

DEVELOPING GUIDELINES FOR EMBEDDING CODES ONTO PRODUCTS PRODUCED USING SELECTIVE LASER SINTERING

T. M. DE JONG - S1590170

MASTER THESIS

DOCUMENT NUMBER: DPM 1994

COLLOQUIUM DATE: 10-03-2023

UNIVERSITY OF TWENTE.

ABSTRACT

In the past decade, additive manufacturing has matured from a prototyping technology into a production technology. One category of additive manufacturing processes is powder bed fusion processes. Powder bed fusion processes have high precision compared to other additive manufacturing methods, offer a lot of design freedom and result in good material properties. Selective Laser Sintering (SLS) is a powder bed fusion process that can be used to produce polymer parts. To be able to automate and scale up SLS printing, it is required to be able to identify the printed products. Identification is problematic when similar but unique products such as personalized medical products like insoles are produced in the same build. To identify the parts, embedded 2D matrix barcodes using self-shadows can be used. In this research guidelines to embed 2D matrix barcodes onto SLS printed products have been developed.

At first, literature research has been conducted on SLS printing, 2D matrix barcodes and embedding codes onto 3D printed products. The advantages and limitations of SLS printing are identified. Guidelines that exist for SLS printing are used to estimate the achievable feature size. Several types of 2D matrix barcodes are researched. All work in roughly the same way, having positive (white) and negative (black) modules (pixels) that are distinguished based on contrast. When looking at the research that has been done on embedding 2D matrix barcodes onto 3D printed products goals and influencing parameters are identified. The goal is to print readable, printable unobstructive codes. To make the code readable, contrast needs to be generated. To do this, shadows are used. Parameters that influence the generated contrast are the lighting conditions, the size of the modules of the code, the carving depth, the surface where the code is embedded and the inclusion of patterns.

Four parameters that influence the successful embedment of the codes have been varied: 1) The module size of the code, 2) The carving depth, 3) The surface shape where the code is embedded and 4) Texture modifications of the surface. These parameters are varied in experiments. The parameters are varied individually from the values of a baseline sample. Samples with varying values for each parameter are produced.

In order to fix the lighting conditions, a setup is created and the same setup and procedure are used in the evaluation of all samples. Light is cast from four directions on the code. The height, distance and intensity of the lighting sources are varied. The main influencing setup parameter is the angle of the light that is cast on the sample. Next to that, a script is developed to process the images of the samples. To evaluate the samples, the generated contrast is quantified as the number of standard deviations between the group of positive and negative modules. This is used as the scoring indication. The sensitivity, specificity and errors per module are included for context. Next to that, it is checked if the code can be read.

A positive correlation between module size and the number of standard deviations between the positive and negative modules is found. The carving depth also has a positive correlation. The surface curvature has a negative correlation. The inclusion of patterns does not influence the generated contrast. A minimal value of 2.5 standard deviations between the means of black and

white modules is strived for. This means that approximately 1.24% of errors occur in both black and white modules. In reality, black modules are more likely to error. The errors mainly occur due to directional printing effects, negative module adjacency and isolated negative modules.

It is concluded that for the successful embedment of a 2D matrix barcode, the module size should at least 0.7 mm, and if there is a lack of space the module size can be reduced to 0.6 mm. The minimal carving depth is 1.6 mm. Ideally, the maximum surface curvature is 20°, but up to 35° is permissible. Surface texture modification did not influence the results. A supporting document with a three-step process to embed the codes is developed to help apply these guidelines and embed codes onto 3D SLS printed products.

More research can be done on the verification of this research and other parameters can be varied as well. Also, the workability of the design guidelines and the supporting document should be evaluated in practice.

CONTENTS

- 1 Introduction** **5**

- 2 Preliminary literature research** **8**
 - 2.1 Additive manufacturing 8
 - 2.1.1 Categories and technologies of additive manufacturing 8
 - 2.1.2 Selective Laser Sintering 14
 - 2.2 Storing information 20
 - 2.2.1 Assessment of data that should be stored 20
 - 2.2.2 Types of codes 21
 - 2.2.3 Conclusion 26
 - 2.3 3D-printing QR-codes 27
 - 2.3.1 Embedding codes 27
 - 2.4 Literature gap 32
 - 2.5 Research questions 33

- 3 Research methodology** **35**
 - 3.1 Procedure 35
 - 3.2 Setup 37
 - 3.3 Post processing code 39
 - 3.4 Experiments 41
 - 3.4.1 Sample set 1, size 41
 - 3.4.2 Sample set 2, carving depth 42
 - 3.4.3 Sample set 3, simple surface curvature 43
 - 3.4.4 Sample set 4, advanced curvature 43
 - 3.4.5 Sample set 5, pattern inclusion 45

- 4 Processing images** **46**
 - 4.1 Processing steps 48
 - 4.2 Determining scoring values 57

- 5 Setup** **62**
 - 5.1 The components 62
 - 5.2 The procedure 65
 - 5.3 Experiments 65
 - 5.4 Results 66
 - 5.5 Conclusion 70

- 6 Experiments** **71**
 - 6.1 Module size 71
 - 6.2 Carving depth 74
 - 6.3 Pattern inclusion 76
 - 6.4 Simple curvature 79

6.5	Advanced curvature	81
6.6	Conclusion	84
7	Design guidelines	85
7.1	Supporting document	85
7.2	Steps	86
8	Discussion	89
8.1	Extra investigation of samples	93
8.1.1	Investigation the 20 mm/base sample	93
8.1.2	Colour difference between samples in the same batch	97
9	Conclusion	100
	Bibliography	102
A	Plagiarism report	106
B	Terminology	107
C	Fused deposition modeling experiments	108
D	Extra plots setup	109
E	Supporting document	112

1 INTRODUCTION

Currently, the production industry is shifting towards industry 4.0. The first main characteristic is that industry 4.0 implements internet technologies in the production process. This can be digital data processing, connecting different machines using internet or utilizing digital twins to monitor and investigate production processes. A second characteristic is that these digitally integrated factories have a high degree of automation. A final characteristic is that the end user is able to customize their products more than ever before. Additive manufacturing (AM) is seen as an ideal manufacturing process for this shift towards industry 4.0, as all the required information is already in the digital environment. [1]. Additive manufacturing is digital from start to finish. This digital nature makes that it requires very little human intervention. This makes it suitable for automation, and combined with the digital information that is used in additive manufacturing the technology is very suitable for industry 4.0.

In the past decade additive manufacturing has matured from a prototyping to a production technology [2]. Additive manufacturing (AM) is fundamentally different from conventional production methods such as sawing or milling because instead of removing material during processing material is added [3]. An advantage of AM is that it can produce almost any possible shape as a result of the layer by layer manufacturing, removing constraints that are normally being associated with actually being able to produce the design [4]. New design freedom is achieved through AM based manufacturing. For different sectors end-use-parts are being produced using additive manufacturing. The production volume ranges from small series up to 100,000 products [5].

According to Goa et al. and Chen et al. additive manufacturing has several advantages compared to conventional 'material removing' production methods. At first, the overall ability to customize products has increased [6]. Next, additive manufacturing enables a lot of design flexibility and complex geometric shapes can be produced with no additional cost, since products don't need to be post processed [7]. Moreover, the need for assemblage can be eliminated, decreasing assembly costs [6]. Also, no start-up costs have to be made when starting production, increasing production flexibility. An application of this advantage could be to make on demand and on location AM available. Goa et al. state that AM technologies are suitable for products that have customized features, are geometrically complex or are produced in low volumes [7].

There is potential for AM to be more widely adopted as a manufacturing technology. Gao et al. looked into the state and challenges of additive manufacturing in 2015 [7] and Dizon et al. reviewed the state of the art of additive manufacturing in 2018 [8]. They described the different AM methods in detail. Gao et al. noted that there is a lack of design guidelines and standardisation of best practises for additive manufacturing [7]. As causes of this development they mention the variety of materials that can be used, the new application areas and the low-cost machines [7]. Dizon et al. [8] focused on the mechanical properties of the 3D printed products. Dealing with anisotropy is the biggest challenge according to Dizon et al. Thomas-Seale et al. conducted a case study to see what barriers impede the adaptation of AM in the industry in the UK. In their study, eighteen barriers were identified: education, cost, design, software, materials, traceability, machine constraints, in-process monitoring, mechanical properties, repeatability, scalability,

validation, standards, quality, inspection, tolerances, finishing and sterilisation [9].

There are seven main technology categories of additive manufacturing. Material extrusion, powder bed fusion, vat photopolymerization, material jetting, binder jetting, sheet lamination and finally direct energy deposition. In the field of additive manufacturing selective laser sintering (SLS) is viewed as the technology with the most production potential for plastic parts in industry [3]. The main reasons for this is that using SLS parts can be produced with high accuracy, the design advantages that SLS offers, good mechanical properties and a good surface finish [7]. Also, it does not require support structures to be present. SLS printing is a type of additive manufacturing that falls in the powder bed fusion category. Powder is added and targeted by a high-power directional heat source such as a laser to partly melt the material [7] [8]. In SLS layer after layer is sintered, building up a 3D product. There are other technologies in the powder bed fusion category. In the process selective laser melting (SLM) the material is melted instead of sintered. This means that it does not exceed the melting temperature, but still fuses. Also there are the processes where the material that is being used is a metal instead of a polymer. This process is named metal laser sintering (MLS). A difference, apart from the material being used, is that MLS requires support structures. These support structures function as heat sinks. Finally, a different energy source can be used, namely a electron beam instead of a laser.

In industrial SLS printing, the whole build volume is filled with as many as possible products that are to be printed. This optimally filling of the build volume is called 3D nesting, and is done to fill up the maximum space. The space should be maximally utilized as the powder decreases in quality as it undergoes more build cycles. The powder decreases in quality because the whole build volume is heated elevated temperatures, causing thermal effects on the powder. Next to optimally filling up the build volume, a recycling strategy is required and/or the powder needs to be replaced after a certain number of cycles. So, filling the build volume is this more economical because more powder is utilized. A build volume typically contains several hundreds of potentially unique parts. This is more efficient, as that way more products per printing volume are produced.

After production, these parts need to be sorted out after printing and cooling down. The sorting step is problematic when a lot of similar but unique product are produced in the same build volume, and the product are hard to distinguish. An example of these products are personalized medical products, such as products for dentalwork, artificial hips or insoles. Because of this large number of products in the same build volume it is necessary to have an identifier on each product. The sorting process has turned out to be hard to automate, as it is necessary to identify what parts are being processed. Advanced 3D scanners that compare product geometry to CAD-models can be used for product identification. However, these scanners are big and expensive. Moreover, the product identification process is slow (several minutes), which would be problematic if hundreds of parts need to be identified. The possibility of embedding an alphanumeric string onto the products is explored. However, this is hard to automate and results in a major design intervention. To make this identification step easier and help the industry further automate the process of 3D printing, an identifier code can be embedded on all produced parts. This should reduce the cost of equipment that is needed and increase the speed at which parts can be identified. Also, it is a smaller design intervention and can store more data than the current alphanumeric strings. So, it would have considerable advantages if it is possible to embed codes onto 3D printed products.

Prior research by Kikuchi et al. showed that it is feasible to embed codes onto 3D printed products [10]. Peng et al. and Papp et al. continued to explore the possibilities on how this embedding should work in practice [11] [12]. They succeed in embedding codes onto freeform

surfaces. However, and the practical applications of this technique have not been looked at. The goal of this research is to link the fundamental potential of the ability to embed QR codes on freeform surfaces to the practical needs of the industry and the customer, making the technology generally applicable. These embedded codes can be used as a means of storing additional information such as a serial number, unique identifier or even statistics. Conventional methods of including this information consist of attaching labels, engraving or marking the part. Embedding a code directly surpasses these methods in terms of flexibility, cost effectiveness, information density and reliability.

For SLS printing several design guidelines have been developed. Seepersad et al. [13] developed guidelines for dimensioning SLS parts. These guidelines include guidelines for the feature resolution, the resolution of small parts and the space needed for clearances. Developing guidelines is a step to make the technology more generally applicable.

The goal of this research is to establish guidelines on how to embed readable codes onto 3D printed products. These codes are to help with product identification, allowing for the picking and sorting process to be automated. This could significantly reduce production cost and required employee time by automating the identification and sorting step. This enables similar but unique products to be produced in a single build volume and allows an increase in the series size for which production by additive manufacturing is a viable option. Moreover, it eliminates the use of labels as an information carrying device since this information can now be contained within the product itself, making it easier to identify the product. By making this identification step easier a factory that uses additive manufacturing could be fully automated. By developing the ability to embed codes onto 3D printed products several of the barriers identified by Thomas-Seale can be overcome. The biggest improvements will be made in the categories cost, inspection traceability and scalability.

The main research question is formulated as: "What design guidelines should be established to allow readable QR codes to be embedded onto industrial products that are produced using SLS 3D printing?"

There are several factors to consider when designing these guidelines. At first, the required contrast in order to read the code and how to guarantee that this contrast is investigated. How to embed the code is highly dependent on the geometry and surface of the product. The suitability of the surface where the code is to be embedded is also to be evaluated. This allows the code to be embedded on an ideal position. This is a second factor that is taken into consideration. Thirdly, there is the size of the code that is to be embedded. Ideally this size would be as small as possible, while not compromising the ability of the robot to find the code nor affecting the readability of the code. Next to that, the type of code that has to be embedded will be selected.

2 PRELIMINARY LITERATURE RESEARCH

An extensive literature research has been conducted. It consists of three sections. The first section reviews the current state of the art of additive manufacturing. In this section, the different AM production technologies are explained. The technology that this project will focus on, selective laser sintering, is discussed in more detail. In the second section, different codes that are used for storing information are reviewed. This section consists of three parts. At first, the information that is currently used for product identification and current ways of storing this information is looked into. Next, the different types of codes are explained. Finally, the codes that can be used are listed. In the third section, the research that has been conducted on 3D printing codes is investigated. Important parameters when embedding codes are listed and the conducted research is categorized and tabulated. After these sections, the gap in the literature is identified, and sub-questions are formulated to be able to answer the research question.

2.1 Additive manufacturing

In this section, the different methods of additive manufacturing, commonly referred to as 3D printing, are discussed. The advantages and limitations of these methods are explained. The limitations are important because next to the camera accuracy, the production accuracy is expected to be a limiting factor when embedding QR codes on 3D-printed products.

The rest of this paragraph is structured as follows. First, an overview of the different types of additive manufacturing is given. Next, the production method that is most relevant for the research, selective laser sintering, is discussed in depth.

2.1.1 Categories and technologies of additive manufacturing

There are several additive manufacturing technologies [7] [8]. These manufacturing methods can be separated into 7 categories. In this section, the different additive manufacturing technologies are discussed according to these categories. The categories are material extrusion, powder bed fusion, vat photopolymerization, material jetting, binder jetting, sheet lamination and directed energy deposition. An overview of the categories, technologies that are grouped under each category, the material that is printed, the power source that is used and the advantages and disadvantages of the different categories are shown in Table 2.1.

Table 2.1: Different forms of 3D printing adopted and revised from Goa et al. [7].

Category	Technologies	Printed material	Power source	Advantages/Disadvantages
Material extrusion	Fused deposition modeling (FDM)	Thermoplastics Ceramic slurries Metal pastes	Thermal energy	+ Inexpensive extrusion machine + Multi-material printing - Limited part resolution - Poor surface finish
	Contour crafting			
Powder Bed Fusion	Selective Laser Sintering (SLS)	Polymer/Polyamides	High-powered laser beam	+ High accuracy and details + Fully dense parts + High specific strength & stiffness
	Direct Metal Laser Sintering (DMLS)	Atomized metal powder	High-powered laser beam	- Powder handling & recycling
	Selective Laser Melting (SLM)	Atomized metal powder	High-powered laser beam	
	Electron Beam Melting (EBM)	Atomized metal powder	Electron beam	- Support and anchor structure (not for SLS)
Vat photopolymerization	Stereolithography (SLA)	Photopolymer Ceramics	Ultraviolet laser	+ High building speed + Good part resolution - Overcuring, scanned line shape - High cost for supplies and materials
Material Jetting	Polyjet / Inkjet printing	Polymer powder Ceramic powder Metal powder	Thermal energy Photocuring	+ Multi-material printing + High surface finish - Low-strength material
Binder jetting	Indirect inkjet printing (Binder 3D P)	Polymer powder Ceramic powder Metal powder	Thermal energy	+ Full-colour objects printing + Wide material selection - Requires infiltration during postprocessing - High porosities on finished parts
Sheet lamination	Laminated object manufacturing (LOM)	Plastic film Metallic sheet Ceramic tape	Laser beam	+ High surface finish + Low cost (material, machine, process) - Debubing issues
Directed energy deposition	Laser engineered net shaping (LENS)	Molten metal powder	Laser beam	+ Repair of damaged / worn parts + Functionally graded material printing
	Electronic beam welding (EBW)			- Requires post-processing

The first category that will be discussed is material extrusion. In processes in this category, the printed material is melted and extruded onto a working space in 2D layers. These layers are stacked, creating a 3D model. The most used technology that falls under this category is fused deposition modelling (FDM). Fused deposition modelling (FDM) is the method that is most commonly used in consumer-level 3D printers [8]. A schematic overview of this technology is shown in Figure 2.1. It is a method in which filament is extruded in 2D layers that are stacked onto each other to create the 3D product. As shown in the Figure, it is also possible to use two types of filament. The filament is heated in the extrusion head and extruded through the nozzle onto the build platform. The diameter of the nozzle through which these materials are extruded varies and influences the production speed and accuracy. The filament is made from thermoplastic materials such as PC, ABS, PLA and PEEK. Although the majority of FDM processes use thermoplastic materials, ceramic materials and metal pastes can also be processed using this method. Dizon et al. state that the use of thermoplastic polymers and the melting of these polymers are the major limitations of FDM [8]. Another limitation of FDM is that it does not allow for overhang. If there is overhang a support structure is required. According to Goa et al. the advantages of FDM are the low cost, wide availability of machines and high printing speed. They add that the limited part resolution, poor surface finish and high porosity are disadvantages of FDM [7].

Another technology in the material extrusion category is contour crafting. Contour crafting utilizes the same concepts but on a larger scale. An example of what contour crafting is used for is the printing of houses. Instead of using thermoplastic materials that are typically used in FDM, contour crafting tends to use quick-setting materials such as concrete or sand.

Secondly, powder bed fusion technologies are addressed. In these methods, a bed of powder is heated to a certain temperature. An energy beam, typically a laser or an electron beam, adds some more energy to melt or sinter the material. This material then consolidates. Next, a new layer of powder is spread on top using a distribution mechanism, adding a new layer that can

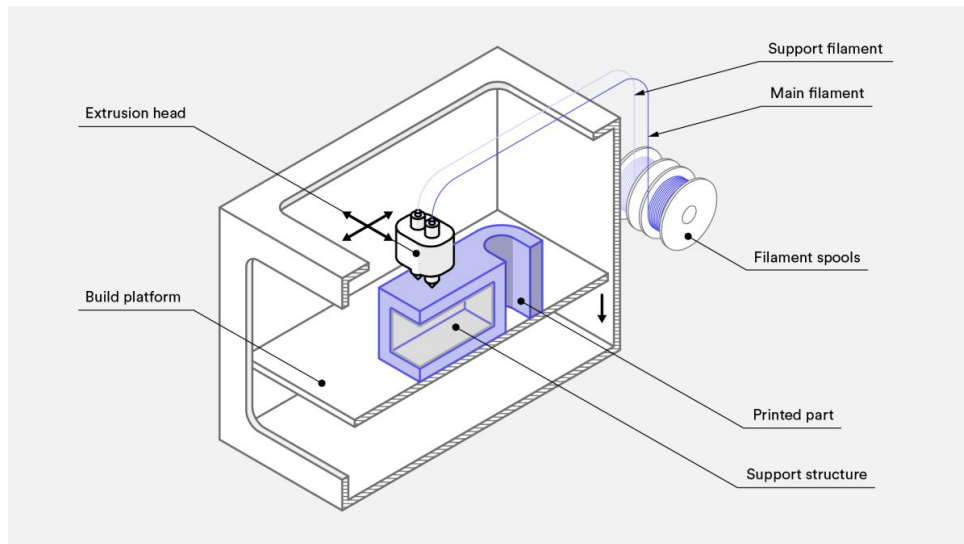


Figure 2.1: A schematic representation of FDM printing from Hubs [14].

be solidified. This method allows for thermoplastic plastics, ceramics and metals to be printed. A technology within this category is selective layer sintering (SLS). An illustration of SLS printing is shown in Figure 2.2. In SLS a polymer is used as the building material. The powder is sintered, partly melted, to form a solid structure. In SLS, the energy source is a laser beam (C), which is projected using laser optics (D) onto the powder bed (B), sintering the material. In the image, this is represented by a different colour of the material. The build platform (A) moves down after one layer is sintered, and new powder is dispensed (E) and spread over the build volume (F). Other technologies differ on what type of material is being used, whether the powder is sintered or completely melted and if a laser or an electron beam is being used. Three other technologies that fall in the category of powder bed fusion technologies are direct metal laser sintering (DMLS), selective laser melting (SLM) and electron beam melting (EBM). Instead of using a polymer, atomized metal powder is used to build the parts in these technologies. In EBM an electron beam is used instead of a laser as the energy beam.

An advantage of powder bed fusion technologies is that it allows for high accuracy and part quality. It results in parts with low porosity, which in turn have a higher specific strength and stiffness. Another advantage of SLS is that it allows for overhang, offering more design freedom. The other technologies, DMLS, SLM and EBM, do require a support structure. This support structure is needed to cool the metal parts that are formed. Solid structures have a higher conductivity than powder, so solid structures are used to cool the product. A disadvantage of processes in this category is that they tend to be expensive, as high-power heating sources are required. Also, powder handling can be challenging and not all powder can be recycled. The precision of the parts is limited by the powder particle size. Further quality is determined by the laser power, scan spacing and scan speed [8].

The third category is the category of vat photopolymerization. In this category, a liquid resin is used that becomes solid under UV light [8]. A technology that falls in this category is stereolithography (SLA). A schematic of SLA printing is shown in Figure 2.3. The liquid form of UV curable resin is added to a basin, the resin tank, and selectively exposed to UV light. In the Figure, a laser is used to 'cure' the resin, exposing it to UV light. As the resin is exposed to UV light, the chemical structure changes and the material becomes solid. The solidified part (cured resin) is then lifted up by the build platform so that new resin can be added to the basin. The solidified layers attach to one another, and the first layer is attached to the build platform.

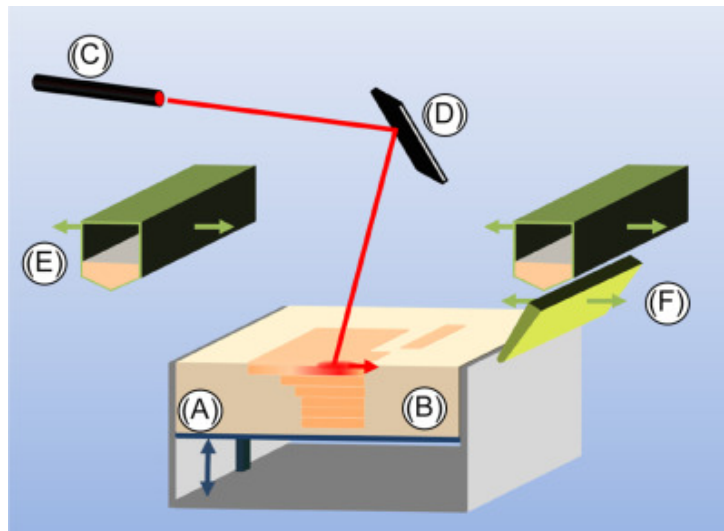


Figure 2.2: A Selective Laser Sintering setup consisting of (A) building platform (vertically movable), (B) powder bed (C) laser source, (D) laser optics, (E) deposition hopper for feeding, and (F) distribution and leveling blade. Adapted from S.C. Ligon, R. Liska, J. Stampfl, M. Gurr, R. Mulhaupt [15].

This process is repeated, thereby forming products. An advantage of SLA is that it produces products with high precision. The building speed of SLA is relatively high at around 1-3 cm per hour. The minimum layer thickness is dependent on the curing depth [7]. Disadvantages of this process are that errors might occur due to overcuring, the layers from which the product is built up are clearly visible and the materials and supplies have high costs. Digital light processing (DPL) is a variation that uses a different lighting technique. It is able to print quicker, but with a lower resolution and a rougher surface finish.

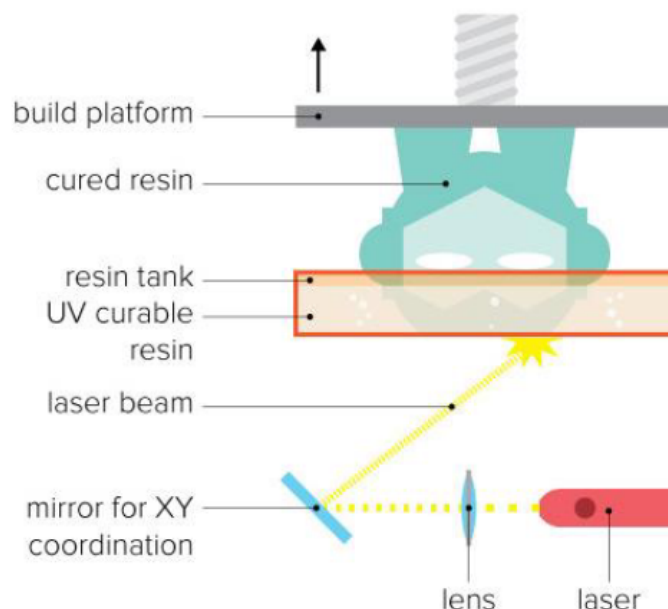


Figure 2.3: Schematic of SLA printing from medium.com [16].

In technologies in the category of material jetting, wax and/or photopolymer droplets are dispensed onto a substrate using drop-on-demand ink-jetting. The phase change of the dispensed

droplets occurs due to heating or photocuring. The viscosity that is required for this to work limits the types of material that are suitable for this process [7]. An example of material jetting is polyjet printing. A schematic of this printing method is shown in Figure 2.4. Polyjet printing utilizes multiple print nozzles to accurately spray the droplets of material on the printing surface, and UV-light cures the droplets instantly. In the image, the print nozzles are clustered in the print head. The print head can move in a plane, and the build platform lowers as each layer is finished. Like stereolithography, this method uses UV light to change the chemical composition of the printed material. It is able to print multiple materials in a high resolution [8].

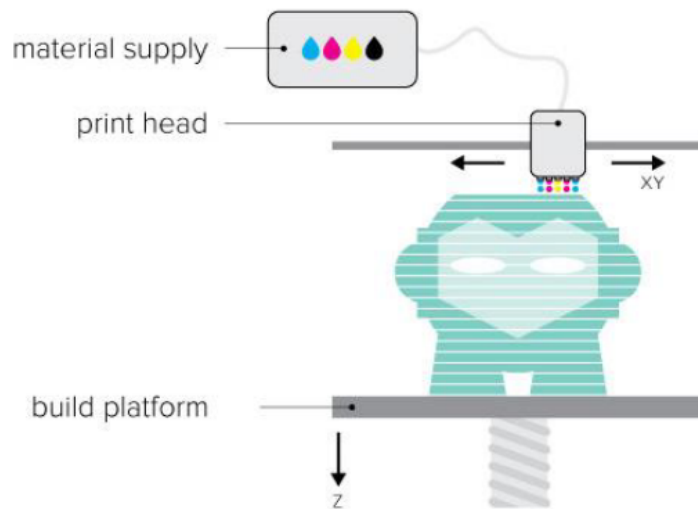


Figure 2.4: Schematic of material jetting from medium.com [16].

The next powder category that is discussed is binder jetting. Binder jetting is similar to powder bed fusion technologies in that there is a powder bed. Instead of using a high-energy source to sinter/melt the powder, a liquid binding material is selectively deposited over the powder. This material binds the powder. An example of a printing process in this category is three-dimensional printing (3DP). A visual representation of binder jetting is shown in Figure 2.5. The powder is added to the powder bed by a roller. The printhead deposits the fluid binding material in the powder bed, selectively solidifying the powder and forming parts. The build platform is then lowered by one layer, new powder is added and the printhead deposits liquid binding material. A technology in the category of binder jetting is indirect inkjet printing. When indirect inkjet printing is used, post-processing is required to achieve sufficient strength [7]. This method is limited by the viscosity of the binder material and suffers from poor surface quality [8].

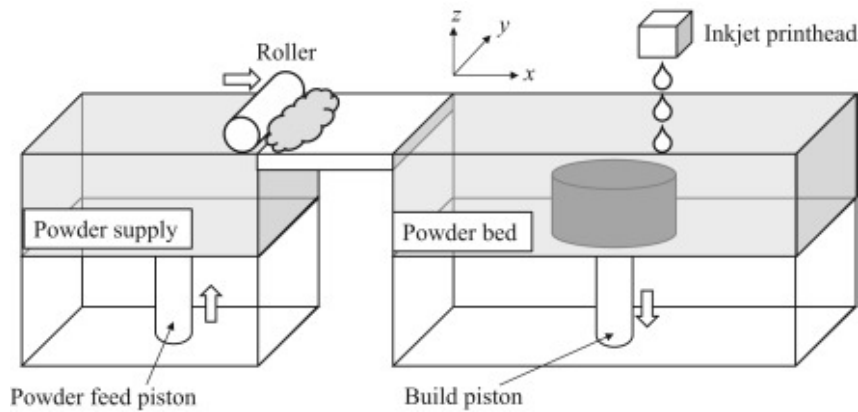


Figure 2.5: Schematic overview of binder jetting, adapted from Zhang et al. [17].

Sheet lamination works by building up the model from heated layers that are cut out using a laser. In Figure 2.6 a schematic overview of sheet lamination is shown. Each layer is cut out in a specific shape, and the layers are stacked to create the final product. According to Gao et al. sheet lamination results in low internal stresses, a high surface finish and lower costs [7]. Dizon et al. state that it allows for freedom of form, but results in products with a poor or average surface finish and resolution [8]. Note that these claims contradict each other.

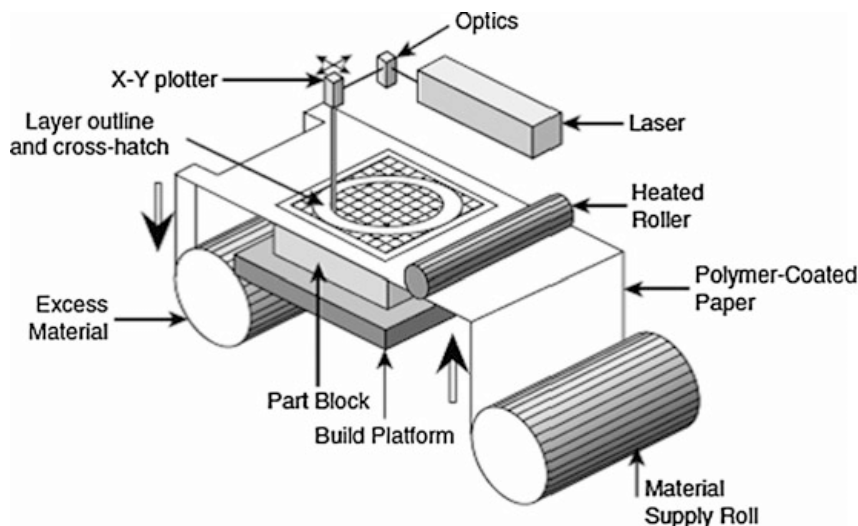


Figure 2.6: Schematic overview of the sheet lamination process from Additive manufacturing technologies by Gibson, Rosen and Stucker [18].

Finally, there is the option of using directed energy deposition. In this process metallic material is fed into an energy source, melting the material. This can be thought of as a 3-dimensional welding machine [7]. The material is cooled after being dispensed. This results in high material strength. This process is especially suitable for the repair of parts or to add coatings to products. In Figure 2.7 a schematic overview of direct energy deposition is shown. The powder feed nozzles supply the material to the point where the laser beam is aimed. The laser beam melts the powder, causing it to solidify. The build platform is able to move in three directions, while the material and energy supply sources are aimed at a single point.

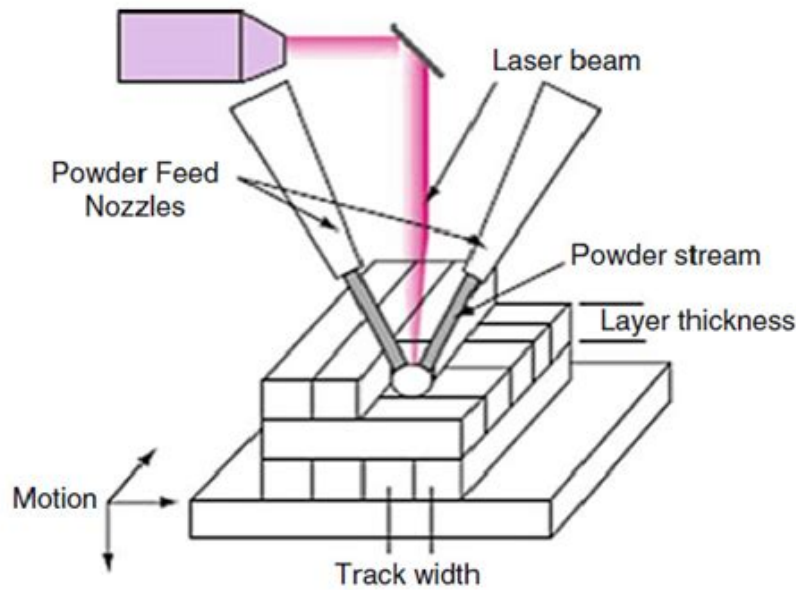


Figure 2.7: Schematic overview of direct energy deposition from Additive manufacturing technologies by Gibson, Rosen and Stucker [18].

2.1.2 Selective Laser Sintering

In this section, the current state of the art of SLS printing is evaluated in-depth. First, the workings of SLS printing are explained in detail. This is done to get a deeper understanding of how the process works. Moreover, it will create a complete picture of what is possible and what mechanisms are taking place in the process and should thus be taken into account. The goal is to answer the questions: "What is currently possible using SLS printing and what are the limitations?" and "What effects should be taken into account when using selective laser sintering for the production of polymer parts".

Next, the processing parameters that play a significant role are identified. Three important influencing factors are looked at: 1) the machine, 2) the user-defined printing settings that play a role when printing, 3) the powder properties. First, the properties of the machine that play a significant role are identified. By identifying and fixing these parameters constant printing quality can be ensured. Thereafter, the relevant settings that play a role for print quality are listed. Finally, the properties of the powder that affect the processing are described. This should serve to answer the question: "What processing parameters are relevant when manufacturing using Selective Laser Sintering?".

After identifying the processing parameters, it is investigated what the expected smallest achievable feature size is for SLS printing. This will be done in general and for the applied case when using the EOS PORMIGA P 110, the SLS machine that will be used in this research. Existing design rules for SLS printing will be looked at. This will give an idea of the resolution that can be achieved using SLS machines. The achievable resolution will be used to get a starting point on the size of the printed samples. It serves as the starting point for investigating what is the minimum module size. Also, this ensures that the design guidelines will be developed in correspondence with these settings. An estimated answer to the question: "What is the minimal size of the modules of the code?" should be given. The exact answer will be determined through experiments.

Selective Laser Sintering Mechanisms

In this section the binding mechanisms that are at work when selective laser sintering will be discussed. For a start, it is useful to recap what Selective Laser Sintering is. Through the use of a laser, specific parts of a powder bed are sintered, turning those parts solid. Afterwards, a new layer is added. By repeating this layer after layer the layers merge and a solid product is manufactured. A schematic design of an SLS machine with the parts of the machine is shown in Figure 2.8.

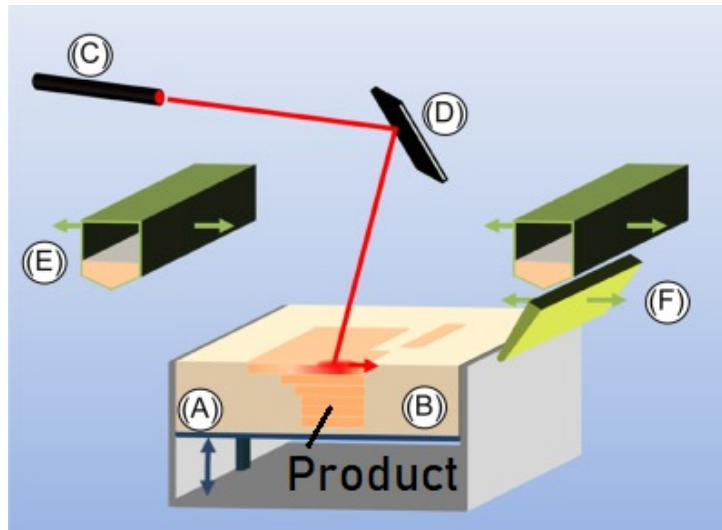


Figure 2.8: A Selective Laser Sintering setup consisting of (A) building platform (vertically movable), (B) powder bed (C) laser source, (D) laser optics, (E) deposition hopper for feeding, and (F) distribution and leveling blade. Adapted from S.C. Ligon, R. Liska, J. Stampfl, M. Gurr, R. Mulhaupt [15].) [19]

Compared to other AM methods SLS has a number of advantages. At first, SLS is able to process a wide range of materials. As long as a powder material absorbs laser energy and is not too viscous it can in theory be used. According to Shi et al. these powders can be polymer or ceramic powder materials [20]. When processing polymers, the polymer should be a thermoplastic material, as thermosets are not suitable. Secondly, the utilization rate of material is high as the unused powder can be recycled. This recycling can only happen to a certain extent as will be discussed in powder properties. Third, SLS printing offers a lot of design freedom as there are few restrictions on what can be printed. Since the powder bed offers support overhang in the model is not a problem. Also, because of the supporting powder bed, no support structures are needed. Next to that, SLS is able to produce parts with high accuracy and details. One disadvantage is that the surface roughness of parts is higher than when a technique such as SLA is used.

In order to better understand the sintering process, a closer look is taken at what happens during the 'selective laser sintering'. The general principle is that the powder bed is preheated to a certain temperature, and the laser will add extra energy at specific places. This will induce sintering at those selected positions. Sintering is different from melting in that the material does not melt completely. Instead, the material is heated to the sintering temperature. It is possible that two phases exist simultaneously [3]. The small particles of the material are then fused into one piece, solidifying the powder. According to Gibson, Rosen and Strucker different powder fusion mechanisms are at work in the process that is called selective laser sintering [18]. Four fusion mechanisms can be at work in powder bed fusion processes: solid-state sintering, chemically

induced binding, liquid-phase sintering or partial melting and full melting. These four binding mechanisms are shown in Figure 2.9.

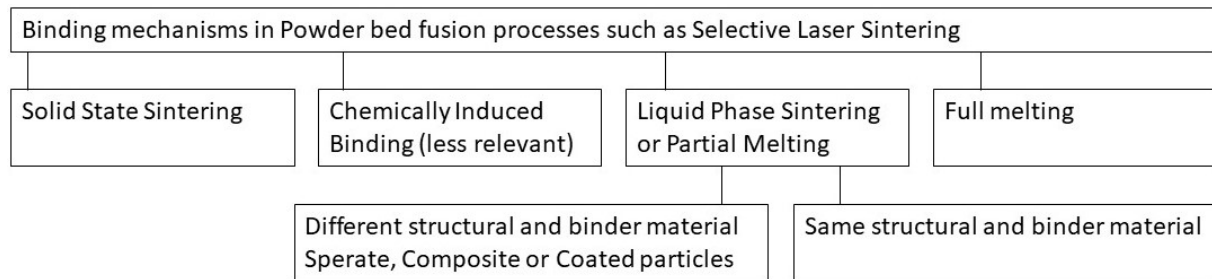


Figure 2.9: Possible binding mechanisms at work in powder bed fusion processes, specified for Selective Laser Sintering, adapted from Gibson, Rosen and Strucker [18].

At first, there is solid-state sintering. Solid-state sintering is a process by which particles are fused without melting at elevated temperatures. Sintering takes place between half the melting temperature (K) and the melting temperature. The process that drives sintering is the reduction of surface energy, which happens by reducing the surface area to a minimum. Sintering increases in speed as temperatures increase. Solid-state sintering is the slowest when selectively fusing a region. Often other fusing mechanisms are used alongside solid-state sintering. This is because the faster a fusing process is, the more economically competitive it is as it takes less time to form a layer.

Even though solid-state sintering may not be the primary fusion mechanism, the products are affected in several ways. Since the powder is held at an elevated temperature, solid-state sintering constantly takes place. This causes the particles in the powder bed to sinter to one another, causing the average powder size to increase. This decrease in powder quality is called (powder) ageing. Because of this ageing powder can't be recycled indefinitely, as the powder properties change. To counter this a recycling strategy is implemented and new (virgin) powder is added to the recycled powder before each new print [18]. How the powder changes as it ages will be discussed in powder properties. Next to ageing, another mechanism that takes place due to solid-state sintering is part growth. What is meant by this is that the powder around the product fuses with the product. This happens because the temperature of the part is higher than the surrounding powder, transferring the heat due to conduction and increasing the speed at which solid-state sintering takes place in the surrounding area. This increases the rate of solid-state sintering. Because of this, particles will attach to the product. If the products are lower on the powder bed, the part is kept at an elevated temperature for a longer time (before the print is finished). This causes solid-state sintering to continue for a longer time. Increasing the amount of part growth that takes place. So, it can be expected that some part growth occurs, especially if the products are lower in the printing volume [18]. One positive effect that solid-state sintering has on prints is that it causes the porosity of parts to go down, thereby increasing the mechanical properties. Just like with part growth, this will happen to a higher degree on the bottom of the printing volume than on the top. This is also caused by the solid-state sintering that is occurring as the part is kept at elevated temperatures.

Chemically induced binding is the second mechanism that can be at work. In chemically induced binding thermally activated chemical reactions form a product which binds the material. This mechanism is mainly at work when ceramic materials are being processed [18]. Since the focus is on polymers this fusion mechanism is expected to be less prevalent.

In liquid phase sintering, a part of the powder particles melt while another part remains solid [18]. This melting results in the fusion of powder particles. The molten parts act as a binding material to keep the solid particles together. There are two material compositions in which liquid phase sintering takes place. There can be distinct binder and structural materials, or indistinguishable binder and structural materials.

If there is a clear distinction between binder and structural materials, two materials are present. Typically, these materials have different melting temperatures. One component, the binder material, melts, while the structural component remains solid. The molten component then fuses with surrounding components, creating a solid structure. This structure tends to have low porosity, as the melting of the particles efficiently fills up gaps. There are three ways to combine the different two different material. They can be combined as as mixed separate particles, as coated particles or as as composite particles [18]. In the first way, the two types of particles are mixed. Well-mixed separate particles are in many cases sufficient for liquid-phase sintering. Often the formed structures have a high porosity as the melted material has insufficient time to flow and draw particles together. If the density of the powders is different segregation might occur. Secondly, there is the option of using composite particles, this means that the particles contain both the structural and the binder material. The third option is the use of coated particles. As the name suggests, one particles contain a core of one material and an outer layer of the other material. In this particle build-up, the structural material is in the core.

If only a single material is present, liquid-phase sintering can still take place. In this case a part of the material melts while another part remains solid. Because of low conductivity, the smaller particles and the outer shell of larger particles can be melted without completely melting the whole material. This causes some of the material to be in a melted state while another part remains solid. It is debatable whether this process should be called liquid phase sintering or partial melting. Another polymer property that has a large role is the glass transition temperature. Above this temperature, but below the melting point, fusion between polymer particles occurs because of the reduced viscosity/liquid-like behaviour. Exceeding the glass transition temperature can enable fusing materials without fully melting them.

Full melting is most commonly used when processing metal alloys and semi-crystalline polymers such nylon polyamide [18]. The region of the material that is subjected to heat, in the case of SLS the lasered area, is melted to a depth larger than the layer thickness, causing the new layer to bond with the solid parts in the previous layer. By fully melting the material it results in a solid product with the highest possible strength. However, since excess heat is added to the product it also results in part growth. Here a trade-off is made between part strength and accuracy.

Gibson, Rosen and Stucker note that any of the four mechanisms that are discussed can be used, and often more than one is present. For the selective laser sintering of polyamide 12 it is most likely that a combination of partial and full melting serves as the primary fusion mechanisms. Solid-state sintering has a secondary role.

Machine parameters

According to Sun, Brandt and Easton several processing parameters are important when using powder bed fusion [21]. They separate these parameters in four categories: Laser, Scan, Temperature and Powder related properties. The first three, Laser, Scan and Temperature are machine related and will be discussed here. User set parameters will be discussed in the next section, and the powder-related properties will be discussed in the section after that.

The laser properties that are relevant are the laser power, wavelength, spot size, pulse dura-

tion and pulse frequency. These are responsible for how much energy is deposited onto the powder bed, thereby sintering the material. The laser parameters determine how much energy is transferred and how large the area is where this energy is deposited.

Scan-related properties are the scanning speed, spacing and scanning pattern. Together with the laser-related properties, this determines the precision with which the energy can be transferred to the powder bed.

Among the temperature-related properties Sun, Brandt and Easton identify three parameters. At first, there are the temperature of the powder bed and the powder feeder temperature. The powder feeder temperature is the temperature at which the powder enters the powder bed, and the powder bed temperature is the temperature of the powder bed. This determines how much the temperature needs to rise to sinter or melt the material and thereby how much energy needs to be provided by the laser. Also, the temperature of the powder bed determines the cooling when using a related technology such as selective laser melting. The third parameter that is identified is the uniformity of the temperature, so if the powder has the same temperature everywhere on the powder bed. These first three categories that Sun, Brandt and Easton identify, the laser, scan and temperature-related properties are dependent on the machine that is being used. If the machine configuration is correct these three categories should not vary from print to print.

User-set parameters

Several user-set parameters influence the quality of printed products. The most relevant property that can be varied is the layer thickness. Layer thickness can be varied from print to print. If the layer thickness is smaller, the resolution is higher but more layers have to be printed, which takes more time. If the layer thickness is higher, the product is printed faster but with a lower resolution. So by changing the layer thickness the balance between speed and product quality is adjusted. Another factor that can be of influence is the part orientation in the machine. In general, upward-facing features are more smooth [18]. This is because they solidify against air while downwards-facing parts solidify against powder and that powder can attach to the product, resulting in higher surface roughness. Applied research from Weiss et al. on feature resolution shows significant differences in resolution with build orientations [22].

Powder properties

The powder-related properties are the properties of the powder that is used that are relevant for the process. At first, the material properties play a significant role. On a grain level, the particle size, shape and distribution of the grains are relevant. Also, the powder bed density is important. In the SLS printer used in this experiment, polyamide 12 with the brand name PA2200 was used, so a closer look is taken at the behaviour of this material.

When reusing powder thermal ageing takes place, degrading the quality of the powder. The particle size, shape and chemical composition are affected by the heat cycles that the material undergoes during production. Several mechanisms are at work during ageing. Gibson, Rosen and Strucker note that the particle size increases as the powder undergoes more heat cycles. This is the effect of solid state sintering causing particles to attach to one another [18]. Chen et al. and Dadbaskhsh et al. looked at the microstructure of polyamide 12 grains and noted that grains crack [23] [24]. Also, when particles are exposed to a heat cycle small particles form around the normal-size particles [23]. Yang et al. state that the chain length of the polymer increases as a result of dehydration [25]. Thus, the powder quality degrades as it is recycled. As a result, when using older powder the quality of the manufactured products decreases.

To counter this, virgin powder is added to the powder mix for each new build cycle. The amount of virgin powder that is being used is important for the powder properties, as it influences the powder quality and thereby the print quality. Every time the machine is turned on 30-50 % virgin powder is recommended to be added in order to reach the resolution [26]. The amount of virgin powder that is used during the prints is fixed at 30% virgin powder.

When a higher powder fraction is recycled powder the material costs go down. In SLS printing the material cost is significant, so utilizing more powder is an advantage. Vock et al. state that powder is the second or third largest cost when using powder bed fusion technologies [27]. Combined with the low recyclability, this high material cost makes making optimal use of the build space essential.

Minimum feature size

In order to be able to determine what the size of the print should be it is relevant to know what the minimum feature size is that can be achieved. On this topic, a lot of research has been conducted already. Ligon et al. state that the minimum feature resolution for SLS printing that can be achieved is 100 μ m under ideal circumstances [15]. Seepersad et al. experimented with the feature size that can be achieved with SLS [13]. They give guidelines on how parts can be dimensioned while still being printable. They were able to print holes as small as 0.5 mm and print a clear font of 20 pt. This gives an idea of the resolution that can be reached by an SLS machine. This is relevant when printing codes, as it is impossible to print smaller than the minimally achievable feature size. When printing codes this will be used as a starting point to determine what the minimum module size should be. The resulting design guidelines from Seepersad et al. are tabulated in 2.2.

Table 2.2: Design guidelines by Seepersad et al. [13] with a wall thickness of 1.8 mm for holes. Vertical means the feature is printed in the height direction.

	Hole vertical	Hole horizontal	Wall vertical	Wall horizontal	Font
Acceptable	0.5 mm	1.3 mm	0.2 mm	0.6 mm	20
Good	0.8 mm	1.3 mm	0.6 mm	0.8 mm	24

The orientation that the feature has in the machines does influence the results. According to the research of Sheepersad et al. the minimal feature size is smaller when the feature is printed in the height direction [13]. This makes sense, as extra material is more likely to attach to the product in the height direction. If the feature is printed in the height direction (vertical orientation) this effect is minimized, while if the feature is printed in the layer direction the extra material attaches in the height direction.

Weiss et al. looked at the minimum feature sizes of negative features [22]. They looked at four negative features and determined a 95% interval at which the feature is successfully produced. The results are tabulated in Table 2.3, and the names are visualized in Figure 2.10. From the Figure, it seems like Weiss et al. define the horizontal direction differently than Sheepersad. The samples of Weiss et al. work better when printed in what Sheepersad calls the vertical direction. This is not what one would expect, as material is more likely to attach in the vertical direction and this should happen more in samples that have a feature in the horizontal direction.

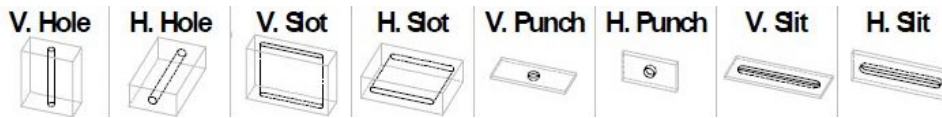


Figure 2.10: Terminology as used by Weiss et al. [22].

Table 2.3: Minimal feature size at 95% interval according to Weiss et al. [22].

	Hole V	Hole H	Slot V	Slot H	Punch V	Punch H	Slit V	Slit H
95% success interval	6.5 mm	6.5 mm	1.8 mm	0.7 mm	1.1 mm	0.4 mm	0.4 mm	0.3 mm

The SLS machine that is used in research is a EOS PORMIGA P 110. According to the specification sheet of this machine, it is able to produce walls that are 0.4 mm thick. Weiss et al. use the same machine and criticize that the specification given by the manufacturer can be ten times more optimistic than the actual size of features that can be manufactured [22]. They conclude that negative features of 0.4 mm can be produced if the carved-out feature is small.

2.2 Storing information

In this section, suitable information storing codes are identified. The information that needs to be stored and the possibilities to store the information are investigated. At first, it is assessed what and how much data needs to be stored. Next, the different types of codes are discussed. Finally, a conclusion is drawn about the type of code that is suitable to store all this information. This type of code is selected to be embedded onto the products. The subquestions that are answered are: "What information storing codes are available?" and "What information needs to be stored?". By combining the answers to these questions, a suitable code can be selected.

2.2.1 Assessment of data that should be stored

In this section, the data that should be stored is assessed so that a suitable code can be selected. When storing data there are several actors that have an interest in what data is stored. The most prominent of these actors are the manufacturers. Another actor that could benefit from specific data being stored are the users of the product. Depending on the user and use case the amount and type of data that needs to be stored differs. The focus of this research is on industry, which means that the main focus is on the manufacturer. This scenario is discussed first. Afterwards, variations and extra possibilities of user interactions along with their implications for the code are discussed.

For manufacturers storing product information is essential. The product information is used to identify the produced parts. So, the product information has to include at least what part is it. For this a serial number could be used, that being a (alpha)numeric string of approximately 12 numbers. This serial number does not require a lot of data storage capacity. Extra information can be added such as the production date, time, material, printer that was used, further required manufacturing steps and the destination of the part. Even work instructions can be added and in the spirit of industry 4.0 product management can be automated. This information can be stored directly (spelled out), through a code (a certain set of characters has a meaning) or through a website portal. Adding the information directly requires a lot of data storing capacity. By encoding the information the data can be stored much more efficiently. However,

this does require a device that is able to process this information. The website portal limits the required string length but an internet connection is needed. This is basically the same principle as encoding the information, but now the code is a website address. The two most suitable options are encoding or using a website, and both don't require a lot of data. If it is assumed that two alphanumeric characters are needed per variable and that there are 6 variables. An alphanumeric string of 24 should be able to contain enough information to store the base and extra. So, for the manufacturers, the most important data is the product information which does not take a lot of data, 24 alphanumeric characters at most.

If the code is used not only at the manufacturing location but also by the user of the final product, more functions can be added. In this use case, the manufacturer should think of what information is relevant for the user and what they would rather keep to themselves. Papp. et al suggest using a QR code to replace user manuals. This can be done by embedding a website portal in the code and embedding the code in the product [28]. Additionally, information that the user might find useful can be added such as product warranty and contact information.

2.2.2 Types of codes

In this section, the different codes that are available are discussed. There are many types of codes available. The first main distinction that is made is between 1D and 2D codes. 1D codes are barcodes, and there are many types of those. These codes represent data by varying the width of and spacing between parallel lines [29] [30]. The length of a 1D code determines how much information can be stored. Generally, the code length is limited from 8 to 15 characters.

The other codes that are available are 2D codes, also referred to as 2D matrix barcodes. There are also several types of 2D matrix barcode codes available. In these types of codes, data is represented with two-dimensional symbols, often pixels. These pixels are called modules. The code works by identifying each module as positive or negative, this is determined based on the contrast between modules. Using the positive and negative modules the data in the code can be decoded. 2D matrix barcodes have two advantages over their 1D counterpart [29] [30]. The 2D code is able to hold more data per unit area than 1D barcodes. Additionally, 2D barcodes are able to correct errors. This means that if part of the code is unscannable, the code as a whole can still be scanned without a loss of data. A disadvantage of 2D barcodes is that sometimes more complex scanning devices are required in order to scan the code. In this research, the focus is on 2D-matrix barcodes as they have more extensive error correction capabilities and are able to contain more information than their traditional 1D counterparts. Because of this, it would be more beneficial to embed 2D matrix barcodes on products.

2D matrix barcodes

There are five types of codes that will be taken a closer look at: QR codes, PDF417, DataMatrix, MaxiCode and the Aztec code.

QR codes

The most common type of 2D-matrix barcode is the quick response (QR) code. The data storage efficiency and quick decodability made the QR-code popular [31]. QR codes were introduced and used since the age of the Internet as a 2D matrix barcode. They are printed on documents and products for tracking, identification, and management [32]. The QR code is originally developed in Japan [11]. It was developed by Denso Wave for Toyota to track car parts during manufacturing.

In a QR code the data is stored in modules, the black and white pixels. The data can be stored in several formats, but is most often stored as an alphanumeric string. The number of characters that this string contains determines how much data can be stored in the QR code. In this section, several types of QR codes are discussed and compared. Models 1 and 2, micro, iQR codes, logo and secure QR codes will be compared. Important information on the different types of QR codes is how much data can be stored and what the error correction capacity of the code is.

In Figure 2.11 a type 2 QR code is shown. Some terminology is explained based on this image. The detection patterns are marked in red and the alignment pattern is shown in blue. Each pixel is called a module. Around the code is a quiet zone, this zone should have one colour. To read a QR code first the detection patterns have to be found. Based on these patterns the size and position of the code can be determined. Next, a module grid is constructed and the value of each module is determined based on the contrast. Several authors worked on decoding robustness of QR codes. Van Gennip et al. worked on regularization to blind, deblur and denoise the QR codes. [33]. Chu et al worked on the stabilisation of QR codes to make them more readable from mobile devices [34]. As a result, there are QR code scanners available that are able to compensate for non-ideal circumstances. The result of this is that 'unclear' QR codes can also be read.

The decoding robustness is how much of the QR code needs to be read in order to decode the QR code. If a part is missing, the QR code can in some cases still be decoded. If the QR error correction code level (ECC), or the robustness, is higher, less of the QR code needs to be read in order to decode the code. The ECC level determines how robust the QR code is, so how much of it is needed to be understood by the scanning device [35]. There are four levels of ECC. If the level is lower, the storage capacities of QR codes are larger. The different error correction levels along with the percentage of the code that can be unreadable while still decodable is tabulated in Table 2.4.

Table 2.4: The different error correction levels of QR codes according to QR code tiger [36].

Error correction level	L	M	Q	H
Error correction capacity %	7	15	25	30

Model 1 and 2 QR codes

The most commonly used type of QR codes are models 1 and 2. These are the codes that you think of when QR codes are discussed. In Figure 2.11 an example of a model 2 QR code is shown. Next to that, in Figure 2.12 a model 1 QR code is shown. The original QR code was a model 1 code. Model 2 is an iteration on model 1, and in this iteration, it has been improved. The improvements are that with the same amount of modules, more data can be stored. Also, more modules can be used in a single QR code. The most important difference might be the addition of an alignment pattern in the right bottom corner, shown as blue in Figure 2.11. This increases detection accuracy. Because of this model 2 QR codes are more commonly used nowadays. Model 2 has 40 different versions.

The model 2 QR codes are classified into 40 versions according to the number of modules they contain, i.e., a QR code of version $V = 1, . . . , 40$ includes $(17 + 4 \times V)^2$ modules. [31]. The storage capacity is dependent on the ECC level (Error Correction Code) and the size of the QR code. The alphanumeric storage capacity ranges from 10, at version 1, to 4,296 at version 40 [37]. In Table 2.5 the properties of the smallest type 2 QR codes are shown as found on qrcode.com. The full table can be accessed at <https://www.qrcode.com/en/about/version.html> [37].



Figure 2.11: An example of a type 2 QR code. Figure 2.12: An example of a type 1 QR code.

Table 2.5: Type 2 QR code properties from [37].

Version	Modules	ECC level	Alphanumeric storage capacity
1	21x21	L	25
		M	20
		Q	16
		H	10
2	25x25	L	47
		M	38
		Q	29
		H	20
3	29x29	L	77
		M	61
		Q	47
		H	35

Micro QR codes

Another type of code is the Micro QR code. It can store less information, but less space is required. Compared to type 2 QR codes the micro QR code is smaller, and it contains only one detection pattern instead of 3. In Table 2.6 the storage capacities of micro QR-codes are displayed. The main advantage of these QR codes is that they take up little space. However, the limited storage capacity means that an online environment is required for this QR code to function. An example of a micro QR code is shown in Figure 2.13.

Table 2.6: Three versions of micro QR codes and the size of the alphanumeric string that can be stored for each version [38].

Symbol version	Number of modules	ECC level L	ECC level M	ECC level Q
M2	13	6	5	-
M3	15	14	11	-
M4	17	21	18	13



Figure 2.13: An example of a micro QR code from Wikipedia.

iQR code

The iQR code was developed more recently and has several advantages over traditional QR codes. At first, it is able to store more information in the same amount of space. This is mainly due to a reduction in position detection patterns. As a result, more of the space can be used for modules that contain information. Because of this higher storage capacity per size, the iQR codes that would be required to store a given amount of information can be smaller than the type 2 QR codes. The reduction in the size of the detection patterns makes the code harder to find and adjust for distortion. Secondly, the iQR code does not have to be square, but can also ah a rectangular shape. Examples of both types are shown in Figure 2.14. This increases the number of places in which it can be used. Replacing a barcode is a possible field of use. Another advantage is that it has a higher decoding capability on cylindrical surfaces. A third feature is that the iQR codes can be larger than type 2 QR codes. It can have 422x422 modules, while the type 2 QR code has a maximum size of 177x177 modules [39]. Finally, it allows for a higher error correction level, being able to correct up to 50% of the code where type 2 can only correct up to 30%. Finally, the iQR code can be turned over (mirrored) and it will still be readable. The main disadvantage of the iQR code is that it can not yet be scanned by a smartphone. Thus, the application field of the iQR code is not as public as the normal QR code [39]. When embedding codes, the loss in robustness as a result of the size reduction of the detection patterns is considered a disadvantage, while the increased error correction capacity is a major advantage.



Figure 2.14: A normal (left) and a rectangular iQR code.

Logo QR code

Logo QR codes have logo's embedded in them. Another name for logo QR code is image QR code. Often a logo of the QR code user is embedded in the QR code, but other images can also be embedded. Often this is done to make the QR codes more recognizable and trustworthy, or even more aesthetically pleasing [40]. Logo QR codes don't provide any significant advantages for their appliance as a part tracker, and will thus not be used.

Secure QR code

The final QR code that will be discussed is the Secure QR code. It can store both public and private information. To do so it will encrypt certain data, allowing only devices that have the decoding key to access that specific information while allowing other information to be obtained by anybody [41]. This effectively allows two classes of data to be stored. In this application it could be ideal to have a publicly available data pool with data that is relevant to the end user and an encrypted data pool with the data that is only relevant for the producer.

PDF417

The PFD417 is a type of stacked linear barcode. This means that there are several barcodes stacked on top of each other. Those stacked codes are the equivalent of 1D barcodes. The code derives its name from the way the code patterns are specified. In this code patterns are specified to consist of four bars, and each pattern is 17 units long [29]. An example of a PDF417

code is shown in Figure 2.15. In Figure 2.15 each row is basically a 1D barcode.



Figure 2.15: An example of a PDF417 code from <https://www.barcodefaq.com/2d/pdf417/>.

DataMatrix

A data matrix is a two-dimensional barcode [29]. In Figure 2.16 an example of a data matrix is shown. It does not have recognizable features like the detection patterns that a QR code has. Instead, the L shape is used for detection. This takes up less space compared to the detection patterns in QR codes. As a result, the DataMatrices have a higher information density than QR codes [42]. Advanced error-checking and correction algorithms are available. The ECC level can be increased, but increasing this level results in a code with a lower information density. In Table 2.7 the data storage capacities of DataMatrix according to jgraph.net are shown [43]. The values in this table have minimal ECC. The relation between data storage capacity and ECC in DataMatrices is flexible, of the code is larger but less data is stored, then it will automatically have a higher ECC. In industrial practices, the DataMatrix is the most commonly used code. For industrial applications, Data Matrices are better because of the more dense data storage capacity. However, advanced scanners are needed. Because of this, for general purposes QR codes are superior.



Figure 2.16: An example of a DataMatrix from https://en.wikipedia.org/wiki/Data_Matrix.

Table 2.7: Code size and maximum data storage capacities of DataMatrices according to jgraph.net [43].

Size	16x16	20x20	24x24	26x26
Alphanumeric storage capacity	16	31	52	64

MaxiCode and Aztec code

The final two types of matrix barcodes that will be discussed in this section are the MaxiCode and the Aztec code. These codes are similar in that the detection pattern is centrally placed in the code. An example of a MaxiCode is shown in Figure 2.17, and an Aztec code can be seen in Figure 2.18. Both have a central detection pattern. A difference is that the MaxiCode uses a hexagonal grid whereas the Aztec code uses a square grid. The size of a MaxiCode is fixed, and it can store up to 93 characters[44]. The size of the Aztec code ranges from 15x15 to 151x151. At its largest size, it can code up to 3067 alphanumeric characters [45]. The sizes and data storage capacities of the smallest Aztec codes according to [barcodebakery](http://barcodebakery.com) are shown in Table 2.8. The ECC of Aztec barcodes can be set between 0 and 99%, it is unclear what the used ECC in the table is, but probably it is 0. The mechanism that is at work here is expected to be similar to the Data matrix, where the data storage capacity reduces as the ECC increases. There also is the option to make the Aztec barcode more compact, which excludes a reference grid. This is not problematic as it concerns codes of small sizes. Because both codes only have a single detection pattern they are harder to find and it is harder to determine the orientation.

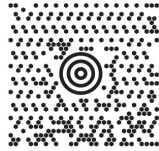


Figure 2.17: An example of a MaxiCode from <https://www.cognex.com/resources/symbologie/2-d-matrix-codes/maxi-codes>. Figure 2.18: An example of an Aztec code from https://en.wikipedia.org/wiki/Aztec_Code.

Table 2.8: Code size and data storage capacities of Aztec barcodes according to barcodebakery [45].

Size	15x15	19x19	23x23	27x27
Alphanumeric storage capacity	-	15	40	68
Compact Alphanumeric storage capacity	12	33	57	89

2.2.3 Conclusion

The code that would be most suitable to embed onto products would be a 2D matrix bar-code. The reason for this is twofold. At first, more data can be stored in a smaller space. Secondly, 2D matrix bar codes have the possibility to have error correction. This error correction capacity is very relevant for the application. Design guidelines can be established for 2D matrix barcodes in general. This is because the working principle of the 2D matrix barcodes are similar. There is contrast between modules, and through this contrast, the code can be scanned. This is the same for all these 2D matrix bar codes. Thus, design guidelines for these types of codes can be generalized as the underlying principle of generating contrast between the modules of the code. In the final guidelines, a guide should be provided to select the most suitable type of code. The code should have an ECC of H (25%) or Q (30%). The highest information density can be achieved using a DataMatrix or an Aztec Code. However, it is unclear how the data storage capacity is exactly affected if the ECC is increased. The Aztec code is less suitable as it is harder to determine the orientation of the code, while QR codes and DataMatrix codes have more elaborate detection patterns. A DataMatrix code is recommended if the scanners and systems are available. If not, a type 2 QR code should be used.

2.3 3D-printing QR-codes

In this research, the goal is to develop guidelines for embedding codes onto SLS 3D-printed products. In the previous sections, the basic workings of selective laser sintering and 2D matrix barcodes have been discussed. The limitations to and the resolution that can approximately be achieved using selective laser sintering are identified. The readability of codes relies on the contrast between the positive and negative modules and the ECC. This basic knowledge is used to understand how to embed codes onto 3D printed products, and research that already has been done on the embedding of codes is reviewed.

In this literature research, the main focus will be on finding literature about embedding codes onto 3D printed products. At first, the process to embed a code is discussed. Secondly, the different approaches that have been taken to generate contrast and predict the lighting of the carved surfaces are discussed. Thereafter, the research that has been conducted on the embedding of QR codes on curved surfaces is reviewed.

2.3.1 Embedding codes

The most common code that is embedded is a QR code. When a QR code is embedded on 3D printed products, it is called a 3D QR code. The concept of a 3D QR code was introduced by Kikuchi et al. [10]. In their research Kikuchi et al. [10] embed QR-codes onto products. Kikuchi et al. use surface modifications to generate contrast. The surface modifications that are made consist of carving out part of the product, and casting a shadow to create contrast. The questions that have to be answered in order to be able to establish design guidelines are: "What goals should be set when embedding codes onto 3D printed products?" "What parameters are relevant when embedding codes onto 3D printed products?" "What should be the values of these parameters?"

The main goal of this section is to investigate what goals should be set for the embedment to work and to map out what parameters influence how well the code is embedded.

The process of embedding a code

In literature, a basic step-by-step approach to embed QR codes onto 3D printed products is proposed by Kikuchi et al. [10]. At first, the position where the code should be embedded as well as the size of the code is determined. Next, Kikuchi et al. preprocess the QR codes by defining the grid on which the QR code is placed [10]. They add a buffer zone alongside the QR code so that the negative modules will not be dragged inward. Thereafter, a 3D Cartesian coordinate system is constructed. The QR code is placed on the xy-axis. Next, knot lines are generated and based on the knot lines the surface is modified and grooved. This groove provides contrast. However, this contrast is not always sufficient. Papp et al. improved upon this method by eliminating the preprocessing step to save time [28]. Only the borderlines are remeshed to connect the QR code with the rest of the mesh.

Generating contrast

There are different options to generate the required contrast. The first and most straightforward option would be to use materials of different colours. Using this option has the obvious advantage that if the colours are different sufficient contrast is created, and the QR code can be read. It does however require the used printer to be able to print using two material types. Since this option is not available when SLS printing, a solution that utilizes single material usage is investigated.

If the printing method would be FDM it would be possible to print material of two different colours. This requires an advanced setup, as consumer-level 3D printers normally print using a single material at a time. Thus, this technology would not be available to all FDM printers and it would be more universally applicable to search for an alternative that uses a single material.

Another option to generate the required contrast is to manipulate the surface of products. The idea is that products cast shadows onto themselves, causing contrast. White modules of the QR code are obtained by casting direct light onto those modules. Black modules are obtained through surface cast shadows, carved surface geometry and through yielding self occultations. This method has been proposed by several authors. Grateguy et al. embed QR codes into images using luminance manipulation (shadows) [46]. They generate contrast by modifying the surface of a single material, causing self-shadows to appear. Kikuchi et al. [10] also make use of this option. Peng et al. and Papp et al. developed this method further [11] [12]. For these guidelines, contrast is generated by self-shadows so that a single material can be used.

Two major surface modifications are often used in literature to generate contrast/shadows. The first is to simply carve out or heighten regions of the QR code, generating shadows. Kikuchi et al. Papp et al., and Peng et al. make use of this carving option [10] [28] [11]. Grateguy et al. Kikuchi et al. Papp et al., and Peng et al. confirmed the possibility to embed codes using self-shadows. Two important parameters in this process are the depth and the width of the carved-out part. These two parameters should be varied in order to determine what settings generate high contrast. When embedding codes, this carved-out surface could be a module of the respective code. As a result, the precision of the printer determines the minimum module size.

A second option is to add a surface texture to the printed surface. The different modules have different roughness and as a result, reflect different amounts of light. Papp et al. suggest surface modifications of larger areas that are supposed to be black to break the surface. When using self-shadows these larger black areas become problematic [28], as in larger low areas no shadows are cast, and as a result, these areas are no longer recognized as black. Papp et al. modified the bottom of the black regions to reflect less light. Several modifications for the black modules are discussed. This ranges from modification of the shape to carving out negative modules with a non-uniform depth.

So, to generate contrast several actions can be taken. It is possible to lower or heighten the surface and to apply a surface texture. When establishing guidelines the creation of contrast is vital. In order to guarantee this surface contrast it should be established what carving depth generates sufficient contrast. Next to that, the minimum module size that can be achieved by the printer should be determined. Finally, to what extent surface textures should be utilized needs to be determined. Three subquestions are formulated:

'What should the guidelines for carving depth be to generate sufficient contrast?'

'What is the minimum module size that can be achieved using the selected printer settings?'

'How can and to what extent should surface modifications play a role in the design guidelines?'

Determining the position

Depending on the product on which the code is to be embedded, the ideal position to embed the code should be determined. Papp et al. propose an algorithm to determine the ideal position of the code. With a specified centre the best size and the position of the QR code can be determined. A preferred position and minimum size are inputs [28]. The position of the code influences two aspects. At first, the surface properties at that location are set by the design of the part, so surface properties such as the surface curvature are dependent on the location.

Kikuchi et al. [10] worked on embedding codes onto freedom surfaces, so non-flat surfaces. Their work was focused on embedding codes on B-spline surfaces. According to Kikuchi et al., the curvature of the surfaces has a large influence on how well the code can be embedded. Secondly, the position determines how easily the code can be found, and thus scanned. The question that should be answered to establish design guidelines is:

"Where on the product should be code be embedded?"

"What are the requirements for the surface on which the code is embedded?"

Curved surfaces

One restriction that remained in place when printing QR codes was that the surface on which the QR codes were to be printed had to be flat. The ideal surface to place a QR code on is a flat surface. This avoids deformation when embedding the code and avoids distortion as a result of curvature. Kikuchi et al. were the first to embed 3D-printed QR codes on curved surfaces [10]. The curvature of the surfaces was represented with a B-spline curve. Their method requires three inputs to generate 3D QR codes. 1) A QR-code of a certain size 2) A freeform B-spline surface 3) A 3D point on the surface that is specified where the QR-code will be projected. Kikuchi et al. assume orthogonal projection when determining the reflection. Papp et al. improved upon the work of Kikuchi et al.[12]. They discuss the projection method that is to be used. Central projection is preferred, parallel projection works also but is less robust. Iteration is used to gradually improve the position of the QR code. The total normal vector of the QR code is maximised. In this way, the flattest surface is found. Li et al. embedded barcodes on cylindrical surfaces. This can help with the decoding of QR codes on curved surfaces that are not looked at from a normal view easier. [47].

Embedding codes onto curved surfaces using self-shadows causes problems for several reasons. At first, it introduces distortion of the code. Secondly, the curvature of the surface causes an uneven light profile. Two additional questions are added:

"How does the surface curvature affect the generation of contrast?"

"How does the surface curvature of the products affect distortion, and thereby the readability of the code?"

It can be concluded that the feasibility of embedding codes using 3D printing onto freeform surfaces has been proven. However, it needs to be clarified what surface curvature is how problematic.

Lighting conditions

The lighting conditions determine how the product is lit, and thereby also where the shadows fall. This in turn, greatly influences the generated contrast. Kikuchi et al. and Papp et al. use ambient lighting. In order for the codes to work in ambient lighting conditions a higher carving depth is required.

Peng et al. used directional light to create the contrast required to scan the 3D-printed QR codes [32]. In this option, light is assumed to come from a certain angle. Their focus is on creating unobstructive QR codes that are minimal in terms of carving depth and shape modification, but sufficient in terms of pattern contrast and casted shadows. They minimize the editing of the original shape, persevering visual appearance and geometry [32]. It results in codes that are much less intrusive. However, it should be noted that the readability of these codes is only high under specific circumstances. The type of lighting influences the amount of contrast that is generated.

It is clear that the lighting conditions have a large influence on how the code is scanned. This prompts two questions:

"What should be the scanning environment?"

"How should the environmental (lighting) conditions be controlled to generate sufficient contrast?"

Context

There is some research that borders this application but is not quite as relevant. Artistic applications of self-shadows are investigated by Alexa and Matusik [48]. They create artistic images using shading due to self-occultation. A material with a uniform albedo is used. The ways that are used to generate contrast, and in this case even images might help when trying to form codes.

Yang et al. try to embed artistic QR codes onto a thin shell. Their research focuses on perforating this QR code from a plate. The resulting model can be 3D printed as well as perforated. It requires no under-laying surface. This means that some of the parts of the QR code would not be connected to the rest of the code. These parts need to be connected somehow. The white modules of the QR code are defined as positive modules: these will be produced while the black modules won't be present. If these modules are not connected in one piece they would simply fall out of the code. They propose an algorithm to connect all the white modules of the QR code. To do so, black modules are made white. This results in a loss in decodability. In their optimization, they attempt to minimize the loss in decodability. However, it does allow the code to consist of a single positive shape. The perforated plate can be seen as a top layer that is placed over the rest of the product, similar to a surface modification [31]. This might be beneficial if it was decided to embed codes using FDM printing because the nozzle does not have to switch on and off as much.

Robustness

There are three levels of robustness in the decoding of the code. At first, is the Salmon Reed error correction. This is the ECC that was discussed in the section on codes. Secondly, there is the binarisation algorithm, transforming the grey tones into either white or black pixels. Finally, there is the symbol constant constraint. This means that during the calculation a distinction is made between the lightest and darkest modules.

Goals for successful embedding

When establishing design guidelines, there are several goals. At first, the applicability of the guidelines should be high. It would be better if the guidelines can be universally applied. This applicability will be an objective of the research. The two main contributors to this applicability are readability (ensuring that the code can be read) and printability (making sure that the details in the code can be printed). The research will strive to develop guidelines to embed readable and printable codes while aiming to avoid negatively affecting the functionality of the mechanics and visual properties of the products.

In the research of Peng et al. [11] there are three objectives for the embedded QR codes: 1) be unobstructive, by what they mean that QR code should be minimally visible when using to product and not intervene with the function of the product, 2) be printable and 3) be readable. These objectives, unobstructive vs readable and printable, conflict and thus it should be seen as an optimization problem. The unobstructiveness will be added to the two goals that were already identified.

In order to be able to read the code, good alignment, sufficient resolution and high contrast are required [11]. These three parameters determine the quality of the embedded code. The alignment and resolution are mainly dependent on the capabilities of the printer, whereas the contrast is mainly determined by cast shadows, which are in turn dependent on the severity of

the surface modifications and the lighting conditions. The minimal size of the modules of the code is determined by printer capabilities. The clarity of the alignment patterns is also dependent on the printer capabilities and the generated contrast. An additional factor that is at play is distortion if the code is placed on a curved surface. To minimize the obstructiveness of the code the code should be minimally visible in everyday use, not interfere with the functionality of the product and be as small as possible. The subgoals are dependent on the severity of the surface modifications and the resolution that can be achieved by the printer. All objectives are tabulated in Table 2.9.

Table 2.9: Goals, subgoals and influencing factors.

Goal	Subgoal	Influencing factors
Readability	High contrast	Lighting, surface properties and surface modifications
	Sufficient resolution	Code size, printer capabilities
	Clear alignment patterns	Code size, printer capabilities, contrast and distortion
Printability	Printer able to meet requirements	Printer capabilities
Unobstruc- tiveness	Minimal visibility in everyday use	Surface modifications
	Don't reduce the functionality	Surface modifications
	Minimal code size	Sufficient resolution

For the design guidelines, the most important influencing factor to research is how to generate contrast at a sufficient resolution. The surface modifications and the conditions lighting should be looked at. Also, the printer capability for this specific scenario should be tested. The design specifications of the specific product is taken into account by looking at how to generate contrast on different surfaces. For example, how do curved surfaces influence the cast shadows? No specific research is done on where on the product the code should be embedded, but some advice is given based on common sense.

2.4 Literature gap

Using a table the aspects to which each of the papers has contributed are mapped. Only the literature research on embedding codes onto 3D printed products is taken into account, as the literature on SLS printing and 2D matrix codes is plentiful. In Table 2.10 the subjects that are discussed are tabulated. The first four subjects are the type of additive manufacturing that is used in the research. The next four subjects are about the location of the code and the lighting used. The seven subjects after that are about how and where the code is embedded. Thereafter, two methods that can be used to in image processing are the subjects and finally, the focus of the papers is shown.

Table 2.10: Subjects related to embedding 3D codes and the amount of literature available.

Subject	Nr of references	References discussing it
Embedding QR codes in 3D printed products	8	[10] [6] [11] [2] [32] [12] [28] [31]
FDM	6	[10] [6] [11] [12] [28] [31]
SLM	2	[6] [2]
SLA	2	[32] [6]
SLS	0	-
Using shadows	7	[10] [11] [2] [32] [12] [28] [6]
On the product surface	6	[10] [11] [2] [32] [12] [28] [6]
Inside products for authentication	2	[6] [2]
Directional lighting used	1	[32]
Algorithm embeds code in CAD model	6	[10] [6] [11] [32] [12] [28]
Size optimization using algorithm	1	[11]
Carving depth optimization using algorithm	1	[28]
Code rotation optimization using algorithm	1	[28]
Location optimization	0	-
Pattern utilization	2	[12] [28]
General principles	0	-
Algorithm available	0	-
Distortion compensation for codes on curved surfaces	2	[49] [47]
Advanced module identification	2	[6] [2]
Focus on consumers	6	[10] [6] [11] [12] [28] [31]
Focus on industry	0	-

In literature, a lot of attention has been paid to the theoretical possibility of embedding codes. Advanced algorithms to embed the codes have been developed, and the principle of embedding codes has been proven. The main focus is on consumer-level printers. However, little attention has been paid to practical applications in industry. The algorithms that are used are not available and general principles are not established. As a result, the technology is not available to the public. To make the technology more accessible, design guidelines based on general principles should be developed. With the establishment of design guidelines, the technology can be applied in industry. In order to establish design guidelines the parameters that have the highest influence on the readability of the code need to be optimized. This is problematic, as it depends on the surface, place and application what is optimal.

In most found articles FDM printing is used. However, in this research, the focus is on SLS. There are several reasons for the switch from FDM to SLS. At first, there is more demand for the technology to be suitable for SLS products, as multiple products will have to be identified

and removed from a build volume. Secondly, when FDM printing adding a code to a product adds production time because the nozzle has to switch on and off a lot and extrude many small amounts of material. It takes approximately one hour extra to embed a code on an FDM product, and as production time increases so do production costs. Next to that, when embedding codes using FDM printing, overhang becomes a problem when codes are not embedded horizontally. Experiments have been run with FDM, but achieving a sufficient resolution was problematic. Some documentation on these FDM experiments can be found in Appendix C. With SLS a higher resolution can be achieved, the overhang is not a problem and there is a use case.

Another aspect that has received little attention in the found literature is how to locate the code on the printed products as they are removed from the printer. If time permits, this will be the second part of this graduation project. If not, further research is required to find out how to implement this aspect when determining the desired position.

2.5 Research questions

Based on the literature research several sub-questions to help answer the are formulated. The main research question is: "What design guidelines should be established to allow readable QR codes to be embedded on industrial products that are produced using SLS 3D printing?"

Several questions have already been answered during the literature research, while some require experiments to be answered. Questions that have already been answered are about the process of SLS printing and the information-carrying devices.

SLS printing will be used as it is the most relevant printing method where this embedment is needed in the industry. It is proven that embedding 2D matrix barcodes onto 3D printed products is feasible. The minimum feature size is roughly determined and will be exactly determined through experiments. The printing parameters that play an important role are identified. Laser, scan and temperature properties are assumed to be fixed if the machine is configured correctly. An important powder property is the amount of recycled powder that is being used, this property will be fixed as well.

About the information-carrying devices, the question that was asked was: "What is a suitable storing device for the information to be carried?" The answer to this question is that a 2D matrix bar code is the most suitable. A 2D matrix bar code functions through the contrast between the different modules (pixels).

The problem of embedding the code can be divided into roughly two parts: how to embed the code and where to embed the code. How to embed the code consist of generating enough contrast. In literature, this is done through the generation of self-shadows. When doing this there are three modifications to be made. The two first two modifications are the depth and width of the carve The other modification that can be made is a texture modification in the carved-out areas. When embedding the code the minimum size of the code is set by the printer capabilities.

Next to that, there is the location of the code that needs to be determined. Two important factors to consider for this are the size that is available and the surface shape at the given location. Next to that, the code should not impede the functionality of the product. Also, it should be findable for the camera.

It was found that the lighting will influence the results. By controlling the scanning environment,

the lighting is controlled.

In order to tackle the problem several sub-questions are formulated, which are listed below. The first three questions serve to answer the "how" part of the embedding. Questions 4 and 5 help to set guidelines on where to embed the code. Questions 6 and 7 are about the scanning environment. The answer to the questions will be determined through experiments:

1. What is the minimum module size that can be achieved using the selected printer settings?
2. What should the guidelines in terms of width and depth for carving depth be to generate sufficient contrast?
3. How and to what extent can surface modifications play a role in the design guidelines?
4. How does the surface curvature affect the generation of shadows through carved sections?
5. How does the surface curvature of the products affect distortion, and thereby the readability of the code?
6. What should be the scanning environment?
7. How should the environmental (lighting) conditions be controlled to generate sufficient contrast?

3 RESEARCH METHODOLOGY

In order to answer the research questions several experiments will be conducted. In the experiments, the parameters that were identified in the literature research will be varied to be able to answer the sub-questions. In that way, these parameters are taken into account when finalising the design guidelines. Different samples will be printed and four images will be taken to evaluate how much contrast is generated and if this contrast is in the correct position. This evaluation happens using a Matlab code. This code generates a score indicating how well the different samples are performing. In this way, the results of the experiments are quantified. There are three important factors that need to be discussed: the setup, the samples and the post-processing using that code.

Several external parameters were identified during the literature research. They can be grouped into three categories. At first, there is the printer. Printer properties, powder properties and the amount of recycled material that is used are relevant factors. These factors can be varied. For these experiments, it is assumed that these are fixed already as the machine is configured correctly. Next to that, there is the scanning environment: lighting, lighting direction the type of scanner, sensing modality of the camera and the scanning angle. An optimal scanning environment for the basic sample will be determined, and the scanning environment will remain fixed that way. Finally, here are product properties: surface roughness and curvature, size, space available, carving depth and surface modifications. The effects on the embedment of the code of these parameters will be tested by varying them from a standard sample. The variation of colour is left out of the research.

To help answer the research questions one parameter will be varied, while the other parameters remain fixed. In order to be able to do this, a baseline situation will be created. This baseline consists of two parts, a standard setup and a standard sample. At first, the setup will be determined. This is done by trying out several setups and seeing how the results in terms of contrast and accuracy of the standard sample differ. The standard sample is selected based on a first iteration, where the code is embedded using trial and error. It is assumed that the results of the embedding parameters are independent of the setup.

Once the standard setup is decided upon, the parameters involved with embedding the QR code will be varied. These parameters are the size of the code, the carving depth and width, the effect of extra surface modifications and the effects of surface curvature. One of these parameters will be varied while the others are at the value of the standard sample to see the effect on the generated contrast and decoding accuracy.

3.1 Procedure

When conducting the experiments, a standard procedure is followed. It consists of three steps. At first, several prints are made in which one parameter is varied from the standard sample.

In order to be able to make these prints different CAD models will be made, and then these will be printed using SLS 3D printing. All parameters will be varied over a certain range. If a single parameter is varied it should have 5 different values while if two parameters are varied together they both should have 3. By varying the parameters this way, it is expected that a correlation between a parameter and the score is shown. Also, it limits the number of samples that are required. It is expected that the variation causes a change in score, this relation being a causal relation. The properties of the standard or baseline sample are tabulated in Table 3.1. SC is the abbreviation for surface curvature, while ST stands for surface texture. The surface textures are extra negative shapes that are in the negative modules. The depth of these extra shapes will be varied, hence both the form of the shape and the depth of the shape is in the table. In Figure 3.1 the parameters that are varied (except for the surface curvature) are shown.

Table 3.1: Properties of the baseline sample.

Sample name	Size [mm]	Module size [mm]	Carving depth [mm]	SC [°]	ST	ST depth [mm]
Base	20	0.69	2	-	-	-

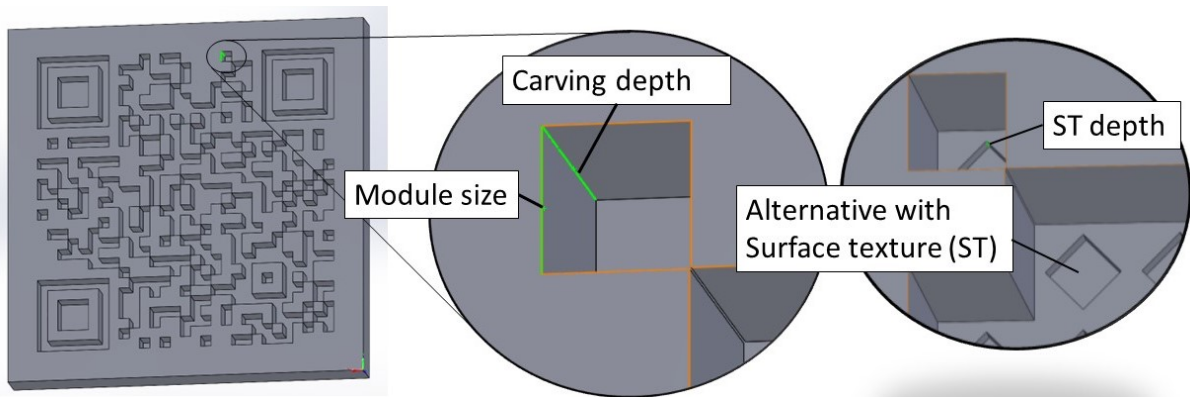


Figure 3.1: Parameters that are varied shown in the CAD model.

Secondly, images of the printed QR codes will be taken in controlled conditions. Five images are taken during this processing. One with all lights on, and four with light from only one direction. This directional lighting is used to increase contrast. Each sample is processed three times to test consistency. In this step images of the samples are taken in a fixed setup. Several setups are proposed, and three samples are processed under each of these setups. Based on this result a standard setup is selected that will be used during the rest of the experiments. How this setup is selected is explained in paragraph 3.2.

As a third step, these images are processed using a Matlab script. The goal here is to determine the amount of contrast generated at each module for each sample, and whether or not this contrast is in the desired location. In that way, more insight is created into why the scanning of a QR code does (not) work. Also, it is possible to observe warping effects. Finally, a score will be determined based on the amount of contrast that is generated and if this contrast is in the right location. How this code is written and what it does is elaborated in paragraph 3.3. The three steps are shown below:

Step 1 Print samples.

At first, several sample sets are printed using the SLS printer. The samples are modelled in SolidWorks beforehand and are modified from the base sample in one or more parameters.

Step 2 Take images in controlled conditions.

Next, the samples are photographed under controlled conditions. This is done using a setup that will be determined and will remain fixed as the samples vary. The setup consists of an industrial camera and controlled lighting. Of each sample four images will be taken, these are to be processed to obtain scores to see how well the sample did.

Step 3 Analyse images & (post) processing results.

Finally, the four images will be processed. This is done to determine how well the different samples generated contrast in the right position. To be able to analyse and process the results a Matlab script is developed. The results of the four images are combined to generate one score. The score consists of two factors: 1) The amount of generated contrast. This encompasses the amount of contrast that is generated on average between the detected white and black modules. In order to be able to read the code a certain amount of generated contrast is required. The goal is to reliably score just above this threshold, as it is expected that the embedment of the code is least intrusive while still reliable at that point. 2) If the generated contrast is in the correct location. This means that the location of the positive and negative modules corresponds to the reference code that was embedded. Through image processing the four images the score is determined. Because this score looks at two values, the generated contrast and accuracy, it will be able to quantify how well the samples perform. This is better than just looking at whether the code does work or not because the reasons why the code does (not) work are explored.

Afterwards Post processing results

With the score of all samples determined, the relations between how well the code is embedded and what modifications were made to that sample will be investigated. In that way, the effect of varying different parameters can be observed. By mapping this effect for all the different parameters it will become possible to develop design guidelines. The steps of this post-processing are listed below:

1. Do contrast analysis through image analysis using the code, generating a score
2. Analyse results per set to see the effects of modifications
3. Relate the results to the different variations
4. Fit a scoring curve with the varied properties, identifying if and how the parameter influences the generated contrast
5. Answer a research question and conclude what this will mean for the design guidelines

3.2 Setup

Several variations of the lighting environment will be created, and multiple pictures of several samples will be taken. In that way, the setup that creates the most uniform contrast can be determined.

At first, the basic setup that will be used needs to be determined. This baseline setup will be used in further experiments. The questions that are supposed to be answered are: '6. What should be the scanning environment?' and '7. How should the environmental (lighting) conditions be controlled to generate sufficient contrast?'. These questions will be answered by testing out several lighting configurations on the baseline sample. The properties of the baseline sample are shown in Table 3.1.

The setup consists of an overhead camera that is adjustable in height. It looks down on a white background where the samples will be placed. The height of the camera is set so that an area of 50x50 is visible. This camera height will remain fixed at 400 mm. An IDS-UI-3590CP camera with an IDS-10M11-C1616 lens is used. An image of the conceptual setup is shown in Figure 3.2.

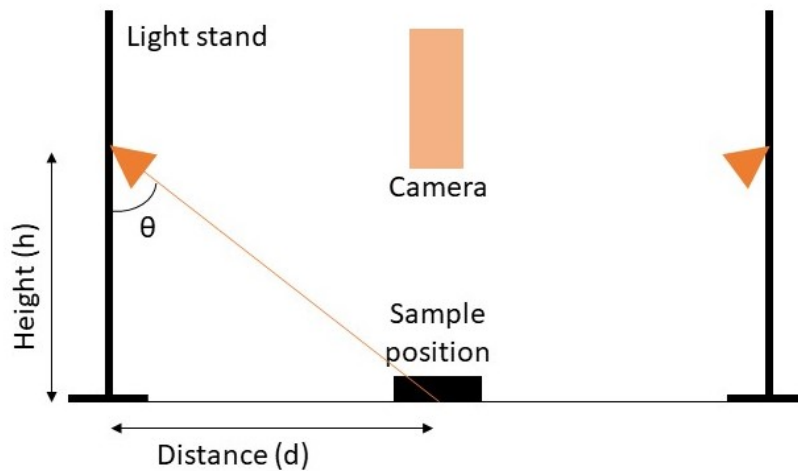


Figure 3.2: Conceptual side view of the setup showing two light stands, the camera and the sample position with the height, distance and angle θ indicated.

A part of the setup that does require some experimentation is the lighting. The idea is to have directional lighting from four directions, taking four images and combining them when post-processing the data. This allows shadows to be cast more clearly than when ambient lighting is being used. For the type of lamp that is used a GU10 TRADFRI LED 345 lm lamp is selected. This lamp is chosen because the light intensity can be varied and there is little diffusion of the light.

A conceptual top view of the setup with the four lamps is shown in Figure 3.3. There are four lighting sources (orange) that each holds a lamp at an angle of 90 °from one another. The position (height, distance) and intensity of these lamps will need to be determined. The light intensity is measured using a voltcraft MS-1300 lux meter that will be placed next to the sample. The light intensity is measured when all lights are on.

Several variations will be made for the height, distance and light intensity. The values of the height, distance and light intensity used in these variations are shown in Table 3.2. All options will be tested with all other options. This means that 36 (4x3x3) different setups are tested. The angle of the lamps is determined to be aimed directly at the sample. The angle of the lamp will be determined by the height and distance. When varying the setup this will be used as the average value, one value being above that and one below it.

Table 3.2: The different values of the parameters concerned with the lamp setup.

Height	70 mm	120 mm	170 mm	220 mm
Distance	220 mm	270 mm	320 mm	-
Light intensity	+/- 1000 lux	+/- 2000 lux	+/- 4000 lux	-

Based on the outcome of these variations an optimal setup will be selected. The most optimal option is the one where the most contrast is generated on the right places with the basic sample.

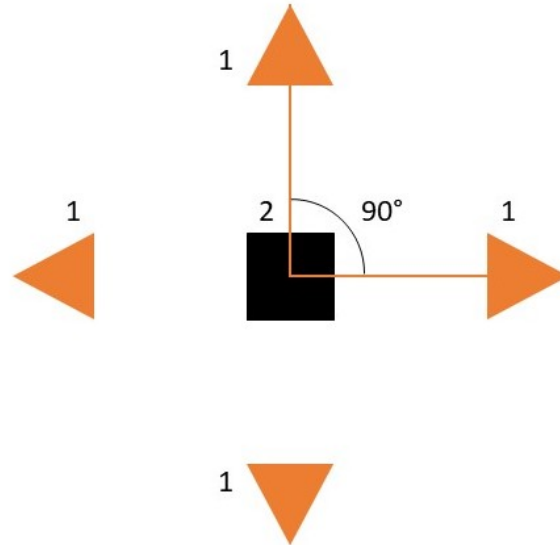


Figure 3.3: Conceptual top view of the setup with the lighting sources (1), the sample (2) and the distance.

3.3 Post processing code

In order to post-process the image a Matlab code is used. Matlab is chosen because the author is familiar with it. This code assigns scores to the different samples and thereby determines the results of the experiments. Thus, the code has a strong influence on the results. In chapter 4, it is explained how the code works and why it was set up that way. Also, the disadvantages are discussed. Here the steps that the code uses are discussed. These are listed and discussed below:

1. Import the different image
The different images of the same sample are imported so that they can be processed.
2. Convert images to grayscale
The images are converted into grayscale. In Matlab, the images are now represented by a 2D matrix instead of a 3D matrix.
3. Combine the four images into one image by adding the matrices together, dividing by four.
The different images are combined into one single image. Because the sample did not move between taking images it is possible to combine the different images to get one combined image. This still needs to be tested, it is also possible to continue to step 7 without combining the images but processing each image individually. This will be tested.
4. Binarize image
A global limit is set for the grayscale image through an algorithm. Using this bar the image is binarized, converting it into a black-and-white image. In the middle three images of Figure 3.5 a code is being converted into grayscale.
5. Find detection and alignment patterns
Next, the detection patterns are located. These will be used to determine where the QR

code is, how large it is and how it is orientated. Using this information the next step can be carried out, transforming the QR code to the reference position so that it can be compared to the reference code. An image with on the left a code with found detection and alignment patterns is shown in Figure 3.4.

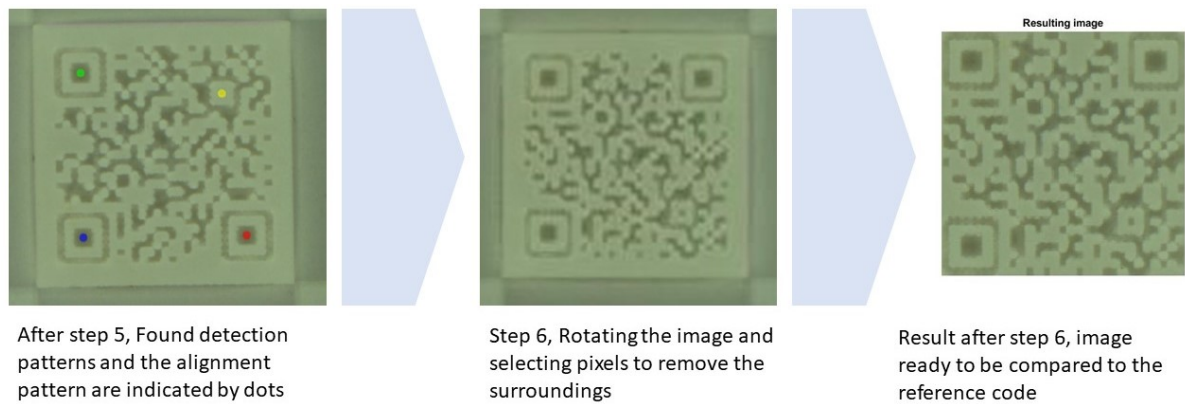


Figure 3.4: Steps 5 and 6. Left: detection and alignment patterns found, the restul after step 5. Middle: rotated image, Right: isolated code, result after step 6.

6. Transform QR code to reference position

In this step, the code is transformed into the same format as the reference code so that it can be compared. Based on the locations of the detection and alignment patterns the code is rotated and transformed. These steps can be seen in the middle and right images of Figure 3.4. The modules consist of several pixels. The average grayscale value of each module is calculated.

7. Compare reference position with actual position

The modules are compared to the reference code to see if they correspond to the reference code. In Figure 3.5 the QR code in several stages is shown. On the left, the reference image is shown. In the middle 3 images, the colour, grayscale and binarized images are shown. On the right pixels that are wrongly categorized are shown. Each pixel and module is grouped based on whether it should be a positive or negative pixel, and the grayscale values in these two groups are used to determine the score.

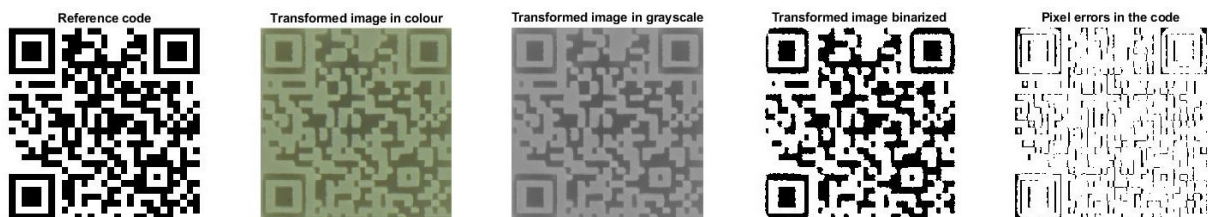


Figure 3.5: The reference code (left), isolated colour code (mid left), grayscale code (mid), binarized code (mid right) and difference between the reference code and binarized code (right).

8. Compute score

In this step, the average grayscale of the positive and negative modules is compared to indicate how much contrast was generated. This is done by looking at the data sets of what are supposed to be positive and negative pixels and modules, and identifying how clearly the two groups can be distinguished. This is based on the number of standard deviations between the average grayscale value. A detailed description of the method that is used and why this is valid can be found in chapter 4. In Figure 3.6 a histogram

of the grayscale values of pixels is shown on the left and a histogram of the average grayscale values of each module is shown on the right. The image on the left shows how the grayscale values of the pixels that are supposed to be black and white are divided. The image on the right shows how the grayscale values of modules are divided. Based on the mean and standard deviations and the number of standard deviations between the two means of the module average grayscale value the score is computed.

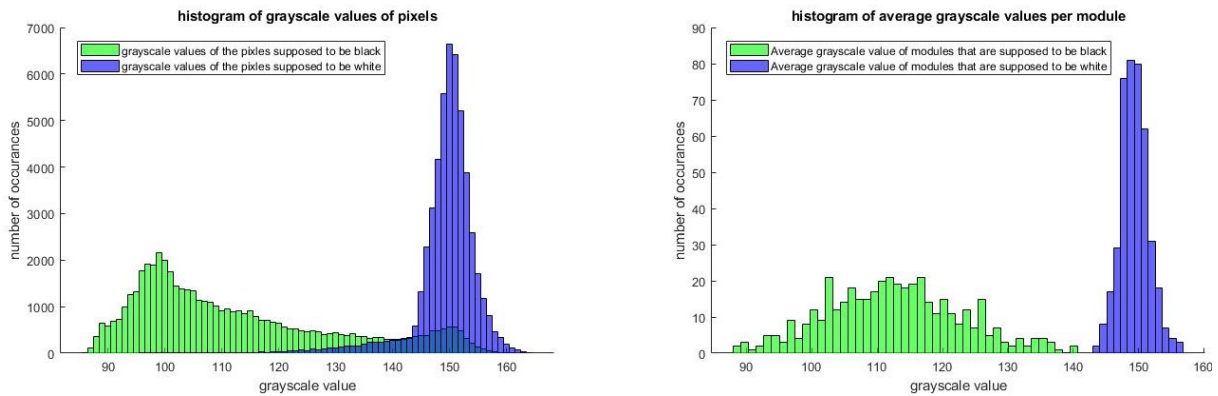


Figure 3.6: A histogram of the grayscale values of pixels (left) and a histogram of the average grayscale values of each module (right).

3.4 Experiments

The experiments will consist of five sample groups, and focuses on answering the sub-questions 1, 2, 4, 5 and 6 of the research question. The sample sets, and what questions are answered by each of the sample sets are discussed below. In the sample sets, one parameter has a significant effect according to the preliminary literature research is varied. The effect that each parameter has on the contrast, and thereby the readability is identified. Also, the prevalence of warping due to surface curvature is identified. All of the samples will be analyzed according to the procedure described in paragraph 3.1. The base sample is described in that paragraph as well, and the properties can be seen in Table 3.1. Some parameters are varied together because it is expected that there are interdependencies.

3.4.1 Sample set 1, size

The goal of sample set 1 is to help answer sub-question 1: 'What is the minimum module size that can be achieved using the selected printer settings?'. To answer this question the size of the QR code is varied on flat surfaces. The carving depth, surface curvature and surface texture will remain constant in the sample set. The proposed samples are shown in Table 3.3. In addition, a second set can be added with a curved surface where the size is also varied. This is done in sample set 4.

The hypothesis is that at a too small size, the printer is unable to accurately print the features, resulting in a loss of accuracy and contrast. At some middle ground, both the contrast and the accuracy will be high. If the modules become too large, not enough shadow is generated, resulting in a loss of contrast. The goal is to find the range of the middle ground where the contrast is generated in the right location, determine a guideline for the size of the code and

Table 3.3: Properties of sample set 1, varying size.

Sample name	Size [mm]	Module size [mm]	Carving depth [mm]	SC [°]	ST	ST depth [mm]
Size10	10	0.34	2	-	-	-
Size15	15	0.52	2	-	-	-
Size16	16	0.55	2	-	-	-
Size17	17	0.59	2	-	-	-
Size18	18	0.62	2	-	-	-
Size19	19	0.66	2	-	-	-
Base	20	0.69	2	-	-	-
Size21	21	0.72	2	-	-	-
Size22	22	0.76	2	-	-	-
Size25	25	0.86	2	-	-	-

thereby answer research question 1.

3.4.2 Sample set 2, carving depth

Sample set 2 is designed to determine what carving depth is required to generate sufficient contrast, answering question 2: 'What should the guidelines in terms of width and depth for carving depth be to generate sufficient contrast?'. The width of the carves is determined by the module size, so by the code size. It is expected that for each module size, there is a different ideal carving depth. However, to start somewhere the carving depth of the standard sample is optimized first. The size of the code, surface curvature and surface modifications will remain constant, only the carving depth is varied. A follow-up step of this experiment is to also introduce a curved specimen with the same modifications in terms of carving depth. The properties of the samples in this set are tabulated in Table 3.4.

Table 3.4: Properties of sample set 2, varying carving depth.

Sample name	Size [mm]	Carving depth [mm]	SC [°]	ST	ST depth [mm]
Base	20	2	-	-	-
CD1.6	20	1.6	-	-	-
CD1.2	20	1.2	-	-	-
CD0.8	20	0.8	-	-	-
CD0.4	20	0.4	-	-	-

The hypothesis is that as the carving depth increases, so does the amount of generated contrast. The goal of these samples is to test if this hypothesis is correct, and figure out how large this effect is. Next to that, an ideal carving depth needs to be selected. This ideal carving depth is the point at which sufficient contrast is generated to reliably read the code with as shallow carvings as possible. Shallower carvings are desirable because that means that the code is less of a design intervention. When designing guidelines it is important to find this minimum to limit the intrusiveness of the embedded code.

A recommendation for future experiments is to vary both code size and carving depth. It is expected that the optimal carving depth is relative to the size of the modules. This would have been tested if more time was available.

3.4.3 Sample set 3, simple surface curvature

In sample set 3 the surface curvature is varied, helping answer research question 4: 'How does the surface curvature affect the generation of shadows through curved sections?' It also should help provide some insight into research question 5: 'How does the surface curvature of the products affect distortion, and thereby the readability of the code?'. To be able to answer these questions the surface curvature is modified while the other parameters stay the same. The properties of this sample set are shown in Table 3.5. This sample set only includes simple surface curvature, that being uniform surface curvature in just one direction. When 3D modelling the samples they were embedded on a cylinder with a radius R. SC R is added as a column. R refers to the radius of a cylinder on which the code is embedded by creating negative modules using the given carving depth. More advanced surface curvatures such as sin waves and curvature in more than one direction are treated in sample set 4.

Table 3.5: Properties of sample set 3, varying surface curvature.

Sample name	Size [mm]	Carving depth [mm]	SC [°] x	SC R	Surf texture
Base	20	2	-	-	-
SC50	20	2	50	R13	-
SC45	20	2	45	R14.1	-
SC40	20	2	40	R15.5	-
SC35	20	2	35	R17.4	-
SC30	20	2	30	R20	-
SC25	20	2	25	R23.6	-
SC20	20	2	20	R29	-
SC15	20	2	15	R38.6	-
SC10	20	2	10	R57	-

It is expected that the curvature greatly influences the results, as the four directional lighting sources will be influenced by the surface curvature. The hypothesis is that surface curvature will introduce large differences between the four images made of a single sample. However, it is expected that by taking four images this problem can be overcome. The correlation that is expected to be found is the higher the curvature the larger this discrepancy becomes.

The other effect that is expected to be observed is the effect of surface curvature distortion of the image. Since the sample does not have the same distance from the camera everywhere this effect will show up. It is expected that this gives problems during the transformation of the code in the script. Errors in the corner without detection pattern are expected, so the error of modules that are not at the right place will go up. It is expected that this does not distort more than 5% of the code. If the code is distorted too much then the script will have to be improved by making the detection roster variable, so that it changes when distortion occurs. It should be noted that this problem might be resolved by normal scanners that have a better decoding algorithm than the Matlab script used in this research.

3.4.4 Sample set 4, advanced curvature

Sample set 4 consists of the samples that have advanced surface curvature. These can be separated into four groups. All samples are shown in Table 3.6. The first group consists of samples that have a constant surface curvature where the carving depth is varied. In the second group, the size is varied on two surface curvatures. The first and second groups combine two parameters that are expected to interact. In the first group, the parameters that are different

from the standard sample are the surface curvature and the carving depth. The second group also varies two parameters: the size and the surface curvature. By conducting these experiments it is tested if the found relations between the scoring value and size and carving depth are also valid on curved surfaces. This helps provide a better answer for sub-questions 2 and 4.

Carving depth and surface curvature are expected to interact because of the effect of the surface curvature with the directional lighting. This is expected to create large differences between the four images that are taken. Since it is expected that the carving depth plays a major role on the shadows that are cast this parameter is tested together with the effect of the surface curvature. In that way, it can be observed how the optimum (minimal interference in the product while generating a high contrast) changes when surface curvature is present. It is expected that the behaviour in this situation significantly differs from the flat situation tested in sample set 2.

Thirdly, the curvature is added not in one but in two directions. In this case, the sample is part of a ball on which a code is embedded. In the fourth case, a sin wave is used in one direction as the surface on which the code is embedded. The amplitude and number of periods this sample has on the size of one sample are varied. These two groups help give more depth to the answer to research questions 4 and 5: 'How does the surface curvature affect the generation of shadows through carved sections?' and 'How does the surface curvature of the products affect distortion, and thereby the readability of the code?'

In Table 3.6 the sample properties of this set are shown. This sample set improves the answer to research questions 2, 4 and 5 by adding another configuration and examining the interdependence of parameters. It will help determine whether a relatively simple model suffices to describe the relationship between the parameters and the embedment success or if a more complicated model is required.

Table 3.6: Properties of sample set 4, varying carving depth on a curved surface.

Parameter	Size [mm]	Carving depth [mm]	Surf curv x	Surf curv y	Surf texture
SC50	20	2	50°, R13	-	-
AC50CD1.6	20	1.6	50°, R13	-	-
AC50CD1.2	20	1.2	50°, R13	-	-
AC50CD0.8	20	0.8	50°, R13	-	-
AC50CD0.4	20	0.4	50°, R13	-	-
AC30s15	15	2	30°, R20	-	-
AC50s15	15	2	50°, R13	-	-
SC30	20	2	30°, R20	-	-
SC50	20	2	50°, R13	-	-
AC50s15	25	2	30°, R20	-	-
ACD3030	20	2	30°, R20	30°, R20	-
ACsinP1A1	20	2	sin[1,2]	-	-
ACsinP1A2	20	2	sin[1,3]	-	-
ACsinP2A1	20	2	sin[2,1]	-	-
ACsinP2A2	20	2	sin[2,2]	-	-
ACsinP2A3	20	2	sin[2,3]	-	-

3.4.5 Sample set 5, pattern inclusion

In this sample set the surface is modified further after carving. An extra shape is added in the negative modules. The variations are shown in Table 3.7. What shape is removed is denoted by ST, surface texture. This shape is removed inside the negative modules. The depth of this carving is denoted by ST depth, the depth of the surface texture modification. Two shapes are tested that are also used in literature: a diamond and a circle. Adding these shapes is expected to increase the amount of contrast that is generated. This set will help answer research question 3: 'How can and to what extent can surface modifications play a role in the design guidelines?'. Another property that is also varied is the carving depth, this is because the surface modification is expected to increase the contrast as is the carving depth. By reducing the carving depth this correlation is expected to be shown more clearly.

Table 3.7: Properties of sample set 5, varying the surface texture, size and carving depth.

Name	Size [mm]	Carving depth [mm]	SC [°] x	ST	ST depth [mm]
Base	20	2	-	-	-
C2/.1	20	2	-	circle	0.1
D2/.1	20	2	-	diamond	0.1
C2/.2	20	2	-	circle	0.2
D2/.2	20	2	-	diamond	0.2
C.8/.2	20	0.8	-	circle	0.1
D.8/.2	20	0.8	-	diamond	0.1
C.8/.2	20	0.8	-	circle	0.2
D.8/.2	20	0.8	-	diamond	0.2

4 PROCESSING IMAGES

In this section, it is described how the images taken from the samples are processed. The difference between ink QR codes and 3D printed QR codes and the effect this has on how the code should be processed is discussed. Also, typical errors are described. Next to that, it is noted which steps of this process influence the results. Also, several recommendations are made on how the script can be improved. Post-processing and digitization of the images are both influential and important steps when determining the results of the experiments. Thus, the results are determined and thereby highly dependent on the post-processing script. To avoid ambiguity four words will be used consistently. When describing the QR code, the words 'QR code' or 'code' will be used. Since code is also a synonymous with script, the word 'script' will be used to refer to the script. The word 'module' is used to refer to the modules in the QR code, while 'pixel' is used to refer to pixels in images that are taken from the samples.

For this processing, a Matlab script is used. In Matlab, the image is treated as a matrix. Each matrix is size $n*m$, depending on the pixel size of the image. The matrices contain the value for a certain colour. To save an RGB colour image, 3 matrices are used. Each contains the respective R, G and B values.

As a starting point, a QR code reader script provided by MathWorks was used. This code was modified almost in all steps, and these modifications are described. Altering the script was necessary because of the differences between reading a QR code made from ink and from shadows. There are several differences when reading a QR code from shadows instead of from ink.

At first, there will be less contrast. Black ink is darker than shadows, so it is more clear where the negative regions are when reading a QR code from ink instead of from shadows. Also, the contrast will be non-uniform in colour. This has to do with light reflection. If light falls from one side of a wall, it will reflect. This causes the negative pixels next to the walls to appear lighter when light is cast onto the adjacent walls.

The second difference between 3D-printed and ink-printed codes is that the code will be less sharp. This has two causes. The achievable resolution of an SLS printer is larger than that of an inkjet printer. Because of the grain size, the lines will be a bit irregular. This results in less clear borders.

Thirdly, there will be more errors in the code. In Figure 4.1 the reference code (left), a binarized result (middle) and the errors in the binarized code (right) are shown. On the right, typical errors are pointed out. It can be seen that most errors are dispersed over the modules. There are several causes of the errors.

The first cause is the adjacency of negative modules. When multiple negative modules are adjacent this can have an effect on the generated contrast. If the length of the adjacency exceeds the shadow length, no shadow is cast on the negative modules, and as a result, the negative

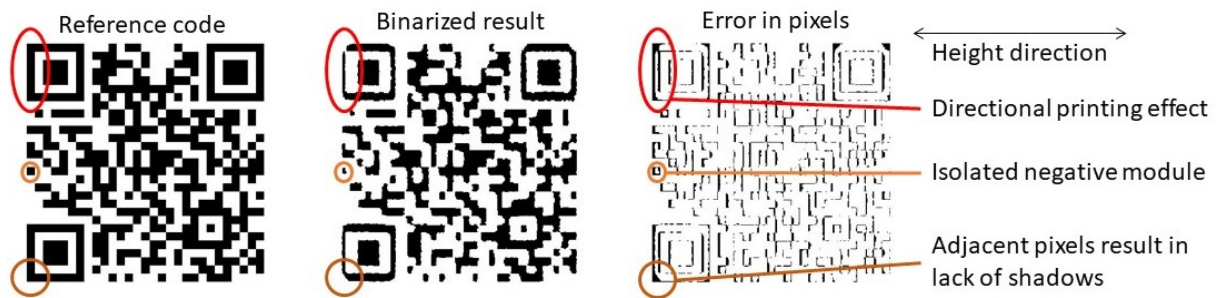


Figure 4.1: The reference code, transformed binarized results and errors that occur on a pixel level with three typical errors that arise in printed samples and the height direction of the printer pointed out.

modules are not darker than the positive modules. This effect is shown in Figure 4.1, and often results in errors in the corners of the detection patterns because there are 7 adjacent modules in two directions.

Secondly, light falls on the wall between the black and white module and reflects on that wall. This creates an overlit area from one lighting angle. In the combined image this results in a higher grayscale value, increasing the chance of an error. This light reflection effect causes errors on the edges of modules.

Another effect that causes errors is that there are some directional differences in the printer. The samples were oriented with the QR code in the horizontal direction. The laser can cause layers below the powder layer that is to melt to also fuse as powder liquefies and leaks down in that layer. This causes material to be present where it should not be, and because of this, the positive areas are larger. This occurs in the height direction due to gravity. This error is also shown in Figure 4.1, where it can be seen that the black modules in the height direction are smaller.

One more category of errors that occurs in isolated negative modules, black modules that have white modules all around them. An isolated negative module is pointed out in Figure 4.1. These modules tend to give more errors, especially around the borders. There are two reasons for this. At first, there is more material present in these modules than there should be. The isolated negative modules tend to get clogged more quickly. Since there is no adjacent open space, it is harder to remove semi-loose material. This causes more powder to stick in these isolated modules compared to non-isolated modules. Part growth can also be a cause since there are 4 walls that can grow. Next to that, the isolated modules have more light reflection, as these isolated negative modules will have the maximum number of walls around them (4).

Based on these differences and typical errors some modifications should be made to the script. In order to be able to process a QR code in which contrast is generated by shadows, a more robust binarization step is needed. Also, the code will be a bit more irregular and on the boundaries, some errors are likely to occur.

The rest of the chapter is structured as follows. At first, the steps that are carried out in the script are described. Next, it is explained how to calculate the scoring values that are used to evaluate the samples and why they are valid is explained.

4.1 Processing steps

Using the setup several images are made. These images are processed in several steps. The following actions take place during the processing. At first, the taken images are imported. Next, the image turned into grayscale and binarized. Afterwards, the detection patterns of the QR code are located. Then the image is transformed so that only the code remains in the image, and the code is rotated to a certain position. At this point, the code can be decoded. The goal of the experiments is not to decode the code, but to evaluate how well the code is embedded. To do this, a score is calculated to reflect how well the code is embedded. This is the final step of the script. How this score is calculated and why it is valid will be discussed in paragraph 4.2.

Importing the image

The first step is importing the image. Four images with four different lighting directions are imported and combined into a single image. After this step, a combined image is obtained, and this image is further processed. The image has the form of a 3D matrix. Each value represents a pixel in the image. Three matrices containing the RGB colour values. The combining of the images is done by adding up the four matrices and dividing by four. The colour values are a number from 0 to 255 representing the RGB values of the corresponding pixel. In Figure 4.2 two images with directional lighting are shown on the left. The combined image resulting from the four directional lighting directions is shown on the middle right and on the right the resulting image when all lights are on is displayed. The contrast is less when all lights are on at the same time. Four different angles are used because it is expected to increase the contrast. In this step, extra actions could be added to the script to take into account the lighting directions and effects of curvature. Initially, this is not done, but it is recognized as a possibility.

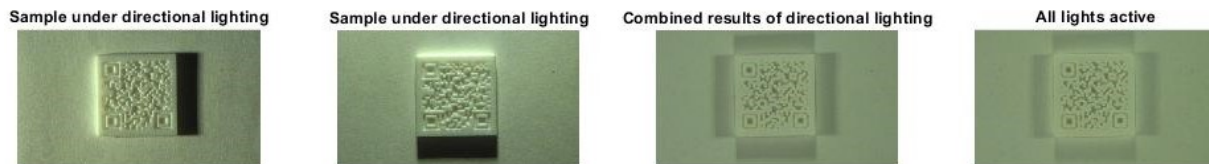


Figure 4.2: Two directional lighting images, the combined result and the result with all lights on.

Grayscale and Binarization

Next, the image is transformed into grayscale. To do this, the degree of Redness, Greenness and Blueness (RGB) is considered and converted into grayscale. Equation 4.1 is used to convert an RGB image into a grayscale image in the Matlab function `im2gray` [50]. A high value means the pixel is more white while lower values are darker.

$$Grayscale = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (4.1)$$

After that, the image is binarized. In the binarization step, Otsu's method is used. Otsu's method is a method of image thresholding. In image thresholding it is determined which pixels are white and which are black based on their grayscale. A threshold is set, and pixels with a higher value (lighter) are categorized as white, while pixels with a lower value a categorized as black. Otsu's method determines the grayscale value at which the variance within the different groups is minimal. The threshold is computed so that the weighted variance in the two groups is minimized. Between the groups, the weighted variance is maximized. Minimizing the variance yields a threshold value to binarize the image. Pixels with a higher grayscale value are categorized as white and pixels with a lower value as black. This is done for each image. A histogram of the black and white pixels that should be identified according to the reference code and the pixels

that are identified using Otsu's method is shown in Figure 4.3.

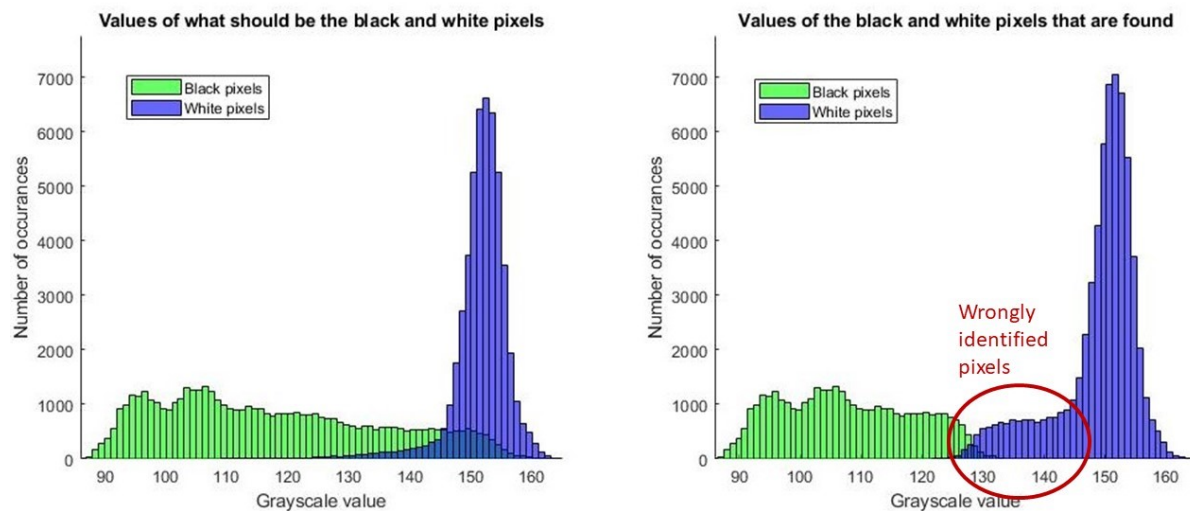


Figure 4.3: A histogram of pixels that are supposed to be black and white next to the pixels that are identified using Otsu's method.

In Figure 4.3 the histogram on the left shows what the grayscale values are of the pixels in black and white modules. This histogram is constructed by comparing the transformed grayscale image to the reference code. In that way, it is determined what pixels correspond to black and white modules. As can be seen in this image, the white pixels are grouped in a grayscale value of around 153 and seem to follow a normal distribution. The black pixels seem to have a bathtub distribution ranging from 90 to 155. The shape of this distribution differs from sample to sample and is not consistent in measurements of the same sample. When selecting a threshold value, these two groups of pixels should be separated as best as possible. Looking at the figure on the left, it can be concluded that it would be best if the threshold value is around 145. In that way, most white pixels will be identified as white and most black pixels as black. In the image on the right of Figure 4.3, it can be seen that using Otsu's method the threshold value is determined to be around 130. This is not ideal since there are a significant number of pixels with a grayscale value between 130 and 145. As a result, a lot of pixels that are supposed to be identified as black are identified as white instead. In Figure 4.3 this can be seen as the blue bar region is larger than it should be. During the binarization, it was a reoccurring problem that too many pixels (and modules) that are supposed to be black were seen as white. Several attempts are done to improve this. Exploration was done to find a different, more suitable, binarization algorithm.

When determining the cut-off value of the binarization step it would be better if instead of minimizing the weighted in-group variance, two distributions are assumed to be present and the break-off point between these distributions is estimated. This would be a better representation of the black-and-white modules. A lot of work has been done on the identification of two groups in image detection. Shirazi et al. [51] combine colour channels with a k-means algorithm (another way of image thresholding) and achieve better results than Otsu's method.

When looking into what makes the image binarization challenging, it turned out that the pixels around the sample have a strong disturbing effect. This is because that region is partly shaded, and this adds a lot of grayscale values that are in the same grayscale range as the black modules. When a code is embedded this is not the case, as there are no shadows around the code. To circumvent this the code could be first isolated and afterwards a second binarization step

could be applied.

It can be concluded that the binarization steps should be improved if the technology is to be used in practice. For practical reasons, a modification is made for the threshold value that results from Otsu's method. As the found value was systematically too low, it is modified to be increased with a factor of 1.08. This value has been empirically determined. A histogram of pixels that are supposed to be black and white and a histogram of pixels that are identified as black or white after multiplying Otsu's threshold value with 1.08 is shown in Figure 4.4. By multiplying with a factor of 1.08 much better results are achieved. As was seen in Figure 4.3, the threshold value that is ideal is around 145 while the value that is found using Otsu's method is 130. Multiplying this found factor results in a value closer to the ideal value. This can be seen in Figure 4.4, as the blue and green regions are similar in the image on the left and right. This means that most pixels are correctly identified as black or white. For comparison, the image on the right in Figure 4.3 does not apply a factor, note that the green region (black pixels) on the right is much smaller than the green region on the left, what it should be.

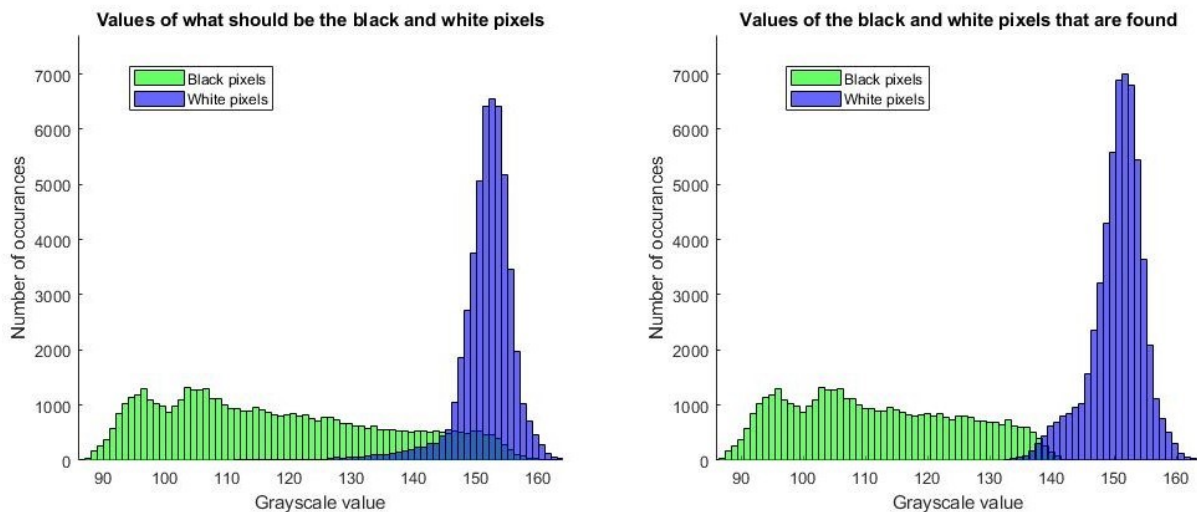


Figure 4.4: A histogram of pixels that are supposed to be black and white next to the pixels that are identified by multiplying Otsu's threshold value with a factor 1.08.

Before the binarization step, a Gaussian blur can be applied to achieve a more homogeneous result by removing small outliers. However, this does seem to result in less accuracy. This is because if the image is modified on pixel level it causes a disturbance in the results since the contrast is one of the key aspects that is being investigated. Changing the pixels lowers the accuracy of the results as the image is being changed from the core image, and thus the results become less representative of what is actually going on. Thereby the Gaussian blur affects the results if applied too broadly.

After these steps, a binarized image is represented as a matrix with ones (white) and zeros (black). The original, grayscale and binarized images are shown in Figure 4.5. The original image is a 3D matrix containing RGB colour information, the grayscale image is a 2D matrix containing grayscale values ranging from 0 to 256 and the grayscale image contains zeros for pixels that are black and ones for pixels that are white.

Code detection

Next, the image is analyzed to find the code. This is done by looking for the detection patterns in the binarized image. The detection patterns have a typical pattern. The ratio of pixels is always

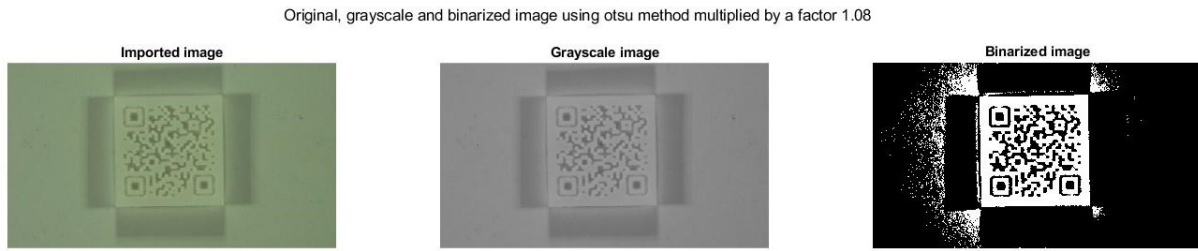


Figure 4.5: On the left the original image is shown, in the middle the grayscale image is shown and on the right, the binarized image using Otsu's threshold value times 1.08 is shown.

1:1:3:1:1, and this property is used to find the patterns. Several steps are carried out to find these detection patterns. At first, the image is inverted so that the detection patterns become positive instead of negative values. Next, the boundaries are removed. This isolates the code. Thereafter, the centre points of the three detection patterns need to be found. This step has been modified.

Originally the Euler characteristic was used to find the centre point of the detection patterns. The Euler characteristic corresponds to the number of objects in the region minus the number of holes in those objects, the formula is shown in Equation 4.2. χ is the Euler characteristic and H stands for holes.

$$\chi = 1 - H \tag{4.2}$$

In the case of a QR code, a detection pattern will contain one object within it with one hole in that object and thus have an Euler characteristic value of 0. By filtering the image to only contain areas with this value several areas are identified. Why this is problematic is illustrated in Figure 4.6, in which a normal detection pattern, a resulting detection pattern from a sample after processing and the calculation of the Euler characteristic is shown. As can be seen, the outer layer of the detection pattern will sometimes not flag as an object with an Euler characteristic of 0. Once the centroids of the objects with an Euler characteristic 1 are found, the ratios of the areas around this centroid are checked to be 1:1:3:1:1. If this is the case, the corresponding centroid of the space is flagged as the location of a detection pattern. This 1:1:3:1:1 ratio is also shown in Figure 4.6, and it is also clear from the image that the resulting detection pattern does not have this ratio. Filtering by Euler characteristic does not work when processing QR codes from images. Problems arise when the squares are broken or contain more than one hole. This causes this Euler characteristic to not be 0, and detection patterns are not found. To solve this, the function was modified. Originally the area around the square was checked for the 1:1:3:1:1 ratio. Since this ratio is not exactly achieved in the samples, this verification step also failed.

In the new method, the first step that is added is to fill small holes of up to four pixels. Next, the image is filtered for the value of the Euler characteristic to be 1, leaving only the objects without holes in them. The three largest square-like objects are selected by looking for areas that have an increasing number of positive values in a square. The detection patterns tend to be the largest squares. Using this method, they are detected successfully in the samples. The centroids of the detection patterns are calculated, and this is used as the position of the detection patterns. Next, it is checked how the position between these detection patterns relates and if this is as would be expected. Based on the position of the patterns it is determined which pattern is which. The distance between the patterns is also used to determine the module size in pixels. Outputs are the centroids of the detection patterns, a slightly filtered image that is used mainly for visualization, and the module size in pixels. An image in which the detection patterns are detected is shown in Figure 4.7. It should be noted that the new method is much

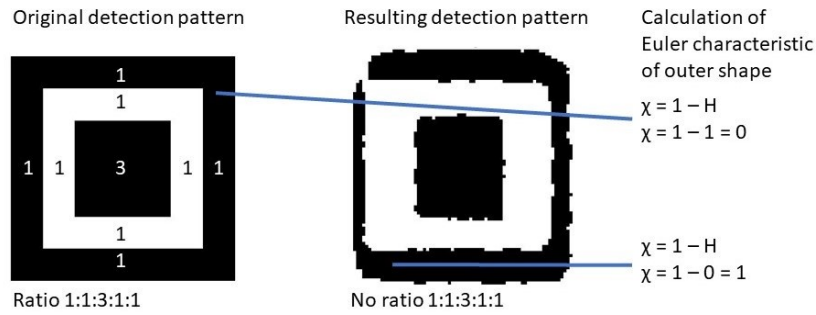


Figure 4.6: Normal detection pattern (left), Binarized results of a detection pattern from 3D printed sample (middle) and the calculation of the Euler number (right).

less robust than the original method, and may yield false results. If a high error percentage is found, it should be checked if the detection patterns are correctly identified.

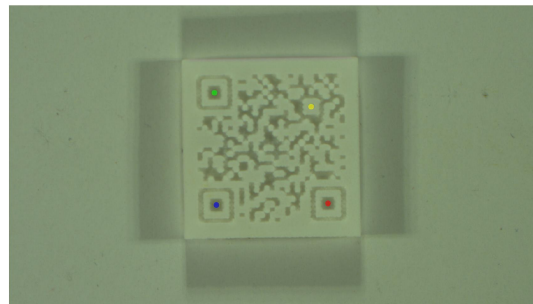


Figure 4.7: The colour image with marked detection patterns.

Transformation

Using the module size and the three positions of the three large detection patterns, the alignment pattern is located. These four points are used to come up with a geometric transformation function to transform the original image. In Figure 4.8 an example of how this transformation function transforms the image is shown with an original (left), transformed and rotated (middle) and resulting (right) image state. In the resulting image, the code is supposed to be centralized and all the modules should be horizontally or vertically adjacent to one another. The image/matrix is transformed and rotated so that the QR is in a normal position (unrotated/straight and the small detection pattern is in the bottom right corner). Now that the image is in this configuration, a grid is created to select pixels for each module of the QR code, isolating the QR code. The original, transformed, rotated and isolated code are shown in Figure 4.9.

Pixel selection

The QR code is isolated through pixel selection. This pixel selection is needed to isolate the QR code, so that histograms can be made and the differences between the groups (black and white) can be calculated. Pixel selection happens by first determining the centre point of each module based on the module size (height and width), and the location of the detection patterns. The module size is determined by looking at the distance between the detection patterns. Since the code is placed in an unrotated manner the centre points are placed in a straight line in a single direction. To extract values from the rotated image, two vectors with integers are created. These integers correspond to the location of the pixels in the image. The formula to construct these vectors is shown in Equations 4.3 and 4.4.

$$\mathbf{x} = \text{round}(\mathbf{g} \cdot m\mathbf{s}\mathbf{x} + \mathbf{x}_{dUL} - \mathbf{s}\mathbf{m}\mathbf{x} \cdot 3) \quad (4.3)$$

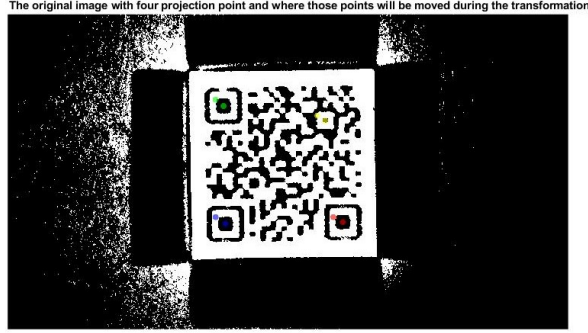


Figure 4.8: A binarized image with the original and transformed location of the detection patterns marked.

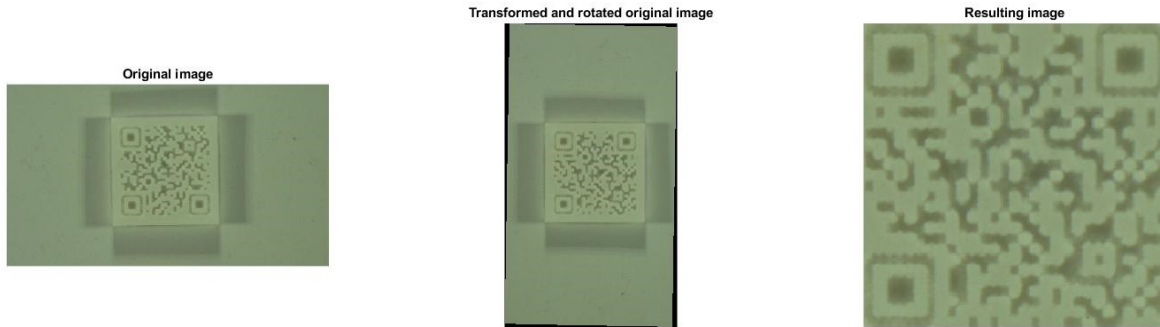


Figure 4.9: Original(left), transformed and rotated(middle) and resulting(right) image.

$$\mathbf{y} = \text{round}(\mathbf{g} \cdot \text{msy} + \mathbf{y}_{dUL} - \mathbf{sm} \cdot 3) \quad (4.4)$$

$$\mathbf{g} = [1, 2, \dots, sQR - 1, sQR]$$

In equation 4.3 and 4.4, \mathbf{x} and \mathbf{y} are the vectors of the x and y coordinates. \mathbf{g} represents the grid that is used for the pixel selection. The variable sQR is the number of modules in the QR code. The average number of pixels per module in x and y is denoted by the terms msx and msy (module size x/y). \mathbf{x}_{dUL} and \mathbf{y}_{dUL} are vectors of size sQR (same as \mathbf{g}) with a constant value of x_{dUL} or y_{dUL} . These are the x and y coordinates of the centre point of the upper left detection pattern. The starting point of the vector, the location of the upper left module, is calculated by subtracting three times the module size from the location of the detection pattern in the upper left corner. \mathbf{smx} is a vector of the size sQR with msx as its value for each point. Likewise, $\mathbf{sm} \mathbf{y}$ is a vector of the size sQR with a value of msy. Since the centre of the detection pattern is the fourth module when looking from top to bottom and from left to right, this gives the centre of the first module. This centre serves as a reference point to where in the image the modules are located. As a final step, the vectors are rounded to the nearest integer so that pixels can be selected using the vectors as locations from the transformed rotated image. How this created grid selects points is shown in Figure 4.10, where individual values of the \mathbf{x} and \mathbf{y} vectors are shown. Also, the supposed location of modules and selected points are shown.

As can be seen in the figure, the \mathbf{x} and \mathbf{y} vectors act as a grid that extracts specific pixels from the transformed and rotated image and assigns that pixel to a position in the QR code.

It is also possible to use extra points in each module (ep). If multiple points per module are used, the step size is modified and another starting point should be used. In equations 4.5 and 4.6 the new formulas to calculate the pixel selection vectors are shown.

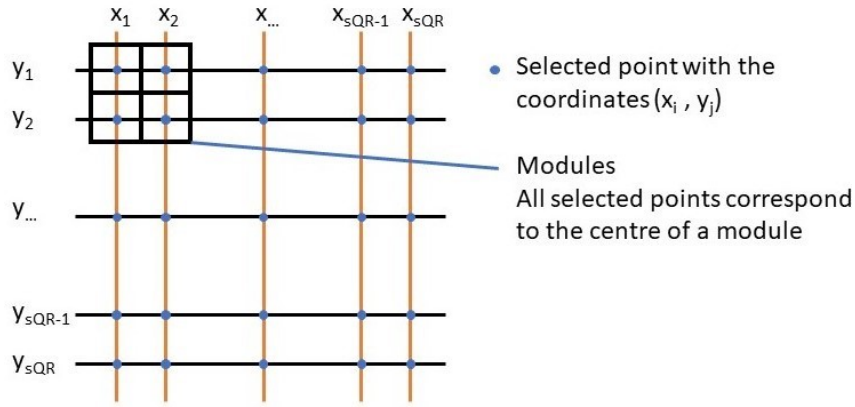


Figure 4.10: Pixel selection grid using a single point per module.

$$\mathbf{x}_{extra} = \text{round}\left(\mathbf{g}_{extra} \cdot \frac{msx}{ep} + \mathbf{x}_{dUL} - \mathbf{smx} \cdot \left(3 + \frac{ep - 1}{2 \cdot ep}\right)\right) \quad (4.5)$$

$$\mathbf{y}_{extra} = \text{round}\left(\mathbf{g}_{extra} \cdot \frac{msy}{ep} + \mathbf{y}_{dUL} - \mathbf{smy} \cdot \left(3 + \frac{ep - 1}{2 \cdot ep}\right)\right) \quad (4.6)$$

$$\mathbf{g}_{extra} = [1, 2, \dots, (sQR \cdot ep) - 1, (sQR \cdot ep)]$$

In equations 4.5 and 4.6 \mathbf{x}_{extra} and \mathbf{y}_{extra} are the vectors that are used for the pixel selection that contains multiple pixels per module. A new variable is introduced, ep . To determine the size of the pixel selection grid \mathbf{g}_{extra} , sQR is multiplied by ep to find the new number of points that are selected. All other vectors have the same size as \mathbf{g}_{extra} . The ep variable represents the number of pixels that will be selected in each module in the x and y direction. Note that per module the ep will be added in both height and width, so the actual number of points per module is ep^2 . The value of ep must be uneven. Otherwise, there are points on the boundaries of modules. These can't be assigned to a module and are extra unreliable. The new base vector becomes \mathbf{g}_{extra} and this vector is multiplied by module size divided by the number of points per module as this is the new distance between points. A new starting point is calculated based on how many module sizes the upper left starting point is from the centroid of the upper left detection pattern, x_{dUL} and y_{dUL} . To calculate this value half minus one of the ep is divided by the ep to find how many modules there should be added to 3 to find the top left pixel (right term in Equation 4.5 and 4.6). By rounding and entering these vectors as input for the rows and columns of the transformed and rotated image, pixels are selected. This pixel selection using multiple points per module is illustrated in Figure 4.11.

The selected pixels form a new image, containing only the QR code. In this image, it is known which pixels are supposed to correspond to what module, and thus a comparison to the reference code can be made. The resulting images are shown in Figure 4.12.

The way that this grid is created has a large influence on the results. What pixels are selected directly determines the result, and thus this is a very important step. In case the image is distorted it could be detected here and the grid could compensate for that distortion. This should happen by modifying the module size in either the x or y direction. Distortion detection and compensation become increasingly relevant when the code is present on curved surfaces.

If the extended grid is used, each module contains multiple pixels. The resulting values of these pixels will have to be converted back into a single value for the module. This is done by looking

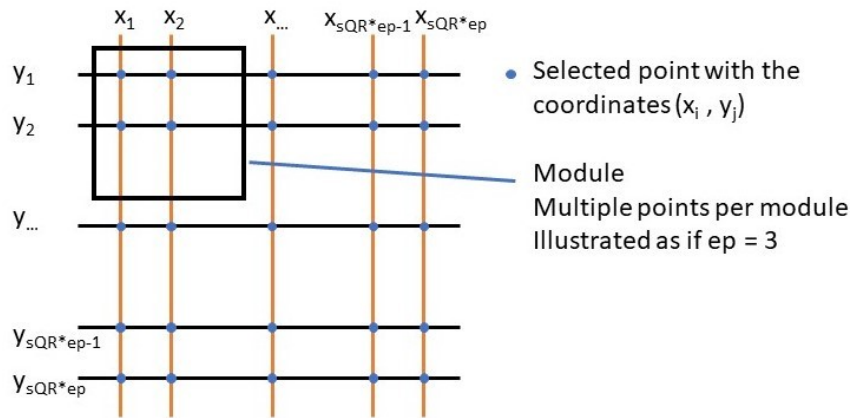


Figure 4.11: Pixel selection grid using multiple points per module.

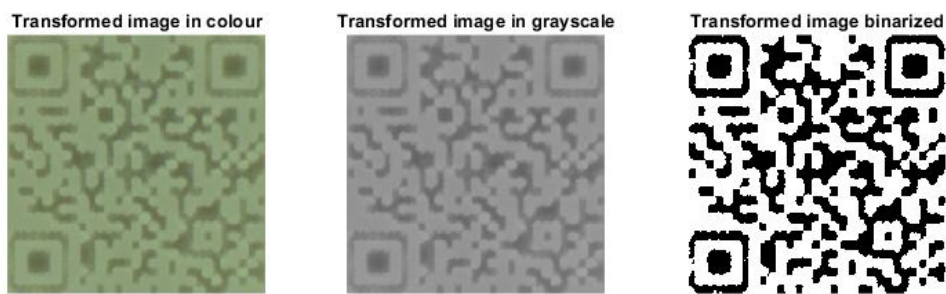


Figure 4.12: The images after transformation and pixel selection in colour(left), grayscale(middle) and binarized(right).

at all the pixels that fall into that respective module. To determine if the module is black or white the number of black and white pixels is counted. Since pixels are systematically less likely to be black due to problems with binarization, modules are registered as black if 30% or more of the pixels in each module are black. The average grayscale value of all pixels is used as the grayscale value of the module.

If the code is embedded onto products and should be read, this is where the decoding process would stop. The code is found and can now be read. However, for the purpose of determining how well the code is embedded onto the product, it will be necessary to compare the found code to the reference code. This comparison and how to come up with a score is discussed in paragraph 4.2

Advanced pixel selection

It was investigated if alternative pixels selection does make a difference. Errors occur around the rims of the modules because that is where printing accuracy influences the results. Thus, it would make sense to only use the inner pixels. To test this, the performance of a 11x11 grid per module is compared to the results where only the inner 7x7 modules are used. The hypothesis is that the inner pixels of a module will result in more contrast, and thus a clearer code than using all the pixels. This was tested. In Figure 4.13 the resulting images with all modules and only the inner modules are shown. The results of the inner modules seem to be sharper around the edges. In this case, both result in 3 errors. Based on the binary image (second column) in Figure 4.13 it seems that the inner module selection method is to be preferred. However, this is very circumstantial. Looking at the histograms of grayscale values will give a better impression

of whether the inner module method is to be preferred.

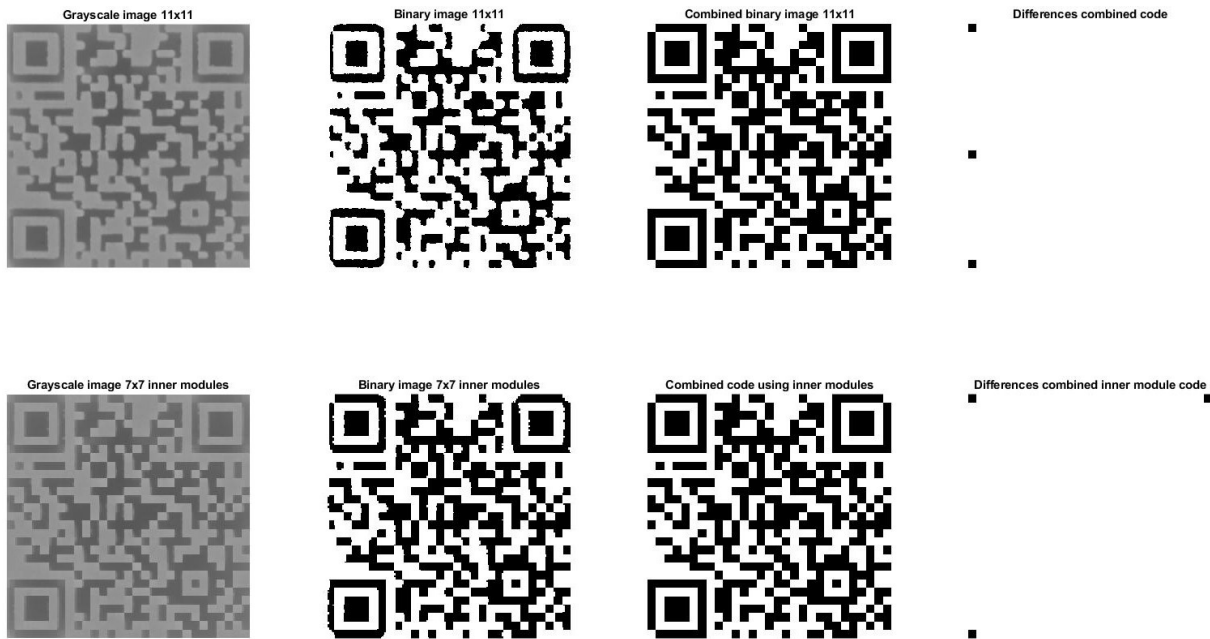


Figure 4.13: A visual comparison between using all pixels(11x11) in a module (top row) and only the inner pixels(7x7) (bottom row).

To be able to extract more information from the in Figure 4.13 a histogram of the average module grayscale values is shown in Figure 4.14 in which all pixels assigned to one module are used (left) and when only using the inner pixels (right). In the histogram the concentration of the

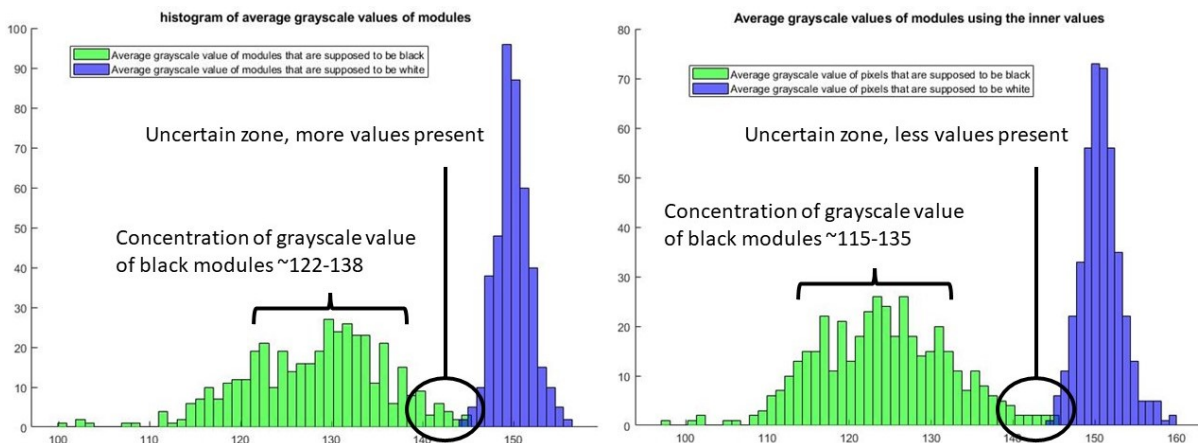


Figure 4.14: A histogram depicting the average grayscale values of the modules using all pixels(11x11) and only the inner pixels(7x7).

grayscale values of the black and white modules are shown. The mean and the deviation are important. If the means are further apart with a low deviation, it is easier to recognise modules as black or white. Grayscale values that overlap are likely to give errors, so the two aspects to look out for when comparing the histograms in Figure 4.14 are the overlap and the main concentration of values. In both histograms, there are some overlapping values, although fewer values are present in the uncertain zone when only looking at the inner pixels. That there is overlap in both histograms is probably why both methods give the same number of errors. However, there is a shift visible in the concentration of the grayscale values of the black modules. When

using only the inner modules, the grayscale values of the black modules decrease. Where on the left (using all modules) the concentration of black modules is in the range of 122-138, on the right the concentration is around 115-135, clearly lower. This is a good development, as it means that it becomes easier to identify modules. Thus, it could be argued that using the inner modules is slightly better. In this case, it does results in a higher sigma (2.25 for 11x11 vs 2.52 for 7x7). It can thus be concluded that using inner modules does improve the results in this case. It makes sense to include the number of standard deviations between the inner modules' average grayscale value in the exported results so that this can be checked for more samples. However, due to a programming flaw that was discovered later this is not done for most samples. This could be done if more time was available. It is also questionable how much extra insight this would give into how successful the code is embedded. It can be concluded that by using advanced pixel selection the readability of the embedded code increases because the selected black pixels are darker.

Second binarization with isolated code

In the binarization step, it was observed that the noise surrounding the code made binarization more difficult. It is tested if binarization yields better results when only the code remains. The transformed image has been binarized again. Otsu's method yields a worse result. Using kmeans, an alternative thresholding method, failed initially due to the surrounding pixels distorting the two groups that it looks for. Kmeans attempts to group the found values (grayscale) into a number of groups, in this case 2. Kmeans also performs worse as it groups all the pixels in one group.

4.2 Determining scoring values

In this section, the considerations and calculations when determining how well the code is embedded are explained. The generated contrast and accuracy are quantified. When determining the objectives of the embedment of the code 'readability' was listed as an objective. The first subgoal was generating contrast. Next to that, generating that contrast at the right location is important. This is called the resolution. A method is proposed to come up with a score, quantifying how well these two goals are being reached. The scores should reflect how well these objectives are being achieved, and it should be clear why these scores represent the goals.

To be able to generate a score, first, some post-processing is needed. The first post-processing step is to enable the comparison of the codes. This is done by creating a reference code that corresponds to the reference grid. Because the scaling function distorted the results a new scaling function was written to do this. The two resulting matrices are compared, identifying errors. Next to that, the grayscale values of the pixels that are supposed to be positive (white) and negative (black) pixels are also assigned a value and grouped. Also, the grayscale values of the pixels that are identified as positive and negative are found.

In Figure 4.15 the reference code, the binary pixelized code, the transformed grayscale image, and the difference between the reference code and pixelized code are shown. It is clear in what modules a lot of values seem to be off, and where the contrast is generated. However, the problem of how and what to quantify is not solved yet. The data needs to be transformed into information. An approach is proposed on how to score the performance of the different samples. This approach is explained in the rest of this paragraph.

Quantifying contrast

When looking at this scoring problem it is clear that there are two distinct groups that need to

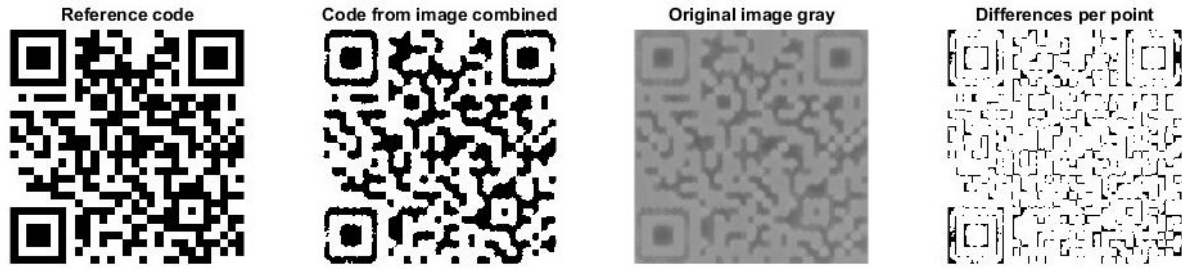


Figure 4.15: From left to right: the reference code, the code pixelized code binarized, the pixelized code in grayscale and the difference between binarized value compared to the reference code.

be identified, namely the black and white modules. How well the sample and circumstances permit this identification is what has to be quantified. Identifying two groups is a problem that can be analysed through statistics.

For the first objective, generating contrast, it seems evident that the colour/grayscale difference between the black and white pixels should somehow be expressed. The difference in grayscale values between black and white pixels will have to be measured and a value needs to be assigned. There are several options to do this. The most straightforward option is to compare the mean value of the black and white pixels. The higher the difference, the better the contrast is. However, this runs into problems because it does not take into account the distribution of the grayscale values. There can still be problematic outliers. Also, it does not give information on the certainty of placing pixels in the correct group. To solve this the two groups can be seen as groups with a normal distribution of which the mean μ and standard deviation σ can be calculated from the data set. The grayscale values are used as the values in the data set. The goal is to find where the two bell curves intersect and calculate how many standard deviations this is removed from the mean. This expresses the likeliness that one value is correctly identified in case the binarization threshold value that would be ideal is found. It is checked if the data is normally distributed. For the grayscale values of pixels, this is not the case, but the mean grayscale values of modules are normally distributed.

At the start of the derivation, the means (μ_b and μ_w) of the black and white modules as well as the standard deviations (σ_b and σ_w) are known. The starting point for the derivation is the point of intersection of the two bell curves. It is known that the fraction of the black modules with a higher grayscale value than the value at which the intersection takes place is equal to the fraction of white modules with a lower grayscale value. Using the property of symmetry of normal distribution, this gives that $z_b = -z_w$. Also, since the means and standard deviations are known z can be calculated. In formula 4.7 these relations are shown. x is the grayscale value at which the two bell curves intersect. There are three unknowns (x, z_b, z_w) and 3 equations, so the system can be solved.

$$\begin{aligned}
 z_b &= -z_w \\
 z_b &= \frac{x - \mu_b}{\sigma_b} \\
 -z_w &= \frac{\mu_w - x}{\sigma_w}
 \end{aligned}
 \tag{4.7}$$

The goal is to obtain an expression for the number of standard deviations between the means of the black and white modules. To derive this, in formula 4.8 z_b and z_w are substituted. Using the

steps shown in the equations 4.9, the formula is rewritten to find an expression for x in equation 4.10.

$$\frac{x - \mu_b}{\sigma_b} = \frac{\mu_w - x}{\sigma_w} \quad (4.8)$$

$$x \cdot \sigma_w - \mu_b \cdot \sigma_w = \mu_w \cdot \sigma_b - x \cdot \sigma_b$$

$$x \cdot \sigma_w + x \cdot \sigma_b = \sigma_b \cdot \mu_w + \sigma_w \cdot \mu_b \quad (4.9)$$

$$x \cdot (\sigma_w + \sigma_b) = \sigma_b \cdot \mu_w + \sigma_w \cdot \mu_b$$

$$x = \frac{\sigma_b \cdot \mu_w + \sigma_w \cdot \mu_b}{\sigma_w + \sigma_b} \quad (4.10)$$

The next step is to substitute this expression of x in either the expression to calculate z_w or z_b . In equation 4.11, x is substituted in the formula for z_b . Rewriting results in equation 4.12, which calculates the number of standard deviations between the means of the average grayscale values of the black and white modules. In equation 4.13 only the resulting equation is shown.

$$z_b = \frac{x - \mu_b}{\sigma_b} = \frac{\sigma_b \cdot \mu_w + \sigma_w \cdot \mu_b}{\sigma_b \cdot (\sigma_w + \sigma_b)} - \frac{\mu_b}{\sigma_b} \quad (4.11)$$

$$z_b = \frac{\sigma_b \cdot \mu_w + \sigma_w \cdot \mu_b}{\sigma_b \cdot (\sigma_w + \sigma_b)} - \frac{\mu_b \cdot (\sigma_w + \sigma_b)}{\sigma_b \cdot (\sigma_w + \sigma_b)} = \frac{\mu_w \cdot \sigma_b - \mu_b \cdot \sigma_b}{\sigma_b \cdot (\sigma_w + \sigma_b)} = \frac{\mu_w - \mu_b}{\sigma_w + \sigma_b} \quad (4.12)$$

$$z_b = z_w = \frac{\mu_w - \mu_b}{\sigma_w + \sigma_b} = \sigma_{betweenmodules} \quad (4.13)$$

The resulting number is the number of sigmas (standard deviations) between the means of the average grayscale values of the black and white modules. From now on, to shortly describe this value the name 'sigma between modules' or $\sigma_{betweenmodules}$ will be used. This is the main scoring indication. The number of 'sigma between pixels' can also be calculated. To do so, the standard deviation of individual pixels is used instead of the combined standard deviation of the modules. This result is unreliable as the grayscale values of the pixels don't follow a normal distribution.

Using sigma between modules the chance of black modules being identified as white and white modules being identified as black can be found. Since a pixel can both be misidentified as black and white, the ideal rate of errors is twice as high as the value associated with the sigma between modules. This is because both black and white modules are equally likely to pass the intersection value. In reality, the error will be larger as the cut-off point is not estimated correctly.

A disadvantage of using this sigma instead of the flat difference between the means is that the absolute difference in grayscale values is lost out of sight. A way to resolve this would be to include this absolute difference as a boundary condition that should have a certain minimum.

When evaluating how well contrast is generated it should be considered whether the results should be evaluated on a pixel level or a module level. Since the grayscale values of the pixels don't follow a normal distribution the sigma between pixels is unreliable. Mainly because of this reason the sigma between modules is used. However, there are some more considerations to be made. Since one module contains multiple pixels and only one value should be selected per module, it could be argued that looking at a module level is more sensible. On the other hand, the pixel level gives insight into what happens inside the module as well as where the errors occur. The main difference between looking at a module level and a pixel level when

calculating the mean and standard deviation is that the standard deviation will be significantly lower when looking at a module level. This is because the combining step filters out the outliers. As a result, the sigma between modules is always higher than the sigma between pixels. On a module level, outliers are not problematic because these values will be compensated by the rest of the module.

The amount pixels that are used can also be varied. The effects of varying the amount and location of pixels that are used in each module will be discussed later.

The scoring value that is chosen is the sigma on the module level because it quantifies the chance that modules will be wrongly identified. Next to that, the difference between the means of the black and white pixels should also be used to see if there is sufficient difference between the two.

Quantifying accuracy / sufficient resolution

The second objective is to identify if the generated contrast is generated in the right location. This is about checking if modules and pixels are correctly identified as respectively black or white. In this problem, two groups need to be identified, and the test/method to do so needs to be evaluated. A statistical approach that is commonly used for similar problems, is to calculate the mathematical accuracy of the respective test. The mathematical accuracy of a test can be described using sensitivity and specificity. Additionally, the positive predictive and negative predictive values are calculated to give more information. For these tests, the white modules are regarded as the positive group and the black modules as the negative group.

In order to understand what these values say here is a short recap of what they mean. Sensitivity is the chance that a test is positive in case the subject is truly positive. In this case, that means how many of the pixels that are supposed to be white are also identified as being white. Likewise, the specificity expresses how much of the true negative values (reference black pixels) are found (identified as black pixels). So, the sensitivity and specificity give information on the likeliness that pixels are correctly identified as their true values. The positive predictive value (PPV) is the fraction of the identified positive values that are true positive values, so that are correctly identified. Likewise, the negative predictive value (NPV) is the fraction of identified negatives that are true negatives. So, the PPV and NPV give information on the likeliness that pixels that are identified as positive or negative are supposed to have that value.

Together, these four values give two types of information that is important. At first, all four values together indicate the likeliness of correctly identifying black and white modules. It shows which groups are more likely to be identified and which might be more problematic. The sensitivity and specificity indicate how well the reference is translated to the image. The PPV and NPV indicate how reliable the values in the image are. The values are interdependent, if black values are less likely to be identified that affects both the specificity, since fewer values that are supposed to be black actually are, and the PPV, because there are more false whites. Because of this interconnectedness, the interpretation can be done with only two values. The values that will be used are sensitivity and specificity because these values directly indicate the likeliness of a module being represented correctly. Thus, the results are more instinctive to interpret. One aspect that normally matters a lot is the occurrence/prevalence of positive and negative values. If only 10% of the group is expected to be positive, then that affects the interpretation. In this case, it is about 50/50 between the white and black modules. Because of this, it is defensible (although debatable) to sum the sensitivity and specificity to give a quick indication of what sample performs best. So, specificity and sensitivity will be used to quantify the accuracy.

Where the number of sigma between the means gives the ideal error value, the sensitivity and specificity give the success rate of the two different values when using the threshold value for binarization that is determined earlier in the process. This should thus be seen as a more realistic error fraction, one that does take into account imperfections in the binarization step. Together, these values also indicate how much this binarization step could be improved, thereby saying something about how well the binarization algorithm performs.

The position of errors on a pixel level plays a major role in how well the code is read on a module level. When translating the pixel level to the module level the distribution of the errors matters, because the more errors per module there are the more likely that that module gives the wrong value. If the errors are distributed randomly across all modules, then there might be a certain error percentage, but all modules will be identified correctly. To give insight into this distribution of errors, the pixel errors per module are calculated. The percentage of wrong pixels per module is calculated for each module. The errors per module are divided in two categories, namely those of black and white modules. For each of these categories, ten brackets are made with an error percentage of 0-10, 10-20 etcetera. The fraction of modules with an error percentage greater than 50% is calculated. Some of these modules with an error percentage below 70% can be saved with compensation. Because of this the percentage of modules with an error greater than 70% is also calculated. These values give insight in how often the errors on the pixel level cause problems on a module level.

It has been considered whether the two objectives (contrast and accuracy) can be represented by a single scoring value. It could be argued that the number of sigma between modules includes accuracy. The sigma between modules is influenced greatly if there are wrong pixels identified in either set. If there are pixels off in a module that does affect that average module grayscale value. Errors move the mean in the 'wrong' direction, increasing the standard deviation and resulting in a lower sigma. However, it does leave out some information as well. First and foremost, it does not give a good indication of how well the actual interpretation of the code goes. This is because only the ideal scenario is compared with the grayscale values and the actual binarization is left out of the results. As a result, the performance will be overestimated. Next to that, it is unclear whether the black or white modules cause the problems.

Scoring

The choice was made to use the sigma between modules as the main scoring indication. For context, the difference between the means of black and white modules and the sensitivity and specificity will be included. Another value that will be extracted for context is the problematic modules that have a pixel error percentage of 50% or more and 70% or more. The sigma between modules quantifies how much contrast is generated, and the other values give some information on how high the expected differences are, where the most errors occur and if there are problematic outliers.

5 SETUP

A test setup has been developed. The goal of this setup is to enable taking pictures under controlled conditions, eliminating the influencing factors and variance from the environment. Because the setup is expected to have a significant influence on the results it needs to be constant across the experiments. The most important condition that needs to be controlled is the lighting. To do this one optimal setup is determined, and this setup will remain fixed when processing the samples with design modifications. Several variations on the setup are tested. The setup is optimized for generating contrast, maximizing the sigma between the black and white modules.

At first, the basic setup and procedure are described. Thereafter, the experiment that will be used is described. The parameters that are fixed and varied will be discussed and results are shown. Based on these results an optimal setup is selected.

5.1 The components

Several components are used in the setup. The setup will consist of one overhead camera, four standards with lamps attached to them, a background and a lux meter. What is used for each of these components is specified in Table 5.1.

Table 5.1: The specific types of components that are used in the setup.

Component	Type
Overhead camera	IDS-UI-3590CP
Camera lens	IDS-10M11-C1616
Light stand	Laboratory stand
Lamp	GU10 TRADFRI LED 345 lm
Background	White paper
Lux meter	VOLTCRAFT MS-1300

A conceptual side view of the setup with a single light stand is shown in Figure 5.1. The light stand is represented by a black stripe with an orange lamp attached to it. The sample is placed in the middle, underneath the camera which is above it. To goal is to find the optimal values for the height, distance, angle θ and light intensity.

The setup consists of an overhead camera that is adjustable in height. It looks down on a white background where the samples will be placed. The height of the camera is set so that an area of 50x50 is visible. This height will remain fixed at 400 mm. An IDS-UI-3590CP camera with an IDS-10M11-C1616 lens is used. An image of this camera as used in the experimental setup is shown in Figure 5.2.

A part of the setup that does require some experimentation is the lighting. The idea is to have directional lighting from four directions, taking four images and combining them when post-processing the data. It is expected that this allows shadows to be cast more clearly than when

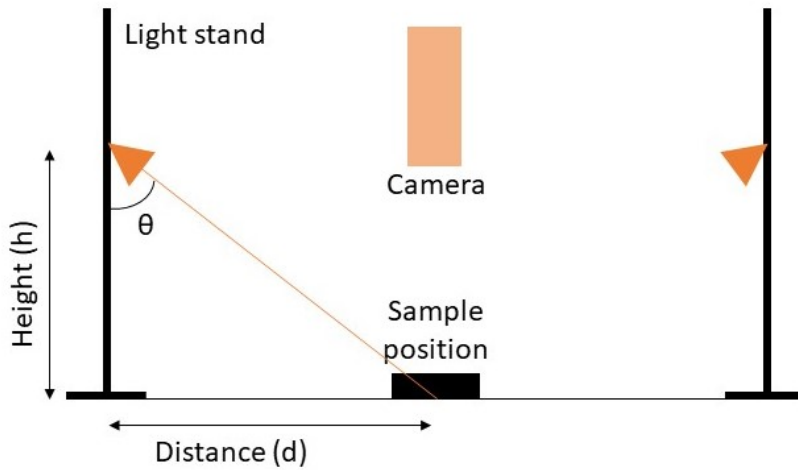


Figure 5.1: Conceptual side view of the setup showing two light stands, the camera and the sample position with the height, distance and angle θ indicated.

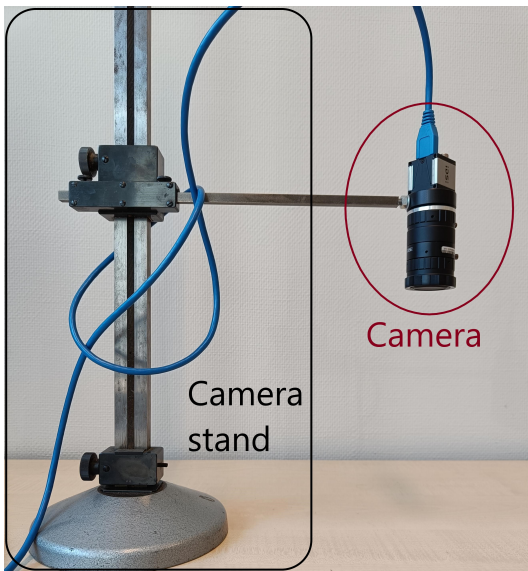


Figure 5.2: The IDS-UI-3590CP camera with an IDS-10M11-C1616 lens as used in the experimental setup.

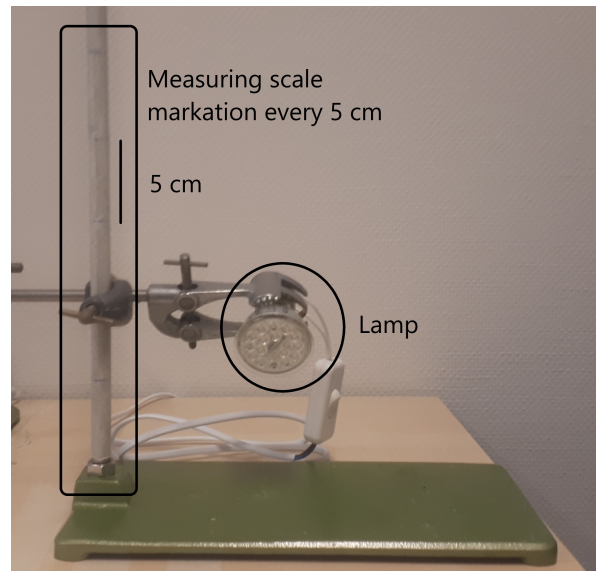


Figure 5.3: The light stand with the GU10 TRADFRI LED 345 lm lamp and height measuring scale.

ambient lighting is being used. One light standard with a lamp and the measuring scale that is used to adjust the height is shown in Figure 5.3. Note that the measuring scale starts where the pole is in the foot, so the lamp is actually 20 mm higher than the measuring scale indicates. Next to that, the distance to the ground is measured, so not to the sample top. The sample thickness is 3 mm, which should also be taken into account. For the type of lamp that is used a GU10 TRADFRI LED 345 lm lamp is selected. This lamp is chosen because the light intensity can be varied and there is little diffusion of the light.

In Figure 5.4 a conceptual top view of the test setup can be seen. The triangles with a 1 represents a lighting source and direction, and the square with a 2 is the area where the sample is placed and where the camera is. There are four lighting sources that have an angle of 90° with respect to one another. All these sources have the same distance to the sample. The sample should also be placed as indicated in the figure. In Figure 5.5 the top view of the realized setup

is shown and the components are indicated. The light intensity is measured using the Lux meter shown in Figure 5.5 when all lights are on.

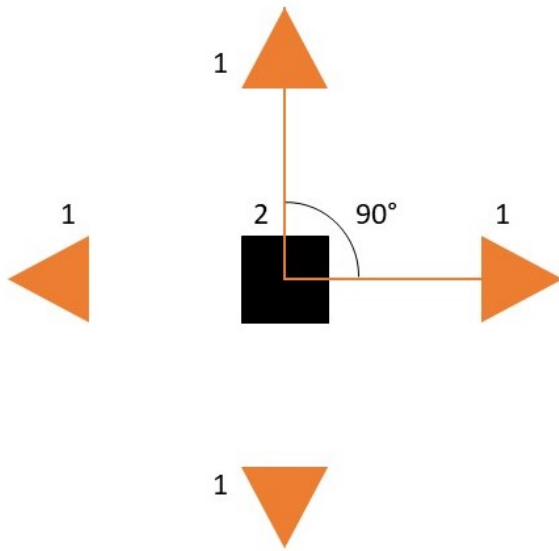


Figure 5.4: Conceptual top view of the setup with the lighting sources (1), the sample (2) and the distance.

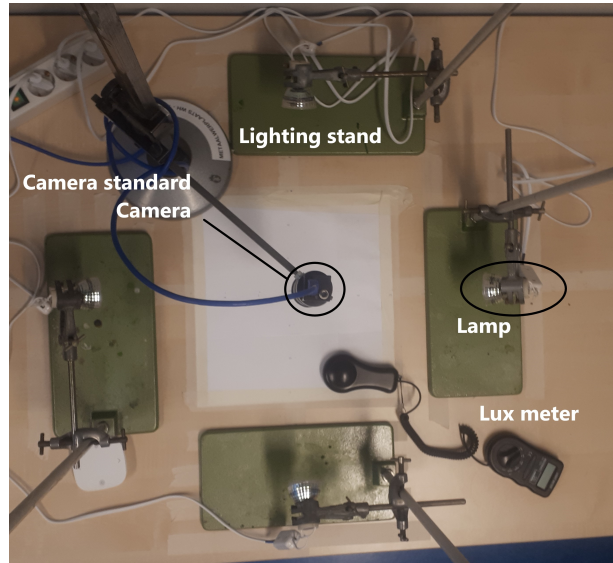


Figure 5.5: The realized top view of the setup.

The desk that was used to realize the setup was of finite size, limiting the distance at which the lighting sources can be placed. In the setup the height and distance were marked to keep track of the position of the lighting source. The distance indications can be seen in Figure 5.6. The point where the camera is above is shown with a plus, and the position at which the light standards will be placed is indicated. The height measuring scale can be seen in Figure 5.3.

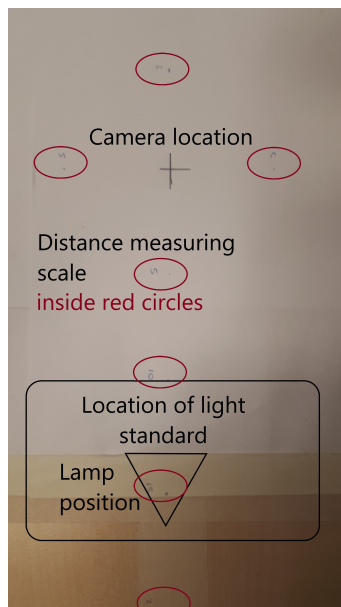


Figure 5.6: Close up of the distance measuring setup with component locations shown.

Several setups will be tested and the one that generates the most contrast using the base sample will be used for all further experiments.

5.2 The procedure

A standard procedure is used to ensure the consistency of the environment and results. When images are made the room is darkened to 10 lux so that the intended light sources are responsible for by far most of the light. The sample is placed on the indicated area to ensure all the lighting sources are at an equal distance. Of one sample, five pictures are taken. One with all four lighting sources on, and four with only one of the lighting sources on. In between taking these pictures the camera and the sample are not moved. After making these images, the sample can be removed. Per sample, 3 measurements are taken to test the consistency of the performance of the sample.

5.3 Experiments

In the experiments three parameters will be varied. These three parameters are the height, distance and the light intensity of the lighting source. The angle of the light will be modified depending on the height and distance so that it casts its light directly onto the samples. Three different heights, four distances and three lighting intensities will be used. The light intensity is measured when all lights are on. When there is directional lighting with only one source on, the lux value is about one-quarter of the value in the table. It is expected that this lighting intensity influences how much contrast is created between the black and the white modules. The values of the parameters that are varied are shown in Table 5.2. The values of the angle θ are shown in Table 5.3.

Table 5.2: The different values of the parameters concerned with the lamp setup.

Height	70 mm	120 mm	170 mm	220 mm
Distance	220 mm	270 mm	320 mm	
Intensity	+/- 1000 lux	+/- 2000 lux	+/- 4000 lux	

Table 5.3: The angle of corner θ with the different height and distance.

Distance/ Height	220	270	320
70	72.3	75.5	77.7
120	61.4	66.0	69.4
170	52.3	57.8	62.0
220	45.0	50.8	55.5

The three parameters will be varied in all configurations, so in total 36 different configurations are tested. The results should show the correlation of individual parameters as well as possible interdependence between the two. It should be noted that 3 to 4 different values are not a lot, however, if more are used then the number of experiments increases significantly. The 3 to 4 variations are expected to give some bearing on how the parameters influence the results. Based on the outcome of the experiments one setup is chosen as standard. This setup is to be

used when looking at different samples.

The other parameters that are expected to influence the results have to be eliminated. The environmental effects are eliminated through the procedure (darkening the room). Next to that, there are the influences that the program has on the results. The same program is used to ensure that these influences are similar. Finally, the sample that is used is expected to have an effect on the results. To eliminate this variation the same sample is used in the experiments, the basic sample of which the properties are shown in Table 3.1.

5.4 Results

The different setups are made, and images of the baseline sample are taken and processed for each of the setups. As discussed in chapter 4, the scoring is determined based on the number of sigmas between the means of the black and white modules which from now on will just be called sigma. In this paragraph the results will be discussed.

The two properties that were inspected closely at first were the sigma over the height (of the lighting source) and the sigma over the distance. It became clear that these parameters did have effects. These effects seemed to be interconnected, so the sigma over the light angle was also plotted. It became clear that the angle of the light has a larger and clearer influence on the generated sigma. In Figure 5.7 the number of sigma (standard deviations) between the black and white modules is shown on the y-axis and the angle of the light source is shown on the x-axis. In Appendix D the plots where the height and distance are on the x-axis are shown.

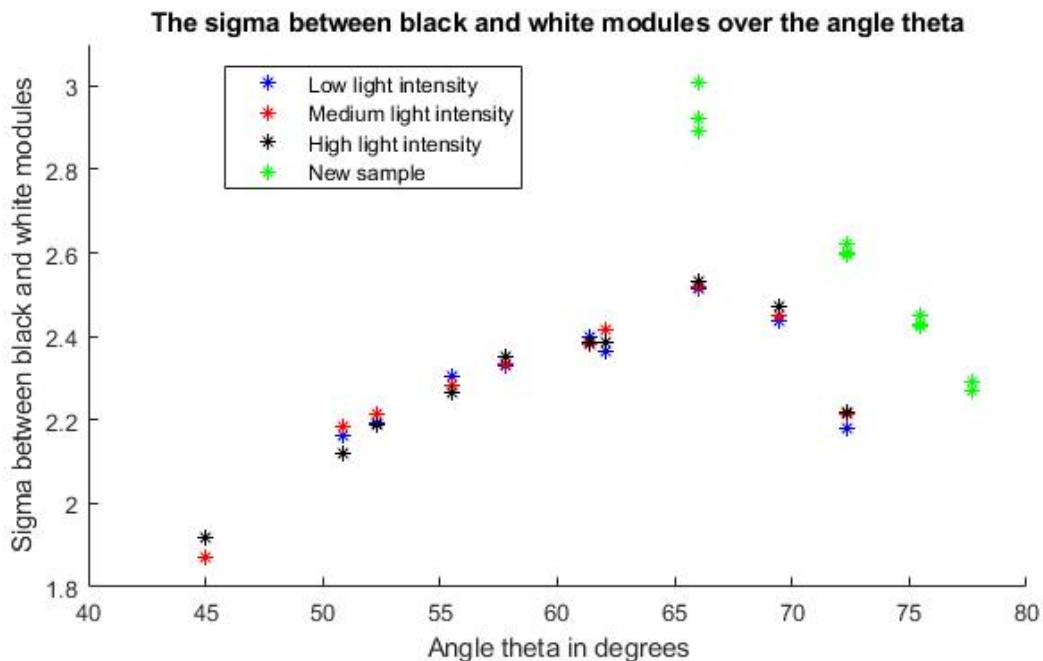


Figure 5.7: The number of standard deviations (sigma) between the black and white modules on the y-axis and the light angle θ on the x-axis with the different light intensities in different colours.

In Figure 5.7 the points represent a found sigma value during one of the experiments. From a small angle the sigma increases up to an angle of about 66° , after which the sigma value

decreases. It seems clear that the angle has a strong predictive value on the sigma and that there is an optimum angle. The measured point seems to peak at around 66° . Based on this experiment, it seems that there is an optimum around this light angle. This angle corresponds to a height of 120 mm and a distance of 270 mm. Based on this information, these settings would be the best option.

One thing that stands out in Figure 5.7 is the high performance of the reprinted sample (green dots). It was expected that something about the original baseline sample was suboptimal. To test this, it was investigated under a microscope. This is described in paragraph 8.1. This resulted in the destruction of the original base sample. A new sample with the same settings was printed, and this one performed much better. The new sample outperforms the old sample. To find out why the sigma between black and white pixels is looked at.

The sigma between black and white pixels over the lighting angle is shown in Figure 5.8. Instead of using the module average grayscale value when calculating the standard deviation all the individual pixels are used. This gives a more accurate idea of how likely pixels are to be correct, but it neglects the effect of clustered pixels. This can alter the results. In this case it can be seen clearly that the trend is the same. So, when looking at single pixels there also is a maximum around 66° .

It is interesting that the peak of the new base sample is not as high as in the sigma over modules. A closer investigation of the data revealed that the mean average grayscale value of the black pixels is lower in the new sample than in the old sample, while the value of the white pixels remains the same. This means that the black modules and pixels are darker. The standard deviation of the black samples is smaller in the new sample as well, this causes an increase in the sigma between modules. The increase in sigma between pixels is not as large because the standard deviation of the white pixels increased in the new sample.

The setups with a distance of 70 mm were initially not all tested. This was added later. However, at that moment the original base sample had been destroyed so the new sample was used.

It should also be tested if the angle theta is the main influencing parameter of the three, or if height and distance play a secondary role. To see if height and distance play a secondary role, points with a similar angle but with a different height and distance are compared. Two groups of points are compared. At first, the points with an angle of 61.4° (D 220, H 120) and 62.0° (D 320, H 170). In Figures 5.7 and 5.8 the points score very similar. Looking at a second group of points, with angles of 50.8° (D 270, H 220), 52.3° (D 220, H 170) and 55.5° (D 320, H 220), the results are almost the same and correspond to the trend. From these results, it is concluded that only the angle of the light plays a major role, and the individual influences of the height and distance are negligible compared to the light angle. Sigma over height and distance graphs were initially also investigated. These are left out of the report, but those graphs were inconclusive.

The sensitivity and specificity over the angle theta are shown in Figure 5.9. This data is only used for context, and in this case, it offers a nice insight into a key difference between the old and new base samples. What stands out is the lower sensitivity and higher specificity that is reached using the new sample (circled green). This is most likely because a more ideal threshold value was selected for the image binarization. Apart from that, it can be concluded that there is no clear correlation between the sensitivity and specificity and the angle of the light source as all measurements stay in the same range without showing a clear trend.

Another parameter that was expected to influence the results of the setup is the light intensity.

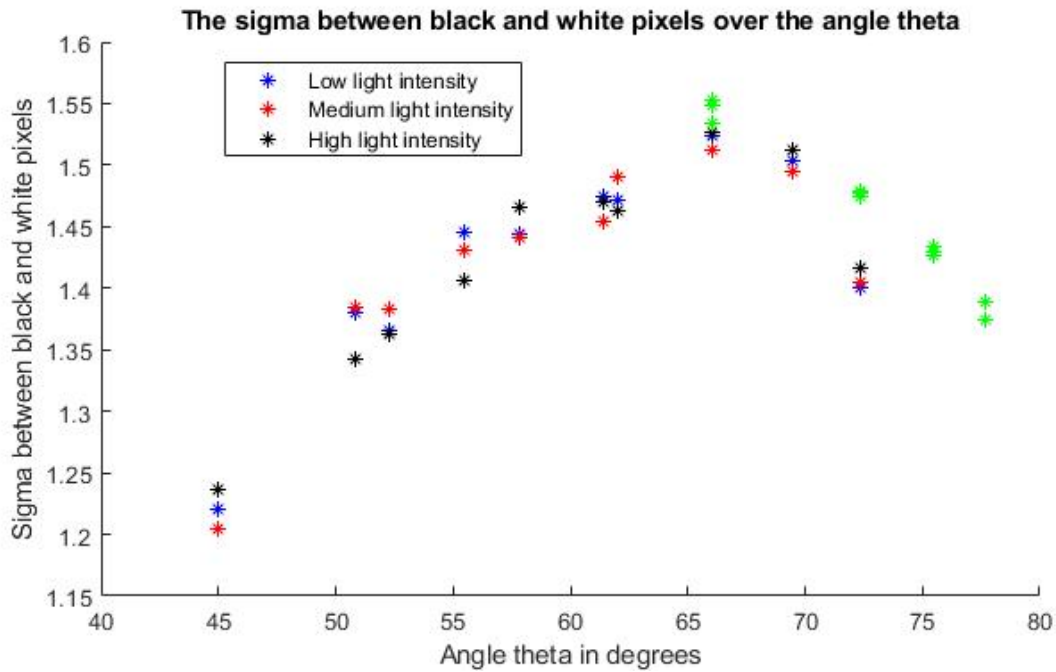


Figure 5.8: The sigma between the black and white pixels on the y-axis and the light angle θ on the x-axis with the different light intensities in different colours.

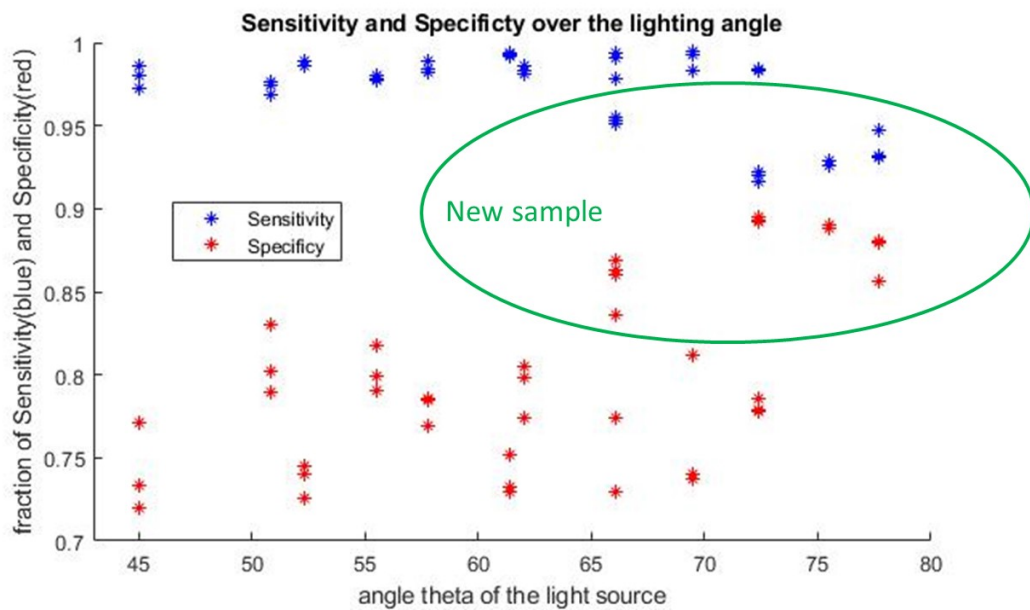


Figure 5.9: Sensitivity and specificity on the y-axis with the lighting angle θ on the x-axis.

For each angle, three different measurements have been made, with 3 different light intensities. In Figures 5.7 and 5.8 the different light intensities were also included. To more clearly show the effect of the light intensity the sigma over light intensity is shown in Figure 5.10. From this figure, it is clear that the light intensity has no significant effect on the sigma between the black and white pixels. The sigma between modules remains constant if the height and distance remain constant while the light intensity is changed. It can be concluded that the light intensity does not matter for the number of sigma that is generated.

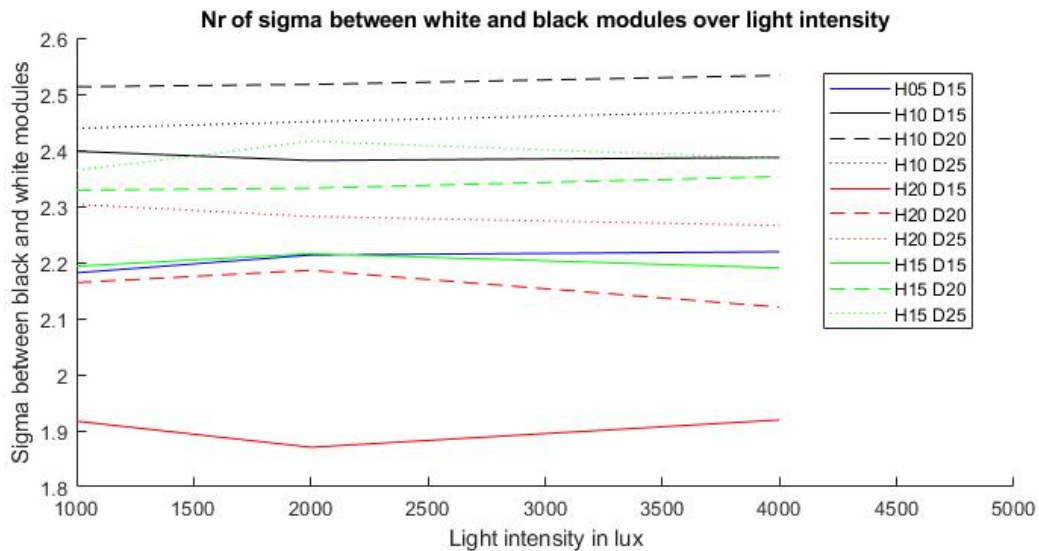


Figure 5.10: The sigma between the black and white pixels on the y-axis and the light angle θ on the x-axis with the different light intensities in different colours.

Light intensity might matter for being able to select a better threshold value in the binarization of the image. To check this, the sensitivity and specificity over the light intensity are shown in Figure 5.11. There is no correlation that is clear enough to draw a certain conclusion. However, some choice needs to be made with regard to the light intensity that is to be used. If anything can be said it seems that the specificity seems slightly higher in low-lighting conditions. Because of this low lighting will be used. However, it should be noted that based on this experiment this effect is not considered to be proven. If the light intensity did matter, one effect that could play a role is light scattering over longer distances. However, this is not the case.

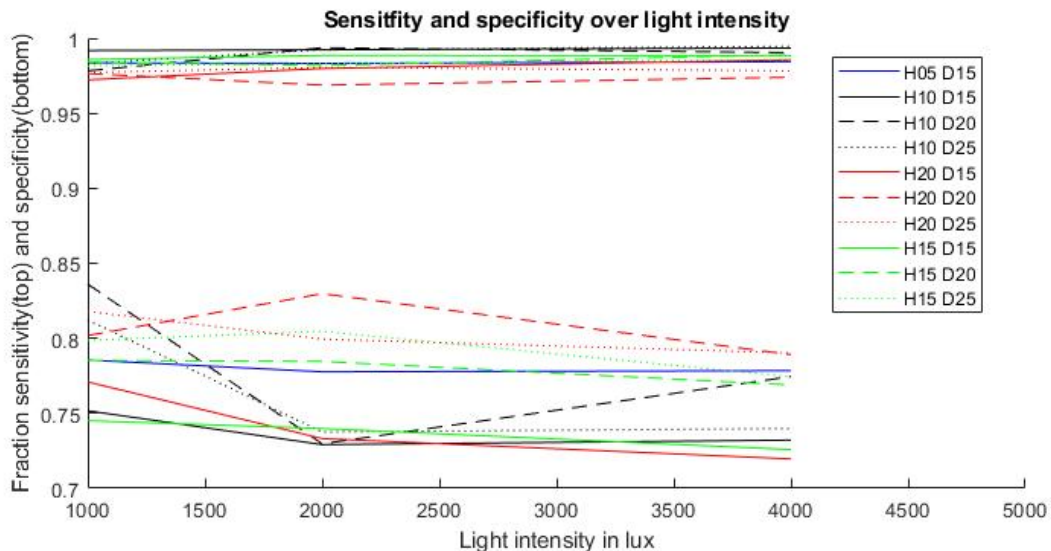


Figure 5.11: Sensitivity and specificity on the y-axis with the light angle θ on the x-axis with the different light intensities in different colours.

5.5 Conclusion

It can be concluded that the setup should have an angle of around 66° to generate the maximum number of sigma between black and white modules. The light intensity does not significantly influence the results. However, a setting has to be selected. In the experiments, low lighting conditions will be used. Low lighting occasionally resulted in higher specificity, a sign that image binarization was more successful.

6 EXPERIMENTS

In this chapter, the results of the experiments in which the design parameters of the samples are varied are shown. Several design modifications have been made to find out what the effects of these modifications are. Four major design parameters are varied. At first, the size of the code, and thereby the modules, is varied. Next, the variation of the carving depth will be discussed. Thereafter, the effect of curvature is described. Finally, surface modifications are evaluated. Correlations between the modification and the scoring will be identified. Conclusions are drawn as to what this implies for the design guidelines.

6.1 Module size

The first parameter that will be discussed is the size of the code. In the experiments, a QR code with a module size of 29x29 was used. Not only the total size is important, the module size is even more important. This is because the module size serves as a general property. The size of the code (module-wise) can change, but if the module size remains the same similar results will be achieved. This is because the printer will be able to reliably produce the smallest features, the modules. Thus, the goal of the size variation is to find the minimal size of the modules at which the code still reliably be decoded. In Table 6.1 the size variations are shown. It is expected that if the module size is smaller the limited resolution and part growth play a larger role, decreasing the results. It is thus expected that the results improve as the size increases. At first, samples with a size of 10, 15, 20 and 30 mm were printed to see what works. From these samples, it was found that somewhere in between 15 and 20 mm the code becomes readable. Because of this, the sizes that are investigated are in this range.

Table 6.1: The size variation of the samples.

Sample name	size 15	size 16	size 17	size 18	size 19	size 20	size 21	size 22
Size [mm]	15	16	17	18	19	20	21	22
Module size [mm]	0.52	0.55	0.59	0.62	0.66	0.69	0.72	0.76
Carving depth [mm]	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0

In Figure 6.1 the standard deviation between the average grayscale value of modules is shown with respect to the QR code size. In the figure, it can be seen that the sigma increases as the size of the code increases. To show this, two predicted curves are fitted. The quadratic one is almost linear and seems to be the most accurate. It can be concluded that there is a positive correlation between size and sigma. It is expected that this does not continue indefinitely, as at some point the modules become too large which results in lighter shadows. However, this point has not been found in this experiment.

It stood out initially that the sample of 20 mm performed worse than the sample of size 19. The causes of this will be analyzed in more depth in paragraph 8.1. To test if there is a performance peak at a size of 19 mm, three additional samples were printed and processed. The sample

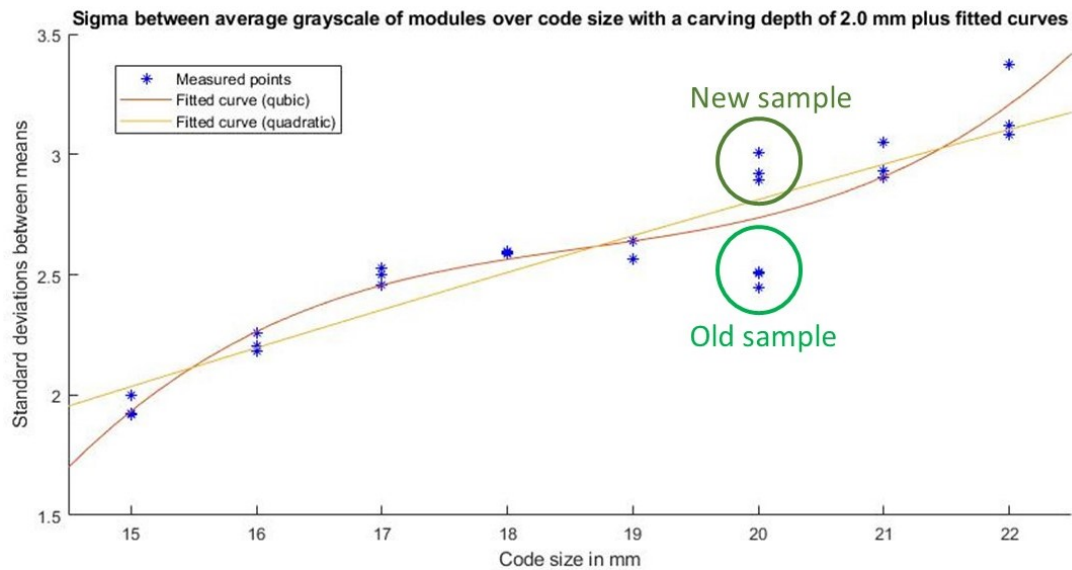


Figure 6.1: The standard deviations (sigma) between average grayscale of the black and white modules on the y-axis and the code size in mm on the x-axis for a sample with a carving depth of 2.0 mm with a fitted curve and the old and new baseline sample indicated.

with size 20 was reprinted. Also, larger samples were produced. These three new points are already added in Figure 6.1. It was observed that the second 20 sample outperforms the initial sample, fitting in well with the expected correlation. The larger samples also result in a higher sigma. A question that should be asked is why the found results are obtained. In this case, the specified resolution can be achieved by the printer as the module size gets larger, while at lower values the printer capabilities are not sufficient and result in errors on the boundaries of modules.

The two different basic samples have different performances. This is because the new sample has a higher difference between the means of black and white pixels/modules. The standard deviation of the black modules is lower in the new sample. However, the standard deviation of the white modules is larger. It still results in a significantly higher difference in generated sigma between module grayscale values. Based on this, it can be concluded that there is variation between samples that are supposed to be identical. It is expected that this difference arises as a result of where the sample is in the printer. If it is on the bottom of the building volume, there is more time for part growth to take place and that can explain the difference between samples.

In order to decode the code, a minimum number of sigma is needed. It has to be determined what the minimum value should be. The code should be robustly decodable. In this experiment all the samples were readable. However, this was under controlled circumstances and with a modified program to read them. The errors per module can be used to help determine what the minimum sigma should be. The smaller samples did contain more errors. The number of errors in each of the samples are shown in Figure 6.2. It is clear that the larger the module size gets, the fewer modules with more than 50% errors there are. It can be observed that the size 18 sample has fewer modules with more than 50% errors than the samples of sizes 19 and 20. Based on these error figures, the samples from 17 contain no modules with more than 70% errors. This means that the most problematic modules can still correctly be identified, and a good performance is achieved. The new sample of 20, 21 and 22 contain almost zero modules with more than 50% errors. This corresponds to a perfect performance. Based on this a module size of 0.59 mm delivers a medium performance, and 0.69 mm gives a high performance. These two numbers are rounded up to 0.6 and 0.7 mm in the design guidelines for size. This

corresponds to a sigma between modules of 2.5 and 2.9.

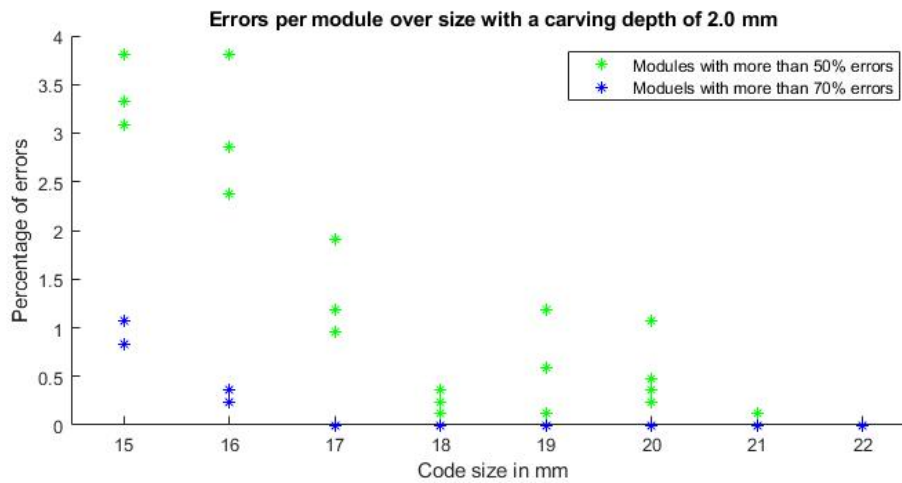


Figure 6.2: Percentage of errors on the y-axis and the module size on the x-axis for a sample with a fixed carving depth of 2.0 mm.

In Figure 6.3 the results of the sensitivity and specificity over size are shown. It can be observed that the sensitivity remains steady across the samples. The specificity increases as the code size increases. However, this seems to happen in steps. The first step happens between 16 and 18 mm, and the second step is between 20 and 21. This supports the earlier found distinction between the two groups (medium and high performance). The explanation for this increase in specificity is that the printer is unable to achieve the specified resolution, and solid parts turn out too large. Part growth and bad removal of material are also at play here.

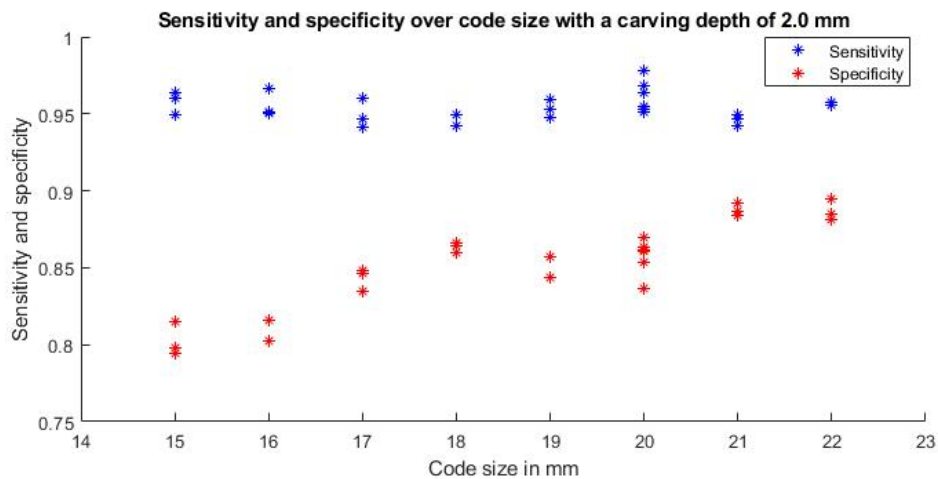


Figure 6.3: Sensitivity and specificity on the y-axis and the module size on the x-axis for a sample with a fixed carving depth of 2.0 mm.

Conclusion

Now that these results are known, conclusions are drawn as to what this means for the design guidelines. To decode the code correctly, a certain number of modules need to be identified correctly. To be able to do so, a certain minimum number of sigma is needed. By looking at Figure 6.1 and 6.2 it seems like a value of 2.5 for sigma between modules. These values are reached at a code size of 17 mm with a module size of 0.59 mm. At a size of 21 mm the

quality increases again, if there is enough space available this should be the preferable size. From Figure 9 in a designer’s guide for dimensioning and tolerancing sls parts by Sheepersad [13] it can be seen that isolated holes of 0.5 give ‘medium’ performance at a vertical orientation. Printing isolated modules is like printing holes, so these results are in line with their observations. Weiss et al. [22] embed slits, which are like multiple modules next to each other. From a thickness of around 0.3 mm slits are successfully embedded. Thus, it can be concluded that with a margin the minimal module size should be 0.6 mm, and 0.7 if there is space available.

6.2 Carving depth

The second parameter that is varied is the carving depth. This is the depth of the negative modules. It is expected that deeper carvings result in darker shadows. If the carving becomes too shallow the distinction between the black and white modules disappears, as there are hardly any shadows cast. The carving depth is varied from 0.4 mm to 2.4 mm. The samples are shown in Table 6.2.

Table 6.2: The properties of the carving depth samples.

Sample name	CD 0.4	CD 0.8	CD 1.2	CD 1.6	CD 2.0	CD 2.4
Sample size [mm]	20	20	20	20	20	20
Carving depth [mm]	0.4	0.8	1.2	1.6	2.0	2.4

The sigma between black and white modules as a function of carving depth is shown in Figure 6.4. As the carving depth increases, so does the number of sigma between the average grayscale value of black and white modules. To show this more clearly, fitted curves are included in the figure.

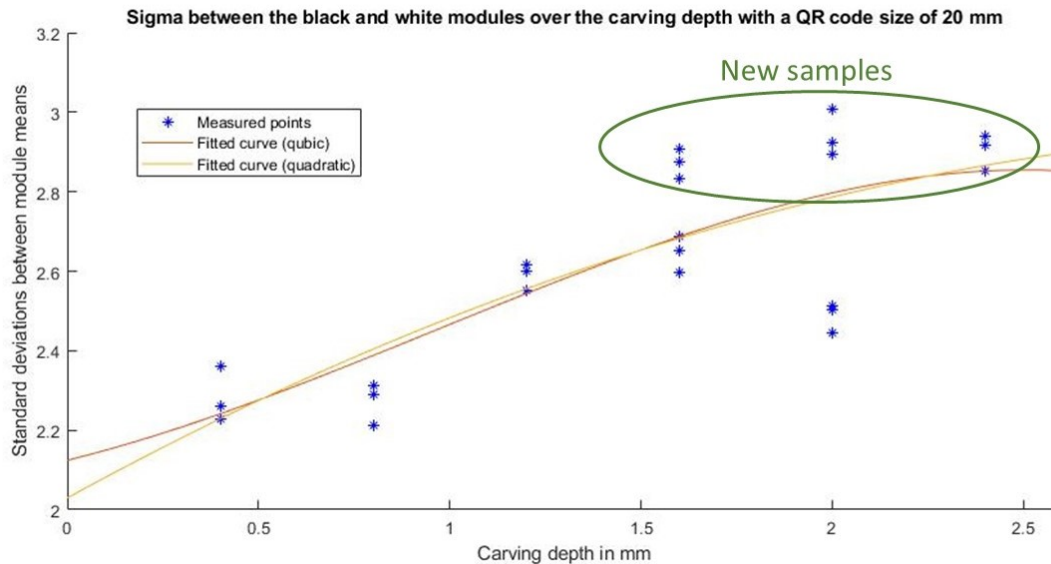


Figure 6.4: Standard deviation (sigma) between the means of the black and white modules and pixels on the y-axis with the carving depth on the x-axis with a QR code size of 20 mm for a 29x29 QR code with the new samples indicated and two fitted curves.

As can be seen from the fitted curves in the figure, there is a positive correlation between the generated sigma and the carving depth. It is unclear if this correlation continues, or if it reaches

a plateau. When fitting using a quadratic polynomial it continues, but a cubic polynomial shows a plateau. The original sample with a carving depth of 1.6 mm was outperforming the basic sample with a carving depth of 2.0 mm. Three new samples were printed with carving depths 1.6, 2.0 and 2.4 mm. The new samples (1.6, 2.0 and 2.4) have a larger difference between the means of the black and white modules than the old samples, which causes the sigma to increase.

These three new samples outperform the old samples, and in Figure 6.4 these are the top performers in terms of sigma between means of modules. The new basic sample does outperform the sample with a carving depth of 1.6 mm, and also outperforms the sample with a carving depth of 2.4 mm. The results of the new sample are all quite similar. This seems to indicate a plateau is reached at a carving depth of 1.6 mm.

The average grayscale value of the black modules being lower (darker) does indicate that there is a difference. This can be because of printing differences between different batches. The position of the samples in the building volume most likely plays a role, if the samples are printed earlier then part growth plays more of a role as they are kept at elevated temperature for a longer time. Next to that, the percentage of virgin powder can vary even though it was assumed to be constant. The fact that there are differences between batches is problematic, as it indicates that there are uncontrolled parameters.

In Figure 6.5 the sensitivity and specificity are shown. It stands out immediately that when the carving depth is 0.4, the specificity is better than the sensitivity for the first time. This means that more modules are correctly detected as black instead of white. This is because there was very little contrast, the total difference between the means of the black and white modules was lower. This made finding a threshold value harder, and caused pixels white modules to be identified as black.

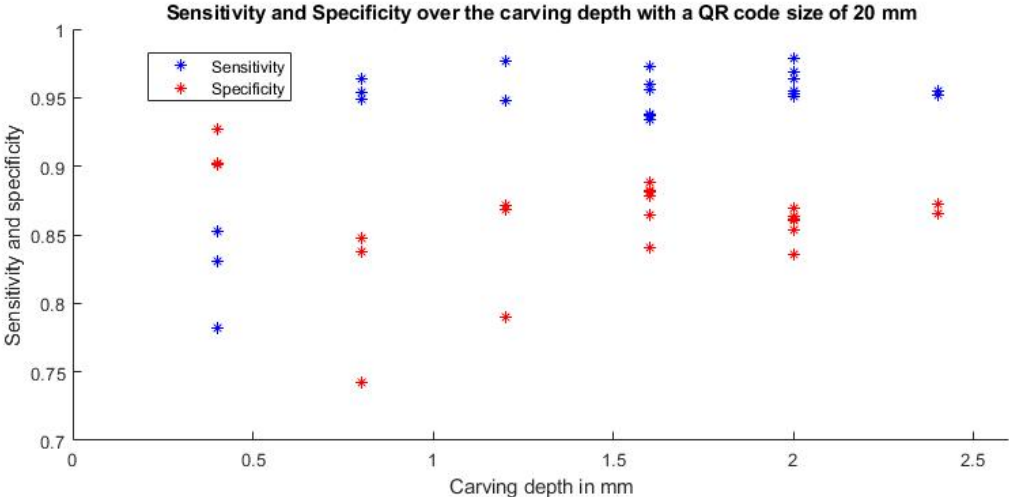


Figure 6.5: Sensitivity and specificity on the y-axis with the carving depth on the x-axis for a sample with a constant code size of 20 mm for a code with 29 modules.

The performance of the plateau starting at 1.6 mm is expected to be good because both samples achieved a sigma higher than 2.5, namely 2.6 and 2.9. The value of 2.5 was found in the previous paragraph when looking at the errors. To check if this boundary condition is valid, the percentage of modules with high error rates is investigated. The percentage of modules with high errors is shown in Figure 6.6. It can be seen clearly that the samples with a carving depth of 1.6 and up outperform the other samples. Thus, the quality is good, and it can be concluded

that a carving depth of 1.6 mm and up works best for flat surfaces.

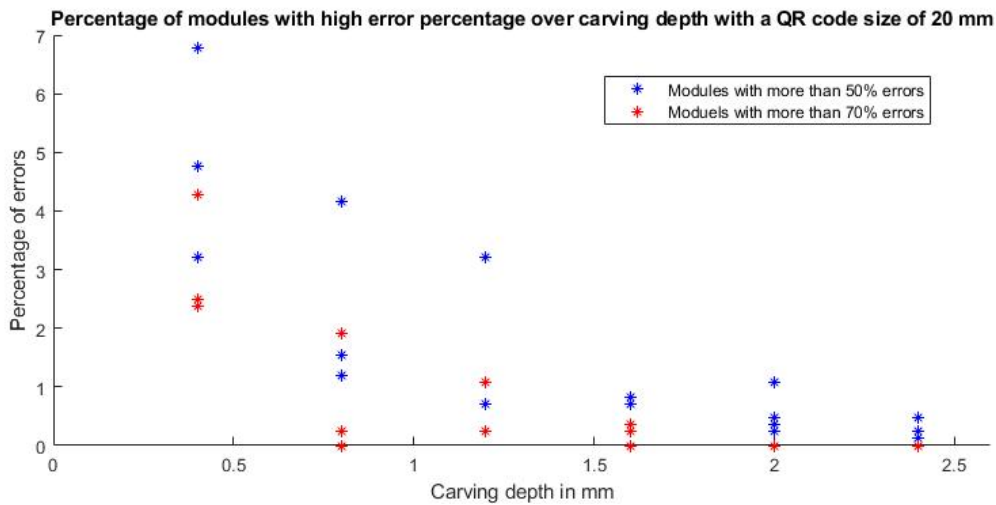


Figure 6.6: Percentage of modules with a high error percentage on the y-axis with the carving depth on the x-axis for a sample with a constant code size of 20 mm for a code with 29 modules.

Conclusion

From this set of experiments, it can be concluded that if the module size is 0.69 from a carving depth of 0.4 mm to 1.6 mm the carving depth and the number of sigma between modules is positively correlated. When the carving depth is greater than 1.6 mm, the performance does not increase by a lot. From a carving depth of 1.6 mm it reaches a plateau. For the design guidelines, it is desirable to be on this high-performance plateau, while having minimal intervention in the product. Thus, a carving depth of 1.6 seems to be the most suitable for flat surfaces.

6.3 Pattern inclusion

In the literature research, it was found that including a surface modification could increase the generated contrast. This feature has been included through what is called here 'pattern inclusion'. In pattern inclusion, a pattern is included at the centre of each negative module. It is expected that this increases the generated contrast. In Figure 6.7 an example of a QR code with circles as included patterns is shown. Next to that, in Figure 6.8 an example of a sample with pattern inclusion is shown.

Two different patterns are tested, one with diamonds and one with circles added in the middle of the modules. These shapes had a depth of 0.1 and 0.2. As an extra variation the carving depth is varied, this is because the pattern inclusion is expected to contribute to extra contrast especially when the carving depth is insufficient. By doing so, it might enable a lower carving depth to generate sufficient contrast. The standard carving depth of 2.0 mm is used. Additionally, a carving depth of 0.8 mm is used. This means that eight samples have been printed to test the effect of pattern inclusion. In Table 6.3 the different properties of the samples are shown.

In Figure 6.9 the sigma over the pattern depth is shown. For completeness, the base sample (CD 2.0 mm) and the sample with a carving depth of 0.8 mm are also included. The samples that included a pattern are represented by that pattern in the Figure, respectively circles and diamonds. The colours represent a carving depth and whether the sigma over modules or the sigma over pixels is shown.

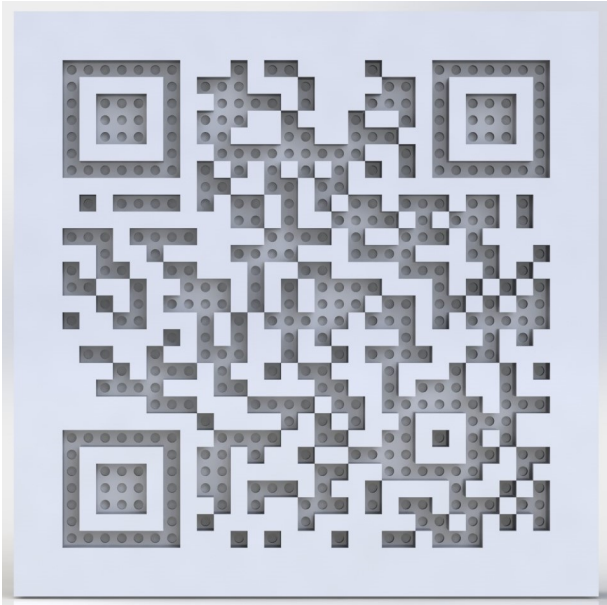


Figure 6.7: A render of a QR code with a circle Figure 6.8: An example of a sample with a pattern (C0.2) and a carving depth of 0.8 mm.

Table 6.3: The different properties of the samples that included a pattern.

Sample name	C2/.1	C2/.2	C.8/.1	C .8/.2	D2/.1	D2/.2	D.8/.1	D .8/.2
QR code size [mm]	20	20	20	20	20	20	20	20
Carving depth [mm]	2.0	2.0	0.8	0.8	2.0	2.0	0.8	0.8
Pattern type	Circle	Circle	Circle	Circle	Diamond	Diamond	Diamond	Diamond
Pattern depth [mm]	0.1	0.2	0.1	0.2	0.1	0.2	0.1	0.2

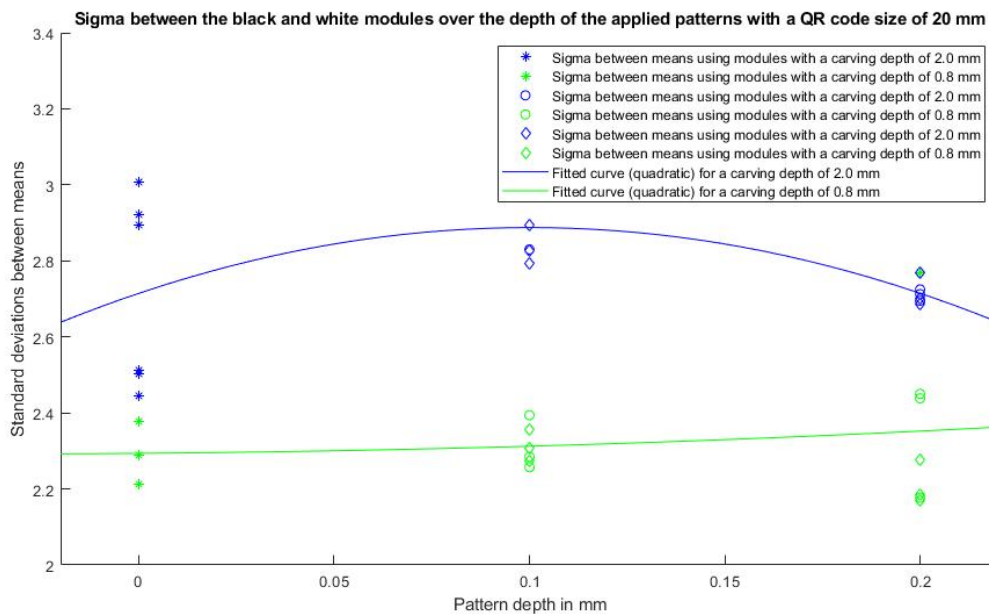


Figure 6.9: Standard deviation (sigma) between modules on the y-axis with the pattern depth on the x-axis for a sample with a QR code size of 20 mm for a 29x29 QR code.

From Figure 6.9 it seems that the inclusion of a pattern does not yield significant effects, as the results stay within the range of the samples without pattern inclusion. This can be seen clearly when looking at the fitted curve with a carving depth of 0.8 mm, as it almost is a straight line. The curve with a carving depth of 2.0 is not straight, but this is a result of the inferior performance of the original baseline sample. Thus, it can be concluded that the pattern inclusion with a pattern depth of 0.1 to 0.2 mm does not influence the results. The most likely explanation is that the printer is not able to reliably achieve the required feature size that is needed for this form of pattern inclusion. It is also noted that the form of the pattern (circle or diamond) does not seem to make a significant difference for the number of sigma that is generated. This can mean several things: the form of the pattern does not matter, circles and diamonds are too much alike or the required resolution is not achieved anyway. That the required resolution can't be met is the most likely explanation, as the module size is already being pushed to the limits and the features are one third the module size.

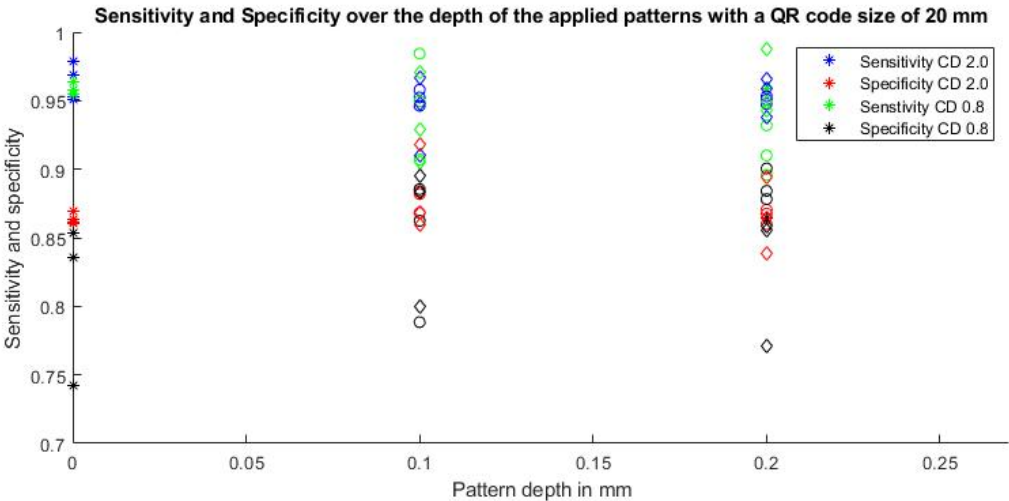


Figure 6.10: Sensitivity and specificity on the y-axis over the pattern depth on the x-axis for a sample with a QR code size of 20 mm for a 29x29 QR code.

A look is taken at the sensitivity and specificity performance of the samples. In Figure 6.10 the sensitivity and specificity are shown over the depth of the patterns. Additionally, to mark the difference in carving depth of the samples colours are used and to indicate what type of pattern is used different shapes are used. This figure is a mess, and no meaningful correlations can be observed. One aspect that can be noticed is that the specificity of the sample with a round pattern with a depth of 0.2 mm and a carving depth of 0.8 (C.8/.2) is high. To see what caused this effect and if this has anything to do with pattern inclusion, individual images of the samples are looked at.

The imported image of the sample C.8/.2 and C2/.2 are shown in Figures 6.12 and 6.13. For good comparison in Figure 6.11 the image with a carving depth of 0.8 without patterns is shown. It can be seen that the pattern inclusion has more effect in the sample with a lower carving depth. It seems that the image of C.8/.2 is more blurry, this is most likely caused by the patterns being darker and thereby the rest of the area that is supposed to be black appears lighter. It seems more like an optical illusion than a conclusive result. The specificity increases, which indicates that the binarization threshold did favour the white pixels less than usual. For evaluation the sigma is used, but as said already that does not show a strong effect. However, the pattern does leave some effect. Perhaps these properties could be used in an advanced algorithm to more precisely identify the modules. However, that is outside the scope of this research.

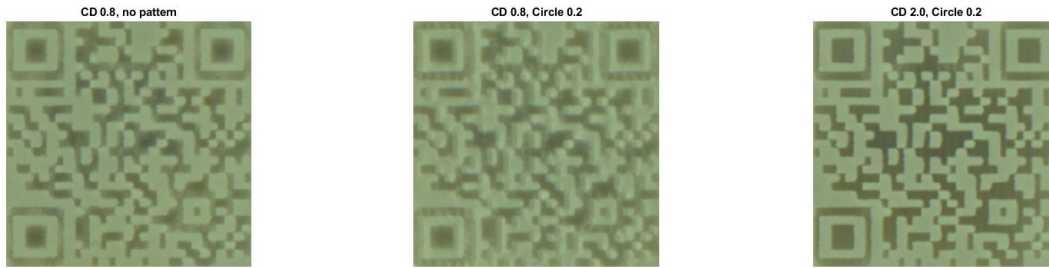


Figure 6.11: Image of CD0.8. Figure 6.12: Image of C.8/.2. Figure 6.13: Image of C2/.2.

It can be concluded that the inclusion of shallow patterns does not significantly affect the generated contrast. Pattern inclusion will not be recommended in the design guidelines.

6.4 Simple curvature

Different angles of curvature are investigated. In this paragraph, the results of samples with a single surface curvature will be discussed. Not all could be evaluated because the program was unable to identify the code. At first, it is useful to explain how the surface curvature is determined. In Figure 6.14 a representation of the angle θ as an expression of the surface curvature is shown. The size of the code remains constant at 20 mm, but the radius r is varied. As a result, the angle θ changes. Samples with a surface curvature from 10° to 50° are produced. The properties of these samples can be seen in Table 6.4. It can also be seen, the code is embedded on a hollow cylinder with radius r . This radius is also included in the table.

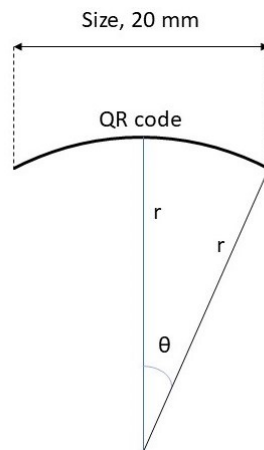


Figure 6.14: Visual representation of simple angle surface curvature.

Not all samples yielded readable results. In Table 6.5 it is shown if the sample was evaluated and whether or not it was readable. Evaluated shows if the script is able to correctly identify and evaluate the QR code. Yes means the code could be processed with a threshold value of Otsu's multiplied by 1.08. Otsu 1.00 means the multiplication value for the threshold was modified to 1.00. No means the code could not be evaluated by the program. 2/3 means 2 out of 3 samples were evaluated. Readable represents if the QR code is readable by a normal QR code scanner from the original images. This was included because the program was unable

Table 6.4: The properties of the samples with simple surface curvature.

Sample name	SC50	SC45	SC40	SC35	SC30	SC20	SC10
QR code size [mm]	20	20	20	20	20	20	20
Carving depth [mm]	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Surface angle [°]	50	45	40	35	30	20	10
Radius outer circle [mm]	13	14.1	15.5	17.4	20	29	57

to evaluate some of the codes, while they did look like an improved algorithm might be able to identify and process the code. The source image of SC20 and SC30 are shown in Figures 6.15 and 6.16.

Table 6.5: The evaluability and readability of the samples with simple surface curvature.

Sample name	SC50	SC45	SC40	SC35	SC30	SC25	SC20	SC15	SC10
Evaluated	No	No	No	2/3 Otsu 1.00	Otsu 1.00	Otsu 1.00	Yes	Yes	Yes
Readable	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes

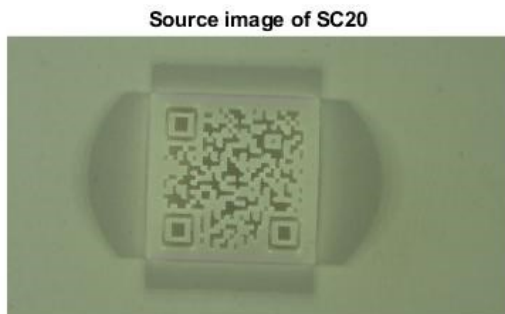


Figure 6.15: Source image of SC20.



Figure 6.16: Source image of SC30.

The results that were obtained are evaluated, and the number of sigma between black and white modules is shown in Figure 6.17. In this Figure, when looking at the fitted curves it can be seen that the number of sigma decreases as the curvature increases. The reason for this is that the directional lighting becomes a disadvantage rather than an advantage, as the shade of the curvature makes a part of the sample much darker. Partly because of this reason the evaluation of the samples with a higher curvature fails, although effects such as distortion might also play a role. It can be seen that the performance drops when the angle of curvature exceeds 20°. From these results, it can be concluded that when selecting an area to embed the code, that area should ideally have a maximum surface curvature of 20°. Areas with a surface curvature of up to 35° can be scanned, but the embedding quality decreases drastically.

As can be seen in Table 6.5, some samples that can be read using a QR code scanner can't be processed or required the image binarization threshold to be modified. This has to do with the way that the program detects where the codes are located. In Figure 6.18 the SC45 sample is shown in several stages. The imported image, grayscale image and binarized (using Otsu's method multiplied by 1.08) image are shown. The program is unable to process this binarized image because the detection patterns are not found by the program. This is because the isolated square is not longer isolated. The program tries to remove the noise from outside of the code, however, because the black shades reach inside the code as well this noise removal re-

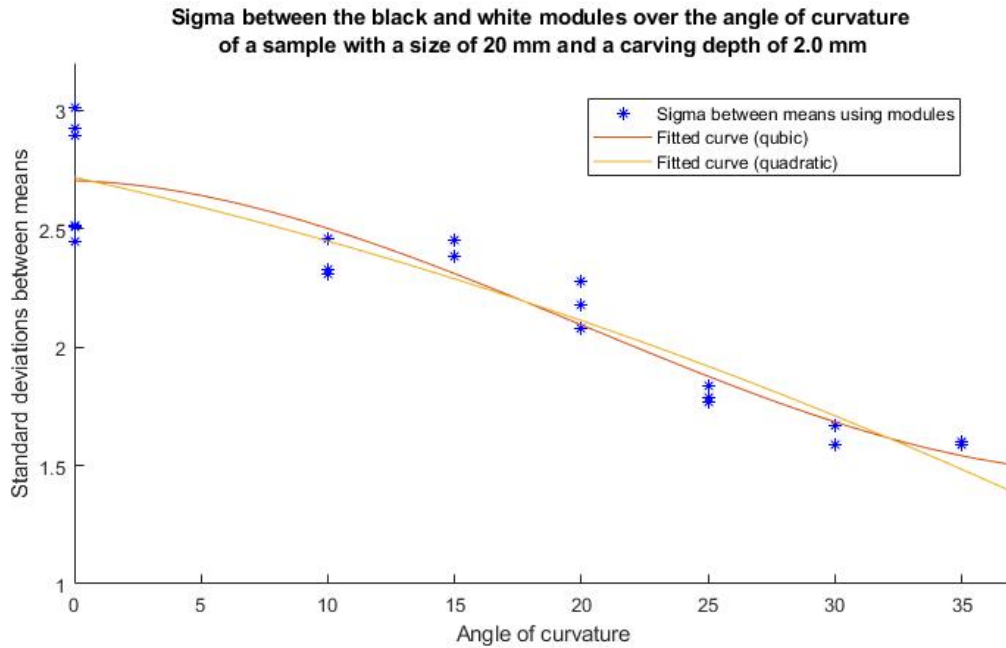


Figure 6.17: Standard deviation (sigma) on the over modules on the y-axis over the angle of the surface curvature on the x-axis of a sample with a size of 20 mm and a carving depth of 2.0 mm.

moves part of the code. These black connections are caused by the shadow cast by the light on the opposite side. This can be solved by using a smart binarization scheme, however, developing that is outside of the scope of this research. It might be beneficial to use a different setup when evaluating the embedment success of curved samples.

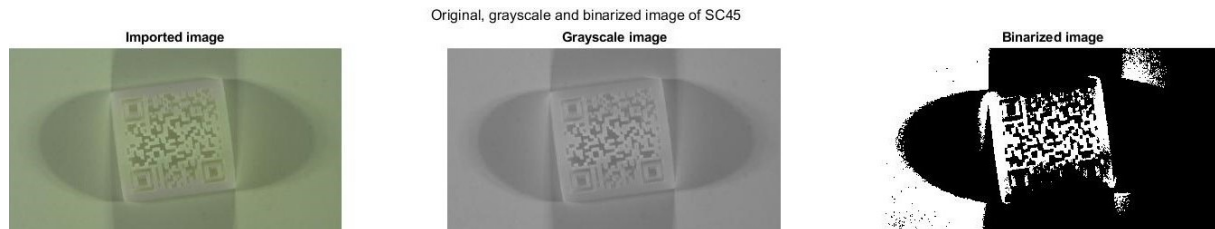


Figure 6.18: Binarization steps of the SC45 sample, note that the binarized image has black areas extending onto the code where it should be white.

Based on these results can be concluded that when selecting an area to embed the code a maximum surface curvature of 20° is permissible. A surface curvature of up to 35° does function, but results in a severe decrease in robustness.

6.5 Advanced curvature

Samples with more advanced curvature are also investigated. All samples are shown in Table 6.6. The samples can be separated in roughly four categories.

In the first category, different code sizes are printed on the same cylinder. The samples with code size 15, 20 and 25 mm on a curved surface with an angle of 30° (cylinder r20) are shown

Table 6.6: Sample names and properties of the advanced curvature samples.

Sample name	Size [mm]	CD [mm]	Surface modification
AC30s15	15	2.0	Simple surface angle of 30° as used for size 20 sample
AC30s25	25	2.0	Simple surface angle of 30° as used for size 20 sample
AC50s15	15	2.0	Simple surface angle of 50° as used for size 20 sample
AC50CD1.6	20	1.6	Simple surface angle of 50°
AC50CD1.2	20	1.2	Simple surface angle of 50°
AC50CD0.8	20	0.8	Simple surface angle of 50°
AC50CD0.4	20	0.4	Simple surface angle of 50°
ACsinP1A2	20	2.0	Sin wave of 1 period with an amplitude of 2 mm
ACsinP1A3	20	2.0	Sin wave of 1 period with an amplitude of 3 mm
ACsinP2A1	20	2.0	Sin wave of 2 periods with an amplitude of 1 mm
ACsinP2A2	20	2.0	Sin wave of 2 periods with an amplitude of 2 mm
ACsinP2A3	20	2.0	Sin wave of 2 periods with an amplitude of 3 mm
ACD3030	20	2.0	Surface curvature of 30° in two directions, like a ball

in Figure 6.19. In the second category, the carving depth is varied with a fixed surface angle. In Figure 6.20 three samples with varying carving depths on a curved surface are shown. In the third category instead of being printed on a cylinder, the code is printed on a sin wave. The amplitude and number of periods are varied. In Figure 6.21 shows the ACsinP1A3 sample, with a period of 1 and an amplitude of 3. Figure 6.21 shows the ACsinP2A1 sample, with a period of 2 and an amplitude of 1. The whole sample (24 mm) consists of a longer period so that the 20 mm on which the code is embedded has 2 periods. In a way, the simple curvature can be seen as a sin wave also. The period would be 1/2 and the curvature can be expressed as some amplitude. However, this description would not be completely accurate, as instead of a sin wave a circle is used. In the fourth category, the sample is curved in more than one direction. The code is now embedded on a ball instead of a cylinder. There is only one sample in this category, with a surface curvature of 30° in two directions. This sample is shown in Figure 6.23.

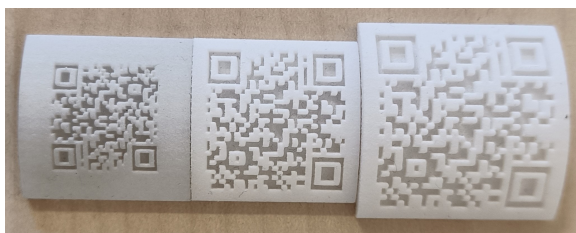


Figure 6.19: Samples with varying code sizes on a curved surface with a curvature of 30°.

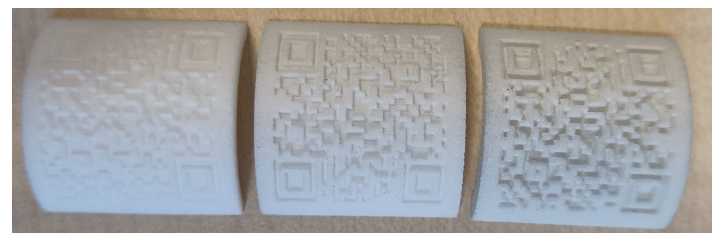


Figure 6.20: Samples with varying carving depths on a curved surface with a curvature of 30°.

The readability and whether or not the codes could be evaluated is tabulated in Table 6.7. Most of the samples with advanced curvature were unreadable and could not be evaluated. For the samples that could be read, the script had to be modified. Because of this, those results will not be compared to the main 'normal' samples, as a factor with large influence is modified.

The main cause for failure in the sin samples was that there are different shade levels on the sample itself. When using a global binarization threshold this causes problems. The same threshold is applied to the shaded areas and light areas. Obviously the light areas are lighter, and the shaded areas are darker. This causes white modules to be unreliable in the shaded



Figure 6.21: The ACsinP1A3 sample, with a period of 1 and an amplitude of 3.



Figure 6.22: The ACsinP2A1 sample, with a period of 2 and an amplitude of 1.



Figure 6.23: The ACD3030 sample with a curvature in two directions.

Table 6.7: The evaluability and readability of the samples with advanced surface curvature.

Sample name	Evaluated	Readable
AC30s15	Script modified, multiplication factor Otsu's of 0.90	Yes
AC30s25	No	No
AC50s15	No	No
AC50CD1.6	No	No
AC50CD1.2	No	No
AC50CD0.8	No	No
AC50CD0.4	No	No
ACsinP1A2	No	Yes
ACsinP1A3	No	Yes
ACsinP2A1	Yes but dealigned, multiplication factor Otsu's of 1.0	Yes
ACsinP2A2	No	No
ACsinP2A3	No	No
ACD3030	Script modified, multiplication factor Otsu's of 0.90	From sample

areas and black modules to be unreliable in the light areas. It can be concluded that if there are multiple depressions the reliability of the code decreases.

It should be noted that in some cases the different shade levels are not problematic. The samples ACsinP1A2, ACsinP1A3 and ACsinP2A1 are readable. It is expected that this is because the slopes are less sharp in those samples. As a result, shadows that were less dark were cast on the unlighted side of those samples. Some light did reach that side, while in the other samples, the slopes were too large and no light from one source reached the surface. This could be a result of the height at which the lighting sources were placed. It could also indicate a maximum permissible height change (slope) and slope concentration. The fastest height changes occurs in sample ACsinP1A3. The formula of ACsinP2A1 is given in equation 6.1. 24 is used as the size of the sample, the size of the code is 20 but there is 2 mm of edge around the code

$$ACsinP2A1 = 3 \cdot \sin\left(\frac{2\pi \cdot x}{24}\right) \quad (6.1)$$

The maximum derivative is $\frac{6\pi}{25}$, so the maximum slope is $\frac{6\pi}{25}$ mm/mm. In the simple curvature samples the maximum slope that still yielded successful results had an angle of 35°. This corre-

sponds to a value of 0.7023 mm/mm, quite similar to the maximum slope of ACsinP1A3. In the standard setup, a height of 12 mm distance of 27 mm is used. This results in the light having a slope of 0.44. It is clear that the maximum slope is larger than the slope of the light. Based on these results it can be concluded that a maximum slope of $\frac{6\pi}{25} = 0.75$ is allowed. However, it can't be said with certainty if the slope or the lighting direction is the critical parameter.

6.6 Conclusion

From the experiments, it can be concluded that for successful (decodable) embedment the minimum module size is 0.6 mm, while ideally, the modules are 0.7 mm or larger. A carving depth of 1.6 mm results in maximized contrast. Pattern inclusion does not significantly influence the results, thus it is unneeded to include patterns. A surface curvature of up to 20° is permissible. It could be argued that up to 35° the code is decodable, and thus that angle is permissible. A maximum slope of 0.75 mm/mm is allowed. Pattern inclusion is not recommended as it does not influence the results.

7 DESIGN GUIDELINES

In this section, the design guidelines that are established are shown. Along the design guidelines, a supporting document that contains all information to embed the codes is developed. This document contains all the required information, and a step-by-step approach to embed QR codes onto SLS-printed products. In this chapter, specific numbers for the are given that are based on the experiments. A layout for the step-by-step is chosen, and this is explained. More elaborate information is given for each step.

The design guidelines that follow from the experiments are tabulated in Table 7.1. It is unknown when the carving depth becomes too low if the size is increased, but since the goal is to have a code that is as small as possible that is not as relevant.

Table 7.1: The design guidelines for embedding 2D matrix barcodes onto SLS printed products.

Parameter	Recommended value	Lower performance value
Module size	0.7 mm or higher	0.6 mm
Carving depth	1.6 mm	-
Maximum surface curvature	20°	35°
Pattern inclusion	Don't apply	-

7.1 Supporting document

The layout that was chosen for the supporting document is a step-by-step visual flowchart. The steps are ordered chronologically. Three steps are required. At first, the code type is selected. Secondly, a location is chosen and finally, the code is embedded in the CAD model. At that point, the product can be printed on which a code is embedded. These three steps and all information required to carry them out is contained in three blocks on two pages. Arrows are added to make the workflow more natural. The supporting document is included in Appendix E.

Each block contains all the required information on the step that is to be taken. The required input, what action is to be taken and what outputs are produced are listed. Where it is possible, information to enable the action to be carried out is provided. This is not always the case, as there are some product and company-specific choices to be made. When this is the case, it is indicated. All the information that belongs together is placed in one box. In that way, it is clear that the information belongs together. By clarifying the input, action to be taken and output it is clear when a step can be initiated and if the step is carried out successfully.

The minimal amount of information to embed codes was included in these guidelines. The idea is that the fewer words that can be used to explain something, the better. This forces the information to be clear and concise. In that way, the guidelines can be applied in a practical manner.

The visual choice was made to place white text in a coloured box. An alternative would be to use an outline and place the information required in that outline.

7.2 Steps

Step 1: Selection of the code type

The first step is selecting the code type. The code type is determined by the amount of data that needs to be embedded, whether or not an internal 2D matrix barcode system is present and the availability of scanning systems. This is the input to be able to select a code type. For some 2D matrix barcodes, special scanners are required. If an internal 2D matrix barcode system is present, it is assumed that the company also has scanners for this system. In that case, they can use whatever barcode they would normally use, as long as it is a 2D matrix barcode.

If there is no 2D matrix barcode system present in the company some type of QR code will be used. The QR code is chosen because it has a high data density, has error correction and scanning is publicly available. Based on the amount of data that needs to be stored different types of QR codes will be selected. In Table 7.2 codes that work up to an alphanumeric string of 47 are shown. To keep it concise, a limited number of options are shown. These options offer a wide range of data storage capabilities. It is also possible to choose a code that is between the QR codes listed in Table 7.2, but if all options were included the table would not be very helpful when making a choice on what type of QR code to select. The table is provided in the supporting document to allow an easy selection of code types.

Table 7.2: Types of codes that are available for selection.

Alphanumeric data storing	Code type	Code size	ECC
5	Micro QR code	13x13	M
11	Micro QR code	15x15	M
13	Micro QR code	17x17	Q
16	Type 2 QR code	21x21	Q
29	Type 2 QR code	25x25	Q
47	Type 2 QR code	29x29	Q

To select a code type from the table the use needs to come up with what information needs to be stored. The most versatile form of information is the alphanumeric string. The user knows how many characters they need, and they choose the smallest number above their requirements. In the row, the code properties that correspond to the information storage need are shown. Based on that, the size of modules of the code that needs to be embedded is known. For decoding robustness, it is recommended to select a code with a ECC of Q or H (25% and 30%). Also, scanning capabilities for QR codes are more widely available than those for micro QR codes. With this information, a code type and size can be selected.

Step 2: Determine the position of the code

After selecting a code type, the next step is to determine where on the product the code should be embedded. The code size and product-specific information like geometry is required information to carry out this step. The size of the surface that is required is determined by the number of modules that need to be embedded. This number is multiplied by the minimum desired module size (0.7 mm). This is shown in formula 7.1 where size is the output module size, DMS is the desired module size and sizeQR is the size of the QR code in modules.

$$Size = DMS * sizeQR \quad (7.1)$$

An area of the size Size in [mm]x[mm] is needed to embed the code. In order to successfully embed the code, an area of this size needs to be found that has a maximum surface curvature of 20°. Additionally, it should not contain more than 5 depressions or a local surface angle higher than 35° from the direction normal to the camera orientation.

Another aspect is that this area needs to be easy to find when the product is removed from the printer. No research has been done on this aspect, but some sensible recommendations can be made. The code should be on an area that is visible when the product is removed from the machine. No parts should be blocking off the area. On the other hand, the intrusiveness should be minimal. It is not ideal to see the code throughout the product's lifetime. To avoid this, the code could be embedded on the bottom or the back of the product. Thin features should be avoided, if possible, as the intrusion of carving out a code might affect the functionality of the product.

It would make sense for the designer to look at unintrusive but still visible locations (like on the bottom of a part) and see if there is a space available there that matches the size and surface properties that were found in the research to be ideal. In case there is no such place available, a choice must be made. Either the code can be sized down by reducing the module size or a location that is in sight should be chosen. Reducing the module size decreases the performance, and thereby the robustness. In the experiments, this smaller code size should still work, but there is less margin of error. If a location in sight is selected, that location is in sight and thereby more intrusive. In Table 7.3 these options are tabulated.

Table 7.3: Design considerations when selecting the code location.

Options	Ideal scenario	Alternative
Minimum module size [mm]	0.7	0.6
Location	Unobstructed camera view and out of view during use	Unobstructed camera view but in view during use
Maximum surface curvature	20°	35°

At the end of step 2, a location where the code is to be embedded on the product should be selected.

Step 3: Embedding the code in the CAD model

The third step is to modify the CAD model to include the code. A step-by-step approach to embed a QR code in a CAD program is provided. To include the code in a CAD model, it is saved in a DXF format and is scaled to the desired size. In order to create a file that can be imported into a CAD program, the code should be a vector file and the modules should be individual. This is needed to be able to scale the code as a whole and the modules individually. First, this vector file should be scaled to the correct size. This is the code size as shown in Table 7.2. When scaling this the whole code should be scaled, so all objects should be selected and transformed together. Next, to ensure that the sketch does not contain doubly nested contours, the individual modules are set to 99.5% size. Now the code can be saved as a DXF and imported. To embed the code in a CAD program the first step is to create a plane parallel to the average direction of the selected surface. On that plane, the DXF of the code is imported and placed in the correct location. Using the offset from surface option when using the remove material option with a depth of 1.6 mm will result in a code with a carving depth of 1.6 mm.

Because all the modules are 99.5% of their original size, there will be small walls between the modules in the CAD program. Because these are extremely small features, the printer will ignore them. Alternatively, if those small features are unwanted, the individual modules can be scaled to 101% size. This ensures that adjacent negative modules overlap. These modules should be combined into one shape. In Inkscape, this can be done with the combine (Ctrl-K) and join all nodes (Ctrl-J) options. Afterwards, save as DXF14. This resulting DXF will contain doubly nested contours. In the CAD program, one extra step is now required. Each shape has to be selected individually for extrusion. This is the method that was used in the research. The main difference is that selecting each shape takes more time.

After embedding the code in the CAD program the product is ready to be printed and the code should be able to identify the product.

8 DISCUSSION

In this chapter aspects that could be improved on the experiments are discussed. Also, suggestions are made on how to continue this research.

At the start of this project, the aim of this research was to develop the guidelines for fused deposition modelling (FDM). This decision was made for two reasons. At first, design guidelines that will work for FDM will work for most other AM methods. This is because FDM has a lower precision and less design freedom than the other AM methods. Because these other methods can often meet the same specification or better than FDM printing, it would be likely that the guidelines will also work for those methods. In fact, the design guidelines could prove to be more reliable when other methods are used, as they typically have higher precision. Secondly, wide availability of the technology is ensured by developing it for FDM 3D printing as this is the most available and commonly used 3D printing method. Some experiments have been conducted, the results are discussed in Appendix C. However, due to FDM not being able to meet the necessary requirements the switch was made to SLS printing. Moreover, there is more need from the industry for an identification technology to be developed for SLS printing. Also, the embedment of codes on SLS-manufactured products is much more practical than on FDM-printed products. For SLS manufacturing it has no cost to embed the codes, whereas when FDM printing is used the production time increases significantly when codes are embedded. There are other additive manufacturing technologies in the powder bed fusion category for which the embedment of codes onto products is also relevant. As a follow-up step, the validity of the developed guidelines for other powder bed fusion techniques can be evaluated, allowing a diversification in terms of the materials on which codes can be embedded.

At the start of the research, it was not always clear what variation of the parameters was required. This caused some extra samples to be added to a sample set later, these samples were printed in a different batch. There is a quality difference between batches of printed samples. This quality difference can be seen in the base sample, size 19 and CD1.6. Those samples were reprinted after they were destroyed and put under the microscope. The new samples performed significantly better than the old samples. This indicates that there is a quality difference between the batches in which the samples were printed. It is assumed that the difference in quality is caused by the location in the printer and the amount of virgin powder that is used. As seen in the microstructure analysis of the microscope in paragraph 8.1, there is some material building up in the carvings of the old base sample. This could be because the sample could have been longer in the printer, which caused more part growth. Also, in the microscope analysis black dots with a higher molecular mass were found, this can indicate that oxidation and/or dehydration took place. The difference in sample quality per batch affected all sample sets, as several samples were added later, and were thus printed in a different batch. Extra samples were added to sample set 4, and these were printed in a different batch. In sample set 4, the quality difference between sample sets was less clear than in sample sets 1 and 2. The fact that there is a significant quality difference between batches indicates that there might be parameters that have not been fixed. In this case, it is most likely that the amount of virgin powder and the location in the printer varies. Two actions should be carried out: on the one hand, the

samples from a sample set should be reprinted in a single batch to verify the results. Next to that, the cause of the quality difference should be identified and eliminated, as the difference indicates that there are parameters that influence the results that are unaccounted for.

An aspect that has not been researched thoroughly, is the interdependence of the parameters. There might be interdependence between the parameters. However, little experimentation in this area has taken place. Moreover, most of the experiments that did take place failed to yield processable results. In one of the experiments that were conducted in this category, the experiments with a varying carving depth and surface curvature of 50° needs to be redone with a surface angle of 20° and 30° , as the sample with an angle of 50° did not yield any useful results. Size and carving depth are likely to be dependent on one another. Not a lot of experimentation has been done to prove this. It could be that it is better to vary the carving depth based on the size of the modules, however, this is not tested. To find out if the ideal carving depth is dependent on the module size, experiments in which the carving depth and module size are varied should be added. The ratio of size/carving depth can show if there is such a dependence. Next to that, experiments, where the size and carving depth are both varied on a sample with a surface curvature, should be added.

Also, since the recommended carving depth is altered, it could be argued that the experiments need to be re-run with a new base sample. The new base sample should follow the guidelines: a carving depth of 1.6 mm and a module size of 0.7 mm. Other parameters can also be added. Colour could be such a parameter, and some more experimentation with surface modifications that are achievable with the printer resolution could be done.

The achieved resolution in the samples is higher than what was to be expected based on Seepersad et al. [13] and Weiss et al. [22]. It could be that a higher resolution can be achieved reliably. However, during the printing, the samples were placed in a location of the printer where small features are printed relatively well. This might have affected the results, as the achievable resolution might be lower in other parts of the printer. To check this, the sample sets could be placed in a position of the printer that is suboptimal to see if the samples perform according to the expectations.

To summarize, if a second iteration was to be carried out several experiments should be included. The second iteration should use the determined specification for the parameters as the baseline, that being a module size of 0.7 mm and a carving depth of 1.6 mm. All samples in a sample set should be printed in the same batch. Several sample sets should be added. These sample sets should investigate the interdependence of the carving depth and size, carving depth and surface curvature (and size) and adding colour variation. Next to that, the surfaces on which the code is embedded could be placed under a corner to evaluate the decoding capacity if the embedded code is under an angle.

The setup was optimized to maximize the generated contrast in flat samples. It could be the case that the optimal lighting angle is different when samples have curvature. This is something that should be tested. Instead of optimizing the setup for a flat sample, the setup could be optimized for the sample with 20° curvature. It could be possible that in order to process the samples with a higher curvature the script will need less alteration because of the change in setup. So, if there is a continuation of this research the setup should be revised to take the surface curvature of samples into account.

One step that can definitely be improved in the processing script is the binarization step. Instead of minimizing the in-group variance, an algorithm that tries to distinguish two groups with

a different variance and means and tries to fit the data to that would most likely be way better. The samples can also be altered, as the sample shadow was a major cause of noise when trying to binarize. Instead of having a straight wall after 2 mm, a slope can be used. However, real products that will be produced will also cast shadows, so the program that is looking for the code will have to be able to deal with that. It might also be an idea to tailor the binarization algorithm to the process of finding a code with positive and negative modules. The colour of a module can be determined based on the properties of the adjacent modules. This is expected to reduce the error rate that occurs due to negative module adjacency.

Another aspect that can be improved in the script is the way the detection patterns are found. This can be done by adding a step that checks if there is a border around the central square, checking for the 1:1:3:1:1 ratio in a less strict way than the original script provided by MatWorks did. Also, the determination of the alignment pattern can be improved. the location of the alignment pattern is now calculated based on the location of the detection patterns It can also be determined by using a more sophisticated algorithm that actually checks the location and corrects it if it is slightly off. With these changes in place, distortion correction could be included, which can improve the reading capabilities of samples with curved surfaces.

When looking at the results, another aspect that should be considered is the usefulness of including sensitivity and specificity. It does show how much of the code is correctly identified, but it does not really contribute to the scoring. Thus, it is more a result that gives some feel but does not significantly influence the results in any way. It is useful when checking what is happening. This is why the choice was made to include the sensitivity and specificity (and also the percentage of errors per module) only for context.

In the design guidelines, the extra options are given to use a module size of 0.6 mm and embed the code onto a surface with a surface curvature of up to 35° if it can't be avoided. This introduces ambiguity, which is unhelpful for guidelines that are supposed to be clear and short.

In the design guidelines, there are some vague recommendations. It could be better if these were more concrete. The choice was made to not include examples so that the guidelines are shorter. Also, the experiments did not prove what the upper range of the module size is. This is not ideal, but also not problematic. Since the focus is to determine the minimal code size, the upper limit is less interesting. When including the code, it is better if it is as small as possible. Thus, the choice as made to not determine the upper limit, but recommend the lower limit.

The design guidelines should be evaluated to test if the communication is understandable and the guidelines work in practice. To evaluate this the guidelines along with the supporting document should be provided to companies that work with SLS printing, and they should be asked to try to embed codes onto their products. Afterwards, the results will be evaluated in two ways. First and foremost, whether or not the code was successfully embedded is measured. Next to that, qualitative research should be done on the user-friendliness of the design guidelines and the supporting document.

One suggestion that was done by Papp et al. [12] was to use individual modules only. In that way, the error occurrence rate due to module adjacency decreases. It has two disadvantages. At first, all modules are isolated as individual modules which increases the error rate. Secondly, more space is required for the code as the minimal achievable resolution remains the same, but more material is added around the modules. This results in a size increase of the code. Because of the size increase this option was not considered to be useful. However, if the module adjacency causes most problems it is a good option to explore.

A major problem with the resulting guidelines is that they are not automatically embedded. Embedding the codes onto products will cost a lot of engineering time. This step needs to be automated if the technology is ever to be used on a larger scale. The research has provided information that can be used to automate the embedment.

A final aspect that should be researched is how to locate the codes on the products. No attention was paid to that aspect in this research, but it is relevant when the technology is used in practice. This research should come up with some scanning setup that includes a camera and lights and looks for the codes that are embedded onto the product.

When expanding further on this research, the design guidelines can be generalized powder bed fusion processes or specified for other technologies in this category. The main difference is that the other technologies require support structures to cool, but that is not a big problem when embedding codes. The achievable resolution does differ, and the optical properties of polymers and metals are not alike. As a results, some changes to the recommended values of the three parameters should be made. The parameters, remain the same. It is possible to generalize even further, modifying the guidelines to be used in other categories. When doing this directional effects should play a larger role in the considerations as overhang is often problematic.

8.1 Extra investigation of samples

Two unexpected results were investigated more thoroughly. At first, the underperformance of the original base sample is investigated by looking at the modules using a microscope. Secondly, the colour difference between samples from the same batch is looked into.

8.1.1 Investigation the 20 mm/base sample

During the size and carving depth analysis, it became clear that the base sample performed worse than the size 19 (CD 2.0) and carving depth 1.6 samples (size 20). This was strange, as the base sample was expected to outperform those samples. To investigate why this happened, the baseline, size 19 and carving depth 1.6 samples were more closely investigated. At first, the source data was examined to find out where the difference came from and after that the samples were investigated under a microscope.

To find out what the cause of this unexpected result was the histograms of the size 19 and base sample are compared. In Figure 8.1 the histogram of the base sample is shown and in Figure 8.2 the histogram of the size 19 sample is shown.

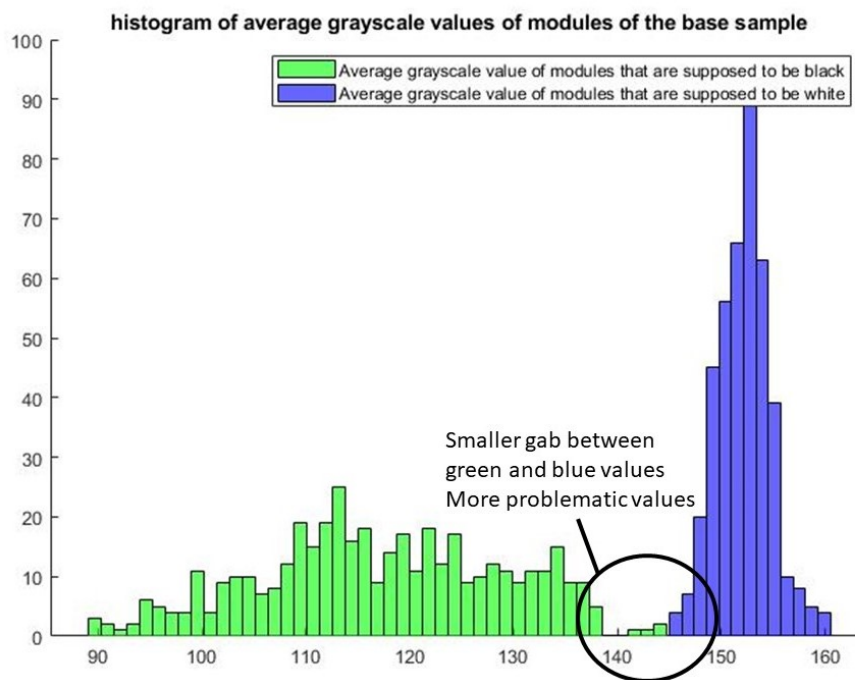


Figure 8.1: Histogram of the average grayscale values of the modules of the base sample.

When comparing these two histograms there are two differences that should be noticed. The gap between the black and white modules is larger for the size 19 sample than for the base sample and there is less overlap between the two groups. Both differences can be seen by looking at the area inside the circle that is present in both figures. In the circle of the histogram of the base sample, it can be seen that four modules that are supposed to be black are closer to the group of white modules. If modules are closer to the other 'wrong' group, it significantly increases the standard deviation, decreasing the number of sigma between the means. The next step is finding out why several black modules in the base sample have a higher average grayscale value.

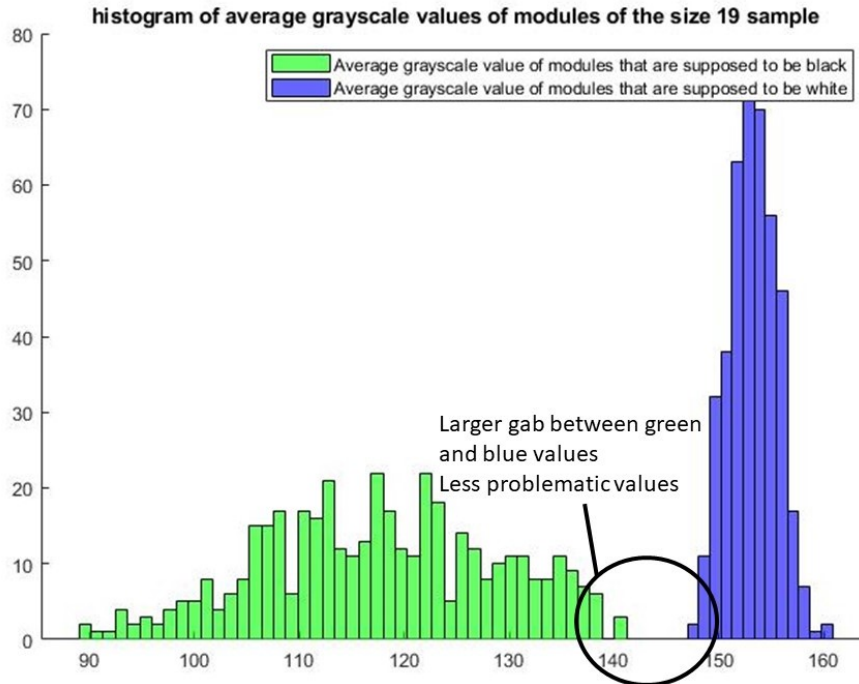


Figure 8.2: Histogram of the average grayscale values of the modules of the size 19 sample.

To find out where the four outliers of the base sample originate, the grayscale image and the image with the difference between the reference code and pixelated code are shown for the base sample in Figure 8.3 and for the size 19 sample in Figure 8.4. It can be seen clearly that the corners of the detection patterns contain more errors in the base sample. These are circled in red. A cause for these errors could be that light is reflected from the walls of the adjacent positive modules, causing increased lighting levels in the corners. From two lighting directions, the corners will be registered as dark, but from the other two directions they are doubly registered as light. Another cause can be that the shadows are not long enough due to negative pixel adjacency, causing no shadow to be cast in these corner modules. Also, the directional effects of the printer play a role, which resulted in more part growth in the height direction.

It is expected that the orientation of the sample with respect to the lighting directions causes this difference. The orientation of the specimen compared to the lighting direction can heavily influence the results, as at a slight angle the light no longer falls directly across the channels of the detection pattern. This diminishes the effect of the double light reflection. To test if this is the effect both samples are placed so that the lighting angles are exactly 90° and they are also placed and processed under a slightly different angle. This did not change the results. It can thus be concluded that the difference is not caused by the lighting angle but must have to do with the specimen itself. A possible cause for this difference could be that the base sample was located at the bottom of the printing volume while the 19 sample was at the top, causing part growth to play a more significant role in the base sample. To check if part growth played a more significant role the base sample, size 19 sample and carving depth 1.6 sample are placed under the microscope.

In the Figures 8.5 and 8.6 holes with the width of one module of the base sample are shown. In the Figures 8.7 and 8.8 holes with the width of one module of the carving depth 1.6 sample (S20,CD1.6) are shown and in Figures 8.9 and 8.10 holes with a width of one module from a sample with size 19 (S19,CD2.0) are shown. Aspects that should be noted are marked.

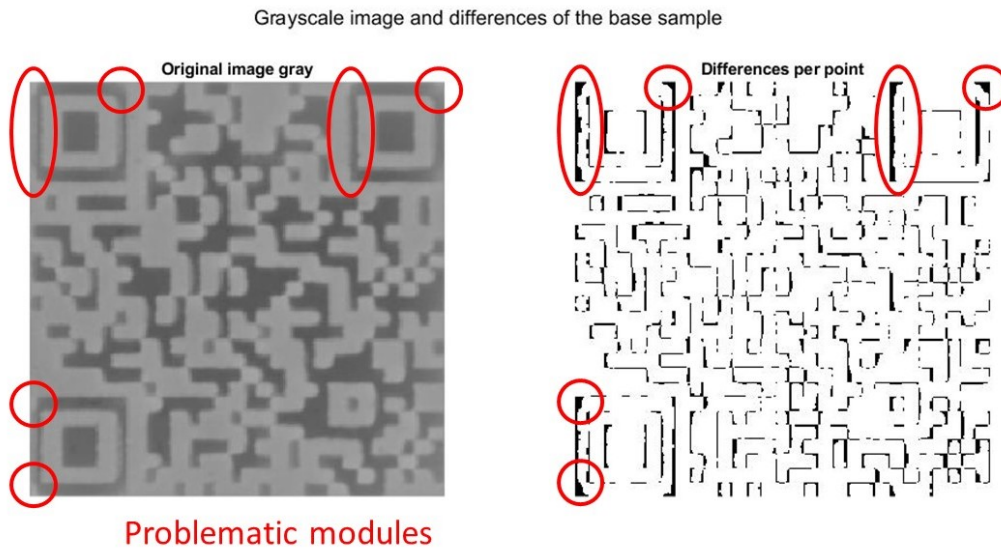


Figure 8.3: Grayscale image and differences compared to the reference of the base sample.

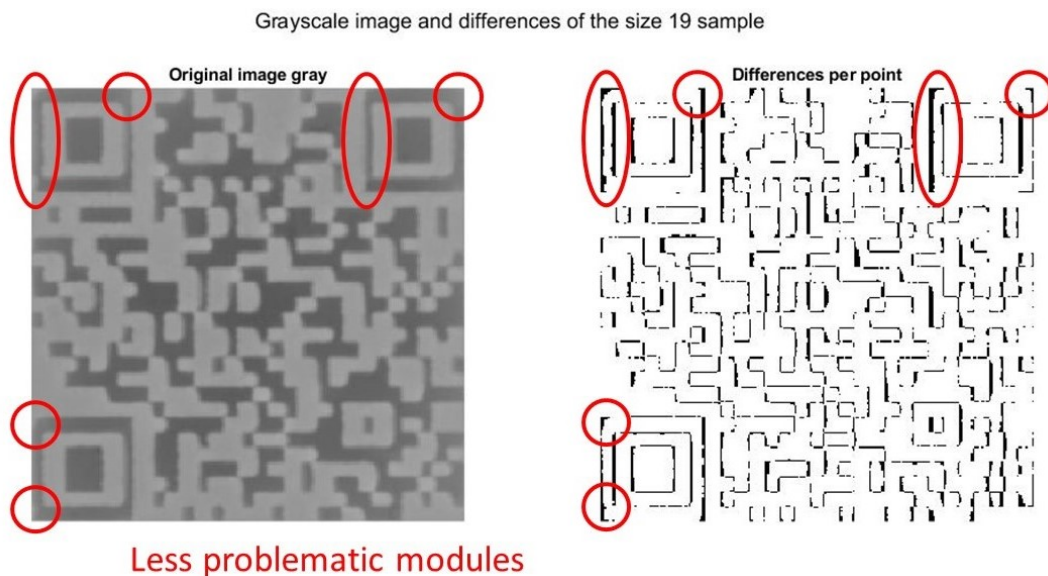


Figure 8.4: Grayscale image and differences compared to the reference of the size 19 sample.

Several things stand out about the base sample. All modules are supposed to be of the same width. As can be seen in the Figures, this is not the case. The hole in Figure 8.6 is wider than Figure 8.5. In Figure 8.5 the width of a negative module (A) and positive module (B) are marked. It is clear that the positive module is wider. Closer inspection seems to indicate that the width of the hole shown in Figure 8.6 is approximately 0.6 mm, while Figure 8.5 seems to have a width of around 0.4 mm. The width of the modules is supposed to be 0.69 mm, so it can be concluded that all negative modules are too small.

There is some build-up of powder in the bottom of the holes, in Figure 8.6 this can be seen most clearly. The loose clogged material is circled. The powder is the porous area at the bottom of the hole. As a result, the hole is more shallow than intended. In Figure 8.5 powder build-up can be seen as well, if the region C and D are compared, it is clear that there is more powder

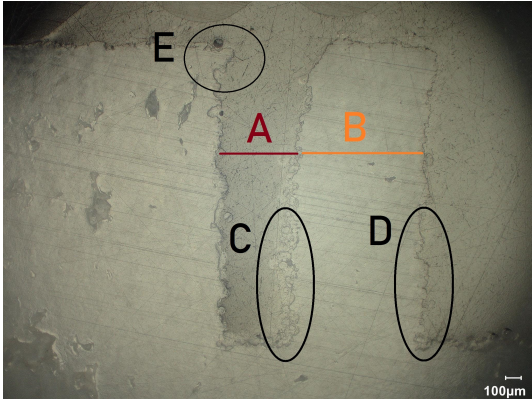


Figure 8.5: A hole with a width of one module in the original base sample with the difference in module with (A & B), differences in powder build-up (C & D) and a shift in the top layer (E) marked.

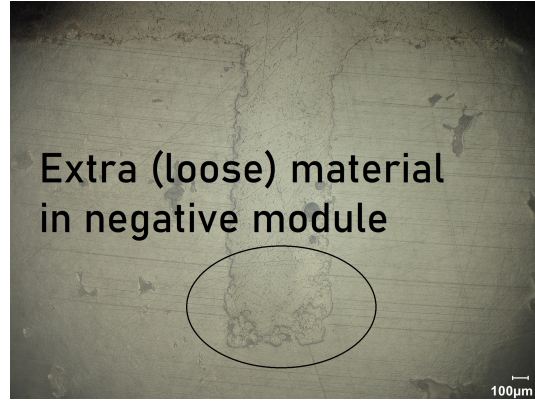


Figure 8.6: A hole with a width of one module in the original base sample with extra loose material that sticks in the negative modules pointed out.

build-up in C. This is because the negative module is smaller there, so the semi-loose powder is harder to remove. Part growth and directional effects can also play a role in this powder build-up on the side of the walls.

The top layers seem to be dealigned with the bottom layers. This can be seen most clearly in Figure 8.5 where it is indicated with the letter E. The top of the hole seems to be in a different place than the bottom hole, there is an overhang above the hole. From a top view, this reduces the size of the deep zone, causing the black modules to appear smaller than they were meant to be. This can cause the performance of the sample to go down. This shift in top layers can be the cause of the lower performance of the base sample.

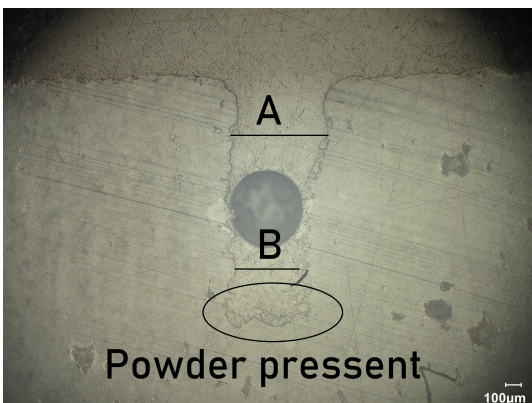


Figure 8.7: A hole with a width of one module in the sample with a carving depth of 1.6 mm with powder in the bottom of the module and a varying module width.

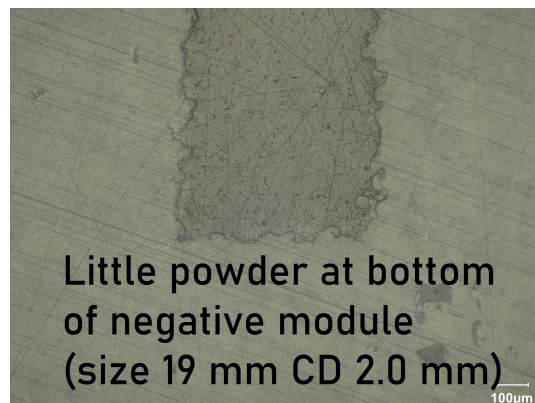


Figure 8.8: A hole with a width of one module in the sample with a carving depth of 1.6 mm with little powder adhesion.

The Figures of the holes of the sample with a carving depth of 1.6, 8.7 and 8.8, are compared to each other and to the base sample. The width of the negative modules is measured using the measuring scale in the image. Again, there is a difference in size. Figure 8.7 has a width of 0.5 mm at the top and 0.4 mm at the bottom. The width of the hole in Figure 8.8 is 0.55 mm. Compared to the base samples less powder build-up is seen. In Figure 8.7 some powder

build-up can be seen, Figure 8.8 is about as good as it gets in terms of powder build-up at the bottom of a carve. In Figure 8.7 it can be seen that the top of the hole seems wider than the bottom. Size A is not equal to size B. Compared to the base sample the overhang is missing, so the whole hole can be seen from the top view. These two observations help explain the difference between the base sample and this sample, and explain why this sample outperformed the base sample. Being able to see the whole hole and wider negative modules are the main contributors to the difference.

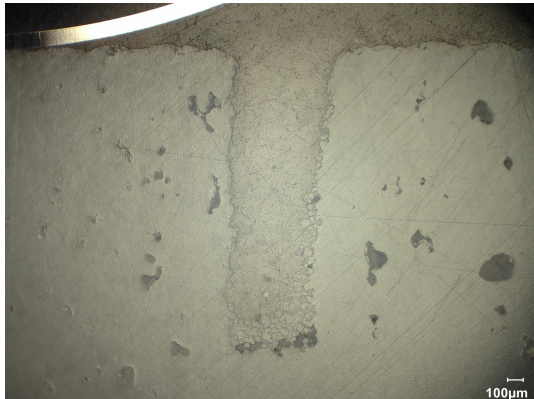


Figure 8.9: A hole with a width of one module in the size 19 mm sample with powder grains at the bottom of the negative module.



Figure 8.10: A hole with a width of one module in the size 19 mm sample without at the bottom of the negative module.

Figure 8.9 and 8.10 are images of negative modules in the size 19 sample. These figures nicely show the difference between powder build-up and no powder build-up. In Figure 8.9 it can be seen that there are a lot of small particles at the bottom and the side of the hole, while in Figure 8.10 this is not the case. With a width of 0.55 the negative modules are of the same size and wider than the base sample.

It should be noted that the width measurements are done with a ruler that was put to a screen, and some accuracy is missing. This was seen most clearly in the size 19 sample. When measuring the positive module size a size of 0.6 mm was measured vs 0.55 mm of the negative module. Clearly, a measuring error is made, as both are supposed to be 0.66 mm.

It can be concluded that the original base sample had a smaller negative module width which caused the performance to decrease. This is a combined result of incomplete removal of material, part growth and a shift in the top layer.

8.1.2 Colour difference between samples in the same batch

There are some strange results where samples from the same batch had a significantly different colour. Because colour affects the generated contrast, it is investigated what caused this colour change. These samples were placed under the microscope to see what was going on. Three samples are investigated using the microscope. These were the darkest, one medium dark and the lightest sample. The microstructure of the darkest sample is shown in Figure 8.11, that of the medium sample is shown in 8.12 and Figure 8.13 shows the light sample.

From the Figures, it can be seen that in the darkest sample, there are more black dots. The black dots most likely cause the colour to change. The more black dots, the darker the sam-



Figure 8.11: Micro structure of the darkest sample. Figure 8.12: Micro structure of a medium dark sample. Figure 8.13: Micro structure of the lightest sample.

ple. It is strange that there is a difference between samples from the same sample set, as the powder is expected to be uniform. To find out what the black dots are, the darkest and lightest samples are placed in an electron microscope. The microscope that is used is a scanning electron microscope (SEM). The image of the electron microscope of the darkest sample can be seen in Figure 8.14, and the lightest in Figure 8.15.

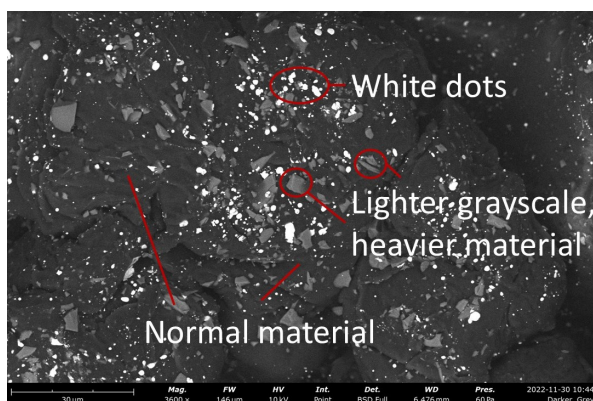


Figure 8.14: Image from the SEM of the darkest sample.

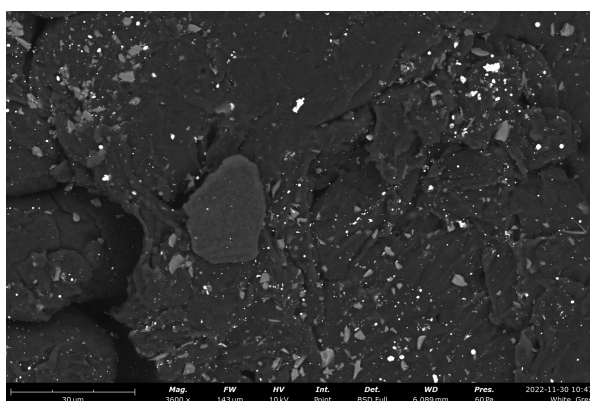


Figure 8.15: Image from the SEM of the lightest sample.

What is seen in these Figures is the amount of electron scattering that each part of the surface causes. Heavier molecules cause more scattering, resulting in a lighter grayscale in the image. So the lighter dots indicate that there is a heavier material present. The most common colour is polyamide 12 which is the material that is used for production. The two other components that are present, the white dots and the lighter units need to be discussed. It seems like the image of the darkest sample contains more light gray spots than the image of the lightest sample.

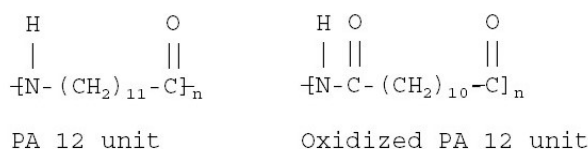


Figure 8.16: Chemical structure of a PA12 unit and an oxidized PA12 unit.

The lighter grey dots are most likely oxidized PA12. The chemical structure changes, which can change the optical properties of a material. To explain what happens during oxidation in Figure 8.16 a PA12 unit and an oxidized PA12 unit are shown. According to Yang et al. the C-H bonds

next to the nitrogen can easily be oxidized [25]. This is thermal oxidation, which despite the chamber being filled with nitrogen still takes place. Next to this change in chemical structure, the molecular chain is also changed due to oxidation. To see what oxidation would look like and if the lighter areas could be oxidation, some other examples of what oxidation looks like were compared. Zappiello et al. [52] use an SEM to capture oxidized and non-oxidized carbon black, as shown in Figure 8.17. The SEM images have a different zoom, Zappiello magnifies 10x more. However, based on the decolourization of the effect it seems likely that the lighter areas in the sample are indeed oxidized. One observation that contradicts the idea that the lighter material is oxidized PA12 is the fact that it has sharp edges. Typically this is something that would be expected in a ceramic material, and not in a polymer. If the chemical build-up of the material changes, the optical properties can also change. Dabdakhsh et al. claim that TiO_2 and silica are added to PA12 to increase whiteness [24], if this TiO_2 contributes to the oxidation that might cause a decrease in whiteness. Dehydration can also be taking place. In dehydration, the length of the polymer chain increases. If two chains fuse it has fewer ends (2 vs 4). The H and OH molecules that are at the end of each chain are separated as water [25].

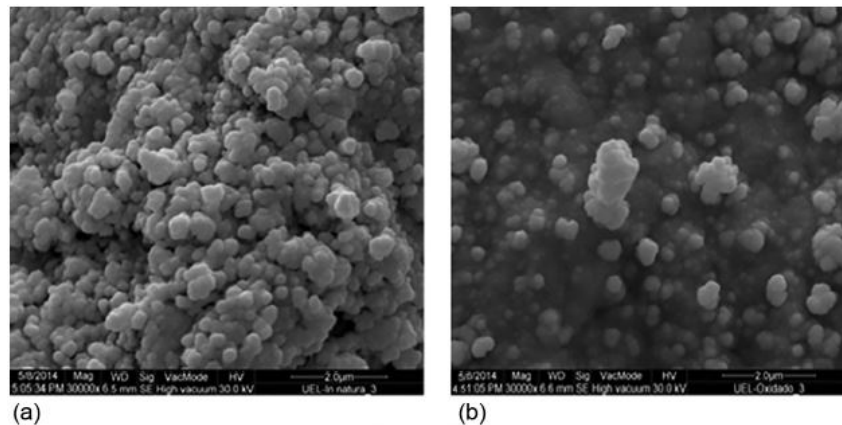


Figure 8.17: Scanning electron microscope (SEM) images of (a) carbon black (CB); (b) oxidized CB from [52].

Dabdakhsh et al. [24] and Chen et al. [23] observe crackling as a result of ageing in PA12. As a result, small particles emerge around the grains. The particle size can affect the optical properties of the grain. It could be that because of the increased surface area oxidation is more likely to happen so the two possible explanations complement each other.

The other aspect to address about the SEM images are the white dots. It seems that the dark sample, Figure 8.14 contains more white dots than the light sample, Figure 8.15. On a forum discussing these white spots Ravish Kumar Jain claimed that these white dots are most likely particulate matter that adhered to the surface [53]. This is probable as the samples were not cleaned before being put in the SEM. He adds that it could also be that during the deposition a high deposition rate of the electrons or arcing the plasma caused these white dots to appear.

9 CONCLUSION

It can be concluded that this research forms a first step towards developing design guidelines for embedding 2D matrix codes onto SLS printed products. The most important parameters are identified and their effects on the embedment success are tested. The product parameters that are identified are the module size, carving depth, surface curvature and surface textures. Next to those parameters, the lighting conditions and processing scripts are controlled. Based on experiments the ideal values for those parameters have been determined. The minimal module size is 0.7 mm. If little space is available, a module size of 0.6 mm can be used but this results in a performance decrease. For the carving depth, a value of 1.6 mm is ideal. The maximum permissible surface curvature without a performance loss is 20°, while a surface curvature of up to 35° can be decoded with a drastic loss in performance. In Table 9.1 the values of the parameters are shown. Surface textures have no significant effect on the results. Using these guidelines 2D matrix barcodes can be embedded onto SLS printed products. More research can be done to make the position options of the code placement more flexible. Also, the communication of the guidelines can be evaluated on a larger scale.

Table 9.1: The design guidelines for embedding 2D matrix barcodes onto SLS printed products.

Parameter	Recommended value	Lower performance value
Module size	0.7 mm or higher	0.6 mm
Carving depth	1.6 mm	-
Maximum surface curvature	20°	35°

Acknowledgements

I would like to thank Tom Vanaker and Robbert Jan Torn for their supervision during the project. Also, I would like to thank Quint Meinders who printed my samples and Nick Helthuis who allowed me to help investigate my samples under the microscope. Next to that, I want to thank the people from the workshop and the Student Laboratories Twente (SLT) that allowed me to borrow materials for much longer than I asked initially.

BIBLIOGRAPHY

- [1] Michael Schmidt, Marion Merklein, David Bourell, Dimitri Dimitrov, Tino Hausotte, Konrad Wegener, Ludger Overmeyer, Frank Vollertsen, and Gideon N Levy. Laser based additive manufacturing in industry and academia. *Cirp Annals*, 66(2):561–583, 2017.
- [2] Fei Chen, Jaime Zabalza, Paul Murray, Stephen Marshall, Jian Yu, and Nikhil Gupta. Embedded product authentication codes in additive manufactured parts: Imaging and image processing for improved scan ability. *Additive Manufacturing*, 35:101319, 2020.
- [3] Manfred Schmid, Antonio Amado, and Konrad Wegener. Polymer powders for selective laser sintering (sls). In *AIP Conference proceedings*, volume 1664, page 160009. AIP Publishing LLC, 2015.
- [4] Julien Gardan. Additive manufacturing technologies: state of the art and trends. *Additive Manufacturing Handbook*, pages 149–168, 2017.
- [5] Tom Vaneker, Alain Bernard, Giovanni Moroni, Ian Gibson, and Yicha Zhang. Design for additive manufacturing: Framework and methodology. *CIRP Annals*, 69(2):578–599, 2020.
- [6] Fei Chen, Yuxi Luo, Nektarios Georgios Tsoutsos, Michail Maniatakos, Khaled Shahin, and Nikhil Gupta. Embedding tracking codes in additive manufactured parts for product authentication. *Advanced Engineering Materials*, 21(4):1800495, 2019.
- [7] Wei Gao, Yunbo Zhang, Devarajan Ramanujan, Karthik Ramani, Yong Chen, Christopher B Williams, Charlie CL Wang, Yung C Shin, Song Zhang, and Pablo D Zavattieri. The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design*, 69:65–89, 2015.
- [8] John Ryan C Dizon, Alejandro H Espera Jr, Qiyi Chen, and Rigoberto C Advincola. Mechanical characterization of 3d-printed polymers. *Additive Manufacturing*, 20:44–67, 2018.
- [9] Lauren EJ Thomas-Seale, Jackson C Kirkman-Brown, Moataz M Attallah, Daniel M Espino, and Duncan ET Shepherd. The barriers to the progression of additive manufacture: Perspectives from uk industry. *International Journal of Production Economics*, 198:104–118, 2018.
- [10] Ryosuke Kikuchi, Sora Yoshikawa, Pradeep Kumar Jayaraman, Jianmin Zheng, and Takashi Maekawa. Embedding qr codes onto b-spline surfaces for 3d printing. *Computer-Aided Design*, 102:215–223, 2018.
- [11] Hao Peng, Lin Lu, Lin Liu, Andrei Sharf, and Baoquan Chen. Fabricating qr codes on 3d objects using self-shadows. *Computer-Aided Design*, 114:91–100, 2019.
- [12] György Papp, Miklós Hoffmann, and Ildikó Papp. Improved embedding of qr codes onto surfaces to be 3d printed. *Computer-Aided Design*, 131:102961, 2021.

- [13] Carolyn Connor Seepersad, Tyler Govett, Kevin Kim, Michael Lundin, and Daniel Pinero. A designer's guide for dimensioning and tolerancing sls parts. In *2012 International Solid Freeform Fabrication Symposium*. University of Texas at Austin, 2012.
- [14] Hubs. What is fdm (fused deposition modeling) 3d printing? <https://www.hubs.com/knowledge-base/what-is-fdm-3d-printing/>, accessed 22-08-2022.
- [15] Samuel Clark Ligon, Robert Liska, Jurgen Stampfl, Matthias Gurr, and Rolf Mülhaupt. Polymers for 3d printing and customized additive manufacturing. *Chemical reviews*, 117(15):10212–10290, 2017.
- [16] Why 3D Printing is the answer for short series production. <https://medium.com/extreme-engineering>. <https://medium.com/extreme-engineering/why-3d-printing-is-the-answer-for-short-series-production-9f7ffe99e4e>, accessed 22-08-2022.
- [17] Yi Zhang, William Jarosinski, Yeon-Gil Jung, and Jing Zhang. Additive manufacturing processes and equipment. In *Additive manufacturing*, pages 39–51. Elsevier, 2018.
- [18] Ian Gibson, David Rosen, and Brent Stucker. *Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing, second edition*. 01 2015.
- [19] Kalim Deshmukh, Aqib Muzaffar, Tomáš Kovářík, Tomáš Křenek, M Basheer Ahamed, and SK Khadheer Pasha. Fundamentals and applications of 3d and 4d printing of polymers: Challenges in polymer processing and prospects of future research. In *3D and 4D Printing of Polymer Nanocomposite Materials*, pages 527–560. Elsevier, 2020.
- [20] Yusheng Shi, Chunze Yan, Yan Zhou, Jiamin Wu, Yan Wang, Shengfu Yu, and Ying Chen. Chapter 2 - polymer materials for additive manufacturing—powder materials. In Yusheng Shi, Chunze Yan, Yan Zhou, Jiamin Wu, Yan Wang, Shengfu Yu, and Ying Chen, editors, *Materials for Additive Manufacturing*, 3D Printing Technology Series, pages 9–189. Academic Press, 2021.
- [21] Shoujin Sun, Milan Brandt, and MJLAM Easton. Powder bed fusion processes: An overview. *Laser additive manufacturing*, pages 55–77, 2017.
- [22] Benjamin Weiss, Olaf Diegel, Duane Storti, and Mark Ganter. A process for estimating minimum feature size in selective laser sintering. *Rapid Prototyping Journal*, 2018.
- [23] Peng Chen, Hongzhi Wu, Wei Zhu, Lei Yang, Zhaoqing Li, Chunze Yan, Shifeng Wen, and Yusheng Shi. Investigation into the processability, recyclability and crystalline structure of selective laser sintered polyamide 6 in comparison with polyamide 12. *Polymer Testing*, 69:366–374, 2018.
- [24] Sasan Dadbakhsh, Leander Verbelen, Olivier Verkinderen, Dieter Strobbe, Peter Van Puyvelde, and Jean-Pierre Kruth. Effect of pa12 powder reuse on coalescence behaviour and microstructure of sls parts. *European Polymer Journal*, 92:250–262, 2017.
- [25] Feifei Yang, Navid Zobeiry, Ramulu Mamidala, and Xu Chen. A review of aging, degradation, and reusability of pa12 powders in selective laser sintering additive manufacturing. *Materials Today Communications*, page 105279, 2022.
- [26] Fredrick M Mwanja, Maina Maringa, and Kobus van der Walt. Mixing and reuse of polymer laser sintering powders to ensure homogeneity—a review.

- [27] Silvia Vock, Burghardt Klöden, Alexander Kirchner, Thomas Weißgärber, and Bernd Kieback. Powders for powder bed fusion: a review. *Progress in Additive Manufacturing*, 4(4):383–397, 2019.
- [28] György Papp, Miklós Hoffmann, and Ildikó Papp. Embedding qr code onto triangulated meshes using horizon based ambient occlusion. In *Computer Graphics Forum*. Wiley Online Library, 2021.
- [29] Comprehensive Guide to 1D and 2D barcodes. www.dynamsoft.com. <https://www.dynamsoft.com/blog/insights/the-comprehensive-guide-to-1d-and-2d-barcodes/#Aztec-Code>, accessed 05-04-2022.
- [30] Types of Barcodes: Choosing the Right Barcode. www.scandit.com. <https://www.scandit.com/blog/types-barcodes-choosing-right-barcode/>, accessed 05-04-2022.
- [31] Jingru Yang, Hao Peng, Lin Liu, and Lin Lu. 3d printed perforated qr codes. *Computers & Graphics*, 81:117–124, 2019.
- [32] Hao Peng, Peiqing Liu, Lin Lu, Andrei Sharf, Lin Liu, Dani Lischinski, and Baoquan Chen. Fabricable unobtrusive 3d-qr-codes with directional light. In *Computer Graphics Forum*, volume 39, pages 15–27. Wiley Online Library, 2020.
- [33] Yves Van Gennip, Prashant Athavale, Jérôme Gilles, and Rustom Choksi. A regularization approach to blind deblurring and denoising of qr barcodes. *IEEE Transactions on Image Processing*, 24(9):2864–2873, 2015.
- [34] Chung-Hua Chu, De-Nian Yang, Ya-Lan Pan, and Ming-Syan Chen. Stabilization and extraction of 2d barcodes for camera phones. *Multimedia systems*, 17(2):113–133, 2011.
- [35] K Saranya, RS Reminaa, and S Subhitsha. Modern applications of qr-code for security. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pages 173–177. IEEE, 2016.
- [36] An overview of QR code error correction feature. <https://www.qrcode-tiger.com>. <https://www.qrcode-tiger.com/qr-code-error-correction>, accessed 05-04-2022.
- [37] Type one and two QR code. Qrcode.com. <https://www.qrcode.com/en/about/version.html>, accessed 02-12-2021.
- [38] Micro QR code. Qrcode.com. <https://www.qrcode.com/en/codes/microqr.html>, accessed 02-12-2021.
- [39] iQR code. Qrcode.com. <https://www.qrcode.com/en/codes/iqr.html>, accessed 02-12-2021.
- [40] Logo QR code. denso-wave.com. <https://www.denso-wave.com/en/system/qr/product/frame.html>, accessed 02-12-2021.
- [41] Secure QR code. denso-wave.com. <https://www.denso-wave.com/en/system/qr/product/sqrc.html>, accessed 02-12-2021.
- [42] A comparison between DataMatrix and QR codes. <https://www.qrcode-tiger.com>. <https://www.qrcode-tiger.com/data-matrix-code-vs-qr-code>, accessed 05-05-2022.
- [43] Chapter 26. Datamatrix (2D-Barcode). <https://jpgraph.net/download/manuals/chunkhtml/ch26.html>. <https://jpgraph.net/download/manuals/chunkhtml/ch26.html>, accessed 05-05-2022.

- [44] MaxiCode 2D Barcode Specification. <https://www.barcodefaq.com/2d/maxicode/>. <https://www.barcodefaq.com/2d/maxicode/>, accessed 05-05-2022.
- [45] Technical Specifications Aztec. <https://www.barcodebakery.com/en/docs/php/barcode/aztec/specification> <https://www.barcodebakery.com/en/docs/php/barcode/aztec/specification>, accessed 05-05-2022.
- [46] Gonzalo J Garateguy, Gonzalo R Arce, Daniel L Lau, and Ofelia P Villarreal. Qr images: optimized image embedding in qr codes. *IEEE transactions on image processing*, 23(7):2842–2853, 2014.
- [47] Xiaochao Li, Zhifeng Shi, Donghui Guo, and Shan He. Reconstruct algorithm of 2d barcode for reading the qr code on cylindrical surface. In *2013 international conference on anti-counterfeiting, security and identification (ASID)*, pages 1–5. IEEE, 2013.
- [48] Marc Alexa and Wojciech Matusik. Images from self-occlusion. In *Proceedings of the International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, pages 17–24, 2011.
- [49] Kuen-Tsair Lay, Lee-Jyi Wang, Pei-Lun Han, and Yi-Sheen Lin. Rectification of images of qr codes posted on cylinders by conic segmentation. In *2015 IEEE international conference on signal and image processing applications (ICSIPA)*, pages 389–393. IEEE, 2015.
- [50] MatWorks. im2gray function documentation. https://www.mathworks.com/help/matlab/ref/im2gray.html#mw_a3f3a58a-3c5c-4a1d-9541-51a6da9b2dea, accessed 05-10-2022.
- [51] Elham Shirazi, Joris Lemmens, Mohammed Gofran Chowdhury, Arttu Tuomiranta, Francky Catthoor, Eszter Voroshazi, and Ivan Gordon. Cloud detection for pv power forecast based on colour components of sky images. In *2021 IEEE 48th Photovoltaic Specialists Conference (PVSC)*, pages 2389–2391. IEEE, 2021.
- [52] Caroline D Zappiello, Daniel M Nanicuacua, Walter NL dos Santos, Daniel LF da Silva, Luiz H Dall’Antônia, Fernanda M de Oliveira, Débora N Clausen, and César RT Tarley. Solid phase extraction to on-line preconcentrate trace cadmium using chemically modified nano-carbon black with 3-mercaptopropyltrimethoxysilane. *Journal of the Brazilian Chemical Society*, 27:1715–1726, 2016.
- [53] Ravish Jain. Researchgate forum, 07 2019. https://www.researchgate.net/post/What_are_these_dark_and_bright_spots_of_SiO2_deposition_layer_on_the_SEM_image_corresponds_Dust_or_pinhole, accessed 31-01-2022.

A PLAGIARISM REPORT

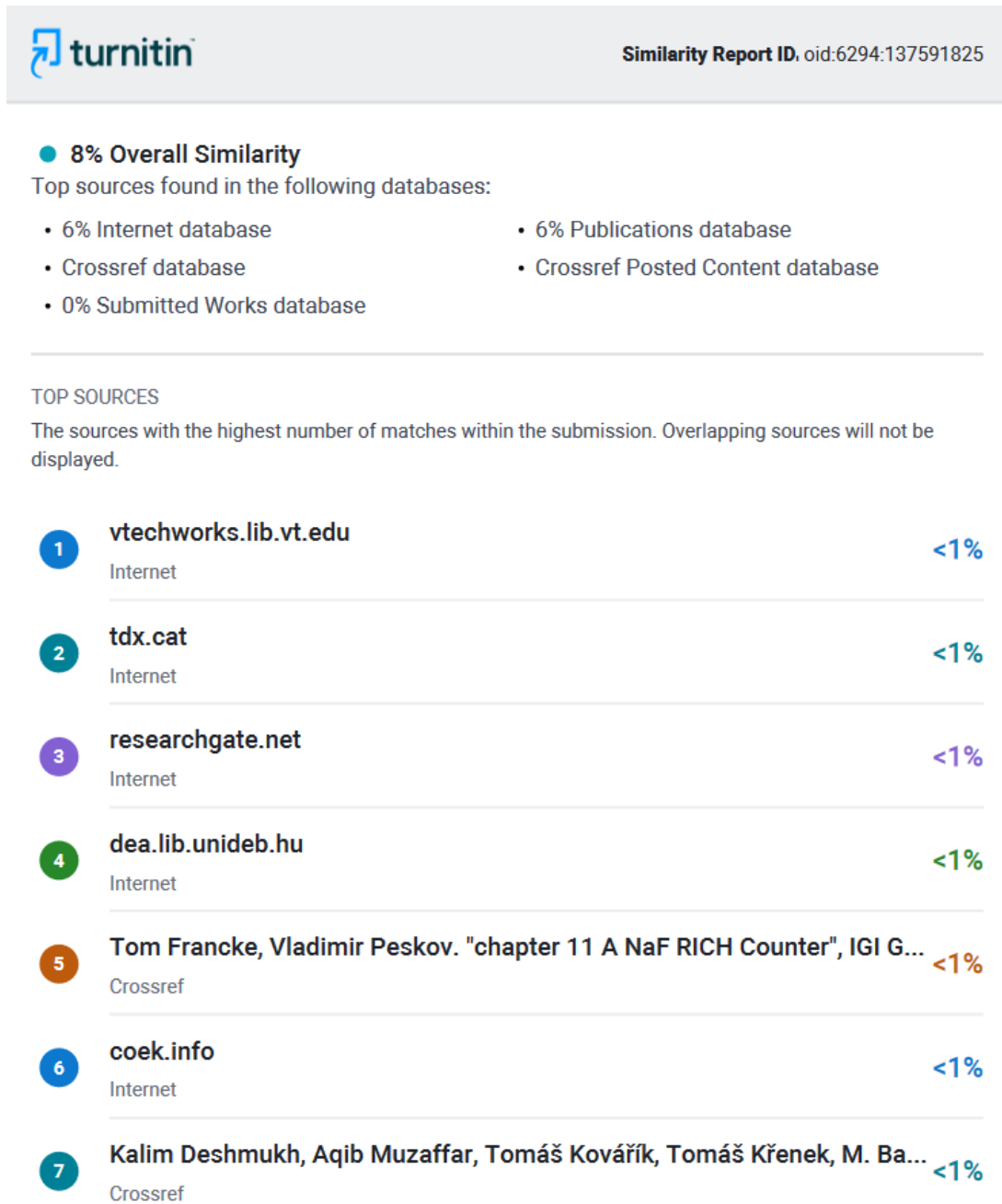


Figure A.1: The plagiarism report as produced by turntin.

B TERMINOLOGY

It is attempted to use consistent terminology throughout the report. Important words that are used are tabulated here to resolve ambiguity that may arise from overlapping terms.

Table B.1: Commonly used words and their meaning in this report.

Code	with code, the QR code that is embedded is meant. Code can normally also refer to a script, but care is taken to refer to this as the script.
Script	Refers to the processing script.
Module	A single pixel of the code.
Positive module	A white module, high grayscale value in Matlab
Negative module	A black module, low grayscale value in Matlab
Pixel	A pixel in an image, typically of a sample. A module consists of multiple pixels.
Sample	A printed piece of material with a code with certain values for each parameter.
Module size	The size of a module, an embedment parameter.
Carving depth	The depth of negative modules as embedded, an embedment parameter.
Surface curvature	The curvature of the surface, an embedment parameter.
Pattern (inclusion)	A small modification in a negative module, an embedment parameter.
Sigma between modules	The number of standard deviations between the means of the grayscale values of the modules that are supposed to be black/white using the average module grayscale value to calculate the standard deviations. This is the main scoring indication.
Sigma between pixels	The number of standard deviations between the grayscale values of the pixels that are supposed to be black/white using the individual pixel grayscale values to determine the standard deviations.
PA12	The type of powder that is used.
PA2200	The commercial name of the powder that is used.

C FUSED DEPOSITION MODELING EXPERIMENTS

At the start of the research, the focus was on fused deposition modelling (FDM). There were several reasons for this. At first, this was the technology that the previous research was focused on so when adding to the previous research it would make sense to use the same technology. Also, the idea was that if guidelines are available for a technology that achieves a lower resolution and has more limitations, it is easier to apply and modify those guidelines for other technologies. They will work for more AM methods and thus are more universally applicable. Next to that, the technology would be available for common user-grade 3D printers, which increases the availability of the technology. If a code is embedded on a product that is printed using FDM the production time increases.

The production method was switched to selective laser sintering (SLS). At first, there is more of a use case from an industrial perspective. SLS is able to print products at a higher precision, and for automatization, it is needed to sort the product from the build volume. Next to that, the resolution that was strived for was not achieved using FDM printing. However, attempts were made to achieve this resolution.

The printers that were used in these early experiments were an ultimaker 2+ and 3 with a nozzle diameter of 0.4 mm. Influencing parameters are identified. The parameters that are important when FDM printing are the filament material, the nozzle size and the slicer settings. The slicer settings determine the layer thickness, printing speed, whether or not there is a z-hop and retraction and the temperature of the nozzle.

The standard settings of the printer did not give satisfactory results. The samples were not printed according to the model specifications. The goal was to print small QR codes with precision and consistency. Stringing was a problem, as this caused the material to be dispensed in areas where there was not supposed to be material. This resulted in false positive modules, resulting in an incorrect code. Oozing was also a problem, were mater

To be able to print more the precise slicer settings were varied. The layer thickness, printing speed, retraction, z-hop and nozzle temperature were varied to reduce the stringing. The idea of raising the nozzle temperature was that maybe the problem was that the material was too cold which made it too viscous and as a result increased the stringing. The idea of decreasing the nozzle temperature was that the material would reach a solid state sooner and thus stop flowing from the nozzle and from the build volume, reducing both oozing and stringing. This did not work. The layer thickness was lowered for more precision. The printing speed was lowered for more precision although this may have had an opposite effect. Retraction and a z-hop were enabled to reduce strining.

The stringing is not avoided. Because of the turning on and off of the printer head the desired accuracy can't be reached. In the end it was not deemed feasible to reach a module size of 0.7 mm.

D EXTRA PLOTS SETUP

In this appendix, a few extra plots are shown. Initially, the sigma over height and distance was investigated. However, due to the clear dependence of the generated contrast on the lighting angle this was left out of the report.

In Figure D.2 the number of standard deviations over the height is shown, and in Figure D.1 the number of standard deviations over the distance is shown.

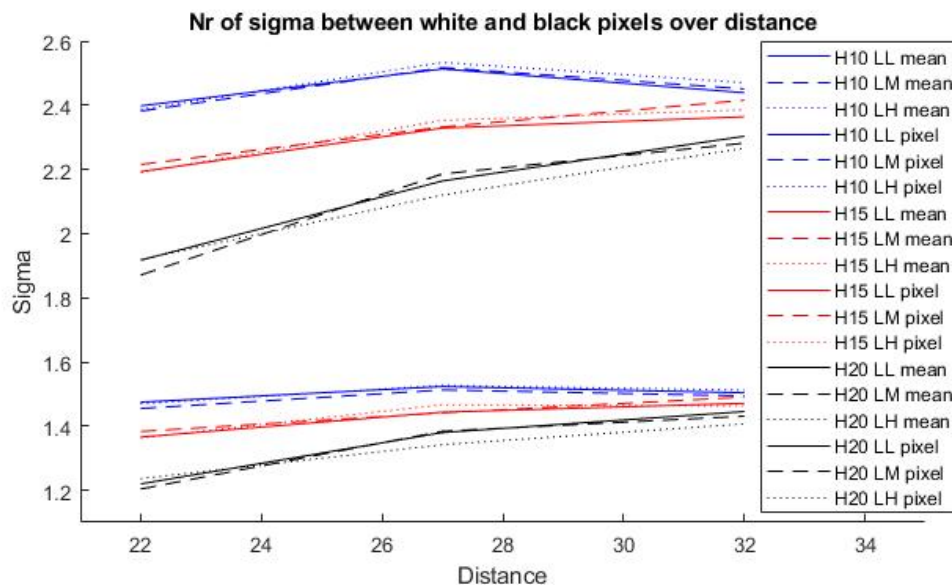


Figure D.1: Number of standard deviations over the distance

In Figure D.1 the number of sigma (standard deviations) between the black and white pixels is on the y-axis and the distance of the standards from the centre point where the sample is placed. The different colours represent the different heights at which the lamp can be placed. The different lining patterns, solid, striped and dotted, represent the different lighting intensities that are used. Abbreviations are used in the legend, if the lighting source is at a height of 10 cm, H10 is used. Vice versa with 15 cm (H15) and 20 cm (H20). The two capital letters behind those numbers indicate the lighting intensity that was used. LL stands for lighting low, with means a lux of 1000. LM stands for lighting medium, which corresponds to a lux of 2000. With LH lighting high is meant, typically with a lux of 4000. It should be noted that the legend contains both the standard deviations between means and between pixels. The same linestyle was used because the conditions are the same, only a different value is used. The standard deviation between means is always larger, so the 9 lines with higher values are the means. It can be argued that only the means should be included. [check with chapter 4] From the plot, it can be seen that as the distance of the lighting source increases, so does the number of sigmas. When the lighting source is at H15 and H20 the sigma increases as the distance increases. At height 10 the sigma seems to peak at a distance of 20 cm. Based on this figure it seems that the light intensity does

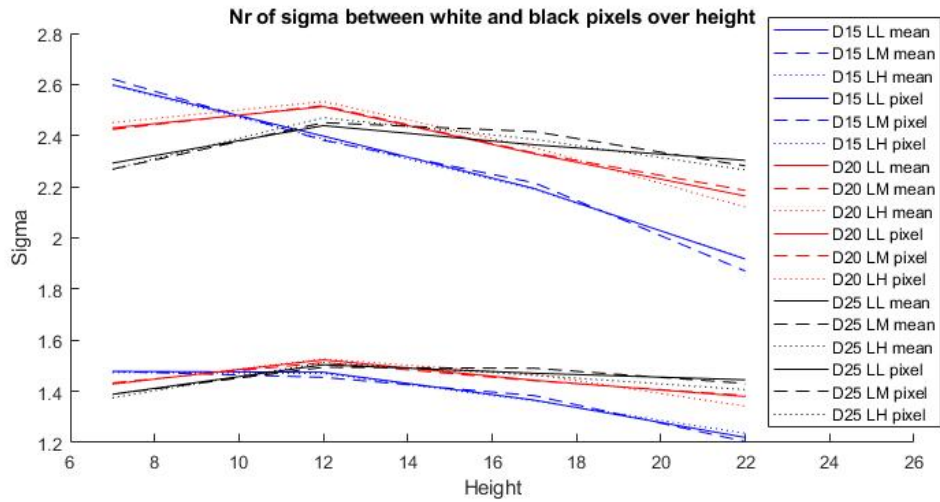


Figure D.2: Number of standard deviations over the height

not have a significant impact on the results. This is because the lines of the same colour show similar behaviour, while the line style does not. This indicates that the lighting intensity has a much smaller impact than the height.

In Figure D.2 the number of sigma that is generated is shown on the y-axis and the height is shown on the x-axis. The plot seems to indicate that as the height increases, the standard deviations between the black and white modules decrease. Again, this plot seems to confirm that the lighting intensity does not play a major role in the number of sigma that is generated. The highest sigma seems to be found at a height of 12 cm for all distances. It is also interesting that the larger the distance the slower this value decreases. This supports the angle theory, since if the distance is larger the change of angle is less if the height increases.

In Figure D.3 the specificity as a function of the height is shown and in D.4 the specificity as a function of the distance is shown.

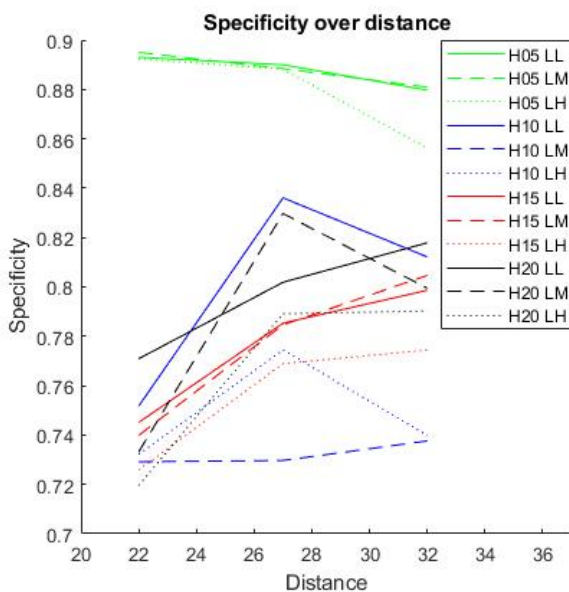


Figure D.3: Specificity over distance

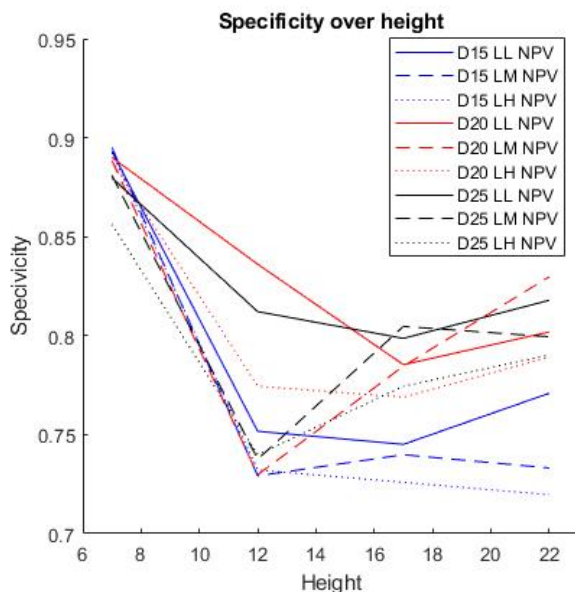


Figure D.4: Specificity over height

It seems like the specificity over the height plot yields some sort of a result. Height 7 is performing better! These images are taken with the new samples, and as seen in the main report those samples had a lower sensitivity but higher specificity. This is most likely because a more ideal threshold value was selected for the image binarization.

E SUPPORTING DOCUMENT

Step 1: Determine the code size

Input:

Amount to of data that needs to be stored
Availability of internal systems
Scanner availability

Action:

Use internal system or use table 1 to select code type and size

Output:

Code size

Recommendation: ECC Q and a normal QR code can be scanned more easily

AN* data storing capacity	Code type	Code size (modules)	Module 0.6 mm	Module 0.7 mm	ECC
5	Micro QR code	13x13	7.8 mm	9.1 mm	M
11	Micro QR code	15x15	9.0 mm	10.5 mm	M
13	Micro QR code	17x17	10.2 mm	11.9 mm	Q
16	Type 2 QR code	21x21	12.6 mm	14.7 mm	Q
29	Type 2 QR code	25x25	15.0 mm	17.5 mm	Q
47	Type 2 QR code	29x29	17.5 mm	20.3 mm	Q

Table 1: Possible codes, *AN standing for Alphanumeric

Step 2: Determine the embedment location

Input:

Code size
The product

Action:

Select suitable location based on requirements

Output:

Location of the code

Location requirements

- Minimum area, dependent on selected code and module size (0.7 recommended)
- A maximum surface curvature of 20°, 35° if unavoidable
- Unobstructed, unintrusive location. Out of view, on the bottom or back of product

Step 3: Embed the code in the CAD model

Input:

Code size
Code location
Embedment properties

Action:

Embed code in
CAD model

Output:

Printable model
with embedded
code

How to embed a QR code onto a CAD model

Step 1: Prepare code in vector form with each module as an individual pixel. <https://qrplanet.com/qr-code-generator-svg> offers vectors with modules as individual pixels. Don't forget to delete the background.

Step 2: Select and scale the whole code according to the size specified in table 1. An example is shown in figure 1.

Step 3: Enlarge the individual modules to 99.5% using a vector editing program, shown in figure 2.

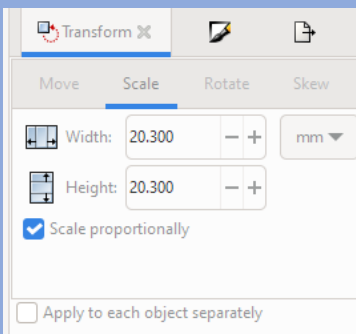


Figure 1, step 2

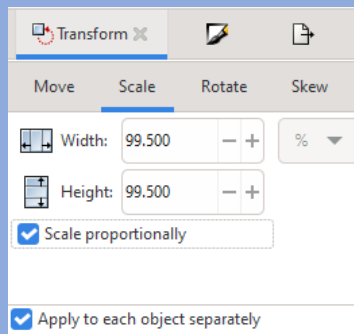


Figure 2, step 3

Step 4: In the CAD program, create plane parallel to surface of area selected for embedment.

Step 5: Start a sketch on the newly created plane and import the DXF file.

Step 6: 'Cut extrude' using the offset from surface with a value of 1.6 mm, the carving depth. Select the areas that need to be extruded. In figure 3 an example of what this looks like in SolidWorks is shown.

Step 7: Done! The model is ready to be printed.

The guidelines for the three influencing parameters

Carving depth	Module size	Max surface curvature
Minimum 1.6 mm	0.7 mm 0.6 mm if no space available	20° ideal 35° if no suitable location

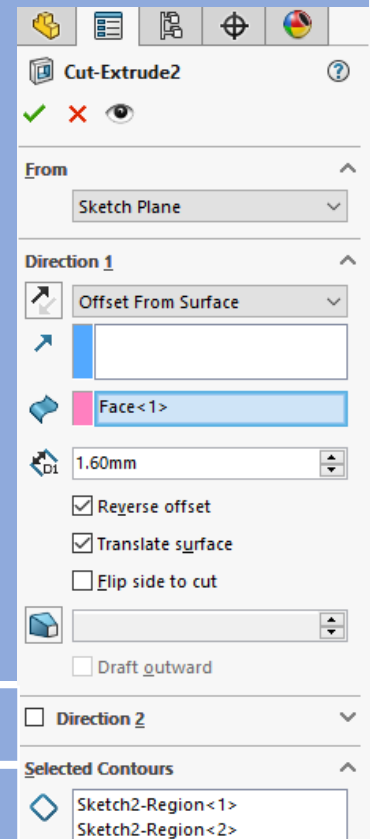


Figure 3