or XOR node. A XOR node has similar behaviour to AND node but without parallelism. It splits the execution flow to two paths, but only one path can be executed. In order the algorithm detect which of the two possible ways is executed, it attempts to validate the first activity of both paths. The one which doesn't fail, is the executed path. The listing 7.4 shows the pseudo-code for this node.

Listing 7.4: Exclusive gateway operator

```
If (validate(path0.activity[0]){
    follow(path0)
} else {
    follow(path1)
}
```

There are two possible exits from the conformance check algorithm. The first is during the execution an error occurs or the validation fails to detect the correct activity in the activity sequence then immediately the latter is considered as non-compliant and the algorithm stops. The second exit is when an "End" node is found which similarly, stops the execution.

The conformance check can only be correct in the second exit when an "End" node is found in the model and the activity sequence is left empty. Otherwise, the activity sequence is considered non-compliant to the model.

The output of this sub-phase is the results of the conformance check. It can be either true or false.

7.5 Output

Due to the second sub-phase of this phrase which is not mandatory, the output of this phase could either be the generated process model or the confirmation of a given activity sequence conforms with the aforementioned process model.

Chapter 8

Evaluation

In this chapter, an evaluation of the proposed approach will be discussed. Aspects like execution time and the accuracy will be discussed to reach a conclusion about the performance of the approach.

The first major aspect is the accuracy rate. As it is reported in the Experimentation section in the Mining Phase, 7.3, there is a big number of variables that affect the total accuracy rate of the approach:

- Object detection tag
- Object starting and ending timestamp
- Optimized crop area
- Optimized time threshold

Consequently, the complexity of taking all the above variables into consideration to calculate the accuracy rate of the approach is enormous. For this reason, the accuracy rate will be defined as the number of the variants that match completely the given process model divided by the total executions.

According to the Disco, as it is explained in the Experimentation section of the Mining Phase, in 7.3, out of 25 executions, 20 of them matched with the process model, given in the discussion of the example process in section 4. This means an 80% accuracy rate. The rest 5 executions were different from what it was expected as at least one of the activities were not identified correctly or in order. More particularly, the misidentification has occurred in the Processing Sub-phase and more specifically in the identification of objects using Barcode tags.

Another important aspect of the approach is the execution time. The execution time includes the time from the moment the video is transferred to the machine until the model is created.

It must be noted that during the Pre-Processing Phase, in Processing Subphase and in Filtering Subphase, it is recommended to run a pilot run in which the optimal size of crop area and the time threshold is defined. Both of these pilot runs are not included in the execution time as they are executed only for the first time or when the environment or the user is changed.

More analytically the execution time includes:

- Conversion from video to still frames
- Crop pictures
- Detect Barcode
- Filtering
- Mapping
- Importing to Disco and creating the model

In the following paragraphs, the execution time will be measured for each of those elements. The PC is powered by an Intel 6700K with 16GB RAM and a Samsung 850 EVO as a solid state disk.

As it mentioned in the Preparing and Optimizing Subphase in Pre-Processing Phase, the extraction of the still frames from the video is achieved using FFMpeg. Having extracted the frames of 10 executions runs the average extraction time takes almost the 1/7th of the total video duration as it is shown in the table 8.1. It should be noted that this speed is an approximation only and it is true only for the video with the same specifications as the videos created by Tobii Glasses (video codec, bitrate, resolution, etc).

Video	Video Duration	Extraction Duration
1	24 sec	7.26x
2	32 sec	6.9x
3	25 sec	6.69x
4	42 sec	6.98x
5	41 sec	6.89x
6	33 sec	7.03x
7	32 sec	6.92x
8	35 sec	6.87x
9	44 sec	7.09x
10	45 sec	7.01x
Average	**35.3 sec**	**6.96x**

Table 8.1: An overview of the video duration and the extraction time to frames

Next, the cropping of the frames takes place. On the same machine, the cropping of the pictures takes 18.78 seconds while the identification of objects with Barcode tags from all the frames takes 31.73 seconds.

After the creation of the object sequence and the definition of the time threshold during the pilot run, the filtering of accidentally observed is executed. This required 0.30 seconds.

The use of regular expressions to map the applications from one object sequence using 1-to-1 mapping and export them requires 0.12 seconds.

The final step, to import the data to the disco is done manually, using the GUI, adds an extra time. Using the GUI version of disco requires around to 7 seconds to import one activity sequence and export the model.

Finally, the total time needed to process one execution results up to 62.2 seconds as it is summarised in the table 8.2.

Activity	Time Duration (s)
Extraction of frames	4.33
Cropping of frames	18.78
Detecting Barcode tags	31.73
Filtering objects	0.30
Mapping to activities	0.12
Creating the model	7
Total	**62.26**

Table 8.2: Overview of the time execution for each of the tasks

Chapter 9

Generalizability of Approach

In this chapter, the ability of the approach to be adjusted in various environments is going to be discussed.

As it is previously reported in the section 3, one among the requirements set for the approach is to be able to be deployed in a plethora of environments.

The approach of this chapter is to broke down the proposal into small parts and discuss the restrictions imposed by each of these parts before drawing a conclusion for the approach as a whole.

9.1 Pre-Processing Phase

There are three major elements in this phase that define the generalizability of this phase.

The first element is the hardware, the Tobii Glasses. As it can be seen in the fig. 4.1, Tobii glasses contain a plastic frame with a camera and eye-tracking sensors to be placed in the user's head, connected to a small portable processing unit attached to the user. Finally, it's connected with an Ethernet cable to a computer.

All these units, have a slight impact in the invasivity as they put some restrictions on user's movement both of his body and his head. These units use batteries and SD card as energy source and storage for saving videos, imposing another restriction on the duration of the operation. Another limitation that is set by the glasses is the technical specifications of the camera sensor. Due to the small size of the sensor, the amount of light that can be captured is limited leading to raising the ISO(noise) into high levels making the recognition of objects more challenging. Furthermore, according to the technical specifications of this sensor can capture video of 1080p with up to 25fps, restricting user's or object fast movements. And on top of that, the compression of the recorded video decreases, even more, the quality of the image.

The second major element is the object identification technology, Barcode, and the used API. Barcode, as it is previously reported, utilizes printed tags to identify objects, which immediately adds a restriction to the size of the tag and the material of the tag, depending on the environment. For instance, if the environment has high humidity, the tag must not be printed on a paper as the paper gets easily degraded under humid conditions. For example, plastic would be a fine replacement for the paper. In addition, due to the nature of Barcode, there is a restriction on the tag recognition. The tag can be folded only on the y-axis while the experimentation showed there is a threshold on the angle between the camera and the tag. This limits the shape of the object that can be used. Last but not least ZXing, the used API, doesn't allow more than two tags to be identified at the same time.

The last major part is the nature of the algorithm. In "Preparing and Optimizing" sub-phase and the "Filtering" sub-phase, the algorithm suggests defining a threshold in a pilot run before running the real-life experiments. This enables this phase to adjust to a better degree to each environment without omitting essential data.

Type of Restriction	Description
Invasivity	the combination of attached objects and the cables
Time	use of batteries and SD cards
Accuracy	camera sensor's technical limitations + video compression
Invasivity	slow fps of exported video
Accuracy	size of the tag
Accuracy	shape of the object
Invasivity	limited detection of more than one tags simultaneously

Table 9.1: Pre-Processing Phase restrictions

9.2 Mapping Phase

In this phase, there are two elements that affect the aspect of adjustability.

The first element is the use of regular expressions. With the use of regular expressions, there is support of advanced queries that enable the detection of complex objects sequence to map them into a sequence of activities.

The second element is the margin of error that can be taken into consideration building the queries. As it is previously reported, before the real-life deployment there is a pilot run that identifies the optimal margin of error for each environment and user. With the support of regular expression to use advanced queries, it is feasible to define the number of wrongly identified objects among the correct without decreasing the accuracy rate of mapping activities.

Both of these elements allows high adjustability of this phase.

9.3 Mining Phase

This phase doesn't impose any limitation on the adjustability of the approach. This phase includes the selection of an application capable to create a process model and an optional compliance check.

9.4 Conclusion

The biggest source of the restrictions comes from the use of Tobii Glasses as they restrict the physical aspect of the approach (no big range movements, time of execution, storage for video, technical restrictions of the camera sensor). Even though the number of the of restrictions is non-negligible, in real-life most of them don't affect the use as the battery and the SD cards can be replaced very fast while longer cables can be used to extend the movement range of the user.

To summarize, this approach even though places a number of small restrictions on the user, it provides high adjustability to various environments and users.

Chapter 10

Discussion

In this part of the thesis, the degree of how much the proposal meets the requirements set in the chapter 3 is discussed.

As a requirement set in the responsive section, the approach must be an automated procedure. The proposal from the Pre-Processing phase until the Mining phase doesn't need any human input except the part in the Mining phase if the selected application to create the process model is Disco. As it is noted before, Disco doesn't support the use of command line or any other scripting language requiring the user to create the model manually using the GUI. However, the ability to use an alternative application for business mining brings the whole approach to be potentially full automated.

Another requirement that the approach should satisfy is to be able to identify user's tasks. As it is reported in the Pre-Processing phase and Mapping phase, the user's tasks are identified through the observed objects using Eye-tracking techniques and finally, they are mapped to activities.

The approach as it is mentioned in the requirements section must be adjustable to many environments. The ability of the user to move around the environment alongside the nature of the Barcode tags, that can be imprinted or enclosed in protective invisible cases gives a major increase in the adjustability of this approach. Currently, the most restrictive element of the approach is the low quality of the camera of Tobii glasses, as in environments with high contrast in the lighting makes the identification of tags challenging.

The approach to be deployed requires to have low user invasivity. This proposal requires Tobii Glasses, to be installed on the user to track down what he observes at each moment which is connected to a small portable processing unit which is connected to a computer. This setup even though it restricts the movement range, enables the user to operate with his tools with no significant restrictions.

Furthermore, another requirement the approach must meet is to detect objects in many physical conditions. In the Pre-Processing phase, the optical identification technology which was used is Barcode. Barcode allowed the process to detect objects attached with tag without any specific limit on distance or in the rotation of the object. However, The evidence from the experimentation has shown that there is a limit on the angle, between the glasses and the tag and after reaching a certain threshold a tag cannot be identified causing to detection rate to decrease.

In addition, the approach must support all mapping types to identify the activities from the sequence of objects. As it is mentioned in the Mapping Phase, the use of regular expressions allowed to perform identification of activities multiple times allowing for all types of mapping.

The last requirement this approach must meet is to create a process model after identifying the sequence of activities. The generation of activities sequence alongside the proposed use of Disco allows the creation of a business model.

Chapter 11

Related Work

At the time of writing this thesis, there are not any reported research work that use optical identification technologies to perform process mining and create a model. Most of the research work is limited to using technologies as NFC, RFID to identify objects in various environments with stationary equipment.

Florian et al. [SMRM17] propose the use of NFC in the hospital environment to detect the activities of nurses specifically to improve the documentation for each patient. More analytically, it is proposed to install an NFC reader in each patient's bed and attach an NFC tag on care utilities. When the nurse is performing each task, is suggested to use the NFC reader to scan each care utility and record the task on the system.

Jennifer A. Pereira et al. in [PQH+14] discusses the process and the results of incorporating Barcode scanning technology into the Canadian public health immunization settings and more specifically into tracking vials in two health organizations, one in Ontario and one in Alberta. The results she concludes are encouraging and she suggests for further improvement of barcode readability and to expand its use into more vaccination settings like schools, pharmacies, and others. Alan C. O'Connor et al also reported the introduction of QR-Code in USA health system in [OKL+13] to enhance accuracy and decrease paperwork.

The approach presented by Jongchul Song et al. [SHC+06] suggests the use of RFID technology in industrial projects. More specifically, the use of RFID is evaluated in automated detection of incoming pipe spools for better accuracy, fewer errors and less time. The identification of incoming pipe spools was achieved using an RFID reader lifted some meters from the ground while the truck with the objects is passing. A similar approach was proposed by Robin G. Qiu in [Qiu07] in which the RFID was a proposed as a technology to track components cross-businesses to report information regarding the status and the location. Moreover, Matthias Lampe challenged the use of RFID in tool management for aircraft maintenance in [LS03].

Requirements	Florian et al	Jennifer A. Pereira et al.	Alan. C. O'Conor et al.	Jongchul Song et al.	Robin G. Qiu	Matthias Lampe	This approach
Automated procedure	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Task identification	Yes	No	No	No	No	No	Yes
Support of various environments	No	Yes	Yes	No	Yes	No	Yes
Low user invasivity	Yes	-	-	No	Yes	No	Yes
Object detection position & shape agnostic	Yes	No	No	Yes	Yes	Yes	Yes
Support all mapping types	No	No	No	No	No	No	Yes
Process model generation	No	No	No	No	No	No	Yes

Table 11.1: Comparison with related work

As it can be seen from the number of related work and the overview in the table 11.1, there were a numerous attempts to utilize optical identification or wireless communication technologies to achieve higher automation. It can be noticed that most of them are focused on a specific environment like hospital or factories like the work of Florian et al. and Jongchul Song et. al while some of them don't provide any details regarding the reading of the tags, like the proposal of Jennifer A. Pereira et al. and Alan C. O'Connor et al. A big percentage of these proposals, is focused on detecting objects with no task identification.

In the table 11.1 when the symbol "-" is listed, it means the research work doesn't provide any details about this aspect.

Chapter 12

Conclusion

12.1 Analysis of the proposal

The proposal meets all of the requirements that were set at the beginning of this thesis. However, it is crucial to stress some of the technical abilities of this approach. This approach presented an algorithm to combine the data from an eye tracking equipment, Tobii Glasses with an optical identification technology, Barcode to identify objects. More analytically, the gazing data from the Tobii glasses, the video with the user's visual field and Barcode tags are combined to identify objects limiting any noise to increase the accuracy rate. Moreover, this approach described an algorithm of mapping the observed objects to activities that has support for all the mapping types and more importantly it presented how the mapped activities can be used to create a process model for further analysis. The most crucial features of this approach are the need of no human interaction rendering it to an automated procedure alongside the high adjustability to many different environments and different users.

Experimentation showed an accuracy rate up to 80%. This rate is highly encouraging as it points to a possible real life deployment in the future with even higher accuracy assuming some improvements were done.

12.2 Limitations

The approach presented in this document, even though it picked the most optimal technologies and solutions for each different phase, it has some limitations. More specifically, in the Pre-Processing phase, Tobii Glasses were used alongside the Barcode. Due to technical specifications of the Tobii Glasses, they can not be used in environments with high lightning contrast as the Barcode become difficult to get recognized. In addition ZXing, the Barcode API has a certain threshold on the angle between the angle and the camera that can support limiting the user movement. Moreover, due to Barcode limitations, objects that have shape with no big surfaces to place the Barcode tag can impose a limit on which objects can be used for this approach.

12.3 Future Work

The proposal even though it is calculated the accuracy rate can be up to 80%, there are still some aspects that need to be improved to increase the accuracy rate even more. One major aspect that could be improved would be the camera sensor of the glasses, to support even more environments with various lighting conditions and limit the noise. Finally, the Barcode API could be improved on the angle threshold of detecting each tag to raise the accuracy rate.

Moreover, the conformance even as an optional step is needed to extend its support to the rest of BPMN operators to be able to process even more models.

Bibliography

[Ado18] Adobe. Buy Adobe Photoshop CC | Best photo, image, and design editing software. https://www.adobe.com/products/photoshop.html, 2018. [Online; accessed July 22, 2018].

[Ali18a] Alibaba. Long Range Contactless 13.56 MHz USB Interface RFID NFC Reader. https://www.alibaba.com/product-detail/Long-Range-Contactless-13-56-MHz_60718769954.html?spm=a2700.7724857.main07.28.78fd7d14WqFsWd&s=p, 2018. [Online; accessed July 22, 2018].

[Ali18b] Alibaba. Newest! UHF RFID NFC Tag for Stock/Access/Book Management/Coupon. https://www.alibaba.com/product-detail/Newest-UHF-RFID-NFC-Tag-for_1498981019.html?spm=a2700.7724838.2017115.13.6d905197jyqhON&s=p, 2018. [Online; accessed July 22, 2018].

[Apa18] Apache. Introducing Complex Event Processing (CEP) with Apache Flink. https://flink.apache.org/news/2016/04/06/cep-monitoring.html, 2018. [Online; accessed July 22, 2018].

[Are18] Ernest Arendarenko. A study of comparing rfid and 2d barcode tag technologies for pervasive mobile applications. 07 2018.

[Bar18] Barcodes. Motorola LS3578-ER20005WR Barcode Scanner. https://www.barcodesinc.com/symbol/

part-ls3578-er20005wr.htm, 2018. [Online; accessed July 22, 2018].

[BCRV16] Massimo Bertolini, Giada Cilloni, Giovanni Romagnoli, and Andrea Volpi. Benchmarking of rfid epc and nfc tags for apparel applications. *Proceedings of the Summer School Francesco Turco*, 13-15-:187–191, 2016.

[Bes16] James E Bessen. How computer automation affects occupations: Technology, jobs, and skills. 2016.

[BK09] Alejandro Buchmann and Boris Koldehofe. Complex event processing. *IT-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik*, 51(5):241–242, 2009.

[BSRS10] Miodrag Bolić, David Simplot-Ryl, and Ivan Stojmenović. Rfid systems: Research trends and challenges. *Rfid Systems: Research Trends and Challenges*, pages XIX–XX, XIX–XX, 2010.

[Cam18] Camcode. How Bar Code Scanners Impact Scanning Distance. https://www.camcode.com/asset-tags/bar-code-scanning-distance/, 2018. [Online; accessed July 22, 2018].

[CSAK14] Trishul Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 571–582, Broomfield, CO, 2014. USENIX Association.

[DBLFF10] Jia Deng, Alexander C. Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 71–84, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[Dis18a] Disco. Import Process Mining Book 2.1 documentation. https://coda.fluxicon.com/book/import.html, 2018. [Online; accessed July 22, 2018].

[Dis18b] Disco. Process Mining and Automated Process Discovery Software for Professionals - Fluxicon Disco. https://fluxicon.com/disco/, 2018. [Online; accessed July 22, 2018].

[Edj18] EdjeElectronics. How to train an object detection classifier for multiple objects using tensorflow (gpu) on windows 10. https://github.com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Wi 6-run-the-training, February 2018.

[Fer16] Andrea Ferri. Object tracking in video with tensorflow, October 2016.

[FFm18] FFmpeg. FFmpeg. https://www.ffmpeg.org/, 2018. [Online; accessed July 22, 2018].

[Fé18] Bruno Félix. Complex Event Processing with Esper—Core Concepts. https://medium.com/@bruno.felix/complex-event-processing-with-esper-core-concepts-f97394b39c07, 2018. [Online; accessed July 22, 2018].

[Gra95] Lisa Grantham. Justifying office automation: benefits and problems. *Industrial Management & Data Systems*, 95(8):10–13, 1995.

[GZ11] Yunhua Gu and Weixiang Zhang. Qr code recognition based on image processing. In *International Conference on Information Science and Technology*, pages 733–736, March 2011.

[Hen03] John M. Henderson. Human gaze control during real-world scene perception. *Trends in Cognitive Sciences*, 7(11):498 – 504, 2003.

[HNA+11] Kenneth Holmqvist, Marcus Nystrom, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost van de Weijer. *Eye Tracking. A comprehensive guide to methods and measures.* Oxford University Press, 2011.

[JBR+16] M. Jiang, X. Boix, G. Roig, J. Xu, L. Van Gool, and Q. Zhao. Learning to predict sequences of human visual fixations. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6):1241–1252, June 2016.

[LF13] Jeng-An Lin and Chiou-Shann Fuh. 2d barcode image decoding. 2013:1–10, 12 2013.

[LS03] Matthias Lampe and Martin Strassner. The potential of rfid for moveable asset management. In *5th International Conference on Ubiquitous Computing (UbiComp)*, 2003.

[MB98] Brad C Motter and Eric J Belky. The guidance of eye movements during active visual search. *Vision Research*, 38(12):1805 – 1815, 1998.

[MZHG15] X. Min, G. Zhai, C. Hu, and K. Gu. Fixation prediction through multimodal analysis. In *2015 Visual Communications and Image Processing (VCIP)*, pages 1–4, Dec 2015.

[OHH04] E. Ohbuchi, H. Hanaizumi, and L. A. Hock. Barcode readers using the camera device in mobile phones. In *2004 International Conference on Cyberworlds*, pages 260–265, Nov 2004.

[OKL+13] Alan C. O'Connor, Erin D. Kennedy, Ross J. Loomis, Saira N. Haque, Christine M. Layton, Warren W. Williams, Jacqueline B. Amoozegar, Fern M. Braun, Amanda A. Honeycutt, and Cindy Weinbaum. Prospective cost–benefit analysis of a two-dimensional barcode for vaccine production, clinical documentation, and public health reporting and tracking. *Vaccine*, 31(31):3179 – 3186, 2013.

[OMG11] OMG. Business Process Model and Notation (BPMN), Version 2.0, January 2011.

[PHL16] Russell L. Woods Peng Han, Daniel R. Saunders and Gang Luo. Trajectory prediction of saccadic eye movements using a compressed exponential model. *Journal of Vision*, 2016.

[PQH+14] Jennifer A. Pereira, Susan Quach, Jemila S. Hamid, Sherman D. Quan, Amanda Jane Diniz, Robert Van Exan, Jeffrey Malawski, Michael Finkelstein, Salim Samanani, and Jeffrey C. Kwong. The integration of barcode scanning technology into canadian public health immunization settings. *Vaccine*, 32(23):2748 – 2755, 2014.

[Qiu07] Robin G. Qiu. Rfid-enabled automation in support of factory integration. *Robotics and Computer-Integrated Manufacturing*, 23(6):677 – 683, 2007. 16th International Conference on Flexible Automation and Intelligent Manufacturing.

[RGL11] Luis Ruiz-Garcia and Loredana Lunadei. The role of rfid in agriculture: Applications, limitations and challenges. *Computers and Electronics in Agriculture*, 79(1):42 – 50, 2011.

[Rob09] Mark Roberti. How much information can an RFID tag store?? `https://www.rfidjournal.com/faq/show?66`, 2009. [Online; accessed July 22, 2018].

[Sch04] Mike Schneider. Radio frequency identification (rfid) technology and its applications in the commercial construction industry. 2004.

[SHC+06] Jongchul Song, Carl T. Haas, Carlos Caldas, Esin Ergen, and Burcu Akinci. Automating the task of tracking the delivery and receipt of fabricated pipe spools in industrial projects. *Automation in Construction*, 15(2):166 – 177, 2006.

[Siv16] N Sivikami. International Journal of Academic Research in Library Information Science. *IEEE Software*, 2, 2016.

[SMRM17] Florian Stertz, Juergen Mangler, and Stefanie Rinderle-Ma. Nfc-based task enactment for automatic documentation of treatment processes. In Iris Reinhartz-Berger, Jens Gulden, Selmin Nurcan, Wided Guédria, and Palash Bera, editors, *Enterprise, Business-Process and Information Systems Modeling*, pages 34–48, Cham, 2017. Springer International Publishing.

[SWB97] Nadine B Sarter, David D Woods, and Charles E Billings. Automation surprises. *Handbook of human factors and ergonomics*, 2:1926–1943, 1997.

[Tip10] S. P. Tipper. From observation to action simulation: The role of attention, eye-gaze, emotion, and body state. *Experimental Psychology*, page 1, 2010.

[Tob17] Tobii. Tobii Pro Glasses 2, Product Description. `https://www.tobiipro.com/siteassets/tobii-pro/product-descriptions/tobii-pro-glasses-2-product-description.pdf`, 2017. [Online; accessed July 22, 2018].

[Too18] ProM Tools. start | ProM Tools. `http://www.promtools.org/doku.php`, 2018. [Online; accessed July 22, 2018].

[TPJ17] P Thanapal, J Prabhu, and Mridula Jakhar. A survey on barcode rfid and nfc. *IOP Conference Series: Materials Science and Engineering*, 263(4):042049, 2017.

[unk10] *Computer Vision, Imaging and Computer Graphics. Theory and Applications, COMPUTER VISION, IMAGING AND COMPUTER GRAPHICS.* Springer Berlin Heidelberg, 2010.

[VBHdDSL+12] Mabel Vazquez-Briseno, Francisco I. Hirata, Juan de Dios Sanchez-Lopez, Elitania Jimenez-Garcia, Christian Navarro-Cota, and Juan Ivan Nieto-Hipolito. Using rfid/nfc and qr-code in mobile phones to link the physical and the digital world. In Ioannis Deliyannis, editor, *Interactive Multimedia*, chapter 12. IntechOpen, Rijeka, 2012.

[vdAAdM+12] Wil van der Aalst, Arya Adriansyah, Ana Karla Alves de Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos Buijs, Andrea Burattin, Josep Carmona, Malu Castellanos, Jan Claes, Jonathan Cook, Nicola Costantini, Francisco Curbera, Ernesto Damiani, Massimiliano de Leoni, Pavlos Delias, Boudewijn F. van Dongen, Marlon Dumas, Schahram Dustdar, Dirk Fahland, Diogo R. Ferreira, Walid Gaaloul, Frank van Geffen, Sukriti Goel, Christian Günther, Antonella Guzzo, Paul Harmon, Arthur ter Hofstede, John Hoogland, Jon Espen Ingvaldsen, Koki Kato, Rudolf Kuhn, Akhil Kumar, Marcello La Rosa, Fabrizio Maggi, Donato Malerba, Ronny S. Mans, Alberto Manuel, Martin McCreesh, Paola Mello, Jan Mendling, Marco Montali, Hamid R. Motahari-Nezhad, Michael zur Muehlen, Jorge Munoz-Gama, Luigi Pontieri, Joel Ribeiro, Anne Rozinat, Hugo Seguel Pérez, Ricardo Seguel Pérez, Marcos Sepúlveda, Jim Sinur, Pnina Soffer, Minseok Song, Alessandro Sperduti, Giovanni Stilo, Casper Stoel, Keith Swenson, Maurizio Talamo, Wei Tan, Chris Turner, Jan Vanthienen, George Varvaressos, Eric Verbeek, Marc Verdonk, Roberto Vigo, Jianmin Wang, Barbara Weber, Matthias Weidlich, Ton Weijters, Lijie Wen, Michael Westergaard, and Moe Wynn. Process mining manifesto. In Florian Daniel, Kamel Barkaoui, and Schahram Dustdar, editors, *Business Process Management Workshops*, pages 169–194, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[War16] Pete Warden. Can a RFID reader read multiple passive RFID tags at once? `https://www.quora.com/Can-a-RFID-reader-read-multiple-passive-RFID-tags-at-once`, 2016. [Online; accessed July 22, 2018].

[ZF07] C. Zang and Y. Fan. Complex event processing in enterprise information systems based on rfid. *Enterprise Information Systems*, 1(1):3–23, 2007.

[ZXi18] ZXing. ZXing ("Zebra Crossing") barcode scanning library for Java, Android. `https://github.com/zxing/zxing`, 2018. [Online; accessed July 22, 2018].