ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
FACULTY OF INFORMATICS

MACHINE VISION BASED INTELLIGENT HUMAN LABOUR ASSISTANT SYSTEM

ÓE-NIK	Student's name:	**Sárvári Marcell**
2022	Student's registration number:	**T/008718/FI12904/N**

Óbuda University
John von Neumann Faculty of Informatics
Software Engineering Institute

DIPLOMA THESIS
TOPIC DESCRIPTION

Student name:	**Marcell Sárvári**
Registration number:	T/008718/FI12904/N
Neptun's code:	J915SG

Title of diploma thesis:

Machine Vision based Intelligent Human Labor Assistant System
Gépi látás alapú intelligens munkafolyamat segítő

Internal supervisor:	Dr. habil. Sándor Szénási Dr. Zoltán Vámossy
External supervisor	
Deadline:	15th of May, 2022.

Topic description:

The aim of this diploma thesis is to design and implement an application based on machine vision and spatial pattern recognition methods to assist the quality controller's work. The student should analyze the already existing similar systems focusing on their advantages and disadvantages. Based on these experiments, he should design a novel system which is able to help labor assistants.
It must be capable to check the worktable with a camera every second and detect and identify boxes. Based on this identification, when the box is opened, the projector must be able to draw numbered shapes where the different items have to be packed into a foam cutout in the desired order. One the box is packed, the system must be capable to check the placements of all items. Finally, the projector must be able to draw a rectangle to the closed box where a barcode needs to be sticked.

The diploma thesis should cover:

- detailed literature review,
- analysis of current state-of-the art techniques,
- specification, design and development of the task,
- implementation of the solution,
- evaluation of the results,
- detailed documentation of the research and development phases,
- summary of the results and further plans.

ÓBUDAI EGYETEM Neumann János Informatikai Kar 1. Budapest

Ph.

Vámossy Z

……………………………
Dr. Zoltán Vámossy
head of institute

Term of limitation: **15th of May, 2024.**

The diploma thesis is adequate and can be submitted:

…………………
external supervisor

Szénási Sándor

……………………
internal supervisor

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

JOHN VON NEUMANN FACULTY OF INFORMATICS
SOFTWARE ENGINEERING INSTITUTE

HALLGATÓI NYILATKOZAT

Alulírott Sárvári Marcell (Neptunkód: J915SG) hallgató kijelentem, hogy a diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

STUDENT'S DECLARATION

I the undersigned student hereby declare that this degree project / thesis is the result of my own work; I have disclosed the references and tools used in an identifiable manner. The results indicated in my degree project / thesis completed may be used by the university and the institution announcing the task for their own purposes free of charge.

Budapest, 2022.05.01

Sárvári Marcell
....

TABLE OF CONTENT

TABLE OF FIGURES

INTRODUCTION

Automation has greatly increased in the global industry over the past years as more and more companies are pushing forward to increase their production efficiency while also manufacturing with constant precision. Quality control is highly involved as it helps with filtering out defective items and further improving efficiency by providing useful information about any step of the manufacturing process. Automation also reduces direct human labour costs and expenses, but there are occasional tasks where full automation is not practical or economically viable. With the Industry 4.0 and newly introduced 5.0 standards, it is preferred to have an insight into every step of the production, but it can be difficult to measure human labour quality. The aim of this thesis is to develop and measure the viability of such a system of which aim is to both aid human labour workers and measure their performance.

Packaging the same type of items into boxes can usually be done with an automated process, but when there are many different and everchanging type of products and packaging, human labour has to be used. The task of the 'Intelligent Human Labour Assistant System' is to detect incoming boxes, notify the worker of the type, help with deciding which item (or items) to put into the box, show sticker placement locations and also measure the quality of the work with the help of image processing.

Before starting the development of the application, the parts of automated inspection systems have to be identified. By looking at how novel applications work, what advantages and disadvantages each solution has will help understand with which components should be selected for the task. The performance of the application will be measured, and a conclusion will be drawn regarding the success of the task.

1. QUALITY CHECKING IN MANUFACTURING

Quality control has been an essential part of any industrial manufacturing line, especially if the manufacturing lines are partly or fully automated. To recognize and manage a quality issue after a product has been manufactured, packaged and/or dispatched may cause more problems both financially and morally for a company than to identify and fix them before the product is finished. Assembly lines can be followed step-by-step by using statistical process control (SPC) software; therefore, the methods may be adjusted as soon as a quality difference of a product is noticed. Having access to (near) real-time quality of the manufacturing during production has a number of benefits, including less unnecessary work, financial savings through higher efficiency, and less customer complaints, recalls.

> One of the most well-known cases of quality control failure has to be connected to the now defunct Takata Corporation. The company was the largest vehicle airbag manufacturing company in the world, but due to an improper seal which could cause the malfunction of inflation and improper discharge, the company had to file for bankruptcy in 2017. This manufacturing error was present in over 42 million cars in the US market alone and since it was unnoticed for over a decade, the company could not cope with the cost of the investigation, leading to its demise. [1]

Before checking the quality of a product, the check-up process has to be decided, since various products require different types of inspections. Manufacturing products with given margin of errors – for example, engine parts, surfaces, precision tools, which have been machined - have to be tested by a measurement tool, as human eyes would not be able to detect a problem without reference. Depending on the type of the measurement tool it might work either analogue or digitally and may be done by a human inspector or could be automated.

Analogue tools do not need external power but require a worker who is able to read out the values properly, which can be time consuming and problematic on scales that is smaller than a millimetre. With sizes less than that – in case of callipers -, the width of the lines printed onto a device cannot be diminished anymore and may affect the value read by the user. Another factor is the lightning conditions that can also affect the readability, furthermore reducing the repeatability correctness. In addition, when the value needs storing, it has to be manually entered into a logging system.

In contrast, digital tools can measure values without any intervention from a worker. After calibrating accurately, these tools will have a high point of repeatability precision, which is essential in quickly measuring products. Another benefit of digital measurement tools is that they are not affected by visual factors, such as incorrect lightning, consequently the precision of measurement can be several levels higher than their analogue counterparts. Digital measurement tools also provide possibilities to store the measured data into logging systems without interruption, not to mention, giving way to further analysation of data later down the line. Digital tools however cost significantly more than their analogue counterparts, especially as the precision level gets higher and higher. Different mechanical measurement processes check the size, shape, material surface or magnetic properties of an object, while spectroscopic inspections check the chemical composition of a material.

In case of electronic products, the correct conductivity has to be tested in order to determine the wiring of an item, and if it is functioning as intended. If there were a terminal unconnected, its component would not operate, whereas shorting could create a potential fire hazard. Electrical testers are used for checking the wiring of the products. This test process is completed by sending electrical signals down one end and measuring the received signal on the other. There are handheld testers so-called multitesters which are handled by a worker, who operates it just by touching the two metal probes to a circuit. As a result, it shows different types of information about the circuit depending on the chosen setting. Generous versatile functions are available during the using of these handheld devices and can be easily adapted to any type of electrical testing. However, it can only test one circuit with one attribute at a given time, making complex testing time consuming. Larger, automated electrical tester cabinets can manage the checking of a complete printed circuit board with hundreds of electrical lines within seconds, but in this case each type of product has to be pre-configured. The inspection time is significantly faster than the manual one, but adaptation of the test bed to new products is costly.

A form of aesthetical testing is also expected from most checking processes, where visual errors are spotted. These faults can either cause functional or aesthetical problems. For instance, in case of injection moulding plastic, incorrect amount of input material can cause deformation in the desired shape; or in electrical components incorrectly soldered components onto a printed circuit board (PCB) may cause contact errors. Surface imperfections such as scratches and dents do not necessarily mean a defect in the product but may lead to customer dissatisfaction.

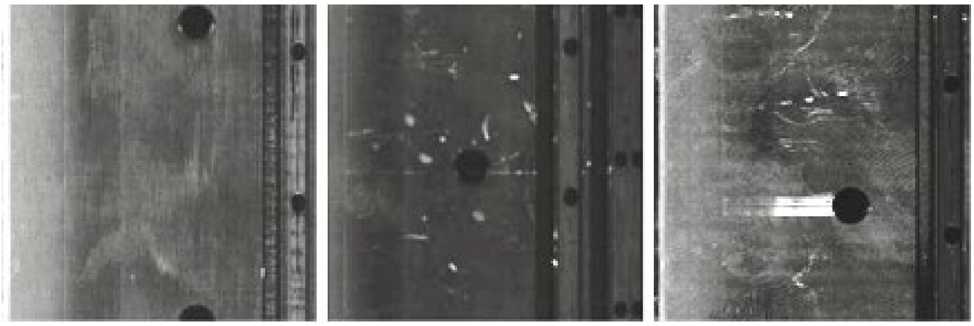


Figure 1.1: Different levels of surface damage

Source: [2]

Many of these quality checking methods can be implemented by using a camera with a computer, where a program calculates these values and gives evaluation based on the image taken of the product. This process is called machine vision.

2. MACHINE VISION AND ITS COMPONENTS

2.1. INTRODUCTION TO MACHINE VISION

Machine vision is an automated process where a computer is used to capture and process images which results in a measurable value of one or more attributes. The use of machine vision is not limited to the industrial sector, it is also found in academic, governmental, and military applications. Industrial vision systems require a high level of stability and reliability, especially compared to academic use-cases, but their costs are less than governmental, and especially military applications.

Machine vision systems are responsible for taking images and giving a comprehensible result. To achieve this, five basic components are needed: an inspected object, a camera, correct lightning, a processing unit, and an input trigger. The result of these kinds of systems are not limited to only ‘Yes’ or ‘No’ decisions, they can also accomplish objective measurements, such as determining the gap size of a spark plug, or getting a location/alignment of an item, and providing the coordinates to an automated robot which repositions the item for the oncoming manufacturing process.

As for the process of taking a picture, a digital sensor is required. This sensor takes an image of the inspected element as soon as a trigger sensor sends out a signal. This image is then transmitted to the main controller unit, which examines it in a pre-configured way and determines a calculated value according to a reference value. If this value is within the set borders, the examined object gets a pass – an ‘OK’ value – but rejected if not – an ‘NG’ value. This result is sent out as an output and can be processed with either a visual representation for a worker or a signal telling a machine to reject the imperfect item.

There are two main categories of quality checking applications in case of an industrial manufacturing process: quantitative and qualitative measurements. Quantitative measurements are based on structured scenes with little or no variation as computers are excellent at determining results with small variation in margins. On production lines a machine vision system can examine hundreds of parts per a minute, and a system with the suitably chosen camera resolution, lightning and optics setup can inspect details that would be too small for the human eyes to see otherwise. Besides, testing the quality of an object without physical contact not only reduces the potential for component damage during the process, but eliminates costs of wear and tear on mechanical parts. Not to mention, machine vision systems can be installed in locations where human workers could not be placed due to high-risk, dangerous environments with high-speed machinery working close.

Qualitative measurements are based on complex, unstructured inspections of an item, which can examine the aesthetical part of a product. As an example, in case of a wooden furniture manufacturing company, the grain of the wooden surface is an essential part of the design, and it is difficult to find two of the same kind of surface textures. Determining the quality of the surface might take no more than a second for a human worker, but it could impose a nearly impossible task for a computer. The type of veins, margins for the size of a cramp and the discolorations of the surface cannot be independently described, as they make up the design of the piece together. Automating such a task has been near impossible up until recently, as static configurations have to be either updated regularly or made extremely robust. This brings the cost of a system up significantly, making human employment far more economical. Machine learning could in theory solve this issue, but an industrial machine vision system has to be highly reliable while remaining stable. Machine learning is simply not yet proven in this regard within the industrial sector where an hour-long outage could cost upwards of 10,000 euros in missed production. On the other hand, sectors where there is no such immediate danger, like in the IT sector, machine learning does get use in computer imaging.

With a complete example, an automated inspection station can be described as such: A company produces electrical motherboards where the quality has to be checked before packaging. To solve this, the products are checked by a two-stage quality checking process. Initially, the different components on the motherboard are identified with a machine vision system, and if each component is placed correctly, an electrical tester checks the functionality of the panel. The products arrive on a conveyor belt one by one. The machine vision system has a camera with the correct resolution to get a clear image of the examined pieces on the motherboard. Correct optical lenses are chosen to provide proper focus and aperture and stable lighting is fitted with optimal positioning and brightness. The camera is connected to the image processing unit with either a proprietary cable, through USB or an RJ49[1] connector. As a first step, the inspected piece is positioned under the camera. Secondly a trigger sensor detects it and has the camera take an image. This image is transmitted to the image processing unit for measuring, which values were pre-configured by an engineer earlier. With a number of different steps, the taken image's measurement values are compared to a reference value. Depending on the complexity of the measurements, it can take anywhere from a tenth of a second to tens of seconds to process depending on the selected hardware of the image processing unit. After the calculations are finished, the tested piece is assigned an 'OK' – pass – value or an 'NG' – reject – value that is sent onto the mainframe of the company where it gets stored. This value may affect how the next station handles the item, and if the item is rejected, instead of sending it into the electrical tester it will be sent to the side where a worker has to take the necessary steps to repair the item. [3]

2.2. COMPONENTS OF MACHINE VISION SYSTEMS

Different types of inspections require different methods, and to reach the complete potential of the capabilities of a system and hit peak efficiency, a correctly configured machine vision system is essential. Depending on the allocated costs of a customer and the complexity-, size-, speed of feeding of inspected items, a wide range of machine vision setups are available. For instance, if a company needs to check the presence or a colour of a component, this problem can be solved perfectly by a simple, less expensive one-camera machine vision system instead of applying a multi camera, high speed processing setup. To find the best solution determining the requirements of the customer in creating a vision system is important, because unsatisfying results can undoubtedly have negative consequences for both the consumer and producer. The producer has to take its previous knowledge into consideration in case of expandability of the system later, as the consumer cannot always identify how to reach the full potential of the product in the first place. Regarding to inspecting or positioning, machine vision applications can

[1] The RJ49 connector is most commonly used for ethernet connection.

range from simple to complex with tasks and selection of the correct hardware has to be done after the application of the machine vision has been determined.

There are a number of different companies that produce industrial machine vision systems, such as Siemens, Cognex, Basler but as the writer have previous experience with the systems created by Omron, they will be used as an example. Omron Vision systems are a popular choice in the industry, as they have a large portfolio of products which can cater to the needs of the customer. These applications range from the Omron FQM systems providing solutions to positioning tasks, through the Omron FQ2 intelligent systems for simple inspections with built-in camera lenses and lightning for easy setups, to the more complex Omron FH Image Processing Units which can cope with even the most intricate of applications. The FH Image Processing Unit is a small industrial-level computer running embedded software. The FH unit can be configured in a number of ways depending on the speed requirements. Its specifications can range from an Intel Atom 1.2Ghz processor with two gigabytes of ram to a more performing unit using an Intel i7 4-core 2.8Ghz processor with eight gigabytes of ram. These units have a large selection of communication methods with different types of protocols, including Parallel Input/Output, Ethernet, EtherCAT, Universal Serial Bus (USB) and a serial port as well. Depending on the specifications, multiple cameras may be connected to the system.

> A car seat manufacturing company located in Hungary has a machine vision system which runs 18 cameras on a high-performance FH system. Its purpose is to check for manufacturing defects and log the results. This is used in tandem with human workers assisting their work. [4]

The cameras used are digital ones with either Charge Coupled Device (CCD) or Complementary metal–oxide–semiconductor (CMOS) sensors. CCD works by exposing the sensors to light, and after the sensors have been charged, the electrons are shifted from each line of sensor value down by one. On the bottom layer the charges are measured one-by-one on each sensor, and an image is taken. As this method does not require the usage of electrical amplifiers so the created image is clearer with less noise. CCD uses global shutter, which means that each sensor is exposed to light at the same time. Here is a disadvantage of CCD, as it has a high-power consumption because of requiring high voltage. As a result, on larger resolutions reading the sensor data may take considerably longer, while also introduce unwanted altering of the charge value during each shifting.

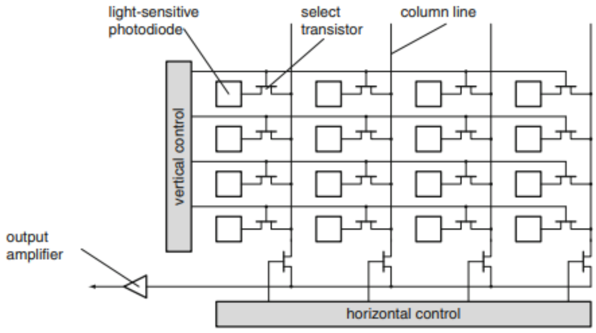


Figure 2.1: Structure of a CMOS sensor

Source: [2]

The other image capturing technology, the CMOS chips have their electron charge to voltage conversions and amplifiers at each sensor, eliminating the need for high power draw. The sensors are exposed to light – one at a time - and the values are read from each line by the processor unit and the image is created. As for the costs, although CMOS chips are more power efficient and are less expensive than their CCD counterparts, but by using rolling shutter, capturing moving objects with high speed can present malformed images. There are CMOS chips which also use global shutter instead of rolling shutter to eliminate artifacts, but the cost of a camera with global shutter is several times more expensive than a rolling shutter one, especially as resolution increases. Objects moving with high-speed require a camera using global shutter, as the image may deform due to the object's change of location while reading the sensor data.

Figure 2.2: The rolling shutter effect

Source: [5]

To operate correctly a camera sensor requires a lens. There are a large set of different lenses available, but the correct one has to be chosen for the application. The lens or optics of a camera is primarily responsible for setting the brightness of the image, the field of view, the depth of field and the optical distortion. A camera lens is made up of multiple glass lenses inside a moveable mechanism, and if the image should be changed, different combination of single lenses can solve this requirement. Numerous different camera lens technology has been developed for machine vision aimed at solving different problems. When choosing a camera lens there are some very significant features to consider. They are the distance from the inspecting object to the lens and to the camera, depth of field, field-of-view, resolution, and last but not least, focal length, which is the distance from the camera's sensor to the lens. As the distance increases between the two, the magnification does so, and smaller objects will seem to be larger. This allows for inspecting smaller items with higher resolution at the cost of less components fitting into the image. The distances between the inspected items, the lens and the camera sensor are significant values when calculating how far the focal point should be. The camera is in focus when the examined object is at its clearest state on the image and the focal point is where the focus is located. With different amounts of depth of field, the range of focus can be narrowed down or extended out. If the components are placed in different positions, in this case a bigger depth of field effect is required because they all have to be in focus. On the other hand, when the environment interferes with the created image, a lower depth of field can hide the disturbing details of the background and the examined object will have no interference with points not located nearby. Most industrial applications use a fixed focus point as incorrect autofocusing could introduce blurriness. The field of view is the number of degrees the camera will see to the side. It means, that the larger value is the wider viewing angle of the camera will be resulted, but the distortion of the image will be higher. The field of view is affected by the focal length of the camera and the distance from the inspected object. So called CCTV lenses are the most commonly used camera lenses because they provide a wide range of options with variable field of view sizes, during keeping the costs of a lens low. Their aperture and focus are adjustable, making them a good solution for most use cases, however they are highly sensitive of vibrations, and they suffer from distortion during inspecting close items. Macro lenses are specifically designed for close inspection scenarios. Their size is compact with higher resistance against vibration, and they have little amount of distortion at close distances. It is worth noting that they only operate at a short given range because of their working principles. A special type of lenses is telecentric optical lens, where the incoming main light beams are parallel with the optical axis. This results in no perspectival distortion and a high precision of measuring – each inspected point is the same distance from the lens. Measurements are easier and more precise, while less computing time is necessary as they create a projection image of the inspected object. Their large size and high purchase cost are the main limiting factor.

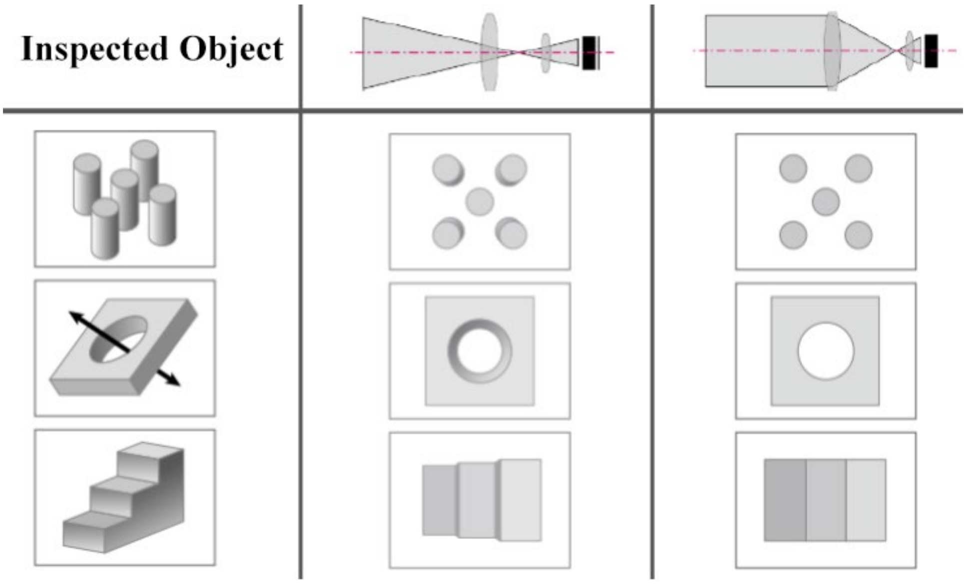


Figure 2.3: Difference of traditional and telecentric lens imaging

Source: [6]

Besides the above-mentioned lenses, there are several other kinds of special lenses available, like zoom, pericentric and hole inspecting lenses. As for zoom lenses, they allow choosing the field of view without changing the distance between the object and camera. Hole inspecting and pericentric lenses allow for viewing the complete inner or outside surface of the inspected item in one picture.

Figure 2.4: Pericentric lens' view

Source: [6]

Choosing the correct lightning setup depends on the examined object. It is highly important to consider the fact that different material surfaces have their own light reflection properties. The most common static lighting solutions for a machine vision setup are the following:

- direct lightning
- indirect lightning
- back lightning
- angle lightning
- low angle lightning
- collimated lightning and
- lightning from a convergent beam.

Direct lightning means that an object is illuminated without disrupting the light wave and the image can be seen with high contrast. In this case, unwanted shadow formations can occur because of strong reflections. If this problem arises, a diffusor can be put facing the light source. As for the material of the diffusor's, plastic or frosted glass can be selected. The diffusor creates indirect lightning, and the diffusors scatter the incoming light beams and also soften their edges. As a result, an even light distribution is created with less shadows and reflections, but with lower contrast and brightness of the image. If the reflected light still causes problems, the light source can be placed at an angle of 45° or 15° for low angle lightning. By using angle lightning, the light waves bounce off the surface of the material and a more detailed examination can take process, but colour information is lost.

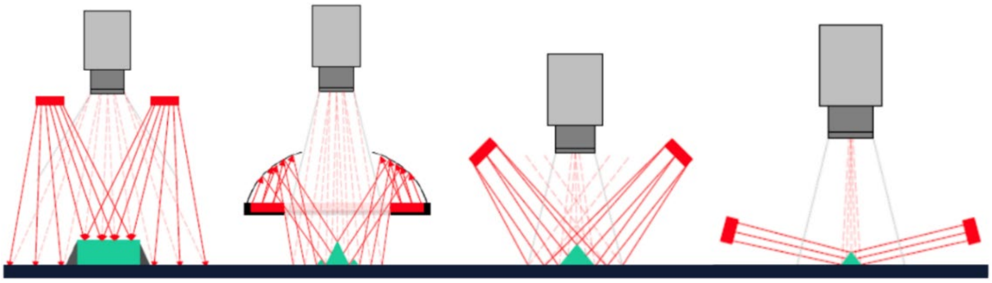


Figure 2.5: Direct, indirect, angle and low-angle lightning setup

Source: [6]

When the source of the lightning is placed behind the object, it is called the back lightning. The shape or alignment of an object can be measured and aimed at the camera, creating a high contrast image. The details about the surface of the material image cannot be seen, but very clear edges and precise measurement tasks are resulted. Collimated lightning can be used in case of telecentric lenses where a semi-permeable mirror is placed in front of the camera at 45° and the light source is placed at 90°. As features, there are no shadows affecting the image, shiny surfaces appear to be brighter, and no perspectival distortion is present. There is also a drawback, because only 1/4th of the light waves can reach back to the sensor, consequently, making the image darker altogether.

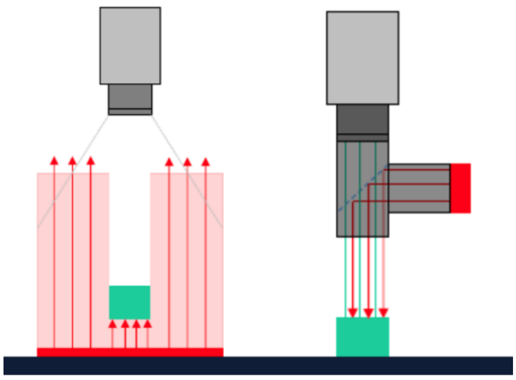


Figure 2.6: Back- and telecentric lightning

Source: [6]

These are the main components of a novel 2D machine vision system, but more and more inspections require the measurement of 3D objects, requiring different hardware. There are a number of different options a system can use with each having their advantages and disadvantages.

2.3. MACHINE VISION IN 3D

An increasingly important role is given to the third dimension in image processing applications. The decision of whether 2D or 3D camera technology should be used to handle a respective inspection task has to be made in the beginning, with a good deal of consideration. While the decision is clear with some applications, it is not the case with others. Both technologies have their respective advantages and disadvantages, and these have to be understood so that they can be used appropriately. 2D image processing is suitable whenever the application offers high contrast or if the structure and colour of the object are decisive for the end result.

A distinguishing feature of 2D and 3D cameras is that the image not only contains values on the X and Y axis, but also on the Z, depth axis as well. This opens up entirely new possibilities to solve intricate tasks, particularly in robotics, factory automation and in the medical sector. 3D cameras are also commonly found in consumer and entertainment products, such as in stand-alone virtual reality and augmented reality headsets. 3D image processing is used in places where shapes, volumes or the 3D position of an object is required. Depth information can also be used to handle tasks in the inspection of objects that do not have enough contrast for 2D but do show a significant difference in height. 3D cameras are typically found in the following applications:

- Detection of obstacles in autonomously driving vehicles in an industrial environment, such as automated carts
- Robot-controlled gripping tasks on conveyor belts
- Presence detection, checking objects in a box, even if they have no contrast at all against the background
- Inspection of position and presence of components on a circuit board
- Volume measurements of a wide range of objects

3D images can be created by utilizing different technologies, each satisfying different application needs. The three main technologies currently in use are as follows:

- Stereovision with structured lightning

- Laser triangulation
- Time-of-Flight Lidar

Each of the technology is based on a different principle to record the depth axis and they each have various advantages and disadvantages. The technologies also complement each other, and the most suitable choice will depend on the requirements of the corresponding application.

Traditionally 2D technology has played a greater role in the image processing market, but the trend towards 3D solutions is constantly increasing. In the coming years, 3D image processing will become increasingly important particularly due to the trend towards Industry 4.0 and the growth of automation.

2.3.1 STEREOVISION

Stereovision imaging is based on the principle of a human pair of eyes. Two cameras are used to record two 2D images of an object, each from two different positions and the depth information is assembled into a three-dimensional image with the aid of the triangulation principle. A matching algorithm is used to search corresponding points in the left and right image and a depth image of the scene is created.

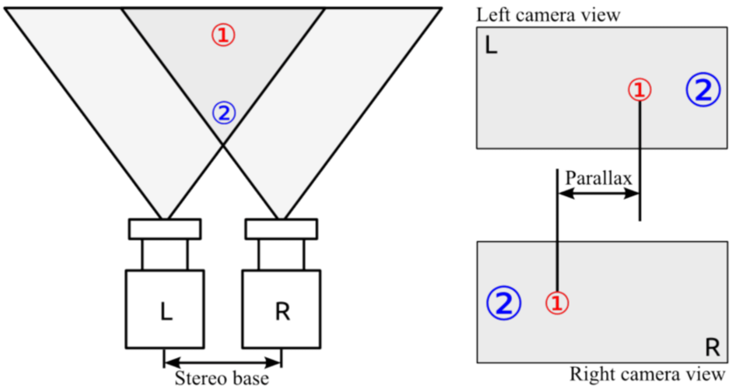


Figure 2.7: Stereovision principle

Source: [7]

The working distance at which this process functions depends on the distance between the cameras and requires calibration beforehand. Stereovision systems' accuracy can range from centimetres down to a millimetre, but this depends on the camera resolution, object distance and lightning. A big factor in the accuracy of the depth estimation is the target object. If the surface is homogeneous and no reference points can be found, or the surface is covered with repetitive texture, the system will have a hard time calibrating the distance. One option for improving the accuracy of a stereo system is to add structured light. Structured lightning is a type of lightning method where a higher control over the brightness/location/shape of the light beam is achieved. With a light source that can project different geometric patterns with control on brightness on the scene, measurement results become more precise and the disadvantages of stereoscopy due to homogeneous surfaces and low light are significantly reduced. Stereovision can often be found in coordinate measurement technology, size measuring of objects and workspaces for applications with industrial, service or robot systems, as well as in the 3D visualization of locations that are inaccessible or dangerous to humans. A well-known example of a consumer stereovision camera is the Microsoft Kinect. [8]

2.3.2 LASER TRIANGULATION

The process of laser triangulation uses a combination of a 2D camera and a laser light source. In this procedure, the laser emits lines, or dots onto the scene in front of the camera. If an object is lit by the light, the difference between heights can be calculated by measuring the distance between the two lines and using the $y = x * tan(\Theta)$ formula. The angle is known and the calculate the height of the object is solved by with measuring '*x*'. Systems using laser triangulation require extensive calibration and are difficult to adapt to different procedures once deployed. [9]

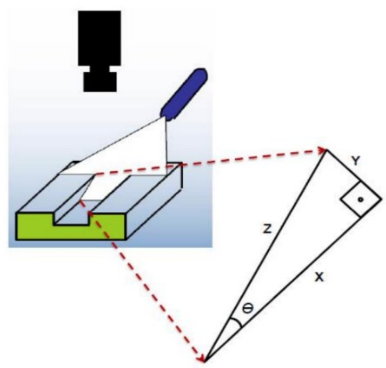


Figure 2.8: Calculating height from perspective

Source: [6]

2.3.3 TIME-OF-FLIGHT METHOD

A Time-of- Flight camera provides two kinds of data for each pixel: the intensity value, stated as the grey value, and the distance of the object to the camera, namely the depth value. This technology uses a highly efficient way to get depth data and measure distances. There are two main methods which use the Time-of-Flight technology: the continuous wave and the pulsed Time-of-Flight method. The continuous wave Time-of-Flight method is based on measuring the phase shift between emitted and reflected light of a brightness-modulated light source while cameras working with the pulsed Time-of-Flight method determine distances based on the time delay between the emitted and the reflected light pulse.

Time-of-Flight cameras are suitable for applications that require a large working distance, high speed and low complexity. They do not require special lightning and are quite versatile, however highly reflective or non-reflective surfaces can give of false depth readings. The average accuracy is within a centimetre, but it can be brought down to a millimetre if the setup allows it. The current use of these cameras includes volume measurements in logistics, palletizing and de-palletizing tasks as well as autonomously driving vehicles in a logistics environment. Due to their relatively low depth precision, systems with Time-of-Flight cameras are more suitable for generalized tasks such as pick and place applications of larger objects in the industrial area. They can also be used for robot control systems or the measuring and position detection of large objects, for example in automotive manufacturing. [10] [11] [12]

3. 3D TECHNOLOGY IN PRACTICE

As previously already stated, different applications require different solutions. There is no such thing as a perfect solution for every application, therefore each application has to be evaluated in terms of its requirements and the suitable technology. The task has to be properly identified to decide whether 2D or 3D imaging is required, and both of the technologies' advantages and disadvantages have to be taken into consideration.

There are a number of points which can help narrow down the possible solutions, such as:

- What level of accuracy does the application require?
- What is the surface condition of the objects?
- What is the working distance?
- What is the required speed of the system?
- Does the system have to work in real-time?
- What are the requirements for installation and setup?
- What is the total budget for the application?
- What is the maximum cost for the 3D solution as an individual component?
- What external factors may affect the system?

After answering these questions and prioritizing, the solution may be decided. Two case studies will be presented to show real-life use-cases of 3D measurements and describe their working methods, which will also help identifying possible problems with the thesis work.

3.1. FERRITE HEIGHT CHECKING

A company located in Hungary manufactures high-power controlling modules which require the use of ferrite cores. These cores are placed onto the printed circuit board by a human worker, but their position may not be perfect. The task of the inspection system is to check the presence of the cores, their position and their orientation. Position checking also includes checking the height of the cores which in turn tells whether they are seated correctly. If it would not be the case, the cores would either get damaged or crush the PCB underneath them when the top cover is installed.

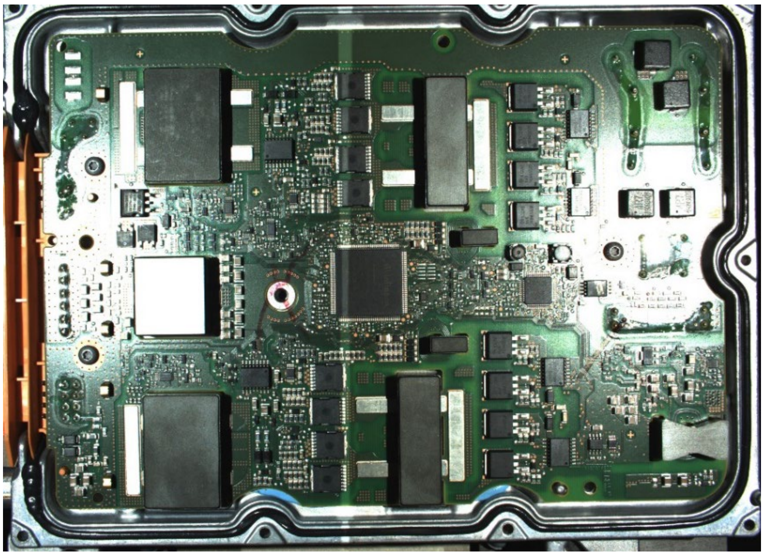


Figure 3.1: Ferrite cores in a PCB

Source: Own source from internship

The main components of the system are a single 2D camera and a structured light source, a projector. Each inspected board can be at different locations underneath the camera, therefore their location has to be identified. This is a very important step, as different perspectival distortion has to be accounted for each location. Once the board is located, the projector displays a cross on the location of each core. With the help of the triangulation principle the height can be calculated by the space between the lines. As mentioned before, the position has to be identified beforehand accurately, as both the projector's and the camera's lens work in a non-linear perspective. The camera's and the projector's angle are unknown with respect to each other, and the height difference is calculated relative to the board. The actual height is only known because of a calibration process when pieces with known height were put into the system, but without this step, the system would not know the true height.[2]

[2] The system does not have to know the exact height difference, it just has to measure whether the pieces are within tolerance.

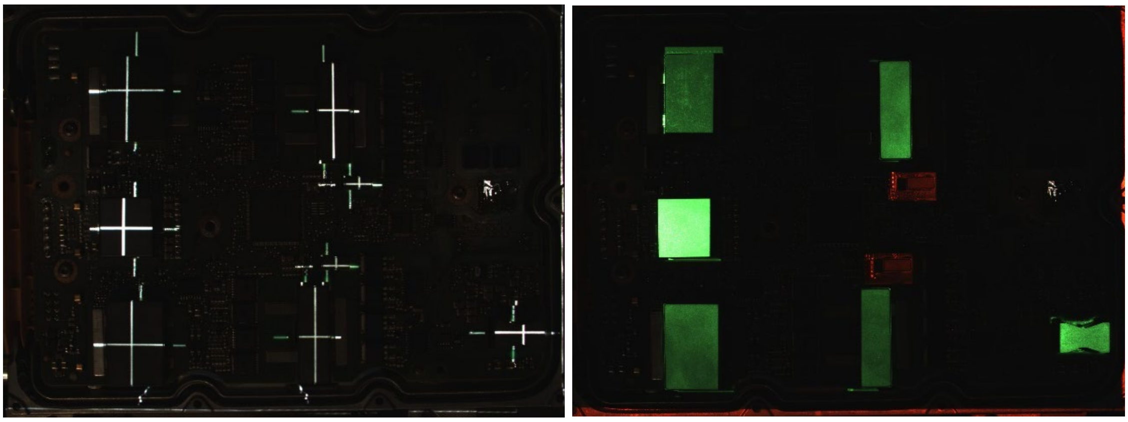


Figure 3.2: Ferrite height measurements and their evaluation

Source: Own source from internship

Once the measurements are finished, the projector displays coloured rectangles on the cores depending on their measurements.

3.2. OBJECT IDENTIFICATION

A different company manufactures metal parts with the help of an automated welding robot. The parts arrive to the robot on a conveyor belt in random order and position placed by an assistant worker. The robot has to know the position, orientation and height of the part to identify and pick up the correct item. Similar to the previous setup, a single 2D camera is used with a line laser beam, but in this case the angle and distance of the laser emitter is known relative to the conveyor belt.

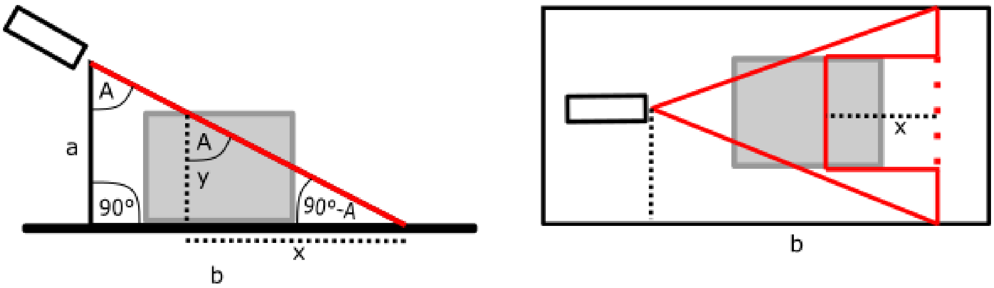


Figure 3.3: Laser triangulation principle

Source: Own creation

As the objects travel under the camera, the line of the laser is continuously recorder and calculations take place. A new image is created from the height data with colours representing height, and further image processing takes place on this image.

Figure 3.4: Raw image and a digital example of the process

Source: Own source from internship

Since the height and angle of the laser is known, all the required numbers can be calculated in an absolute way. However, if the alignment changes the values will be incorrect.

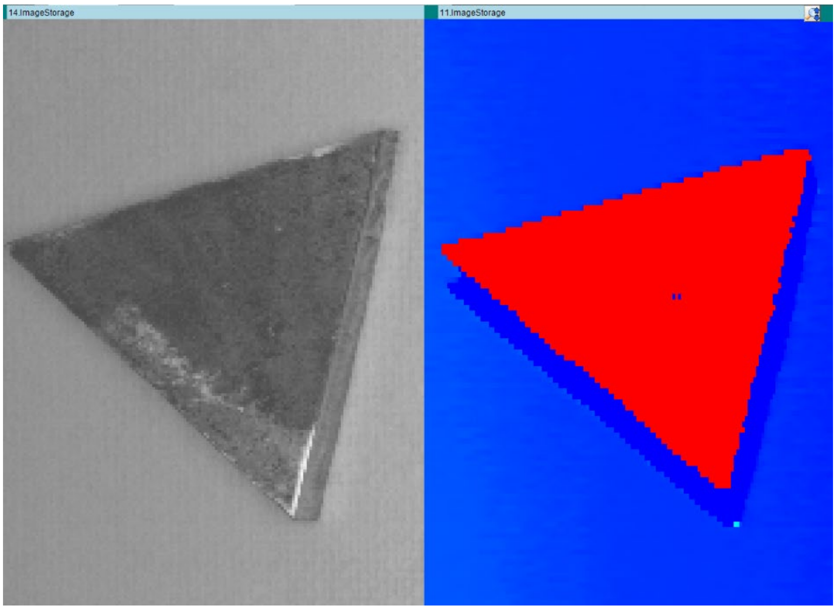


Figure 3.5: Meshed image (grey) and generated image based on height data (colour)

Source: Own source from internship

The two cases demonstrated how triangulation can be used with the help of a controlled light beam and depending on the setup, relative or absolute values can be gathered. These systems are only using 2D cameras with the measurements depending on the correct environment. Stereovision and Time-of-Flight systems are yet to be common in the small-scale inspection area as the disadvantages often do not allow these systems to work reliably. With the possible hardware solutions introduced the development of the thesis work can start.

4. INTELLIGENT HUMAN LABOR ASSISTANT SYSTEM

4.1. DESCRIBING THE TASK

There are occasions when a task is not economically viable to automate. Such tasks are when an everchanging variety of items have to be processed, requiring constant reconfiguration of the system. In this case human labour is used, but it is difficult to measure the quality of a human worker without also distracting them from their job.

This thesis would like to discover whether it is physically and economically possible to create a system which can cope with the following tasks:

- Measure the quality of the worker
- Aid the worker
- Help with the training of the worker
- Does not involve additional processes, which could hinder performance
- Safe for working environment

An example application was determined to be the following: A human worker needs to pack boxes with items, each different shaped box requiring different types and amounts of items to be packed. The system should identify the box based on its shape and texture and load up the corresponding program. Once loaded, the worker should be notified of the item that needs to be placed inside the box at a determined position. Once finished with packaging, the system should check the presence of all the necessary items and their location and act accordingly. Finally, a sticker location has to be displayed for the worker and checked once placed on the top of the box.

Figure 4.1: Workflow of the application
Source: Own creation

The system should also be able to work in a wide range of environments to be adaptable to different tasks, as the box packaging task is just an example.

4.2. CHOOSING THE COMPONENTS

The system should consist of the following four main components:

- A camera for recording images for processing and quality control
- Lightning for proper illumination of the work area
- Feedback system for the human worker
- Processing unit capable of running the software

The task of the camera is to identify the boxes and their content. This requires separation of the box from the workbench and measuring its parameters. Since it has to identify the contents as well, it should be located above the box, which makes it difficult to measure the height of the box with a simple 2D camera. Structured lightning could be of help, but constantly flashing bright lines and other shapes could pose a health risk on the worker. Stereovision cameras can measure distance without additional lightning, but they do find it difficult to identify the object when the texture is homogeneous, which is often the case with cardboard boxes.

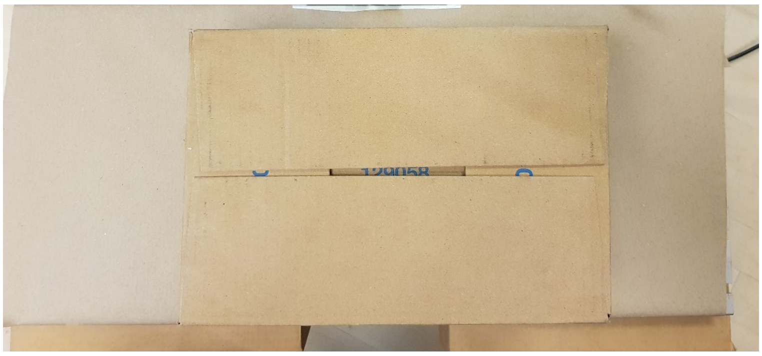


Figure 4.2: A cardboard box on a cardboard paper

Source: Own creation

Time-of-Flight (or Lidar) cameras also do not require additional lighting and the object's texture plays little in gathering depth information[3]. The chosen camera for the task was an Intel RealSense L515 Time-of-Flight which houses a 2MP RGB sensor and a depth sensor with a working range of 0.25 to 9 meters with accuracy up to 5mm at 1 meter. The

[3] As mentioned before, highly or non-reflective surfaces do not cooperate well with ToF cameras.

camera also has a high precision gyroscope but will not be utilized in this project. [13] [14]

Figure 4.3: The Intel Time-of-Flight camera
Source: Own creation

The communication between the worker and the computer could be done through a monitor and a mouse, but that would mean that the worker has to take their eyes off the workbench after each step. To provide a feedback system and illumination, a projector should be used, as it can display all kinds of information on the workbench and on the box itself without disrupting the worker. The chosen projector for the task is a Texas Instruments DLP 4710 EVM full HD projector with high refresh rate and sufficient lighting power[4].

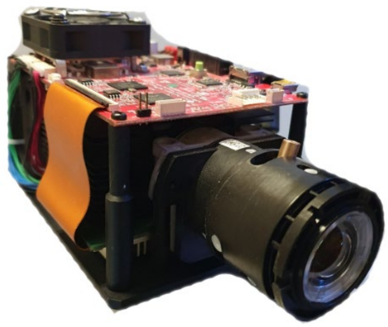


Figure 4.4: The projector
Source: Own creation

[4] Additional information is found in the appendix.

The software orchestrating the hardware will be written in C++ and C# with the Intel RealSense SDK giving access to the camera's sensor data and the OpenCV library providing image processing functions. The computer running the software was chosen to be a Dell OptiPlex 3040 running embedded Windows 10 with its compact size and relative high performance. The components were fastened to a frame which will serve as a test bench for the thesis.

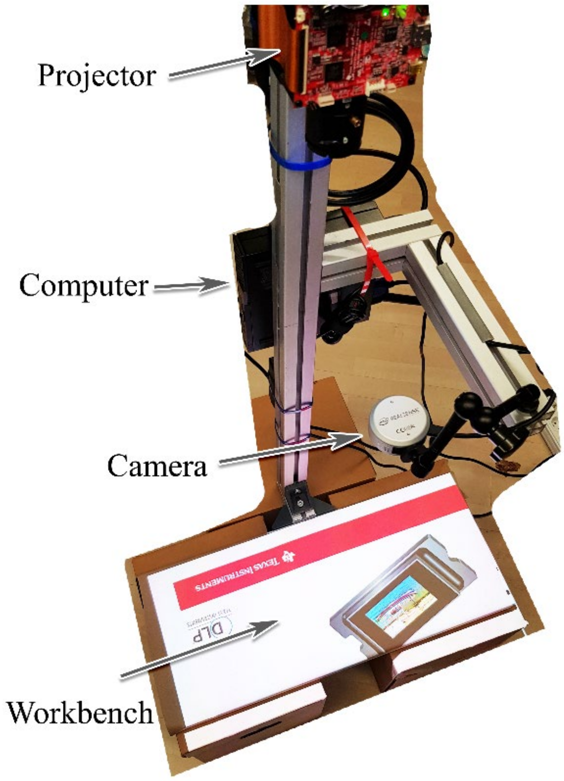


Figure 4.5: The test bench assembled

Source: Own creation

4.3. PRINCIPLE TESTING

The software was initially written in C++ as it is the language the Intel RealSense SDK and OpenCV uses, with smaller tests done in Python with relation to OpenCV. It was vital to check whether the camera's colour and depth resolution is sufficient for accurate measurements, therefore testing had to take place before committing to the whole software. [15] [16] [17]

The camera produces a full-HD colour image with a depth resolution of 1024x768. Colour data is stored in a 3 channel 8-bit image, while the depth data is stored in a single channel 16-bit image, where the values represent the depth (in a non-linear fashion). The SDK gives access to the properties of the camera, to the live feed of the sensors and to basic functions which help with decoding the depth data. OpenCV is a free tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and other image processing. C++, unlike C# or Visual Basic, does not have built-in graphical user interface (GUI), therefore displaying visual information requires external libraries. OpenCV includes a lightweight UI called the HighGUI module, but this only allows rudimentary visualisation options. Once the application working principle is proven, a more capable GUI will have to be chosen. [18] [19] [20]

Intel includes a program for displaying live image called the RealSense Viewer. With the help of this program, the sensors' basic capabilities can be tested. The depth image can be displayed in a greyscale format, which is easier to work with but is difficult to distinguish objects by eye, or by converting the depth to an RGB image where the values are put onto the colour spectrum.

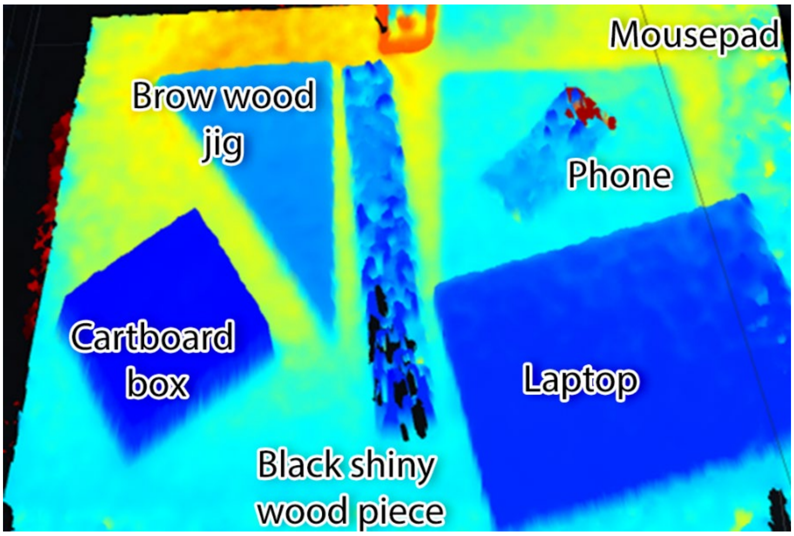


Figure 4.6: How different types of textures are seen by the sensor.
Noise means an uncooperative surface.

Source: Own creation from the RealSense Viewer application

Black spots are where the sensor could not get a reading of the depth value. This could happen because of three main things: The surface is non-reflective, and the IR signals emitted by the camera do not get reflected back; the surface is highly reflective, and the signals get bounced around which results in incorrect readings; or an interference occurs from a different, stronger IR source, such as the sun. It is normal to encounter a small percentage of incorrect readings from an image, especially when working with uncooperative surfaces, such as metal.

As seen on the image, noise is to be expected from some of the objects. This can be reduced by the use of a temporal filter, but this introduces artifacts and slows down the process significantly. [21] [22]

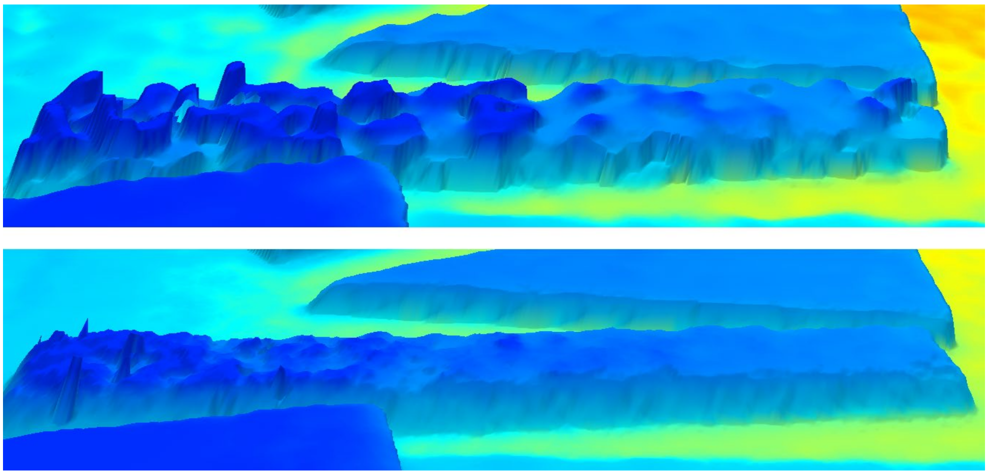


Figure 4.7: Before and after applying a temporal filter

Source: Own creation from the RealSense Viewer application

Accessing the frames from the created application requires a pipeline to be opened with the camera. Since the camera houses its own processing unit, it is usually not dependent on the processing capabilities of the host computer, but this also means that an asynchronous communication has to take place. After the pipeline is established, the program can listen for new frames and processing can take place. To use with OpenCV, the depth image has to be converted from the Intel proprietary *depth_frame* type to a 16-bit matrix. After converting the depth image from the Intel format to OpenCV matrix, it can be displayed. Light colours represent objects further away, while darker colours represent objects closer to the sensor. To help with visualisation, the image contrast and brightness can be increased. [23]

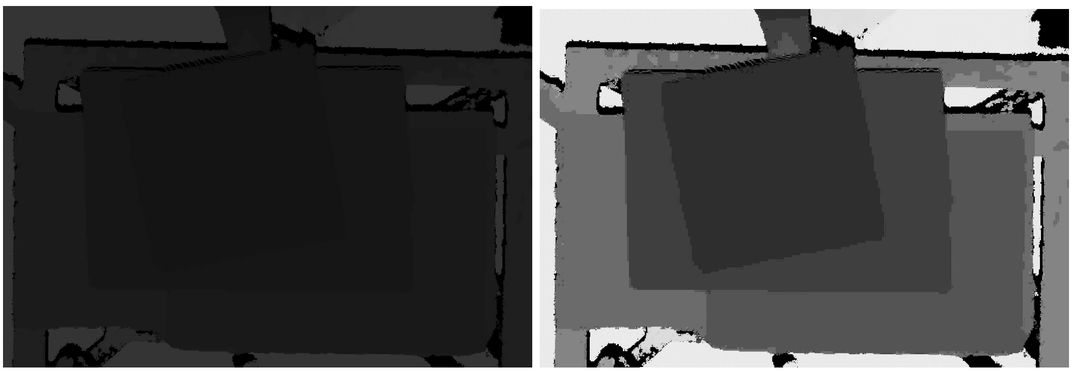


Figure 4.8: Raw and adjusted depth image

Source: Own creation from the C++ test application

The application now has access to the depth image of the camera, which raises a question: Why does it need the depth image in the first hand? It is clearly not usable for surface inspection, as there is no colour data. The camera does have an RGB sensor, but the two sensors are slightly out of alignment. Fortunately, the SDK provides a function which aligns the depth and colour image, making a near perfect overlay of the two.

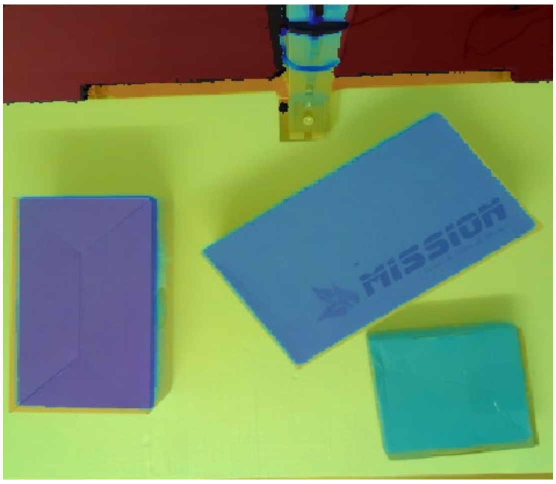


Figure 4.9: Aligned depth and colour frames with 50% overlap.
Flat colours represent depth data.

Source: Own creation from the RealSense Viewer application

As for the usefulness of the depth frame, image processing can take place on both the colour and the depth image, and since they are aligned, measurement coordinates from one frame can be translated to another. The depth image can be used as a mask in cases where the colour image could not determine the object's properties. This substitutes the task where traditional 2D cameras would require different lightning to get the contour of an item.

Take the following example: To measure the size of the box, a clear contour has to be achieved. Looking at the colour image it is clear, that the background and the lightning is sub-optimal, making it a very tedious process to separate the box from the image. The depth image is not affected by any of those factors; therefore, a clear separation is visible.

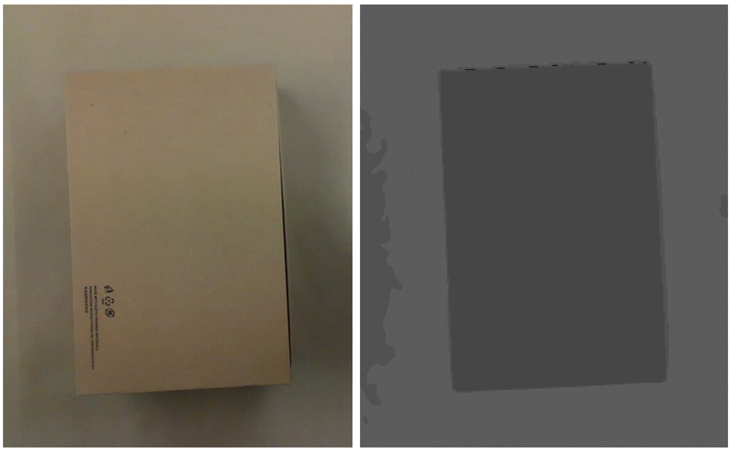


Figure 4.10: Colour and depth image of a box

Source: Own creation from the C++ test application

To create a mask, the depth image has to go through a number of steps. First, a small Gaussian blur is applied to even out the values. Then, a threshold filter[5] is applied with a value just above the workbench's height. This thresholded image is then eroded to minimise noise and dilated back to its original size[6]. The resulted binary image then can be used to mask out the background of the colour image, giving a clear cut-out of the box in RGB.

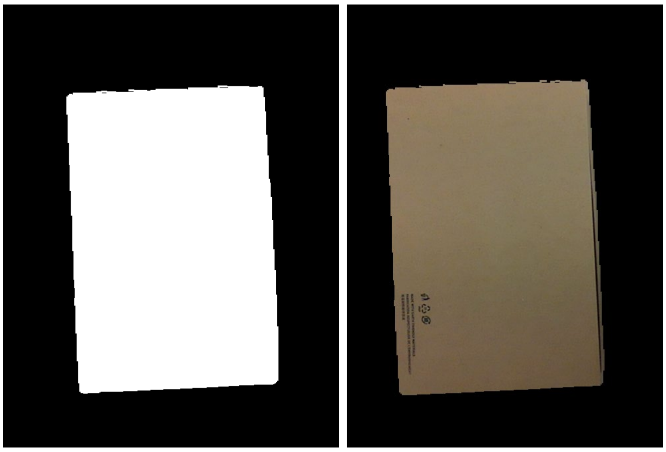


Figure 4.11: Thresholded depth and masked colour image

Source: Own creation from the C++ test application

[5] Thresholding means that anything below a given value gets discarded and values above get turned to white, resulting in a binary image.

[6] Erosion narrows the shape while dilation grows it with a given radius.

Additional benefit of the depth image is that to calculate the size, no previous calibration is required. To measure the size with the 2D image, a calibration board could be used with known distances, but higher the box, the bigger the size distortion would be due to the nature of perspective.

Figure 4.12: The same boxes at different heights

Source: Own creation

The SDK provides functions to gather absolute distance values from the sensor, and by subtracting the box’s distance from the workbench’s distance, the box height can be calculated. To get the points where the distance has to be measured, the shape of the object has to be determined. The canny edge detection algorithm helps with finding the contour and by approximating a polygon to the contour, a shape can be determined. This polygon is made up of a number of points, depending on the strictness of the parameters, and by counting the number of points, the shape can be determined. Further calculations can be done by measuring the angle and distance between the points and object recognition can be achieved by comparing these measured values to pre-defined ones.

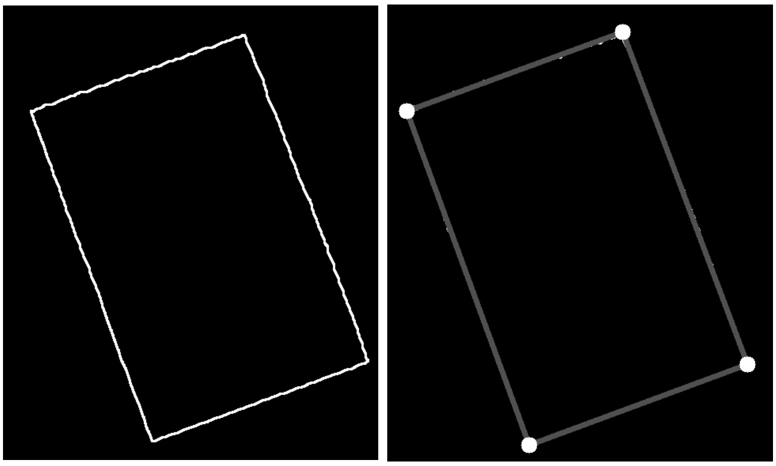


Figure 4.13: Canny edge detection and polygon estimation

Source: Own creation from the C++ test application

Relative measurements can take place on the colour image, but by using the depth information, absolute values can be determined. The values are printed out for showcasing, with side values being white, and diagonal values being grey. The box's true size is 34.5 cm x 23 cm. The slight undershooting is due to the erosion being slightly larger than the dilation. This has to be done to improve the accuracy as the corners of the polygon may be in a point where falloff happens[7].

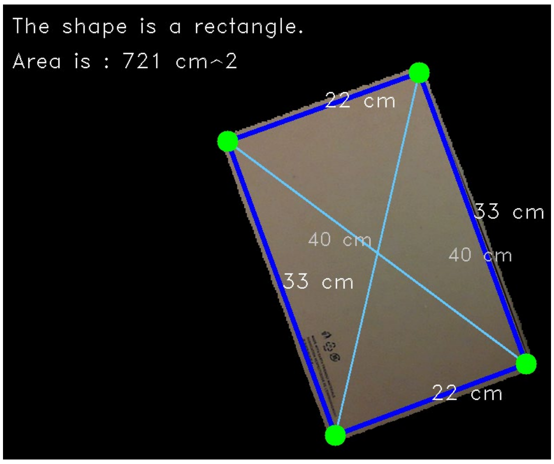


Figure 4.14: Measurements by the application

Source: Own creation from the C++ test application

[7] Sharp edges have blind zones beneath them, especially with tall objects. Reading depth values at these points can either give the distance of the box, the workbench, or completely invalid numbers. Example image of this phenomenon is found in the appendix.

This proves that the camera can successfully be used as a measurement unit, the depth image is both accurate within a reasonable margin and stable enough to provide a mask. A second threshold filter can be applied to the image with inverted functionality, where images closer than a value can be truncated. This way taller objects can be left out of the image.

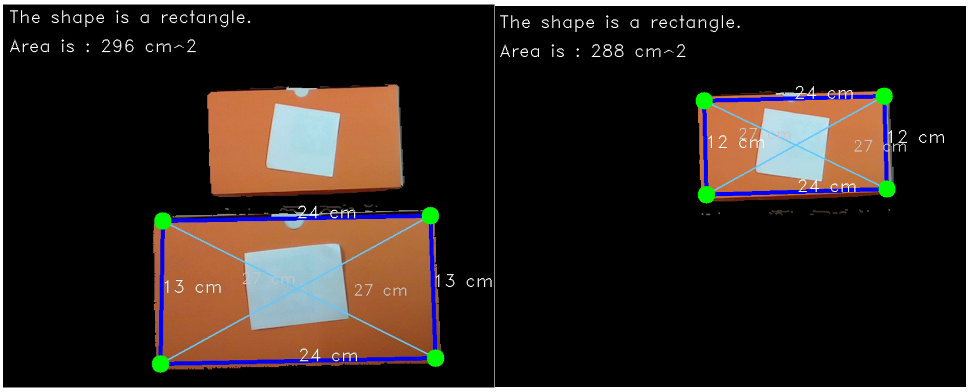


Figure 4.15: Taller box thresholded out.

Note that the size of the box remains the same, even though the taller one appears to be larger.

Source: Own creation from the C++ test application

Some problems were encountered during the testing phase. One of them was separating the workbench from the inspected item. Thresholding works by binarizing an image at a given value, but if the table itself is not parallel to the camera, different parts will be at different distances, which in turn means different colours.

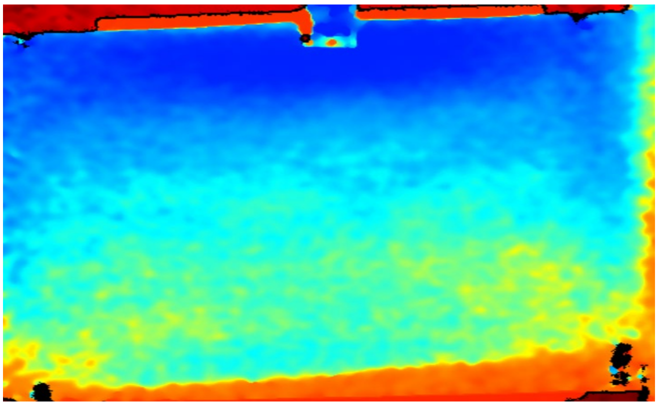


Figure 4.16: Distances of the workbench when the camera is not parallel to it

Source: Own creation from the RealSense Viewer application

This can be solved by a number of different ways:

- Set the camera parallel to the workbench

This is the easiest solution, but since the aim of this application is to be as versatile as possible, it brings more problems with deployment than it helps with programming. Not every workbench is flat as a glass panel and other fixed objects could also impose interference. Even though the lens distortion is hard coded into the camera, some impurities still remain, also posing problems.

- Implement a multi-level thresholding function, where different parts of the get thresholded by their distance

This would eliminate the problem of the camera not being parallel, but still wouldn't solve the problem of interfering objects found in the work area. Depending on the amount of local thresholding blocks used, processing power would also be significantly impacted.

- Creating a clean plate of the workbench and subtracting that from the live image

This solution is what the application ultimately uses, as it solves most of the problems. A photo is taken of the clear, empty workbench and gets stored in a file. The live image is then subtracted from this clean image, and only the difference of the two image remains (Which is anything that was not present on the clean plate). This process both solves the non-parallel and interfering objects problem but requires an initial calibration step. This also means that if the camera gets moved or the workbench gets changed, the calibration has to be done again. This is not a significant problem, as it can be implemented to compare the clean plate to the empty workbench on system start-up and create a new plate if a difference is found. As this calibration step only requires a single frame and the amount of time it takes to save the file, it is a negligible time addition.

The box detection has been implemented by using the contour of the object, which works well for boxes, but the contents of the box require a higher level of checking. Different methods are available with different performance and accuracy properties. One of the simplest ways of checking the presence of a pre-defined object is to compare the live feed of a pre-determined image, similarly to the clean-plate concept. When the box is being learnt by the system, the objects are placed in their corresponding locations and between each step, both a colour and depth image is taken. These images are then compared to the live image of the camera and by calculating the difference, an estimation is made. The simplest way of comparing two images is by using the *norm* function in OpenCV, which calculates the difference norm[8] of two image arrays. This method compares the identical pixels of the two images, therefore comes with a lot of drawbacks.

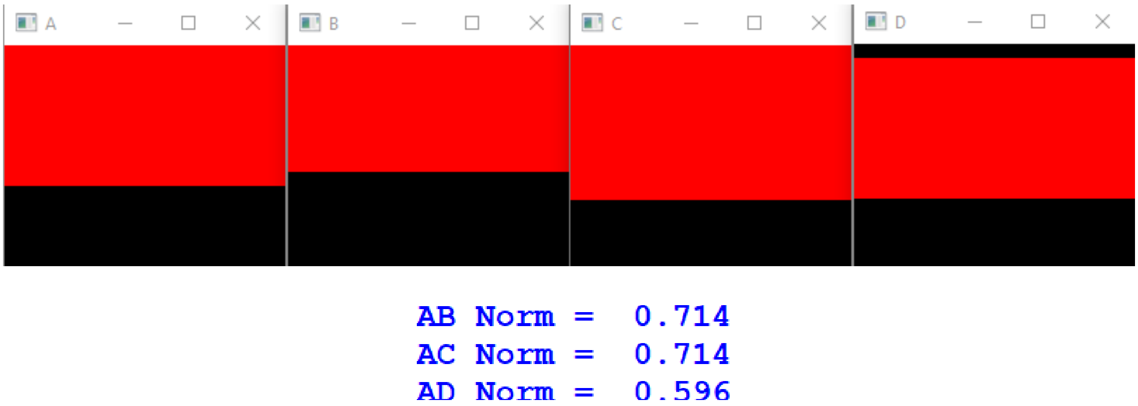


Figure 4.17: Comparison of 4 slightly different images using the *Norm* function

Source: Own creation from the Python test application

While the brightness of the pixel has less effect on the final value, once the images get shifted to a different location, the similarity value gets decreased drastically. If the image is altered in a significant way, such as rotation, the result will be unusable.

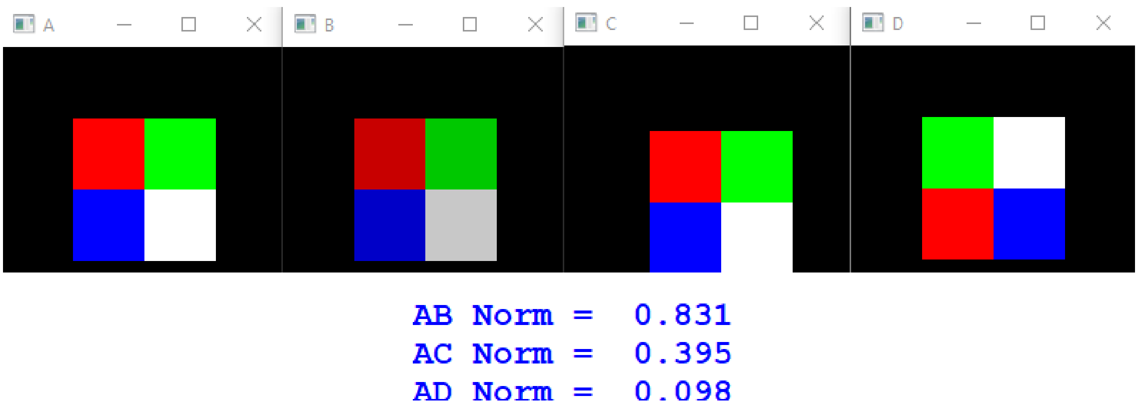


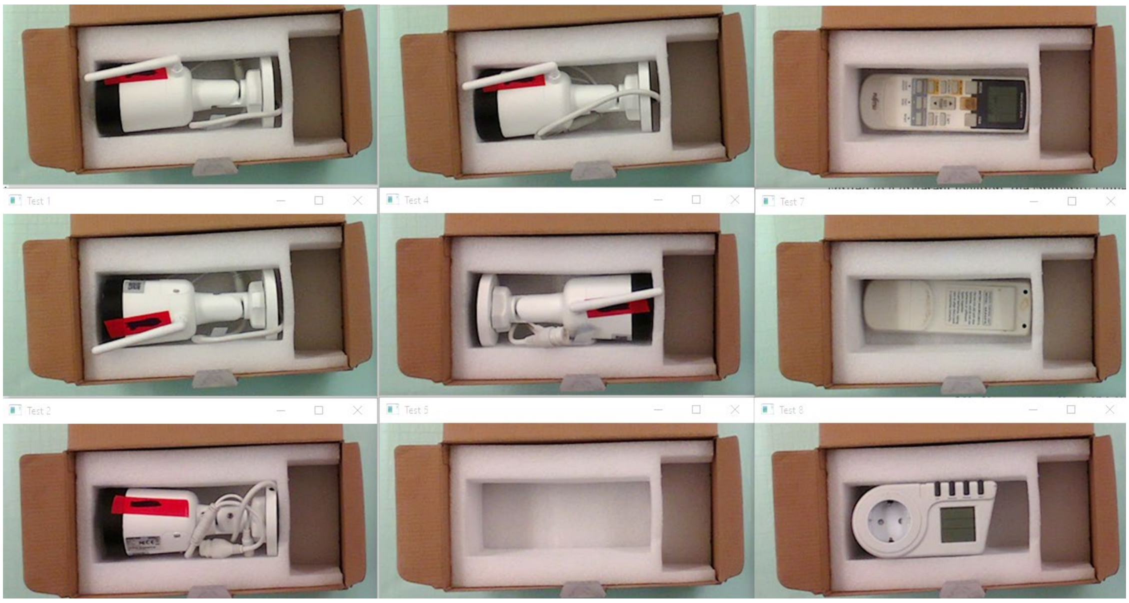
Figure 4.18: Original, darkened, shifted and rotated image

Source: Own creation from the Python test application

[8] The norm of a mathematical object is a quantity that in some way describes the length, size, or extent of the object. In the case of images, it describes its properties and its content.

This method is not reliable for a live environment, where the location and angle of the tracked object gets changed constantly, with different lightning depending on the time of the day.

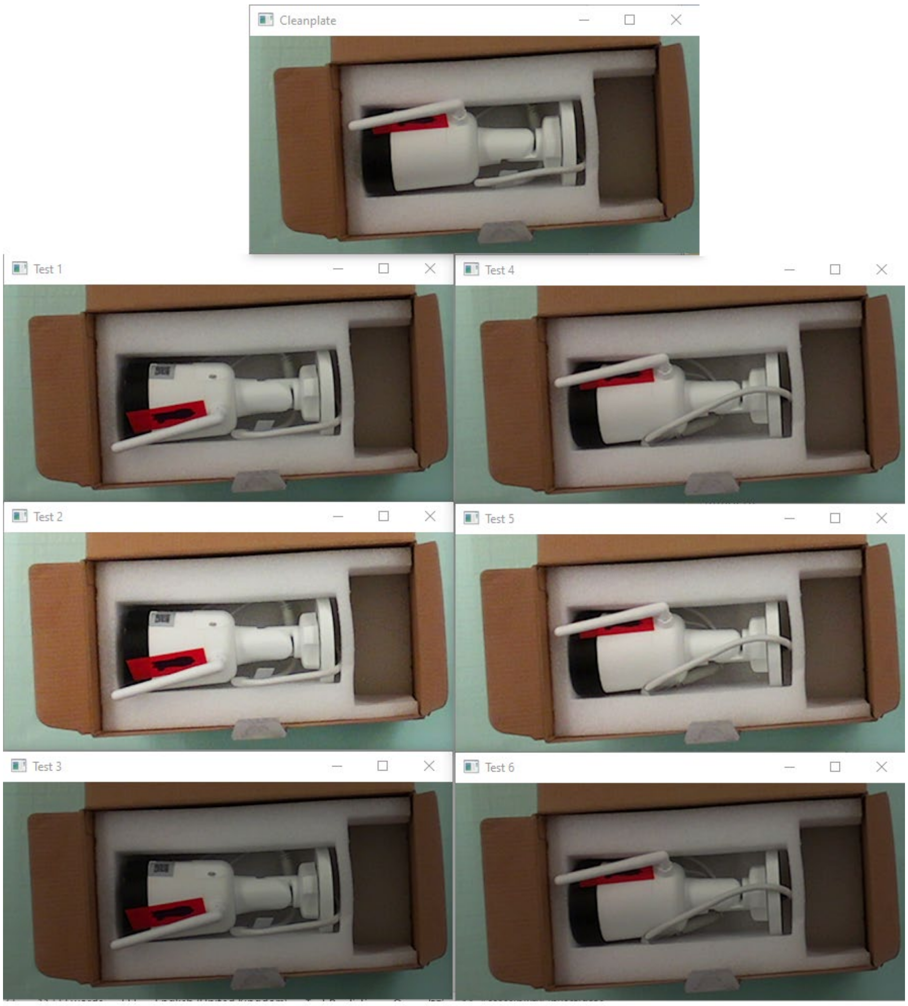
To overcome the problem of different locations and angles, the image can be cropped down to the size of the box and rotated so it always stays upright. The pixel-shifting issue can be solved by comparing the histogram of the images based on their colour. This only compares the colour difference of the whole image; therefore, the shape does not matter. This could pose an issue if two products are similar in colour, so a test was put together. A number of images were compared, and the results were recorded with the images visible below. The top left image is the so-called clean plate image, and the others were compared to it. The second and fourth images were both within 90% of the original image. The bottom left and middle images were purposely put in a wrong orientation and were not determined to be similar. The middle right image was found to be rather similar incorrectly, which shows the weakness of comparing just the colour histogram.



```
base_base Similarity =  1.0
base_test1 Similarity =  0.936
base_test2 Similarity =  0.250
base_test3 Similarity =  0.908
base_test4 Similarity =  0.438
base_test5 Similarity =  0.247
base_test6 Similarity =  0.525
base_test7 Similarity =  0.846
base_test8 Similarity =  0.217
```

Figure 4.19: An array of images and their histogram comparisons

Source: Own creation from the Python test application

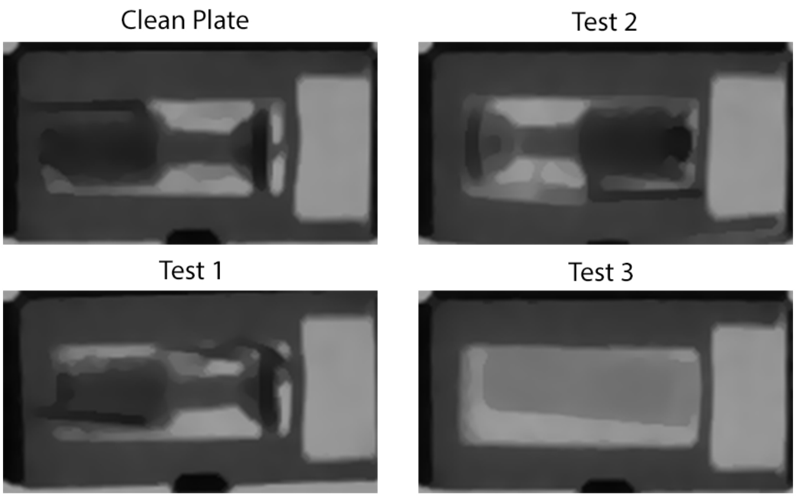



```
base_base Similarity =  1.0
base_test1 Similarity =  0.936
base_test2 Similarity =  0.756
base_test3 Similarity =  0.921
base_test4 Similarity =  0.908
base_test5 Similarity =  0.847
base_test6 Similarity =  0.896
```

Figure 4.20: Adjusted brightness of two sample images

Source: Own creation from the Python test application

Adjusting the brightness of the two correct samples show a slight decrease in reliability, however the live image's brightness can be adjusted to have the same tones as the clean plate. Without adjustment the remote turned upside-down would still be classified as accepted at 75% acceptance rate on the colour histogram, but the same calculations can be done on the depth image as well to improve the accuracy.




```
base_base Similarity =  1.0
base_test1 Similarity =  0.883
base_test2 Similarity =  0.897
base_test3 Similarity =  0.774
```

Figure 4.21: Histogram of the depth images

Source: Own creation from the Python test application

There is a considerable difference between the images with the camera and the one with the remote, but this would still pass with the 75% acceptability value. To overcome this problem, the image should be cropped in even more to only observe the small cut-out of the foam. If a box requires multiple objects to be placed, first the location has to be identified where an object was placed, and once that is determined, the checking can occur. The key of having a high level of accuracy is taking smaller samples where the differences are expected rather than using the whole image as one. Dividing up the inspected area into smaller chunks and measuring the local difference gives a higher reliability. The test also consisted of mostly white objects placed on a white foam, which is fairly uniform, making measurements purposefully more difficult. If the inspected object is made up of more colourful components, recognition is easier and more reliable. Shape detection is commonly used in situations where the object is placed in a fixed location, but due to the wide range of accepted angles of a given object reliability would be low in this case. The principle would be similar to the box recognition detection by estimating the polygon of the object, saving the corner angles and finding an object with similar parameters. [24] [25]

In case an object is determined to be at a bad location incorrectly by the software, the human worker should have the opportunity to override the system (while also logging the process). As previously defined, a projector shall be used as an output device and to avoid having to use a keyboard/mouse, hand-tracking can be implemented as an input device. With the use of thresholded depth image, the hand shape can be easily separated from the background, further helping implementation. The previous RealSense SDK included a built-in hand recognition function, but this was not upgraded to the latest version. However, OpenCV also provides options to implement this function with the help of the MediaPipe library. MediaPipe is a framework mainly used for building audio, video, or any time series data. With the help of the framework, versatile pipelines can be created for media processing functions, such as hand detection. This function was not implemented in the final software due to incompatibility, but initial testing proved promising. [26]

The basic concept of using a depth camera has been tested out with sufficient results, which leads to integrating the projector to work with the system.

4.4. CREATING THE APPLICATION IN UNITY

Since the GUI of the included OpenCV library is very basic, a different library or method had to be found to test the projector. A large set of graphical interfaces are available to the C++ language, but Unity was chosen as a development platform in the end due to seemingly being the most versatile to work with. The RealSense SDK has a sub-library which ports the functions into C# and some of the camera's parameters can be accessed through the graphical interface of Unity. OpenCV also has a C# implementation, but that implementation is based on a Java implementation of the C++ one, which means that some of the functions don't work quite how they should, and another few are not included at all. The MediaPipe framework is also not directly supported by Unity and C#, making hand-tracking more difficult. These shortcomings were found well into the development of the application, therefore changing to a different development platform was not practical. An alternative and more capable choice would be to stay with the C++ libraries and use a GUI designer like Qt.

Unity provides a lot of useful spatial functionalities as it is primarily designed to be a game development platform. Both 2D and 3D elements can be created with ease and functionalities can be implemented through both a graphical programming interface and through writing code in C#. Unity can also export to a number of platforms, making multi-platform applications possible.

The application has the following menu structure:

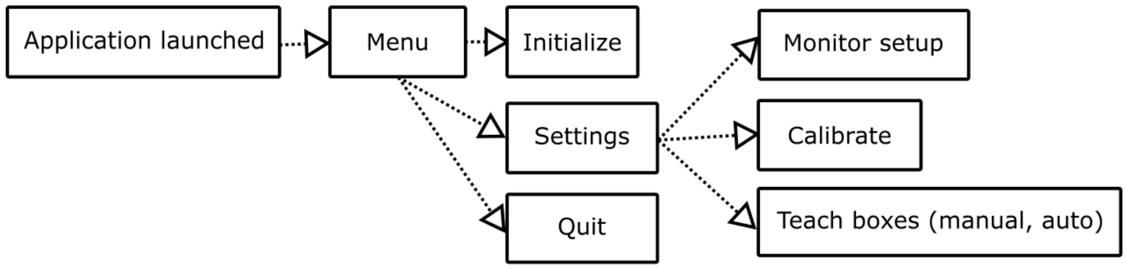


Figure 4.22: The menu structure of the application

Source: Own creation

There are three main menus available from the Settings menu:

- Monitor setup is responsible for allowing the user to choose which connected displays should display what information. The application requires at least one connected display, but two is preferred. The projector output is responsible for displaying information onto the workbench, while the main monitor output is there to give additional control over the application if needed.

Figure 4.23: Display selection menu

Source: Own creation

- The calibration step is required to be done by the operator before using the application. This step includes creating the clean plates of the workbench with both sensors and also selecting out the total working area, or so-called area of interest (ROI). This is done by clicking four times on-screen to select the corners. Anything outside of this area will not be considered by the application. This is an optimization step as the camera has a wide field-of-view, which helps with making deployment easier, but impacts performance by also processing areas which do not require scanning. The calibration also includes a step where the projector screen is synced to the camera's screen. This is necessary as the coordinate system of the projector has to be translated to the camera in order to display the necessary shapes onto the objects at the right location.

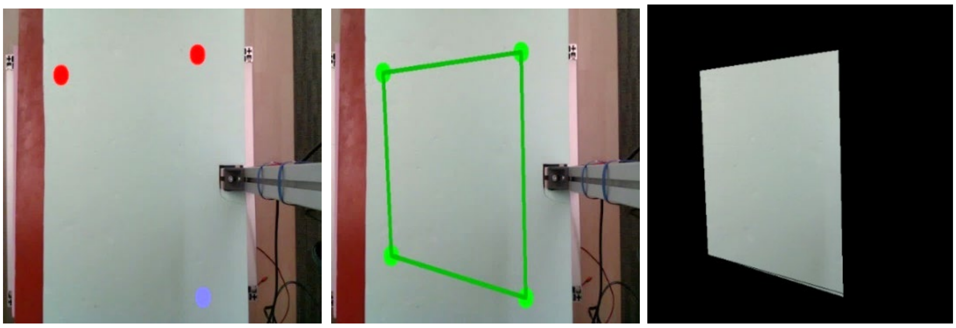


Figure 4.24: Selecting the work area of the camera

Source: Own creation from the Unity-based application

Calibrating the projector to the camera was tested in four ways:

1. Manually selecting the edges of the projector screen to fit the camera similarly to how the ROI is selected. This is the simplest way, but it also calculates the distances in a linear fashion, not accounting for the distortion of the projector lens.
2. By displaying a square in the middle of the projector and growing it automatically until the shape stops growing from the perspective of the camera. This would mean that the shape is either out of the camera's or the projector's range. This is repeated on each axis and each corner until the calculation is done. This process is fully automatic but takes a considerable amount of time. Its accuracy was also found to be unreliable, and another option was assessed.

3. One of the most reliable ways of calibrating the projector to a camera is by using homography, also called plane matching. This process uses a checkerboard image to calculate the intrinsic and extrinsic[9] parameters of the camera. This method can take the distortion of lenses into account, making it highly accurate, however the projector does not have a sufficiently large cover of the camera image in the current setup, as multiple images have to be placed to fully map the camera. Testing did not provide sufficient accuracy, and without change of hardware, this method was deemed unusable. [27]
4. A different method was developed based on the checkerboard method, where the shortcomings of the small display size are addressed. A white rectangle is displayed onto the centre of the projector screen with a known pixel size. This is measured by the camera and the size is calculated based on the width/height of the rectangle. Skewed images are also compensated by using the vector of the origin and each corner. The method is fully automatic and works in a range of setups while the initial box is fully visible by the camera. One shortcoming of this method is that it does not account for lens distortion unlike the third option, however testing proved that distortion is less than 5mm from the centre to the edge of the image.

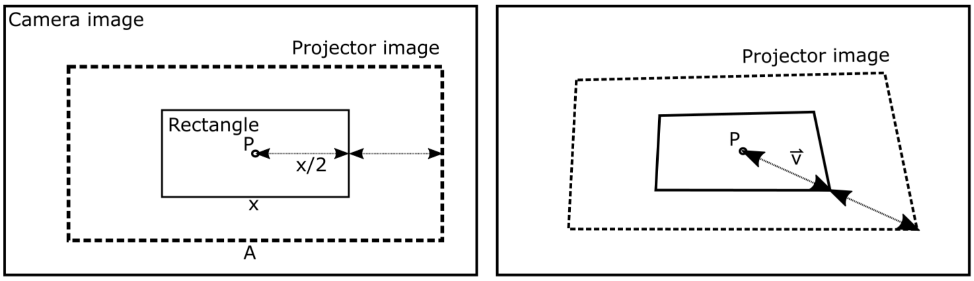


Figure 4.25: Projector calibration methodology at 50% scale

Source: Own creation

[9] Also called internal and external parameters, these values help with calculating where a 3D point is in the 2D image.

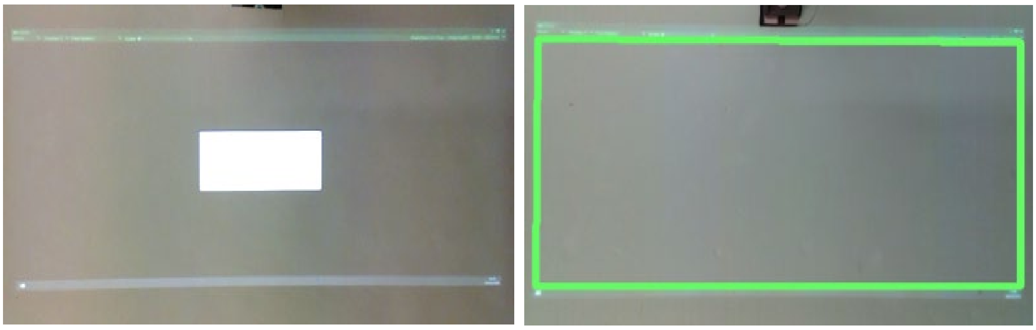


Figure 4.26: The projector calibration sequence and the calculated area

Source: Own creation from the Unity-based application

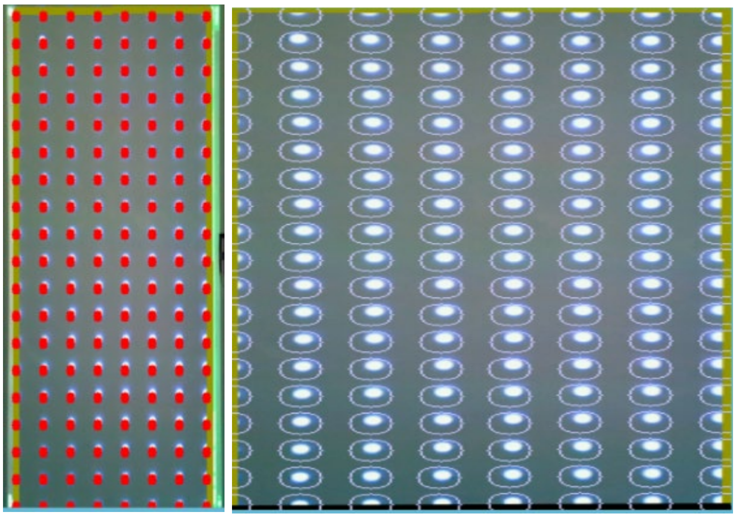


Figure 4.27: Point estimation of the projector.[10]

Source: Own creation from the Unity-based application

- The application can learn the size of the boxes in two ways:
    1. By manually measuring and entering the values into a database where the information is stored. This makes remote configuration possible, but the contents still have to be taught on location.
    2. By letting the application measure the box's parameters and store them by itself. Multiple measurements are taken of the box at different positions and angles and an average of the values is stored.

[10] Red circles are drawn onto the camera image. The blue hollow circles are drawn onto the projector image as an estimation where the red circles should be placed. The blue filled circles are the actual drawn circles by the projector.

Once the box is learnt, the contents can be put into the box, one at a time. The application takes an image of the empty opened box and between each step. By subtracting the latest image from the previous one the difference is found, and the ROI is located.

Figure 4.28: The boxes can be also digitized by the learning process.

Source: Own creation from the Unity-based application

Once the output displays are selected, the clean plates taken, the projector calibrated and the boxes learnt, the main part of the application can be initialized.

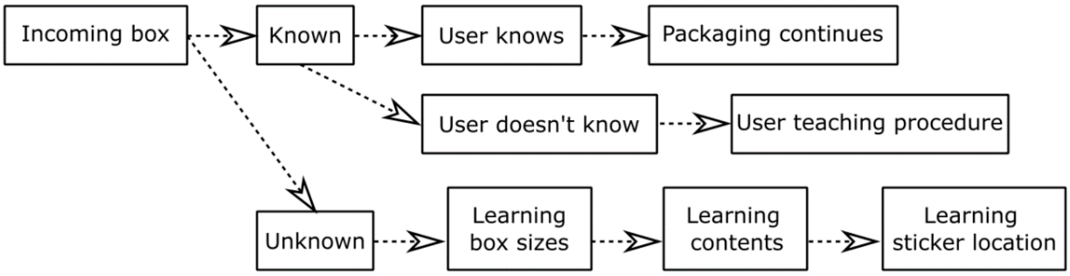


Figure 4.29: Possible steps of the application

Source: Own creation

Reflecting on the initial goal, the final application has the following tasks:

- Quality control: Checking whether the correct item was placed into the box.
- Logging: Storing the images for further quality assurance.
- Assistance: Showing the correct item placement for the worker

If the setup allows the identification of each individual user[11], their progress can be tracked. In case a box which they have not yet encountered, the application can

[11] Can be implemented by logging into the computer, by using an RF card or by using the camera as an identification method by scanning their card if the resolution allows it.

automatically load up a training sequence where the packaging order is accompanied by additional information (such as orientation of objects, what to look out for, etcetera). The logging process keeps track of the performance by measuring the time for each step and the number of errors which have been made. Slow processes can be recognized and by comparing the performance of different users, the source can be identified. Slow performance could be attributed to user error in case of a single difference, but uniform slowness may mean an inefficient way of packaging. Analysing these performance numbers can be highly valuable to a company.

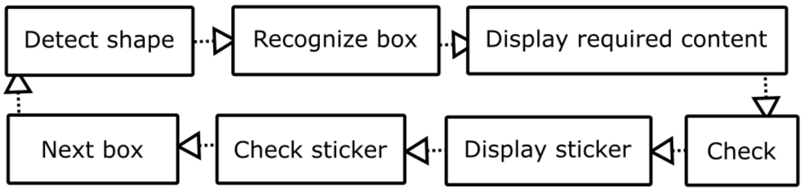


Figure 4.30: The packaging process steps

Source: Own creation

5. MEASUREMENT RESULTS, ANALYSIS

5.1. PERFORMANCE OF THE SYSTEM

The system is responsible for quality checking, therefore a substantial amount of liability is involved. The application has to be tested and a margin of error has to be established. There are three main components which include variability:

- Box recognition
- Content checking
- Projector alignment

The box detection algorithm measures the size of the box by creating two separate shape detection matrixes. One is responsible for measuring the sides of the box, while the other is responsible for measuring the height of the box. Both require different erosion and dilation values as due to the blind zone effect of the depth sensor, the points found in the side measuring matrix are not stable enough to measure the height of the box. The height is measured at various locations and an average is taken to even out any possible noise.

The current implementation works with fixed erosion/dilation values and does not consider the height of the object. This introduces slight inaccuracies with objects with different heights. Two similarly tall boxes are identified within 1.5 cm of their true width size, but if a taller box is measured with the same values, the precision drops measurably. If the setup is configured for boxes with tallness range from 5 to 10 cm, a box with a tallness of 15 to 25 cm encounters a width measuring error of 3 to 6 cm. If the setup is adjusted to the height, the accuracy is brought down within 1.5 cm, but the shorter objects' measurement error increases.

This can be fixed in two ways:

1. First measuring the height and adjusting the erosion/dilation parameters based on the value. This requires the measurement of the perspectival distortion to calculate the adjustment value.
2. By storing the Intel depth frame type and measuring the side values on the workbench by translating the coordinates from the live images. This does not require additional calibration, but the C# port of the RealSense SDK does not allow direct access to the sensor values. This problem will be further addressed in the following chapter.

The content checking is based on the previously described method of comparing the histogram of the base image to the live one. The implementation required two notable extensions: a matrix divider and a super-sampling function for stabilizing the image. To localize the histogram of the image to smaller blocks, a function was created which divides the base and live matrix to a 3x3 block. These blocks are then compared to each other. This is necessary as with the basic histogram comparison, the whole image's content is measured. This means that if a blue object is placed at a different location, but still present in the image, the histogram could not be used to recognize the error. By dividing the image into smaller blocks and comparing the respective blocks together, both colour and location data can be tracked to a degree. The number of blocks used greatly increases the trackability of an object, but if the image encounters slight shifts, the recognition reliability drops significantly.

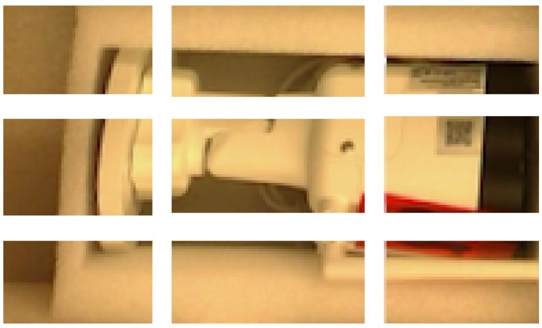


Figure 5.1: Block images of the base image

Source: Own creation from the Unity-based application

Due to the noise of the edge detection, the cropping of consequent images encounter slight differences. To address this issue, a super-sampling function was implemented, which averages the crop points of the image. This smoothens out any sudden movement at the cost of a slight additional settling time.[12] Both functions greatly improve the accuracy and reliability of the content measurement step.

To examine the stability of the content checking process, the following test was done: Similarly to the testing on page 37, a number of objects were placed and measured with the additional functions implemented. Three different object placement was pre-determined to be OK, while two different placements were purposefully put in the wrong way. Three additional objects were also placed into the box with the addition of the empty box, to check the capability of item swapping detection. Each object was measured at 3 distinct locations with at least 4 consequent measurements. These values were then averaged by location and by object. The original object which was serving as the base image was recognized with 98% on the same location but with only 62% on the edge of the workbench. The average of the accepted boxes was 72.7% with the worst valid performance at 50.4%. The negative average was 31.9% with the highest resemblance at 49.7%. The accepted and declined values are technically separated at the 50% line, but a difference of only 0.7% is not acceptable.

[12] When a box is put under the camera, the application requires a settling time of ~2 seconds when the measured numbers' average reaches an equilibrium.

	1	2	3	4	5	AVG	AVG/OBJ
OK							
original	98	98	98	98		98	84.53
original	94	93	93	92		93	
original	62	64	62	62	63	62.6	
r1 ok	84	81	82	82		82.25	70.55
r1 ok	65	63	63	65	65	64.2	
r1 ok	65	67	67	65	62	65.2	
r2 ok	49	51	52	50	50	50.4	63.07
r2 ok	72	72	73	74	73	72.8	
r2 ok	66	66	65	66	67	66	
NEG							
r3 neg	41	41	38	40		40	36.42
r3 neg	20	19	19	20		19.5	
r3 neg	51	48	49	51		49.75	
r4 neg	47	47	45	46		46.25	37.33
r4 neg	20	21	20	21		20.5	
r4 neg	43	48	45	45		45.25	
empty	32	32	32	34	33	32.6	34.95
empty	35	38	37	36		36.5	
empty	40	27	39	37		35.75	
remote	13	12	12	13		12.5	22.42
remote	39	42	41	41		40.75	
remote	15	14	14	13		14	
remote upside	18	19	19	18		18.5	24.50
remote upside	38	36	37	37		37	
remote upside	18	18	18	18		18	
plug	38	37	37	37	38	37.4	35.98
plug	37	38	39	37	38	37.8	
plug	32	33	32	34		32.75	

AVG OK	72.72
AVG NEG	31.93
Min AVG OK	63.07
Max AVG NEG	37.33
Min OK	50.40
Max NEG	49.75

Figure 5.2: The measurements' result table

Source: Own creation from the Unity-based application

To understand the source of the inaccuracy, the images have to be compared. The depth images did not encounter any significant differences based on the location, but the colour images experienced a gradual alteration due to different lightning angles. The boxes were moved in a work area size of 50 x 70 cm; therefore, lot of external variabilities can be introduced. It is seen on the sample images that the original base image is much darker than the images which were taken at the edge of the workbench. Additional reflection is also visible, further decreasing similarities.

Figure 5.3: Original, original at the edge, rotated at edge and wrongly oriented image.

Source: Own creation from the Unity-based application

The wrongly oriented image also has a lot of similar colour components in the same area, which can deceive the histogram comparison.

Three solutions are available which can improve the accuracy of the measurement:

1. Reducing external variabilities by reducing the work area. This depends on the items being packaged and the variety in size of allowed boxes.
2. Instead of automatically creating a 3x3 mesh of the image, parts of the object could be manually selected based on identification points. This requires manual selection of areas-of-interest on each step by a person.
3. Currently the depth images' results are averaged with the colour images'. This could be adjusted by weighting the depth image result, making colour difference less controlling.

An additional problem limited to the Unity based application is that the sensor settings are not accessible, therefore exposure, brightness and white balance has to be on automatic. This is highly unwanted as even the slightest automatic image processing can alter the results.

The projector alignment calibration process provided a surprisingly accurate result, no adjusting was required. A slight non-linearity can be noticed due to lens distortion of the projector, but even at the most pronounced areas it only contributes to a misalignment of 3 mm on the workbench. However, due to the non-aligned camera and projector, a height misalignment is quickly observed.

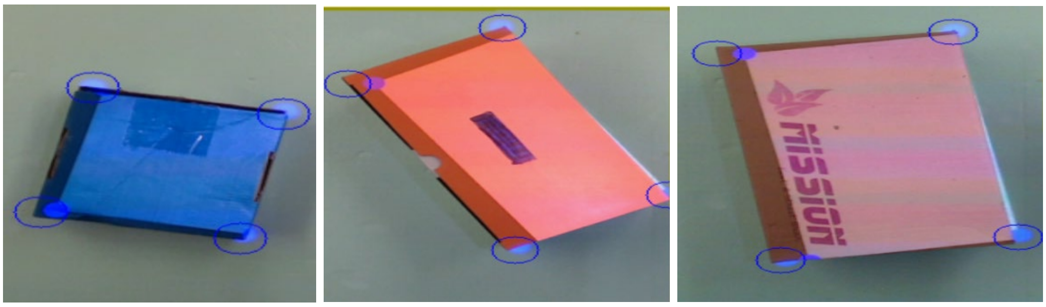


Figure 5.4: Projector height misalignment

Source: Own creation from the Unity-based application

This error is due to the projector not considering the point's height. By observing the figure above, it is visible that the camera correctly recognizes the point, but due to perspectival distortion the same point is at a different coordinate on the workbench level. The projector then projects its image onto that coordinate, which further gets distorted by the obstructing object. To adjust for this, the height of the object has to be calculated and the projector's coordinate has to be increased by the same amount. This either requires a function which calculates the pixel size, or a pre-set adjustment number based on the height. While the second option is easier to implement, it does not account for non-linear distortions unlike the first one. Another significant note is that the projector's vertical projection image is not symmetrical. The bottom of the screen is parallel to the lens while the top is projected at a high angle.

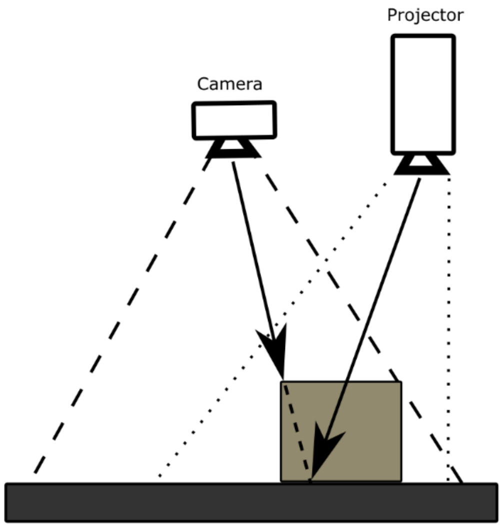


Figure 5.5: Camera and projector misalignment principle

Source: Own creation

5.2. ASSESMENT OF THE SYSTEM

The development of the application was affected by many different internal and external factors. The aim was to create a versatile application which has the general capabilities of checking the quality of the human worker and by providing assistance for them. The biggest internal problem was the chosen development platform, Unity. While Unity is an undoubtedly versatile development platform, it is mainly focused on videogame development. Compatibility with external libraries which were not specifically designed to work with Unity is questionable. While it is possible to use them, they require special implementation of the functions and do not always behave as expected. Documentation is also very sparse in these matters.

The biggest problem was the lack of control over the sensors of the camera with the RealSense SDK. When testing was done in C++, a low-level access was provided, which also meant a higher control over the sensors' image. The Unity port of the SDK provides access to the camera by texture files, shaders and prefab objects. These objects have minimal amount of C# scripts, as they call the functions from a base C++ library, but many of the C++ functions are not implemented. Access to the frames is through texture types, which is acceptable for displaying the image frame, but additional information, like the actual sensor data is lost. In case of the depth image, this means that depth calculations cannot be done directly. The documentation did not provide any insight into this matter, therefore a custom method had to be implemented. This did not affect the accuracy of the sensor but impacted the performance noticeably. Additional filters, like temporal filters could also not be used, which could've helped with more accurate depth measuring [21] [22]. The resolution of both the depth sensor and the colour sensor also had to be reduced to 640 by 480 px, which greatly impacted the measurement performance.

The OpenCV port was more-or-less in par with the C++ implementation, but 16-bit image handling was not implemented reliably. The library also required a monetary investment as it is not free.

These shortcomings were found well into the transition from C++ to Unity and due to time constraints, changing was not practical. The Unity application did provide sufficiently measurable results and with greater control over the internal components, a higher accuracy would be achieved. An optimal choice would be to use the native SDK and OpenCV library and use a similarly capable C++ based development platform, like Qt.

The camera itself provided exceptional performance regarding its depth accuracy. Material uncooperation was not significantly detected while using cardboard boxes, foam padding and plastic objects, but highly reflective materials, like glass components could pose a potential issue. A notable fact is that Intel has discontinued the production of its Lidar based Time-of-Flight cameras, including the L515 camera during the development of this thesis. Intel will continue to produce stereovision cameras for the foreseeable future, but this raises the question: what could be the reasoning behind it? The official statement states a shift from this technology, which implies economical or technical reasons. The bigger problem is that cameras which similar capabilities are non-existent in the price range of the camera used. The closest product is from a sensor called Avia from the company Livox, but it costs four times as much with a completely different range and accuracy target and does not come with an RGB sensor.

The projector also proved to work greatly, even though its brightness was only set to 50% of its nominal rating. The only issue is its low field-of-view, which could be changed with either putting the projector higher up, or by different optics. The height option greatly depends on the current setup, while a different lens would most likely account for a greater lens distortion. However, with a greater FoV the homology method could be used which would solve that issue while also providing exceptional accuracy.

The biggest external factor is the fact that the application has to work with humans. Every inspection method aims at minimising the number of external variables (such as uncontrolled lightning, position differences of the object and the camera itself, etcetera), but any work involving humans involve a certain degree of uncertainty. It is possible to create an application where the human worker has to attend to strict rules about placement, lightning, shadow control and such, but that means that the person is helping the system, not the other way around. The measurements were affected more than expected by the large workbench size. In case of the orange box, which was showcased, the workbench was 10.7 times larger than the area of the box. The measurements were done in the entire available area, thus a very wide range of external lightning changes should've been accounted for. The workbench was tested in a light-controlled environment with fluorescent lamps providing even lightning, but as seen from some of the examples, some uneven reflections did impose problems. This could be minimized by reducing the work area depending on the size of the inspected box.

An additional way of improving the accuracy of the measurements is by instead of using histograms, dense vector representation could be used. This technique uses machine learning to classify objects but requires a database of types. Public databases are available, but they mostly consist of rudimentary objects like cats, vehicles, flowers, chairs, etcetera. To build a reliable database of the measured objects, thousands or even tens of thousands of images would be required, making setup highly time-consuming. [28]

The cost regarding to the hardware of the developed system is relatively minimal compared to industrial machine vision solutions. The computer, camera and projector all-together costs less than a single industrial-grade camera with 5mp resolution, and costs 7 times less than hardware cost of the ferrite core inspection solution.

Comparing the application to currently in-production systems is difficult, as those systems work with different parameters and with highly specific goals. While they do have the same task of measuring the properties of the product, the human worker is just an assistant to the system, behaving as a pick-and-place robot. Systems, such as the ferrite core inspection stations are commonly found in places where robotic automation is not economically profitable, but human labour would be eliminated if made possible. The aim of those applications is to strictly measure the position / quality of the inspected item, the human aspect is not accounted for. This satisfies the connectivity need of the Industry 4.0 standard, but with the newly introduced 5.0 standard, human-centric work has been in focus. The thesis application has been designed around this fact and has proven that, - even with its current shortcomings compared to highly specialized solutions - it is in fact possible to create such a system, which assists human labour.

6. ABSTRACT

Industrial manufacturers are looking for ways to automate every step of their production line more than ever to increase their productivity. Machine vision is commonly used to help this automation, but some processes are still done by human labour. Measuring of the quality of human work is difficult as a number of variabilities have to be accounted for, and systems which currently exist have little emphasis on the human element. This thesis looks at the current state of machine vision technology, how currently in-production systems work and aims to design and implement a solution which's purpose is to assist the human worker while also measuring their work quality. The performance of the application will be measured, and a conclusion is drawn.

7. TARTALMI KIVONAT

A gyártóipari cégek minden eddiginél jobban automatizálják gyártósoruk minden lépését termelékenységük növelése érdekében. A gépi látást általában az automatizálás elősegítésére használják, de bizonyos folyamatokhoz továbbra is emberi munka szükséges. Az emberi munka minőségének mérése nehéz, mivel számos változóval kell számolni, és a jelenleg létező rendszerek kevés hangsúlyt fektetnek az emberi elemre. Ez a szakdolgozat a gépi látástechnika jelenlegi állását, a jelenlegi gyártási rendszerek működését vizsgálja, továbbá egy olyan megoldás kerül megtervezésre és megvalósításra, amelynek célja a humán dolgozó segítése, munkaminőségének mérése. Az alkalmazás teljesítménye mérésre kerül, és az eredményekből következetés kerül levonásra.

8. REFERENCES

[1] K. Segal, „Total Recall: A Data-Driven Analysis of the Takata Airbag Recall,” Dietrich College, Carnegie Mellon University, 2019.

[2] C. Demant, B. Streicher-Abel és C. Garnica, „Industrial Image Processing,” Germany, Springer, 1999, p. 299.

[3] M. Sonka, V. Hlavac és R. Boyle, Image Processing, Analysis and Machine Vision, 1993.

[4] S. Marcell, „Industrial applications of image processing based automation methods,” Budapest, 2019, pp. 31-35.

[5] Dicklyon, „Wikipedia - Rolling shutter,” 13 7 2021. [Online]. Available: https://commons.wikimedia.org/wiki/File:Propellor_with_rolling-shutter_artifact.jpg.

[6] F. Callejon, “FH/FZ5 Training Material,” 2014.

[7] S. Grehl, M. Donner, M. Sastuba és M. Ferber, Towards virtualization of underground mines using mobile robots, 2015.

[8] A. Zabatani, V. Surazhsky, E. Sperling és S. B. Moshe, „Intel RealSense SR300 Coded light depth Camera,” Intel RealSense Group, 2018.

[9] Keyence, „Vision System with Pattern Projection Lighting,” Keyence, 2019.

[10] V. TADIC, A. ODRY, I. KECSKES, E. BURKUS and Z. KIRALY, “Application of Intel RealSense Cameras for Depth Image Generation,” University of Dunaujvaros, 2019.

[11] M. Gramatke, „High-Resolution Time-of-Flight Cameras,” Basler, 2019.

[12] M. Gramatke, “2D or 3D Camera? Which 3D Camera Technology Fits Your Application?,” Basler, 2019.

[13] Intel, “Depth Testing Methodology,” Intel, 2018.

[14] Intel, „L515 Datasheet,” Intel, 2021.

[15] P. Deitel and H. Deitel, C++, How to program, Deitel, 2014.

[16] J. Lospinoso, C++ Crash Course, No Starch Press, 2019.

[17] C. A. Shaffer, Data Structures and Algorithm Analysis, Virginia Tech, 2011.

[18] A. Kaehler és G. Bradski, Learning OpenCV, O’Reilly Media, Inc, 2017.

[19] D. M. Escrivá és R. Laganiere, OpenCV 4 Computer Vision Application Programming Cookbook, 2019.

[20] C. László, Képfeldolgozás, Miskolc: Miskolci Egyetem, 2007.

[21] D. T. Vo, Spatio-temporal filtering for images and videos, UC San Diego, 2009.

[22] H. J. Park and K. B. Kim, “Depth image correction for intel realsense depth camera,” Institute of Advanced Engineering and Science, 2020.

[23] T. Sonoda and A. Grunnet-Jepsen, “Depth image compression by colorization,” 2019.

[24] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2018.

[25] G. Kertész, D. Z. Vámossy és D. h. S. Szénási, „Multi-Directional Image Projections with Fixed Resolution for Object Recognition and Matching,” Óbuda University, 2019.

[26] A. K. Bojja, F. Mueller, S. R. Malireddi, M. Oberweger and V. Lepetit, “Automatically Labeled Dataset for Hand Segmentation from Depth Images,” University of Victoria, 2019.

[27] D. Moreno és G. Taubin, „Simple, Accurate, and Robust Projector-Camera Calibration,” Brown University, 2012.

[28] A. Dadhich, Practical Computer Vision, Packtr.

[29] D. J. Katupitiya és K. Bentley, Interfacing with C++, 2006.

[30] A. Dadhich, Practical Computer Vision: Extract insightful information from images using TensorFlow, Keras, and OpenCV, 2018.

9. APPENDIX

The application's source code related to the thesis consists of the following:

- ~480 lines of C++ code
- ~175 lines of Python code
- ~2430 lines of C# code

Figure 7.1: Blind zones of an object.

Source: Own creation from the RealSense Viewer application

Measurements on the missing points will result in invalid values.

Figure 7.2: The projector and its lightning power.

Source: Own creation

It is still visible from a distance of 4 meters on a wall with direct sunlight. The projector's image is not considered by the camera, therefore 'just enough to be seen by the worker' brightness is enough.