Computer Vision Based System for Grasping Control of Transradial Prostheses.

With the use of segmentation and shape recognition algorithms

Project Report

Group 863

Aalborg University
Electronics and IT

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Computer vision based system for grasping control of transradial prostheses.

Theme:
Collaborative Robotics

Project Period:
Spring Semester 2022

Project Group:
863

Participant(s):
Bjarke Gjerlev Stück
Marco António de O. Q. Ferreira Alemão
Marek Raška
Robin Leo Emil Cotman
Reshad Zadran

Supervisor(s):
Strahinja Dosen
Miguel Nobre Castro

Copies: One

Page Numbers: 55

Date of Completion:
June 20, 2022

Abstract:

This project proposes a vision system that can be used to control the grasping pose of a transradial prosthesis, to grab a hand held object. The solution identifies a graspable space on the hand held object and computes a grasp pose. It does so by removing background and foreground as well as people, thereby isolating the objects depth information. It can recognise when a person enters the camera frame, even if just partially, such as a hand. The results of the tests show that it is possible to visually isolate an object with mean success ratio of 72%. The shape of the object is determined by using RANSAC to fit a point cloud. The tests showed that it is possible to visually isolate an object and determine the correct shape using at least 50% of the point cloud data. With the lowest mean of the object correctly being determined, being 66%. This was proven to be a success, even despite multiple grabs in the testing, where the hand holding the object was covering large parts of it. When the camera parameters can be adjusted correctly, the system can prove capable of being used to receive an object, thereby improving social engagement for amputees.

Contents

Preface

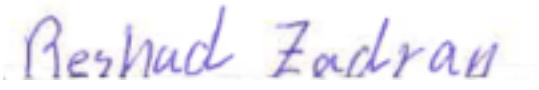
Aalborg University, June 20, 2022

Bjarke Gjerlev Stück
<bstack18@student.aau.dk>

Marco António de O. Q. F. Alemão
<malema18@student.aau.dk>

Marek Raška
<mraka21@student.aau.dk>

Robin Leo Emil Cotman
<rcotma18@student.aau.dk>



Reshad Zadran
<rzadr21@student.aau.dk>

Chapter 1

Introduction

People affected by upper-limb transradial amputation experience limitations when performing their daily rituals, due to the prosthesis control not reaching the level of dexterity a person possesses. The main problem with prosthesis control is the lack of robust control algorithms. Currently, a prosthesis can be controlled with direct myoelectric control, classification-based control and regression-based control [1]. The constraint of the different approaches lies in the Degrees of Freedom (DoFs) that can be controlled simultaneously, to reach a human-level dexterity that limitation must be surpassed. The limitations of the algorithms are described further in Chapter 2. The dexterity of a prosthesis also has an impact on a person's quality of life. Hagberg et al. [2] performed a study where amputees were asked a series of question and one of them was regarding the control of the prosthesis, that question was: "Difficulty directing and keeping control of the prosthesis". Out of 90 amputees, 42% experienced a moderate or worse effect on their quality of life because of the lack of prosthesis control.

Social interaction can affects a person's quality of life [3] and health [4]. Social interaction are a common and important element of life. It supports speech when expressing intention, which can also be expressed without speech e.g., pointing in a direction, is an indication of wanting to go in a desired path. People can often interpret the intention of a interaction. Whether the interaction is perceived positively or negatively depends on the action e.g., pointing at people is seen as a rude interaction, overlapping hands when receiving an object from another person can be off putting. People affected by amputations desire greater control of their prosthesis to allow for better social interactions [5].

Current solutions exist to partially alleviate the burden of a missing limb. Several types of prosthesis exist. The most prominent being myoelectric prostheses. Myoelectric prostheses can have opening and closing functionalities of the hand that can be controlled with signals such as Electromyography (EMG). Further functionality can be added to the prosthesis by mounting various sensor that provide information about the surrounding of the

prosthesis. Sensors that could be used are force sensors, ultrasound, cameras and tactile sensors. The information from the sensors can be used in a machine learning scheme to make the prosthesis semi-autonomous, thus the prosthesis can automatically adjust itself to grasp objects with different shapes, dimensions, and orientation. In [6], [7], [8], [9], [10], [11], [12] and [13] reports the process of achieving a semi-autonomus prosthesis system by combining various technologies, the process is elaborated on further in 2.4.

The first chapter of this report is the problem analysis, where the problem with prosthesis control is described, related work in the prosthesis field is looked into, and a final problem formulation is given at the end. Afterwards the system requirements chapter lists the requirements and delimitations of the envisioned system. Next, is the design, methods, and implementation chapters. The design chapter gives an overall overview of the system, the methods chapter describes the hardware and methods the system uses, and the implementation entails the process of implementing the system. At the end of the report are the testing, discussion, and conclusion chapters. The testing chapter shows the testing setup and the results from the performed tests. The discussion chapter talks about the achieved results and further improvements. Finally the conclusion chapter concludes and summarises this project.

Chapter 2

Problem Analysis

This chapter starts by formulating an initial problem formulation, then it describes the problem in greater depth. Next, some of the active prosthesis control schemes are explained. Finally, it continues by reviewing existing related work solutions which are related to the field of semi-autonomous prosthesis control for object grasping. The chapter is then summarized and a final problem formulation will be derived.

2.1 Initial Problem Formulation

How can the current robotic prosthesis control paradigm be improved to enhance social interactions?

2.2 Problem Description

2.2.1 Amputation

An estimate from 2008 states that there are approximately 10 million people suffering from a loss of limb worldwide[14]. Some of the main contributors to limb loss are vascular diseases, diabetes and trauma[15]. From the 10 million total amputee patients about 30% or 3 million suffer from an upper limb amputation. From those upper limb amputees, an estimated 1.4 million suffer from some form of amputation below the elbow, making it the most frequent type of upper limb amputation.[14] In Figure 2.1 the different types of upper limb amputations can be observed. Since the majority of the upper limb amputations are located below the elbow this project will focus on the case of a transradial amputation.

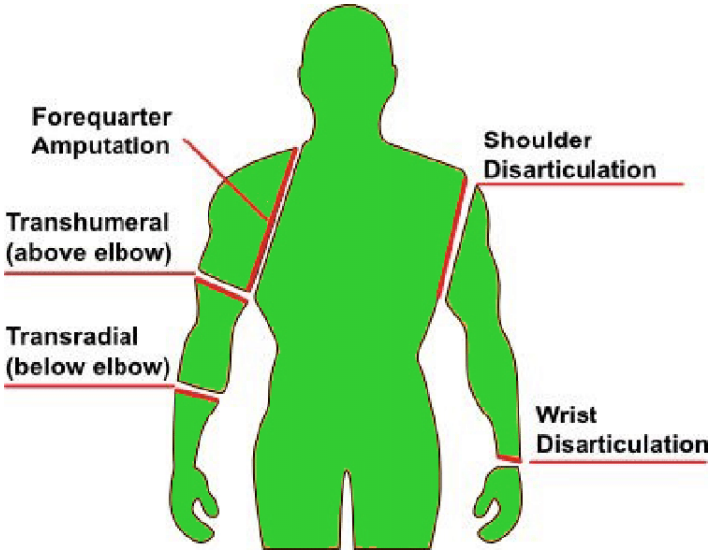


Figure 2.1: Different levels of upper limb amputations, with the focus of this project, transradial amputations, being located below the elbow.[16]

Not only does an amputation impact the physical capabilities of a person but it also impacts their psychological well being. The loss of limb can lead to depression in some patients as well as; anxiety, loss of self-esteem, etc.[17][18]

2.2.2 Grasp type variation for objects

The ability to grasp objects is an intuitive task for most, but with the loss of a limb even the simplest of tasks, such as grasping objects, can become impossible. The vast complexity, capabilities and amount of control of the human hand is difficult to replicate. In the context of grasping objects, the hand has a vast amount of different types of grasps for different objects.

A paper by Feix et al.[19] has researched several taxonomies containing the categorisations of different grasp types. The study has made a new taxonomy by collecting all their findings together, their findings were that around 33 different grasp types exist. These grasp types are categorized based on different object shapes and the type of force that should be applied. For most applications, the palmar, lateral and pinch grasps are most relevant to consider since they are the most widely used[20]. Besides categorizing the different grasp types, the study included a frequency analysis which show that some grasps are more frequently used than others in daily life activities. Based on their analysis the grasp types shown in Figure2.2 are seven of the most frequently used grasp types[19].

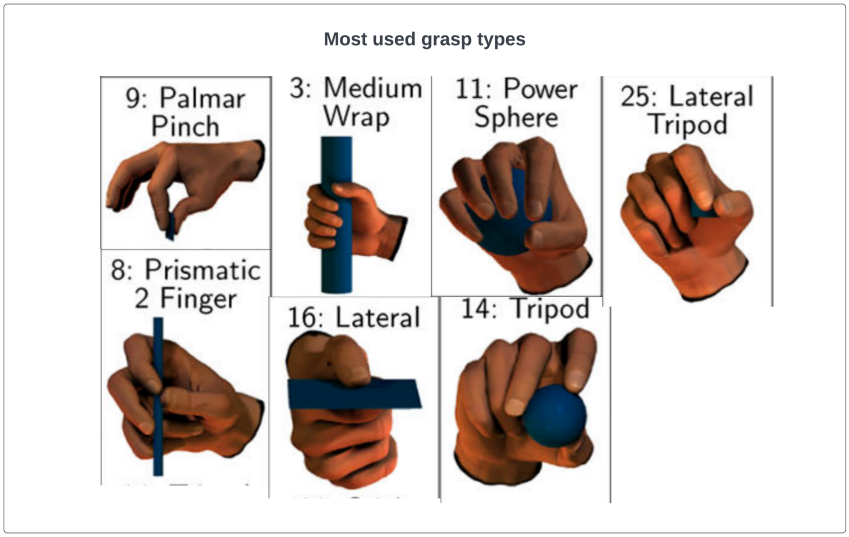


Figure 2.2: This Figure showcases seven of the most commonly used grasp types, used in daily life to grab objects.[19]

The grasp types shown in Figure 2.2, are each used for specific shaped and sized objects. The **Medium Wrap** grasp, is mostly used to grasp cylindrical shaped objects with an intermediate diameter, such as a can. In contrast to this is the **Prismatic 2 Finger grasp**, which is mostly used for holding thin objects, such as pens. The previously mentioned grasp type is also a type of precision grasp, where less power is asserted on the object. Another example of such a precision grasp is the **Palmar Pinch** grasp, which is mostly used for objects of a very small size, like a marble for example. Flat objects, like a card or a key are mostly grasped using the **Lateral** grasp type. Medium sized spherical objects, such as a tennis ball can be grasped using the **Tripod** grasp. For bigger spherical shapes with a grasp where more power is needed, the **Power Sphere** grasp is most commonly used. Lastly the **Lateral Tripod** is also a precision grasp which can be used to grasp smaller sized cylindrical shaped objects.

2.2.3 Prosthetics

Prostheses are commonly used to relieve the patients from their physical impairments, as well as their psychological impairments. Direct myoelectric controlled prostheses are widely used in commercial solutions, an example of how such a prosthesis can look like and what its main components are can be seen in Figure 2.3.

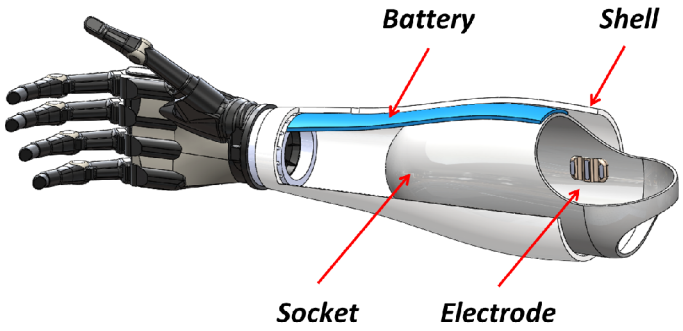


Figure 2.3: Figure of a myoelectric prosthesis, together with its main components.[21]

This proportional direct EMG control makes use of the muscle activity in the residual limb of the user. Although it is used by various commercial solutions, it remains burdensome for the user to operate due to the amount of degrees of freedom which need to be controlled with the limited amount of control. The Cybathlon competition [22] can be used as an example. This is a competition where amputees have to perform various tasks with their prosthesis as fast as possible. It has been observed that it is not the active control based prostheses that win these competitions, but rather body powered prostheses, as seen in Figure 2.4. These prostheses can be used using other parts of the body, e.g shoulder, with cables to perform an open and close action.

Figure 2.4: Figure of a body-powered upper limb prosthesis, with its main components shown.[23]

Even though prostheses can alleviate some of the physical and mental problems of amputees, the control and intuitiveness is still lacking. The control complexity increases with the amount of DoF, potentially becoming cumbersome for the user. With the current hardware available, a prosthesis could reach the same level of DoF as a human hand[24]. The problem remains in the control of the amount of DoF whilst keeping it intuitive for the

user. The level of dexterity in a capable human hand is not achieved with the current methods of prosthesis control[25][26]. Attempts to create more efficient and robust ways for the control of prosthesis have been made. Two such methods are pattern recognition and regression based myo-control. Some of the prosthesis control methods are explained in further detail in the following section 2.3.

2.3 Active Prosthesis Control Strategies

2.3.1 Direct Myoelectric Control

Myoelectric prostheses are controlled using EMG-signals gathered from surface electrodes placed on the residual limb of the user, usually two electrodes are placed on the volar and dorsal muscles of the forearm as shown in Figure 2.5.

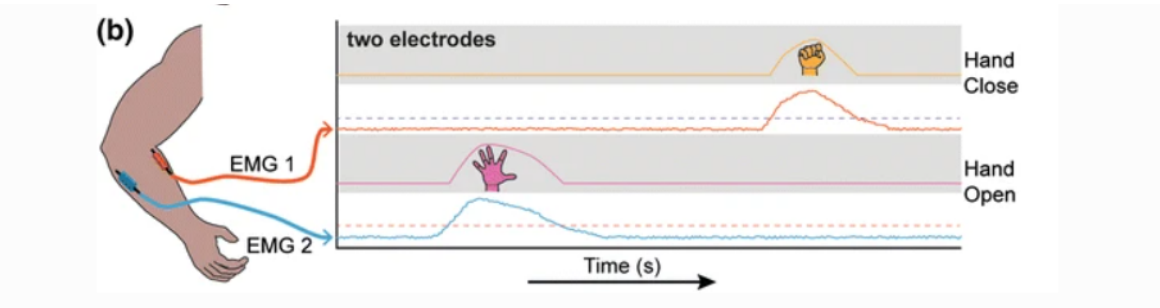


Figure 2.5: Illustration of how two channel myoelectric control works using two opposing electrodes.[27]

EMG-signals are signals which show the amount of electrical current is generated in the muscle when it contracts. The electrodes will record the EMG-signals which are then decoded by the controller unit located in the prosthesis. After the EMG-signals have been decoded they are used to activate the prosthesis to perform an action such as opening or closing of the hand[28]. By varying the intensity of the EMG-signals recorded from the opposing muscles, the prosthesis's grasp can be controlled continuously or proportionally [6]. This method of prosthesis control is a commonly used in commercial prostheses[28]. Even though it is a common method of control it can be lacking if one wants to perform multiple actions at once, such as simultaneously opening/closing of the hand and rotating the wrist. The amount of actions that are able to be performed is limited by the amount of electrodes that are placed on the limb. Since most solutions make use of the direct EMG control using two electrodes, prosthesis actions have to be performed sequentially. This means that for a user to open/close a prosthesis hand and pronate/supinate the wrist, a contraction has to be used to switch in between controlling the hand and controlling the wrist. This is a major disadvantage since it takes some time to perform these actions sequentially. [29][27]

2.3.2 Pattern Recognition Control

Pattern recognition of EMG-signals aims to make the control of a prosthesis more efficient in comparison to direct myoelectric control. A controller with use of pattern recognition, first needs recorded EMG-signals as input. Similar to direct myoelectric control approach these signals can be recorded using surface electrodes placed on the residual limb of the user. When the EMG-signals have been recorded, the useful features are extracted, which can then be classified[28]. With this method, muscle activation patterns can be recognized when performing certain actions, such as opening or closing of the hand. The classifier algorithm can be trained to recognize the muscle activation patterns when performing these actions. When the classifier has been sufficiently trained, it can relate the pattern with the specified action. In the context of prosthesis control this means that when a specific pattern occurs the prosthesis can be controlled to make the related movement [30]. Using pattern recognition a predefined set of commands can be trained, these commands can include the opening of the hand, closing of the hand and turning of the wrist as shown in Figure 2.6. Using only two electrodes can limit the amount of actions which can be recognized by the muscle activation pattern. More electrodes or an increase in input channels provides a broader input range from more muscle groups, not only expanding the amount of actions which can be recognized but also the classification accuracy.

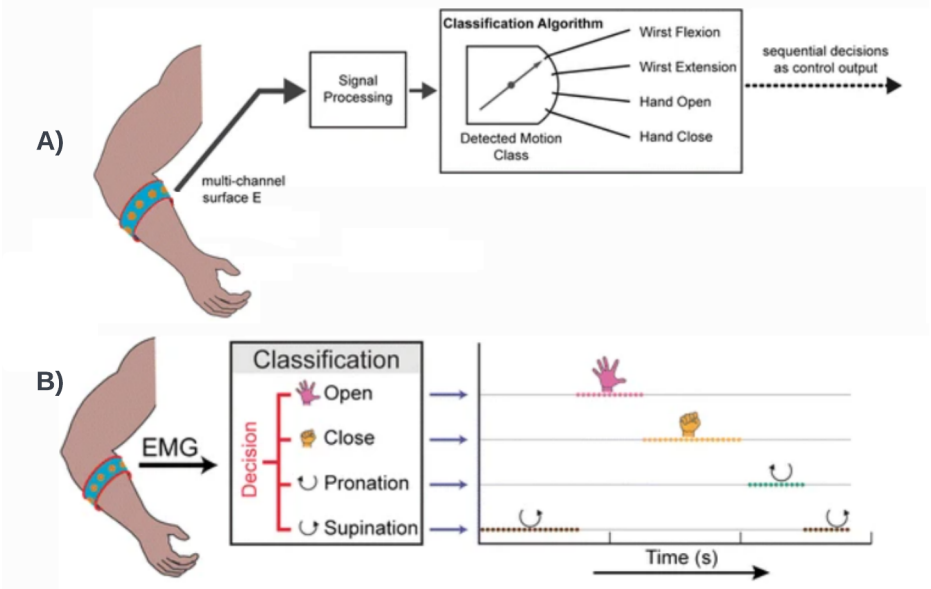


Figure 2.6: A)Shows the structure of a pattern recognition control scheme. B) Shows how the different movement patterns are classified using pattern recognition, note that this is only possible in a sequential way. [27]

Although it is slightly more efficient way of control compared to direct myoelectric control,

this method of control demands tedious training to be able to achieve smooth control[31]. Similarly to direct control, pattern recognition still suffers from only being able to perform sequential actions. Besides only being able to perform actions sequentially it can not be controlled proportionally, meaning the amount of force a user will give will not influence how fast the prosthesis will perform the specified action. Research for control strategies to solve some of the shortcomings of pattern recognition control, such as the lack of continuous control has been done[27].

2.3.3 Regression Control

Control strategies such as regression based myo-control have been proposed to solve some of the shortcomings of pattern recognition control. Unlike pattern recognition control, regression gives a continuous output. This allows the user to control a prosthesis with simultaneous and proportional actions. Regression algorithms, unlike classification algorithms in pattern recognition, give a continuous and multivariate output. This means that instead of a certain preset class, the output of a regression algorithm can have a continuous value. This solves one of the main issues with pattern recognition control, where there is only so many actions that can be trained by the classifier, regression on the other hand allows for a much more robust way of control. Meaning that a user could potentially control their prosthesis hand to simultaneously open/close and pronate/supinate the wrist as shown in Figure2.7. Even though regression allows for more complete control, it still requires training and a major drawback is that only 2 DOFs can be controlled by the user.[32]

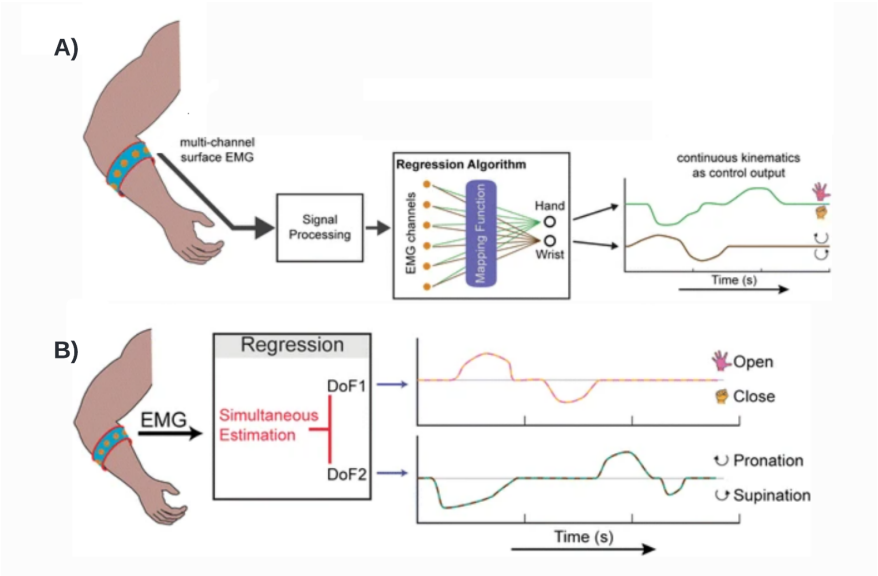


Figure 2.7: A) Shows the structure of a regression control scheme. B) Regression control is capable of identifying simultaneous movements, making it more robust control method compared to pattern recognition. [27]

2.3.4 Findings

Even with current advances in myoelectric control methods, the user is still required to give manual input for most of the operations. Operations such as choosing the appropriate grasp type, or triggering the prosthesis with an EMG-signal to switch between different DOFs can become tedious for the user. Especially when the task of grasping an object is something intuitive for an able bodied person. The problem remains to integrate an intuitive way of control to make the act of grasping as effortless as possible for a prosthesis user. Semi-autonomous prostheses aim to automate many of the manual processes current prosthesis users have to perform. As of now there has been extensive research in the topic of semi-autonomous prostheses, however this project searches to improve in the aspect of social interactive capabilities of prosthesis users. This project aims to build upon what already has been achieved by some state of the art semi-autonomous prosthesis systems. Some of the state of the art semi-autonomous systems are explored in the next section 2.4.

2.4 Related Work

This section presents the related work for semi-autonomous prosthesis control methods. For the context of this project, works related to adding a computer vision sensor to the prosthesis control scheme have been researched. The section starts with Semi-autonomous prosthesis control, where some of the works which laid a foundation in the field of semi-autonomous prosthesis control are mentioned. Afterwards a more in depth research into the field of object recognition and object grasping in the context of prosthesis control is covered. Finally a summary will be given and the final problem formulation will be derived.

2.4.1 Semi-Autonomous Prosthesis Control

As described in the earlier section 2.3, research is being done for semi-autonomous prosthesis control. Research has been done in replacing/enhancing the current direct surface EMG based control[33]. The ability for a prosthesis control system to recognize the object it is intended to grab can open many possibilities in automating the process of grasping. Therefore an additional source of information such as adding a computer vision element to the prosthesis control scheme has potential.

For the context of this project, studies related to the addition of computer vision to automate the grasp type selection process for a prosthesis to grasp objects has been researched. One such work by Došen and Popović [20], combines multiple sensors, such as an ultrasound sensor to get depth information together with an RGB camera. Their system proposes to use the combination of the two sensors to create a practical way of control for a transradial prosthesis. The system works by first aiming the sensors in the direction of the desired object. Afterwards, using computer vision methods the size and orientation of

the object are found. When the appropriate data has been collected the system can output the appropriate grasp type and grasp size to grasp the desired object as well as to how much the wrist should rotate to be able to grasp the object. Another work by Ghazaei et al.[10] proposed a system with the same objective to enable a transradial prosthesis user to grab household objects making use of a computer vision system. However their approach was to train a Convolutional Neural Network(CNN) with images of over 500 graspable objects. In contrary to [20] which determined the dimensions of the object,[10] classifies the objects in regards to their accompanying grasp type.

2.4.2 Object recognition

Object recognition can be done, with several techniques. Feature based visual systems, which is one of the most researched representation method in this field according to Carvalho et al. [34]. These techniques have lately been outperformed by neural networks, which have the advantage of being able to simultaneously predict bounding boxes and probabilities. As of now some of the new versions of convolutional neural network (CNN) systems for object recognition like YOLO and YOLOv3 are good candidates for the implementation of an object recognition algorithm. These systems are capable of running at more than 45 frames per second(FPS), but it is not ideal for object segmentation [35]. For object recognition to be usable with prostheses in mind, the object has to be separated in the image from other objects, such as hands holding the object. This is the concept that Bolya et al.[36] introduced with a work called, YOLACT. YOLACT can segment instances in real-time compared to YOLOv3 that can only obtain bounding boxes. Real-time in this case is stated as FPS that the eye can no longer recognise sequence of pictures and instead sees motion, which is according to them above 30FPS. With an object separated and isolated, model fitting can be performed. A method introduced by Fischler and Bolles[37], called Random Sample Consensus(RANSAC) is a great candidate for model. RANSAC can generate candidate solutions for model parameters even with minimum number of observations.[37] [38]. In some scenarios object recognition and grasping can be combined together with neural networks as presented by Zhong et al.[39], but this could lead to a person with a prosthesis of this type to dissociate from it, because of no user input, which is why it is avoided.

2.4.3 Object grasping

In the field of, grasping an object with a prosthesis a lot of work exists [6], [7], [8], [9], [11], [12], [13]. In general, information on the shape, dimension, and orientation of the object is used to determined the grasp required to grip the object. Furthermore the prosthesis user must be able to close and open the prosthesis when attempting to grasp the object. The required functionalities are archived by making the prosthesis semi-autonomous. The work by Castro and Dosen [9], presents a semi-autonomous prosthesis by using the depth data from a depth sensor placed on the dorsal side of the hand. The depth data is used in their

vision algorithm to pre-shape the prosthetic hand when the sensor is pointed at an object, thus the prosthesis automatically adjusts its configuration (wrist orientation, grasp type, and size) when grasping objects with different shapes, dimensions, and orientation. Their semi-autonomous prosthesis is capable of rotating the wrist, opening/closing its fingers in palmar and lateral grasps.

2.5 Summary

This chapter introduced the initial problem formulation, and some of the struggles of amputees and the variety of different grasp configurations has been mentioned, giving an understanding of the complexity of the human hand. The methods of active prosthesis control have been explored and the shortcomings of these methods of control are mentioned. Related work in the field of semi-autonomous prostheses control has been found. The research from the related work yielded in finding papers regarding; addition of computer vision to enhance a prosthesis control system, object recognition in the context of prosthesis control and object grasping. From the analysis of the papers, no papers were found to be about grasping an object with a prosthesis from the hand of another person, hence that area was chosen to be the novelty of this project. Additionally, a final problem formulation was derived from the related work and the scope of this project.

2.6 Final Problem Formulation

How can a computer vision system be used to automate the grasp type selection and size for a transradial prosthesis when receiving an object from a persons hand?

Chapter 3

System Requirements

This chapter proposes the requirements and delimitations to the solution answering the final problem formulation presented in Section 2.6. The requirements are derived from research made in the related work 2.4 and from the scope of this project, in order to ensure that the requirements are quantifiable and testable in a real life scenario.

3.1 Requirements

1. *The system must be able to visually isolate a graspable object from the hand holding it, at least 70% of the times.*

   In order to receive an object given by another individual, a distinction between the human hand and the object must be made. This is to get the object information with as little clutter as possible. A graspable object, being an object that can be held in one hand.

2. *The system must be capable of determining a 3D shape of an object, using 50% of the object depth information.*

   The identification of an appropriate grasp type when picking up an object, is largely dependent on the shape of the object. It was decided that 50% inliers is enough to ensure a robust classification.

3. *The object must be correctly classified with success rate of 90% as either a cylinder, sphere or cuboid.*

   The model classification of 90% success rate was defined as acceptable, as it represents the last stochastic evaluation needed for grip shape classification.

4. *The model must be estimated within an overall size error of 10%*

   Another important factor when deciding the grasp type to be used on an object, are the dimensions of the object. This requirement will verify how accurate the object

dimension representation is in the system. The size error of 10% was determined to be enough to ensure the correct dimensions and still be graspable.

5. *Choose identical grasp type as the one assigned for the object, depending on the shape and dimensions and have a success rate of over 80% of attempts.*

   This requirement will verify whether or not the system is capable of choosing an appropriate grasp type for an object, even if the model is incorrectly classified. The assigned grasp type for the objects have been chosen beforehand. The 80% success rate was chosen as acceptable.

6. *The system must be able to acquire visual information and determine the appropriate grasp type and size, in less than one second.*

   This requirement ensures that the system is capable of running in a real time application.

3.2 Delimitations

1. *Processing limitations* The system should run on a laptop similar to one this project is using for visual information processing. This means that online processing or any other elevated processing capabilities will not be considered in this project.

2. *Orientation limitation* Due to limited time constraints for this project, the hand giving the object will be assumed to hold the object in a vertical angle, perpendicular to the camera.

3. *RGB-D Camera limitations* Stereo cameras fail to reliable compute the depth of non-Lambertian or pattern filled surfaces. They also have a minimum distance at which it is not possible to compute depth information.[40]

4. *Exclusion of EMG signal processing and prosthesis control* Because of the limited time, the control of the prosthesis will not be considered when making the solution. The solution will display an estimated grasp type, when presented with an object held in another individuals hand.

5. *Camera placement* The camera placement in the prosthetic hand will not be considered.

Chapter 4

Design of System

This project aims to design a computer vision system to be used on a prosthesis, to aid in the task of grasping objects from another person's hand. By automating the grasp type and grasp size selection, the burden of manual control is alleviated from the user. In this chapter the overall system architecture is explained. The methods that will be used for the implementation of the system, are introduced in this chapter, but they will be further explained in the Methods chapter. 5

The flowchart represented in Figure 4.1 showcases a system which can possibly integrate the solution presented in this report.
Here, a user could point the camera at a desired object, in another person's hand, then the prosthetic hand would be able to identify the object shape and change to an appropriate grasping pose. An EMG trigger could be used to correct any mismatch of the grasp type.

The system proposed by this project aims to address the issues regarding the items from one to five on the flow chart. Here, the RGB-D camera(1) passes an image wich will be used by a CNN(2). Semantic segmentation will be performed on the RGB-image received from the camera. Any part of a person will be defined as a Region of Interest(RoI). A mask of the objects class **person** will be returned. It also generates a coordinated depth map. In the image processing(3) step the depth masks are used to set to the depth pixels of class person, to zero.

Later, a threshold is used, which sets both foreground and background pixels to zero. The threshold is implemented to further isolate the object of interest. This will create a clean depth map, only containing the depth data of the object held in the hand(3). For the object recognition and size estimation of the held object, the depth data is first converted to a point cloud. The shape of the objects are found by model fitting using RANSAC, afterwards the size of the object is also estimated with methods such as Principle Component Analysis(4). The shape and dimensions will then be used to select the appropriate grasp type and grasp size. The appropriate grasp type can be predetermined for the shapes.

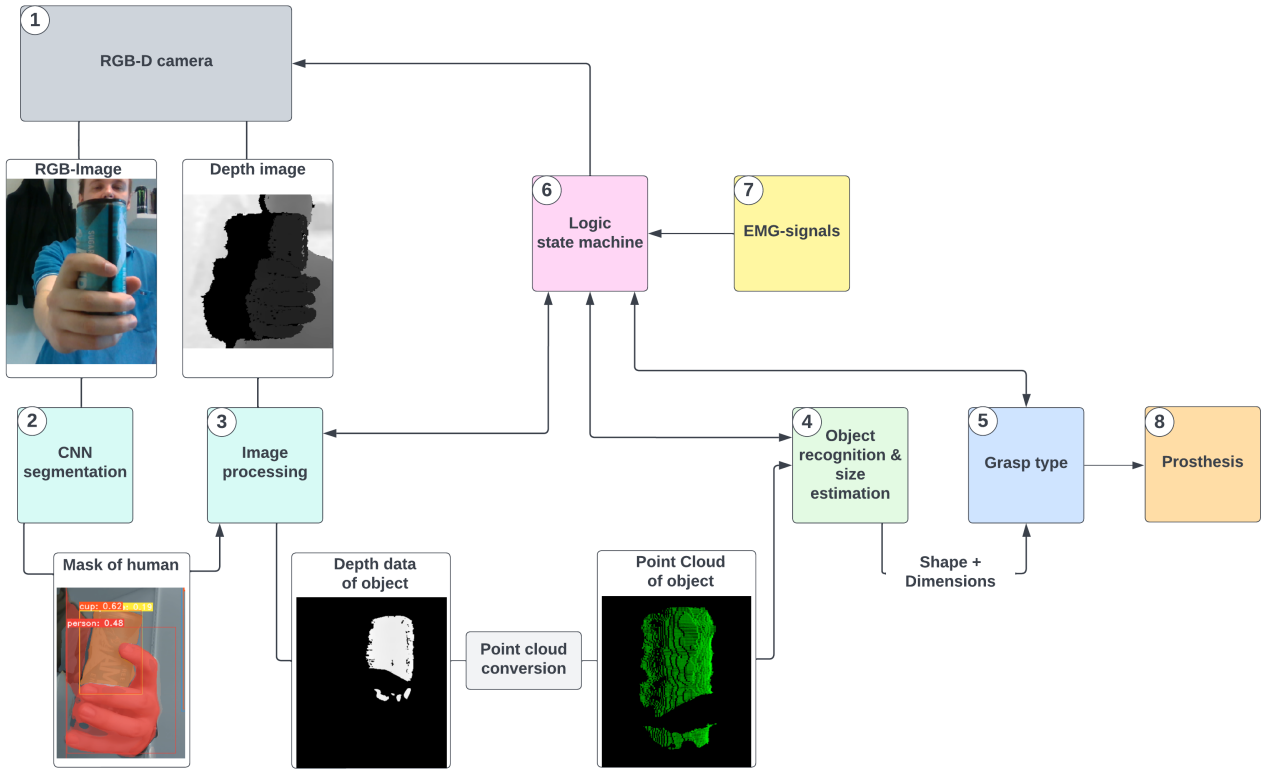


Figure 4.1: General overview of the system. The numbers do not necessarily describe the order of the system, but it is used to refer the specific part in the documentation.

The aforementioned processes,(3)(4)(5), are controlled using a state machine. In coordination with input from the user, the state machine decides when one process starts and the other ends. This is done in order to prevent the grasp process to change uncontrollably and to reduce computations.

4.0.1 Delimitation of system

This project limits the scope to develop a computer vision system, focusing on the person detection and object recognition. The processes (7),(8) in 4.1, namely the implementation on the prosthesis and the EMG-signal processing for the activation of the logic state machine are not in within the scope of this project. Therefore the Computer vision processes will be handled manually and not by a logic state machine. This also means that in the current project the camera will run continuously which is in contrast with what was initially planned.

Chapter 5

Methods

This chapter describes the methods used in this project and the hardware necessary for their implementation. First the specification of the hardware is given. Afterwards, the methods used for image segmentation and object shape recognition are explained. In the hand object segmentation section, the following topics are discussed; convolutional neural networks, regions of interests, semantic segmentation, and YOLACT. In the shape recognition section, point cloud, RANSAC, and PCA are explained.

5.1 ROS

Robot Operating System (ROS) provides communication tools that allows for data exchange between software and hardware. ROS allows for communication across different nodes through topics, using publishers and subscribers. Nodes can perform multiple operations, including interact with hardware. Each topic has a unique name and provides unique information, further, the same topic can be published or subscribed by multiple nodes. A node that shares information uses a publisher to send data through a topic. If a node needs the published data it can subscribe to that specific topic to receive the information. For example, a camera node can provide topics such as color image, depth image, and camera information. Subscribing to the depth image topic will give the depth data that is published by the camera node to depth image topic. The process is explained further in chapter 6. The reason why ROS was used for this project, was because it can communicate reliably between nodes, and work across multiple machines within a network.

5.2 Hand-Object segmentation

5.2.1 Convolutional Neural Networks

Convolutional neural network (CNN/ConvNet) is part of deep neural network family. ConvNets are mostly used for image/video processing. The structure of ConvNets can

be split into an input part, a feature learning part and a classification part, which can be seen in Figure 5.1. In the feature learning part, high level features of the image are learned by using convolutions, non-linear activation function and pooling. The classification part takes the high level features and flattens them into a single long feature vector. The feature vector is then fed into the fully connected layer with softmax function at the end of it to output an array of likelihood that the image belongs to a certain class. The architecture of a ConvNets depends on the application. Some of the well known ConvNets are AlexNet[41], LeNet[42], VGG[43], and ResNet[44].

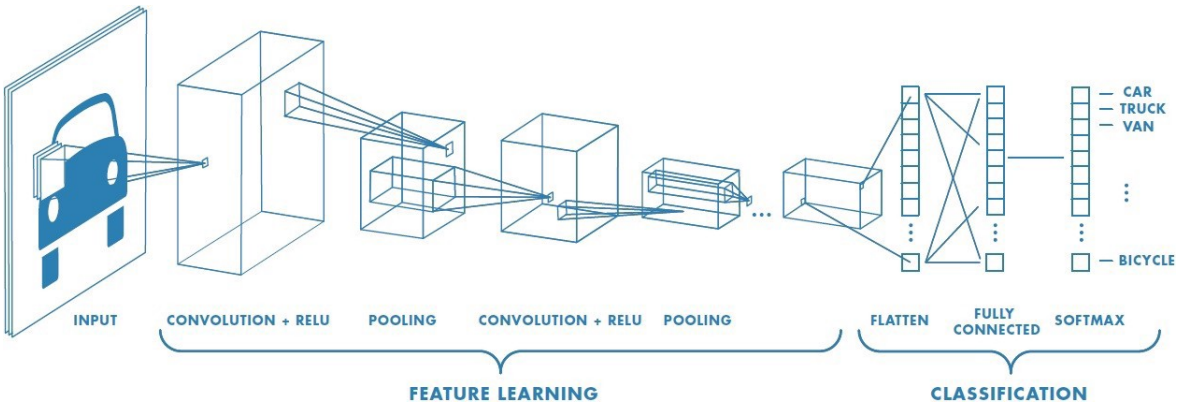


Figure 5.1: General architecture of a Convolutional Neural Network[45].

Among the ConvNets that can be used for instance segmentation are Mask R-CNN[46], FCIS[47], RetinaMask[48], and PA-net but these algorithm have greater focus on performance and lack speed. The trade off between performance and speed of the instance segmentation algorithms is shown in Figure 5.2. Performance is indicated as mean average precision (mAP) and spped as frames per second (FPS). Speed is an important factor in real-time application and "You Only Look At CoefficienTs" (YOLACT) or YOLACT++[49] are algorithms that can reach real-time speed, because of that YOLACT is chosen to be used in this project.

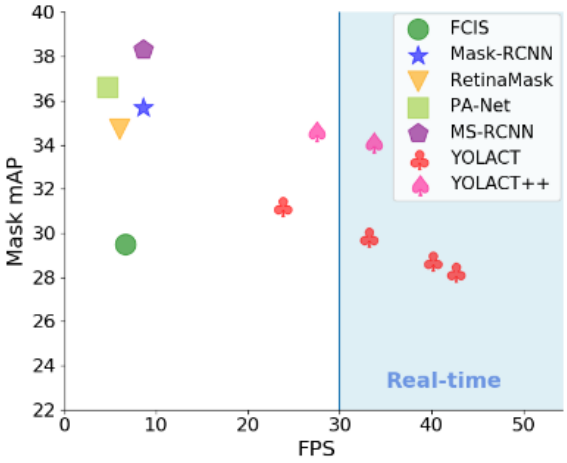


Figure 5.2: Mask mAP and FPS characteristic of various instance segmentation algorithms [36].

5.2.2 Regions of Interest

Isolating regions of interests (RoI) is used to remove noise and regularize data to speed up processing
In an RGB image, a RoI can be represented by a mask, which is a binary image where each pixel has a Boolean value, corresponding to whether or not it belongs to the RoI. The concept is illustrated by Figure 5.3.

Figure 5.3: RGB image of a person holding a can with the mask of the can of the can without the hand.

The concept of RoI is not limited to two-dimensional images. The binary pixels are then converted to points within a point cloud. An example of this can be found, by using an RGB-D camera, as it enables a mapping between the RGB data and the point cloud data.

5.2.3 Image Segmentation

When exchanging objects between two individuals, an area of the object is occluded by the person handing it over. The occluded area of the object can not be grasped by the receiving person.
With this in mind, a spacial representation of the receiving object can be made, and it can be isolated in regards to everything else around it e.g. isolating the object from the hand and the environment around the object.

Semantic Segmentation

Semantic segmentation can be used to label each pixel in an image with a class. Semantic segmentation focuses on materials with undefined shapes, which possess characteristic patterns or homogeneous colours. Figure 5.4 highlights the background, using semantic segmentation.[50]
Semantic segmentation is often used to classify the background, but it can also be used to define objects that are closer to the camera, such as occluding objects.

Figure 5.4: Example of semantic segmentation [50].

YOLACT

YOLACT is a computer vision algorithm proposed by Bolya et al. [36]. YOLACT uses a fully convolutional model for real-time instance segmentation. YOLACT uses semantic segmentation to segment and classify items, people, animals and more within the scene and can draw a bounding box around a desired class. The YOLACT process can be explained in five steps with the functionality of its nine parts. The nine parts of the architecture are shown in Figure 5.5.

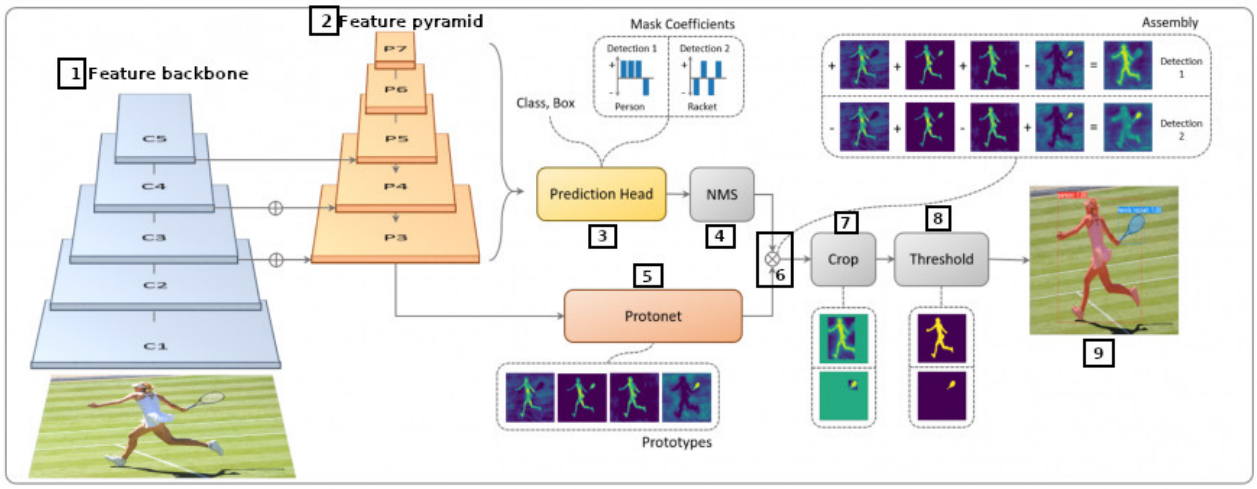


Figure 5.5: Overview of the complete YOLACT architecture. The building blocks of the architecture are labeled from one to nine. (1) is the Feature backbone, (2) is the Feature pyramid network, (3) is the prediction head, (4) is the NMS, (5) is protonet, (6) is linear combination, (7) is Crop, (8) is Threshold and (9) is the processed image. [36]

The first step in the YOLACT architecture takes in an image and feeds it into the fully convolutional neural network backbone(1).The backbone used is ResNet-101, as it deals with the vanishing gradient problem while maintaining high accuracy due to its depth (101 convolutional layers). Furthermore, ResNet-101 is inherently translation variant, which is relied on heavily by YOLACT.
The second step involves the Feature Pyramid Network(2) (FPN). The FPN process features deeper (from C3 and up) in the backbone to produce more robust masks and better performance on smaller objects.
The third step includes part (3),(4) and (5). Protonet makes a guess as to how the mask looks like, by making multiple prototype masks based on the features from the FPN. In parallel with protonet generating masks, prediction head predicts class, bounding box and mask coefficients for each instance of the FPN. After the prediction non-maximum suppression is applied to remove overlap of bounding boxes.
In the fourth step part (6) takes the result from part (4) and (5) linearly combines the prototype masks with the mask coefficient to predict instance masks. In the last step the different instances, in the instance mask are cropped(7) and thresholded(8), the mask are then applied to the image and the final result is shown in part (9).

5.3 Object shape recognition

5.3.1 Point Cloud

When a potential object is selected, it is necessary to identify the shape, in order to plan a grasping strategy. A shape can be represented by a set of points in a 3D space called **point cloud**. [51]. A teapot, for instance can be represented by a point cloud and its points can be rendered such as in Figure5.6.

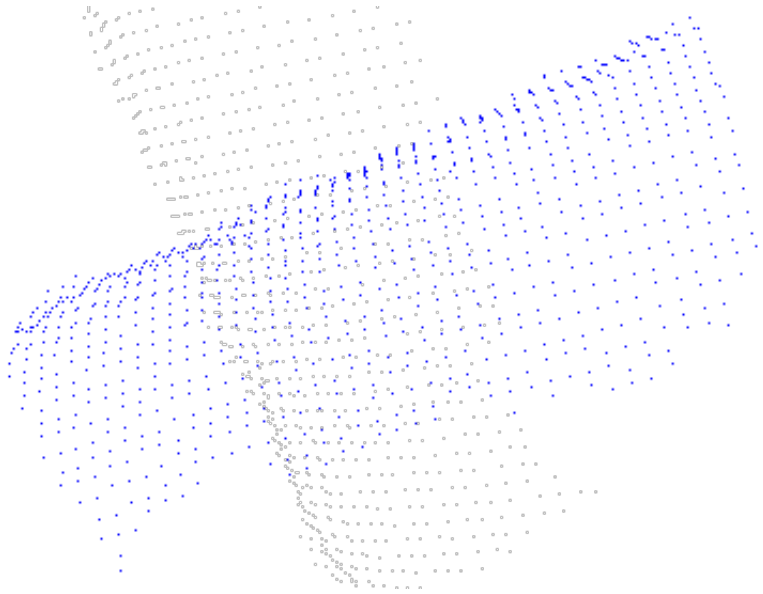


Figure 5.6: Visualization of point cloud. [51]

There is a library called Point Cloud Library (PCL) [52], which can run on C++ and has multiple algorithms to process point clouds. For example, it can be used to create, segment and filter of point clouds. For this project, PCL was chosen to process the point cloud and determining the object characteristics.

5.3.2 Object shape recognition

The way objects are grasped depends on their shape, dimension and orientation. Most machine and human made objects can be simplified to represent geometric primitives such as spheres, cylinders or a combination of these primitives[53]. Based on the fact that different objects can be represented by their geometric primitives, it is possible to group different objects based on this. The following subsection will look into different methods that can be used to define the shape of an object.

RANdom SAmple Consensus(RANSAC)

A work presented by Fischler and Bolles[37], introduces a method called Random Sample Consenus(RANSAC). RANSAC can be used when processing images modelling tool. This algorithm can be used on point cloud data to find the desired model. RANSAC works by randomly selecting a certain amount of points in the data and a minimum amount of points needed to represent the model. After this, every point in the data set is tested against a model to see which are inliers and which are outliers to the model. The previous step is then repeated for a set amount of iterations, where the model with the most inliers will be picked or until a goal amount of inliers has been reached. An example of RANSAC on a point cloud dataset can be seen in Figure 5.7. PCL also includes the function to use RANSAC in point clouds.

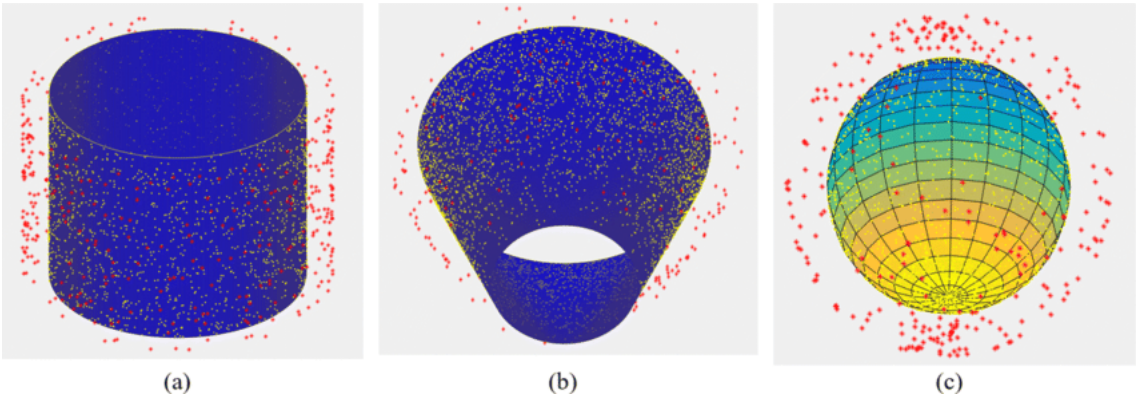


Figure 5.7: Example of RANSAC trying to fit a 3D model to a point cloud. The red points are from a captured point cloud, while the geometrical objects are RANSAC models.[54]

Principle Component Analysis (PCA)

Principle Component Analysis or PCA is a dimensionality reduction method, it can bring out strong patterns in a dataset by focusing on variation. PCA can be used to reduce the dimensionality of data while maintaining the most amount of the information. It can be applied for visualization purposes, since data higher than three dimensions can not be visualized. It can also be used to filter out noise from data, removing dependent data which is redundant information. Lastly, reducing dimensions with PCA can have the benefit of decreasing the amount of processing time since a large amount of dimensions can slow it down.

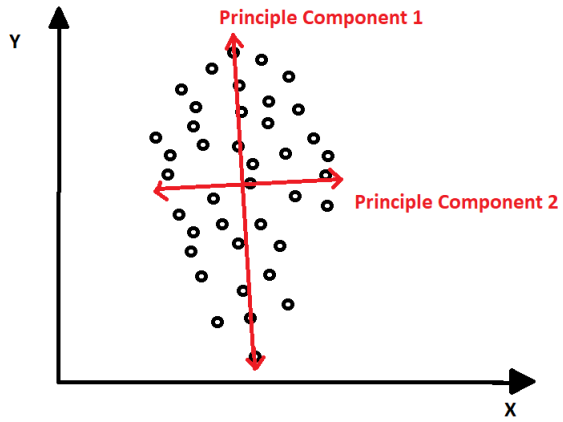


Figure 5.8: Example of Principle components on scattered data points in 2D space. The x and y-axis represent different features that differentiate the data points.

Using linear combinations, PCA can reduce the dimensionality by creating new features using the already existing ones. These new features, called principle components, will contain the most amount of data. An example of how these principle components look like can be seen on 5.8. A good feature can be defined to have high variance and have a small reconstruction error.

To perform PCA, the data should always first be centered around zero. When the data has been normalized, the covariance matrix can be calculated. The covariance matrix shows the correlation between the different features. Low correlation ensures good features, since this means there is less redundancy. When the covariance matrix is found, the eigenvectors and their corresponding eigenvalues can be computed. The eigenvectors will give information about the direction of the axes where the most amount of variance lies, this is what was mentioned earlier as the principle component. The eigenvalues show the actual amount of variance of each of the principle component. By selecting the highest eigenvalues, the principle components with the highest variance or significance can be found. The most amount of data will always be put in the first principle component that PCA finds. This method will be used later for size estimation of the models.

5.4 Hardware

This section will go through the different hardware used for the implementation and testing of the system.

5.4.1 Intel Realsense d435

The camera used for this project is the Intel RealSense D435 [55]. It is a stereo vision depth camera with USB-C connection. Version D435 is distinct with an Intel Realsense Vision processor D4, Wide Stereo Imagers, Wide Infrared Projector and RGB color sensor. This version has these minimum system requirements to be operated: USB 2.0/USB3.1 Gen 1, Ubuntu 16.04/Windows10 and newer. This camera features up to 1280x720 stereo depth resolution, up to 1920x1080 RGB resolution with 30FPS RGB camera, Depth diagonal Field of View(FOV) over 90°, Dual global shutter sensors for up to 90 FPS Depth streaming and range from 0,2m to over 10m dependent on lighting conditions[56]. The camera with tripod attached can be seen in Figure 5.9.

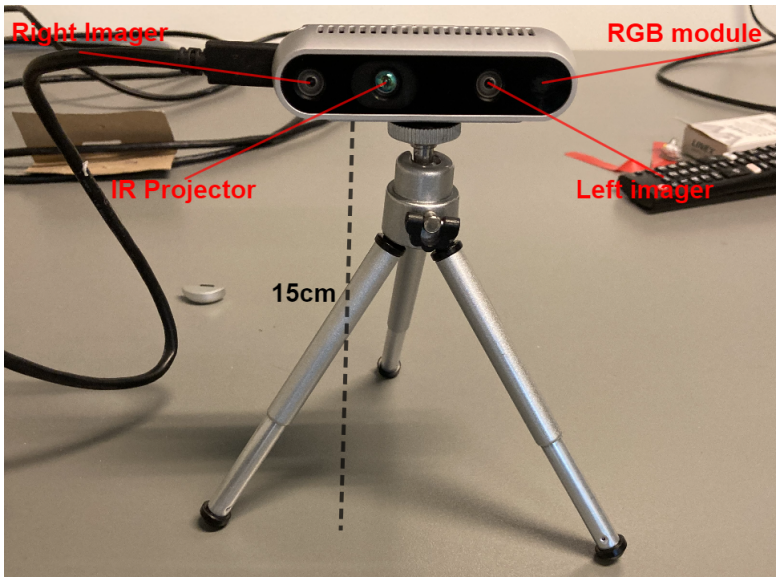


Figure 5.9: Image of the Intel Realsense d435 camera on the tripod stand included. The right and left imager, IR projector, RGB module and the height of the stand being highlighted.

5.4.2 Computer specs

The project uses a laptop with the following specifications. Operating system: Ubuntu 20.04.4 LTS, processor AMD Ryzen 5 4600H (3,0 GHz), RAM 16 GB DDR4-3200MHz (SODIMM), graphics card NVIDIA GeForce GTX 1650 Ti connected to the camera with a 1m long USB 3.1 extension cable made by COXOC.

Chapter 6

Implementation

This chapter details the implementation process of various parts used in this project. First a complete system overview is given. Afterwards, the implementation of hand object segmentation is explained. Next, the implementation process of object dimension and shape recognition is elaborated. ROS is used in this project to communicate across hardware and software. One of the things ROS is used for, is to get data from the camera_node, alter the data, and publish it back to camera_node to display the changes.

6.1 System Overview

Figure 6.1 shows an implementation of the system. The Realsense D435 camera publishes RGB and depth images at 5 frames per second (fps). These images are coordinated both chronologically and pixel-wise in 640x480 resolution for both depth and RGB. The 5 fps rate was chosen, in order to obtain a system which is both responsive and computationally light.

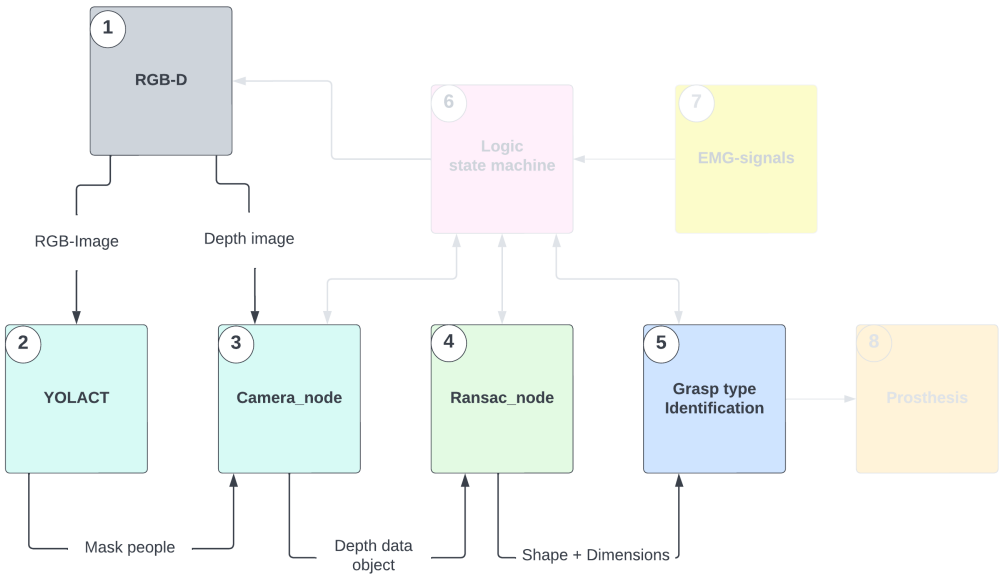


Figure 6.1: This flowchart shows the processes which were implemented in this project. Note that some of the processes which were shown in the Design of system 4 diagram are grayed out since they will not be implemented. The reason that they are not implemented is due to that fact the focus of the current project is image processing.

6.2 Hand object segmentation

Realsense ROS package The ROS Realsense package allows for integrating the functionality of the Intel Realsense D435 camera, to the ROS environment. The camera topics that are subscribed to are:

- /camera/aligned_depth_to_color/image_raw
- /camera/color/image_raw

The */camera/color/image_raw* topic gives the RGB image of the camera. The topic is subscribed to by YOLACT, to check whether there is a person or not in the frame. The */camera/aligned_depth_to_color/image_raw* topic give a depth image of the camera. The topic is also used by YOLACT to segment out the detected person.

6.2.1 yolact_ros and yolact_ros_msgs

For each RGB image frame received by *yolact_ros*, a topic called */yolact_ros/detections* is published, containing a set of four object detections, this may be empty if the system detects no objects.

Each detection contains four variables: **class_name**, **score**, **box** and **mask**. The **class_name** is a string which contains the name of the class. The **score** represents a percentage of certainty about the class name. The **box** describes the bounding box. The **mask** contains an unsigned 8-bit integer buffer, a width and height.
To reduce the computations, the file *yolact_ros/scripts/yolact/layers/output_utils.py* was changed to prevent further processing of any detection belonging to a non-**person** class.

6.2.2 camera_node

The overall workflow of the camera_node node can be seen in Figure6.2.
This node subscribes to the topic *camera/aligned_depth_to_color/image_raw*, which contains a **depth map** and saves it to a global variable called depth.

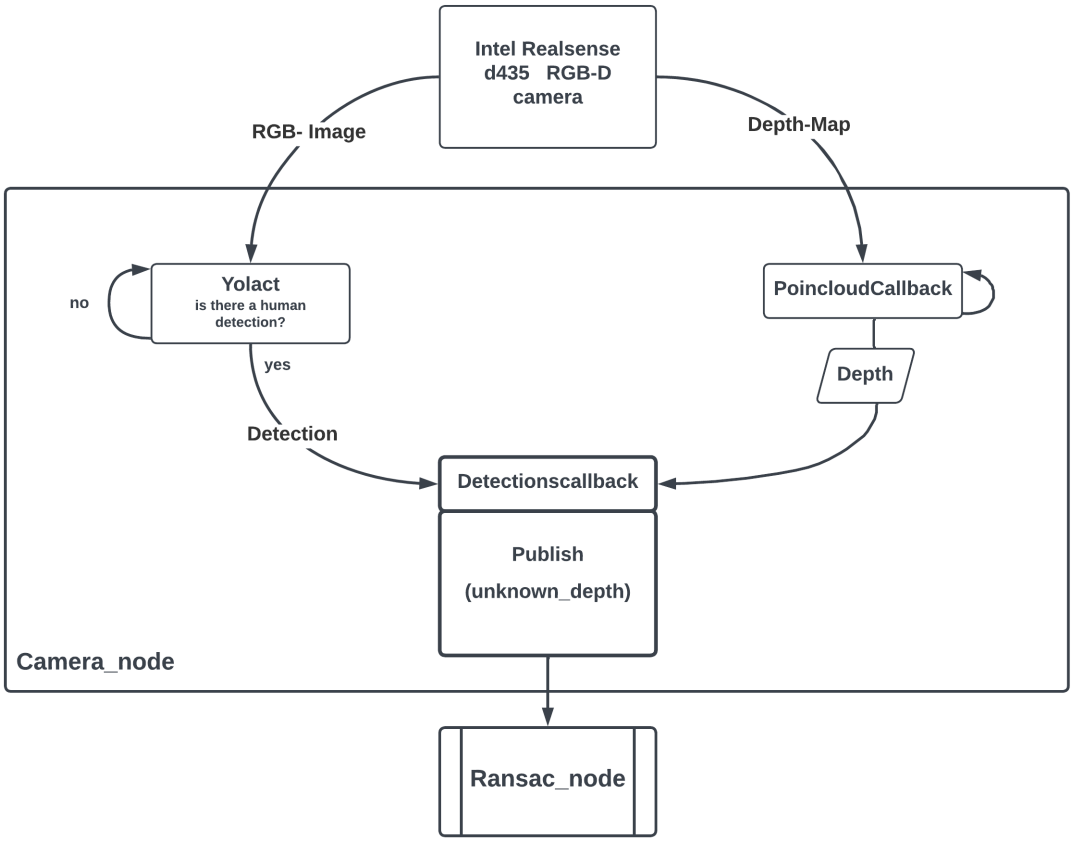


Figure 6.2: Camera_node workflow

The node also subscribes to the topic */yolact_ros/detections* described in 6.2.1. The subscriber calls a function that compares the depth map with the masks from the detections,

turning every depth map pixel into zero, if one of two conditions were fulfilled: there is a non-zero value in the mask or the depth is outside the range between 20.00*cm* and 40.00*cm*.

Figure 6.3:
Left: RGB image of an object being held within the threshold depth.
Right: Depth map coordinated with the RGB image.
Bottom: Depth map of isolated object.

The resulting depth map ideally contains the object being held, a visual example of the process cont of this can be seen in Figure6.3. This is depth map is publish in the topic *unknown_depth*.
The result is a depth image of an object containing nothing but the object of interest without the hands being seen, which is then sent to the next node, the ransac_node.

In the top left Figure 6.3, it can be seen the coordinated RGB and depth image pair. In the top left you can see only depth image with all information present. On the bottom of the Figure 6.3, the filtered depth image of only the object can be seen which is the product sent from camera_node to the ransac_node.

6.3 Object dimension and shape recognition

This section describes the implementation of the ransac_node, created to find the appropriate grasp type, based on the point cloud received from the camera_node.

6.3.1 ransac_node

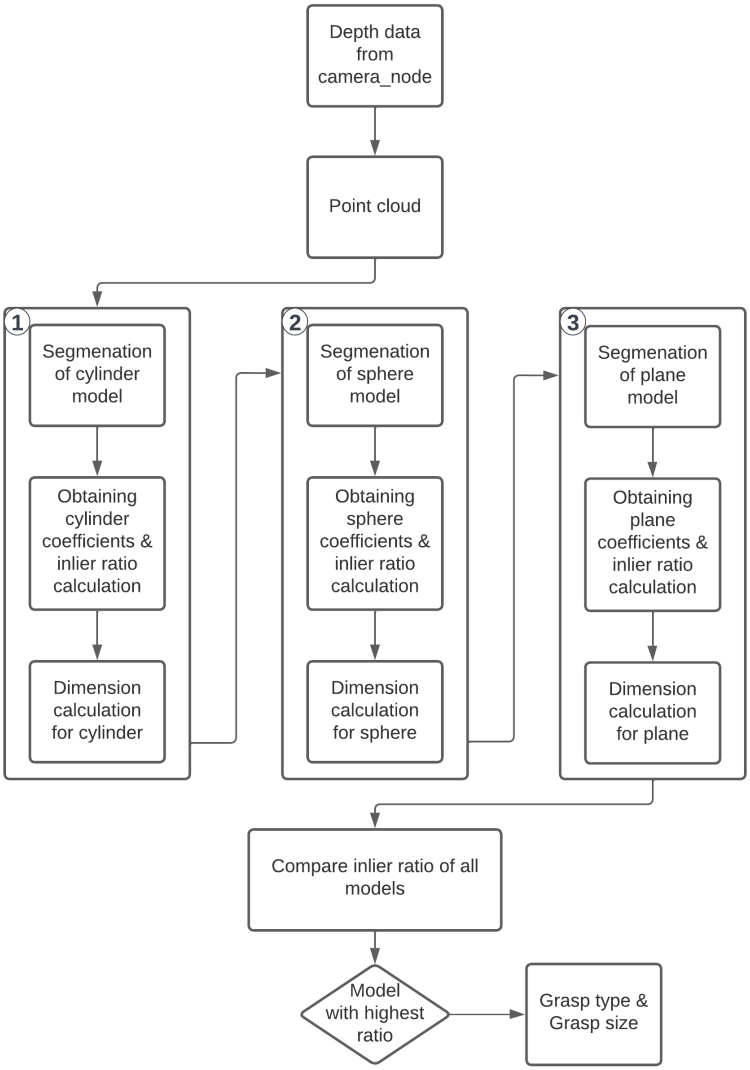


Figure 6.4: Ransac_node process.

Segmentation by shape fitting

After receiving the depth data from the camera_node, the data is transformed into a point cloud of the isolated object. Next, the point cloud is segmented. The segmentation is done to gather points located near each other, to form clusters in the point cloud. The clusters are used for further processing. After the clustering is done, three different RANSAC models are fitted sequentially, from the center of the cluster. The models are expanded until it is no longer possible to locate a point within a specified distance from the model. Said distance being dependent on the models in use. The three models are a cylinder and a sphere. Additionally, a plane which is combined with other planes of the point cloud to form a cuboid, due to the PCL not having a cuboid segmentation model.

Point Extraction

After the RANSAC models have been fitted onto the point cloud cluster, the outliers of the fitted models are removed. The outliers are accounted for in the process to get the total amount of points. The inliers and outliers are used to get the ratio of points included inside the model. This process is identical for the cylinder and sphere model, but requires up to three loops for the plane model, because multiple planes cannot be fitted at the same time.

Cylinder and sphere

The points located in the cylinder and sphere shape are obtained in order to acquire the coefficients of the shape. The outliers of the shape are removed, to only have the inliers remaining. Thus the coefficients for the model are computed. The seven coefficients representing the cylinder are the radius, three coordinates of a point in the longitudinal axis and three coordinates of the vector representing the direction of the longitudinal axis for the cylinder. The four coefficients for the sphere are the coordinates of the center point, and the radius of the sphere. These coefficients are used to calculate the dimensions of the shapes.

Cuboid

As mentioned, PCL does not contain a RANSAC model for a cuboid, thus multiple planes are utilized instead. Depending on the angle of which the object is seen by the camera, one to three planes are capable of being created. A while loop is used to construct a plane onto a point cloud, and continues until there are less than 30% of the original points in the point cloud. In order to check if it is possible for more planes to be fitted to a point cloud, it is verified if there is enough points left, to construct a new plane. Meaning that at least 30% of the original points from the point cloud, has to remain. Once the first plane has been fitted, the inliers of the plane are removed, to ensure the same plane is not fitted

twice. The planes that have been fitted their outliers are removed, which also happens to the cylinder and sphere.

Ratio

After each process is done, the total amount of points are divided by the amount of inliers, and then multiplied by 100, to obtain a percentage. The ratio for the cuboid is obtained by taking the total amount of inliers for all three planes, and add those together as the total amount of inliers. These three ratios are then be compared to one another, and the highest ratio determines the shape of the point cloud.

6.3.2 Size estimation

To correctly identify which grasp type should be selected, the size of the objects should be known. The following sections describe how the size estimation of each model was done. The parametric models for the sphere and the cylinder contain coefficients which has information about the radius of each model, however cuboids are not therefore PCA is used to estimate the dimensions.

Sphere

The parametric model of a sphere is already supported, it is described with four coefficients with the PCL library, namely:

- Radius r of the sphere
- (X,Y,Z) coordinates of the center P of the sphere

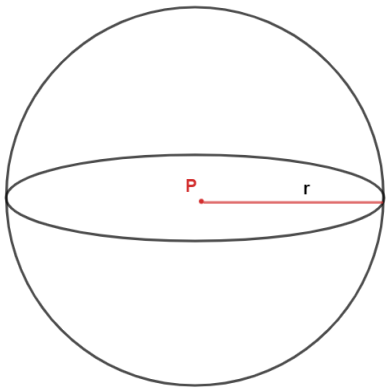


Figure 6.5: Figure of a sphere, showing the center coordinate P and the radius r.

The radius and center point can be seen in Figure 6.5. The four coefficients are given, hence no further calculations are needed to calculate the dimensions of the sphere. The radius is then multiplied by two.

$$D = 2r \tag{6.1}$$

Cylinder

The cylinder has seven coefficients, there is no specified height in its parametric model. The height of the cylinder can be seen as an infinite line along the longitudinal axis of the cylinder. The seven coefficients of the parametric model are:

- Radius r of the cylinder

X,Y,Z Unit vector ($\vec{u}$) along the longitudinal axis

- (X,Y,Z) coordinates of a point P on the longitudinal axis of the cylinder

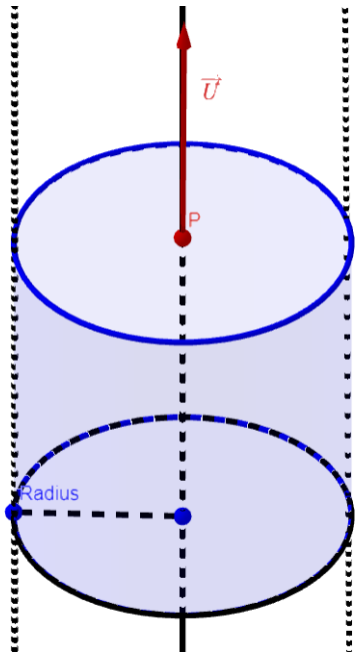


Figure 6.6: Figure of a cylinder, showing radius r of the cylinder, unit vector ($\vec{u}$) and point P on the longitudinal axis.

A Figure of the parametric model of the cylinder is seen in Figure 6.6.

The height h of the cylinder is calculated by using the x,y and z components of the directional vector.

$$h = \sqrt{x^2 + y^2 + z^2} \tag{6.2}$$

Cuboid

In order to calculate the dimensions of the cuboid, Principal Component Analysis (PCA) was used on each of the detected planes. The covariance matrix was found for each of the plane's point clouds, which contains the inliers for each of the planes. After the covariance matrices has been found, the eigenvectors, and corresponding eigenvalues are then calculated, in order to determine the x, y and z dimensions of the planes. Using these, it was possible to calculate the height, depth and width of the cuboid RANSAC model.

6.3.3 Grasp type identification

After the geometrical shape and dimensions of the shape has been determined, the identification of the grasp type is initialized. A decision tree has been made, to determine the grasp type out of seven options. A diagram of the decision tree is found in Figure 6.7. The object is sorted into a category, depending on the geometrical shape. Once this has been decided, the diameter/width of the object determines which one of two grasps are most fitting. Additionally, the height of the object is used, to find additional directions to grasp the object, in case the individual giving the object is fully blocking the horizontal dimension of the object. If the object is too tall, it is assumed an area above the givers hand is available for grasping.

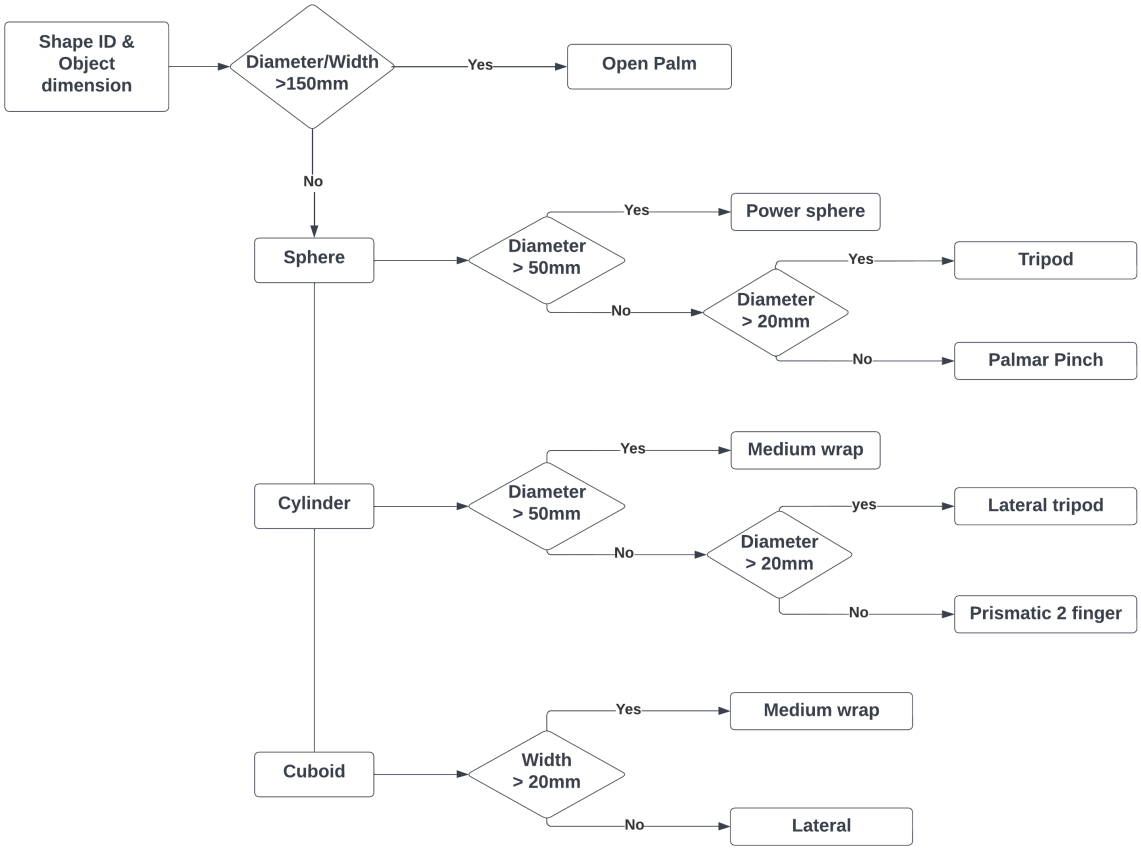


Figure 6.7: This flowchart depicts the grasp type selection process. The prosthesis would preform to make the hand completely open when it encounters an object with dimensions too large to grasp.

6.3.4 Implementation conclusion

This chapter went through the two nodes used for the system, and how the different methods were implemented. The final system, along with a video demonstration can be found in the following github link: [57]

Chapter 7

Testing

This chapter attempts to validate the solution presented in this project by testing the different requirements presented in 3. For this goal, a sets of tests will be performed on the system and the results will be presented and then summarized in the end of the chapter. All the additional test information can be found in the github.

7.1 Test Setup

Nine objects are chosen as graspable objects, for the testing. Among those, three have a spherical shape, three have a cylindrical shape and the last three have cuboid shapes. A visual overview of the nine objects can be found in Figure 7.1, and a table of the object dimensions, as well as the designated appropriate grasp type chosen, can be found in Table 7.1 .

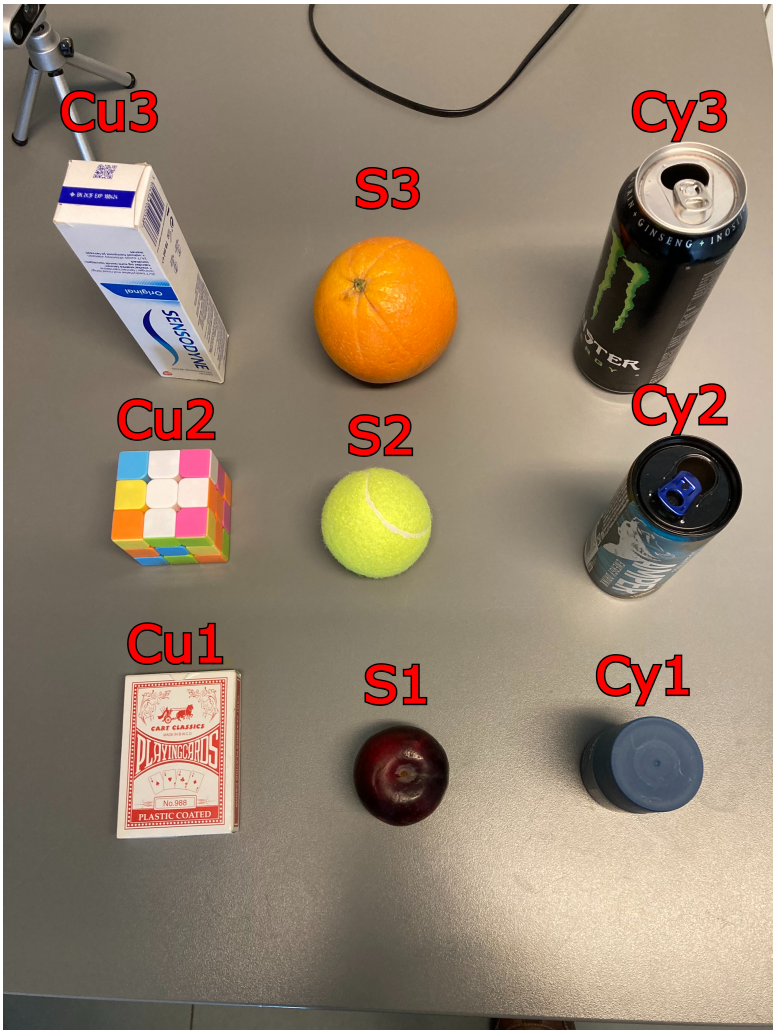


Figure 7.1: The testing objects used for this project. The captions above indicating the object name. The letter being the shape, and the number indicating the volume. For example, Cu2, is Cuboid number two.

	Diameter/Width	Height	Depth	Appropriate Grasp Type
Cylinder 1 (Cy1)	45mm	85mm	45mm	Lateral tripod
Cylinder 2 (Cy2)	50mm	135mm	50mm	Medium wrap
Cylinder 3 (Cy3)	65mm	170mm	65mm	Medium wrap
Sphere 1 (S1)	50mm	50mm	50mm	Tripod
Sphere 2 (S2)	65mm	65mm	65mm	Power sphere
Sphere 3 (S3)	86mm	86mm	86mm	Power sphere
Cuboid 1 (Cu1)	19mm	90mm	65mm	Lateral
Cuboid 2 (Cu2)	56mm	56mm	56mm	Medium wrap
Cuboid 3 (Cu3)	45mm	170mm	38mm	Medium wrap

Table 7.1: Dimensions as well as determined appropriate grasp types for the specified object

The objects will be held by a person at a distance between 200mm to 400mm away from camera. The camera is placed statically on the tripod. The hand holding the object will have different grasp types.The holder's hand will in some of the tests be partially cover the objects, and have different grasp types, to verify the robustness of the segmentation and point cloud reconstruction.

7.2 Human hand segmentation

This test will verify if the system is capable of segmenting the hand when holding an object, described in requirement 1. Furthermore, enough of the object held should be reconstructed in a point cloud, saved to a PCD file format. A total of 90 tests were performed, ten for each of the nine objects. The objects were held in using different grasps, where the hand was in different positions on the object. The objects were also held in various positions in the frame of the camera.
Since the nodes are tested separately, the generated point clouds will be judged through visual inspection for if the point clouds are detailed enough to recognize the designated object shape. This means, that a point cloud of a cylinder must have a curvature, a sphere must have a partially spherical curve, and a cuboid must contain one or multiple planes. Background object accidentally picked up, and noise will be ignored for this test, since they do not affect the desired point cloud object.

The point clouds are judged in one of four categories. Correct Point Cloud (CPC), when a point cloud is generated and can be identified as the correct shape. Wrong Point Cloud (WPC), a point cloud is created, but does not resemble the correct object shape. No Point

Cloud (NPC), when the node fails to create a point cloud of the test object. Hand Point Cloud (HPC), when the hand holding the object is significantly represented as part of the point cloud. Figure 7.2 shows the results from the testing.

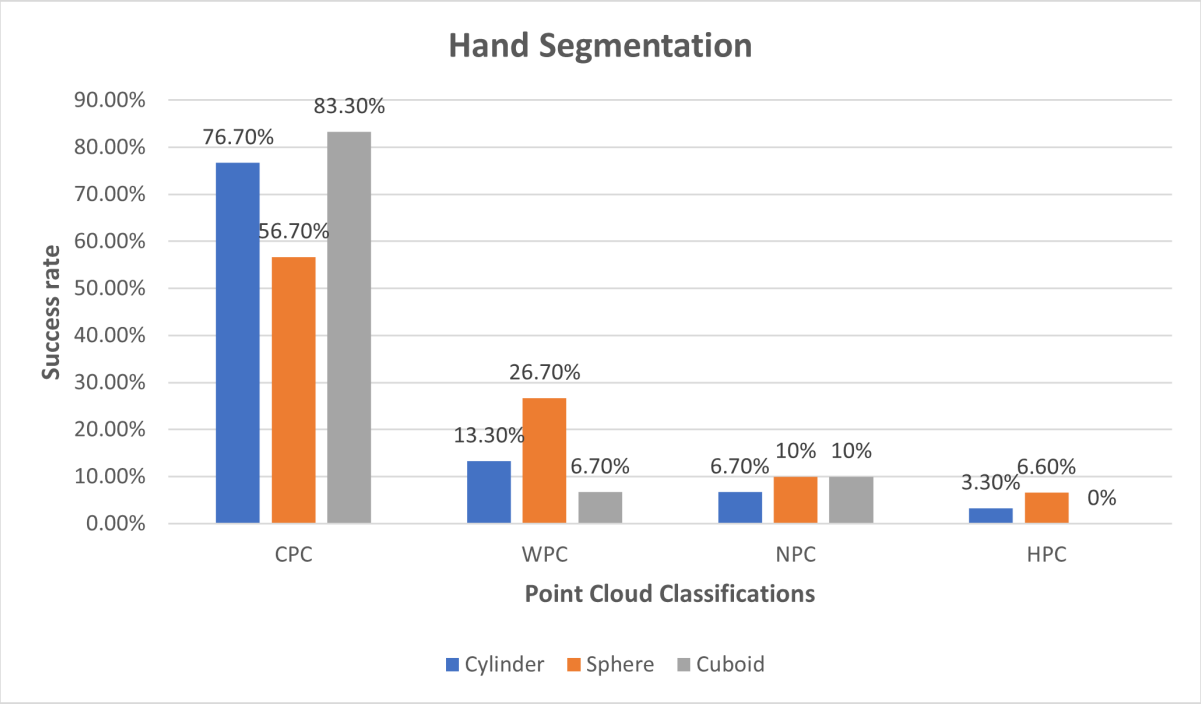


Figure 7.2: CPC= Correct Point Cloud | **WPC=** Wrong Point Cloud | **NPC=** No Point Cloud | **HPC=** Hand Point Cloud

After the testing, the three cylinders had a CPC rate of 76.6%, and the spheres had a CPC rate of 56.6% and lastly the cuboid with a CPC rate of 83.3%.

7.3 Shape identification

This test will demonstrate the capabilities of the system to identify the shape of the object, by fitting a RANSAC model. This test is based on requirement 2 and requirement 3.
Before initializing the testing, the parameters of the three segmentation models were modified, using a point cloud of each of the nine test objects, without any occlusion from a hand. Next, the point clouds from the previous tests which were saved as PCD files, were used to verify the robustness of the classification, when working with point clouds with different levels of occlusion. The 90 tests were performed, with some extra being added to accommodate for the eight missing point clouds from the previous test. A test was then performed to observe the ransac_node's capability to fit the correct model onto the point cloud, thereby estimate the shape of the object. The results can be found in Figure 7.3

represent in a pillar diagram.

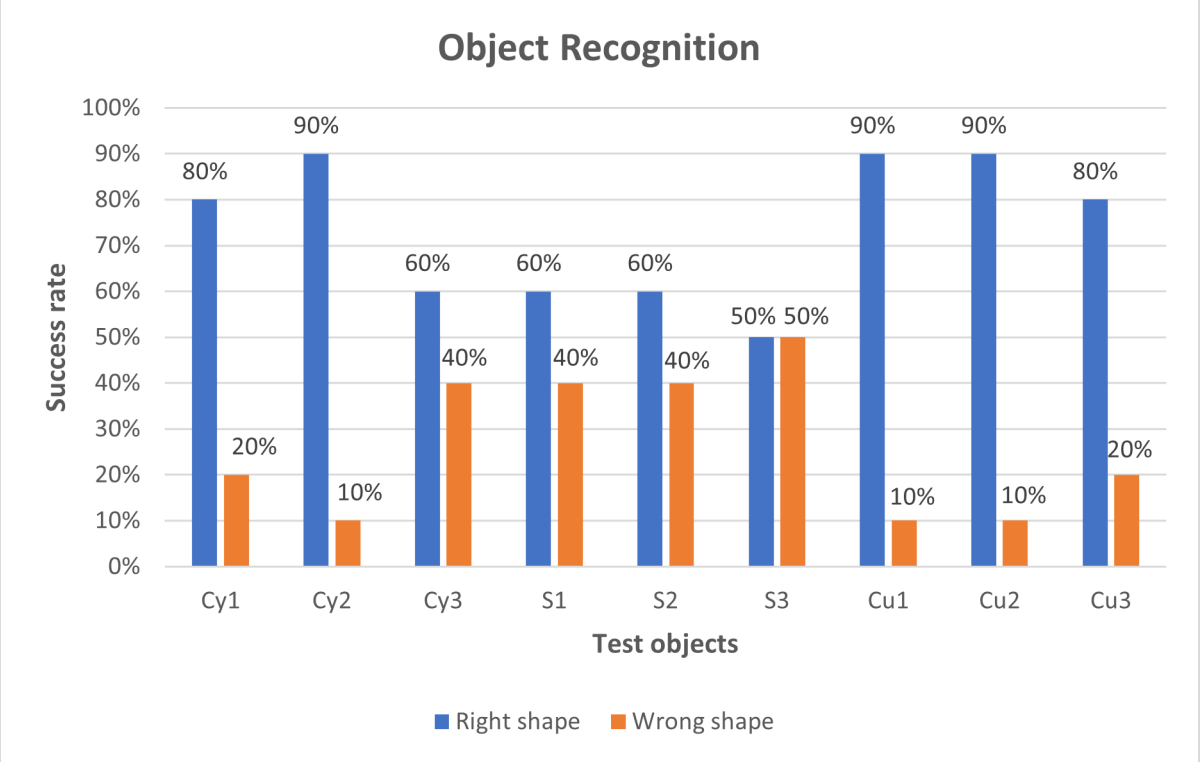


Figure 7.3: Pillar diagram over the shape identification results, for the three Cylinders (Cy), three Spheres (S) and three Cuboids (Cu)

As can be seen in the figure, the cylinders had an overall success rate of correct shape identification on 76.7%, the spheres had 56.7%, and the cuboids had 86.7%. A table of the mean success rate and mean shape ratio can be seen on Table 7.2.

	Success rate	Mean shape ratio
Cy1	80%	98.6%
Cy2	90%	89.0%
Cy3	60%	66.0%
S1	60%	96.2%
S2	60%	89.9%
S3	50%	77.03%
Cu1	90%	88.92%
Cu2	90%	83.19%
Cu3	80%	97.2%

Table 7.2: Table summarising the tests done for the shape identification, containing individual object success rate and mean shape ratio

7.4 Shape dimensions

In order to identify the grasp type meant for the object, the dimensions of it must be known. This test aims to test the systems ability to calculate the dimensions of the test objects, when handed over by a person. This test will verify whether or not the system passes requirement 4.

During the testing, the camera parameters not being set up correctly to accurately estimate the dimensions, it was not possible to get an accurate estimation of the size for the objects in the point cloud. The PCA calculation were not possible to get fully integrated as well. Though, when ran, the RANSAC node would provide the coefficients of each RANSAC model, as well as the rough calculations from the PCA implementation, which can be seen in Table 7.3 .

	Diameter /width
Cy1	112.3mm
Cy2	140.4mm
Cy3	151.2mm
S1	132.2mm
S2	170.3mm
S3	199.6mm
Cu1	11930.6mm
Cu2	42070.2mm
Cu3	33014.6mm

Table 7.3: Table showcasing the results of the shape dimensions test, in the form of the average radius/width of the recognised objects.

7.5 Grasp identification

The final step of the identification process is identifying the appropriate grasp for the object. This test will verify whether or not the the system passes requirement 5 in identifying the grasp type for the object, and whether or not it matches with the designated grasp type assigned the individual objects. Secondly, requirement 6 will also be verified, by measuring the time it takes for the system to estimate the grasp type, from when the operation has been initiated, to the grasp identification process has ended.
Since the results from the previous test were not accurate enough, it was not possible to accurately test and determine the appropriate shape for each of the objects.

Chapter 8

Discussion

8.1 Human hand segmentation

As mentioned in chapter 7.2, the three shapes had a CPC rate on 76.7% 56.7% and 83.3%, and an overall success rate on 72%, which is considered a success according to the requirements. The WPC classification, which had a total classification in 15% of the results, along with the NPC classification with 8.8%, might be due to the hand blocking the object too much, leading to an incomplete and small point cloud. This was not as much of an issue for the cuboids, which could be due to it generally being more difficult to gain a full grasp around the objects, in comparison to the spheres, which could more easily be blocked by the fingers of the hand. Additionally, reflection from the objects could also have been a benefactor to the incorrect point clouds. Lastly, an occurrence which appeared in a total of three tests, where the fingers holding the objects getting recognised as part of the point cloud. When observing the RGB image of the point cloud, it can be seen that the fingers are sticking more out, not being completely wrapped around the object. This can lead to an angle of the hand, which the YOLACT can not recognise.
Furthermore, a dragging effect was observed, while the object was moving fast. It hypothesised that that there is lag between the masks from the CNN and the depth map. This can be solved by pairing only masks originated from images with the same time stamp as the depth map.

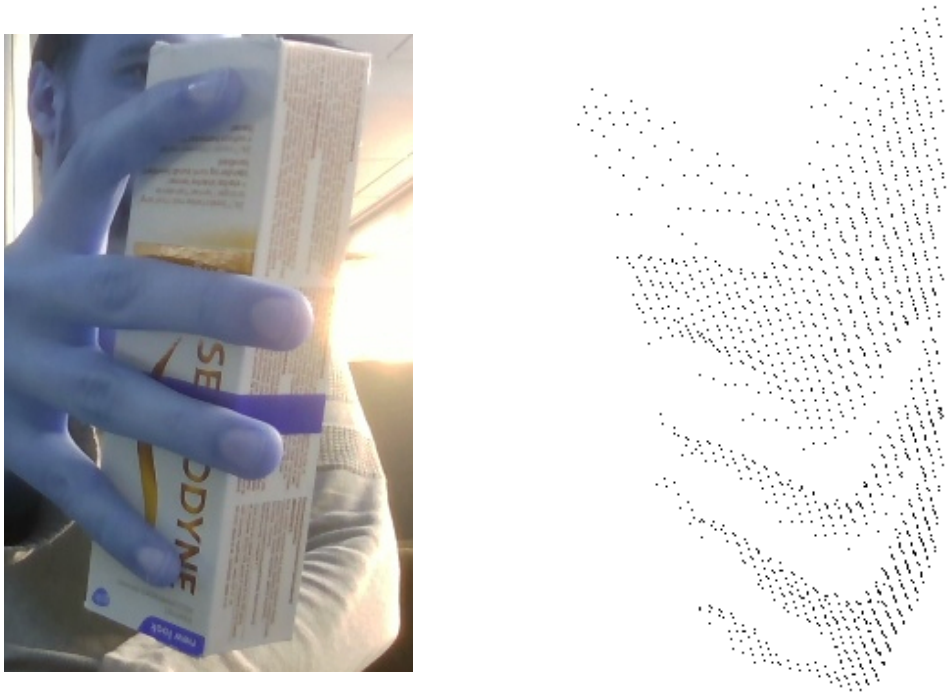


Figure 8.1: RGB image from YOLACT with resulting successful Point Cloud

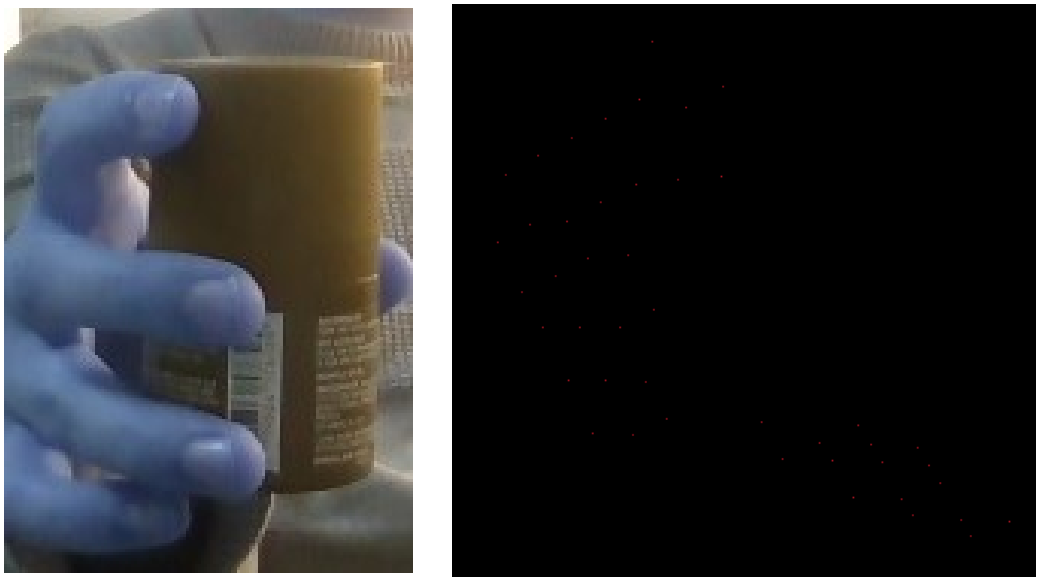


Figure 8.2: RGB image from YOLACT with resulting failed Point Cloud. The point cloud containing very few points. Too few to determine a shape.

8.2 Shape identification

The results from testing 7.3 showed a strong ability for the ransac_node to recognize the points clouds obtained from the camera_node, with non of the objects error rate exceeding the success rate. Only the S3 ended up having an error rate equal to the success rate. The spheres showed the overall poorest performance out of the three shapes. This was mostly due to a problem with the ransac sphere segmentation, causing no inliers to be found in the model. This could be fixed for future development, by altering the parameters for the sphere segmentation. Furthermore, errors in the point cloud itself would also cause shapes to be wrongly classified as a different shape. Most prominently, the error was because of a lack of points in the point cloud of the object. Additionally, the reflective surface of C3 caused the shape to be wrongly identified as a cuboid on four of the tests. However, the overall performance showcases positive results, that is capable of identifying object shapes more accurately than when human inspection analyses the point cloud, with a total success rate of 73%, over the 72% from the hand segmentation test.

8.2.1 Shape dimensions

As mentioned, the camera parameters were not estimated correctly during the testing, meaning the dimensions of the point cloud objects were not similar to the real life counterparts. However, the result dimensions obtained from the current implementation showed a mostly consistent output, meaning the system is capable of identifying dimensions of an object on a stable basis.

8.2.2 Shape identification

Since it was not possible to get the accurate radius/width of the test objects, it was not possible to test the grasp identification. Altough, since the decision tree is only dependant on the shape identification and shape radius/width, it can be assumed that the system would be capable of identifying the appropriate grasp for an object, given the two variables are accurate.

8.3 Future work

As mentioned in the delimitation of system section in the Design 4, some of the processes were not implemented in the current project. To limit the scope of the project, there was a focus on only implementing the person recognition an object recognition without the integration on a prosthesis.

Further development for this project would be to implement a working logic state machine, capable of controlling the separate parts in the overall pipeline. The logic state machine,

as explained before, would be able to be activate the processes whenever the previous processes are completed. This includes the activation of the camera by an EMG-signal input, and the control of the prosthesis to select the appropriate grasp type.

Another aspect to focus on for future work, would be social gestures. Social gestures like handshake, waving, and fist bumping is common way to greet in everyday life. In the context of this report, further development could include the ability to perform such social gestures. This could be implemented by recognizing the specific social gesture, such as a handshake, a wave or a fist bump, and then make the prosthesis replicate this action. With modern computer vision techniques, hand shape recognition is made possible and implemented in areas such as robotics [58], virtual reality [59], augmented reality [60] and prosthetic enhanced control. In prosthetic control, a hand holding an object can be separated from it using these techniques. This leads to better object classification, which in turn gives more precise measurements for model fitting and thus overall amplified control of prosthetic. The vision techniques used for detecting the hand and its shape is reviewed by Munir et al. [61], with explicit advantages and disadvantages of these systems listed.

Chapter 9

Conclusion

The scope of this project was to create a computer vision system to enhance the current prosthesis control. More specifically the focus was made on the act of grasping objects from another persons hand. This was the main novelty of this work, to find out which methods can be used to improve on this aspect in the field of semi-autonomous prosthesis control.

To get a better grasp of the overall problem, a detailed research of the current struggles amputee patients face has been made in the problem analysis chapter. Furthermore some of the shortcomings of current prosthetics has been mentioned. It was found that the majority of the current commercial prostheses were controlled using myoelectric control. Besides direct myoelectric control, other methods of control for myoelectric prostheses was explored. Methods such as signal pattern recognition and regression based control, have shown promising results regarding the improvement in control of prostheses. It was found that while these methods alleviated some of the burden for amputee patients, it was still lacking in regards to the control over the prosthesis. Findings were made that the addition of an additional sensor such as a camera could prove beneficial to enhance the control of a prosthesis. Works relating to the semi-autonomous control of prosthesis with the use of computer vision have been mentioned. The findings were that most of these works focus on object grasping from freestanding objects, such as objects placed on a table.

The final problem formulation asked: *How can a computer vision system be used to automate the grasp type selection and size for a transradial prosthesis when receiving an object from a persons hand?*

The system also proved capable of estimating consistent measurements to a varied set of point clouds of the same object and determine the correct shape using at least 66% of the points.
Despite multiple grasps being used to grab the testing object where the hand covered large

portions of it, as well as different positions in the frame, the results show that it is possible to visually isolate an object with mean success ration of 72%.

With further work, it will be possible to increase the accuracy of the shape classification as well as dimension estimations.
The results show that the system can successfully be used to automate the grasp type selection and size for a transradial prosthesis when receiving an object from a persons hand. Therefore, the project was considered a success.

Bibliography

[1] Aimee Cloutier and James Yang. "Control of hand prostheses: A literature review". In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 55935. American Society of Mechanical Engineers. 2013, V06AT07A016.

[2] Kerstin Hagberg and R Brånemark. "Consequences of non-vascular trans-femoral amputation: A survey of quality of life, prosthetic use and problems". In: *Prosthetics and orthotics international* 25.3 (2001), pp. 186–194.

[3] Debalina Datta, Pratyay Pratim Datta, Kunal Kanti Majumdar, et al. "Role of social interaction on quality of life". In: *National Journal of Medical Research* 5.4 (2015), pp. 290–292.

[4] Lynn Bloomberg, James Meyers, and Marc T Braverman. "The importance of social interaction: a new perspective on social epidemiology, social risk factors, and health". In: *Health Education Quarterly* 21.4 (1994), pp. 447–463.

[5] Caroline C Nielsen. "A survey of amputees: functional level and life satisfaction, information needs, and the prosthetist's role". In: *JPO: Journal of Prosthetics and Orthotics* 3.3 (1991), pp. 125–129.

[6] Strahinja Došen et al. "Cognitive vision system for control of dexterous prosthetic hands: experimental evaluation". In: *Journal of neuroengineering and rehabilitation* 7.1 (2010), pp. 1–14.

[7] David P. McMullen et al. "Demonstration of a Semi-Autonomous Hybrid Brain–Machine Interface Using Human Intracranial EEG, Eye Tracking, and Computer Vision to Control a Robotic Upper Limb Prosthetic". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22.4 (2014), pp. 784–796. DOI: 10.1109/TNSRE.2013.2294685.

[8] Julia Starke et al. "Semi-autonomous control of prosthetic hands based on multimodal sensing, human grasp demonstration and user intention". In: *Robotics and Autonomous Systems* 154 (2022), p. 104123. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2022.104123. URL: https://www.sciencedirect.com/science/article/pii/S0921889022000689.

[9] Miguel Nobre Castro and Strahinja Dosen. "Continuous Semi-autonomous Prosthesis Control Using a Depth Sensor on the Hand". In: *Frontiers in Neurorobotics* 16 (2022). ISSN: 1662-5218. DOI: 10.3389/fnbot.2022.814973. URL: https://www.frontiersin.org/article/10.3389/fnbot.2022.814973.

[10] Ghazal Ghazaei et al. "Deep learning-based artificial vision for grasp classification in myoelectric hands". In: *Journal of neural engineering* 14.3 (2017), p. 036025.

[11] Marko Markovic et al. "Stereovision and augmented reality for closed-loop control of grasping in hand prostheses". In: *Journal of Neural Engineering* 11.4 (2014), p. 046001. DOI: 10.1088/1741-2560/11/4/046001. URL: https://doi.org/10.1088/1741-2560/11/4/046001.

[12] Marko Markovic et al. "Sensor fusion and computer vision for context-aware control of a multi degree-of-freedom prosthesis". In: *Journal of neural engineering* 12.6 (2015), p. 066022.

[13] Jeremy Mouchoux et al. "Artificial perception and semiautonomous control in myoelectric hand prostheses increases performance and decreases effort". In: *IEEE Transactions on Robotics* 37.4 (2021), pp. 1298–1312.

[14] Maurice LeBlanc. "Give Hope - Give a Hand" - The LN-4 Prosthetic Hand". University Lecture. 2008. DOI: https://web.stanford.edu/class/engr110/2011/LeBlanc-03a.pdf.

[15] Kathryn Ziegler-Graham et al. "Estimating the Prevalence of Limb Loss in the United States: 2005 to 2050". English (US). In: *Archives of Physical Medicine and Rehabilitation* 89.3 (Mar. 2008). Funding Information: Supported by the U.S. Centers for Disease Control and Prevention (grant no. R04/CCU322981-02). Copyright: Copyright 2008 Elsevier B.V., All rights reserved., pp. 422–429. ISSN: 0003-9993. DOI: 10.1016/j.apmr.2007.11.005.

[16] Muhammad Jameel Mohamed Kamil, Sarah Moi, and Mohd Abdullah Sani. "Reassessing the Design Needs of Trans-Radial Amputees in Product Design Innovation". In: *Wacana Seni* 19 (Dec. 2020), pp. 61–71. DOI: 10.21315/ws2020.19.5.

[17] Mussarat Jabeen Khan, Sarah Fatima Dogar, and Uzma Masroor. "Family relations, quality of life and post-traumatic stress among amputees and prosthetics". In: *PAFMJ* 68.1 (2018), pp. 125–30.

[18] Beth D. Darnall et al. *Depressive Symptoms and Mental Health Service Utilization Among Persons With Limb Loss: Results of a National Survey*. https://www.archives-pmr.org/action/showPdf?pii=S0003-9993%2804%2901316-4. 2005.

[19] Thomas Feix et al. "The GRASP Taxonomy of Human Grasp Types". In: *IEEE Transactions on Human-Machine Systems* 46.1 (2016), pp. 66–77. DOI: 10.1109/THMS.2015.2470657.

[20] S. Došen and D. B. Popović. "Transradial Prosthesis: Artificial Vision for Control of Prehension". In: Jan. 2011, no. 1,pp. 37–48.

[21] Martin Vilarino. "Enhancing the Control of Upper Limb Myoelectric Prostheses Using Radio Frequency Identification". In: 2013.

[22] CYBATHLON. https://cybathlon.ethz.ch/en.

[23] Samreen Hussain, Sarmad Shams, and Saad Khan. "Impact of Medical Advancement: Prostheses". In: Nov. 2019. ISBN: 978-1-78984-383-5. DOI: 10.5772/intechopen.86602.

[24] M. Laffranchi et al. "The Hannes hand prosthesis replicates the key biological properties of the human hand". In: *Science Robotics* 5.46 (2020), eabb0467. DOI: 10.1126/scirobotics.abb0467. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.abb0467. URL: https://www.science.org/doi/abs/10.1126/scirobotics.abb0467.

[25] Mohammadreza Asghari Oskoei and Huosheng Hu. "Myoelectric control systems—A survey". In: *Biomedical Signal Processing and Control* 2.4 (2007), pp. 275–294. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2007.07.009. URL: https://www.sciencedirect.com/science/article/pii/S1746809407000547.

[26] Anders Fougner et al. "Control of Upper Limb Prostheses: Terminology and Proportional Myoelectric Control—A Review". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 20.5 (2012), pp. 663–677. DOI: 10.1109/TNSRE.2012.2196711.

[27] Aidan D. Roche et al. "Prosthetic Myoelectric Control Strategies: A Clinical Perspective". English. In: *Current surgery reports* (Mar. 2014). ISSN: 2167-4817. DOI: 10.1007/s40137-013-0044-8.

[28] Ning Jiang et al. *Myoelectric Control of Artificial Limbs— Is There a Need to Change Focus?* 2012.

[29] Miki Fairley. *UPPER-LIMB PROSTHETICS: PATTERN RECOGNITION SHOWS PRACTICAL PROMISE*. https://opedge.com/Articles/ViewArticle/2018-09-01/upper-limb-prosthetics-pattern-recognition-shows-practical-promise. 2018.

[30] Enzo Mastinu et al. "An Alternative Myoelectric Pattern Recognition Approach for the Control of Hand Prostheses: A Case Study of Use in Daily Life by a Dysmelia Subject". In: *IEEE Journal of Translational Engineering in Health and Medicine* 6 (2018), pp. 1–12. DOI: 10.1109/JTEHM.2018.2811458.

[31] P. Geethanjali. "Myoelectric control of prosthetic hands: state-of-the-art review". In: *Medical Devices (Auckland, N.Z.)* 9 (2016), pp. 247 –255.

[32] Carles Igual et al. "Myoelectric Control for Upper Limb Prostheses". In: *Electronics* 8 (Oct. 2019), p. 1244. DOI: 10.3390/electronics8111244.

[33] Dario Farina and Sebastian Amsüss. "Reflections on the present and future of upper limb prostheses". In: *Expert Review of Medical Devices* 13 (2016), pp. 321 –324.

[34] L. E. Carvalho and A. von Wangenheim. "3D object recognition and classification: a systematic literature review". In: *Pattern Analysis and Applications* 22 (Feb. 2019), pp. 1243–1292. DOI: 10.1007/s10044-019-00804-4. (Visited on 11/21/2020).

[35] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement*. arXiv.org, 2018. URL: https://arxiv.org/abs/1804.02767.

[36] Daniel Bolya et al. "YOLACT: Real-Time Instance Segmentation". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9156–9165. DOI: 10.1109/ICCV.2019.00925.

[37] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6 (1981), 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: https://doi.org/10.1145/358669.358692.

[38] Konstantinos G Derpanis. "Overview of the RANSAC Algorithm". In: *Image Rochester NY* 4.1 (2010), pp. 2–3.

[39] Boxuan Zhong, He Huang, and Edgar Lobaton. "Reliable Vision-Based Grasping Target Recognition for Upper Limb Prostheses". In: *IEEE Transactions on Cybernetics* 52.3 (2022), pp. 1750–1762. DOI: 10.1109/TCYB.2020.2996960.

[40] Faraj Alhwarin, Alexander Ferrein, and Ingrid Scholl. "IR Stereo Kinect: Improving Depth Images by Combining StructuredLight with IR Stereo". In: Dec. 2014. ISBN: 978-3-319-13559-5. DOI: 10.1007/978-3-319-13560-1_33.

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[42] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[43] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[44] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[45] *ConvNets convolutional networks*. https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html. Accessed: 2022-05-22.

[46] Kaiming He et al. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[47] Yi Li et al. "Fully convolutional instance-aware semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2359–2367.

[48] Cheng-Yang Fu, Mykhailo Shvets, and Alexander C Berg. "RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free". In: *arXiv preprint arXiv:1901.03353* (2019).

[49] Daniel Bolya et al. "Yolact++: Better real-time instance segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* (2020).

[50] *COCO-Common Objects in Context*. https://cocodataset.org.

[51] *Point Cloud Processing*. https://se.mathworks.com/help/vision/point-cloud-processing.html.

[52] Radu Bogdan Rusu and Steve Cousins. "3d is here: Point cloud library (pcl)". In: *2011 IEEE international conference on robotics and automation*. IEEE. 2011, pp. 1–4.

[53] Aksel Sveier. *Primitive Shape Detection in Point Clouds*. https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2402578/15648_FULLTEXT.pdf?sequence. 2016.

[54] Yufan Zheng et al. "A primitive-based 3D reconstruction method for remanufacturing". In: *The International Journal of Advanced Manufacturing Technology* 103 (Aug. 2019). DOI: 10.1007/s00170-019-03824-w.

[55] *Depth Camera D435*. https://www.intelrealsense.com/depth-camera-d435/.

[56] *Intel ® RealSense TM D400 Series Product Family Datasheet Intel ® RealSense™ Vision Processor D4, Intel ® RealSense™ Vision Processor D4 Board, Intel ® RealSense™ Depth Module D400, Intel ® RealSense™ Depth Module D410, Intel ® RealSense™ Depth Module D415, Intel ® RealSense™ Depth Camera D415, Intel ® RealSense™ Depth Module D420, Intel ® RealSense™ Depth Module D430, Intel ® RealSense™ Depth Camera D435, Intel® RealSense™ Depth Camera D435i*. 2019. URL: https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf.

[57] Alemão M. Raška M. Stück B. *Computer vision based system for the enhancement of grasping for transradial prostheses.* https://github.com/ROB8-very-NOICE/P8_project.git. 2022.

[58] Jochen Triesch and Christoph Von Der Malsburg. "Robust classification of hand postures against complex backgrounds". In: *Proceedings of the second international conference on automatic face and gesture recognition*. IEEE. 1996, pp. 170–175.

[59] Michael Zeller et al. "A visual computing environment for very large scale biomolecular modeling". In: *Proceedings IEEE International Conference on Application-Specific Systems, Architectures and Processors*. IEEE. 1997, pp. 3–12.

[60] James Crowley, François Berard, Joelle Coutaz, et al. "Finger tracking as an input device for augmented reality". In: *International Workshop on Gesture and Face Recognition*. Citeseer. 1995, pp. 195–200.

[61] Munir Oudah, Ali Al-Naji, and Javaan Chahl. “Hand gesture recognition based on computer vision: a review of techniques”. In: *journal of Imaging* 6.8 (2020), p. 73.