Full Color High Definition Fused Filament Fabrication

B. Botterman
Full Color High Definition Fused Filament Fabrication

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft University of Technology

B. Botterman

August 11, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology
The work in this thesis was supported by Leapfrog 3D Printers. Their cooperation is hereby gratefully acknowledged.
Abstract

Although Full Color (FC) High Definition (HD) Additive Manufacturing (AM) machines are available on the market today, a FC HD Fused Filament Fabrication (FFF) printer has yet to enter the competitive FFF market. Leapfrog, a FFF printer manufacturer, holds a patent that allows the application of coating on 3D printed FFF filament after it has been deposited. In this thesis work a prototype has been realized that serves as proof of the FC HD FFF concept and forms the basis of the commercial product which will be launched in the near future. Using inkjet technology high development costs can be avoided if existing coating devices are used in the machine.

The challenges are to achieve a maximum outer surface coating of the 3D printed object and minimize the time it takes to coat a 3D printed object. This time is increased dramatically by applying a coating layer after every 3D printed layer compared to no coating.

Using the prototype machine several coating methods have been explored of which the most promising method is perpendicular coating. Here, after 3D printing a layer, droplets are jetted onto the outward facing surface of the object.

The total coating time is minimized by grouping contours which can be coated in one swath and sorting the order in which groups are coated. The grouping is done by formulating and solving a bin packing problem. Mixed Integer Linear Programming (MILP) and First Fit Decreasing (FFD) are compared and used to solve the problem, achieving more than 80% time reduction compared to no grouping on test models. A minimal movement time between all the groups is found by solving a Traveling Salesman Problem (TSP). The state of the art solver Concorde (CC) is compared to a Nearest Neighbor (NN) heuristic. A movement time reduction of 56% up to 83% is achieved compared to the standard 3D printing order on the test models. CC gives a maximum improvement of 2.3% over NN.

The perpendicular coating method will be used to coat the filament. A significant total print time reduction is possible by grouping. MILP can have a hard time finding a solution for certain data sets, therefore FFD is preferred. For movement time reduction, a 2.3% performance increase by CC generally means that printing times are reduced by seconds. This is negligible considering 3D printing a layer generally takes minutes making NN a suitable heuristic for solving the TSP.
Table of Contents

1 Introduction 1
  1-1 Problem Formulation ........................................ 1
  1-1-1 Current State and Significance ......................... 1
  1-1-2 Prototype Development ................................. 2
  1-1-3 Print Time Reduction ................................. 3
  1-2 Outline .................................................. 3

2 Fused Filament Fabrication 5
  2-1 FFF Principles .......................................... 5
    2-1-1 Work Flow ........................................... 6
    2-1-2 Slicer ............................................... 7
    2-1-3 GCODE .............................................. 7
    2-1-4 Terminology ....................................... 8
  2-2 Creatr HS ................................................. 9
    2-2-1 Arduino ............................................. 10

3 Full Color FFF printer 11
  3-1 Prototype Machine ...................................... 12
    3-1-1 Patent .............................................. 12
    3-1-2 Direct Versus Layer by Layer ...................... 13
    3-1-3 UV Ink ............................................. 13
    3-1-4 CMYK Coloring .................................... 14
  3-2 Experiments and Results ................................ 15
    3-2-1 Methodology ......................................... 17
    3-2-2 Parallel Coating .................................. 19
    3-2-3 Perpendicular Coating ............................ 19
    3-2-4 Oozing ............................................. 23
  3-3 An Additional Coating Method: Absorption ............ 25
  3-4 Discussion .............................................. 26
# Table of Contents

## 4 Grouping

4-1 Coating a Contour .................................................. 30
4-2 Notation ............................................................... 31
   4-2-1 Contour Objects ............................................. 33
   4-2-2 Inward Facing Normal Vector .............................. 33
   4-2-3 Angle Distance ............................................... 34
   4-2-4 Rotation Direction .......................................... 34
   4-2-5 Aligned State ................................................ 34
4-3 Constraints .......................................................... 35
   4-3-1 Distance Constraint ......................................... 35
   4-3-2 Grouping Angle Constraint ................................. 35
   4-3-3 Potential Group Set .......................................... 36
4-4 Bin Packing Problem Formulation ................................. 37
4-5 Implementation ...................................................... 39
   4-5-1 Set Partitioning .............................................. 41
   4-5-2 Partition Merging ............................................ 42
4-6 Experiments .......................................................... 42
   4-6-1 Set-up .......................................................... 42
   4-6-2 The Effect of Partition Size on Solution Quality and Computing Time ........................................ 42
   4-6-3 Effect of the Grouping Angle on Node Reduction ...... 47
4-7 Discussion ............................................................ 47

## 5 Sorting

5-1 Traveling Salesman Problem ....................................... 51
   5-1-1 Notation ......................................................... 51
   5-1-2 TSP Solvers ..................................................... 51
5-2 Formulation ................................................................ 52
   5-2-1 Cities and Edges ............................................... 52
   5-2-2 Cost ............................................................... 52
5-3 Experiments and Results ............................................ 55
   5-3-1 Experiments ...................................................... 55
   5-3-2 Results ........................................................... 56
   5-3-3 Discussion ....................................................... 57

## 6 Conclusions and future work

6-1 Conclusions ........................................................... 59
6-2 Future Work ........................................................... 60

## A Datasets

A-1 Monte San Pietro ...................................................... 63
A-2 Nautilus Gear .......................................................... 64
A-3 Charlotte Web .......................................................... 64
A-4 Wire-frame Skull ...................................................... 65
B  Full Color High Definition Prototype Machine  67
   B-1  The Machine ................................................. 67
   B-1-1  Epson L805 ............................................. 68
   B-1-2  L805 and Creatr HS Communication ............... 70
   B-1-3  Activation of a L805 Print Via the Creatr HS .. 71
   B-2  Printing with the Prototype .............................. 72
   B-2-1  Coordinate Frames ...................................... 73
   B-2-2  Printer Y-motion ....................................... 73
   B-2-3  Buffering Moves ........................................ 75
   B-2-4  Full Color High Definition 3D Printing .......... 75
   B-3  Branch and Bound .......................................... 76

Bibliography  79

Glossary  83
   List of Acronyms ............................................... 83
   List of Symbols .............................................. 84
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>A major development in FFF: a Full Color High Definition image on a 3D print.</td>
<td>2</td>
</tr>
<tr>
<td>2-1</td>
<td>Schematic of a typical extrusion-based AM process. [1]</td>
<td>5</td>
</tr>
<tr>
<td>2-2</td>
<td>Surface roughness of a 3D printed object.</td>
<td>6</td>
</tr>
<tr>
<td>2-3</td>
<td>A model of a sphere (left) represented in Stereo lithography (STL) format for varying resolutions. luxexcel.com</td>
<td>6</td>
</tr>
<tr>
<td>2-4</td>
<td>Workflow of FFF.</td>
<td>6</td>
</tr>
<tr>
<td>2-5</td>
<td>3D model of a nautilus gear.</td>
<td>7</td>
</tr>
<tr>
<td>2-6</td>
<td>3D model of a nautilus gear. The first layer is printed with a lower feed rate(blue) than the rest of the layers(green) to ensure a proper adhesion to the printer bed surface.</td>
<td>8</td>
</tr>
<tr>
<td>2-7</td>
<td>Schematic of the Leapfrog Creatr HS 3D printer.</td>
<td>9</td>
</tr>
<tr>
<td>2-8</td>
<td>Data flow of the Creatr HS.</td>
<td>10</td>
</tr>
<tr>
<td>3-1</td>
<td>The FC HD FFF prototype machine. The original bed of the Creatr HS has been replaced by a glass disc that can rotate the 3D printed object. This way, all faces of the object can be coated by the Coating Device (CD).</td>
<td>11</td>
</tr>
<tr>
<td>3-2</td>
<td>Full Color prototype printer schematic view. The bed and hot end can be recognized from the schematic view of the Creatr HS shown in Figure 2-7. An Epson L805 inkjet carriage system (CD Carriage) is mounted on the Y-axis carriage of the Creatr HS. The area both the hot end and CD can reach is indicated by the yellow square surface.</td>
<td>12</td>
</tr>
<tr>
<td>3-3</td>
<td>Patent No. 2012940 illustrated for a layer by layer approach. Step 1: deposit a layer of filament. Step 2: Coat the layer. Step 3: Apply the next layer and repeat the process.</td>
<td>13</td>
</tr>
<tr>
<td>3-4</td>
<td>Schematic illustration of the direct coating approach.</td>
<td>13</td>
</tr>
<tr>
<td>3-5</td>
<td>Print of the Leapfrog logo on a block using the thermal inkjet prototype machine. The arrow points to ink smeared out over the layer because it was not dry when a new layer was printed on top of it.</td>
<td>14</td>
</tr>
</tbody>
</table>
3-6 A FC HD image printed on a flat surface of Polylactic Acid (PLA) using a 2D Ultra Violet (UV) inkjet printer. The surface is printed on the way a regular desktop inkjet printer would print paper. .................................................. 15
3-7 Cyan Magenta Yellow Key (CMYK) color model. .................................................. 15
3-8 Changing the intensity of a base color using amplitude modulation or frequency modulation. The lightest perceived color is at (a). Color intensity increases from (a) to (d)[2]. .................................................. 16
3-9 Creating an image using a CMYK color model. Amplitude modulation is used to create color tints. Notice how the grid of each base color has a different angle to counter moiré patterns. .................................................. 16
3-10 Coating contours from above. .................................................. 17
3-11 By zooming in on the eye of the frog, the effect illustrated in Figure 3-10 can be seen clearly. .................................................. 17
3-12 The viewing angle will determine the intensity of colors coated on the object. Eye 1 will see a more intense color than eye 2. .................................................. 18
3-13 A coated wall with 30° overhang. Overhang decreases the intensity of the image because a larger white surface will be seen by the eye as the overhang angle increases. .................................................. 18
3-14 3D model of the object used to uncover the best coating coverage. .................................................. 19
3-15 Undercoat and overcoat terminology illustrated. The cross sections illustrate how ink lands on the outer contours of the filament. The dashed lines indicate which of situation (undercoat, desired, overcoat) corresponds with the schematic cross section view and a real coated 3D object. Filament misalignment is exaggerated in the overcoat situation to better illustrate what happens when overcoating. .................................................. 20
3-16 Parallel coating. .................................................. 20
3-17 Parallel coating on the test subject block having a 0.2 mm layer height setting. Both view angles of Figure 3-12 are captured. The lower view angle image is created by flipping the block and mirroring the image for a more easy comparison of the effect of the view angle on the coating intensity. The best coated surface is indicated by the dashed square. .................................................. 21
3-18 Coating under an angle. .................................................. 21
3-19 The perpendicular coating strategy. By jetting droplets with a horizontal velocity the side of the filament can be coated. .................................................. 22
3-20 The perpendicular coating strategy on the test object having a 0.2 mm layer height setting. By jetting droplets with a horizontal velocity the side of the filament can be coated. .................................................. 22
3-21 A single line cross printed on a horizontal surface of a test object having a 0.2 mm layer height setting. .................................................. 23
3-22 Jetting of a droplet using an inkjet device. .................................................. 23
3-23 Perpendicular and parallel coating methods compared. Both objects have a 0.2 mm layer height. Top: perpendicular coating. Bottom: parallel coating. .................................................. 24
3-24 The oozing coating strategy. A droplet of ink oozes along the surface of the filament, coating the side of the filament surface. .................................................. 24
3-25 Giving a pixel a perceived color using the oozing coating method. .................................................. 24
3-26 Oozing experiment for 7 layers of ink, curing on the 8th layer. .................................................. 25
3-27 Coloring the filament by ink absorption. .................................................. 26
3-28 A single absorption pixel. .................................................. 26
4-1 Data flow diagram. The slicer slices GCODE from an STL file. Printing time is reduced by extracting contours from the GCODE and minimize the coating time by grouping and sorting the contours such that a minimum amount of swaths is needed and the traveling time between swaths is minimized.

4-2 Top view schematic of the prototype with the rotating bed installed.

4-3 Top view of the prototype machine in coordinate frames. The red FRAME is the original bed coordinate frame of the Creatr HS 3D printer. The green 3D frame is the reduced 3D printing area due to installation of the L805 inkjet system on the Creatr HS. The purple 2D frame is an A4 paper sized working area of the L805 inkjet system. The dashed circle provides a visual aid to indicate the area which both the 3D printer and inkjet system can reach in any configuration. The inkjet (the red rectangle) is aligned to a contour, indicated by the black arrow. The grey line shows the coating path of the inkjet device.

4-4 Zoomed view of a the contour line pointed at in Figure 4-3. The contour line is indicated in red. Curved contours are approximated by many small straight contour lines.

4-5 By rotating the bed and translating the inkjet the system is changed from the aligned state of contour line 1 (as illustrated in Figure 4-3) to the aligned state of contour line 2.

4-6 Distance constraint visualized on the largest contour isle of the Nautilus Gear model. Red lines indicate contour lines that fall within the bounds of the constraint.

4-7 Schematic illustration of the angle constraint. Contour lines which fall within $\alpha_{max}$ of $C_i$ are shown in green. The Contour line which falls outside the bounds imposed by $\alpha_{max}$ is shown in red.

4-8 Contour lines that fall within the angle constraint $\alpha_{max} \leq \frac{\pi}{16}$ are indicated in red. Red contour lines which are very small are indicated by arrows.

4-9 Intersection of the distance and angle constraints visualized for one of the contours of the largest contour isle of the Nautilus Gear model. Contours that satisfy both the distance and angle constraints of $C_i$ are shown in red. The figure shows the intersection of Figure 4-6 and Figure 4-8.

4-10 The MILP algorithm run on the largest contour isle of the Monte San Pietro model (Appendix A-1). Background processes running on Windows disturb computing time polling. Therefore, each poll is repeated 10 times. Each box represents a set of 10 repeated polls on the same data set. The boxplot shows the median as a red line and the mean as an asterisk. 25th and 75th percentile of data are the lower and upper blue lines of the blue box. Whiskers indicate the minimum and maximum of the data. $N_c = 20$ has the largest spread of data of 0.1% of the mean for $2\sigma$. The figure shows the exponential increase of the median of the computing time for increasing $N_c$. The sets are subsets of the complete contour isle, and each smaller set is a subset of a larger set. The test is run on an ASUS UX305F Notebook PC, detailed in Table 4-1.

4-11 Computing time and reduction percentages of the MILP approach for varying partition size $N_p$. Test are conducted using $\alpha_{max} = \frac{\pi}{16}$ rad.

4-12 A visualization of partitions of the biggest contour of the Monte San Pietro model for $N_p = 92$. Each partition has a color. Corresponding colored boxes behind the contour partitions make a more clear image of the partitions. Branch and bound computing times are displayed near each partition. It can be seen that the yellow partition has the longest branch and bound time, whereas the red partition has a 0 s branch and bound time. This means that an optimum was found before the branch and bound phase.

4-13 A visualization of partitions of the largest contour isle of the Monte San Pietro model for $N_p = 82$ and $N_p = 102$. Branch and bound computing times are displayed for each partition.
4-14 Total time reduction percentages. The MILP solution displayed as box plots with their mean as an asterisk is compared to the FFD solution which is shown as a straight line because it is independent of $N_p$. 46

4-15 Computing times and node reduction percentages as a function of $\alpha_{\text{max}}$. Tests have been repeated 10 times to account for the spread in computing time. Results are shown as boxplots. The red line is the median, upper and lower blue box bounds represent the 25th and 75th percentile of data. Top and bottom whiskers represent the lower and upper bound of the data after removing outliers. Data outliers are given as red plus signs. The mean is shown as an asterisk. Grouping is done by MILP with $N_p = 62$ on the largest contour isles of Monte San Pietro (MSP), Nautilus Gear (NG) and Charlotte Web (CW). 48

5-1 Velocity profile generated by the stepper driver in the step domain $n$ and time domain $t$. $n_a$ is amount of steps to taken to accelerate to $F_{\text{max}}$. Because the acceleration is constant this is the same distance to decelerate back to zero. $n_c$ is the number of steps a constant velocity is maintained. In the same way $t_a$ denotes the acceleration time and $t_c$ the constant velocity cruising time. In the time domain a piecewise linear model is followed. The step model is not linear in the acceleration and deceleration phase but will always have the set velocity in $n_c$ if $n \geq 2n_a$. 54

5-2 In Figure 5-2a lower and upper bound of the blue boxes show the 25th and 75th percentile of the data respectively. Whiskers shows the minimum and maximum of the data after outliers have been removed. The median is shown as a red line. Plus signs indicate removed outliers. Figure 5-2b shows the normal distribution of the reduction percentages. 57

5-3 Performance of the NN compared to CC for an increasing number of vertices. 58

A-1 3D model of Monte San Pietro. 64
A-2 3D model of the Nautilus Gear. 64
A-3 3D model of the Charlotte web. 65
A-4 3D model of the wire-frame skull. 65

B-1 Top view schematic of the prototype. 67
B-2 Schematic of the L805 carriage as installed on the Creatr HS. Motors and sensors are not displayed. 69
B-3 L805 print head 70
B-4 L805-CreatrHS communication scheme. 70
B-5 Encoder pulse differential. Encoder values are sampled with $t_s = 200 \mu$s. The differential of this data is polled roughly every 10 ms. 72
B-6 Coordinate frames relative to 3D printer frame coordinates. The purple dotted line shows the lowest point the CD can reach. The black dotted circle highlights the area both 3D and 2D devices can reach. This is the working area. 73
B-7 A major development in FFF: The first Full Color High Definition image on a 3D print. 74
B-8 Data flow in the prototype machine systems. 75
B-9 Branch and bound of the MILP. $u$ is an upper bound of the solution found heuristically. $f_x x$ is an Linear Programming (LP) solution with one or more variables fixed. 77
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>3D printer jargon.</td>
<td>9</td>
</tr>
<tr>
<td>2-2</td>
<td>Creatr HS specifications.</td>
<td>10</td>
</tr>
<tr>
<td>4-1</td>
<td>ASUS UX305F Notebook PC.</td>
<td>40</td>
</tr>
<tr>
<td>4-2</td>
<td>Experiment testing machine.</td>
<td>42</td>
</tr>
<tr>
<td>4-3</td>
<td>Computing time and node percentage extrema of the MILP results presented in Figure 4-11</td>
<td>44</td>
</tr>
<tr>
<td>4-4</td>
<td>Maximum time reductions of the MILP and FFD approaches on the largest contour isles of the MSP, NG and CW test models.</td>
<td>47</td>
</tr>
<tr>
<td>5-1</td>
<td>Velocity profile parameters</td>
<td>56</td>
</tr>
<tr>
<td>A-1</td>
<td>Slicer settings.</td>
<td>63</td>
</tr>
<tr>
<td>A-2</td>
<td>Monte San Pietro contour data.</td>
<td>64</td>
</tr>
<tr>
<td>A-3</td>
<td>Nautilus Gear contour data.</td>
<td>64</td>
</tr>
<tr>
<td>A-4</td>
<td>Charlotte Web contour data.</td>
<td>65</td>
</tr>
<tr>
<td>A-5</td>
<td>Wire-frame Skull contour data.</td>
<td>65</td>
</tr>
<tr>
<td>B-1</td>
<td>L805 specs.</td>
<td>69</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1-1 Problem Formulation

1-1-1 Current State and Significance

Additive Manufacturing (AM) is the process of joining materials to make physical objects from three-dimensional (3D) model data, usually layer by layer, as opposed to subtractive manufacturing methodologies [3]. Development of AM started in the 1980’s. Since then significant progress has been made and the market for AM machines has been rapidly growing [4][5]. AM can be used to create complex shapes from a wide array of materials with near zero material waste and little post processing.

Among AM technologies Fused Filament Fabrication (FFF) printers are most commonly used by consumers and small companies due to the low machine cost. With the expiration of the original patent of Stratasys on FFF printing [6], the market of FFF printers has increased rapidly. To get ahead of the competition FFF machine manufacturers have to keep improving and innovating their products.

One innovation the FFF industry has yet to see is the projection of Full Color (FC) High Definition (HD) imagery on FFF printed objects. As of today, color FFF printers are only able to print gradients, no details. Projecting HD imagery on a 3D printed object can aid in the visualization. Think of architects, for example, who will be able to print maquettes straight from their computer without having to do any post processing on their prints. Or small batch products with complex shapes. These shapes can be hard to color after printing as some areas might be hard to reach. Coloring during printing makes sure all outer surfaces of the object can be colored.

A future application is the possibility to apply coating materials and fillers to 3D printed objects. A filler material can be applied to each layer to smooth the ridged surface typical of FFF printed objects. Another example is that an object can be made more weather resilient if it is coated in a material less sensitive to weather conditions. These coatings and fillers can be applied locally in contrast to dipping an entire object in a bath of coating material.
As of today, AM machines capable of full color HD printing are the Stratasys Objet260 Connex3 and the Mcor iris. These are not FFF machines and are in a much higher price range than the average FFF machine, leaving a gap to be filled in the low price segment ($\leq 10,000). The FC HD 3D printer is highly anticipated by the industry and 3D printing community. It is considered the holy grail of FFF and any manufacturer developing such a machine first will be in the money. Leapfrog aims to fill this gap with the development of this machine.

1-1-2 Prototype Development

Leapfrog has acquired a patent on coating/coloring printed material after it has been printed. This is explained in detail in Section 3-1-1. Based on this patent a commercial machine will be build. Before this can be realized a suitable coating method has to be found, satisfying the constraints put up by the patent, cost of manufacturing and limits of technology.

Coloring 3D printed filament is unknown territory. Whatever effect placing ink on the round surface of the filament has is not known and has to be investigated. This is a reason to keep the printing technique simple by coating in a layer by layer approach, as opposed to direct coating, where both the printing and coating happen at the same time. One can imagine that direct coating would be the fastest approach of the two. This technique should be explored in a later stadium, when the biggest hurdles of coating 3D objects have been identified.

By mounting an Epson L805 piezo inkjet 2D desktop printer on a Leapfrog Creatr HS 3D printer it is possible to proof the concept of coating after 3D printing. With this machine, a milestone was reached in FFF: a FC HD image on a 3D printed surface, shown in Figure 1-1.

![Figure 1-1: A major development in FFF: a Full Color High Definition image on a 3D print.](image)

The prototype is also used to investigate coating strategies, highlighted in Section 3-2, which can provide a maximum coating of the outer surface of the round filament. Perpendicular coating is the most promising. By giving ink droplets a velocity it is possible to slap them to the round surface. This way the side of the filament can be coated without tilting the print assembly.

To coat a 3D printed object from all sides the object has to be rotated. This is done by printing the object on a rotating bed, rather than rotating the entire Coating Device (CD) assembly. Using the rotating bed prototype droplets can be jetted on the side of printed filament of 0.1mm in diameter.
1-1-3 Print Time Reduction

The layer by layer coating approach increases the total print time significantly. This is because every contour line of every layer is coated individually. By grouping lines which can be coated in one swath the print time can be reduced. By finding a minimum amount of groups this time can be minimized.

After groups of contours have been formed they have to be coated. To coat a group the machine has to align the object such that the group can be coated in one swath. By sorting the order in which the groups are coated the time it takes to align the machine can be minimized.

The prototype printer is a first iteration of what the commercial machine will be. How the commercial machine will operate will be very similar to the prototype machine. Therefore, the scope of the optimization algorithms is constrained by the prototype machine.

1-2 Outline

In Chapter 2 the basics of FFF are explained. This section highlights basic operation of a 3D printer as well as relevant details of the inner workings of the machine.

A suitable method of coating the filament is presented in Chapter 3. Here, design decisions resulting in the final prototype machine are discussed. The prototype serves as a basis for the coating time reduction algorithms.

FC printing in a layer by layer fashion significantly increases the total printing time. In Chapter 4 this time is reduced by grouping contours which can be printed in one swath. Chapter 5 attempts to further reduce the printing time by changing the order in which the groups are coated such that the total movement time is minimized.

Conclusions and future work are presented in Chapter 6.
Chapter 2

Fused Filament Fabrication

This chapter covers the basics of Fused Filament Fabrication (FFF). The process of turning a CAM/CAD model into a 3D printed model is explained step by step. Relevant machine parameters and printer firmware operations are explained in some detail.

2-1 FFF Principles

Extrusion based FFF is illustrated in Figure 2-1. A filament of amorphous print material is fed through to a heated liquifier by the feed pinch rollers. In the liquifier the feed material is heated to the glass transition temperature. The material is extruded onto a previously deposited layer and cooled immediately. When a layer is completed, the part is lowered and the process is repeated on the next layer. Typical of FFF is the ridged surface structure.
of printed objects, shown in Figure 2-2. This is due to the stacking of layers of extruded filament. The liquifier has a round exit. This gives the filament a rod like shape as it leaves the liquifier.

(a) Outer surface of a print.  
(b) Cross section of a print with the outer layers on the right. - [7]

**Figure 2-2**: Surface roughness of a 3D printed object.

### 2-1-1 Work Flow

The basic FFF printer work flow is illustrated in Figure 2-4. A designer creates a 3D object in his preferred 3D modeling software. This object is exported in Stereo lithography (STL) file format. In this format, 3D objects are represented as triangulated surfaces. Figure 2-3 shows this triangulation for a sphere in different resolutions.

![Figure 2-3](luxexcel.com)

**Figure 2-3**: A model of a sphere (left) represented in STL format for varying resolutions. - luxexcel.com

Slicer software slices this model into layers. Each layer is represented as series of straight lines and print head moves. The slicer also calculates the volumetric flow rates for extrusion moves and determines infill patterns. All information is stored in a time linear instruction list called a GCODE. Line by line the GCODE is processed by the 3D printer, creating the physical 3D printed object.

![Figure 2-4](luxexcel.com)

**Figure 2-4**: Workflow of FFF.
2-1-2 Slicer

The slicer software translates a 3D model into GCODE. Slicing is an important step in the 3D printing process as printing strategies and settings here have a major influence on the quality of the final print. You can have the best 3D printer in the world, but if your slicer settings are off you still get a poor print.

In this thesis Simplify 3D is used for slicing. It has a myriad of slicing settings and a real time communication window with the printer. Some prominent slicer settings include layer height, feed rate and object infill percentage. These settings have a significant impact on the surface quality and stiffness of the print. Optimal print settings are dependent on the individual object. It requires a certain amount of skill to make a decent print. This is one of the reasons why 3D printing is not yet widely used among consumers.

To illustrate how a model is sliced, a 3D model of a nautilus gear is loaded into the slicer (Figure 2-5).

![3D model of a nautilus gear.](image)

Figure 2-5: 3D model of a nautilus gear.

The slicer slices the model into layers and filament lines. Outer contours are sliced as one continuous line. This results in a smooth looking outer surface. In Figure 2-6 a preview of the slicing of the first and second layer of the nautilus gear is shown.

The preview of Figure 2-6 is translated into a set of machine instructions called GCODE.

2-1-3 GCODE

GCODE is a list of machine instructions. It is processed line by line by the 3D printer. 3D printer firmware is based on CNC milling machine firmware. This is seen back in GCODE terminology. Extruders are indicated as tools and velocities as feed rates, for example.

Depending on the size and shape of the model and slicer settings a GCODE can consist of thousands of instructions. This is due to FFF printers relying completely on the feed forward nature of stepper motors. Every move, every extrusion and every setting has to be parsed to the printer line by line.

Listing 2.1 shows a typical fragment of GCODE. The \( X \) and \( Y \) are the next coordinates the print head has to move to. \( E \) is the amount of filament that has to be extruded during the move and \( F \) is the desired feedrate. \( X \), \( Y \) and \( E \) are in mm. Extrusion volumes have been calculated in the slicer. The \( E \) instruction tells the extruder motor how many mm to displace.
Beside movement and extrusion instructions, GCODE can be used to alter printer settings. Temperatures and fan speeds, for example. Even Electrically Erasable Programmable Read-Only Memory (EEPROM) memory data of the printer controller board can be saved and read using these commands. This can be used to quickly change nozzle offsets for dual extrusion without having to alter and re-upload the firmware.

Listing 2.1: Fragment of GCODE

```
1 ; layer 30, Z = 4.5
2 ; tool H0.150 W0.400
3 ; inner perimeter
4 G1 X73.059 Y113.096 F6000
5 G1 Z4.500 F1000
6 G1 E287.0552 F24000
7 G1 X72.936 Y112.786 E287.0631 F2950
8 G1 X72.647 Y112.750 E287.0700
9 G1 X69.503 Y108.869 E287.1883
10 G1 X68.673 Y107.914 E287.2183
```

2-1-4 Terminology

Throughout this report some technical jargon reoccurs. This jargon is included in Table 2-1. It should be clear to the reader that a curved shape is approximated by a number of straight lines. This can be seen in Figure 2-3. A layer of the sphere is a circle. Its contour consist of straight lines. These are referred to as contour lines. If a model would consist of three spheres, a layer would contain a disconnected contour for every sphere. In such case the contour of a single sphere is referred to as a contour isle.
<table>
<thead>
<tr>
<th>term</th>
<th>unit</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feed Rate</td>
<td>mm s(^{-1}) or steps/s</td>
<td>Term used for the carriage and extruder stepper motor velocity.</td>
</tr>
<tr>
<td>Contour</td>
<td>-</td>
<td>Outward facing filament lines of a layer.</td>
</tr>
<tr>
<td>Contour line</td>
<td>-</td>
<td>Single element of a contour or contour isle.</td>
</tr>
<tr>
<td>Contour isle</td>
<td>-</td>
<td>A closed loop of contour lines.</td>
</tr>
</tbody>
</table>

Table 2-1: 3D printer jargon.

2-2 Creatr HS

The Leapfrog Creatr HS, shown in Figure 2-7, will be used as a basis for the prototype machine. It is a dual extruder desktop 3D printer.

![Figure 2-7: Schematic of the Leapfrog Creatr HS 3D printer.](image)

**Actuators and heaters** The printer head and bed are actuated by NEMA 23 stepper motors. The printer head is belt driven over the \(XY\)-plane by the \(X\)- and \(Y\)-motors respectively. On both axes, the step size is 15 \(\mu\)m. The \(Z\)-axis motor is connected to the bed using a spindle which can move the bed with 1 \(\mu\)m per step. Other relevant printer specifications are given in Table 2-2.

The heater elements in the hot-ends (liquifier of Figure 2-1) heat the printing material, Polyvinyl Alcohol (PVA) for example, to a glass transition temperature. This is 190°C[8] for PVA. A third heater heats the print bed (build-stage of Figure 2-1). By heating the bed the first few layers of the print, the object adheres firmly to the bed. If the object would come loose during printing the print would be ruined. By cooling down the bed at the end of the print, the object can be removed more easily from the bed.
Table 2-2: Creatr HS specifications.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build volume</td>
<td>280 x 270 x 180 mm</td>
</tr>
<tr>
<td>Nozzle diameter</td>
<td>0.35 mm</td>
</tr>
<tr>
<td>Rated min. layer height</td>
<td>50 µm</td>
</tr>
<tr>
<td>Max. Feed rate Y-stepper motor</td>
<td>400 mm s⁻¹</td>
</tr>
<tr>
<td>Max acceleration Y-stepper motor</td>
<td>3000 mm s⁻²</td>
</tr>
<tr>
<td>XY-step size</td>
<td>15 µm</td>
</tr>
</tbody>
</table>

**Sensors** Each axis has an end stop used to zero the stepper motors. For the XY-plane these end stops are micro switches. Setting the zero position of the steppers is called homing. It is not necessary to have a high repeatability between prints, only the relative position is important. Therefore cheap micro switches are used in the XY-plane. In the Z-plane the absolute position is important, as the first layer always has to be of the same thickness for every print otherwise the nozzle can drill into the bed or be too far away from the bed, causing the first layer not to adhere properly. An induction sensor is used in the Z-axis to get an accurate bed starting position. Thermistors in each nozzle and the bed provide temperature data.

### 2-2-1 Arduino

![Data flow of the Creatr HS](image)

The printer is controlled by a 16MHz Arduino MEGA 2560. The Arduino runs open source 3D printer firmware, Marlin, which parses and buffers GCODE fed by either a Personal Computer (PC) or USB storage device via an Olimex A13-OLinuxXino computer. The Arduino transforms GCODE instructions into pulses to the stepper driver which sets the stepper motors in motion.

Parsing and buffering GCODE, processing this GCODE into movements, driving the stepper motors and other processes, part of the Marlin firmware, require a lot of memory and computational power from the Arduino. Additional computational load on the system should therefore be avoided.
Figure 3-1: The FC HD FFF prototype machine. The original bed of the Creatr HS has been replaced by a glass disc that can rotate the 3D printed object. This way, all faces of the object can be coated by the CD.

During this thesis work a FC HD FFF prototype has been realized, shown in Figure 3-1. The prototype consists of an Epson L805 inkjet carriage mounted on the Y-axis carriage of the Creatr HS. This system is shown schematically in Figure 3-2. Here, the bed is shown as a static square surface because the rotating bed was installed after completion of the tests in this chapter.

The goal of this chapter is to find a suitable coating method to apply coating to a 3D printed object using the prototype machine.
Figure 3-2: Full Color prototype printer schematic view. The bed and hot end can be recognized from the schematic view of the Creatr HS shown in Figure 2-7. An Epson L805 inkjet carriage system (CD Carriage) is mounted on the Y-axis carriage of the Creatr HS. The area both the hot end and CD can reach is indicated by the yellow square surface.

3-1 Prototype Machine

The way coating is applied to 3D objects is restricted by a patent. This section explains the patent and why coating is applied layer by layer rather than directly after the filament has been extruded. If the coating is a color, the Cyan Magenta Yellow Key (CMYK) color palette is used to generate a color. A basic knowledge on CMYK coloring is provided to give the reader a better understanding of the challenges faced in FC HD FFF.

3-1-1 Patent

Leapfrog holds the Dutch national patent (Patent No. 2012940) on coloring filament after extrusion. One way of interpreting this is illustrated in Figure 3-3. In the first step a layer of filament is extruded. In the next step this filament is coated. If the coating has dried, the next layer is applied. This is a very simplistic representation of the patent but it explains the layer by layer approach, referred to as layer by layer coating, used in this thesis work.

The patent allows coating right after the nozzle as well. In this approach the layer of filament is coated before it is finished. This process is referred to as direct coating.
Figure 3-3: Patent No. 2012940 illustrated for a layer by layer approach. Step 1: deposit a layer of filament. Step 2: Coat the layer. Step 3: Apply the next layer and repeat the process.

Besides layer by layer coating or direct coating the way the ink is applied to the filament is covered by the patent as well. This is referred to as a coating method. Coating methods are described in more detail in Section 3-2.

3-1-2 Direct Versus Layer by Layer

A direct coating approach is shown schematically in Figure 3-4. The advantage of direct coating over layer by layer coating is a shorter coating time because printing is directly followed by coating.

Figure 3-4: Schematic illustration of the direct coating approach.

The difficulty of direct coating is that a moving part must be attached to the 3D print head. This will make the system complex because of the introduction of extra dynamics, small moving parts and sophisticated control schemes. As coating 3D printed objects is still relatively unexplored the choice was made to keep the prototype as simple as possible such that the number of variables is kept to a minimum. Therefore a layer by layer approach is used in this thesis work.

3-1-3 UV Ink

When a layer of filament is coated and the ink has not dried before the next layer is printed on top of it, the ink will smear. This can be seen in Figure 3-5. The frog was printed using a thermal inkjet device. The ink coming from this device is water based. Ink pigments are supposed to fall in the cavities of a porous material, but the material printed with, Polylactic Acid (PLA), has no porous structure. Therefore pigments and filament are not consolidated,
causing the pigments to be smeared over the filament surface when the next layer is 3D printed.

This issue can be solved by printing with a porous material. This puts a limit on the type of materials the FC HD printer will be able to print with.

![Image](image.png)

**Figure 3-5:** Print of the Leapfrog logo on a block using the thermal inkjet prototype machine. The arrow points to ink smeared out over the layer because it was not dry when a new layer was printed on top of it.

To overcome this Ultra Violet (UV) curable ink is used. This ink can be cured almost instantly by exposing it to UV light. Not only does it harden, it also adheres firmly to smooth surfaces. Using UV curable ink prints on a flat surface of glass have been made which could not be removed by hand. From these observations and UV printing on a PLA testing surface, shown in Figure 3-6, it is likely that UV curable ink can be used for coating FC HD imagery on FFF filament.

### 3-1-4 CMYK Coloring

To print a wide array of colors on a non radiating surface (paper for example) the CMYK subtractive color model is used in inkjetting technology. The CMYK color model uses a white surface as a base. Colors are subtracted from the reflected white light using the CMYK base colors. This can be seen in Figure 3-7.

By varying CMYK color amounts intermediate colors can be created. In a jetting device colors can not be mixed before they are jetted onto a surface. To be able to create a tone by jetting CMYK droplets, base color tints are overlapped. This process is called halftoning. A tint can be created using ink droplets by either increasing the number of droplets on a surface, known as frequency modulation, or by increasing the droplet diameter, known as amplitude modulation. These modulation patterns are illustrated in Figure 3-8.

By overlapping the different tints and patterns of the base colors a FC image is created. This is shown in Figure 3-9. The grid of each base color is rotated under a unique angle to prevent moyree patterns from occurring. By mixing halftone patterns of CMYK base colors, a full spectrum of perceived colors can be created[2][9][10].
Figure 3-6: A FC HD image printed on a flat surface of PLA using a 2D UV inkjet printer. The surface is printed on the way a regular desktop inkjet printer would print paper.

Figure 3-7: CMYK color model.

Black ink, Key, is used to darken dark locations of a print. This is to save ink. A dark pixel in the image would require three droplets of the base colors. This is replaced by one droplet of Key ink.

3-2 Experiments and Results

As can be seen in Figure 2-2b in Chapter 2 the extruded filament is shaped like a rod. If the filament is coated from the top, the outer surface of the filament will not be completely coated. Figure 3-10 shows this schematically. Figure 3-11 shows this effect on a real print.
Figure 3-8: Changing the intensity of a base color using amplitude modulation or frequency modulation. The lightest perceived color is at (a). Color intensity increases from (a) to (d)[2].

Figure 3-9: Creating an image using a CMYK color model. Amplitude modulation is used to create color tints. Notice how the grid of each base color has a different angle to counter moyree patterns.

The intensity of the color will be dependent on the angle from which the object is viewed. In Figure 3-12 eye 1 will see a better coated object than eye 2.

Ideally, the entire outward facing surface of the filament is coated, such that the image quality is not dependent on the viewing angle. This will give the best looking print from all directions. In 3D printing, objects are likely to have overhang. Overhang is illustrated in Figure 3-13. The overhang shows the need to coat the entire outward facing surface of the filament. If the overhang angle increases the eye in Figure 3-13 will see less of the coated part of the filament.

In this section, several methods attempting to maximize the coated outer surface are tested.
3-2 Experiments and Results

3-2-1 Methodology

In this thesis work a number of coating methods have been investigated. The investigation involves printing on a rectangular cuboid on which a black line is printed, shown in Figure 3-14. The black line has a thickness of one nozzle of the L805 print head. The side of the cuboid that will be coated is shaped like a staircase such that alignment errors are accounted for. The steps of the staircase have a height of 15 $\mu$m. This is the minimum step size the 3D printer steppers can make. A layer height of 0.2 mm is used to print the test object.

Overhanging structures have been investigated by sheering the block of Figure 3-14 with 15 $^\circ$ and 30 $^\circ$.

Coating in black Control over the L805 is limited. Whatever halftoning patterns the L805 uses is unknown. To rule out any unknowns, testing is done with black only.
Figure 3-12: The viewing angle will determine the intensity of colors coated on the object. Eye 1 will see a more intense color than eye 2.

Figure 3-13: A coated wall with 30° overhang. Overhang decreases the intensity of the image because a larger white surface will be seen by the eye as the overhang angle increases.

**Best surface**  The best coated surface of the test cuboid is the coated surface which is darkest, but shows no misjetting of droplets from one layer onto another layer. When 3D printing a vertical wall, filament lines are not stacked perfectly. There is some misalignment present, causing some layers to stick out further than the rest. If the inkjet is calibrated in such a way that it only hits the filament lines that are sticking out, these lines will appear darker than the rest because they have been coated numerous times. This is referred to as **overcoating**. Opposed to overcoating, if ink is jetted too far inward the next layer will cover most of the coat, causing the coated image to fade. This is referred to as **undercoating**. Overcoating and undercoating are illustrated in Figure 3-15. In this figure the misalignment of layers is exaggerated to give the reader a better understanding of what is going on during over- or undercoating. When the 3D object is well aligned with the inkjet device this misalignment is less clear.
Quality  Coating quality is determined by the darkness of the best coated surface of the test cuboid. No numeric value can be assigned to the quality, but by placing the best results of all strategies next to each other it is clear to determine which has the best coverage.

3-2-2 Parallel Coating

In parallel coating the contour line and coating direction are parallel. This is illustrated in Figure 3-16. This printing method was used to create the frog of Figure 3-11. Although it is a great first step to FC printing, it can only be used to coat the top of a layer.

The test object printed using the parallel coating method is shown in Figure 3-17. It shows the viewing angle issues discussed in Section 3-2. Although the best square shows a uniform coating, it appears gray rather than black. This is more apparent when the object is viewed from below. This is due to the outer filament being only coated half, causing the observer to see a color that is somewhere halfway between white and black when the object is viewed from a distance.

A larger side contour surface can be coated if the CD would be held under an angle as shown in Figure 3-18. This would require the entire assembly to be held under an angle.

During testing, inkjet manufacturers gave mixed signals about the possibility printing with a tilted inkjet. This, combined with the fact that controlling and aiming the L805 is very difficult and tilting the assembly would require a significant reconstruction and recalibration of the prototype, caused this test to be postponed to a later stage in the project. An extra downside to tilting the assembly is that if the CD has an angle it can no longer coat large flat surfaces. However, this can be solved by making the tilting of the head dynamic. This will introduce extra calibration variables in a system that is already hard to calibrate.

3-2-3 Perpendicular Coating

With perpendicular coating the coating device direction of motion has a $90^\circ$ angle to the contour line. By giving the inkjet droplets a horizontal velocity, they are jetted to the side of the object.
Figure 3-15: Undercoat and overcoat terminology illustrated. The cross sections illustrate how ink lands on the outer contours of the filament. The dashed lines indicate which of situation (undercoat, desired, overcoat) corresponds with the schematic cross section view and a real coated 3D object. Filament misalignment is exaggerated in the overcoat situation to better illustrate what happens when overcoating.

Figure 3-16: Parallel coating.
Figure 3-17: Parallel coating on the test subject block having a 0.2 mm layer height setting. Both view angles of Figure 3-12 are captured. The lower view angle image is created by flipping the block and mirroring the image for a more easy comparison of the effect of the view angle on the coating intensity. The best coated surface is indicated by the dashed square.

Figure 3-18: Coating under an angle.

the filament. This is illustrated in Figure 3-19.

Figure 3-20 shows the test object printed using the perpendicular coating strategy. Ink has been deposited on a grid. This is because with perpendicular coating the angle between the
Figure 3-19: The perpendicular coating strategy. By jetting droplets with a horizontal velocity the side of the filament can be coated.

nozzle array of the L805 and the filament is 0°. With parallel coating this is angle is 90°. The droplet spacing corresponds with the nozzle spacing in the nozzle array of the L805 inkjet head. The L805 nozzle array can be seen in Figure B-3 of Appendix B.

The darkest regions of the test subject show some yellow contamination. This is because yellow and black nozzles lie closely together on the L805 print head. The cleaning mechanism of the L805 is not functioning at 100% due to the installation on the Creatr HS. This causes the cleaning station to sometimes wipe some yellow ink on the black nozzles in stead of removing it.

Figure 3-20: The perpendicular coating strategy on the test object having a 0.2 mm layer height setting. By jetting droplets with a horizontal velocity the side of the filament can be coated.

To illustrate what is going on when droplets hit the filament and to show that it is possible to aim droplets on a contour with the prototype machine a cross is printed on a horizontal wall. This is shown in Figure 3-21. The cross has a single droplet horizontal line and a two droplet wide vertical line. On the vertical axis, every filament contour has a droplet. If the filament line is stacked more to the back, the droplet appears more on the outer face of the filament and vice versa.

On the horizontal line the imperfect alignment of inkjet and filament shows droplets on the inner and outer surface of one filament line. Outer surface droplets appear elongated and sometimes even two or three droplets can be seen, where one droplet would be expected. The shape of a droplet leaving the inkjet is shown in Figure 3-22. In frame 7 of Figure 3-
22 a second droplet is almost formed. Formation of secondary droplets could explain why sometimes multiple droplets appear when only one droplet is expected. The elongation of droplets on the filament surface is due to the smearing of the droplet when it lands on the smooth curved surface of the filament.

Figure 3-23 shows how the perpendicular coating method compares to the parallel coating method. The figure shows the objects of Figure 3-20 and Figure 3-17. It can be seen that the best surface of the perpendicular coated object is much darker than the best surface of the parallel coated object. Even though the perpendicular coating is limited to printing in a grid whereas parallel can draw continuous lines, the ink on the outer surface of the filament creates a much darker object.

Besides a darker coating, an advantage of perpendicular coating over parallel coating is that the inkjet head does not have to be tilted in order to coat the side of the filament.

3-2-4 Oozing

By depositing multiple droplets on the same position droplets of ink will start to ooze over the filament surface, possibly coating the sides of the filament. This method is illustrated in Figure 3-24.
Figure 3-23: Perpendicular and parallel coating methods compared. Both objects have a 0.2 mm layer height. Top: perpendicular coating. Bottom: parallel coating.

Figure 3-24: The oozing coating strategy. A droplet of ink oozes along the surface of the filament, coating the side of the filament surface.

The filament gets a perceived color pixel by pixel. This is shown in Figure 3-25. By overlapping oozing droplets the bands of the CMYK components can be altered, achieving a range of perceived colors.

Figure 3-25: Giving a pixel a perceived color using the oozing coating method.
By applying 7 coating layers and curing the ink on the 8th layer, ink droplets start to form on the filament. If a next layer is deposited the ink droplet is pressed out from between the filaments. This is can be seen in Figure 3-26. Due to the limitations imposed by the prototype machine this effect can not be explored beyond 8 layers, but it is expected that after a certain amount of layers the ink will indeed start to ooze over the filament. Too little ink can be applied by the prototype machine and the ink is too viscous to achieve an oozing effect. This could be helped by lowering the viscosity of the ink or increase the amount of ink deposited.

![Figure 3-26: Oozing experiment for 7 layers of ink, curing on the 8th layer.](image)

In the test subject of Figure 3-26 droplets of ink are pushed outward by application of the next layer of filament. If droplets are pressed out between two layers of filament it is likely droplets are smeared over the top surface as well due to the application of the next filament layer. Smearing has to be prevented as explained in Section 3-1-3. It occurs because the lower ink layers are not cured because the top ink layers absorb most of the UV light. A more powerful UV lamp or curing for a longer period of time could solve this issue.

Applying 7 or more layers of ink increases the coating time compared to the other methods which only require one layer of ink. This can be solved by having an inkjet that is able to jet a sufficient amount of ink in one swath.

### 3-3 An Additional Coating Method: Absorption

A promising coating method which could not be investigated with the current set-up is the absorption coating method.

If ink deposited on the filament would be absorbed by the filament hopes are an area over the entire circumference of the filament is colored. This is illustrated in Figure 3-27. A droplet deposition area can be considered one pixel, as shown in Figure 3-28. By depositing different amounts of CMYK ink components this pixel is given a perceived color.

This method proved promising in an early experiment with the material Polyvinyl Alcohol (PVA) and water based ink of a single color. PVA is used as a support material in 3D printing and is water soluble.

A major challenge using this method is finding a suitable combination of ink and filament material. The ink needs to be absorbed before the next layer is deposited, otherwise it will...
be smeared when the next layer is extruded. The ink also needs to absorb in the vertical direction rather than the horizontal direction to keep equal pixel sizes. It is unknown how the CMYK components will mix when absorbed. This has to be investigated in order to get the right color in the filament.

![Diagram](image)

**Figure 3-27:** Coloring the filament by ink absorption.

The absorption method will limit the material types that can be used in the FC HD FFF machine because the material needs to be able to absorb ink.

### 3-4 Discussion

Using parallel coating a FC HD image was printed on an FFF object using UV curable ink, proving the FC HD FFF concept. Because the frog is coated from the top, the image quality differs depending on the angle the object is viewed from. To overcome this issue the coating methods perpendicular coating and coating by oozing were investigated.
Discussion

**Perpendicular coating** Of all methods, perpendicular coating is the most promising. In contrast to absorption and oozing there is a high degree of control over where individual droplets, and thus color pigments, are placed on the filament. This means this method is not bounded by pixels, allowing a possibly high resolution image quality without having to make the layer height very small. Whether this is indeed the case will turn out once full control over the inkjet is possible. The black grid printed on the test block created a darker surface than the best surface printed using the parallel method. A parallel printed line is continuous whereas a perpendicular line is interrupted. If the white space would be filled by a second swath the perpendicular line would be even darker.

**Oozing** The desired oozing effect was not achieved in prints made with the prototype machine. The ink used was too viscous and the amount of ink required for an oozing droplet was too much to drop using the prototype machine. Several studies have to be conducted to uncover the true potential of this method. One involves finding an ink that has the right viscosity in combination with a jetting device that can jet the right amount of ink in one droplet to achieve a straight band of color when the droplet oozes along the filament surface. It is suspected that the surface roughness of the printed material will play a role here as well. This means a suiting material or class of materials has to be found.

**Absorption** Using the current prototype it is not possible to fully investigate the potential of the absorption method. Several studies must be conducted before the true potential of this technique is uncovered. These studies include a material study to find the best ink/filament material combinations, an absorption study to see how ink is absorbed by the material and whether it is possible to create and control a full color spectrum using CMYK base colors. To overcome the material limitation a multi-material printer could be used to print the bulk of the object of a material of choice and the outer contour of the ink absorbing material [11].

**Method of choice** Only the parallel coating method is proven to be able to print FC imagery on 3D printed objects. The perpendicular coating method shows an intensity improvement over the parallel method and based on the observations of this section it is likely that the perpendicular method is able to print colors as well, because it is possible to place individual ink droplets. An advantage of perpendicular coating over parallel coating is that the CD remains leveled to coat the sides of the contours. This allows the coating of large flat surfaces. The top layer of most 3D objects is a large flat surface, for example.

Because of the before mentioned advantages the perpendicular coating method is believed to hold the highest potential for success. It shines in its simplicity compared to the oozing and absorption methods. Therefore, it is used as the default printing method throughout the rest of this thesis project.
Chapter 4

Grouping

In the previous section a feasible coating strategy, perpendicular coating, was identified for a Full Color (FC) High Definition (HD) Fused Filament Fabrication (FFF) printer. Coating will increase the total printing time dramatically. In the worst case, every contour of a 3D object will have to be coated. If coating a contour would take 1 second, an object like the Nautilus Gear (Appendix A-2), which can be expected to be printed with the machine, will take an extra 7 hours to complete. Compared to a 7 minute printing time for 3D printing the object alone, this is a significant increase. This section and Chapter 5 aim at reducing the coating time by optimizing the number of coating swaths needed to coat a layer and reduce the distance the inkjet has to move between swaths by re-arranging the order in which they are executed (sorting). Computing times are taken into consideration as well. If a coating time reduction of 1 hour would cost 2 hours hours to compute the net printing time is increased.

Algorithms presented in this chapter and Chapter 5 are designed for the prototype machine with a rotating bed installed in order to coat all surfaces of the 3D object. Perpendicular coating is used as a coating method. One swath takes 6 seconds to execute and contour lines can only be coated by swaths from right to left on the X-axis. Because in the prototype the Coating Device (CD) is an inkjet system, the term 'inkjet' refers to the CD.

Section outline This section starts off with an explanation of how contours are coated in the prototype. A notation on how contours are represented mathematically is given. This notation is used in the grouping and sorting algorithms. With a mathematical representation of the contours a grouping algorithm is developed, minimizing the amount of swaths needed based on physical constraints imposed by the inkjet device. The effect of these constraints on the coating time and optimization quality is investigated. In Chapter 5 the groups are reordered to reduce the traveling time between swaths.

Data flow In Chapter 2 it was explained that the Arduino is at its limits regarding computing power. Solving optimization algorithms is therefore done offline as part of the slicer on the Personal Computer (PC) of the user.
Figure 4-1 shows when the optimization steps happen. The slicer first slices GCODE from the STL as in regular 3D printing. Contour data is extracted in the pre-processing step and stored in $\bar{C}$. The contours are grouped in and stored in set $S$. The groups are ordered by the sorting algorithm. This sorted list of groups is stored in $S^*$. The original GCODE is augmented with inkjet movements between layers. How this GCODE is printed using the prototype machine is explained in Appendix B.

Grouping  The CD nozzle array can be wider than a contour. This allows some contours to be coated in one swath if certain criteria are met. On the prototype machine the average coating time of one single contour is about 6 seconds (Not taking into account the time it takes to align the CD). Every coat move less would mean a decrease in coating time of about 6 seconds. Contours are grouped in sets that can be coated in one swath. A minimum amount of swaths is found by formulating the problem as a bin packing problem. The bin packing problem is formulated as a Mixed Integer Linear Programming (MILP) problem which can be solved by a MILP solver algorithm. MILP is compared to a First Fit Decreasing (FFD) heuristic to compare its performance against a more simplistic algorithm.

This section shows the significant coating time decrease when multiple contours are printed in a single swath. Some parameters of the CD are unknown. The effect of these parameters on the quality of solutions found by the minimization is investigated. For large data sets the MILP algorithm has trouble finding a solution in feasible time. Therefore the data has to be split into partitions. For some large data partitions the computing time outweighs the time saved by group reduction. A trade-off between computing time and group reduction is uncovered and compared to the performance of a FFD heuristic.

4-1 Coating a Contour

To coat all contour lines in the same fashion the 3D object will have to be rotated. This is achieved by rotating the bed on which the object is placed. The original bed of the Creatr
HS is replaced by a rotating disc. Figure 4-2 shows a schematic top view of the prototype printer with the rotating bed installed.

![Figure 4-2: Top view schematic of the prototype with the rotating bed installed.](image)

In Figure 4-3 the coordinate frames of the prototype printer are shown. Contours of a layer of the Nautilus Gear (NG) model are placed on the rotating bed, shown in blue. A small contour line, located at the tip of arrow 1, is about to be coated. This contour line is shown in Figure 4-4. To coat this contour line, the inkjet (CD) center has to be at the same \( y \) position as the center of the contour line and the coating direction and contour line have to be perpendicular. This is called the **aligned state** of contour 1.

To coat the next contour, indicated by arrow 2 in Figure 4-5, the inkjet has to make a translation and the bed a rotation to transform the system to the aligned state of contour 2.

Each contour has an aligned state. Most aligned states are unique, so an object with a lot of contours requires a lot of rotations and translations in order to be completely coated. Equal aligned states can be coated in one swath. Non-unique aligned states can occur in objects which have parallel contours. A ladder, for example, is symmetrical and has parallel contour lines. Curved or organic objects are far less likely to have parallel contours.

### 4-2 Notation

In this section a mathematical notation is given for contour lines and their properties. This notation is used to explain the algorithms of this chapter and Chapter 5.

The contours of the object to be coated are stored as coordinate points in the GCODE file. Contour coordinate points are given in \( XY \) coordinates for each layer. All information needed for the optimization process can be extracted from these contour lines.
Figure 4-3: Top view of the prototype machine in coordinate frames. The red FRAME is the original bed coordinate frame of the Creatr HS 3D printer. The green 3D frame is the reduced 3D printing area due to installation of the L805 inkjet system on the Creatr HS. The purple 2D frame is an A4 paper sized working area of the L805 inkjet system. The dashed circle provides a visual aid to indicate the area which both the 3D printer and inkjet system can reach in any configuration. The inkjet (the red rectangle) is aligned to a contour, indicated by the black arrow. The grey line shows the coating path of the inkjet device.

Figure 4-4: Zoomed view of a the contour line pointed at in Figure 4-3. The contour line is indicated in red. Curved contours are approximated by many small straight contour lines.
4-2 Notation

Figure 4-5: By rotating the bed and translating the inkjet the system is changed from the aligned state of contour line 1 (as illustrated in Figure 4-3) to the aligned state of contour line 2.

4-2-1 Contour Objects

A contour object \( C_i \) represents a single line of the contour. This object contains the following information:

\[
\begin{align*}
C_i &= \begin{cases} 
C^l \in \mathbb{R}^{2 \times 2} & \text{Bed coordinates of the endpoints of the line.} \\
C^c \in \mathbb{R}^2 & \text{Bed coordinates of the center of the line} \\
p \in \mathbb{R}^2 & \text{Inward facing normal vector}
\end{cases}
\end{align*}
\]

The rows of \( C^l \) contain the \((x, y)\) coordinates of the endpoints. These are extracted from the GCODE. The center of this line \( C^c \) is a coordinate point \((x_c, y_c)\).

A complete contour is the set \( \bar{C} = \{C_1, C_2, \ldots, C_{N_c-1}, C_{N_c}\}_k \). Here, \( N_c \) is the number of contour lines in a contour \( k \) of the 3D object. Throughout this report, \( N_c \) is used to indicate the total number of contour lines in a set.

4-2-2 Inward Facing Normal Vector

The inward facing normal vector is used to calculate orientation angles between contours. The calculation of the inward facing normal vector \( p \) is done in two steps. First the normal vector is calculated. This normal vector \( p' \) can point in two directions. Either inward or outward of the object:

\[
\begin{align*}
\Delta x &= C^l_{x2} - C^l_{x1} \\
\Delta y &= C^l_{y2} - C^l_{y1} \\
p' &= \{(-\Delta y, \Delta x), (\Delta y, -\Delta x)\}
\end{align*}
\]

The inward pointing normal vector is determined using the crossing numbers algorithm. A line is drawn in the direction of the vector, starting from the origin of this vector, \( C^c \). If the
number of times this line crosses an edge of the contour polygon is odd the vector is pointing to the inside of the polygon. This is vector $\mathbf{p}$. Vector $\mathbf{p}$ is normalized, as it is only used to indicate the direction. Normalizing $\mathbf{p}$ simplifies some calculations, as can be seen in Section 4-2-3.

**4-2-3 Angle Distance**

The difference in orientation of two contours $\mathbf{C}_i$ and $\mathbf{C}_j$ is given by:

$$\cos(\alpha_{ij}) = \frac{1}{\|\mathbf{p}_i\|\|\mathbf{p}_j\|}(\mathbf{p}_i \cdot \mathbf{p}_j) \quad (4-1)$$

With $\|\mathbf{p}\| = 1$, (4-1) reduces to:

$$\alpha_{ij} = \cos^{-1}(\mathbf{p}_i \cdot \mathbf{p}_j) \quad (4-2)$$

(4-2) gives the absolute shortest angle $\alpha_{ij}$ between $\mathbf{p}_i$ and $\mathbf{p}_j$.

**4-2-4 Rotation Direction**

Using (4-2) to calculate the angle, the direction of rotation is unknown. The direction from $\mathbf{C}_i$ to $\mathbf{C}_j$ is determined using mathematical operations only by calculating the cross product. The direction of rotation is given by the sign of the $k$ component of the cross product $\mathbf{p}_i \times \mathbf{p}_j$. Because $\mathbf{p}_i$ and $\mathbf{p}_j$ are on the same plane, their $k$ components are equal to zero. The $k$ component of $\mathbf{p}_i \times \mathbf{p}_j$ is given by:

$$dir(\alpha_{ij}) = \text{sign}(\mathbf{p}_{i,x}\mathbf{p}_{j,y} - \mathbf{p}_{i,y}\mathbf{p}_{j,x}) \quad (4-3)$$

Subscripts $x$ and $y$ indicate the $x$ and $y$ component of a vector respectively.

**Definition.** A rotation is to the left if $dir(\alpha_{ij}) >= 0$ and to the right if $dir(\alpha_{ij}) < 0$.

**4-2-5 Aligned State**

An important state of the system is when $\mathbf{C}_i$ is aligned perpendicular to the coating direction. In this state $\mathbf{C}_i$ can be coated. In this situation the bed is rotated by an angle $\alpha^0_i$ from the starting configuration of which the layer was 3D printed.

To calculate $\alpha^0_i$ the angle between the starting configuration and the coating direction is calculated. The coating direction is always to the left. This is expressed by $\mathbf{p}_0 = \begin{bmatrix} -1 & 0 \end{bmatrix}$. $\alpha^0_i$ is calculated using (4-2) and (4-3) with $\mathbf{p}_i = \mathbf{p}_i$ and $\mathbf{p}_j = \mathbf{p}_0$.

Rotation of contours is done by multiplication with a 2D rotation matrix:

$$\mathbf{R}(\alpha_{ij}) = \begin{bmatrix} \cos(\alpha_{ij}) & -\sin(\alpha_{ij}) \\ \sin(\alpha_{ij}) & \cos(\alpha_{ij}) \end{bmatrix}$$
4-3 Constraints

Contour lines can be grouped if they lie close together such that they can be coated in one swath and their orientation falls within certain bounds of the orientation of a base contour line $C_i$. These two constraints, the distance constraint and angle constraint, are used to identify contour lines which can be grouped together.

Constraint sets are calculated as a pre-processing step of the MILP solver. This is because the distance constraints have to be calculated by aligning the contour lines to the CD. This introduces nonlinear sine and cosine terms due to rotations needed for the alignment. By calculating these constraints beforehand nonlinear terms are eliminated in the MILP. By combining the angle constraint set and the distance constraint set all constraints are contained in one compact set. This reduces the number of constraint equations in the MILP problem.

4-3-1 Distance Constraint

From the aligned state of $C_i$ a band can be drawn in the moving direction of the coating device. The width of this band is the nozzle array width of the CD. All contours that fall completely within this band can be reached by the CD in a single swath if $C_c$ is at the center of the CD. This is illustrated in Figure 4-6. The CD nozzle array width is defined as $d_{CD}$. Upper and lower bounds, $b_u$, $b_l$, on the reach of the CD are given in bed coordinates by:

$$b_{u,i} = \left( R(\alpha_{0i}) C_c^i \right)_y + \frac{1}{2} d_{CD}$$

$$b_{l,i} = \left( R(\alpha_{0i}) C_c^i \right)_y - \frac{1}{2} d_{CD}$$

Subscript $y$ of $\left( R(\alpha_{0i}) C_c^i \right)_y$ indicates the $y$ component of the rotated $C_c^i$. All contours $C_j$ rotated by $R(\alpha_{0i})$ with all $y$ components within the bounds belong to the set $D_i$:

$$C_j \in D_i \text{ if } b_{l,i} \leq \left( R(\alpha_{0i}) C_j^i \right)_y \leq b_{u,i}$$

4-3-2 Grouping Angle Constraint

There is a limit to the angle under which a contour line can be coated by the CD. The effect of this angle on the grouping results is investigated in Section 4-6-3.

If the orientation of $C_j$ differs less than a specified grouping angle $\alpha_{max}$ from $C_i$ it belongs to the set $A_i$:

$$C_j \in A_i \text{ if } \alpha_{ij} \leq \alpha_{max}$$

This is shown schematically in Figure 4-7. In Figure 4-8 the angle constraint is shown on the largest NG contour isle. Here the same $C_i$ as in Figure 4-6 to determine which contours fall within the constraint bounds.
4-3-3 Potential Group Set

For each contour line in a contour layer grouping candidates are determined and stored in a set $P_i$. Index $i$ indicates that $P_i$ is constructed of contour lines that fall within the distance
and angle constraints calculated from the aligned state of $C_i$. For each $C_i$ a $P_i$ exists so the cardinality of the complete potential group set $P$ is $N_c$: $P = \{P_1, \ldots, P_{N_c}\}$. $C_i$ can belong to $P_j$ and is always a member of $P_i$: $C_i \in P_i \forall i$.

The contents of $P_i$ are the contour lines which fall within the distance constraint bounds and angle constraint bounds:

$$P_i = C_i \cup (A_i \cap D_i)$$

A visualization of $P_i$ is given in Figure 4-9.

### 4-4 Bin Packing Problem Formulation

The grouping of contour lines into clusters that can be coated in one swath can be seen as a Variable Size Bin Packing Problem (VSBPP). The basics of the standard Bin Packing Problem (BPP) are simple: suppose we have a fixed number of items with each a volume $V_i$ that have to be placed in a certain number of bins with a fixed volume $V_B$. Then we want to place the items in the bins such that we end up using a minimum amount of bins and that the cumulative volume of the items placed in the bin does not exceed $V_B$.

The contour lines can be viewed as items and groups that can be coated in one swath as bins. The volume constraint has been reformulated into the sets of $P$. In stead of placing items in a bin as long as their cumulative volume does not exceed $V_B$, it is known beforehand which items are allowed to be placed inside a certain bin.

The VSBPP is a combinatorial optimization problem. This problem is Nondeterministic Polynomial Time (NP)-hard [12][13]. This means that the problem is at least as hard as any
NP problem. An NP problem is a problem which can be solved in polynomial time by a deterministic Turing machine.

In literature, numerous optimal solution and heuristic algorithms have been studied to solve the standard bin size packing problem [14]. The variable bin size bin packing problem is less well studied [15]. In this thesis, two algorithms which allow and intuitive transformation from a standard size to a VSBPP are investigated. A computationally light and simple first fit algorithm is compared to a MILP approach.

**First Fit Decreasing** The FFD algorithm sorts the bin sizes in decreasing order and places contour lines in the first bin it fits in until all contour lines are assigned to a bin. Literature provides a bound on the performance of FFD on a standard bin packing problem of $\frac{11}{9} (B_{opt}) + 4$ [16]. It is not proven if this bound holds for a VSBPP.

**Integer Linear Programming** The VSBPP can be formulated as an integer linear programming problem. Integer solutions are often of very good quality, but finding an optimum can become difficult as decision variables of the optimal solution of the relaxed Linear Programming (LP) problem lie close to unity [17].
The VSBPP is formulated as a MILP:

\[
\begin{aligned}
\text{minimize} & \quad B = N_c \sum_{j=1}^{N_c} y_j, \\
\text{subject to} & \quad \sum_{i=1}^{N_c} x_{ij} \leq N_c y_j, \quad \forall j \in \{1, \ldots, N_c\} \\
& \quad \sum_{j=1}^{N_c} x_{ij} = 1, \quad \forall i \in \{1, \ldots, N_c\} \\
& \quad p_{ij} x_{ij} = 0, \quad \forall i \in \{1, \ldots, N_c\}, \forall j \in \{1, \ldots, N_c\} \\
& \quad y_j \in \{0, 1\}, \quad \forall j \in \{1, \ldots, N_c\} \\
& \quad x_{ij} \in \{0, 1\}, \quad \forall i \in \{1, \ldots, N_c\}, \forall j \in \{1, \ldots, N_c\}
\end{aligned}
\] (4-4)

The constraint of (4-5) activates a group \( j \) if it contains a contour \( C_i \), by switching the binary variable \( y_j \) (4-8) to 1. The sum of active groups is the objective value of (4-4) which is to be minimized.

Which contour belongs to which group is given by a binary variable \( x_{ij} \) (4-9). Subscript \( i \) is a pointer to a contour \( C_i \) and subscript \( j \) points to which group \( C_i \) belongs. The constraint of (4-6) guarantees that every contour is assigned to one group only, but multiple contours can be assigned to a group.

The constraint of (4-7) is the grouping constraint. In Section 4-3 a constraint set \( P \) was constructed which contains the intersection of the distance constraint and angle constraint sets. The reason for this is because by pre-determining which contours can be stored in which groups non-linear terms are eliminated in the MILP problem formulation. \( p_{ij} \) is defined as:

\[
p_{ij} = 0 \quad \text{if} \quad C_i \in P_j \quad \text{else} \quad p_{ij} = 1
\] (4-10)

Equations (4-7) and (4-10) constrain which \( C_i \) can belong to which group by satisfying the grouping constraints stored in \( P \).

## 4-5 Implementation

In this thesis, computations are done using MATLAB. Equation 4-4 is solved using the MILP solver \( X = \text{intlinprog}(f, \text{intcon}, A, b, Aeq, beq, \text{LB}, \text{UB}) \):

\[
X = \text{intlinprog}(f, \text{intcon}, A, b) \text{attempts to solve problems of the form}
\]

\[
\begin{align*}
\text{min} \ f^\prime x & \quad \text{subject to:} \quad Ax \leq b \\
x & \quad Aeq x = beq \\
\text{lb} \leq x \leq \text{ub} \\
x(i) \text{integer, where } i \text{ is in the index vector intcon (integer constraints)}
\end{align*}
\]

It is important to note that using the formulation of (4-4), vector \( x \in \mathbb{Z}^{N_c(N_c+1)} \), \( A \in \mathbb{Z}^{N_c \times N_c(N_c+1)} \) and \( A_{eq} \in \mathbb{Z}^{N_c+1 \times N_c(N_c+1)} \). The matrix size depends on \( N_c \) quadratically.
Figure 4-10: The MILP algorithm run on the largest contour isle of the Monte San Pietro model (Appendix A-1). Background processes running on Windows disturb computing time polling. Therefore, each poll is repeated 10 times. Each box represents a set of 10 repeated polls on the same data set. The boxplot shows the median as a red line and the mean as an asterisk. 25th and 75th percentile of data are the lower and upper blue lines of the blue box. Whiskers indicate the minimum and maximum of the data. $N_c = 20$ has the largest spread of data of 0.1% of the mean for $2\sigma$. The figure shows the exponential increase of the median of the computing time for increasing $N_c$. The sets are subsets of the complete contour isle, and each smaller set is a subset of a larger set. The test is run on an ASUS UX305F Notebook PC, detailed in Table 4-1.

<table>
<thead>
<tr>
<th>CPU</th>
<th>Intel(R) Core(TM) M-5Y10c CPU @ 0.80GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Memory</td>
<td>4Gb @ 1.6GHz</td>
</tr>
<tr>
<td>Physical Memory available for optimization</td>
<td>0.94Gb</td>
</tr>
<tr>
<td>Virtual Memory</td>
<td>7.2Gb</td>
</tr>
<tr>
<td>OS</td>
<td>Windows 8</td>
</tr>
</tbody>
</table>

Table 4-1: ASUS UX305F Notebook PC.

This causes the computing time to rise exponentially as $N_c$ increases. This is demonstrated in Figure 4-10.

The optimization returns a set $S = \{V_k, \ldots, V_m\}$. $S$ is a set of disjoint groups $V_i$ satisfying
the following conditions:
\[ \bigcap S = \{\}, \]
\[ C_i \in S, \forall C_i \in \bar{C}, \]
\[ \forall V_i \in S, \exists P_j : V_i \subseteq P_j \]
Subscript \( k \) of \( V_k \) indicates that this group uses \( C_k \) as a base contour line to align the CD to. From this base contour line the entire group can be coated in a single swath. Each \( C_i \) is assigned to one group and one group only. Contour lines are grouped in such a way that there is always a \( P \) of which the group is a subset.

**Solution strategies**  
`intlinpog()` takes the following steps to come to a solution.

- **Step 1**: a pre-processing step removes redundant constraints.
- **Step 2**: Find a lower bound on the solutions by solving the problem as a linear programming problem.
- **Step 3**: Mixed integer pre-processing checks feasibility of the solutions found in step 2, strengthens or removes redundant inequalities, tightens bounds and fixes integers wherever possible.
- **Step 4**: Add cutting planes to restrict the feasible area closer to integer solutions.
- **Step 5**: Apply heuristics to find a feasible solution. This places an upper bound on the branch and bound process.
- **Step 5**: Use branch and bound to find a minimal solution satisfying the integer constraints

Default solver settings are used during testing except for the `NodeSelection` setting which is set to a minimum objective function. The branch and bound process is illustrated in Appendix Section B-3.

### 4-5-1 Set Partitioning

Contours isles can contain a large amount of contour lines. Optimization over a large data set comes with a long computing time when solving the MILP problem. The computing time can be reduced by dividing the set into smaller subsets. Suppose the number of function evaluations is \( N^2 \), then splitting this set in two parts would result in \( 2 \left( \frac{N}{2} \right)^2 < N^2 \) function evaluations. This can be applied to the optimization as well. This will come at the cost of a less optimal solution quality, as the optimization algorithm will be limited to a smaller subset, whereas solutions found overlapping different subsets are neglected.

Partitioning happens on a first fit basis. \( \bar{C} \) is divided up in partitions \( \bar{C}_p \) with \( |\bar{C}_p| = N_p \). The remainder is a partition \( \bar{C}_{pr} \) with \( |\bar{C}_{pr}| < N_p \). The total number of partitions is \( N_P \).
4-5-2 Partition Merging

Each partition has an optimal grouping set $S_p$. A complete solution is the union of all partition solutions $S = S_{p,1} \cup \cdots \cup S_{p,N_P}$. Sometimes it is possible to merge the last group of $S_{p,i}$, $V_{end}^i$, and the first group of $S_{p,i+1}$, $V_{1st}^{i+1}$. The merging is possible if there exists a $P_j$ for which $V_{end}^i \cup V_{1st}^{i+1} \subseteq P_j$. If multiple $P_j$’s satisfy this condition, the $P_j$ with the lowest subscript index defines the base contour of the merged group.

4-6 Experiments

The algorithm is tested on contour isles of 3D objects that are bigger than the prototype is able to print. This is done to test the algorithm to its limits. If these contour isles can be processed, smaller contour isles can be processed as well. The largest contour isles of three 3D models, Monte San Pietro (MSP), NG and Charlotte Web (CW) highlighted in Appendix A are used for testing. These models were chosen based on their size, complex shape or large number of contour isles. The skull model is not used since it contains a large amount of small contour isles which are too small to test the algorithm to its limits.

The performance of the MILP approach is compared to the FFD approach in terms of total print time reduction. This includes the time reduced by grouping and the time it takes to find a solution. Because the data is too large for the MILP to solve it has to be partitioned. To find a suitable partition size $N_p$ is varied to test its effect on the MILP solution quality. This is treated in Section 4-6-2.

$\alpha_{max}$ is not fixed. This will be a quality setting in the commercial machine. What the effect is of increasing $\alpha_{max}$ is treated in Section 4-6-3

4-6-1 Set-up

Scripts are run on the machine specified in table 4-2. This is a modern desktop computer. During experiments the only process running is MATLAB. Windows always has some background processes running. These influence computing time polling. Therefore, tests involving computing times are repeated 10 times in order to get a more accurate polling.

| CPU | Intel(R) Core(TM) i5-4670K CPU @ 3.40GHz |
| Physical Memory | 2x 4Gb @ 1.6GHz |
| Physical Memory available for optimization | 4.78Gb |
| Virtual Memory | 39.74Gb @ 7200rpm - 210Mb/s |
| OS | Windows 7 |

Table 4-2: Experiment testing machine.

4-6-2 The Effect of Partition Size on Solution Quality and Computing Time

In section 4-5-1 an unknown parameter $N_p$, the partition size, was identified. In this test the effect of increasing the partition size on the solution quality is investigated. By increasing the
partition size, the algorithm has more freedom to explore solutions. This comes at the cost of a longer computing time. The goal of this experiment is to find a trade-off between computing time and solution quality of the MILP approach by altering the partition size parameter $N_p$.

![Graphs showing computing time and reduction percentages for varying partition size $N_p$.](image)

**Figure 4-11:** Computing time and reduction percentages of the MILP approach for varying partition size $N_p$. Test are conducted using $\alpha_{max} = \frac{\pi}{16}$ rad.

The computing time as a function of $N_p$ is shown in the left column of Figure 4-11. The test is run up to a partition size of $N_p = 122$ because beyond this partition size the test computer is unable to solve the problem due to memory limitations. The test starts at $N_p = 2$ because for $N_p = 1$ only the merging algorithm is active.

The right column of Figure 4-11 shows the percentage of remaining nodes after grouping. This is a percentage of the absolute worst case scenario without grouping when every contour line is coated individually. Table 4-3 shows the extreme values of the figures in Figure 4-11. The lowest computing time is found for $N_p = 22$ for all models. The computing time of
the merging algorithm and data pre-processing steps decreases linearly as a function of $N_p$. Computing times increase exponentially as a function of $N_p$. At $N_p = 22$ the sum of these functions is at a minimum.

The highest computing times are found at $N_p = 122$ for all models. This corresponds with the expectation that computing times increase exponentially as a function of $N_p$. It is expected that the lowest node reduction percentages are found at these values as well. This is not the case. Minimal percentages lie at other large $N_p$ values. This is due to breaking up large solution bins by unfortunate partitioning, but also due to the remainder partition becoming of a size that results in poor grouping. The CW has a gap of 2.3% between the remaining node percentage at $N_p = 122$ and the lowest remaining node percentage at $N_p = 102$. The NG has a gap of 0.9% and MSP behaves as expected with a gap of 0%.

The computing times lie scattered along an exponentially increasing path. This scattering is not due to background processes as can be seen by the size of the boxplots of Figure 4-11 and Figure 4-10. These experiments have been repeated numerous times and these outliers lie at the same partition size every repetition. It is most visible in computing time outliers at $N_p = 92$ of MSP, $N_p = 112$ of NG and $N_p = 92$ of CW. The partition of $N_p = 92$ of MSP is visualized in Figure 4-12. The branch and bound phase of the MILP solver is timed and plotted for each partition. It can be seen that for the yellow partition it takes a long time to find an integer optimal solution. If this is compared to the branch and bound time on partitions of $N_p = 82$ and $N_p = 102$ in Figure 4-13 it can be seen that optimal integer solutions are found much faster in the same region, even for the larger $N_p = 102$. This is because the unfortunate partition of $N_p = 92$ has a lot of decision variables in the optimal solution of the LP relaxation that are a fraction of unity, and it may not be so easy for heuristics to find the integer optimal solution[17].

Grouping by MILP does not always reduce the nodes more than the merging algorithm. This is seen in the remaining nodes of NG in Figure 4-11. Here, $N_p = 12$ yields a less optimal reduction than $N_p = 2$ where the merging algorithm has a relatively larger role in the computations.

### Total Time Reduction

From the results of the experiments of Section 4-6-2 it is clear that a larger node reduction comes at the cost of a longer computing time. This raises the question if a maximal node reduction is desired. If the time it takes to compute a solution is longer than the time reduction in printing, the net result is worse than before the optimization.
Figure 4-12: A visualization of partitions of the biggest contour of the Monte San Pietro model for $N_p = 92$. Each partition has a color. Corresponding colored boxes behind the contour partitions create a more clear image of the partitions. Branch and bound computing times are displayed near each partition. It can be seen that the yellow partition has the longest branch and bound time, whereas the red partition has a 0 s branch and bound time. This means that an optimum was found before the branch and bound phase.

Figure 4-13: A visualization of partitions of the largest contour isle of the Monte San Pietro model for $N_p = 82$ and $N_p = 102$. Branch and bound computing times are displayed for each partition.
It takes the prototype machine roughly 6 seconds to make a single 2D print move. The reduction percentage of the original printing time is given by:

\[ \eta_{redux} = \left( 1 - \frac{t_p n_{red} - t_{cmp}}{t_p N_c} \right) \times 100\% \]

Where \( \eta_{redux} \) is the total time reduction percentage, \( t_p \) the time it takes to make a 2D print move, \( n_{red} \) the number of remaining nodes after grouping and \( t_{cmp} \) the computing time, in seconds.

For the computation of reduction percentages of the MILP solution as a function of \( N_p \), reduced nodes and computing times found in Section 4-6-2 are used.

![Figure 4-14: Total time reduction percentages. The MILP solution displayed as box plots with their mean as an asterisk is compared to the FFD solution which is shown as a straight line because it is independent of \( N_p \).](image)

Figure 4-14 shows the time reduction percentages of the MILP as a function of \( N_p \) on the largest contour isles of MSP, NG and CW. The FFD result is plotted as a straight line as it is independent of \( N_p \). Maximum reductions and the range for which the MILP approach results in a higher reduction than the FFD approach are shown in Table 4-4.

It can be seen that partition size for maximum time reduction and maximum node reduction (see Table 4-3) do not correspond. This is due to the long computing time to find the maximum node reduction. The maximum time reduction is an optimal trade-off between...
computing time and node reduction for this particular contour isle. The optimal \( N_p \) differs per model. Ranging from \( N_p = 82 \) for NG to \( N_p = 112 \) for MSP.

If the maximum time reduction of MILP is compared to the FFD solution it can be seen that MILP finds a 2.28\% up to 2.42\% better solution. However, for the CW contour isle, the FFD finds a solution which is a 3.89\% improvement over the most optimal MILP solution. This is because the CW has a lot of contour lines compared to MSP and NG. The FFD computing time scales better for larger data sets than the MILP computing time.

For NG the reduction percentage is more for \( N_p = 2 \) than \( N_p = 12 \). This does not correspond with the trend of the rest of the curve. This outlier was seen in Section 4-6-2 as well and is due to the merging algorithm finding a better reduction than MILP for small \( N_p \).

### 4-6-3 Effect of the Grouping Angle on Node Reduction

In the previous experiment an 84\% time reduction was achieved. This high reduction is partly due to the grouping angle that was used in the experiments. Figure 4-15 shows the effect of increasing \( \alpha_{\text{max}} \) on the remaining node percentage and accompanying computing times when using the MILP grouping approach.

The remaining node percentage shows an exponential decrease as \( \alpha_{\text{max}} \) increases. The angle constraint is the most constraining of the angle and distance constraints. Increasing \( \alpha_{\text{max}} \) is relaxing the angle constraint, allowing for larger group sizes and thus less remaining nodes after grouping.

The computing time shows that for all test models \( \alpha_{\text{max}} = \pi/64 \) has the lowest computing time. This is due to the fact that a small \( \alpha_{\text{max}} \) constrains the group sizes to be close to 1. This means the algorithm has not much freedom to assign nodes to groups and a solution is quickly found. For the remaining \( \alpha_{\text{max}} \) computing times are scattered. \( \alpha_{\text{max}} \) has a significant influence on the bin sizes and thus the solution space of the bin packing problem. Some solution spaces allow heuristics to find an optimal solution faster than others, as was explained in Section 4-6-2.

### 4-7 Discussion

Although a MILP approach for grouping results in a higher time reduction percentage for two of the three test pieces than the FFD approach, using the MILP as a grouping algorithm in a commercial product is not recommended. The partition size range for which MILP

<table>
<thead>
<tr>
<th>Model</th>
<th>Max time reduction</th>
<th>Range MILP &gt; FFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSP</td>
<td>86.67% @ ( N_p = 112 )</td>
<td>84.25%</td>
</tr>
<tr>
<td>NG</td>
<td>90.32% @ ( N_p = 82 )</td>
<td>88.04%</td>
</tr>
<tr>
<td>CW</td>
<td>84.6% @ ( N_p = 102 )</td>
<td>88.49%</td>
</tr>
</tbody>
</table>

Table 4-4: Maximum time reductions of the MILP and FFD approaches on the largest contour isles of the MSP, NG and CW test models.

Master of Science Thesis CONFIDENTIAL B. Botterman
Figure 4-15: Computing times and node reduction percentages as a function of $\alpha_{\text{max}}$. Tests have been repeated 10 times to account for the spread in computing time. Results are shown as boxplots. The red line is the median, upper and lower blue box bounds represent the 25th and 75th percentile of data. Top and bottom whiskers represent the lower and upper bound of the data after removing outliers. Data outliers are given as red plus signs. The mean is shown as an asterisk. Grouping is done by MILP with $N_p = 62$ on the largest contour isles of MSP, NG and CW.

MILP outperforms FFD differs for each model and it is not to say beforehand what this range is and what the optimal $N_p$ is going to be.

For some data MILP can have a hard time reaching a solution. Not only is this dependent on the contour isle data itself, an unfortunate choice of $N_p$ can result in long computing times for some partitions as well.

The performance of FFD is only model dependent. The computing time is low compared to MILP. Because this algorithm functions "out of the box", meaning that no matching $N_p$ has to be found for every individual model, it is much better suited for the varying models a commercial machine is going to have to print. In this thesis a basic FFD was implemented. This algorithm can be improved upon as is shown in [18].

A user will have no idea what the optimal $N_p$ is going to be. He could end up with a very unfortunate model on which MILP has a hard time finding a solution. His computer could
be old, having only a limited memory therefore he can not use the possibly larger optimal $N_p$ should this be known. These are all factors FFD does not have to deal with making it a more suitable algorithm.

The larger $a_{max}$ can be, the more contour lines can be allocated to one group, resulting in less groups. Even for small $a_{max}$, reduction by grouping is around 50% for the test subjects. This further increases as this angle becomes larger.
After the contours have been grouped, coating time can be reduced even more by optimizing the order in which groups are processed. By formulating the sorting as a traveling salesman problem, optimal or near optimal ordering can be computed, effectively reducing the total printing time further.

5-1 Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is a well studied combinatorial optimization problem. Suppose we have a number of cities connected by roads, the TSP involves finding the shortest tour, visiting each city exactly once and ending in the starting location. Like the Bin Packing Problem (BPP), computational complexity of the TSP is NP-hard[19].

5-1-1 Notation

Cities are represented as nodes or vertices \( V \), and roads between the cities are arcs or edges \( E \). All edges and vertices belong to a graph \( G = (V, E) \). \( c_{ij} \) is a distance associated with an edge in \( E \), it represents the distance or cost of moving from \( V_i \) to \( V_j \). All edge distances are stored in a cost matrix \( C \). The cost matrix forms a basis for the TSP solvers used in this chapter. A tour is an ordering of groups where each group is visited exactly once. The aim is to find a tour that minimizes the sum of the distances.

5-1-2 TSP Solvers

There are numerous ways of finding a solution or approximate solutions to a TSP using exact algorithms or heuristics[20][21]. Exact algorithms are able to guarantee optimal solutions, but will take much longer than heuristics to find solutions as the number of cities goes up[19]. 3D models will have a number of cities varying from very few to a very large amounts of cities.
For this reason, if one algorithm is to solve all the TSP problems, it will be a heuristic. In general, a heuristic will find an answer in a shorter amount of time than an exact algorithm at the cost of not being able to guarantee that the solution is the optimum.

In this chapter two heuristics will be investigated. The Nearest Neighbor (NN) algorithm and an implementation of the chained Lin-Kernighan algorithm Concorde (CC). The NN is considered a very simple heuristic and CC a state-of-the-art implementation of a complex heuristic. This will give an indication of the range of the time reduction a TSP solver can give.

**Nearest Neighbor** As the name states, the NN algorithm picks the closest not already visited city to move to, to be next city in the tour until all cities have been visited. For metric TSP instances Hougardy and Wilde prove that the solution quality is bounded by \( \frac{\text{NN}(S)}{\text{OPT}(S)} \leq \frac{1}{4}(\log_2(N) + 1) \). Here, \( \text{NN}(S) \) denotes a NN tour found for dataset \( S \) and \( \text{OPT}(S) \) is the optimal solution.

**Concorde** The Concorde TSP solver is capable of finding optimal solutions for a large number of cities. It is an implementation of the chained Lin-Kernighan algorithm. This is an improvement on the Lin-Kernighan local search heuristic, widely regarded as a champion heuristic in solving TSP problems. Johnson and McGeoch extensively tested the algorithm. Although a bound on the solution can not be proven, the Lin Kernighan algorithm was able to find solutions within 3% of the Held-Karp lower bound for Euclidean distance test cases of the TSPLIB, a library with test problems used in scientific research involving TSP problems. In this same paper, it indirectly outperformed the NN heuristic in terms of finding a solution close to the lower bound.

Concorde is able to find an acceptable solution to TSPLIB problems with less than 2000 instances within minutes on a dated 500MHz computer. It can be used for academic purposes only.

### 5-2 Formulation

Contours are formulated into the standard TSP notation of Section 5-1-1. This is done by defining cities and edges along with the cost attached to move from city to city.

**5-2-1 Cities and Edges**

The groups of \( S = \{V_1, \ldots, V_j\} \) calculated in Chapter 4 are the vertices of the TSP.

The machine is free to rotate and translate between coating moves. This means that the system is fully connected. This means that all groups can be reached from all other groups.

**5-2-2 Cost**

Ultimately, the total printing time has to be minimized. By expressing the cost attached to an edge in terms of time units the algorithm will attempt to minimize the total traveling time.
There are two time factors in a traveling move:

- $t_{bed}$ Bed rotation time.
- $t_y$ Stepper translation time.

These factors represent the time it takes for the rotating bed and Coating Device (CD) $y$ translation to align with the next group to coat, as was illustrated in Figure 4-3 and Figure 4-5 of Section 4-1. Both moves can be executed in parallel. The cost attached to an edge is:

$$c_{ij} = \max(t_{bed}, t_y)$$
$$c_{ii} = \infty$$  \hspace{1cm} (5-1)

This is a maximum metric distance transformed to the time domain. The time it takes to move from $V_i$ to $V_j$ is the same as moving from $V_j$ to $V_i$: $c_{ji} = c_{ij}$. The time it takes to move from $V_i$ to $V_i$ is set to infinity to force the algorithm not to dwell on one group. The cost satisfies the triangle inequality: $c_{ik} \leq c_{ij} + c_{jk}$. The problem is metric in the time domain.

**Time and distance relation**

Bed motors are driven by the stepper driver. This driver is used to control the motors during 3D printing as well. In Marlin a path linking algorithm generates velocity profiles for the stepper motors to follow. These velocity profiles are followed by the stepper motor without error. If the stepper motor fails to follow this profile, missing steps are the result. These missing steps are seen back throughout the print as the feed forward nature of the steppers does not allow the correction of errors once these have been made. The firmware dictates the time it takes to make a move, not the physical world dynamics. For example, one could have big motors with a small mass attached and compare this to small motors with a big mass attached. If the firmware settings were the same, they both would follow the same velocity profile and finish the move in the same amount of time.

The velocity profile generator of Marlin is simulated to determine the time it takes to make a move. Figure 5-1 shows such a profile in two domains: the step domain and time domain. The step domain is the profile followed by the stepper driver. The step domain profile is generated from the trapezoidal velocity profile. The trapezoidal velocity profile is generated with constant accelerations.

The velocity profile is generated using the following known constants:

- **Printer Setting**: Acceleration $a$ in steps/s$^2$.
- **Printer Setting**: Max feed rate $F_{max}$ in steps/s.
- **GCODE** Distance to move $n_s$ in steps.

Dynamic constants can be translated to the step domain by multiplying with a motor dependent steps per millimeter constant.
Figure 5-1: Velocity profile generated by the stepper driver in the step domain and time domain. \( n_a \) is amount of steps to take to accelerate to \( F_{\text{max}} \). Because the acceleration is constant this is the same distance to decelerate back to zero. \( n_c \) is the number of steps a constant velocity is maintained. In the same way \( t_a \) denotes the acceleration time and \( t_c \) the constant velocity cruising time. In the time domain a piecewise linear model is followed. The step model is not linear in the acceleration and deceleration phase but will always have the set velocity in \( n_c \) if \( n \geq 2n_a \).

Movement time  The time it takes to make a move is calculated using basic kinematic equations for constant acceleration. In the generation of the velocity profile two situations can occur. When the move is long enough to accelerate to \( F_{\text{max}} \) or it isn’t. \( n_a \) is the steps needed to accelerate to \( F_{\text{max}} \). If the distance is long enough to accelerate to \( F_{\text{max}} \) the total time is \( t = 2t_a + t_c \) with \( t_a \) the acceleration time and \( t_c \) the constant feed rate cruising time. Otherwise it is the time it takes to accelerate and decelerate back to zero over the length of the move: \( t = 2t_a \). The travel time is given by equation (5-2):

\[
    t = \begin{cases} 
        2\sqrt{\frac{n_s}{a}} & \text{for } n_s < 2n_a \\
        \frac{F_{\text{max}}}{a} + \frac{n_s}{F_{\text{max}}} & \text{for } n_s \geq 2n_a 
    \end{cases} \quad (5-2)
\]

Angle distance  The rotation angle aligning \( V_j \) from the alignment of \( V_i \) is relative. This angle is \( \alpha_{ij} \), calculated as explained in section 4-2-3. The motor and rotating bed are connected with a gear ratio of \( g_b = 2 : 3 \). The number of steps a motor needs to take for a full rotation is \( n_{\text{motor}} \). The number of steps to take is \( n_{\alpha_{ij}} \):

\[
    n_{\alpha_{ij}} = \text{nint} \left( \frac{n_{\text{motor}} \alpha_{ij}}{g_b 2\pi} \right), \quad n_{\alpha_{ij}}, n_{\text{motor}} \in \mathbb{Z}
\]

The nint() function rounds to the nearest integer.
Carriage distance  The carriage distance $n_y$ is the distance the CD carriage has to move over the $Y$-axis to align to the aligned state of $V_j$ from the aligned state of $V_i$, expressed in steps:

$$n_{yij} = \text{nint} \left( \left\lfloor \left( R(\alpha^0_j)C_y^j \right)_y - \left( R(\alpha^0_i)C_y^i \right)_y \right\rfloor s_{y,motor} \right), \quad n_y \in \mathbb{Z}$$

$s_{y,motor}$ is motor steps per mm of $y$ carriage displacement.

Cost matrix  To get $t_{\text{bed}}$ and $t_y$, $n_s$ of equation (5-2) is substituted by $n_{yij}$ and $n_{\alpha_{ij}}$. For each $V_i \in S$, the cost to every other $V \in S$ is calculated using (5-1) and stored in a cost matrix $C \in \mathbb{R}^{N_c \times N_c}$. Because $c_{ij} = c_{ji}$ the cost matrix is symmetrical. The diagonal of $C$ is $\infty$.

The coating is done after 3D printing a layer. 3D printing is always done in the same bed configuration, $\alpha_{\text{bed}} = 0$. After 3D printing a layer, the CD will be on a certain location on the $Y$-axis. When coating is done, the print will start at a certain starting location on the $Y$-axis. In order for the algorithm to start and end in these configurations two extra nodes are introduced: $V_{\text{start}}$ and $V_{\text{end}}$. The cost matrix is augmented with these nodes such that $C \in \mathbb{R}^{N_c+2 \times N_c+2}$. When a tour is generated, $V_{\text{start}}$ and $V_{\text{end}}$ will have to lie next to each other. That way the tour can be opened to create a path that starts with $V_{\text{start}}$ and ends in $V_{\text{end}}$. To make it very likely these nodes will be next to each other in the tour $c_{\text{start,end}} = 0$.

5-3 Experiments and Results

To uncover a possible time reduction by sorting, sorting algorithms are tested on the layers containing the biggest contour isles used in the investigation of Chapter 4. Two algorithms are investigated: the CC solver and a NN algorithm. The performance of the algorithms is compared to a random sorting and sorting in the order the contour was printed. A sorting in the order the contour was printed is referred to as the standard tour. The algorithms are tested on absolute time reduction and a time reduction percentage taking the standard tour as a baseline.

5-3-1 Experiments

The CC algorithm is used to provide an upper bound on performance a possible solver written by Leapfrog is able to achieve. It is unrealistic to expect to achieve the same performance as CC since years of scientific research went into developing this algorithm. It has been perfected by numerous experts in the fields of combinatorial optimization and computational efficiency.

The NN is the most intuitive way of solving a TSP. This simple algorithm is used to provide a lower bound on algorithm performance.

The velocity profiles are generated using the parameters given in Table 5-1. Parameters are given in meters and revolutions. The parameters have been translated to steps to provide a better comparison between the motors. Groups are formed using Mixed Integer Linear Programming (MILP) with a partition size of $N_p = 62$. 

Master of Science Thesis  CONFIDENTIAL  B. Botterman
<table>
<thead>
<tr>
<th>motor</th>
<th>parameter</th>
<th>Value</th>
<th>Step domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y-Carriage</td>
<td>$a_y$</td>
<td>1 m/s²</td>
<td>67000 steps/s²</td>
</tr>
<tr>
<td></td>
<td>$F_y$</td>
<td>0.15 m/s</td>
<td>10050 steps/s</td>
</tr>
<tr>
<td>Bed</td>
<td>$a_{bed}$</td>
<td>5 rev/s²</td>
<td>24000 steps/s²</td>
</tr>
<tr>
<td></td>
<td>$F_{bed}$</td>
<td>1 rev/s</td>
<td>4800 steps/s</td>
</tr>
</tbody>
</table>

Table 5-1: Velocity profile parameters

**Absolute time reduction** The first experiment conducted is an absolute time reduction. This experiment shows the magnitude of the time reduced by the algorithms compared to random tours and the standard tour. 1000 random tours are generated to get an accurate impression of its time distribution. For the NN algorithm, tours are generated for the number of $V$’s in $S$. This way, all starting locations for NN search are explored, giving an indication of the average and best and worst case scenarios. For CC, 20 tours are generated.

**Time reduction percentage** The algorithm performance is expressed as the time reduction percentage the algorithm gives over the standard printing tour. Here, computing times are taken into account as well:

$$\eta_{rdx} = 1 - \frac{t_{a,r} - t_{a,c}}{t_{std}}$$

$t_{a,r}$ is the time it takes to make moves using a tour generated by algorithm $a$ and $t_{a,c}$ is the time it takes to compute a tour using algorithm $a$. $t_{std}$ is the movement time using the standard tour and $\eta_{rdx}$ is the reduction percentage. $t_{a,r}$ and $t_{a,c}$ are distributions. This is because the algorithms generate tours of varying quality. For NN this is due to the starting node used and for CC because different starting tours are used each run of CC. As was seen in Chapter 4 computing times vary each run as well. Therefore, the normal distribution of $t_{a,c}$ is used.

5-3-2 Results

The absolute time reduction for the test layers is shown in Figure 5-2a. For the layers of the Monte San Pietro, Nautilus Gear and Charlotte Web models a random tour yields the longest movement time. For the Skull model the standard tour movement time falls within the range of the random tour movement time.

A clear gap can be seen between the random and standard tours and the heuristic solvers. For the Monte San Pietro and Nautilus Gear this gap is about 60 seconds. For the Charlotte Web the gap is roughly 800 seconds and for the Skull it is about 500 seconds.

For all test models, the movement times of tours generated by NN and CC lie close together compared to the standard tour and random tours.

Figure 5-2b shows the distributions of the time reduction percentages. For the test models, time reduction percentages are achieved of 56% to 84% with the NN yielding the lowest on the Monte San Pietro model and CC yielding the highest on the skull model. The difference in the mean of the solutions is largest for the Natulius Gear model with a 2.3% difference and smallest in the skull model with a 0.5% difference. For all models, NN has the highest standard deviation.
5-3 Experiments and Results

(a) Absolute time reduction.

Figure 5-2: In Figure 5-2a lower and upper bound of the blue boxes show the 25th and 75th percentile of the data respectively. Whiskers shows the minimum and maximum of the data after outliers have been removed. The median is shown as a red line. Plus signs indicate removed outliers. Figure 5-2b shows the normal distribution of the reduction percentages.

5-3-3 Discussion

Solutions found by NN and CC only differ by a few percent. This is way better than what was expected. The CC will be very close to the optimum, but the NN is only guaranteed to lie with a fraction of $\frac{1}{3}(\log_2(N) + 1)$ from the optimum. For the Monte San Pietro (MSP) model this bound would be around 80s and for the skull this would be 300s. What can be seen is that the NN solution is close to the CC solution. It is suspected the NN algorithm performs above expectation due to the fully connected metric cost matrix of the problem. To show this, a number of random cost matrices have been generated on which the performance of both the CC and NN have been tested. Two types of matrices have been generated. One is a symmetric
random distance matrix and the other is a symmetric random distance matrix satisfying the triangle inequality, simulating a symmetric metric cost matrix. Figure 5-3 shows the excess percentage of cost(NN) over cost(CC) plotted against an increasing number of vertices. What can be seen is that in the metric case the average fraction differs from the CC solution by 26%. This is in line with results found by Johnson and McGeoch[30] for random euclidean distance matrices. Here the average was an excess of 25% over the Held-Karp lower bound. For the non metric matrix the NN solution differs at best 144% from the CC solution and 285% in the worst case. This result combined with the quicker computing time of NN compared to CC explains why the time reduction percentages of both solutions are so close together.

The Skull and Charlotte Web contours have in common that they have more contour isles than the Monte San Pietro and Nautilus Gear model. This can explain why they have a better time reduction percentage, as hopping from one contour isle to another is a relatively larger distance than moving from one contour to its neighboring contour. The 3D printer does print contour isles in a consecutive stream, but hopping between isles may be far from optimal. The algorithms do take this into account and this yields a higher movement time reduction.

---

**Figure 5-3**: Performance of the NN compared to CC for an increasing number of vertices.
Chapter 6

Conclusions and future work

6-1 Conclusions

**Proof of concept** In this thesis work the concept of Full Color (FC) High Definition (HD) Fused Filament Fabrication (FFF) was proven by realizing a prototype which is able to print the Leapfrog logo onto a vertical 3D printed wall (see Figure 1-1 and Figure 3-11). Although a promising result, it showed that coating from the top will not coat the entire outer surface of a print.

**Coating method** Using the prototype machine promising coating methods have been extensively tested on their ability to coat the entire outward facing surface of a print. Perpendicular coating is identified as the most feasible coating method due to the possibility to coat outer surfaces without tilting the Coating Device (CD) and control of droplet placement over the surface. An extra advantage of perpendicular coating over the parallel methods is that it allows contour lines to be grouped and coated in one swath. This can be used to reduce the printing time.

To coat all contour lines of a 3D object in the same fashion, the bed of the Creatr HS has been replaced by a rotating disc, rather than a rotating CD because this allows a more simple machine design.

**Grouping** One swath of the CD can coat multiple contour lines. How all contour lines are divided among swaths is determined by solving a Variable Size Bin Packing Problem (VSBPP). Although a Mixed Integer Linear Programming (MILP) approach is able to find a smaller number of groups than First Fit Decreasing (FFD), FFD is much more reliable in finding a solution because its solution quality is only dependent on the model, rather than a partition size and possible difficult solution spaces the MILP algorithm has to deal with. This makes the MILP unsuitable in a practical situation, where models to be printed are different every time the machine is used. For this reason, FFD should be used for grouping.
Grouping showed possible large time reductions (80% or more) if $\alpha_{\text{max}}$ increases. Even for $\alpha_{\text{max}} = \pi/64 \text{ rad}$ a 50% time reduction could be achieved on the test models. This shows that the possibility to group can greatly reduce the printing time. This should be taken into consideration when a definite coating method is selected for the commercial machine.

**Sorting** After contour lines have been grouped the movement time of the CD in between swaths is minimized by solving a Traveling Salesman Problem (TSP). Movement time for the test models is reduced from 55% up to 85%. The impact of minimizing the traveling time on the overall print time reduction is small compared to the impact of grouping. Movement time reduction on the test models involves minutes whereas grouping involves hours. A Nearest Neighbor (NN) heuristic was compared to the award winning Concorde (CC) solver. Performance difference between NN and CC is such that in practice a user will have to wait a few seconds more for his print to finish when a solution using NN is found. This can be neglected, making the NN a suitable movement time reduction algorithm to be used in the commercial machine.

### 6-2 Future Work

**Coating methods and color** The perpendicular coating method can be used when printing only in black. Whether this is the case if colors become involved has to be investigated. Should the perpendicular method fail, the oozing method is the next method to be investigated. If this method is disqualified as well Leapfrog can always fall back on the original parallel printing method, as this has been proven to work in this thesis. In this case a study should be conducted on printing with a tilted head to be able to coat the outward facing contour lines.

**Improve FFD performance** The performance of FFD can be improved. There are a few directions in which an improvement can be found. One improvement over FFD for an VSBPP is Iterated First Fit Decreasing (IFFD) [18]. As this algorithm is an improved version of the algorithm used in this thesis a performance increase can be expected. In [15] heuristics are proposed which can find solutions close to optimum for VSBPP. In [32] two algorithms are proposed which can outperform FFD on a standard Bin Packing Problem (BPP). Whether this holds for VSBPP should be investigated.

**Swath time reduction** A swath takes 6 seconds on the prototype machine. This is unnecessarily long and has to be reduced in the commercial machine. In the prototype machine coating only occurs in the negative direction on the $X$-axis. On the way back contours could be coated as well. This can be included in the TSP by adapting the cost matrix to include entry and exit points. These entry and exit points guarantee that droplets have the right velocity when they are jetted onto the filament. This will mean that the cost matrix will no longer be symmetrical and the performance of NN can not be guaranteed, requiring a study for a more suitable heuristic if the performance decreases too much.
Direct coating  A future iteration of the commercial machine should use direct coating as this is expected to eliminate the printing time the most.
Appendix A

Datasets

This appendix gives a detailed overview of the data used to test the grouping and sorting algorithms. The data is generated from open source models found on http://www.thingiverse.com.

3D printing times are predicted by Simplify 3D based on the settings given in Table A-1. The settings are based on regular slicer settings used to print under normal operating conditions of the Leapfrog Creatr HS 3D printer.

A-1 Monte San Pietro

Monte San Pietro is a landscape model of a mountain range in Italy. This GCODE has a complex surface with a lot of different types of curves. The number of contours and contour isles can differ per layer.

<table>
<thead>
<tr>
<th>Layer Height</th>
<th>0.2 mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default feedrate</td>
<td>9000 mm min⁻¹</td>
</tr>
<tr>
<td>Movement Speed</td>
<td>9000 mm min⁻¹</td>
</tr>
<tr>
<td>Infill</td>
<td>3%</td>
</tr>
<tr>
<td>Top Solid Layers</td>
<td>2</td>
</tr>
<tr>
<td>Bottom Solid Layers</td>
<td>2</td>
</tr>
<tr>
<td>Outline/perimeter shells</td>
<td>4</td>
</tr>
</tbody>
</table>

Table A-1: Slicer settings.
A-2 Nautilus Gear

The Nautilus gear is a mechanical part that is characterized by a large number of smooth curved contour lines. The contour lines differ in curvature throughout the object. Individual layers differ little.

A-3 Charlotte Web

The Charlotte Web is characterized by the high number of contour isles. With 2687 contours in a single isle it also has the largest isle of all data sets. The surface of the web is rough like the Monte San Pietro model. This object is larger than the build platform of the prototype printer. It is used to test the performance of the algorithms on large contours.
A-4 Wire-frame Skull

The Wire-frame Skull is an art object. Every layer is composed of a high number of small isles. This object is larger than the build volume of the prototype printer. Due to the large amount of contours and isles it is used to stress test the algorithms.
Appendix B

Full Color High Definition Prototype Machine

B-1 The Machine

The Full Color (FC) prototype machine consists of an Epson L805 inkjet carriage mounted on a Leapfrog Creatr HS 3D printer. This is shown schematically in Figure B-1. The inkjet head is referred to as the Coating Device (CD). The flat bed of the Creatr HS has been replaced by a glass disc, which can be rotated using a stepper motor. This allows the rotation of a 3D printed object such that all sides of the contour can be coated by the Epson CD in the same fashion. The original water based ink of the Epson printer is replaced by Ultra Violet (UV) ink. This is cured during printing by a UV light source mounted on the CD.

Figure B-1: Top view schematic of the prototype.
B-1-1  Epson L805

The best CD for this project would be to develop a custom CD, allowing full control over droplet size and placement. This requires a long development time as a third party manufacturer would have to make one for Leapfrog. To proof the full color concept an existing inkjet system is used. This system can be acquired fast and cheap, if a common desktop inkjet system is utilized.

Epson uses piezo driven inkjet technology in their desktop printers. A more common alternative, thermal inkjetting, can not be used because thermal inkjetting uses the thermal expansion of water to jet ink droplets and UV ink is not water based. Piezo driven inkjet systems are also used in industry. These systems are made for mass production and in general too large to be used in the Creatr HS.

**Black box**  Epson does not allow access to the L805 firmware. This means there is no control over how the L805 will behave. Also, the printer driver on the Personal Computer (PC) side is not available. Whatever halftoning patterns, color schemes or movement optimization the system uses is unknown and will remain unknown unless it is reverse engineered extensively, which might take a very long time to achieve. Only a limited amount of settings can be adjusted on the PC, like a color correction scheme or the Dots Per Inch ( DPI). Based on these settings the L805 gives the paper a certain feed rate, drops ink in unknown halftoning patterns in an unknown order. Start-up sequences can differ as well. There are a number of cleaning methods which are triggered at seemingly random times and events. This is seemingly random because they are based on internal counters and clocks which are inaccessible.

For these reasons, the data transition of an image or text on a PC to a print on paper must be treated as a black box.

Even though this system is a black box, it is possible to get the system in a certain state and print using the same series of events every time. This is exploited in Section B-1-3, where the Creatr HS and L805 are combined to make a fully automatic FC printer.

**Carriage**  Because the firmware of the Epson is not available to Leapfrog the L805 needs to be tricked into thinking it is working under normal operating conditions. The complete L805 inkjet carriage including the cleaning station, mode selection axis, controller board, motors and sensors required for operation under normal circumstances are installed on the Creatr HS. This assembly is illustrated in Figure B-2. The APG unit seen in this figure is a gearbox which, in combination with the mode switch, is used to switch states of the system. Piezo inkjet technology requires a cleaning station to keep the nozzles clean. It consists of a wiper and a waste bin which can create a vacuum around the nozzles and pump them clean using a peristaltic pump.

The L805 has two motors connected to a rotary encoder and a linear encoder. These are the two axes of this system. When printing, the linear encoder is used to determine the X-position (position on the width of an a4-paper) of the CD and the rotary encoder to determine the Y-position (position on the length of an a4-paper). The axes are not only used for printing. Using a complex series of motions in combination with the gears in the APG unit and paper in, out and mode sensors, the printer can zero its axes position, drive a peristaltic pump
system in the cleaning station, run a wide arrangement of cleaning schemes, grab a new sheet of paper and eject printed pages.

The entire paper transportation system has to be removed. The paper transportation system, consisting of a holder for the paper stack, a paper grab system able to grab one piece of paper from the stack and transportation line to the exit feed opening, makes up the bulk of the L805. This system contains crucial sensors required for a normal operation of the printer, but has to be removed if the L805 carriage is to fit the Creatr and replace paper for the printing bed.

![Schematic of the L805 carriage as installed on the Creatr HS. Motors and sensors are not displayed.](Figure B-2)

**Figure B-2:** Schematic of the L805 carriage as installed on the Creatr HS. Motors and sensors are not displayed.

**Printing Head** The L805 print head prints in cyan, light cyan, magenta, light magenta, yellow and black. The light variants were not available during testing so these colors are not used. For each color the print head has a 90 nozzle array, shown in Figure B-3.

Specifications are shown in Table B-1. It can be seen that the Creatr HS is capable of achieving the same DPI on the Y-axis because the smallest step the Creatr HS can make is 0.0150 mm which is smaller than the 0.0176 mm step the L805 makes on the Y-axis.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>max DPI Y-axis</td>
<td>1140</td>
</tr>
<tr>
<td>max mm per droplet L805</td>
<td>0.0176 mm</td>
</tr>
<tr>
<td>CD width</td>
<td>40 mm</td>
</tr>
<tr>
<td>Nozzle spacing</td>
<td>≈ 0.3 mm</td>
</tr>
</tbody>
</table>

**Table B-1:** L805 specs.
B-1-2 L805 and Creatr HS Communication

Because the L805 only uses two motors to execute all printer actions, all parts are highly dependent of each other. The controller unit uses timing windows to receive pulses of certain sensors at the right time in order to know what is going on. To have the Creatr HS start the next buffered print in the L805 the following signals have to be connected.

**Paper Feed sensor** The L805 is a consumer system so it has to be safe to be operated by someone with limited knowledge of the machine. This means that even the smallest deviation of the normal operating mode triggers errors, leading to a full stop and requiring the system to be rebooted. In many error cases, system failure is detected by missing or wrong signals within certain timing windows. One of these timing windows involves triggering the paper feed sensor during printing. This sensor is triggered when the L805 pulls a new paper from the paper stack. If the paper feed sensor is not triggered during a pull action the L805 returns an error stating that the paper stock is exhausted. Only if this pulse is timed just right will the L805 commence printing.

**Rotary Encoder** The optical 4-phase rotary encoder is connected to the same axis that pulls the paper. It is also used to drive the APG unit to switch between operating modes. This encoder contains a lot of information on the operation of the L805. Because the paper feed...
needs to be pulsed at the right moment it is crucial to know when this moment occurs. This information can be extracted by reading the encoder during printing.

**Paper reset button** Under normal operation a user would press the paper reset button after reloading the paper stack with new paper. This way the L805 knows when to resume normal printing operation.

Figure B-4 shows a schematic view of the communication channels between the Creatr HS and L805. By reading the encoder the Creatr HS can determine the exact moment the paper feed sensor and paper reset button need to be triggered in order to make a print.

**B-1-3 Activation of a L805 Print Via the Creatr HS**

Two L805 states relevant for the prototype are the "ready" and "no paper" states. From these states a print job can be executed.

Printing starts in the "ready" state. Before the printing commences, a print assignment is buffered. This can be a number of pages. Once buffering is done the L805 commences printing but soon finds it is missing the paper feed signal pulse. If this occurs the printer switches to "no paper". In "no paper", the L805 remains idle until it receives a pulse from the paper reset button. If the paper reset pulse is received the printer continues printing the paper as normal. Under normal operation of the L805, this would mean the paper stack is empty and the user has to reload paper and press the paper reset button to tell the L805 new paper is in the stack.

**Pulse Timing** If the L805 receives a signal to start printing, either by sending a job or a paper reset pulse, it does not always immediately start the printing sequence. Sometimes a cleaning cycle starts before a print and these cleaning cycles can be different, depending on timers within the L805 controller. This introduces a difficulty in timing the paper feed pulse.

Figure B-5 shows the differential of pulses when a print command is sent to the L805. When the L805 starts to print, it tries to grab a new paper(1a). This is the moment the paper feed pulse has to be sent. If the pulse misses the timing window the paper transport system will run on a low velocity discarding any paper to the exit feed tray of the printer (2a). This process is repeated a second time (1b, 2b). If it fails the second time the printer returns to 'no paper'. Figure B-5.

On the y axis of Figure B-5 the differential of pulses of the encoder wheel taken on a sampling interval of $t_s = 200 \, \mu s$ is plotted. Data is sent to a computer via a serial protocol. Data polling on the PC side happens with $t_{PC} = 10 \, ms$, this is plotted on the X-axis. This is a rough indication of time because the serial communications protocol of the Arduino has a low priority. Other processes running on the Arduino disrupt the exact 10 ms data send timing. This could be fixed by logging the data in the same interrupt function that polls the encoder and have the serial communications send the logs. The downside to this is that the polling interrupt is already running at very high frequency and any extra tasks would require reducing $t_s$. Otherwise the Arduino can not keep up and the interrupt will restart while it is still running the previous iteration, resulting in some serious artifacts in polling data.
Figure B-5: Encoder pulse differential. Encoder values are sampled with $t_s = 200$ µs. The differential of this data is polled roughly every 10 ms.

In Figure B-5 it can be seen that at first a positive differential spike is followed by a depression to values of -6 and -7 for about 20 polls. This first dip is the paper grab phase (1a). The roller is stopped for a moment and then another depression is seen to values ranging to -5 and -6. This is the paper discarding phase (2a). This process is repeated two times (1b,2b). If the paper feed pulse is sent during the paper grab phase the L805 starts printing the job.

Unique for the paper grab phase is that differential values lie between -6 and -8. No other event has this velocity. This fact is used to trigger the Arduino to send a pulse at the right time. If a series of 20 differential values is found between -6 and -8 the L805 is in the paper grab phase for sure. A 300 ms delay, found empirically, between the 20th sample and sending the paper feed pulse ensures the L805 receives the pulse well within the window.

It can be seen that the differential has some spikes in its values as if noise was present in the measurement. This however, is caused by a poor reconstruction of the L805 on the Creatr HS. The reconstruction has introduced some unwanted friction, causing the velocity to show some minor fluctuations where a constant velocity is expected.

### B-2 Printing with the Prototype

In order to use the L805 as a part of the FC machine, the right buttons have to be triggered at the right time. As explained previously, Epson has not provided the firmware of the L805. By manipulating the printer into thinking it is operating under normal printing conditions, it can still be used to drop ink at the desired positions.
B-2-1 Coordinate Frames

The prototype has three coordinate frames. These frames are illustrated in Figure B-6. Highlighted in orange is the 3D printing area of the unmodified Creatr HS printer. The purple area shows the 2D printer coordinate frame. This area is equal to an A4 paper. The dotted purple line shows the upper bound the lowest row of nozzles can reach in the 3D frame. The green area is the 3D printing area of the modified Creatr HS. Installing the CD has reduced the 3D printing area. The blue circle is the rotating bed area. The black dotted circle is the effective working area. This is the area the 3D and 2D printers both can reach.

![Coordinate frames of 3D, 2D and rotating bed](image)

**Figure B-6:** Coordinate frames relative to 3D printer frame coordinates. The purple dotted line shows the lowest point the CD can reach. The black dotted circle highlights the area both 3D and 2D devices can reach. This is the working area.

By translating print bed coordinates to A4 paper coordinates the L805 can be controlled to place droplets on desired location on the bed.

B-2-2 Printer Y-motion

There are two modes the 2D and 3D printer can be connected. 2D listener mode and fulls top mode.

**2D listener mode** In 2D listener mode the Y-axis stepper motors follow the pulses of the encoder of the L805 one to one. This means that the entire carriage moves over the bed the way a piece of paper would move through the paper transportation line of the L805. This is almost the same as printing the way the L805 was intended to print, except now the carriage is moved in stead of the paper.

The advantage of printing this way is that the black box of the L805 firmware is not hindering the outcome of the print. If a green line has to be printed at a certain position, this green
line will be printed there and it will be green. Whatever halftoning patterns or color maps used to print a green line on paper is the same as it will be on the 3D object layer.

The drawback is that the heavy accelerations involved moving the paper through the transportation system are the same for the entire carriage. One can imagine that these accelerations introduce much larger forces in the prototype system than the L805 because the mass of the entire carriage plus the print head of the 3D printer is much more than the mass of a piece of paper. Because filament contours have a diameter in the range of 0.2 mm, aiming the droplets at the exact desired location is crucial. Vibrations introduced by the forces needed to get the carriage at the right position in time make it impossible to meet this requirement.

Even with this drawback it is possible to get a print on the surface of a 3D object. Using a specific location of the 3D object on the printer bed such that the paper carriage accelerations are minimal. By limiting an image to be printed on one face of a cubic object and making image lines wider such that the accuracy no longer plays a role, a major result in this thesis work has been achieved. The frog shown in Figure B-7 is the first ever Full Color High Definition 3D print. It proves the concept of the Full Color High Definition 3D printer.

![Figure B-7: A major development in FFF: The first Full Color High Definition image on a 3D print.](image)

**Full stop mode** To overcome the problem of vibrations, the carriage can be brought to a complete standstill before printing commences. By disconnecting the encoder from the Y-steppers, the carriage can only be moved by the 3D printer.

The advantage of this method is the elimination of vibrations during printing. If no vibrations are present in the system droplets can be placed with a high enough accuracy to coat a single line of filament.

The drawback is the complete loss of control of color and halftoning patterns. Control over the thickness and amount of times the print head moves over the droplet location is greatly reduced.

Because this mode gives me most accurate droplet placement and is most likely the way 2D printing will happen in the fully developed commercial machine it is the printing mode used in the research in this thesis project.
B-2-3 Buffering Moves

To fully automate the printer, every single print command is buffered in the L805. This way all data has to be loaded only once and the 3D printer can send a paper reset signal to the L805 every time the next print assignment is required. If a 3D object is to be coated, every contour line our group will have a separate print job. This means that the printer will have to buffer a large amount of jobs. A custom slicer will have to slice the 3D object and generate corresponding print jobs needed to coat the object. An Stereo lithography (STL) format able to handle STL and color data is the STL Object (STLO) format. The buffering and data flow is illustrated schematically in Figure B-8.

B-2-4 Full Color High Definition 3D Printing

When all jobs have been buffered and GCODE is generated, the FC printing is executed in the following order:

Creatr HS - Finish layer and send a reset pulse to the L805.
L805 - Start executing the first job in the buffer.
Creatr HS - Read the paper grab velocity from the rotary encode and send a paper feed pulse to the L805.
L805 - Begin printing the job.
Creatr HS - After two seconds move the carriage to the next print location or commence printing the next layer.
**B-3 Branch and Bound**

This section describes how `intlinprog` finds the optimal integer solutions using branch and bound.

Figure B-9 describes the process.

**Step (1)** First, an initial relaxed Linear Programming (LP) problem is solved. This serves as an initial node and lower bound. Using heuristics a feasible solution is constructed. This serves as an initial upper bound. A branching rule, the maximum pseudo cost, selects which variable in $x$ raises the upper bound the most if it is fixed to its nearest integers. For the problem in this thesis, this would be 0 or 1.

**Step (2)** The branching rule constructed two new solutions, $fx_2$ and $fx_3$. Here, $x_x$ is the initial solution $x$ with $x(i) = 0$ or $x(i) = 1$. These two nodes have raised the lower bound since $fx$ is the optimal solution. The node which has the lowest objective value is selected to branch from. This is node 3 in Figure B-9. Then the process is repeated.

**Step (3)** Branching from node 3 yields lower bounds which are higher than the bound of node 2. Therefore, node 2 is the next node to be explored.

This process is repeated until one of the stopping criteria is reached:

- The algorithm exceeds the `MaxTime` option.
- The difference between the lower and upper bounds on the objective function is less than the `AbsoluteGapTolerance` or `RelativeGapTolerance` tolerances.
- The number of explored nodes exceeds the `MaxNodes` option.
- The number of integer feasible points exceeds the `MaxFeasiblePoints` option.
Figure B-9: Branch and bound of the MILP. \( u \) is an upper bound of the solution found heuristically. \( f x_\text{fixed} \) is an LP solution with one or more variables fixed.
Bibliography

turing processes: I. Process design and modeling,” Rapid Prototyping Journal, vol. 20,


modelling approaches: a critical review,” The International Journal of Advanced Manu-
facturing Technology, 2015.


properties of fused deposition modeling ABS,” Rapid Prototyping Journal, vol. 8, no. 4,


technology FDM: exploring build process variations,” Rapid Prototyping Journal, vol. 20,
no. 3, pp. 236–244, 2014.


### Glossary

#### List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>Additive Manufacturing</td>
</tr>
<tr>
<td>BPP</td>
<td>Bin Packing Problem</td>
</tr>
<tr>
<td>CD</td>
<td>Coating Device</td>
</tr>
<tr>
<td>CC</td>
<td>Concorde</td>
</tr>
<tr>
<td>CW</td>
<td>Charlotte Web</td>
</tr>
<tr>
<td>CMYK</td>
<td>Cyan Magenta Yellow Key</td>
</tr>
<tr>
<td>DPI</td>
<td>Dots Per Inch</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Electrically Erasable Programmable Read-Only Memory</td>
</tr>
<tr>
<td>FC</td>
<td>Full Color</td>
</tr>
<tr>
<td>FFF</td>
<td>Fused Filament Fabrication</td>
</tr>
<tr>
<td>FFD</td>
<td>First Fit Decreasing</td>
</tr>
<tr>
<td>HD</td>
<td>High Definition</td>
</tr>
<tr>
<td>IFFD</td>
<td>Iterated First Fit Decreasing</td>
</tr>
<tr>
<td>LP</td>
<td>Linear Programming</td>
</tr>
<tr>
<td>MILP</td>
<td>Mixed Integer Linear Programming</td>
</tr>
<tr>
<td>MSP</td>
<td>Monte San Pietro</td>
</tr>
<tr>
<td>NN</td>
<td>Nearest Neighbor</td>
</tr>
<tr>
<td>NG</td>
<td>Nautilus Gear</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
</tbody>
</table>
PLA  Polylactic Acid
PVA  Polyvinyl Alcohol
STL  Stereo lithography
STLO STL Object
TSP  Traveling Salesman Problem
UV  Ultra Violet
VSBPP Variable Size Bin Packing Problem
NP  Nondeterministic Polynomial Time

List of Symbols

\( \alpha_{ij} \)  Angle between \( C_i \) and \( C_j \).
\( \alpha_{\text{max}} \)  Maximum angle deviation to enter the constraint set.
\( C^c \)  Coordinates of the center of the contour line.
\( C^l \)  Coordinates of the endpoints of the contour line.
\( C_i \)  A contour line object.
\( \bar{C}_{pr} \)  Remainder partition.
\( D_i \)  Distance constraint set constructed from \( C_i \).
\( \eta_{rdx} \)  Reduction percentage.
\( \eta_{rdux} \)  Total time reduction percentage.
\( P_i \)  Set containing group candidates of \( C_i \).
\( p \)  Inward facing normal vector of a contour line.
\( p' \)  Set of normal vectors of a contour line.
\( x \)  X component of a geometric vector.
\( y \)  Y component of a geometric vector.
\( a \)  Stepper motor acceleration.
\( A_i \)  Angle constraint set constructed from \( C_i \).
\( b_l \)  Lower bound on the distance constraint.
\( b_u \)  Upper bound on the distance constraint.
\( C \)  Cost matrix of the TSP.
\( c_{ij} \)  Distance associated with an edge in \( E \).
\( d_{CD} \)  Coating nozzle array width.
\( E \)  Edges of the TSP.
\( F_{\text{max}} \)  Stepper motor maximum feed rate.
\( n_a \)  Steps needed to accelerate to the maximum feed rate.
\( N_c \)  Total number of contour lines in a set.
\( n_c \) Cruising distance.
\( n_c \) Cruising distance.
\( N_P \) Total number of partitions.
\( n_s \) Total moving distance.
\( n_y \) Alignment distance.
\( n_{\alpha ij} \) Number of steps needed to rotate the bed with \( \alpha_{ij} \) rad.
\( n_{motor} \) Steps to take for a stepper to make a full rotation.
\( n_{red} \) Reduced node set.
\( p_{ij} \) MILP formulation of \( P_i \).
\( s_{y,motor} \) Steps per mm of the stepper motor.
\( t_a \) Acceleration time.
\( t_c \) Constant feed rate cruising time.
\( t_p \) Time it takes to make a 2D print move.
\( t_y \) Stepper translation time.
\( t_{a,c} \) Time it takes to compute a tour using algorithm \( a \).
\( t_{a,r} \) Movement time of a tour generated by algorithm \( a \).
\( t_{bed} \) Bed rotation time.
\( t_{cmp} \) Computing Time.
\( t_{std} \) Movement time of a standard tour.
\( V_i \) Grouping with \( C_i \) as a base node.
\( x_{ij} \) Bin allocation variable.
\( y_j \) Bin activation variable.
\( \bar{C} \) Complete contour.
\( \mathbb{R} \) Set of real numbers.
\( \mathbb{Z} \) Set of integers.
\( \mathbb{R}(\alpha_{ij}) \) 2D rotation matrix around \( \alpha_{ij} \).
\( G \) Graph of vertices and edges.
\( N_P \) Partition size.
\( S \) Minimum set of groups.