Computer-Aided Process Planning for Surface Quality in Point-Wise Constructive Manufacturing by Three Dimensional Printing

by

Sang-Joon John Lee

B. S. Mechanical Engineering, Stanford University, 1990
M. S. Mechanical Engineering, MIT, 1992

Submitted to the Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy at the Massachusetts Institute of Technology

September 1996

© 1996 Massachusetts Institute of Technology All rights reserved

Signature of Author: ____________________________

Department of Mechanical Engineering
August 20, 1996

Certified by: ____________________________

Emanuel M. Sachs
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by: ____________________________

Ain A. Sonin
Chairman of Graduate Committee

APR 15 1997
Computer-Aided Process Planning for Surface Quality
in Point-Wise Constructive Manufacturing by Three Dimensional Printing

by
Sang-Joon John Lee

Submitted to the Department of Mechanical Engineering September, 1996
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mechanical Engineering

ABSTRACT

Computer-aided process planning has been implemented for high-quality, high-rate fabrication of solid freeform objects by Three Dimensional Printing. 3D Printing is a manufacturing process that produces complex parts directly from computer-aided design (CAD) models, by applying ink-jet technology to selectively bind thin, cross-sectional layers of fine powder.

The objective of this research is to develop information-driven printing enhancements that improve surface finish and dimensional accuracy, consistent with machine capability and process behavior. A key hypothesis for process planning is that the final geometry of printed parts is independent of temporal history. Experiments show that this hypothesis is useful for initial prediction of surface boundaries. The work begins with a basic understanding of the physical process, and the constraints imposed by machine architecture and printhead technology. Printing enhancements include boundary offset compensation, feature snapping, incline tracing, and the use of internal fill patterns. These techniques have been implemented in the software that converts CAD models into machine instructions. The benefits of these enhancements have been proven by printing a wide variety of complex engineering parts.

Many of the improvements employ a control mechanism called proportional deflection, in which binder droplets are placed with a resolution (~10 microns) much finer than that of printed scan lines (~200 microns). Proportional deflection enables detailed tracing of geometric boundaries in the CAD model, and thereby significantly enhances surface quality. For example, roughness on shallow inclines have been reduced from 40 microns to below 20 microns (arithmetic average of profile deviations). Dimensional accuracy in some directions have been reduced from ±100 microns to ±20 microns.

Thesis Committee
Professor Emanuel M. Sachs, Mechanical Engineering
Thesis Supervisor and Committee Chair
Professor Michael J. Cima, Materials Science and Engineering
Professor Nicholas M. Patrikala, Ocean Engineering
Acknowledgments

This work is wholeheartedly and most joyfully dedicated to my parents.

I would especially like to thank Professor Ely Sachs, for direction and inspiration over the last six years. My sincere appreciation also goes to Professor Michael Cima and Professor Nick Patrikalakis for additional guidance and input.

This research was only made possible by the tremendous help so readily given by the 3D Printing staff members, Dave Brancazio, Jim Serdy, and Mike Rynerson.

I also thank my fellow students on the 3D Printing team for contributing both research efforts and camaraderie throughout our progress.
# Table of Contents

1. INTRODUCTION

1.1 Three Dimensional Printing
    1.1.1 Process Description
    1.1.2 Machine Architecture

1.2 Process Planning

1.3 Objective

2. RELATED TECHNOLOGIES

2.1 Solid Freeform Fabrication (SFF)

2.2 Computer Numerically-Controlled (NC) Machining

2.3 Ink-Jet Printing

2.4 Computer-Aided Engineering (CAE)

2.5 Very Large Scale Integration (VLSI)

2.6 Expert Systems in Manufacturing

3. PROCESS CHARACTERIZATION

3.1 Basic Printed Features
    3.1.1 Primitives
    3.1.2 Lines

3.2 Principal Profiles
    3.2.1 Profiles on Faces Normal to the Vertical Axis
    3.2.2 Profiles on Faces Normal to the Slow Axis
    3.2.3 Profiles on Faces Normal to the Fast Axis

3.3 Inclined Profiles
    3.3.1 Layer Stair-Step
    3.3.2 Raster Aliasing

3.4 Principal Dimensions
    3.4.1 Dimensions Along the Vertical Axis
    3.4.2 Dimensions Along the Slow Axis
    3.4.3 Dimensions Along the Fast Axis
4. PROCESS MODELING

4.1 Geometric Modeling
   4.1.1 Primitive Diameter
   4.1.2 Line Diameter

4.2 Temporal Independence
   4.2.1 Line Stitching
   4.2.2 Saturation Tolerance

5. MACHINE CAPABILITY

5.1 Printhead Control
   5.1.1 Proportional Deflection
   5.1.2 Pattern Memory

5.2 Droplet Availability

5.3 Deflection Performance
   5.3.1 Droplet Imprints
   5.3.2 Deflection Accuracy
   5.3.3 High-Frequency Deflection

5.4 Leading Edge Effects
   5.4.1 Aerodynamic Drag
   5.4.2 Mutually Induced Charge
   5.4.3 Leading Edge Concealment

6. DATAFILE PROCESSING

6.1 Geometric Representations
   6.1.1 Surface Facets
   6.1.2 Polygonal Slices
   6.1.3 Raster Segments
   6.1.4 Printing Transitions

6.2 Data Interpretation
   6.2.1 Slicing
   6.2.2 Rastering
   6.2.3 Encoding
7. PARAMETER SELECTION

7.1 Fixed Parameters
   7.1.1 Flow Rate
   7.1.2 Droplet Frequency
   7.1.3 Powder Volume Fraction
   7.1.4 Jet Spacing

7.2 Control Parameters
   7.2.1 Layer Spacing
   7.2.2 Line Spacing
   7.2.3 Printhead Speed

7.3 Derived Quantities
   7.3.1 Droplet Spacing
   7.3.2 Saturation

8. SURFACE ENHANCEMENT TECHNIQUES

8.1 Layer Spacing
   8.1.1 Analytical Prediction of Roughness
   8.1.2 Experimental Results

8.2 Boundary Offset
   8.2.1 Offset Techniques
   8.2.2 Problematic Cases
   8.2.3 Experimental Quantification
   8.2.4 Non-Uniform Offset

8.3 Sub-Raster Feature Snapping

8.4 Incline Tracing
   8.4.1 Methodology
   8.4.2 Experimental Verification

8.5 Diffuse Line Termination

9. FUTURE WORK

9.1 Three-Dimensional Extensions
   9.1.1 Incline-Specific Surface Offset
   9.1.2 Sub-Layer Scanning
   9.1.3 Gap-Fill Layers

9.2 Alternative Printing Styles
   9.2.1 Finishing Passes
   9.2.2 Condensed Jet Spacing

9.3 Comprehensive Process Planning
10. CONCLUSIONS

10.1 Process Characterization  216
10.2 Process Modeling  216
10.3 Machine Capability  217
10.4 Datafile Processing  217
10.5 Parameter Selection  217
10.6 Surface Enhancement Techniques  218
APPENDIX A DERIVATIONS

Appendix A1 Footprint Radius of a Spherical Section Cut by a Plane

Appendix A2 Footprint Radius of a Cylindrical Section Cut by a Plane

Appendix A3 Roughness Profile from an Analytic Circular Model

Appendix A4 Uniform Spacing of Sub-Levels Between Print Lines

APPENDIX B SOFTWARE PROGRAMS

Appendix B1 raster

Appendix B2 encode

Appendix B3 slcOffset

Appendix B4 stlView

Appendix B5 slcView

Appendix B6 rstToTiff

Appendix B7 3dpView

Appendix B8 slcExpand

Appendix B9 rstExpand

Appendix B10 3dpExpand

Appendix B11 slcConstruct

Appendix B12 3dpConstruct

APPENDIX C FILE FORMAT SPECIFICATIONS

Appendix C1 SLC Polygonal Slice Format

Appendix C2 RST Raster Segment Format

Appendix C3 3DP Machine Data Format

APPENDIX D DEFLECTION CALIBRATION
List of Figures

Figure 1.1 Example of a Three-Dimensional CAD Model 20
Figure 1.2 Example Parts Fabricated by Three Dimensional Printing 20
Figure 1.3 Three Dimensional Printing Cycle 21
Figure 1.4 Basic Information Flow for 3DP Process Planning 24
Figure 3.1 Single-Droplet Primitive 38
Figure 3.2 Primitive Position with Respect to Powder Surface 39
Figure 3.3 Single Line by 3D Printing 40
Figure 3.4 Test Part Design for Examination of Individual Lines 40
Figure 3.5 Top View of a Printed Line 41
Figure 3.6 Side View of a Printed Line 42
Figure 3.7 Line Printed with 10-micron Droplet Spacing 42
Figure 3.8 Line Printed with 20-micron Droplet Spacing 43
Figure 3.9 Line Printed with 40-micron Droplet Spacing 43
Figure 3.10 Primitive Undulations Along a Printed Line [Arthur] 43
Figure 3.11 Diminishing Droplet Concentration at Ends of Lines 44
Figure 3.12 Number of Droplet Impacts Along a Printed Line 45
Figure 3.13 Taper Effect at the Trailing Edge of a Line 45
Figure 3.14 Faces Normal to Principal Axes 47
Figure 3.15 Principal Profiles on a Face Normal to the Vertical Axis 47
Figure 3.16 Profile on a Face Normal to the Vertical Axis and Along the Fast Axis 48
Figure 3.17 Profile on a Face Normal to the Vertical Axis and Along the Slow Axis 49
Figure 3.18 Profile on a Face Normal to the Slow Axis and Along the Vertical Axis 49
Figure 3.19 Profile on a Face Normal to the Slow Axis and Along the Fast Axis 50
Figure 3.20 Profile on a Face Normal to the Fast Axis and Along the Vertical Axis 51
Figure 3.21 Profile on a Face Normal to the Fast Axis and along the Slow Axis 51
Figure 3.22 Stair-Step Layer Profile 52
Figure 3.23 Aliased Raster Profile 53
Figure 3.24 Layer Spacing vs. Layer Thickness 54
Figure 3.25 Layer Positions and Physical Dimensions Along the Vertical Axis 55
Figure 3.26 Line Positions and Physical Dimensions Along the Slow Axis 56
Figure 3.27 Droplet Positions and Physical Dimensions Along the Fast Axis 57
Figure 4.1 Circular Model of Profile Roughness 60
Figure 4.2 Profile Roughness vs. Spacing Between Circles 61
Figure 4.3 Line Diameter vs. Binder Dose 68
Figure 4.4 Process Time Scales in 3D Printing 70
Figure 4.5 Lines with Uniform Spacing 71
Figure 4.6 Evidence of Line Pairing 72
Figure 4.7 Line Pairing Mechanism 72
Figure 4.8 Structural Failure Along Fault Planes Between Jets 73
Figure 4.9 Saturation Test Part Design 75
Figure 4.10 Saturation Test Results 77
Figure 5.1 Continuous-Jet Printhead Controlled by Electrostatic Deflection 81
Figure 5.2 Deflection Axis 82
Figure 5.3 Pattern Memory vs. Explicit Deflections 83
Figure 5.4 Pattern Memory Example 84
Figure 5.5 Droplet Availability Along Fast Axis
Figure 5.6 Droplet Positions vs. Encoder Resolution
Figure 5.7 Droplet Imprint on Emulsion-Coated Substrate
Figure 5.8 Contact Angle
Figure 5.9 Contact Angle Between Binder and Emulsion-Coated Substrate
Figure 5.10 Analytic Geometry of a Circular Arc Cut by a Plane
Figure 5.11 Line Imprint on Emulsion-Coated Substrate
Figure 5.12 Proportional Deflection Test Design
Figure 5.13 Image of Deflection Imprint
Figure 5.14 Reduced Image of Deflection Imprint
Figure 5.15 Deflection Performance Results
Figure 5.16 Binder Mass vs. Droplet Spacing
Figure 5.17 Leading Edge of a Droplet Train
Figure 5.18 Aerodynamic Drag Effect
Figure 5.19 Droplet Charging Mechanism
Figure 5.20 Mutually Induced Charge Effect
Figure 5.21 Leading Edge Concealment
Figure 6.1 Triangular Facet Representation of a Three-Dimensional Object
Figure 6.2 Vertex-to-Vertex Rule for STL Facets
Figure 6.3 Approximation Error Between Planar Facets and a Circular Arc
Figure 6.4 Directional Sense of Polygons
Figure 6.5 Non-Minimal Representation of Polygons
Figure 6.6 Polygon Self-Intersection
Figure 6.7 Polygon Self-Intersection and Ill-Defined Sense
Figure 6.8 Mutually Intersecting Polygons
Figure 6.9 Raster Data
Figure 6.10 Raster Boundary Segments
Figure 6.11 Run-Length Compression of Raster Data
Figure 6.12 Printhead Transitions Example (Two-Jet Printhead)
Figure 6.13 Slicing and Rastering
Figure 6.14 Relative Positions of Triangle Vertices and a Horizontal Plane
Figure 6.15 Relative Positions of Edge Vertices and a Horizontal Line
Figure 6.16 State Changes of Raster Intersections
Figure 6.17 Scan Line Intersection through a Vertex
Figure 6.18 Degrees of Overlap
Figure 6.19 Adjacent Raster Segments
Figure 6.20 Improper Arrangement of Polygons
Figure 6.21 Initial Position of Multi-Jet Printhead Along Slow Axis
Figure 6.22 Progression Sequence for Multi-Jet Printing (Two-Jet Example)
Figure 7.1 Selection of Line Spacing with Respect to Jet Spacing
Figure 8.1 Rectilinear Model of Profile Roughness
Figure 8.2 Analytical Representation for Rectilinear Model of Profile Roughness
Figure 8.3 Solving Rectilinear Model of Profile Roughness by Area Under a Curve
Figure 8.4 Analytic Prediction of Roughness vs. Layer Spacing
Figure 8.5 Test Design for Inclined Profiles Using Rotated Cubes
Figure 8.6 Profile Data for a 10-Degree Incline at 170-Micron Layer Spacing
Figure 8.7 Profile Data for a 10-Degree Incline at 100-Micron Layer Spacing
Figure 8.8 Profile Data for a 5-Degree Incline at 170-Micron Layer Spacing
Figure 8.9 Profile Data for a 5-Degree Incline at 100-Micron Layer Spacing
Figure 8.10 Profilometer Results for Shallow Angles to Horizontal
List of Tables

Table 3.1 Experimental Conditions for Examination of Individual Lines 41
Table 4.1 Experimental Conditions for Primitive Diameter 64
Table 4.2 Quantities for Calculating Primitive Diameter Based on Mass 64
Table 4.3 Quantities for Calculating Line Diameter Based on Binder Dose 67
Table 4.4 Experimental Conditions for Line Diameter 68
Table 4.5 Experimental Conditions for Saturation Tolerance 75
Table 4.6 Saturation Test Results 76
Table 5.1 Experimental Conditions for Single-Droplet Imprints 88
Table 5.2 Quantities for Calculating Footprint Radius of a Droplet Imprint 91
Table 5.3 Experimental Conditions for Deflection Segments 92
Table 5.4 Quantities for Calculating Footprint Radius of a Line Imprint 93
Table 5.5 Deflection Accuracy, Measured via Imprint Analysis 96
Table 5.6 Experimental Conditions for High-Frequency Deflection 98
Table 5.7 Measured Mass of Lines under High-Frequency Deflection 99
Table 6.1 Relative Positions of a Triangle Vertices and a Horizontal Plane 122
Table 6.2 Relative Positions of Edge Vertices and a Horizontal Line 125
Table 8.1 Profilometer Results for 10° Angle to Horizontal 154
Table 8.2 Profilometer Results for 5° Angle to Horizontal 155
Table 8.3 Experimental Conditions for Boundary Offset 165
Table 8.4 Boundary Offset Measurements Along Fast Axis 166
Table 8.5 Boundary Offset Measurements Along Slow Axis 166
Table 8.6 Examples of Improved Accuracy by Sub-Raster Feature Snapping 173
Table 8.7 Surface Finish Improvement by Incline Tracing 189
Table 8.8 Experimental Conditions for Diffuse Line Termination 195
< blank page >
1. Introduction

1.1 Three Dimensional Printing

The emergence of solid freeform fabrication (also known as rapid prototyping) technologies [Jacobs] brings the potential for revolutionary advances in the speed and flexibility of product development. Solid freeform fabrication represents a new generation of computer-controlled fabrication methods that creates three-dimensional objects directly from a software model. These technologies are often described as "desktop manufacturing", in analogy to desktop publishing in 2-D. A fundamental approach to freeform fabrication is to build three-dimensional structures layer-by-layer. Each layer is a thin cross-section of the final geometry, and the layers are laminated in a vertical stacking order until the part is complete. Three Dimensional Printing is a freeform fabrication technique that forms the cross-sections by using a binder printhead to adhere particles in a powder bed.

Figure 1.1 is a rendering of a computer-aided design (CAD) model in the shape of an eight-cylinder engine block, and Figure 1.2 is a photograph of scale miniature parts fabricated from the model by 3D Printing. The printed parts show the ability to handle complex geometric features including overhangs, undercuts, curved slopes, and holes at arbitrary angles. The parts were fabricated from beginning to end with no part-specific tooling, fixtures, or support structures.
Figure 1.1 Example of a Three-Dimensional CAD Model
(Courtesy of Michael Rynerson, MIT)

Figure 1.2 Example Parts Fabricated by Three Dimensional Printing
1.1.1 Process Description

Three Dimensional Printing builds parts in layers, and forms the appropriate cross-sections by using a binder printhead to selectively adhere powder particles in each layer. Figure 1.3 illustrates the operating sequence and the basic build axes. Each cycle begins with fine powder spread into a thin layer, typically 100 to 200 microns thick. A slicing algorithm draws detailed information for each layer from a computer-aided design (CAD) model. Then a raster-scanning printhead applies a binder material to join particles where the object is to be formed. The supporting piston lowers the powder bed so that the next layer of powder can be distributed. This building cycle repeats until the part is complete. Removal of the unbound powder (after heat treatment for some materials) reveals the finished product.

![Three Dimensional Printing Cycle Diagram](image)

**Figure 1.3 Three Dimensional Printing Cycle**

The layer-by-layer approach gives 3D Printing the ability to fabricate parts with complex geometric features, including overhangs, undercuts, and blind passageways. The process is extremely flexible because it may utilize any material that can be produced in powder form. This
includes ceramics, metals, carbides, plastics, and waxes. Some engineering applications that would benefit dramatically from the capabilities of 3D Printing include the following:

- Metal tooling for plastic injection molding
- Ceramic shells for metal casting
- Structural ceramics for electronics packaging
- Bio-materials for implant devices

Three Dimensional Printing has already demonstrated added value in these and other applications, but development efforts must continually strive to improve surface finish and dimensional accuracy in order to offer highest benefit to engineering industries.

1.1.2 Machine Architecture

The standard 3D Printing machine has distinct build axes, as labeled above in Figure 1.3. Fabrication proceeds in a layer-by-layer fashion, as an elevator platform increments along the vertical axis (z-coordinate). The process is then simplified to a collection of two-dimensional cross-sections. The task of printing the cross-sections has been developed primarily using a raster architecture. In a raster approach, each layer is divided into a grid of scan lines, and each scan is traversed by a printhead pass over the powder bed. A slow axis (y-coordinate) advances the printhead step by step to different positions for printhead passes. The printhead passes run rapidly back and forth along the fast axis (x-coordinate). A multi-jet printhead is able to cover several of these scans in a single pass, thus greatly reducing the build time.

A raster machine architecture favors high production rate, large build volumes, and low cost. However, the raster design presents key challenges with respect to surface finish and dimensional
control. Raster-scanning and continuous-jet technology (Chapter 5) introduce constraints with respect to resolution, addressability, and droplet availability.

1.2 Process Planning

Process planning is the task of preparing a set of instructions that describe how to fabricate a part or build an assembly which will satisfy engineering design specifications. It is the link between design and manufacturing. In a broad sense, the instructions may include the specification of operations sequence, machines, tools, materials, tolerances, notes, process parameters, material treatment, fixtures, time standards, setup details, inspection criteria, gauges, and graphical representations of the part in various stages of completion [Bedworth]. Process planning is a very fundamental and extensive activity in manufacturing.

Figure 1.4 is a simple information flow diagram that shows the role of process planning for 3D Printing. Process planning takes the design specifications in the CAD model, incorporates process knowledge, and delivers detailed instructions to the 3D Printing machine. The design specifications also include criteria, priorities, and preferences. The knowledge base is a collection of facts about process capabilities, resolution, and constraints.
In principle, 3D Printing may exercise control over each and every binder droplet during the fabrication of a part. The number of droplets in a rectangular volume is equal to the number of layers, times the number of rows per layer, times the number of droplets per row. A filled cubic volume with 10 cm per side would contain over 1.7 billion droplets under typical parameters (170 µm layer spacing, 170 µm row spacing, 20 µm droplet spacing). Even the simple case of using toggle instructions for the start and stop position of each line segment results in nearly 700,000 instructions. Special smoothening techniques such as proportionally-deflected edge tracing (Chapter 8) can easily increase the number of instructions by an order of magnitude. The computational intensity of 3D Printing makes it essential to decide which droplets require precise control and which can be treated in a bulk sense, and the process planning algorithms must make this fundamental distinction.

Computer-aided process planning (CAPP) is the automated implementation of process planning, and it is motivated by several factors. The most obvious concern is the computational intensity described above. Droplet placement instructions must be automated because the number of discrete decisions is completely intractable otherwise. There are also several other motivating factors for automated process planning, as listed below.
• Minimize set-up time
• Document decision-making steps
• Provide consistent, repeatable performance
• Reduce requirements for operator training
• Capture knowledge from experiments and case examples

Three Dimensional Printing is unique from all other manufacturing technologies with respect to process planning. A key difference from machining, for example, is the absence of a need to recognize specific features, such as holes or slots. This flexibility bypasses traditional planning steps such as fixture design and tool selection. Process planning for 3D Printing requires very little consideration of the particular features on a part. This independence is vitally important because 3D Printing targets applications which often require very complex geometric features, such as conformal cooling passages around a heavily-contoured tooling cavity.

1.3 Objective

Three Dimensional Printing is a process that selectively binds powder on a layer-by-layer basis. The binder is deposited as discrete droplets, which adheres powder particles together. The machine has a multi-jet, raster-scanning architecture, and the control mechanism is electrostatic deflection of a continuous-jet printhead.

The objective of this research is to develop process planning algorithms that improve surface quality, consistent with process behavior and machine capability. Surface quality is evaluated primarily in terms of surface finish and dimensional accuracy. Development efforts focus on
enhancements to information content, as a CAD model is progressively reduced to discrete machine instructions.

There are several other criteria for the 3D Printing process, and efforts to improve surface quality should not violate those requirements. Maintaining high production rate and insuring internal structural integrity are two particular challenges.
2. Related Technologies

2.1 Solid Freeform Fabrication (SFF)

Solid freeform fabrication refers to a class of manufacturing technologies that are capable of producing complex geometric parts with little or no part-specific tooling. These technologies are also often known as rapid prototyping processes. 3D Printing represents one particular approach, but there are also several different approaches to freeform fabrication, a few of which are discussed below.

Stereolithography is a rapid prototyping technology that build parts by using a laser to locally solidify liquid polymer [Jacobs]. It is the most well-known commercial implementation of freeform fabrication. Selective Laser Sintering uses powder material, as does 3D Printing. However, this process uses a high-power laser to melt or otherwise fuse particles together [Marcus]. Laminated Object Manufacturing is a paper-based approach which determines layer cross-sections by cutting profiles on each sheet, and Fused Deposition Modeling extrudes thin wax filaments from a nozzle and winds the wax upon itself in space [Wohlers].

Three Dimensional Printing faces common issues with many of these technologies. For example, nearly all rapid prototyping techniques process information on a layer-by-layer basis, and the integrity of the slice information is crucial to successful fabrication. Some similarities from a process planning viewpoint are listed below. Improvements to the integrity of model data and the quality of the finished product are continually sought in both industry and academia [Crawford], [Frank], [Li], [Suh].
• Original CAD model begins as 3-D surface facets.
• Data is processed on a layer-by-layer basis.
• Macro-finish and micro-texture are sensitive to build orientation.

However, there are also several key differences, that represent process planning challenges for 3D Printing in particular.

• Control of droplets is highly discrete.
• Multiple jets must be effectively coordinated.
• High-speed raster demands that each point be scanned only once.
• It is possible to locally deposit different materials.

2.2 Computer Numerically-Controlled (NC) Machining

Computer numerically-controlled (CNC) machining removes material with the use of cutting tools, whose paths are directed by computer-generated instructions [Kalpakjian]. Of all manufacturing processes, CNC machining has received the greatest attention with respect to computer-aided process planning. The development of CNC machining and computer-aided manufacturing (CAM) in general has pursued a number of branches. One basic effort is the capture of process planning knowledge that had previously been recorded in machining tables or handbooks (e.g. preferred cutting speed for a given feed rate through aluminum). Another branch investigates computational geometry representations and algorithms for describing accurate tool paths. Feature recognition is yet another category which has received much attention as the most challenging of the above goals.
A large fraction of commercial CAPP systems utilize group technology concepts for feature recognition. Group technology is an engineering and manufacturing principle that groups parts together based on their similarities, in order to achieve economies of scale in a small-scale environment normally associated with large-scale production [Bedworth]. One reason why feature recognition is such an important issue for machining is the fact that geometric features almost always dictate the tool to be used. For example, a large planar area might best be leveled by a broad-sweeping tool called fly-cutter, whereas a simple hole is best created by a drill bit of the correct diameter. This lies in stark contrast to 3D Printing, in which the same basic element--a droplet of binder in powder--is used regardless of part geometry. It is the placement of these droplets that requires careful planning.

The research work behind this thesis focuses primarily on the part of process planning that is analogous to tool-path planning in CNC machining. In 3D Printing, a more appropriate name for this specific task would be drop placement planning. Other aspects of process planning, such as the selection of operating parameters, are also addressed.

Process planning for 3D Printing is in many ways easier than for CNC machining. For example, there are no major interference constraints between the "tool" and any point in the workpiece. However, planning for 3D Printing is more challenging in other ways because a single "tool" is expected to achieve high surface quality for all possible geometric conditions, and is also responsible for internal mechanical properties. The following lists summarize some key differences between tool-path planning and drop placement planning.
<table>
<thead>
<tr>
<th>CNC - Tool Path Planning</th>
<th>3DP - Drop Placement Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtractive Process</td>
<td>Additive Process</td>
</tr>
<tr>
<td>Interference Constraints</td>
<td>Point-Wise Access in Layers</td>
</tr>
<tr>
<td>Workpiece Shape/Orientation</td>
<td>Powder Base Material</td>
</tr>
<tr>
<td>Feature-Based</td>
<td>Geometry-Based</td>
</tr>
<tr>
<td>Tool Selection</td>
<td>Universal &quot;Tool&quot;</td>
</tr>
<tr>
<td>Exterior Definition</td>
<td>Exterior &amp; Interior Definition</td>
</tr>
</tbody>
</table>

Despite key differences, much can be learned and adapted from geometric investigations in tool-path planning. In particular, numerous discontinuity conditions for tool offset compensation (for cutter diameter) must be considered carefully both in CNC path planning [Held], [Pham] and in 3D Printing. Section 8.2 pursues more detailed discussion of the boundary offset issue.

### 2.3 Ink-Jet Printing

Three Dimensional Printing applies binder onto the powder bed using some of the same basic technology that is used for printing ink onto paper. Specifically, 3D Printing employs continuous-jet technology, which offers high rate (typically 40,000 to 80,000 droplets per second) and ability to handle a wide variety of fluid properties [Heinzl, Johnson].

3D Printing learns from advances in the design and control of ink-jet printheads in the two-dimensional realm, and extends development for printing binder onto powder. Examples of fine-tuning include compensation for inadvertent electrostatic charge and adjustment for aerodynamic drag [Fillmore]. A fundamental challenge for 3D Printing is the ability to handle fluids that have high solids loading and relatively low stability (e.g. flocculation and gelling may occur within the binder).
High-level graphics languages such as HP-GL and page-description languages such as PostScript have improved the ability to capture design intent before it is transferred onto paper [Brown]. Low-level enhancements such as smoothening character fonts for raster machines have also been addressed for 2-D printing [Hersch]. Basic principles of raster sampling for 2-D printing are directly relevant to 3D Printing as well [Duff], [Pratt].

2.4 Computer-Aided Engineering (CAE)

Processing a CAD model into machine instructions naturally requires a great deal of computational geometry. Applied computational geometry is used extensively in the fields of computer-aided engineering (CAE) and computer graphics. Computer-aided engineering uses digital computers to perform activities such as conceptual design, structural/thermal analysis, and design-for-manufacturing. The representations and algorithms developed in CAE offer a wealth of knowledge for data processing in 3D Printing.

Efficient handling of geometric data is critical in 3D Printing because of requirements related to both data storage and processing speed. A complex surface model for rapid prototyping may contain hundreds of thousands of surface patches (see Chapter 6). Upon slicing into layers, these surface patches translate into millions of polygon edges. The number of significant computations increases to billions when the polygons are interpreted as printhead instructions. This explosion of information clearly underlies the need for efficiency. Computations of this magnitude are not uncommon in computer-aided engineering and computer graphics. So the principles learned in these fields greatly benefit the development of 3D Printing software.
Most of the techniques developed in this thesis treat each layer separately, and each layer is represented by a pool of polygons. General issues for polygon manipulation have been addressed in CAE for both representations [Burton] and algorithms [Barton]. Basic approaches to converting polygon data into raster data are also well-developed [Duff]. More specialized techniques, such as the accurate offsetting of 2-D boundaries have also been studied as interesting problems in computational geometry [Held, Maekawa].

2.5 Very Large Scale Integration (VLSI)

*Very large scale integration* (VLSI) is the technology that enables the placement of a vast number of integrated circuits (typically over 100,000) on a single semiconductor chip. Such a challenging task has benefited greatly from a well-formed separation between design efforts and the actual processing steps during fabrication. A designer is equipped with precise knowledge about process capability in the form of *design rules*. Design rules are often presented in the form of minimum allowable values for widths, separations, extensions, and overlaps of geometric objects [Mead].

Three Dimensional Printing shares much in common with VLSI technology, and thus may learn from some of the paradigms developed in that field. A most powerful attribute of the 3D Printing process is that it is driven strictly by geometric data on a point-by-point scale, independent of the net shape of complete parts. This opens the possibility of a clean separation between fabrication steps at the manufacturing level and design efforts at the CAD level. Conformity to these rules during design stages leads to features that are known to be within the capabilities of the process. Chapter 7 discusses some important considerations among process planning parameters.
An example of a design rule in VLSI fabrication is that the depth/width aspect ratio of a gap may not exceed a particular limit if a subsequent material deposition must completely fill the gap. An example of a basic length unit in 3D Printing is the diameter of printed lines, which must be considered during process planning. Process planning must also address and facilitate quantities such as the overlap between lines for adequate stitching and structural integrity. Overlap is controlled by line spacing, droplet spacing, and diffuse printing patterns (Chapter 5).

2.6 Expert Systems in Manufacturing

The application of artificial intelligence principles is one of the recent trends in computer-aided process planning. An expert system is a tool which has the capability to understand problem-specific knowledge and use domain knowledge intelligently to suggest alternate paths of action [Kumara]. Expert systems are implemented as computer programs which combine the knowledge of an expert (or experts) in a particular domain, and attempt to function at such a level of expertise in problem solving.

In an expert system, a declarative knowledge base (facts) is separated from a procedural knowledge base (what to do), and an external inference mechanism activates these procedural rules. Such a separation has distinct advantages because changes in either the declarative knowledge base or the procedural knowledge base does not affect the other.

Below is a brief survey of expert systems in manufacturing applications (from [Kumara], except where noted). The survey provides examples of how expert systems are applied in other manufacturing processes. A declarative knowledge base for 3D Printing has not yet been
formally established. However, the principles of expert systems still serve as a model for the
development of the process planning system.

GARI is a knowledge-based system for process planning of metal cutting. Knowledge is
represented by production rules [Descotte]. The advice is provided with a weighting system to
resolve conflicting recommendations. The knowledge base includes a description of features,
dimensional and geometrical relations. Manufacturing rules are written in IF-THEN format, and
are parameterized by simple variables.

SIPP is a knowledge-based system for generative process planning of machined parts [Nau]. It
uses a frame-based knowledge representation system. The knowledge base consists of
machinable surfaces and capabilities of various machining operations and a control structure
which manipulates the knowledge base in order to construct process plans. The SIPP system uses
a best-first strategy based on Branch and Bound, and produces least cost process plans based on
user-specified criteria.

TOM is a rule-based system that focuses on producing detailed machining plans for a limited
type of feature, typically holes [Matsushima]. TOM uses a backward chaining mechanism to
generate a machine sequence. The conflict among rules is resolved using heuristic methods, such
as taking the rule most often used. It also has an explanation module which shows the line of
reasoning.

CUTTECH was developed for selection of cutting tools, pass sizes, speeds, and feeds, that
require machining expertise. The knowledge base comprises machining rules and machining and
tooling data. The system is not an expert system per se, but incorporates techniques of artificial
intelligence. Rules are stored in decision tables and algorithms, and are applied in descending order of importance.
3. Process Characterization

Three Dimensional Printing is an additive process that builds objects by selectively placing binder onto a powder bed. This chapter begins by introducing the basic printed features of this additive process. The most fundamental building element is a single droplet of binder in powder. The binder adheres powder particles into a cluster, and the resultant feature is called a primitive.

A train of consecutive droplets that fall during a single pass of the printhead forms a printed line. All of the printed area at a fixed vertical axis position is called a layer.

The sections below present some preliminary discussion of issues related to the size and shape of primitives and lines (more quantitative investigation is pursued in the next chapter). Reference positions for the placement of printed features are also discussed. Factors that may affect surface finish and dimensional accuracy are described in relation to the principal build axes. The principal axes are treated separately because they often involve different governing mechanisms for surface quality.
3.1 Basic Printed Features

3.1.1 Primitives

Figure 3.1 shows a primitive formed by a single droplet of colloidal silica binder and aluminum oxide powder. The droplet volume was approximately 250 pL, 30% silica by weight.

![Image of a single droplet primitive](image)

**Figure 3.1 Single-Droplet Primitive**

As shown in the figure, a primitive is roughly spherical (apart from the roughness among individual particles). The uniform shape agrees with physical expectation for fluid equilibrium (gravitational forces are negligible at this small scale). The spherical shape assumption is limited to dry powder only, however. A few applications of 3D Printing may deposit the powder in alternate ways, such as slip casting or spray deposition of powder in slurry-like form.

The position of a primitive is specified by layer position (z), slow axis position (y), and droplet release position along the fast axis (x). The primitives are printed *downward* into the powder.
bed. Therefore, it is the top-center positions of these spheres that can be specified in the data file. The following diagram illustrates the position of a primitive with respect to the powder surface. Complex impact phenomena which may distort the ideal layout in the figure have been investigated by [Fan].

![Diagram of primitive position with respect to powder surface]

Figure 3.2 Primitive Position with Respect to Powder Surface

The size of a primitive may depend on several geometric factors and material properties including droplet volume, packing density, particle size/shape, surface tension, and wetting conditions. Process planning has very little control over these factors (other than via material selection), so it is not within the scope of this thesis to study the material interactions in detail. The size of primitives for a given material system is typically characterized experimentally in advance (Chapter 4), and then used as a basic parameter for drop-placement algorithms.

3.1.2 Lines

A line is the next basic level of 3D Printing construction. A printed line is the feature formed by a consecutive series of binder droplets during a single pass of the printhead. The figure below show a scanning electron microscope (SEM) image of a printed line sample. The sample consists of aluminum oxide powder bound by colloidal silica binder. Droplets are spaced about 20 microns apart along the line.
Figure 3.3 Single Line by 3D Printing

The shape of lines were examined by printing test samples as shown in the following diagram. The lines are approximately 16 mm long and were printed simultaneously with mounting frames for structural support. The frames are printed in two orthogonal directions, so that the lines may be viewed from the top or from the side. Table 3.1 gives the experimental conditions.

Figure 3.4 Test Part Design for Examination of Individual Lines
Table 3.1 Experimental Conditions for Examination of Individual Lines

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Powder</td>
<td>Aluminum Oxide, 28 μm</td>
</tr>
<tr>
<td>Binder</td>
<td>Colloidal Silica 30%</td>
</tr>
<tr>
<td>Powder Solid Fraction</td>
<td>35% (Dry Packing)</td>
</tr>
<tr>
<td>Binder Flow Rate</td>
<td>0.75 ml/min</td>
</tr>
<tr>
<td>Droplet Frequency</td>
<td>43 kHz</td>
</tr>
<tr>
<td>Printhead Speed</td>
<td>0.43 m/s</td>
</tr>
<tr>
<td>Datafile</td>
<td>19960312.3dp</td>
</tr>
</tbody>
</table>

The following images show optical microscope images of lines from top and side views, respectively (droplet spacing was 10-microns). The lines appear to be cylindrical, despite some flattening of the up-facing profile, visible in Figure 3.6. As a line is printed, the up-facing surface is exposed to air (hence the leveling effect), while the rest of the line is embedded in powder. All parts of the line that are embedded in powder are subject to the randomness of particle arrangement and the flow of binder among the crevices.

![Figure 3.5 Top View of a Printed Line]
Figure 3.6 Side View of a Printed Line

The spacing of droplets may affect the shape of a line, as well as the diameter. Images showing the shapes of line are presented here, and line diameter is studied quantitatively in Chapter 4. If the droplet spacing along a line is very large, the undulations within individual primitives may become visible. Test samples were printed with different values of droplet spacing to check for any severe change in shape. Typical droplet spacing for the 3D Printing process is between 10 microns and 20 microns.

Figure 3.7 Line Printed with 10-micron Droplet Spacing
Lines with very wide droplet spacing could not be collected because of limited structural integrity. Lines printed by binder into dry powder disintegrate if the spacing is too wide. However, physical reasoning suggests that at some level of droplet spacing the individual primitives will become distinguishable. This has been verified using a more fixed material base (slip-cast alumina) with a droplet spacing of 120 microns, as shown below [Arthur]. Again, typical droplet spacing for the 3D Printing process is between 10 and 20 microns, an order of magnitude below the scale at which undulations are visible. Therefore the undulations are not a concern for normal printing conditions in dry powder.
A final important issue is the shape of lines is the detail at the ends. The droplets along a printed line typically have a high degree of overlap. Any given point along a line may be influenced by 10 or more droplet impacts. However, the number of droplet impacts diminishes at the ends of lines. The figure below gives an example showing the number of overlapping droplet impacts along a line.

![Diagram of droplet concentration](image.png)

**Figure 3.11 Diminishing Droplet Concentration at Ends of Lines**

The number of droplet impacts along a line is constant over most of the length; however, it diminishes linearly at the ends. The figure below shows the number of droplet impacts that may affect a primitive, assuming 200-micron primitive diameter and 20-micron droplet spacing. The number of droplet impacts at "steady state" is simply the primitive diameter divided by the droplet spacing. This phenomenon was also noted by [Curodeau].
The next image shows the tapering effect at the end of a line. This example shows a trailing edge (printing left to right). Leading edges are more complicated because of electrostatic and aerodynamic effects discussed in Chapter 5. The taper effect is typically not visible when printing with narrow line spacing. A solution for overcoming this problem with wider line spacing is investigated in Chapter 8.
3.2 Principal Profiles

There are several factors contributing to the roughness along any profile of a part fabricated by 3D Printing. Many of these factors affect all surfaces equally and are closely dependent on the material interaction between binder and powder. Examples include particle shape, particle size distribution, wetting behavior, and surface tension.

Surface finish can also be greatly improved by strategic droplet placement, in consideration of the machine axes and the printhead transitions. This is the approach taken in this thesis. The motion axes of the 3D Printing machine differ significantly, leading to roughness profiles that are dependent on orientation. It is convenient, therefore, to define a set of principal profiles for the study of surface finish. The principal profiles run along two orthogonal directions on each of the six principal faces of a rectangular box (making twelve in all). Each face is concisely referenced by the direction of its outward normal vector. The z-axis corresponds to the movement of the vertical platform of the machine, the y-axis is the slow axis for advancing scan positions, and the x-axis is the fast axis along which the printhead travels. The faces normal to the positive principal axes are visible in below.
Figure 3.14 Faces Normal to Principal Axes

Each of the faces normal to the principal axes has two principal profiles which are mutually orthogonal. The next diagram shows the two principal profiles on a face normal to the positive z-axis.

Figure 3.15 Principal Profiles on a Face Normal to the Vertical Axis

Some principal profiles share very similar characteristics, but more detailed investigation shows that the roughness along each principal profile may be governed by unique physical circumstances. Strategic algorithms for droplet placement may be able to smoothen many of these profiles. However, the strategies differ depending on which profile is being addressed.
3.2.1 Profiles on Faces Normal to the Vertical Axis

The roughness of a profile along the x-axis is affected by primitive geometry (size and shape) and droplet spacing along this axis. Process planning has limited control over droplet spacing, but no practical control over the primitive geometry. Figure 3.16 highlights the location of this principal profile on a printed part. The continuous-jet technology of 3D Printing insures that the droplet spacing is considerably less than other characteristic dimensions of the fabrication process. For example, typical droplet spacing is between 10 and 20 microns, while typical row spacing and layer spacing are each between 150 and 200 microns. Both physical argument and empirical observations agree that this profile should be the smoothest among all principal profiles.

![Diagram of Profile Orientation](image)

**Figure 3.16 Profile on a Face Normal to the Vertical Axis and Along the Fast Axis**

The roughness of a profile along the y-axis is affected by line geometry (size and shape) and row spacing along this axis. This profile is identified by Figure 3.17. Process planning has control over the row spacing and even the shape of the lines (via proportional deflection). The default condition, however, is for droplets to be arranged in a straight line within each row, in which case there is no control over the shape. Fill patterns (Chapter 5) may be used to diffuse the profiles along the slow axis.
A face normal to the positive z-axis (up-facing) may differ from a face normal to the negative z-axis (down-facing). The difference may attributed to two factors: condition of the powder bed and downward migration of binder.

3.2.2 Profiles on Faces Normal to the Slow Axis

The shape of a profile along the z-axis is highlighted below. This profile is affected by layer spacing and line shape. The layer spacing is a fundamental parameter selected during process planning. However, acceptable values are typically restricted to a narrow range because of requirements imposed by powder spreading and inter-layer stitching.
A profile along the x-axis on this face is expected to be very similar to a profile along the same axis but on a face normal to the z-axis. This profile is also governed by droplet spacing and primitive shape.

![Diagram of axis system](image)

Figure 3.19 Profile on a Face Normal to the Slow Axis and Along the Fast Axis

Printed parts have revealed that the surface finish on faces normal to the positive y-axis are slightly better than the finish on faces normal to the negative y-axis. The difference has been attributed to the way powder is spread across the bed. A spreading device pushes a powder pile from one side to the other, always in the direction of the negative y-axis. However, the present machine configuration does not allow process planning to alter this arrangement.

3.2.3 Profiles on Faces Normal to the Fast Axis

Faces normal to the x-axis are formed by the ends of printed lines, and therefore profiles along both the z-axis and the y-axis are affected by the shapes of the ends of lines. Profiles along the z-axis are also affected by layer spacing, while those along the y-axis are also affected by row spacing. As mentioned, process planning has control over both layer spacing and row spacing. Greater flexibility lies with row spacing because of more restrictive constraints on layer spacing. These profiles are highlighted below.
The shapes of the ends of a line segment differ significantly, depending on the direction in which the line was printed. The leading edge is subject to complex aerodynamic effects and electrostatic interactions [Curodeau]. These effects are considered machine-related issues and are discussed in Chapter 5. Differences between leading edges and trailing edges will clearly result in different surface profiles between faces normal to the positive x-axis and those normal to the negative x-axis, assuming uni-directional printing. In bi-directional printing the leading and trailing edges of lines are alternated and both of these faces appear the same. Process planning algorithms may alter the appearance of this profile using a technique called diffuse line termination (Chapter 8).
3.3 Inclined Profiles

The principal profiles cover basic orthogonal profiles that are found on engineering parts, but nearly all designs also include several inclined profiles. In fact, many surfaces have an infinite number of inclined tangent planes. Two particular categories of inclined profiles are particularly challenging with respect to surface finish in 3D Printing. The first category includes all of the profiles that cross from layer to layer, and exhibit a "stair-step" effect. The second type of profiles occurs within a single layer, and cross from line to line.

3.3.1 Layer Stair-Step

The following diagram illustrates the stair-step problem that is inherent to all layer-based manufacturing processes. This effect is common to all layer-based solid freeform fabrication processes, and is most noticeable on shallow inclines.

![Diagram of stair-step layer profile]

Figure 3.22 Stair-Step Layer Profile

Strategic orientation of the part with respect to the vertical axis helps to avoid shallow-angle slopes, but this approach is inadequate for complex shapes with a wide variety of contours. Reducing layer spacing is the most direct method for reducing the severity of stair-step roughness. Chapter 8 contains a quantitative analysis of the relationship between layer spacing...
and surface roughness. Layer spacing can not be set arbitrarily, however, for reasons discussed in Chapter 7.

### 3.3.2 Raster Aliasing

In each plane of a raster printing, piece-wise continuous slopes are replaced by discrete lines, resulting in jagged edges. The next figure illustrates this aliasing problem. This situation is analogous to aliasing in computer graphics and 2-D printing (Chapter 2). The jagged effect is most visible on shallow inclines.

![Figure 3.23 Aliased Raster Profile](image)

Proportional deflection (Chapter 5) overcomes the raster aliasing problem by adjusting droplet positions such that they accurately trace the intended contour. Edge tracing is one of the specific process planning techniques discussed in Chapter 8.

### 3.4 Principal Dimensions

Dimensional control over printed parts is highly dependent on orientation, so accuracy is examined separately for each principal direction. The scope of investigation is primarily
concerned with as-printed dimensional accuracy, ignoring distortion effects such as shrinkage and warping. More detailed study of shrinkage and other distortions in 3D Printing have been pursued by [Charnnarong] and [Yoo].

3.4.1 Dimensions Along the Vertical Axis

Dimensional accuracy along the vertical axis is determined by the layer spacing during slicing of the CAD model, layer thickness as formed in the powder bed, and stability of the powder layer positions. Of these factors, only the CAD slicing aspect is directly under the control of process planning software. A model for powder layer displacement is presented [Lee].

The layer spacing between two consecutive layers is specified in the data file, and can be chosen arbitrarily in the file processing software. The layer thickness, however, is determined by the interaction of powder and binder. In fact the thickness of any given layer must be greater than the spacing between its reference position (top surface) and the corresponding position of the previous layer. The overlap of a layer onto its predecessor is necessary for proper stitching between layers.

![Diagram](image)

**Figure 3.24 Layer Spacing vs. Layer Thickness**
Process planning specifies the position of the top surface of each layer. Figure 3.25 identifies layer positions in an example part, followed by an overlay of how the corresponding printed part is likely to appear. Predicting the physical dimension of a printed part requires tracking layer positions, with special note of any down-facing surfaces.

![Layer Positions and Physical Dimensions Along the Vertical Axis](image)

**Figure 3.25 Layer Positions and Physical Dimensions Along the Vertical Axis**

### 3.4.2 Dimensions Along the Slow Axis

Accuracy factors along the slow axis include line positions on each layer, the width of lines, and the performance of edge tracing by proportional deflection. Process planning has direct control over the line positions, and offsets may be applied to compensate for line width (Chapter 8). Edge tracing improves accuracy by allowing droplets to trace polygon boundaries exactly, without being constrained to the regularly-spaced scanning grid. Chapter 8 discusses edge tracing in more detail.

By default each printhead pass consists only of straight line segments, without the benefit of edge tracing. This result in a coarse printing mode, because the spacing between lines is relatively wide, about 200 microns. Reasons for this constraint on line spacing are discussed in Chapter 7. Special planning algorithms offer the greatest relative benefit for dimensions along the slow axis.
(compared to the vertical axis and fast axis) because the scanning resolution of proportional
deflection (~10 microns) is so much finer than that of binary (on/off) printing.

It is necessary to consider both line position and line width when predicting a dimension along
the slow axis. The figure below contrasts the center positions of several lines, and the physical
outcome. When special droplet deflections are used and lines are not placed exactly on an even
grid, these deflections must also be incorporated.

![Diagram of line positions and physical dimensions along the slow axis]

**Figure 3.26** Line Positions and Physical Dimensions Along the Slow Axis

### 3.4.3 Dimensions Along the Fast Axis

Process planning affects accuracy along the fast axis primarily by specifying droplet release
positions. Positions along the fast axis are recognized by a linear encoder with 10-micron
resolution. Therefore it is possible to have very high accuracy along this axis. However, droplet
availability is a limiting factor (Chapter 5).

Droplet spacing $\Delta x$ along the fast axis is a function of the printhead speed $v$ and the droplet
generation frequency $f$. Process planning sets droplet spacing indirectly by specifying printhead
speed. The droplet generation rate is tuned for reliable printhead operation and cannot be
selected arbitrarily.
\[ \Delta x = \frac{v}{f} \]

Equation 3.1

Maximum resolution is achieved when the printhead is traversed at a slow rate and the number of droplets is equal to or greater than the number of addressable encoder positions. However, it is not always practical to use a slower printhead speed for reasons because the higher binder concentration results in wider lines and more coarse resolution along the slow axis.

The length of a printed line segment along the fast axis should be equal to the difference in positions of the first and last droplets, plus the radii on each end, as shown below. In reality there are complex aerodynamic and electrostatic interactions that adversely affect accuracy by distorting the shape of the leading edges of lines. Process planning has bypassed this problem by using a technique called leading-edge concealment (Chapter 5).

![Figure 3.27](image)

Figure 3.27 Droplet Positions and Physical Dimensions Along the Fast Axis
4. Process Modeling

As stated in the opening chapter, the objective of this research is to develop information-driven printing techniques that improve surface finish and dimensional accuracy. The design of these techniques requires not only knowledge of machine capability (Chapter 5), but perhaps more fundamentally it requires a model for process behavior. A central responsibility for 3DP process planning is to specify droplet placement instructions. Fundamental considerations for primitive assembly include position, size, shape, and timing. The number of decisions related to positions alone could exceed one billion for the current build capacity of the 3D Printing machine. Therefore, the design of drop placement arrangements necessitates a working understanding of the geometry and temporal dependence of basic elements.

4.1 Geometric Modeling

This section presents the idealizations related to the size and shape of the component elements in 3D Printing. For example, primitives are assumed to be spherical in shape. The assumptions presented here and throughout this thesis are restricted to feature formation in dry powder only (a contrasting example would be cusp-shaped features formed in a slip-cast powder layer [Arthur]). However, the process planning approach is intended to be generally applicable to any identifiable primitive shape. Complex shapes make analytical predictions for profiles more difficult but not impossible.
Geometric idealizations make it possible to plan primitive and line arrangements according to specific objectives. For example, roughness on certain profiles can be predicted analytically as a combination of elements with circular cross-section. Key parameters are simply the diameter of each circle and the relative positions of their centers, as shown in Figure 4.1. This simple model could be applied not only to individual primitives, but lines as well (longitudinal axis normal to the plane of the page). The circular model is considered to be a worst-case scenario, because capillary smoothening and binder migration may reduce the roughness between individual elements. The center-to-center spacing between circles must be less than the diameter for a valid analytical solution.

![Figure 4.1 Circular Model of Profile Roughness](image)

Convenient geometric shapes make it possible to predict quantities such as roughness based on element spacing. This is useful because process planning dictates the positions of droplets and thus the spacing between printed elements. A common statistic for describing surface roughness along a profile is the arithmetic average of the deviations from the mean, abbreviated as $R_a$. Fine surface finish is associated with a low $R_a$ value. The arithmetic mean of the absolute values of the distances $y$ from the mean to the actual profile within a length $L$ is denoted $R_d$, and is determined by the equation below.
\[ R_a = \frac{1}{L} \int_0^L |y - \bar{y}| dx \]

**Equation 4.1**

In the discrete case, \( R_a \) is approximated by summations over \( n \) samples.

\[ R_a \approx \frac{1}{n} \sum_{i=1}^{n} |y_i - \bar{y}| \]

**Equation 4.2**

The chart below shows the relationship between profile roughness and the spacing between circles. The plot was generated by entering numerical values into the analytical expression for a circle.

![Chart showing Profile Roughness vs. Spacing Between Circles](image)

**Figure 4.2 Profile Roughness vs. Spacing Between Circles**

Figure 3.1 showed a particular example of a primitive with a diameter of about 150 microns, and Figure 3.3 showed a line with a diameter of almost 200 microns. The next two sub-sections take a more general and quantitative approach to assessing the diameters of primitives and lines.
Process planning algorithms would then use this information in selecting appropriate spacing of elements to achieve better surface profiles.

4.1.1 Primitive Diameter

Precisely quantifying the size of a primitive is difficult because of the small dimensional scale and the multitude of determining factors (Chapter 3). Optical measurements are useful, but are complicated by occlusion and by the difficulty of obtaining a significantly large number of samples. Therefore, diameter is evaluated by using a bulk measurement technique with a large number of samples. The procedure is as follows:

- Calculate the volume of a droplet based on flow rate and droplet frequency.
- Measure the mass of a primitive, as an average of a large number of samples.
- Calculate the mass of binder (after evaporation) in a primitive.
- Calculate powder mass from the difference between total mass and binder mass.
- Determine the envelope volume of a primitive based on powder bulk density.
- Calculate diameter from the volume of a sphere.

The volume of a single binder droplet $V_{\text{droplet}}$ is the flow rate $Q$ divided by the droplet generation frequency $f$, as expressed below.

$$V_{\text{droplet}} = \frac{Q}{f}$$

Equation 4.3
The mass of a single primitive $m$ is determined by measuring a large quantity of samples, and dividing the total mass by the number of primitives $n$. The large number of samples improves measurement resolution.

$$m = \frac{\sum_{i=1}^{n} m_i}{n}$$

Equation 4.4

The mass of a primitive is the sum of the binder mass and the powder mass. The binder mass $m_{\text{binder}}$ is the mass of solids remaining after evaporation of the transport fluid (water). The solids content of binder per unit volume is typically specified by the manufacturer, and can be verified by off-line experimentation (expressed in mass per unit volume). The mass of binder in a single primitive is simply the droplet volume $V_{\text{droplet}}$ times this solids content.

The mass of powder $m_{\text{powder}}$ in a primitive must be the total mass of the primitive minus the mass of binder. Given the mass of powder, the bulk density $\rho$ (mass per unit volume of the powder bed) is then used to calculate the envelope volume of a primitive. Powder bulk density is used because the powder dominates the space occupied by a primitive. Bulk density is typically measured using only dry powder in the bed, so this calculation assumes that the bulk density does not change significantly when binder is introduced.

$$V_{\text{primitive}} = \frac{m_{\text{powder}}}{\rho}$$

Equation 4.5

Having the envelope volume of a primitive, the diameter $D$ is calculated using the general analytic expression for the volume of a sphere.
\[ V = \frac{4}{3} \pi R^3 = \frac{1}{6} \pi D^3 \]

Equation 4.6

The spherical shape assumption offers a convenient way to predict a characteristic dimension (diameter) for 3D Printing primitives. An experiment was conducted by printing and collecting 24,000 primitives. Table 4.1 shows the experimental conditions and Table 4.2 lists the calculation parameters that were entered into the equations above.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Spacing</td>
<td>170 μm</td>
</tr>
<tr>
<td>Line Spacing</td>
<td>240 μm</td>
</tr>
<tr>
<td>Binder Material</td>
<td>Colloidal Silica 30%</td>
</tr>
<tr>
<td>Powder Material</td>
<td>Aluminum Oxide, 28 μm platelet</td>
</tr>
<tr>
<td>Powder Solid Fraction</td>
<td>35% estimated</td>
</tr>
<tr>
<td>Binder Flow Rate</td>
<td>0.7 ml/min</td>
</tr>
<tr>
<td>Droplet Frequency</td>
<td>64 kHz</td>
</tr>
<tr>
<td>Printhead Speed</td>
<td>0.64 m/s</td>
</tr>
<tr>
<td>Data File</td>
<td>19950830.3dp</td>
</tr>
</tbody>
</table>

Table 4.1 Experimental Conditions for Primitive Diameter

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binder Flow Rate</td>
<td>1.17E-08</td>
<td>m^3/s</td>
</tr>
<tr>
<td>Droplet Frequency</td>
<td>64000</td>
<td>1/s</td>
</tr>
<tr>
<td>Droplet Volume</td>
<td>1.82E-13</td>
<td>m^3</td>
</tr>
<tr>
<td>Total Mass</td>
<td>4.57E-05</td>
<td>kg</td>
</tr>
<tr>
<td>Number of Primitives</td>
<td>24000</td>
<td></td>
</tr>
<tr>
<td>Total Mass of One Primitive</td>
<td>1.90E-09</td>
<td>kg</td>
</tr>
<tr>
<td>Binder Solids Content</td>
<td>363</td>
<td>kg/m^3</td>
</tr>
<tr>
<td>Binder Mass in One Primitive</td>
<td>6.62E-11</td>
<td>kg</td>
</tr>
<tr>
<td>Powder Mass in One Primitive</td>
<td>1.84E-09</td>
<td>kg</td>
</tr>
<tr>
<td>Powder Bulk Density</td>
<td>1429</td>
<td>kg/m^3</td>
</tr>
<tr>
<td>Volume of One Primitive</td>
<td>1.29E-12</td>
<td>kg</td>
</tr>
<tr>
<td>Primitive Diameter</td>
<td>1.35E-04</td>
<td>m^3</td>
</tr>
</tbody>
</table>

Table 4.2 Quantities for Calculating Primitive Diameter Based on Mass

64
The total mass measured in the experiment was 45.7 mg. A predicted diameter for 28-micron alumina powder with colloidal silica binder under these conditions is 135 microns. Microscopic examination of actual primitives (as in Chapter 3) shows that this is a fair estimate.

The diameter for all primitives is considered to be constant for a given material system. The constant size is attributable to the fact that all droplets have very consistent volume, (Chapter 5). The actual diameters of individual primitives vary only by the locally random arrangements of powder particles.

4.1.2 Line Diameter

Factors that affect the size of individual primitives (e.g. droplet frequency, flow rate) do not vary greatly from run to run for a given material system. Therefore it was sufficient to quantify a single value primitive diameter (for a particular material system). In the case of lines, however, the spacing between binder droplets is also quite relevant in determining the binder content and consequently the line diameter. The line diameter may vary from run to run, even with a fixed material system. Line diameter may furthermore vary within a single run, depending on the local arrangement of droplets. Therefore, it is necessary to develop a model relating line diameter to binder concentration.

The diameter of a line would be expected to vary as the square root of the binder dose $A_{binder}$ in a line, based on the geometry of a circular cylinder. The binder dose is the volume of binder in a line $V_{binder}$ divided by the length of the line $L$. It is equivalent to the volume of a droplet divided by the droplet spacing.
\[ A_{\text{binder}} = \frac{V_{\text{binder}}}{L} \]

Equation 4.7

The volume of binder is related to the overall volume of a line by saturation \( S \), which is the fraction of available void space that is filled with liquid binder. The void space in the original powder bed is that space which is not occupied by powder. Equation 4.7 defines saturation \( S \) in terms of the binder volume \( V_{\text{binder}} \) and powder volume fraction \( \psi \) (the volume of powder divided by total volume) in a particular envelope volume \( V_{\text{envelope}} \). An experimentally determined value for line saturation under conditions similar to those explored commonly in this thesis (28-micron aluminum oxide powder packed to 36% with colloidal silica binder) is roughly 0.6 [Bredt]. Saturation in bulk regions have typical values between 0.6 and 0.9.

\[ S = \frac{V_{\text{binder}}}{(1 - \psi) \cdot V_{\text{envelope}}} \]

Equation 4.8

A circular cylinder is used to model the shape of a line. The general equation relating cylinder diameter \( D \) and length \( L \) to volume \( V \) is given below. Rounded ends are typically insignificant since the length of a line is usually much longer than its diameter.

\[ V = \pi R^2 \cdot L = \frac{\pi}{4} D^2 \cdot L \]

Equation 4.9

However, the volume of a line is difficult to quantify. Therefore the overall line diameter \( D \) is then expressed in terms of binder dose \( A_{\text{binder}} \) and saturation \( S \), by equating expressions for
volume \( V_{envelope} \) in Equation 4.7 and volume \( V \) in Equation 4.9 above. The saturation in a line is assumed to be constant for a given material system. The expression states that the line diameter varies as the square root of the binder dose. Process planning benefits from this model as a way to predict geometrical boundaries based on a controllable quantity. Binder dose is controlled by the proximity of droplets.

\[
D = \sqrt{\frac{4 \cdot A_{binder}}{\pi \cdot (1 - \psi) \cdot S}}
\]

Equation 4.10

The following table contains an applied example of this equation, showing the key values used in computation. Different values of droplet spacing affect the binder dose and hence thereby change the expected line diameter.

| Table 4.3 Quantities for Calculating Line Diameter Based on Binder Dose |
|-------------------|------------------|------------------|
| Quantity          | Value            | Units            |
| Binder Flow Rate  | 1.15E-08         | m³/s             |
| Droplet Frequency | 64400            | 1/s              |
| Droplet Volume    | 1.79E-13         | m³               |
| Droplet Spacing   | 1.00E-05         | m                |
| Binder Dose       | 1.79E-08         | m²               |
| Saturation        | 0.6              |                  |
| Powder Volume Fraction | 0.36         |                  |
| Line Diameter     | 0.000243         | m                |

This model is verified by comparison with experimental results from an independent methodology. An alternate way of estimating line diameter is to make weight measurements of a large number of lines, and to derive the diameter based on the calculated volume. This is the approach taken for estimating primitive diameter in the previous section. This weight method
offers a way to verify the binder-dose method above. The weight experiment was conducted with 28-micron aluminum oxide powder and 30 weight percent colloidal silica. Binder dose was controlled by varying the droplet spacing. Each sample at each level of binder dose contained 128 lines, 80 mm long. The following figure compares line diameters calculated from weight measurements against the diameter predicted by binder content, as a function of increasing binder dose. Experimental conditions are listed in the table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Spacing</td>
<td>170 µm</td>
</tr>
<tr>
<td>Line Spacing</td>
<td>320 µm</td>
</tr>
<tr>
<td>Binder Material</td>
<td>Colloidal Silica 30%</td>
</tr>
<tr>
<td>Powder Material</td>
<td>Aluminum Oxide, 28 µm platelet</td>
</tr>
<tr>
<td>Powder Solid Fraction</td>
<td>35% estimated</td>
</tr>
<tr>
<td>Binder Flow Rate</td>
<td>0.7 ml/min</td>
</tr>
<tr>
<td>Droplet Frequency</td>
<td>64 kHz</td>
</tr>
<tr>
<td>Printhead Speed</td>
<td>0.64 m/s</td>
</tr>
<tr>
<td>Data File</td>
<td>19950831.3dp</td>
</tr>
</tbody>
</table>

![Figure 4.3 Line Diameter vs. Binder Dose](image_url)
The results show that the diameters predicted from binder dose are slightly less than those determined from weight measurements. A key assumption for the calculation from weight measurements is that the powder solid fraction (packing density) remained the same after printing as it would be in a dry state. This assumption is imperfect because there is some local densification as the binder draws powder particles inward. Therefore, the measured mass values in the experiment were higher than appropriate for the calculation of line diameter. This offers an explanation why the data points calculated from weight are higher than those calculated from binder dose. In any case, the difference between the two methods (15 microns on average) is not much larger than the particle size (28 microns).

4.2 Temporal Independence

Thus far the geometry of printed features have been examined, and the key responsibility of process planning has been stated to be the placement of those features. However, the combinatorial behavior of those features over various time scales must also be addressed. Time dependence is an extremely complicated issue because it involves impact dynamics [Fan], capillary equilibrium [Charnnarong], and powder-binder chemistry [Bredt]. The combination of primitive features occurs in several different time stages, as illustrated in Figure 4.4. Two consecutive droplets from the printhead land onto the powder surface 15 to 20 microseconds apart. Two consecutive, adjacent lines from the same jet are printed about one second apart. In the multi-jet stitching arrangement, the last line printed by one jet "catches up to" the first line printed by the next jet in about 30 seconds (refer to Chapter 6 for multi-jet sequencing). Finally, two consecutive layers are completed about one or two minutes apart, depending on the size of the production run.
Various physical balances may occur at these time stages which span seven orders of magnitude. Furthermore, droplets are generated at the rate of tens of thousands per second. Temporal dependence would clearly make the design of printing algorithms quite intractable. Process planning, therefore, assumes the following key hypothesis:

*Printed surface geometry is independent of the time history of drop placement.*

To the extent that the hypothesis is an effective statement, process planning needs to consider only the geometry and positions of primitive elements. The remainder of this chapter examines limitations of this time-independence hypothesis.
4.2.1 Line Stitching

The temporal independence assumption implies that all lines should exhibit identical stitching behavior if the spacing between lines is constant. Figure 4.5 shows several lines printed in a single layer that have been positioned with uniform spacing.

![Uniform Spacing Diagram](image)

Figure 4.5 Lines with Uniform Spacing

However, in some cases the lines have been observed to exhibit a "pairing" phenomena, which has been noted in earlier 3D Printing research by [Lauder]. Figure 4.6 shows evidence of this effect.
Figure 4.6 Evidence of Line Pairing

Figure 4.7 illustrates the suspected mechanism behind line pairing. [Lauder] has conducted experiments to verify this suspicion. As a line of liquid binder is printed into powder, the combination of ballistic impact and subsequent capillary pull results in a trench. Depending on the size of the trench and the position of the next line of binder, a pairing effect may occur as the new line falls inside the trench. The third line has enough dry powder so it is not affected by the first two, but a fourth line would be drawn toward the third. The sequence continues resulting in the pairing of lines.

Figure 4.7 Line Pairing Mechanism
Line pairing refutes the hypothesis of time independence. However, this phenomena is not observed frequently. In fact, the lines in Figure 4.6 were printed with from the same data file as those in Figure 4.5, but the photograph was taken in a region that showed the pairing phenomena most dramatically. Most other regions appeared uniform, as in Figure 4.5. The reason for the difference is likely attributed to local variations in powder stability.

Temporal dependence for line stitching becomes a serious problem if the gaps are aligned such that they form fault planes. This threatens the structural integrity of printed parts. The figure below shows two tooling cavities. The tooling parts have internal cooling passages and are somewhat hollow. The part on the left collapsed along fault planes between jets of the multi-jet printhead.

![Fault Planes and Corrected with Fill Patterns](image)

**Figure 4.8 Structural Failure Along Fault Planes Between Jets**

It is suspected that poor stitching occurs because of time-dependent issues. The spacing between jets (4233 microns) is significantly wider than the spacing between lines (typically about 200
microns). Therefore, several lines must be printed before the last line of one jet "catches up" to the first line of the next jet. The delay (about 30 seconds) apparently allows capillary pull and drying effects to compromise line stitching.

Diffuse fill patterns (Chapter 5) have been implemented to successfully overcome the line pairing and jet stitching problems. When lines are printed with wide droplet distribution instead of straight lines, flexural strength has been markedly improved from 4.5 MPa to 25 MPa along the slow axis under some conditions [Bang].

4.2.2 Saturation Tolerance

As droplets are placed in strategic arrangements to improve surface quality, it may not be possible to maintain a constant saturation throughout a printed part. In fact most of the surface enhancement techniques discussed in Chapter 8 cannot be implemented without changing the local saturation at outer boundaries. A key question is the extent to which locally high saturation affects measured dimensions. This study extends beyond the geometric model for the diameter of individual lines (Chapter 4) because the printing history in neighboring regions may affect binder migration and final dimensions.

A set of test parts were designed with locally high levels of binder saturation on the surface. The higher saturation levels were achieved by using double passes of the printhead to print twice as much binder along selected lines on external faces. The figure below shows the basic design of the test parts, indicating the heavily saturated lines and the dimension along which they were measured. In the final case, the first line is re-printed with an additional printhead pass after all of the lines have been printed (approximately a two-minute delay).
Figure 4.9 Saturation Test Part Design

The following table lists the experimental conditions. There are 100 lines per layer and 60 (identical) layers. The designed spacing between the centers of the first and last lines is 18400 microns. Eight replicates were printed for each experimental condition. The powder material was stainless steel, bound by acrysol binder. This material system was used because it is more directly relevant to tooling applications than the ceramic system used in other experiments, and tooling applications are vitally concerned with dimensional accuracy. The saturation in bulk regions is calculated to be 93%. The double-print regions are super-saturated and may experience substantial dimensional change if the surplus binder wicks away into the dry powder.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Spacing</td>
<td>170 μm</td>
</tr>
<tr>
<td>Line Spacing</td>
<td>184 μm</td>
</tr>
<tr>
<td>Binder Material</td>
<td>Acrysol 25%</td>
</tr>
<tr>
<td>Powder Material</td>
<td>Stainless Steel 316L, 170-400 mesh</td>
</tr>
<tr>
<td>Powder Solid Fraction</td>
<td>60% estimated</td>
</tr>
<tr>
<td>Binder Flow Rate</td>
<td>1.0 ml/min</td>
</tr>
<tr>
<td>Droplet Frequency</td>
<td>44 kHz</td>
</tr>
<tr>
<td>Printhead Speed</td>
<td>1.5 m/s</td>
</tr>
<tr>
<td>Data File</td>
<td>19960222.3dp</td>
</tr>
</tbody>
</table>
The next table compares the measured dimension under normal circumstances against the dimensions for heavily saturated samples. Measurements were made using digital calipers with a resolution of 10 microns. The data reference is the center-to-center spacing between the first and last lines. The average dimension of the parts for the normal printing case was 244 microns larger than the reference dimension. This represents the width of the printed lines (Chapter 3).

<table>
<thead>
<tr>
<th>Condition</th>
<th>Reference (μm)</th>
<th>Width (μm)</th>
<th>Difference (μm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Printing</td>
<td>18400</td>
<td>18644</td>
<td>244</td>
</tr>
<tr>
<td>Double First Line</td>
<td>18400</td>
<td>18663</td>
<td>263</td>
</tr>
<tr>
<td>Double Last Line</td>
<td>18400</td>
<td>18685</td>
<td>285</td>
</tr>
<tr>
<td>Double First Line, Delayed</td>
<td>18400</td>
<td>18660</td>
<td>260</td>
</tr>
</tbody>
</table>

The figure below shows the results graphically. The pooled variance (weighted to compensate for number of replicates in each case) is 1064 microns$^2$ and the variance of the mean is 266 microns$^2$, corresponding to a standard deviation of 16.3 microns (at which the error bars are placed).
The first important observation is that the amount by which the dimension has changed is about 40 microns for the last line, and about 20 microns for the first line. This should be compared to expectations based on the relationship between binder dose and diameter for individual lines

\[ D = \sqrt{\frac{4 \cdot A_{\text{binder}}}{\pi \cdot (1 - \psi) \cdot S}} \]

from a formula involving the square root of 2. If either the first or last line of a test part receives double the binder dose and the line diameter is, the equation predicts that the line diameter will increase by a factor of the square root of 2. If all of the lines in a part are 244 microns wide, a line with twice the binder dose would be 345 microns wide.
The measured differences in dimension are not as high as would be predicted by the binder dose model. In fact, the dimension change is less than half of what was predicted for the case of doubly-printed first lines. Small dimensional change in response to locally high binder concentration on the exterior of a part means that the process has some tolerance for locally high levels of saturation. Consequently, process planning algorithms do not have to precisely control particular saturation levels at all points throughout a printed part. Even if all lines were considered to be 244 microns in the example above, the dimensional error would be small. The total dimensional change (40 microns in the worst case) is only a fraction of a line diameter, and causes only a small deviation with respect to the scale of the entire part (in this case 18400 microns). A 40-micron dimensional change on these test parts represents only a 0.2% error with respect to the full dimension along the slow axis.

The relevance of line sequencing deserves some discussion. If the process was completely independent of time history, all three test cases above would have the same deviation from the default case (Figure 4.10). Each of the three test cases contain one line that has a double concentration of binder, and differ only in line sequence. A possible explanation is as follows. In "double first line" case, excess binder causes the first line to widen uniformly because all surrounding material is dry powder. The "double last line" is printed under very different circumstances because there is printed material to one side of the last line, and this region is nearly fully saturated in this example (93%). Therefore, the binder bleeds outward and the final dimension is wider than the "double first line" case. If the bulk saturation had been lower, there may have been less of a difference between the two cases. When the first line is double-printed after a delay, the outward bleeding is less of a problem because the binder has time to dry.
This experiment shows that the time-independence assumption may leave discrepancies of approximately 20 microns. The discrepancy is non-zero, but still small compared to the overall dimension of typical parts.

5. Machine Capability

The techniques by which process planning pursues its objectives must be consistent with machine capability. This chapter discusses key issues related to the 3D Printing machine. Section 5.1 and Section 5.2 introduce basic topics related to functionality. Section 5.3 verifies controllability at a fine scale, and Section 5.4 examines some process disturbances that require compensation via a special print style.

5.1 Printhead Control

Three Dimensional Printing employs a continuous-jet printhead, controlled by electrostatic deflection. The expression "continuous-jet" explains that a stream runs continuously through a nozzle, and droplets are formed as the fluid jet breaks apart. The droplets that are not intended to impact the powder bed are re-directed into a catcher. Piezo excitation of the droplet generator assists in maintaining a stable droplet release frequency. Another basic type of printhead is a drop-on-demand printhead, in which droplets are released only when triggered by the control electronics. Continuous-jet technology typically offers much higher rate than drop-on-demand approaches, and has been found to be more robust to a wide variety of binder materials [Williams].
Several on-line control techniques have been developed to bring drop placement accuracy to within ±10 microns along the fast axis and ±3 microns along the slow axis [Sachs]. The high accuracy is achieved by controlling jet stability and the time-of-flight of droplets. However, the on-line control techniques are not discussed here in detail because they are implemented strictly at the machine level. Process planning focuses primarily on the basic architecture and operating constraints of the machine, while the on-line systems address drop-placement control. Planning algorithms attempt to arrange printed features in such a way that dimensional errors and surface roughness of parts are reduced to this fine level of controllability (about 10 microns), rather than the relatively coarse scale of printed features presented in Chapter 3 (over 100 microns).

5.1.1 Proportional Deflection

The droplets that exit the droplet generator are controlled by electrostatic deflection. A charging cell first induces precise charge levels onto droplets as they break away from the jet. As these droplets pass through a deflection cell, they are deflected in a direction transverse to the line of flight. The figure below shows a schematic of basic printhead operation. A more detailed description of the charging mechanism is given later in the chapter.
An absence of charge allows the droplets to land straight onto the powder bed. A high level of charge causes extreme deflection into a catcher. The figure also shows one of the most powerful features of 3D Printing technology: the ability to partially deflect droplets in flight. The magnitude of deflection is roughly proportional to the charge applied to each droplet, so this control feature is often called proportional deflection. The droplets are not constrained to the binary states "on" or "off". Rather, they may be placed at any position within a range of approximately 500 microns (depending on configuration details).

The printhead on the 3D Printing machine is oriented such that the deflection axis (left-right in Figure 5.1) is perpendicular the fast scanning axis of the machine (in and out of the page). In this arrangement, proportional deflection adds another degree of freedom with respect to droplet placement, and enables each jet to span a region wider than a single print line as it scans over the powder bed. The deflection axis is parallel to the slow axis, but its origin travels with the
printhead as shown below. Its resolution (~10 microns) is significantly finer than typical line widths (~200 microns).

![Diagram of deflection axes](image)

**Figure 5.2 Deflection Axis**

### 5.1.2 Pattern Memory

The design of the printhead and its control electronics make it possible to assign deflection instruction to each individual droplet. However, it is very impractical to specifically control every droplet because there may be a billion or more droplets that make up a complete part. Therefore, straight lines are printed simply with an "on" instruction at one particular position, and an "off" transition at a later position along the fast axis.

Oftentimes it becomes useful to print along the fast axis with lines that are not straight. One reason would be to improve stitching behavior. Diffusing a line may also be very valuable to control the saturation of binder within bulk regions of parts. Dedicated *pattern memory* has been designed into the control electronics of the machine to address these needs. Pattern memory refers to a repeating set of deflection sequences that are stored in computer hardware memory. The next figure shows simple example of a dashed line, contrasting explicit instructions and
pattern memory. A fixed sequence of transitions is defined, and the sequence is repeated as long as the pattern is active.

![Diagram showing pattern memory and explicit deflections](image)

**Figure 5.3 Pattern Memory vs. Explicit Deflections**

Pattern memory clearly offers a more efficient way to specify repetitive sequences. With explicit deflection instructions, the data file would have to specify "on at position 40, off at position 50, on at position 60, off at position 70, ...". However, pattern memory can substitute a concise instruction like "activate pattern #2 at position 40". Examples of other defined patterns may include sinusoidal waves and sequences from random distributions. The fill patterns used to enhance strength in many tooling parts were generated as a random sample from a triangular distribution [Bang].

A pattern sequence repeats as often as necessary until the pattern is deactivated. Also, the pattern has a fixed alignment with respect to the machine. The illustration below shows how a pattern would be implemented as a zig-zag sequence. The figure shows only the centerline of the droplet path.
The patterns that can be printed are limited by the number of steps that are stored in memory. The current implementation uses a sequence 256 discrete steps. The only other limiting factor (aside from deflection range and resolution) is droplet availability, which is the topic of the next section.

5.2 Droplet Availability

Process planning selects the positions of each of the droplets that are released from the printhead. The objective of this research is to arrange these droplets in a way that minimizes surface roughness and maximizes dimensional accuracy. Therefore, droplet availability at any given position and time is a fundamental concern for process planning.

Droplet availability refers to both the timing and positioning of droplets. There are three important facts to consider: (1) droplets are released at a pre-determined, constant rate, (2) there may be only one droplet at any position along the fast axis in a given printhead pass, and (3) addressing the droplets requires speed-matching with a position encoder.

Liquid binder exits the printing nozzle as a stream, and fluid stability requirements naturally break the stream into droplets. Tuned piezoelectric excitation helps to maintain stable break-off.
at a controlled distance from the jet orifice. The droplet generation rate \( f \) is the number of droplets that are formed per unit time. Typical values are between 45 and 80 kHz, depending on the particular droplet generator, but the value is constant throughout any given production run. [Bredt] presents a more detailed discussion of resonant printhead physics, but in the context of process planning it is most important simply to know that the droplet generation rate is an externally-imposed constant value. Droplet generation rate presents a fundamental constraint for process planning because it limits the number of droplets available within a particular time span, and sets the time increment between them.

The vertical spacing of droplets in flight is translated into horizontal spacing as they fall onto the powder bed beneath a moving printhead. The expression below relates droplet spacing \( \Delta x \) to droplet generation rate \( f \) and printhead speed \( v \).

\[
\Delta x = \frac{v}{f}
\]

Equation 5.1

The droplet spacing is constant for a given printhead pass because both the droplet generation rate and the printhead speed are fixed. The droplet spacing is always greater than zero because the printhead cannot be stopped over the powder bed. This limits the freedom to place droplets along the fast axis during a given printhead pass, as illustrated in the next diagram. The droplets may be deflected to any position along the deflection axis (parallel to the \( y \)-axis), but the spacing along the \( x \)-axis is fixed. The figure only shows droplet centers; in actuality there is much overlap among droplets because droplet diameter (about 70 microns) is significantly larger than droplet spacing (less than 20 microns).
Availability of droplets in precise placement schemes is only useful to the extent that they are addressable as well. Deflection instructions are pre-loaded into hardware memory associated with positions along the fast axis. The position of the printhead is registered with respect to the powder bed as it passes linear encoder ticks along the fast axis. A unique deflection instruction may be assigned at each encoder tick. The encoder ticks on the current 3D Printing machine are spaced 10 microns apart. Since the droplets fall at a fixed rate, the positions with respect to the encoder ticks can only be controlled by printhead speed. Depending on the relative rates of droplet generation and printhead travel, the droplets positions may or may not line up with the encoder ticks. The next figure shows three basic scenarios, based on varying printhead speed for a fixed droplet generation rate. The "match" speed is the one in which the droplet spacing (refer to Equation 5.1) is exactly the same as the encoder resolution. There is a surplus of droplets at slower speed and a deficiency of droplets at higher speed. As in Figure 5.5, only droplet centers are shown.
Deflection instructions may be issued at any encoder position. Ideally, there is a droplet available for each possible instruction. This is the case when the printhead is traversed at the "match" speed. If there is a surplus of droplets, some deflection levels may be over-represented (slower speed case, instruction C). If there is a deficiency, some instructions may be missed entirely (faster speed case, instruction E). Often the macroscopic shape of printed parts is insensitive to one or two instructions at this fine scale (10 microns). However, the matching between droplet availability and encoder positions must be considered when extremely precise droplet addressing is desired.

### 5.3 Deflection Performance

Process planning relies on the ability to precisely control the positions of droplets, according to information specified in software. This section investigates how well proportional deflection is actually controlled. Section 5.3.1 describes an experimental tool to track the placement of droplets. Section 5.3.2 then summarizes a test of deflection accuracy, repeatability, and linearity.
Finally, Section 5.3.3 examines deflection performance when instructions occur at high-frequency.

5.3.1 Droplet Imprints

Several complex interactions take place when binder droplets impact the powder bed and primitives are formed [Bredt], [Fan]. These interactions inhibit clear examination of droplet positions. Therefore, it has been most useful to capture the placement of individual droplets on flat substrates rather than the actual powder bed. Glass slides and plastic film used in photography are convenient substrates because the surfaces are coated with an emulsion to lock the droplet imprints upon impact. The image below shows the top view of an imprint collected on a glass substrate. The printing conditions are listed in the table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binder Material</td>
<td>Acrysol 25%</td>
</tr>
<tr>
<td>Substrate</td>
<td>IMTEC Poly-Edged Hi-Res Plate 1A</td>
</tr>
<tr>
<td>Binder Flow Rate</td>
<td>1.0 ml/min</td>
</tr>
<tr>
<td>Droplet Frequency</td>
<td>45 kHz</td>
</tr>
<tr>
<td>Printhead Speed</td>
<td>0.45 m/s</td>
</tr>
<tr>
<td>Data File</td>
<td>19960309.3dp</td>
</tr>
</tbody>
</table>
Figure 5.7 Droplet Imprint on Emulsion-Coated Substrate

Before using this imprint technique to study deflection control, however, it is first necessary to establish a quantitative understanding of the imprint dimensions. The footprint diameter formed by a droplet that is deposited onto a flat surface can be predicted by measuring the contact angle, $\beta$. The contact angle is a repeatable effect observed in a fluid between its interface with a solid and its free surface [White].

![Figure 5.8 Contact Angle](image)

The volume of any droplet released from the printhead is precisely known from the binder flow rate $Q$ and the droplet release frequency $f$, as expressed below. This original volume remains constant as the droplet comes to rest on a flat surface, assuming that absorption and thermodynamic effects (such as evaporation) are not relevant at the time of measurement.
Equilibrium causes the fluid on the flat base to take the shape of a spherical cap. The height of the droplet is small (~10 microns), so pressure differences from gravity are negligible. The photograph below shows a quantity of 25% acrysol binder on a glass substrate. The image verifies the assumption of spherical shape, and the fact that the interface is wetting ($\beta < 90^\circ$). The droplet captured in this image was deposited by a syringe and is actually several times larger than a printed droplet. However, wetting angle is not dependent on volume, so the angle is applicable to smaller droplets as well.

![Image of a droplet with contact angle](image)

**Figure 5.9 Contact Angle Between Binder and Emulsion-Coated Substrate**

The figure shows a contact angle of about 27 degrees. This value is determined by using a compass to follow the radius of curvature $R$ and a ruler to measure the footprint radius $a$, as drawn in the diagram below. The sine of the angle equals the ratio $a / R$.
Figure 5.10 Analytic Geometry of a Circular Arc Cut by a Plane

Manipulating the geometric relations in above diagram results in an expression for the footprint radius $a$ of a binder droplet in terms of droplet volume $V$ and the contact angle $\beta$. The equation below shows the final expression (derivation in Appendix).

\[
a_{\text{droplet}} = \left[ \frac{3V \sin^3 \beta}{\pi \left[ 3(1 - \cos \beta)^2 - (1 - \cos \beta)^3 \right]} \right]^{1/3}
\]

Equation 5.3

The following table presents a summary of key quantities used in calculating the footprint radius based on known quantities. The flowrate, droplet frequency, and printhead speed are the machine parameters during printing. The contact angle is estimated from Figure 5.9.

| Table 5.2 Quantities for Calculating Footprint Radius of a Droplet Imprint |
|-----------------------------|-------------|------------------|
| Quantity                   | Value       | Units            |
| Binder Flowrate            | 1.67E-08    | ml/min           |
| Droplet Frequency          | 45000       | Hz               |
| Contact Angle              | 0.471       | radians          |
| Droplet Volume              | 3.7E-13     | m$^3$            |
| Footprint Radius            | 9.88E-05    | m                |
| Footprint Diameter          | 198         | $\mu$m           |
The footprint diameter predicted in this example is 198 microns, which is slightly larger than an estimate of 170 microns from in Figure 5.7. A contact angle of 40 degrees would better match the 170-micron prediction. Discrepancies may have been caused by imprecise estimation of contact angle and measurement of flow rate.

The footprint radius of a printed line (several droplets in a straight line) can also be derived analytically (derivation in Appendix). The line is assumed to take the shape of a section of a circular cylinder, and the cross-section is identical to that shown in Figure 5.10. The volume $V$ of binder in a line is equal to the volume of a single droplet times the number of droplets in the line. The ends of the cylinder do not significantly affect the footprint diameter if the length of line $L$ is much greater than the width. The ratio $V / L$ is equal to the binder dose $A_{\text{binder}}$ mentioned in Chapter 4.

$$a_{\text{line}} = \left[ \frac{V / L}{(\beta - \sin \beta \cos \beta) / \sin^2 \beta} \right]^{1/2}$$

Equation 5.4

A line is a train of individual droplets. The figure below shows a line printed onto a flat substrate. Specific details for this sample are given in the table. The footprint diameter estimated from this image is about 300 microns. Calibration was done by taking an additional image of a silicon carbide fiber with 80-micron diameter at identical magnification.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binder Material</td>
<td>Acrysol 25%</td>
</tr>
</tbody>
</table>
The next calculation table lists the key quantities used in calculating the footprint diameter of a line imprint. The binder dose is calculated as the droplet volume divided by the droplet spacing, which is entirely equal to the total line volume \(V\) divided by the line length \(L\).

Table 5.4 Quantities for Calculating Footprint Radius of a Line Imprint

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binder Flowrate</td>
<td>1.33E-08</td>
<td>0.8 ml/min</td>
</tr>
<tr>
<td>Droplet Frequency</td>
<td>44000</td>
<td>Hz</td>
</tr>
<tr>
<td>Contact Angle</td>
<td>0.471</td>
<td>radians</td>
</tr>
<tr>
<td>Droplet Volume</td>
<td>3.03E-13</td>
<td>m³</td>
</tr>
<tr>
<td>Printhead Speed</td>
<td>1.5</td>
<td>m/s</td>
</tr>
<tr>
<td>Droplet Spacing</td>
<td>3.41E-05</td>
<td>34 μm</td>
</tr>
<tr>
<td>Binder Dose (V/L)</td>
<td>8.89E-09</td>
<td>m²</td>
</tr>
<tr>
<td>Footprint Radius</td>
<td>1.66E-04</td>
<td>166 μm</td>
</tr>
<tr>
<td>Footprint Diameter</td>
<td>331</td>
<td>μm</td>
</tr>
</tbody>
</table>
The predicted diameter of 331 microns supports the estimate of 300 microns measured from the image in the preceding photograph. These results offer consistency to the relationship between quantities of material deposited and actual physical dimensions. Having established this consistency, the next sections proceed by using this imprint technique to verify aspects of deflection control.

5.3.2 Deflection Accuracy

The imprint technique onto film substrates has been applied to study the accuracy, repeatability, and linearity of deflection. The figure below illustrates the test geometry, in which selected segments are deflected in graduated magnitudes away from the centerline. Each deflected segment is 2000 microns long. The deflection magnitude varies between 20 and 160 microns from the center line. Experimental conditions are as listed in Table 5.3. The printhead passes are made with a single jet, to avoid any variability from jet-to-jet.

\[ y \]
\[ x \]

\[ \begin{array}{cccccccc}
20 \, \mu m & 40 \, \mu m & 60 \, \mu m & 80 \, \mu m & 100 \, \mu m & 120 \, \mu m & 140 \, \mu m & 160 \, \mu m \\
\end{array} \]

**Figure 5.12 Proportional Deflection Test Design**

The following photograph shows a sample image of the printed lines, taken with a digital camera through an optical microscope. This particular selection shows a deflection that was designed to be 160 microns.
Quantitative examination is conducted by digital image analysis. A reduced image for such analysis is shown in below. Centerline data points are then extracted using a custom graphics program [Arthur]. The images have a resolution of 640 data points across the profile, with a resolution of 240 steps along the vertical direction.

The table below shows that deflection accuracy over the entire range lies within 2 microns. All units are expressed in microns. The average centerline of the deflected segment is computed using 100 data points in the center of each image. To guarantee leveling, 100 points were
collected from the left side and balanced against 100 points from the right. Deviation from a horizontal datum is corrected before averaging the deflected centerline.

<table>
<thead>
<tr>
<th>Design (µm)</th>
<th>Measurement (µm)</th>
<th>Difference (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>18.29</td>
<td>-1.71</td>
</tr>
<tr>
<td>40</td>
<td>41.67</td>
<td>1.67</td>
</tr>
<tr>
<td>80</td>
<td>81.49</td>
<td>1.49</td>
</tr>
<tr>
<td>160</td>
<td>160.28</td>
<td>0.28</td>
</tr>
</tbody>
</table>

The following plot summarizes the results of the deflection imprint experiment, showing graphical evidence of a high level of accuracy and linearity over a wide range.

![Figure 5.15 Deflection Performance Results](image)

96
Repeatability was examined by printing four identical samples with +160 microns. The standard deviation across these four measurements is 0.43 microns. Symmetry was studied by comparing positive and negative deflections of equal magnitude. The average of the four samples having a designed deflection of +160 microns was +161.3 microns, while the average of the four samples at -160 microns was -165.2 microns.

5.3.3 High-Frequency Deflection

The deflection accuracy experiment above shows that proportional deflection behaves correctly for line segments. The deflected line segments above contain approximately 100 droplets, but only transition changes in deflection level (e.g. from 0 to +160, back to 0). However, it has not yet been verified that the printhead properly handles a large number of deflection instructions that occur at high-frequency.

One possible need for high-frequency deflection is the use of a diffuse printing mode, in which only a fraction of the available droplets are actually printed (every other droplet, for example). The remaining fraction is deflected into the catcher using pattern memory (Section 5.1.2). Diffuse printing may be motivated by the need to maintain acceptable binder saturation. A number of factors lead to increased values of saturation (Chapter 7), and in some cases diffusing the droplet concentration in each line is the simplest way to keep saturation at a proper level.

An experiment has been conducted to verify the performance of high-frequency deflection, using diffuse lines. Binder droplets are collected onto plastic film rather than in the powder bed. The expected accumulated volume is known based on the number of droplets that are instructed (in
the data file) to fall onto the film. This is translated into expected mass based on the solids concentration of the binder, and then compared to the measured mass of the printed material. Measurements are made after the aqueous portion of the binder is allowed to evaporate. The table below lists the experimental conditions.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binder Material</td>
<td>Colloidal Silica, 30%</td>
</tr>
<tr>
<td>Substrate</td>
<td>Kodak Plus-X Pan Film 4147 ESTAR</td>
</tr>
<tr>
<td>Binder Flow Rate</td>
<td>0.69 ml/min</td>
</tr>
<tr>
<td>Droplet Frequency</td>
<td>64.4 kHz</td>
</tr>
<tr>
<td>Printhead Speed</td>
<td>0.64 m/s</td>
</tr>
<tr>
<td>Data File</td>
<td>19950908.3dp</td>
</tr>
</tbody>
</table>

Each measurement consisted of 256 lines with 8000 droplets per line. The first level of droplet spacing (10 microns) is printed with no special deflection instructions; all droplets between the start and stop positions of printed lines are allowed to fall onto the collection substrate. The second level of droplet spacing (20 microns) is achieved by instructing the printhead to catch every other droplet. Likewise, the fraction of droplets allowed to fall onto the substrate is halved for the next two cases. Proper deflection performance would require that the measured mass for the 20-micron case be half of the value for the 10-micron case.

The following table presents the measured results (in milligrams) after adjusting for the tare weight of the substrate film. Each sheet of film was precisely measured to within 0.1 mg. Two replicate measurements were made at each value of droplet spacing. Each time the droplet spacing is doubled, the mass should be reduced to half of the previous value. The binder used in the experiment was colloidal silica with density 1.2 g/ml, 30% of which is silica. The droplet volume is calculated from the flow rate and droplet generation rate as 179 pL.
Table 5.7 Measured Mass of Lines under High-Frequency Deflection

<table>
<thead>
<tr>
<th>Spacing (µm)</th>
<th>Expected (mg)</th>
<th>Sample 1 (mg)</th>
<th>Sample 2 (mg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>167.2</td>
<td>165.2</td>
<td>169.1</td>
</tr>
<tr>
<td>20</td>
<td>83.6</td>
<td>82.7</td>
<td>83.7</td>
</tr>
<tr>
<td>40</td>
<td>41.8</td>
<td>42.6</td>
<td>43.0</td>
</tr>
<tr>
<td>80</td>
<td>20.9</td>
<td>23.0</td>
<td>22.9</td>
</tr>
</tbody>
</table>

The plot below confirms that the deflection instructions are realized correctly. The continuous reference line begins at the average of the two measurements for the 10-micron droplet spacing and continues as a curve constrained to the characteristic form $f(x) = 1/x$. The 10-micron case was printed with no special deflection instructions, so it offers a reliable reference measurement. The measured data points with larger values of droplet spacing (20, 40, 80) fall closely along the curve with the expected reciprocal behavior.
5.4 Leading Edge Effects

The previous section has shown that the machine performs as expected, with respect to deflected line segments. However, some physical effects have been observed to locally distort the shapes of the leading edges of droplet trains. The following image shows the expected flight arrangement of a train of droplets. The droplets directed toward the powder bed are on the left and have no charge. On the right are a few deflected droplets that will be collected in a catcher.

![Figure 5.17 Leading Edge of a Droplet Train](image)

The leading edge of the droplet train on the left is subject to distortion from aerodynamic drag and from unintentional electrostatic interaction. The possible interactions are extremely complex as noted in [Fillmore], and many of the physical effects have been studied in relation to 3D Printing by [Curodeau]. Two of the most basic observations are described in the sections below. Process planning is not specifically concerned with correcting these problems because they are considered machine control issues. However, in the interest of measurable improvement, a
special printing technique (Section 5.4.3) has been implemented to overcome these leading edges effects.

5.4.1 Aerodynamic Drag

Several models have been developed in the field of fluid mechanics to determine aerodynamic drag force on a body moving through a medium (such as air). Many of these models are based on empirical data. The drag force has been found to increase with the frontal area at which the bodies fly through the medium. In a train of droplets, the lead droplet has much more frontal area exposed than do the subsequent droplets. Most other droplets are hidden within the wake of their predecessors, and thus subject to less drag force.

This difference in drag force results in the possibility of merging droplets. In fact at typical flight distances (~20 millimeters), three or four of the leading droplets may merge before hitting the powder bed. The next image shows two lead droplets that have already combined, and merging with the third droplet is pending. This behavior for the 3DP printhead has been studied in quantitative detail by [Curodeau]. Merging at the leading edges of droplet trains is a problem because it distorts the shapes of printed line segments. Furthermore, the merged droplets have higher ballistic momentum as they impact the powder bed.
A fluid environment with very low drag coefficient (e.g. helium) would reduce the consequences of drag, but such an approach is very difficult from a practical standpoint.

5.4.2 Mutually Induced Charge

By design, the only mechanism by which a droplet should receive deflection information is via electrostatically induced charge from the charging cell of the printhead. The diagram below illustrates the proper charging mechanism. The charging cell acquires a voltage level from a computer-controlled digital-to-analog source, based on the desired deflection specified in the data file. The continuous part of the fluid stream is grounded by contact with the nozzle from which it exits. This electrode arrangement allows a local imbalance of charge near the break-off point of the fluid stream. When the charging cell has a positive voltage level, a negative charge will be induced upon a droplet that breaks away from the stream. Each droplet retains its charge level at the time of break-off, because there is no means for charge transfer once the droplet is in flight. Increasing the magnitude of charging cell voltage will increase the magnitude of charge (with opposite sign) on the droplets.
This control scheme offers a fast, accurate way to control droplet deflection because the time constant for adjusting the charging cell voltage is very short (typically 5 to 10 nanoseconds [Curodeau]). However, mutually induced charge among neighboring droplets presents a problem. Evidence of this effect is shown below.

Figure 5.20 Mutually Induced Charge Effect
The last few droplets in the deflected train (right) carry a very high level of charge (enough to send them into a catcher). The first droplet in the undeflected train (left) receives its induced charge not only from the charging cell as intended, but also from the droplets immediately below it. Even if the charging cell is set to zero voltage, the last few (highly charged) droplets of the deflected train induce a charge of opposite sign upon the new leading edge. After this leading edge breaks from the stream, this inadvertent charge causes it to be deflected, even though no deflection was intended.

### 5.4.3 Leading Edge Concealment

There are a number of strategies that attempt to compensate for leading edge effects [Fillmore], and in fact many of these techniques have been investigated by [Curodeau]. For example, the mutually induced charge effect can be nullified by deliberately applying a "counter-charge" to lead droplets. Likewise, the merging problem can be alleviated by special charge sequences. One strategy is to remove a few droplets behind the lead droplet, so that when it is retarded by drag it does not merge with other droplets.

Unfortunately, special droplet correction sequences require a tremendous amount of overhead in terms of characterization, computation, and/or data storage. A common approach for implementing special printing sequences is to use look-up tables based on the desired outcome. [Curodeau] has done some initial work towards characterizing what correction sequences should be used under various circumstances.

A printing technique called *leading edge concealment* (also called "exits-only" printing) has been developed, in pursuit of an immediate remedy for the problems mentioned above. When printing
with lead droplet concealment, all leading edges of droplet trains are hidden within the interior of the printed cross-sections. All exterior boundaries are defined using trailing edge droplets only, since these droplets do not suffer from merging and mutually induced charge. The following diagram illustrates the concept of leading edge concealment.

![Diagram](image)

**Figure 5.21 Leading Edge Concealment**

Hiding the leading edges improves surface definition. However, the improvement in surface quality comes with a penalty in build rate. Printing in this "exits-only" fashion requires twice as many printhead passes per layer, compared to bi-directional printing. The point at which lines are split is randomized, in order to avoid any visible seam or structural fault line. A threshold may be selected to specify the minimum length that may be split, because it is impractical to split very short line segments.
<blank page>
6. Datafile Processing

Three Dimensional Printing is an exceptionally flexible manufacturing process because it has the capability to direct each step of fabrication on a point-by-point basis. Each instruction must be represented in digital format for control of the machine and printhead. Therefore datafile processing is a central concern for 3D Printing. CAD-driven information is pre-processed off-line, and only real-time adjustments are made at the machine level. The interpretation of overlapping volumes in the CAD model is an example of a pre-processing responsibility, while adjustments for droplet trajectory is an example of an on-line control issue. The surface-enhancement techniques in this thesis focus specifically on the improvements that can be made during off-line processing.

Parts are designed using many different CAD systems which utilize a wide variety of representations including solid modeling, boundary representations, and spline surfaces. However, the rapid prototyping community has adopted a de facto standard called the STL File Format [Jacobs] for the exchange of 3-D models. This 3-D representation must be first sliced into layers and then translated into printhead passes and transitions, before it can be fabricated by the 3D Printing machine. The conversion of 3-D surface information into low-level printing instructions is the topic of this chapter. The first part of the chapter describes the geometric representations used at various stages, and the remaining sections explain the algorithms by which file processing is executed.
6.1 Geometric Representations

There are four main levels of geometric representation used in file processing for 3D Printing:

- Surface Facets
- Polygonal Slices
- Raster Segments
- Printing Transitions

Each level represents a transformation of the CAD model into a more discrete and machine-specific form. The sections below discuss each representation in greater detail. This part of the chapter is primarily provided to define a collection of terms which are be used extensively in Section 6.2.

6.1.1 Surface Facets

Surface information from various CAD systems is divided into triangular patches and exported as an STL file. Each patch is called a *facet*. The name STL is an acronym for stereolithography [Jacobs], the process for which the file format was originally established. The division into triangular facets is often called *tessellation*. The illustration on the right side of figure below is an STL representation of a square-base pyramid.
An STL file contains a list of triangular facets, defined by the coordinates of its three vertices. Each triangle shares two common vertices with each adjacent triangle, and each vertex is common to at least three triangles. Each facet has a corresponding outward normal vector that identifies the exterior of the 3-D surface. Also, the vertices are ordered in a "Right Hand Rule", such that when the fingers of the right hand curl from the first vertex through the second to the third, the thumb points in the direction of the outward normal.

The STL file format is defined explicitly in the Sterolithography Interface Specification [3D Systems, Valencia, CA]. The file may be stored as a text file or a binary-compressed file, but each type of file contains the same basic information and both are arranged in a similar fashion. The units are unspecified and negative coordinates are not supported. The facets are typically not ordered, and appear as a simple list of grouped coordinates as shown below:

<table>
<thead>
<tr>
<th>Triangle 1</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal i</td>
<td>j</td>
<td>k</td>
</tr>
<tr>
<td>Vertex 1</td>
<td>x1</td>
<td>y1</td>
</tr>
<tr>
<td>Vertex 2</td>
<td>x2</td>
<td>y2</td>
</tr>
<tr>
<td>Vertex 3</td>
<td>x3</td>
<td>y3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Triangle 2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal i</td>
<td>j</td>
<td>k</td>
</tr>
<tr>
<td>Vertex 1</td>
<td>x1</td>
<td>y1</td>
</tr>
<tr>
<td>Vertex 2</td>
<td>x2</td>
<td>y2</td>
</tr>
<tr>
<td>Vertex 3</td>
<td>x3</td>
<td>y3</td>
</tr>
</tbody>
</table>
The total number of facets in an STL file is stated explicitly within the file in the binary version, but needs to be counted by parsing the entire file in the text version. The total number of vertices in an STL file is three times the number of facets. Edges are not written explicitly within the file, but are implicitly located between each pair of vertices in a triangle.

An properly constructed STL object must have a completely closed shell. Every edge is the boundary between exactly two faces, and every facet shares exactly three edges with exactly three other facets. Every vertex contacts at least three facets and at least three edges. Each facet must meet all adjacent facets along a common edge. A vertex may not lie on the edge of an adjacent facet. This restriction is named the "vertex-to-vertex" rule, and the principle is illustrated below.

![Valid Faces](image1.png)  ![Valid Faces](image2.png)  ![Invalid Face](image3.png)

**Figure 6.2 Vertex-to-Vertex Rule for STL Facets**

The triangular facet representation has several limitations and enhanced representations have been proposed and implemented for some applications [Vuyyuru, Wonzy]. For example, planar facets are imperfect because there is an error associated with using planar patches to approximate curved surfaces. The next figure illustrates an example of approximation error for a circular arc.
(more complex curves are certainly possible). The modules which export STL files from the native CAD software typically allow the designer to specify a maximum chord length, also shown in the diagram.

![Diagram showing approximation error and chord length]

**Figure 6.3 Approximation Error Between Planar Facets and a Circular Arc**

The magnitude of the error can be decreased by using a shorter chord length (and consequently more triangles). For example, if the curve is circular with radius $R$ and the chord length is $L$, then the maximum error $\varepsilon$ is as follows.

\[
\varepsilon = R - \sqrt{R^2 - (L/2)^2}
\]

Equation 6.1

The algorithms specific to 3D Printing are developed downstream of this facet approximation and are not directly concerned with this approximation error. The routines, however, must be designed to process a very large number of triangles if necessary (as high as several hundred thousand).

### 6.1.2 Polygonal Slices
Slicing the 3-D facets into layers results in a stack of 2-D planes which represent cross-sections of the original 3-D object. Each layer contains a collection of polygons. A polygon is here defined as the region of a plane defined by an ordered sequence of vertices that form a simple closed curve. A 2-D vertex is simply a point in the 2-D coordinate plane. An edge of a polygon is the straight line segment connecting two consecutive vertices. Each edge has a direction implied by sequential order of its vertices. The first vertex of an edge is considered the source vertex and the second is the destination vertex.

Alternative representations offer some performance advantages for operations such as finding intersections, especially if each polygon has a large number of edges [Burton]. However, the speed of traversing polygons is not a critical issue in the processing stages of 3DP data files, so the ordered vertex representation is still preferred for conciseness.

The sense of a geometric object indicates its direction of rotation. The sense of a polygon is used to indicate whether the polygon bounds a filled region or an empty region. The direction is determined by the order in which the vertices are listed. A polygon with counter-clockwise sense is interpreted as filled, and one with clockwise sense is considered empty. When traversing the boundary of a polygon in the direction of its ordered vertices, the region to the left side will always be filled, and the region to the right will always be empty.
The slice file format which has been used by 3D Printing developers is included as an appendix. File names in this format carry a suffix ".SLC" (internal to MIT, not to be confused with a similar naming convention used by 3D Systems). Nearly any file format that can store multiple layers of polygon data is adequate for this level of representation. There are many restrictions governing a "proper" polygonal slice file. Robust algorithms may be able to interpret violations of these rules (in fact the rastering program described later in this chapter does decipher many ambiguities). However, meeting the following requirements greatly increases the chances for flawless design interpretation.

- Positive-space coordinates
- Closure
- Minimal representation
- No self-intersection
- No mutual intersection (overlap)

Closure is automatically insured by the SLC file format, because the last vertex is always assumed to be connected to the first. This may not be true for all polygon file formats. Another requirement related to closure is that each polygon must have at least three vertices.

A polygon is in its minimal representation if it has no redundant vertices and no sequence of three (consecutive) vertices on a line. The following diagram contrasts non-minimal polygons with their minimal counterparts for a simple square example.
Figure 6.5 Non-Minimal Representation of Polygons

A polygon exhibits *self-intersection* if any non-consecutive edges are in contact, including endpoints, or if two consecutive edges intersect at more than one point. The next figure shows basic ways in which a polygon may contain a self-intersection. A small gap is drawn at the intersection to distinguish separate entities; the actual data would be flush. A pathological polygon may contain several of these self-intersections in combination.

Figure 6.6 Polygon Self-Intersection

Self-intersection often leaves a polygon with ambiguous sense, as shown below. Ill-defined sense is a major problem because it is unclear which regions were intended to be printed and which were not. The absence of self-intersection guarantees that the polygon sense will also be correct.
Figure 6.7 Polygon Self-Intersection and Ill-Defined Sense

The next diagram shows basic ways in which two polygons may intersect. Again, a small artificial gap is drawn at the intersection to distinguish separate entities. A polygon wholly contained in another is not considered to be overlapping. More complex intersections may occur as combinations of the simple examples shown here.

Figure 6.8 Mutually Intersecting Polygons

6.1.3 Raster Segments

A raster representation of geometric information in a space is a set of discrete samples [Brown]. Most often the data is organized in a grid of rows and elements as shown below. In computer graphics each element is often called a pixel. An uninterrupted sequence of elements with the same value in any one row is considered a raster segment.
The major characteristic of raster data is that it is discrete, making it clearly distinct from the relatively continuous representation used for polygonal slices. This difference has both advantages and disadvantages. Major disadvantages are the inability to follow contours exactly and the lack of associative information among individual elements. Key advantages include the fact that the information is readily interpreted in digital form and the requirement that every addressable element must have an explicit, discrete state.

In the most basic implementation of raster data, each element holds only one of two binary values, in which case the raster space is called a bitmap. However, it is often necessary to enhance the richness of information with a larger number of possible values per element. A common example is the representation of darkness levels in 2-D gray-scale images.

Geometric data in 3D Printing must be converted ultimately into a digital representation, so the polygon data is translated into raster segments (on a layer-by-layer basis). The interpretation of geometric data such as polygon vertices into raster data is called scan conversion. It is possible to perform the scan conversion directly into encoded printhead transitions because in many ways the transitions are like raster segments. However, this alternative is not chosen for important reasons discussed in Section 6.2.
The pursuit of high surface quality requires that great attention be given to boundaries. Therefore the raster representation for 3D Printing uses multiple levels per element, to identify not only on/off information but also boundary segments. Boundary segments are best explained pictorially, as below. In the figure, boundary segments are highlighted with thicker lines. A vertical sequence of elements is not considered a segment because it crosses multiple rows, so by definition there are no vertical boundary segments. In some cases a segment must be divided because only part of it lies on a boundary.

![Figure 6.10 Raster Boundary Segments](image)

Raster data almost always requires compression because high-resolution demands a very large number of elements. A single layer of data for the current 3D Printing machine with 10-micron resolution in each axis may occupy nearly half a billion spatial elements, and complete files often have hundreds of layers. Run-length compression [Brown] is a straight-forward technique used by 3D Printing for managing data size. Repetitive sequences are replaced by a single value and a run length. The following is an example of run-length compression.
The raster file format currently used by 3D Printing is called an *RST* file. The specification of this format is included as an appendix.

6.1.4 Printing Transitions

The representations presented thus far are all process-independent and machine-independent, except for the fact that the polygon data is organized in layers. The layer spacing is usually influenced by process constraints. Raster data is typically processed with a resolution at least as fine as would be needed by any machine, so it is therefore considered independent of machine type. The final level of representation, however, is entirely machine-specific. Machine-encoded data consists of axis positions, jet assignments, and printhead instructions.

The printing cycle begins with the powder bed platform positioned at a specified vertical axis position. Then the slow axis is positioned step by step across each layer. The actions during each pass along the fast axis are dictated by information stored in *transitions*. A typical 3D Printing transition consists of a position along the fast axis, a jet number, and a deflection value. A deflection value of zero corresponds to printing on-center with no deflection. An arbitrarily designated high value (e.g. 1000) corresponds to no printing at all, which is achieved by
deflecting the stream into a catcher. Intermediate values enable partial deflection. A special transition type for pattern memory is used to initiate fill patterns.

Each addressable position along the fast axis may hold a unique transition, although the data file only specifies "new" transitions that would effect a physical change. The current machine has approximately 30,000 positions, separated by 10 microns. The density of new transitions along a pass is typically sparse, so the remaining memory addresses are simply filled with information that indicates "continue previous" transition. The next figure presents an example of how transitions are interpreted by printhead electronics. The printhead begins at one end of the fast axis with all jets in a catch (off) position. The tilde symbol (~) represents a "continue previous" condition, and only the explicit entries are what would be specified in the datafile. The solid straight lines represent the deflection positions as the printhead passes from left to right along the fast (X) axis.

![Graph showing printhead transitions](image)

**Figure 6.12 Printhead Transitions Example (Two-Jet Printhead)**

The spacing between jets in the example above is not to scale, because in reality the jet spacing (~4000 microns) is much greater than the fast-axis resolution (10 microns). It is also important to note that the transitions are sensitive to the direction in which the printhead travels. The
outcome would not be the same if the printhead were moving from right to left in the example above.

The position, jet number, and deflection value for each transition are compressed into a data "word". Each transition is encoded in a binary format designed for fast downloading to hardware memory at the machine level. A machine-encoded file is called a 3DP file, and a detailed specification is given as an appendix.

6.2 Data Interpretation

The previous sections described the basic levels of representation used in datafile processing: surface facets, polygonal slices, raster segments, and encoded transitions. This section explains how each of these representations is derived from its predecessor. The three transformations are slicing, rastering, and encoding. Slicing is the process of dividing a 3-D surface model in STL file into 2-D layers. Rastering converts polygon data from an SLC file into a complete digital scan of the information. The figure below illustrates a simple example of slicing and rastering. Encoding (not shown) reduces the geometric raster data into instructions designed for the 3D Printing machine. Some information loss is incurred as the data becomes more machine-specific, so the data conversions are not reversible in most cases.

![Diagram of Slicing and Rastering](image)

**Figure 6.13 Slicing and Rastering**
Another important activity in file preparation is the *composition* of a production batch, because it is economically more efficient to print several parts per run, taking advantage of the powder bed capacity. Composition includes the arrangement, orientation, and scaling of multiple parts. Compensation for shrinkage during heat treatment is also performed during the composition stage of file preparation.

### 6.2.1 Slicing

Slicing is the procedure by which 3-D surface geometry is interpreted on a layer-by-layer basis. A slicing grid is established with a fixed layer spacing, and two-dimensional cross-sections are extracted at each slice plane. Slicing performs the functions listed below. The facets in an STL model do not have any connectivity information, so it is necessary to establish this connectivity in the 2-D plane in order to form closed polygons.

- Organize the CAD model in layers
- Determine plane intersections
- Connect polygon edges

The only major parameter for the slicing operation is the resolution. The *slice resolution* \((dz)\) is the spacing at which a 3-D STL model is divided into a stack of 2-D planes. It has the same value and practical meaning as layer spacing, but has a more purely geometric connotation.

There are many possible strategies for actually performing the slicing operations, but the fundamental operation is determining the intersection of a horizontal plane and the triangle formed by three unique points in space. The table below lists the complete set of possible spatial
relationships between a plane and a triangle. Each vertex can only have exactly one of three states with respect to the plane: below, within, or above. There are three vertices with three possible states, so there must be $3^3 = 27$ possibilities.

Table 6.1 Relative Positions of a Triangle Vertices and a Horizontal Plane

<table>
<thead>
<tr>
<th>Vertex 1</th>
<th>Vertex 2</th>
<th>Vertex 3</th>
<th>Vertex 1</th>
<th>Vertex 2</th>
<th>Vertex 3</th>
<th>Vertex 1</th>
<th>Vertex 2</th>
<th>Vertex 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Below</td>
<td>Below</td>
<td>Below</td>
<td>Within</td>
<td>Below</td>
<td>Below</td>
<td>Above</td>
<td>Below</td>
<td>Below</td>
</tr>
<tr>
<td>Below</td>
<td>Below</td>
<td>Within</td>
<td>Within</td>
<td>Below</td>
<td>Within</td>
<td>Above</td>
<td>Below</td>
<td>Within</td>
</tr>
<tr>
<td>Below</td>
<td>Below</td>
<td>Above</td>
<td>Within</td>
<td>Below</td>
<td>Above</td>
<td>Above</td>
<td>Below</td>
<td>Above</td>
</tr>
<tr>
<td>Below</td>
<td>Within</td>
<td>Below</td>
<td>Within</td>
<td>Within</td>
<td>Below</td>
<td>Above</td>
<td>Within</td>
<td>Below</td>
</tr>
<tr>
<td>Below</td>
<td>Within</td>
<td>Within</td>
<td>Within</td>
<td>Within</td>
<td>Within</td>
<td>Above</td>
<td>Within</td>
<td>Within</td>
</tr>
<tr>
<td>Below</td>
<td>Within</td>
<td>Above</td>
<td>Within</td>
<td>Within</td>
<td>Above</td>
<td>Above</td>
<td>Above</td>
<td>Within</td>
</tr>
<tr>
<td>Below</td>
<td>Above</td>
<td>Below</td>
<td>Within</td>
<td>Above</td>
<td>Below</td>
<td>Above</td>
<td>Above</td>
<td>Below</td>
</tr>
<tr>
<td>Below</td>
<td>Above</td>
<td>Within</td>
<td>Within</td>
<td>Above</td>
<td>Within</td>
<td>Above</td>
<td>Above</td>
<td>Within</td>
</tr>
<tr>
<td>Below</td>
<td>Above</td>
<td>Above</td>
<td>Within</td>
<td>Above</td>
<td>Above</td>
<td>Above</td>
<td>Above</td>
<td>Above</td>
</tr>
</tbody>
</table>

The following figure summarizes these possible conditions, organized based on how many vertices are below, within, and above the slice plane. The diagram is only two-dimensional, so the triangles may have arbitrary projections perpendicular to the page.
Figure 6.14 Relative Positions of Triangle Vertices and a Horizontal Plane

The triangles are listed in the STL file independent of each other. Therefore searching the data space otherwise becomes quite crude and time-consuming because the STL models contain only vertex information. Often it is beneficial to reinterpret the STL data with connectivity information before intersection computations [Wonzy]. Another approach would be to process each facet serially from the file, storing only the resulting intersections in computer memory. Polygon connectivity is established later, by joining segments in close proximity. A number of commercial vendors offer software packages that perform slicing, so it has not been necessary to develop such a program at MIT. Examples of current systems that perform STL slicing are
Rapid Prototyping Module by Imageware (Chelmsford, MA) and Rapid Tools by DeskArtes Oy (Finland).

6.2.2 Rastering

Rastering reduces a polygonal slice file (SLC) into a raster data file (RST). Region boundaries are converted into digitized point data with specific states. The fundamental rastering functions are as listed below.

- Scan Edge Intersections
- Resolve Overlap States
- Identify Boundary Segments

Essential rastering parameters are the resolutions along the two orthogonal axes in each slice. Raster resolution refers to the grid size during the rastering stage of file processing. This resolution is most typically, but not necessarily, uniform along the y-axis and x-axis. Along the y-axis the resolution refers to the spacing between rows, and is labeled $dy$. Along the x-axis the resolution refers to the spacing between elements in each row, and is abbreviated as $dx$. The values for $dy$ and $dx$ are constant across all layers in a file.

The raster resolution along the x-axis is typically set to 10 microns, which equals the resolution of the position encoder on the machine. The y-axis resolution is typically between 10 and 20 microns. The y-axis resolution should not be made unnecessarily fine because it may drastically increase the file size. Emphasis on surface quality is usually placed above minimization of production rate or file size, so the y-axis resolution is often chosen to be very fine (10 microns).
Printing in a "draft" mode for less demanding applications would be an exception to this preference.

**Edge Intersections**

Analytic geometry with lines and slopes conveniently finds the edge intersections between raster scans and polygon edges. Edges are defined by two endpoint vertices. Each vertex must have one of three possible states with respect to the scan line: below, within, or above. Therefore, there are $3^2 = 9$ possible arrangements, as listed here.

<table>
<thead>
<tr>
<th>Vertex 1</th>
<th>Vertex 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Below</td>
<td>Below</td>
</tr>
<tr>
<td>Below</td>
<td>Within</td>
</tr>
<tr>
<td>Below</td>
<td>Above</td>
</tr>
<tr>
<td>Within</td>
<td>Below</td>
</tr>
<tr>
<td>Within</td>
<td>Within</td>
</tr>
<tr>
<td>Within</td>
<td>Above</td>
</tr>
<tr>
<td>Above</td>
<td>Below</td>
</tr>
<tr>
<td>Above</td>
<td>Within</td>
</tr>
<tr>
<td>Above</td>
<td>Above</td>
</tr>
</tbody>
</table>

These cases are summarized below, based on the number of vertices that are below, within, or above the scan line.
The most common type of intersection (other than no intersection at all) is one in which the scan line crosses the polygon edge through exactly one point. This condition is detected by the following rule.

\[
\text{IF the scan line lies above one vertex of the edge} \\
\text{AND the scan line lies below the other vertex} \\
\text{THEN the scan intersects the edge at exactly one point.}
\]

Equation 6.2 shows one standard way of determining the x-intercept for a scan at \( y \) through a line segment from \((x_1, y_1)\) to \((x_2, y_2)\). The equation is valid as long as the segment does in fact cross the scan line at \( y \). The relative positions of the two endpoints (which one is higher) does not matter. All coordinates are in positive space.

\[
x = x_1 + \frac{x_2 - x_1}{y_2 - y_1} \cdot (y - y_1)
\]

Equation 6.2
Raster scanning not only detects an intersection, but also the way in which the intersection affects state change. The state changes are expressed in terms of movement from left to right, in the direction of the positive x-axis. These state changes are contrasted below.

![Diagram showing "ON" and "OFF" state changes](image)

**Figure 6.16 State Changes of Raster Intersections**

It is always possible to determine the way in which the state changes for simple intersections. The order in which the two vertices are listed (in the SLC file) reveals this information, as explained in the following two rules:

1. **IF** the first vertex of a polygon edge is below the scan line,  
   **AND** the second vertex is above the scan line,  
   **THEN** the state changes from filled to empty as the scan crosses the edge.

2. **IF** the first vertex of a polygon edge is above the scan line,  
   **AND** the second vertex is below the scan line,  
   **THEN** the state changes from empty to filled as the scan crosses the edge.

**Vertex Intersection Rules**

Computing intersection values when a scan line runs through one vertex (but not the other) is trivial, because the coordinates are known exactly. However, not all of the intersections of scan lines through vertices are stored. Some vertex intersections must be ignored to insure that the number of on-off state changes through a polygon is always even. The figure below shows the
numerous ways in which a scan line may intersect a vertex. Only the cases for which the center vertex contributes relevant raster information are shaded.

Figure 6.17 Scan Line Intersection through a Vertex

The cases form a comprehensive set of "approaches" and "departures" with respect to a central vertex. Using a compass analogy, the approaches are made in the directions north, south, west, east, northeast, southwest, northwest, and southeast. Whether the vertex is retained (shaded cases) or ignored depends on the relative positions of the neighboring vertices.
The following rules identify a fully-consistent set of conditions for which a scan intersection through a polygon vertex should be stored (i.e. not ignored). These conditions are chosen so that

*a scan line can only intersect a polygon with an even number of intersections.* The rules are written in the affirmative sense, but actual code implementation it may even be more convenient to examine them in the negative sense (if not...then...).

IF a scan intersects a vertex,
AND the next vertex is below the scan,
AND the previous vertex is above the scan,
THEN the state changes from empty to filled going left to right.

IF a scan intersects a vertex,
AND the previous vertex is below the scan,
AND the next vertex is above the scan,
THEN the state changes from filled to empty going left to right.

IF a scan intersects a vertex,
AND the scan also intersects the next vertex,
AND the previous vertex is above the scan,
AND the next vertex is right of the original
THEN the state changes from empty to filled going left to right.

IF a scan intersects a vertex,
AND the scan also intersects the next vertex,
AND the previous vertex is below the scan,
AND the next vertex is left of the original
THEN the state changes from filled to empty going left to right.

IF a scan intersects a vertex,
AND the scan also intersects the previous vertex,
AND the next vertex is below the scan,
AND the previous vertex is right of the original
THEN the state changes from empty to filled going left to right.

IF a scan intersects a vertex,
AND the scan also intersects the previous vertex,
AND the next vertex is above the scan,
AND the previous vertex is left of the original
THEN the state changes from filled to empty going left to right.
Overlap States

A polygonal slice file that was created properly should have no overlapping polygons. However, such conditions are often encountered in practice, because the sliced polygon files may come from a variety of sources that do not exercise strict conformity. The degree of overlap measures the number of times a point is encircled by polygons. Filled polygons (counter-clockwise) contribute +1 to the degree of overlap, and empty polygons (clockwise) contribute -1. This is analogous to the "wrap number" used in [Barton]. The degree of overlap is a particularly useful concept when interpreting intersections and unions among polygons, as shown in this example.

![Degrees of Overlap](image)

**Figure 6.18 Degrees of Overlap**

Each scan through a field of overlapping polygons maintains a record of the degree of overlap. Every ON intersection adds +1 to the overlap state, and every OFF intersection contributes -1. After a scan is completed, only those raster segments that have a positive value of this overlap state are considered to be filled regions.
Boundary Segments

The rastering procedure has the enhanced feature of identifying those scan segments which are on the exterior of the geometric data. These exterior raster segments are called *boundary segments* (Figure 6.10). A straight-forward method of determining whether or not a segment is on the boundary is to check for the presence of and adjacent segment on one side, and lack of an adjacent segment on the other. Two raster segments are considered adjacent if they occur on consecutive raster scans and overlap along the x-axis. Figure 6.19 shows the nine ways in which raster segments can be adjacent, with respect to the relative positions of their start and stop points. There are nine cases because there are two pairs of vertices which can be examined, a start pair (shown with black squares) and a stop pair (shown with white squares). Each pair may have three possible arrangements with respect to the x-axis (one before the other, equally aligned, or one after the other).

\[ \begin{array}{ccc}
  & y & \\
  x & \quad & \\
  \quad & \quad & \quad \\
 \end{array} \]

![Figure 6.19 Adjacent Raster Segments](image)

It is also possible to detect border segments by marching around polygon edges during raster-scanning, but the adjacency approach is more convenient for identifying partial border segments. The following rules check if two segments are adjacent. The segments are adjacent if any of the rules apply. If none of the rules apply, the segments are not adjacent.
IF two raster segments occur on consecutive scan positions AND the start of the first lies between the endpoints of the second THEN the segments are adjacent

IF two raster segments occur on consecutive scan positions AND the stop of the first lies between the endpoints of the second THEN the segments are adjacent

IF two raster segments occur on consecutive scan positions AND the start of the second lies between the endpoints of the first THEN the segments are adjacent

IF two raster segments occur on consecutive scan positions AND the stop of the second lies between the endpoints of the first THEN the segments are adjacent

IF two raster segments occur on consecutive scan positions AND their endpoints are identical THEN the segments are adjacent

It may be preferable to check for non-adjacency to reduce the number of computations while identifying border segments. A segment can be removed from consideration as a segment is found to be non-adjacent. The following rules check for non-adjacency.

IF two raster segments do not occur on consecutive scan positions THEN the segments are non-adjacent

IF the start of a segment is greater than the stop of another THEN the segments are non-adjacent

IF the stop of a segment is less than the start of another THEN the segments are non-adjacent

Segments that are only partially on a boundary are divided into shorter segments, so that each segment in the raster table is either an interior segment or a boundary segment. In Figure 6.10, two longer segments were "broken" to identify those portions that were on the exterior. The boundary segments are marked so that they may receive special attention when placing droplets.
6.2.3 Encoding

Encoding takes a raster data file (RST) as input and creates a file (3DP) that is readily interpreted by the 3D Printing machine. It is fully possible to encode the polygonal slices directly, and in fact the initial implementation of the data processing programs did function in such a manner. However, experience with a large number of CAD models and numerous flaw conditions proved that having a distinct rastering step was invaluable for detecting inconsistent data and for correcting errors. The illustration below shows an example of a complex, improper arrangement of polygons.

![Figure 6.20 Improper Arrangement of Polygons](image)

Rastering polygons into scan lines is a purely geometric operation, whereas encoding involves managing physical parameters and constraints such as the spacing between jets on a particular multi-jet printhead. The raster domain is a much simpler environment for applying rules that deal with exceptions and flaws in geometric logic. Fundamental encoding responsibilities are listed below. Key parameters are lane width (or row spacing), number of jets, jet spacing, and print direction.

- Assign scan positions along slow axis
- Divide printing bands and coordinate jet assignments
- Interpret directionality
- Binary-encode transition data
The encoding process operates on a layer-by-layer basis. Each layer contains only raster scan lines, as output by the rastering procedure. Interpretation of the data proceeds in a hierarchical manner, beginning with a layer. Attention then narrows to a single (multi-jet) printhead pass, and finally focuses on the data covered by a single jet.

After all of the raster scan lines in a layer are loaded, the slow axis position is initialized, either to the lowest position of the layer data or to a fixed machine reference. Initializing with respect to the layer data offers the greatest efficiency in production rate (by using the minimum number of passes to cover the layer), but initializing with respect to a machine reference may be preferable for physical constraints in machine design. The following figure contrasts these initialization alternatives. In either case, printing the layer should terminate as soon as the jets pass beyond the relevant cross-section.

![Diagram](image)

**Figure 6.21 Initial Position of Multi-Jet Printhead Along Slow Axis**

The fast axis origin and directionality are also pre-determined upon initialization. Every odd-numbered pass is a forward pass with increasing x-coordinate, and every even-numbered pass is a reverse pass with decreasing x-coordinate. The most simple directional mode is described as
uni-directional, in which printing occurs during forward passes only, and reverse passes are used simply to return the printhead to its fast axis origin. A more time-efficient printing mode is described as bi-directional, in which printing occurs during both the forward and return passes. Adjustments for projectile motion must be made by on-line control systems so that the bi-directional passes align correctly.

The printhead position along the slow axis is referenced according to the position of the first jet. Once the position and direction of the first jet is established for a particular pass, the positions of all other jets is necessarily fixed. The jets are numbered from 0 to n-1, assuming there are n jets. The expression below computes the slow axis position \( y_j \) for any jet, as a function of the first jet position \( y_0 \), the jet number \( j \), and the jet spacing \( \Delta j \).

\[
y_j = y_0 + j \cdot \Delta j
\]

Equation 6.3

The next diagram shows the progression sequence for a two-jet printhead. The passes are numbered 1 through 8. The printhead advances at an increment equal to the line spacing unless it completes a band, at which time the printhead advances to the next band. A band is considered complete when any further advancement of the first jet would place it at a row position equal to or greater than a position already covered by the second jet.
Given a slow axis position $y$ and the positions of all jets for the pass, writing the output instruction is simply a matter of interpreting the raster information. The raster data is organized by row, and each row includes a position along the $y$-axis. For each jet, the encoding program needs only to look up the scan line data at its particular $y$-position. The raster procedure has already insured that the line segments are non-overlapping, so the encoding program simply translates on-off instructions from the raster representation to printhead toggle instructions.

The above sections have described the basic file processing steps in 3D Printing, without attention to the benefits offered by proportional deflection. Chapter 8 discusses more advanced techniques, many of which focus on more detailed droplet placement.
7. Parameter Selection

There are many basic parameters that govern the 3D Printing process. The parameters span the domains of file processing, machine settings, and material behavior. This chapter discusses those parameters that are most immediately relevant to process planning, and highlights the trade-offs among them. Some of these parameters have been mentioned briefly in earlier chapters as necessary. Here they are centralized and discussed more thoroughly. It is important to understand the basic relationships among parameters, because a primary challenge for process planning is to satisfy constraints. The special droplet placement algorithms developed in the next chapter may only be implemented within the restrictions set by the parameters discussed here.

Section 7.1 begins by describing process variables that can not be controlled at the process planning level, or at least can not be controlled on a run-by-run basis. Section 7.2 describes the parameters that may be selected before each production run, within a known range of values. Process planning has the responsibility of selecting these control parameters to satisfy numerous objectives including feature resolution and structural strength.

Parameters related to characteristic physical dimensions have been introduced in Chapters 3 and 4, and are not discussed in detail here. These include primitive diameter, line width, and layer thickness. Parameters related strictly to datafile processing have been described in Chapter 6, so these are also not included.
7.1 Fixed Parameters

The following is a description of fixed parameters that are important to process planning. Labeling these quantities as "fixed" does not imply that they are constant values, but rather makes a distinction from those values which are selected on a run-by-run basis. Some of these parameters are driven by machine and printhead requirements (e.g. droplet frequency), and others are the result of material behavior (e.g. packing density).

7.1.1 Flow Rate

The binder flow rate \( Q \) measures the volume of liquid that passes through a jet orifice per unit time. The 3D Printing machine has a closed-loop flow control system that is designed to maintain a constant time-of-flight for binder droplets. The target set point for flow rate is influenced by requirements related to printhead stability. For example, the point at which a droplet breaks away from its source stream should lie within the charging cell (refer to Chapter 5) for proper operation, and flow rate is one of the factors that affect this break-off point. Typical values for the current printhead are between 0.7 and 1.2 milliliters per minute.

The flow rate through a nozzle is easily measured by collecting the binder into a container over a known period of time (one minute has typically been used as a convenient standard). The binder dispensed from all jets is collected at once, and the total volume is divided by the number of jets to estimate the flow rate through each jet.
7.1.2 Droplet Frequency

The \textit{droplet frequency} ($f$) is the rate at which droplets are released from the droplet generator. Factors that determine the frequency of spontaneous drop formation from a stream include surface tension and density of the liquid. This "preferred" frequency is discussed in greater detail in [Heinzl]. The frequency may be regulated by an amplified signal from a function generator, and it is tuned in order to achieve the most reliable droplet break-off performance. Typical values for droplet frequency are between 40 and 80 kHz. The exact value is measured using an oscilloscope.

The droplet frequency indicates the number of droplets that are available per unit time. Process planning must consider this availability when specifying transition positions and printhead speed. In particular, the droplet frequency and printhead speed determine the droplet spacing along the fast axis (Chapter 5).

7.1.3 Powder Volume Fraction

The \textit{powder volume fraction} ($\nu$) in a given space is a measure of its packing density. The volume fraction is conveniently measured in the 3D Printing powder bed by weighing the mass of a known volume of powder that has been distributed into the bed. Numerous factors related to the way in which the powder is layered may affect the packing density [Michaels]. However, once the material and the machine settings have been fixed, the volume fraction has been observed to be consistent [Lee].
Process planning is concerned with powder volume fraction because it determines the amount of void space that is available for binder. In more complex ways, powder packing also affects feature formation and powder-binder interaction [Bredt, Charnarrong].

7.1.4 Jet Spacing

Jet spacing ($\Delta j$) is the center-to-center distance between any two consecutive jets on the printhead. This distance is the same for any pair of neighboring jets. The current printhead has a jet spacing of 4233 microns (1/6 of an inch). The limits are constrained only by printhead design. Design, however, faces the special challenge of arranging several fluid and electrical components with reliable performance in a small space. The jet spacing influences the selection of line spacing and the width of the printing bands (Chapter 6).

7.2 Control Parameters

The parameters listed here are specified each time a datafile is prepared. These variables are considered to be fundamentally important to the way in which printing is executed. The list is not comprehensive; Chapter 8 introduces a few more parameters as they become relevant to special printing techniques presented therein.

7.2.1 Layer Spacing

Layer spacing ($\Delta z$) is the top-to-top distance between two consecutive layers. The layer spacing is by default equal to the slice resolution (Chapter 6), and is typically constant throughout any given production run. The layer spacing is chosen to be an integer multiple of the positioning
resolution of the vertical axis (10 microns). Typical values for layer spacing lie between 100 and 200 microns. The lower limit for layer spacing is affected by the vertical stage resolution, particle size, and powder spreading capability. The upper limit is constrained by the need to stitch layers together.

Layer spacing affects build rate because it determines the number of layers which must be printed. The number of layers required to print an object is equal to the height of the object divided by the layer spacing. Layer spacing also influences saturation (discussed below). Finer layer spacing results in reduced layer spacing, and therefore increases the saturation.

7.2.2 Line Spacing

*Line spacing* ($\Delta y$) is the center-to-center distance between two non-deflected lines printed on adjacent printhead passes. This parameter is selected during the encoding stage of file processing. Typical values are between 170 and 212 microns.

The particular value for line spacing is largely determined by the ability to stitch lines together. Lines can not be placed arbitrarily far apart, because there must be adequate lateral overlap to provide structural continuity. Process planning is most simple when the line spacing is slightly less than the line width (Chapter 4), because stitching is guaranteed in such a case. Although wider fill patterns are made possible by proportional deflection, wide line spacing may make it difficult to accurately capture geometric features that may lie inside the grid. Saturation is another factor that affects the choice for line spacing. Lines can not be spaced arbitrarily close together, because closer line spacing results in a higher concentration of binder.
The range of possible values for line spacing are not continuous. Line spacing is typically chosen such that the number of lines divide evenly with respect to the jet spacing. This offers better spacing uniformity, as illustrated below.

![Diagram showing line spacing with respect to jet spacing](image)

**Figure 7.1 Selection of Line Spacing with Respect to Jet Spacing**

The jet spacing is fixed for a given printhead, but the line spacing is selected during file preparation. Line spacing remains uniform throughout the layer if the number of (equally-spaced) lines fits evenly into the spacing between jets. For example if the jet spacing is 1200 microns and the line spacing is 200 microns, each jet prints exactly 5 lines before the last line printed by Jet #0 "catches up" to the first line printed by Jet #1. This hypothetical situation is illustrated as "Preferred Line Spacing" in the figure above. However, 175 does not divide evenly into 1200, and would result in uneven spacing at the interfaces between jets.

### 7.2.3 Printhead Speed

The *printhead speed* \( v \) is the rate at which the jets pass over the powder bed. A closed-loop servo control system maintains constant speed over the entire bed. Maximum speed is naturally
limited by characteristics of the dynamic system. The top speed that can be maintained uniformly is approximately 1.5 meters per second.

Printhead speed is important to process planning because it determines droplet spacing and it directly affects binder saturation. The traverse speed of the printhead does not necessarily have a direct link to production speed. Build rate is dictated by the number of droplets that can be properly placed per unit time. The droplet generation rate is fixed, so it is possible to traverse slower without any loss in build rate. This presumes the ability to maintain a uniform concentration of droplets and an adequate level of saturation.

### 7.3 Derived Quantities

The following are important quantities in process planning, but ones that can be controlled only indirectly by altering other variables. It is often challenging to achieve desired values for these parameters, because there are many trade-offs as discussed previously in this chapter.

### 7.3.1 Droplet Spacing

*Droplet spacing* ($\Delta x$) is the center-to-center distance between two undeflected, consecutive droplets along the fast axis. The spacing is determined precisely by the droplet frequency $f$ and the printhead speed $v$ as given by the expression below. Typical values for droplet spacing are between 10 and 25 microns. Droplet spacing is distinct from the raster resolution $dx$ (Chapter 6) and the fast axis encoder resolution (fixed at 10 microns).
\[ \Delta x = \frac{v}{f} \]

Equation 7.1

The droplet spacing is not directly specified in standard practice. However, for advanced placement techniques (Chapter 8), it becomes necessary to control droplet spacing indirectly by specifying the printhead speed. The target droplet spacing is either 10 microns or 20 microns when the special algorithms are used. This corresponds to an alignment with the fast axis encoder ticks, at which specific deflection instructions can be triggered.

7.3.2 Saturation

Saturation \( (S) \) is the fraction of space not occupied by powder that is filled by liquid binder. Saturation may be computed in many equivalent ways, but for parameter selection it is most conveniently defined below. Other terms are flow rate \( Q \), layer spacing \( \Delta z \), line spacing \( \Delta y \), powder volume fraction \( \psi \), and printhead speed \( v \).

\[ S = \frac{Q}{\Delta z \cdot \Delta y \cdot (1 - \psi) \cdot v} \]

Equation 7.2

This defines saturation in a bulk sense, by using quantities such as layer spacing and line spacing. Saturation was introduced from a local standpoint when examining line diameter in Chapter 4. Bulk saturation has significant effect on the strength of printed parts [Bang], so each material system has a preferred range of saturation based on empirical results. Typical values for target saturation are between 70% and 100%. Saturation may influence roughness and accuracy if it becomes so high that the binder begins to bleed excessively.
Printing at a slower speed takes greater advantage of the positioning resolution along the fast axis. However, the closer proximity of droplets along the fast axis demands particular attention to prevent excessive saturation. The number of droplets per unit volume should be maintained at an acceptable concentration. For example, when printing with a standard system of colloidal silica binder into aluminum oxide powder, a combination of 170 µm layer spacing, 170 µm row spacing, and 20 µm droplet spacing works successfully. These parameters correspond to 1730 droplets per cubic millimeter. An increase in layer spacing and/or line spacing would be required in order to maintain a similar saturation level if 10 µm droplet spacing was desired. An alternative approach would be to discard a significant fraction of available droplets by deflecting them into the catcher of the printhead. This disposal of excess droplets would be used within the bulk regions of parts, while the maximum resolution would be applied at the surfaces.

Increasing the grid spacing (layers and/or lines) is preferred to discarding droplets because of the greater efficiency with respect to production rate. A wider grid spacing requires fewer layers and/or printhead passes in order to deposit the same amount of binder. However, greater layer spacing is undesirable because of its adverse effects on surface finish and resolution along the vertical axis. Increasing line spacing has its own complications, however. Depositing droplets along a line with tighter spacing results in a wider line, but adjacent lines do not necessarily stitch completely. Unless the droplets are diffused laterally, a layer may suffer from anisotropic strength and even disintegration.
<blank page>
8. Surface Enhancement Techniques

With the context and constraints discussed in the previous chapters, it is now appropriate to explain the process planning algorithms that have been designed to improve surface quality. Section 8.1 begins by examining how reducing layer spacing improves surface finish for shallow inclines from layer to layer. Section 8.2 then explains the application of boundary offset to each layer, in order to best achieve target dimensions. The remaining sections discuss enhancements that take advantage of proportional deflection to improve surface quality along various profiles within each 2-D layer.

8.1 Layer Spacing

The stair-step profile caused by the stacking of discrete layers has been introduced previously in Chapter 3. This stair-step profile occurs not only in 3D Printing, but also in all layer-based solid freeform fabrication processes. Rough surface finish is most prominent on shallow inclines, and can be reduced by using thinner layers. Process planning specifies the layer spacing explicitly (via machine axis position) and affects the layer thickness indirectly (via binder concentration). This section examines the improvements in surface finish that can be attained by using thinner layer spacing.

8.1.1 Analytical Prediction of Roughness

The diagram below presents a straightforward idealization of the layer stair-step. This simplified model focuses on the stair-step as the dominant factor in determining the shape of the profile.
Details such as capillary smoothening and powder particle roughness are treated as insignificant relative to the dimensional scale of layer spacing. The parameters of interest are the layer spacing $\Delta z$ and the angle $\theta$ with respect to the horizontal.

![Figure 8.1 Rectilinear Model of Profile Roughness](image)

A common measure of roughness is $R_a$, defined as the arithmetic average of profile deviations from the mean. The following equation defines $R_a$ more precisely. The evaluation is performed along a direction $x$ for a length $L$. Deviations are measured as the absolute values of distances between the actual profile $y$ and the mean.

$$R_a = \frac{1}{L} \int_{0}^{L} |y(x)| dx$$

Equation 8.1

A rectilinear stair-step model is made to fit conveniently in this framework by aligning the mean profile with the horizontal axis, as shown below (here the x-axis and y-axis are used for analytic geometry only, and have no relation to the fast axis and slow axis of the 3D Printing machine).
The length $L$ is given by the following expression:

$$L = \frac{\Delta z}{\sin \theta}$$

Equation 8.2

Solving the expression for $R_a$ is made easier by applying the fact that an integral can be interpreted as the area under a curve. The absolute value is captured by "flipping" any portions that lie below the horizontal axis, as shown in Figure 8.3.

The area under the curve above is the sum of the four distinct triangular regions. The height of each triangle is $\Delta z \cos \theta / 2$. The expression is further simplified by the fact that two pairs of
triangles are identical, so the calculation only involves the sum of the areas of two rectangles. The equation below contains the base of one small triangle times the height plus the base of one large triangle times the height.

\[ A = \frac{\Delta z \sin \theta}{2} \cdot \frac{\Delta z \cos \theta}{2} + \frac{\Delta z \cos \theta}{2 \tan \theta} \cdot \frac{\Delta z \cos \theta}{2} \]

Equation 8.3

The \( R_a \) is the area under the curve divided by the length \( L \). Simplification and substitution leaves the final expression:

\[ R_a = \frac{\Delta z \cos \theta}{4} \]

Equation 8.4

This rectilinear model for roughness according to layer spacing has been used to predict roughness on inclined profiles as a function of layer spacing. The next figure presents the relationship for several inclines with respect to the horizontal. The plot was generated from the final equation for \( R_a \) above.
This figure highlights the fact that the $R_a$ is directly proportional to the layer spacing $\Delta z$ for a given angle. Also, at shallow inclines, the actual value for the angle does not have a strong effect (a direct consequence of the cosine relationship).

8.1.2 Experimental Results

The model for roughness versus layer spacing has been verified using printed test parts. A convenient method for collecting printed samples of surfaces at various inclines has been to use cubes rotated in 3-D space. Applying this standard test shape avoids having to re-design a large number different test parts to examine each different surface orientation. The orthogonal shape of the cubes make it very easy to mount the cubes under microscopes or profile measurement devices after removal from the powder bed. The cubes are labeled and tabulated carefully to record orientation. The following illustration shows a collection of such cubes.
Figure 8.5 Test Design for Inclined Profiles Using Rotated Cubes

The profiles below present measurement data for inclines at 5 degrees and at 10 degrees with respect to the horizontal. The measurements were made using a mechanical profilometer (Dektak 8000, Veeco Instruments, Santa Barbara, California). The scan length was 5 mm with a resolution of 1.282 microns per sample. Two different values for layer spacing were compared, 100 microns and 170 microns. The cubes were printed using aluminum oxide powder and colloidal silica binder.
Figure 8.6  Profile Data for a 10-Degree Incline at 170-Micron Layer Spacing

Figure 8.7  Profile Data for a 10-Degree Incline at 100-Micron Layer Spacing
These measurements were taken on the up-facing surfaces of the cubes. Down-facing surfaces were not measured because they are sensitive to bleeding from high binder concentration in the 100-micron case. The following tables contain the results from the profilometer scans at a 10-degree incline and at a 5-degree incline.
Table 8.1 Profilometer Results for 10° Angle to Horizontal

<table>
<thead>
<tr>
<th>Measured Ra (μm)</th>
<th>Predicted</th>
<th>Sample 1</th>
<th>Sample 2</th>
<th>Average</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Spacing 170 μm</td>
<td>41.9</td>
<td>36.8</td>
<td>41.3</td>
<td>39.0</td>
<td>4.5</td>
</tr>
<tr>
<td>Layer Spacing 100 μm</td>
<td>24.6</td>
<td>25.2</td>
<td>23.2</td>
<td>24.2</td>
<td>2.1</td>
</tr>
</tbody>
</table>

Table 8.2 Profilometer Results for 5° Angle to Horizontal

<table>
<thead>
<tr>
<th>Measured Ra (μm)</th>
<th>Predicted</th>
<th>Sample 1</th>
<th>Sample 2</th>
<th>Average</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Spacing 170 μm</td>
<td>42.3</td>
<td>46.0</td>
<td>42.8</td>
<td>44.4</td>
<td>3.2</td>
</tr>
<tr>
<td>Layer Spacing 100 μm</td>
<td>24.9</td>
<td>26.0</td>
<td>24.6</td>
<td>25.3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

The following plot superimposes the average measured values for $R_a$ upon predicted values from Equation 8.4. The cosines of 5 degrees and 10 degrees differ by only 0.01, so the predicted lines are nearly coincident. The magnitude of layer spacing clearly makes a significant difference.
Figure 8.10 Profilometer Results for Shallow Angles to Horizontal

Thinner layer spacing is clearly preferable for achieving finer surface finish. However, feature geometry (Chapter 4), saturation, the ability to spread uniform powder layers, and the production rate present practical limitations. Under most circumstances production runs are executed with the thinnest layer spacing that results in adequate production rate and acceptable bulk saturation.

8.2 Boundary Offset

Chapter 3 explained that the dimensions of a printed part include physical considerations such as primitive diameter and line width. The figure below shows that offsetting polygon boundaries
can provide compensation for such features in the 2-D plane. Without offset, external
dimensions are too large and internal dimensions are too small. An analogous situation is
encountered when performing milling operations in traditional machining [Held]. Polygons that
encircle filled regions should be offset inward, and those that define empty regions should be
offset outward.

![Diagram of Polygon Boundary Offset](image)

**Figure 8.11 Polygon Boundary Offset**

### 8.2.1 Offset Techniques

There are two basic approaches for offsetting polygons in the 2-D plane. The first approach
generates offset lines parallel to the edges of the original polygon, finds intersections, and
defines the offset polygon using the intersection points as vertices [Saeed]. The second approach
determines resultant normal vectors at each vertex, and moves the vertex along this vector such
that the proper offset distance is achieved. In most cases the offsetting procedure is not
reversible, because special adjustments need to be made for problematic conditions discussed later.

The parallel lines approach forms lines parallel to each edge of the polygon. Each line is offset from its corresponding edge by the appropriate distance. Then the intersections of the new lines can be determined, as long as they are not parallel. These intersections are the defining vertices of the offset polygon.

![Diagram showing original polygon, parallel lines, intersection points, and offset polygon.](image)

**Figure 8.12 Polygon Offset by Parallel Lines**

The geometric procedures required for offsetting with parallel lines are listed below. These operations are well documented in analytic geometry textbooks such as [Edwards]. Actual implementation in software code is not trivial because all limiting cases must be treated accordingly. Examples include lines with infinite slope and lines that are parallel (and thus have no intersection).

- Find the equation of a line given two points on the line (twice).
- Find a line parallel to a given line, at a prescribed directional distance.
- Find the point at which two lines intersect.
The translated vertices approach is illustrated next. A normal vector may be associated with each polygon edge. The vertex that joins two edges is translated along the line of the resultant of the two normal vectors from each edge. The line along which the vertex is moved bisects the angle formed at the vertex, if the offset is uniform for all edges. The vertices are moved inward for filled polygons and outward for empty polygons (holes). Recall that the filled/empty side of each edge is indicated by the order in which the endpoints are listed (Chapter 6).

![Diagram showing the translated vertices process](image)

**Figure 8.13 Polygon Offset by Translated Vertices**

The geometric procedures for determining offset polygons using the translated vertex approach can be outlined as follows:

- Find the equation of a line given two points on the line (twice).
- Find the slope of a line that bisects the angle between two other lines.
- Find the equation of a line given the slope and one point (the apex).
- Find a point on a line a given directional distance from another point on the line.
The working offset program in 3D Printing was implemented using the parallel lines approach because the number of computations were fewer and the data structures were more immediately compatible with existing software modules.

### 8.2.2 Problematic Cases

There are several problematic situations encountered when offsetting boundaries. These problems are common to most types of planar boundaries including polynomial curves, spline curves, and vertex-to-vertex polygons. Examples of problems for planar curves are cited in [Maekawa], [Pham], and [Tiller]. This section describes cases that have been encountered when offsetting polygons as represented in the SLC files of 3D Printing. Solution ideas are discussed here, but not all have been implemented in software code.

- Exaggerated Apex
- Self-Intersection
- Mutual Intersection

The apex between two edges of an offset polygon may disproportionately exaggerate the dimensions of the original polygon if the angle is very acute. The following diagram illustrates a condition in which this exaggeration occurs.
Figure 8.14 Exaggerated Apex Caused by Polygon Boundary Offset

The offset distance \( d \) runs perpendicular to the edges of the original polygon. The offset edges are by definition parallel to the corresponding original edges. These characteristics result in a condition in which a line along segment \( d' \) always bisects the angle \( \theta \). Therefore, the length of segment \( d' \) can be expressed conveniently in terms of the angle \( \theta \) and the offset distance \( d \).

\[
d' = \frac{d}{\sin(\theta / 2)}
\]

Equation 8.5

In the context of polygon edges the interior angle can always be restricted to a range between 0° and 180°. The discrepancy between the corner offset \( d' \) and the standard offset \( d \) is near zero when the angle is near 180°, but most severe as the angle approaches 0°. [Barton] presents a polygon representation that allows the corners to be closed with circular arcs. The polygonal slice format used in 3D Printing does not support circular arcs, but can apply a similar approach using an additional line segment to "cap" the apex. However, such a feature is not currently implemented. Typical offset magnitudes are only about 100 microns, so the exaggeration is not severe with respect to the scale of complete parts.
Another concern is that offsetting a polygon may cause its edges to intersect, even if the original polygon had no such problems. Figure 8.15 illustrates two examples of self-intersection. Offset polygons are shown with dashed lines. The problem to the right of the figure is caused by the fact that the four edges of the offset polygon were generated based on four original edges, but one was exceptionally short with respect to the offset distance. It would occur for both the parallel lines approach and the translated vertex approach to generating offsets.

![Diagram](image)

**Figure 8.15 Self-Intersection Caused by Polygon Boundary Offset**

Self-intersections result in malformed polygons and may be corrected by post-processing of the new data points. Checks and corrections are discussed briefly in [Barton], and oftentimes require deletion of crossover features.

One more concern for offsets involves multiple polygons. In some cases the offsets of two polygons may be large enough that the new polygons intersect. The next figure shows a simple example in which the offsets of two non-intersecting polygons do overlap.
The rastering program used in 3D Printing (Chapter 6) has been designed to automatically interpret overlapping polygons, so this problem does not present an serious obstacle. However, ideally the improper polygon arrangements should be corrected by performing union operations to the polygons before rastering.

8.2.3 Experimental Quantification

Test parts were printed to determine the amount of offset required for highest dimensional accuracy. As stated in Chapter 3, the offset makes adjustments for line width and primitive diameter. The figure below shows the basic design, which has tiers so that any deviation from the designed dimensions can be divided into an additive component and an overall scaling factor.
This common test design can be used to characterize dimensions along any of the three build axes. For example, the same design can be printed in a vertical orientation to examine dimensional control in the vertical direction. The positions of layers, lines, and printhead transitions are specified exactly, because the data file is prepared using the Direct File Construction method (see appendix).

The appropriate offset for a given material system is calibrated by comparing measured dimensions with corresponding reference positions in the data file (refer to Chapter 3). Dimensions along the slow axis are referenced by the spacing between centers of the outermost lines. Dimensions along the fast axis are referenced by the spacing between the center positions of the first and last trigger positions of droplets along the fast axis. Digital calipers with a resolution of 10 microns are typically used for the measurements.

Boundary offset is applied in cases where dimensional accuracy is highly critical. An example application is the fabrication of metal tooling for injection molding. To achieve the best accuracy, the characterization experiment is conducted at parameter values that exactly match the settings to be used in actual production runs. This includes both material conditions and machine settings.
Table 8.3 lists the experimental conditions and the next two tables present the measured results.

Eight replicates were printed, and each part had four tiers.
### Table 8.3 Experimental Conditions for Boundary Offset

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Spacing</td>
<td>170 µm</td>
</tr>
<tr>
<td>Line Spacing</td>
<td>184 µm</td>
</tr>
<tr>
<td>Binder Material</td>
<td>Acrysol 25%</td>
</tr>
<tr>
<td>Powder Material</td>
<td>Stainless Steel 420, 230-325 mesh</td>
</tr>
<tr>
<td>Powder Solid Fraction</td>
<td>60% estimated</td>
</tr>
<tr>
<td>Binder Flow Rate</td>
<td>1.0 ml/min</td>
</tr>
<tr>
<td>Droplet Frequency</td>
<td>44 kHz</td>
</tr>
<tr>
<td>Printhead Speed</td>
<td>1.5 m/s</td>
</tr>
<tr>
<td>Data File</td>
<td>19960205.3dp</td>
</tr>
</tbody>
</table>

### Table 8.4 Boundary Offset Measurements Along Fast Axis

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10000</td>
<td>10179</td>
<td>179</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>20000</td>
<td>20158</td>
<td>158</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>30000</td>
<td>30158</td>
<td>158</td>
<td>49</td>
</tr>
<tr>
<td>4</td>
<td>40000</td>
<td>40149</td>
<td>149</td>
<td>43</td>
</tr>
</tbody>
</table>

### Table 8.5 Boundary Offset Measurements Along Slow Axis

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6440</td>
<td>6755</td>
<td>315</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>13064</td>
<td>13356</td>
<td>292</td>
<td>37</td>
</tr>
<tr>
<td>3</td>
<td>19688</td>
<td>19943</td>
<td>255</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>26312</td>
<td>26581</td>
<td>269</td>
<td>19</td>
</tr>
</tbody>
</table>

Parts printed in this material system for tooling applications undergo a heat treatment, to burn out binder and to achieve higher density (by powder sintering). However, the parts are measured in their green (unfired) state to isolate the additive component of the difference between reference positions and measured dimensions. The boundary offset attempts to compensate for this additive component only. The next plot shows the measured dimensions against the reference dimensions from the data file.
Performing a least-squares linear fit to the above data identifies the additive components along each axis. From the above data the additive component along the fast axis is 183 microns and the value along the slow axis is 326 microns. This corresponds to offset magnitudes of 91.5 microns and 163 microns, respectively.

To confirm that there is no overall shrinkage in the green state, all data points were plotted to look for trends. The next two figures show the differences between measured dimensions and
reference positions. The plots show no clear trends that depend on the length of the original reference dimension. This suggests that there is no appreciable shrinkage in the green state.

Figure 8.19 Examination of Possible Green Shrinkage Along Fast Axis Tiers

Figure 8.20 Examination of Possible Green Shrinkage Along Slow Axis Tiers
The method above has offered a means of quantifying the magnitude of offset to be applied to polygons before printing. This would result in more accurate printed dimensions with respect to the original CAD model. However, the data shows that the amount of offset may not be uniform with respect to the fast and slow axes. This motivates the implementation of non-uniform offset, which is discussed next.

8.2.4 Non-Uniform Offset

An offset is uniform if it can be represented by running a circular outline along the boundary of a curve (or polygon), as was done in Figure 8.11. The locus of points drawn by the outer boundary of the circle forms the offset curve. This is a common situation encountered in manufacturing and two-dimensional graphics. However, the experimental results in the previous section show that the offset in 3D Printing may in some cases be non-uniform. Depending on conditions such as droplet spacing and line width, the necessary compensation along the fast axis may be different from that along the slow axis. A special non-uniform offsetting technique has been developed to address this issue.

In pocket machining the cutting tool spins much faster than it translates, so the offset is commonly modeled by running a circle around the desired pocket boundary. In 3D Printing the uniform circular shape is inadequate to capture the difference between axes, so an elliptical model is used. The following diagram illustrates the difference.
Figure 8.21 Elliptical Model for Application of Non-Uniform Offset

An ellipse in the x-y Cartesian coordinate system is a the locus of points that fits the quadratic expression (where constants $a$ and $b$ are non-zero):

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

Equation 8.6

Figure 8.22 Analytic Geometry of an Ellipse

The distance $d$ between the origin and any point on the boundary is a function of the angle from the x-axis:
\[ x = d \cos \theta \]  \hspace{1cm} \text{Equation 8.7}

\[ y = d \sin \theta \]  \hspace{1cm} \text{Equation 8.8}

Substituting for \( x \) and \( y \) in Equation 8.6 gives an expression for \( d \) in terms of the angle \( \theta \) and the extrema \( a \) and \( b \) along the horizontal and vertical axes.

\[ d = \frac{a^2 b^2}{\sqrt{(b \cos \theta)^2 + (a \sin \theta)^2}} \]  \hspace{1cm} \text{Equation 8.9}

### 8.3 Sub-Raster Feature Snapping

Droplets may be placed more accurately by using proportional deflection to snap line positions onto polygon boundary features. In this context the term "feature" refers to macro-scale dimensions larger than about 1 mm. The term "sub-raster" specifies that the snapping occurs within the space between any two consecutive raster lines.

The standard encoding process as introduced in Chapter 6 uses a uniform line spacing, which may not be adequate for capturing boundaries precisely because the grid resolution is coarse (typically about 200 microns). Proportional deflection allows line positions to be adjusted according to the finer resolution available in the raster file (as low as 10 microns). This section focuses on adjusting the positions of line segments. Section 8.3 brings the principle to a finer scale, to improved inclined boundaries.
The next figure illustrates the concept of feature snapping, in which the line positions closest to the polygon boundary are modified to improve accuracy. In this example the step feature at the upper portion of the target geometry is smaller than the width of a standard printing lane. A lane is the space for which one jet is responsible during a printhead pass. The default procedure for filling a lane would be to position the line at the center of the lane (shown shaded in the figure). However, this is inadequate for capturing the step feature. The software program that designates line positions must perform additional scan operations within the lane to correctly follow the boundary.

![Diagram](image)

**Figure 8.23 Sub-Raster Feature Snapping**

The diagram reveals that the concentration of binder may not be uniform when feature snapping is implemented with the outer-most line only. The saturation experiment in Chapter 4 has shown that locally high binder concentration does not severely affect the overall dimensions. On the low side, however, there may be a deprivation zone as wide as half a line width. This may result in a weaker "shell", because of the potential gap underneath. Therefore, wider internal fill patterns should be used on lines just inside the boundary. Chapter 6 introduced the distinction between boundary segments and interior segments.
Feature snapping has been implemented for the printing of metal tooling cavities and has markedly improved the ability to meet target dimensions. Dimensional errors without feature snapping may be as large as the line spacing (half on either side of the measurement). Table 8.5 shows some case examples of the improved accuracy along the slow axis. The parts were printed in stainless steel with 202-micron line spacing, and measured in four places. As shown in the left of Figure 8.22, the absence of line snapping may cause a discrepancy that is either larger or smaller than the target. The worst-case scenario occurs when the printed part is 202 microns farther from the target.

<table>
<thead>
<tr>
<th>Target (μm)</th>
<th>Measured (μm)</th>
<th>Std Dev (μm)</th>
<th>Difference (μm)</th>
<th>Error (%)</th>
<th>Reference</th>
<th>Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td>49167</td>
<td>49060</td>
<td>10</td>
<td>107</td>
<td>0.22</td>
<td>AMP 5/7/96</td>
<td>CMM</td>
</tr>
<tr>
<td>129286</td>
<td>129030</td>
<td>30</td>
<td>256</td>
<td>0.20</td>
<td>3M 5/20/96</td>
<td>Calipers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target (μm)</th>
<th>Worst-Case (μm)</th>
<th>Std Dev (μm)</th>
<th>Difference (μm)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>49167</td>
<td>48858</td>
<td>n/a</td>
<td>309</td>
<td>0.63</td>
</tr>
<tr>
<td>129286</td>
<td>128828</td>
<td>n/a</td>
<td>458</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Feature snapping not only improves dimensional accuracy, but also enhances surface finish. The surfaces that benefit most directly are those on planes parallel to the fast axis. The side-view figure below shows how feature snapping reduces surface roughness along certain profiles. The vantage point is such that the printed lines are perpendicular to the page. Without feature snapping, the lines are constrained to whatever raster grid was initiated for each layer.
Figure 8.24 Surface Finish Improvement by Sub-Raster Feature Snapping

Line justification is a more elegant way to perform feature snapping for a limited class of shapes. The principle of justification is borrowed from desktop publishing, in which the spacing between characters (or lines, for 3DP) is adjusted to generate more uniform spacing between the first and last characters. The next diagram illustrates the application of line justification for a simple rectangular polygon (top view). Instead of using a pre-determined line spacing for the entire production run, justification chooses the spacing based on local geometry.

Figure 8.25 Line Justification
Line justification has not been implemented for printing 3DP layers. Justification is computationally expensive, because no one raster grid is applicable to all parts. Even if the slow axis were moved in regular increments, proportional deflection would have to be utilized extensively (every printed line segment). Another fundamental problem is that complex shapes may need to be treated in many subdivisions, in order to achieve proper justification between horizontal boundaries on a local basis, as shown here.

![Diagram of justification](image)

**Figure 8.26 Line Justification by Sub-Division of Complex Shapes**

The parts printed by adjusting exterior lines only have shown no adverse effects from the local non-uniformity of binder. The computational expense is low compared to the justification approach because only boundary segments require special attention. This approach has demonstrated measurable benefit (Table 8.6), making it the method of choice for sub-raster feature snapping.

### 8.4 Incline Tracing

One of the most prevalent sources of surface roughness comes from the aliasing effect between lines within each layer (Chapter 3). Shallow inclines accent the discrete step changes from line
to line when using a simple on/off printing style. The on/off printing style is described as binary deflection. The aliasing problem can be reduced dramatically by tracing inclines with proportional deflection. The figure below illustrates the principle of incline tracing. In the figure only one line is shown deflected for clarity, but in the actual implementation all lines apply tracing as necessary.

![Figure 8.27 Incline Tracing by Proportional Deflection](image_url)

The following image compares incline tracing with proportional deflection against default printing with binary deflection. The test design has a shallow incline of 5 degrees from the horizontal. These parts were printed using colloidal silica binder in aluminum oxide powder, and the line spacing in both cases is 170 microns. The image is from a top-view vantage of the powder bed and the fast axis runs horizontally. The part in the upper portion of the photo was printed with simple toggle instructions and the ragged profile is clearly visible. The lower part was printed with incline tracing and has a smoother profile.
The photograph below shows the application of incline tracing on a complete part. The rocker arm on the left was printed using binary deflection, but the one on the right benefited from proportional deflection.
Proportional deflection enables each jet to cover a region wider than a mere straight line during each pass of the printhead. The region for which a jet is principally responsible during any particular pass is called a lane. Each layer is divided into several such lanes, as shown below. The lane width is the distance between the upper and lower bounds of the lane, and is equal in magnitude to the line spacing.

![Figure 8.30 Printing Lane](image)

The basic concept of incline tracing is very straightforward. The scan resolution $dy$ used in rastering (Chapter 6) is significantly finer than the lane width (or line spacing) used during encoding. The line spacing $\Delta y$ may be constrained by several parameters and criteria (Chapter 7), but the scan resolution may be arbitrarily small (limited only by computation power and disk storage space). Therefore, the intersection of each lane with a polygon boundary may contain several intermediate data points. The printhead transitions are encoded to trace these data points, thereby smoothening the profile.

Incline tracing is a technique that has been implemented in standard file processing. The deflection resolution $\Delta w$ is a variable parameter that effectively controls the spacing of features on the surface. This deflection axis is parallel to the slow axis at all times (Chapter 5). The number of sub-levels within a printing lane is the lane width divided by the deflection resolution.
In effect it is the deflection resolution and not the line spacing that dictates printing resolution along the slow axis.

### 8.4.1 Methodology

This section discusses some of the details involved in converting digitized geometric information into printhead transitions. Background knowledge of the file processing steps (Chapter 6) is essential for understanding the discussion in this section. In some cases the decision logic is expressed in conditional rule syntax (IF ..., THEN ...). This syntax as presented here is not intended to be a rigorous symbolic language (compared to the C programming implementation). However, the rules help to concisely bridge the gap between decision principles and logical expression. The tracing algorithms have been made progressively more complete and robust to so-called "software glitches" by re-examining rule logic and augmenting or correcting when necessary.

The encoding process (Chapter 6) divides each layer into lanes, based on the jet spacing and desired line spacing. A *trace* refers to the series of available data points along a polygon boundary that lie within a printing lane. The following illustration shows a scan lane intersecting a polygon edge and the corresponding trace points.

```
\begin{figure}[h]
\begin{center}
\includegraphics[width=\textwidth]{trace_points.png}
\end{center}
\caption{Trace Points}
\end{figure}
```

179
The trace points are read from an RST file (Chapter 6), which has already stored discrete points as the start and stop positions of raster segments. Typically the resolution in the raster file (10 to 20 microns) is significantly finer than the lane width (about 200 microns), so several trace points are available. To interpret state changes, the traces are categorized into two fundamental types, either entry or exit. The next figure distinguishes an entry trace from an exit trace. By convention the trace type is interpreted from left to right, in the direction of the positive x-axis. Each trace point is derived either from the start position or the stop position of a raster segment in the RST file. These are distinguished by different colors in the diagram. For every entry trace point, there is a corresponding exit trace point.

![Diagram of entry and exit traces](image)

**Figure 8.32 Entry and Exit Traces**

Encoding printhead transitions is accomplished by interpreting the trace points on a lane by lane basis. Some look-ahead procedures may necessary to manage state changes, as explained below. The first trace point encountered will always represent an entry transition for left to right printhead motion (the interpretation is reversed when printing occurs from right to left). The design of the RST file guarantees that raster segments are properly bounded by paired on/off endpoints. Deflection positions are encoded into the machine data file (3DP) as long as the next
data point is also an entry transition. This concept is illustrated below and can be expressed in rule syntax as follows.

\[
\text{IF the current data point in a lane is an entry trace point,}
\AND \text{the next trace point along the fast axis is also an entry trace point,}
\THEN \text{encode a deflection at the y-axis position of the current trace point.}
\]

![Diagram showing continuity among trace points](image)

**Figure 8.33 Continuity Among Trace Points of the Same Type**

The encoding of deflection instructions continues until the next trace point represents an exit point. Then deflection needs to directed to a neutral position between the two traces. The material printed in this neutral position is called a *bridge*. A bridge may be a simple straight-line segment as shown in the figure, or it may contain more complex deflection sequences for internal fill patterns (Chapter 5). In either case, the bridge transition is simply a deflection instruction like any other (Chapter 6). The bridge ends before encountering the first exit trace point. The figure below shows an example of a bridge between an entry trace and an exit trace, and the following statement presents the logic. The "sufficiently long" requirement is used as a filter for physical constraints such as the shortest line segment or gap that can be printed. The issue of "spanning" is deferred until later in this section.
IF the current data point in a lane is an entry trace point, 
AND the next data point along the fast axis is an exit trace point, 
AND the distance between the two points is sufficiently long, 
AND both traces span the full lane width, 
THEN a bridge should be placed between the these two points.

Figure 8.34 Insertion of a Bridge Between Entry and Exit Traces

The continuation of exit traces is very similar to the continuation of entry traces: if the current data point an exit trace and the next one is an exit point also, then write a transition at the current deflection position. A catch (off) instruction is written to the data file after the last exit point.

The final exit point is known to be "last" because it is either the last point within the lane, or because the next point is an entry trace point.

IF a data point is the last one along the fast axis for this lane, 
THEN encode a catch (off) transition at this point.

IF the current data point in a lane is an exit trace point, 
AND the next data point along the fast axis is an entry trace point, 
AND the distance between the two points is sufficiently long, 
AND both traces span the full lane width, 
THEN encode a catch (off) transition at the current data point.
An important exception case to consider is the occurrence of traces that are perpendicular to the fast axis. A simple solution is to collapse all the points into a single transition at the center of the lane. A more sophisticated approach is presented later in this chapter.

Figure 8.36 Collapsing a Vertical Trace into a Single Transition
The preceding diagrams and rules summarize the most common scenarios in the implementation of incline tracing. However, the concept of "spanning" was included often. A collection of trace points span the printing lane if the points extend from the lower limit to the upper limit (or vice versa). The next figure contrasts a spanning trace from a non-spanning trace. Non-spanning traces occur at the lateral boundaries of polygon data.

![Diagram of spanning and non-spanning traces](image)

**Figure 8.37 Spanning Traces vs. Non-Spanning Traces**

Non-spanning traces require special consideration. For example, fill patterns are not inserted between entry and exit traces that do not completely span the lane width. Also, the bridge connecting the entry and exit traces does not get deflected back to center (as in Figure 8.34). Another problem is caused by short vertical trace points, even if the trace does span the lane. These must be collapsed because of droplet availability, but may not be "averaged", as was done in Figure 8.36. In such a case each point must be evaluated in relation to the rest of the trace, in order to determine where to place the step change.
Incline tracing was initially driven by the need to smoothen broad-scale surfaces. A key challenge remaining for the encoding software is proper handling of geometric features that have non-unique \( y \) values for each \( x \) value inside a single lane (vertical lines are represent a subset that is manageable by exception). The rules presented thus far are not comprehensive enough to handle cases as shown below. Surface features on this small a scale (less than a lane width) were not encountered frequently, leaving them with lower priority. However, applications that include designed surface textures [Curodeau, Jee] or embossing may soon require such cases to be handled, especially for printing with wider lane widths.

Figure 8.39 Complex Geometric Features Within a Printing Lane
The principle of incline tracing is not restricted to interpreting raster lines, although this is the approach taken for the current implementation software. Incline tracing can be processed directly from a polygon (SLC) file. Bypassing the raster (RST) file reduces the number of operations and results in much faster execution. This direct approach is completely viable, and in fact the rocker arm demonstration parts in Figure 8.29 were processed directly from polygon files without rastering. The computations involve following the pieces of polygon edges that lie within a printing lane. Data points can be taken at increments along the slow axis (as in standard practice), fast axis, or even along the polygon boundary. The figure below distinguishes these increment strategies.

![Increment Strategies for Incline Tracing](image)

**Figure 8.40 Increment Strategies for Incline Tracing**

Each approach has strengths. Incrementing along the slow axis is conceptually simple and matches data organized in raster form. Incrementing along the fast axis can be used to address the physical consideration of droplet availability along this axis, but may lead to very large data files for shallow angles. Following the polygon boundary is perhaps the best way to capture small undulations that occur inside individual print lanes (solving the internal ripple problem).
However, pathological polygon arrangements as was shown in Chapter 6 were encountered frequently, and the raster-based approach (increment along y) was chosen for its robustness.

8.4.2 Experimental Verification

An experiment has been conducted to quantitatively assess the extent to which incline tracing improves surface finish. The test parts were printed as rotated cubes, much like the samples used to examine layer-to-layer profiles (Figure 8.5). The material system was 28-micron aluminum oxide powder bound by colloidal silica binder. The line spacing (lane width) was 170 microns, but the deflection resolution was set to 20 microns. Control samples with the same line spacing but no incline tracing were also printed for comparison. Shallow angles at 10 degrees and 5 degrees off the fast axis were examined. The next set of figures show mechanical profilometer traces along the various samples.
Figure 8.41 Profilometry Trace Along 10° Raster Incline with Binary Deflection

Figure 8.42 Profilometry Trace Along 10° Raster Incline with Proportional Deflection
The following table provides quantitative results for $R_a$, computed numerically from the profilometry data. Printing was conducted uni-directionally. Leading-edges are distinguished from trailing-edges because of the potential leading-edge problems (Chapter 5). The trailing-edge profiles are considered more indicative of a well-controlled implementation. Measurements were also taken parallel to the fast axis, to establish a "best case" reference.
Table 8.7 Surface Finish Improvement by Incline Tracing

<table>
<thead>
<tr>
<th>Printing Mode</th>
<th>Angle</th>
<th>Edge</th>
<th># Samples</th>
<th>Avg Ra (μm)</th>
<th>Std Dev (μm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>0°</td>
<td>Side</td>
<td>8</td>
<td>14.5</td>
<td>2.2</td>
</tr>
<tr>
<td>Toggle Printing</td>
<td>5°</td>
<td>Leading</td>
<td>8</td>
<td>38.7</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40.5</td>
<td>3.8</td>
</tr>
<tr>
<td></td>
<td>10°</td>
<td>Leading</td>
<td>8</td>
<td>38.9</td>
<td>2.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40.4</td>
<td>2.8</td>
</tr>
<tr>
<td></td>
<td>45°</td>
<td>Leading</td>
<td>8</td>
<td>31.2</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>35.2</td>
<td>5.0</td>
</tr>
<tr>
<td>Incline Tracing</td>
<td>5°</td>
<td>Leading</td>
<td>8</td>
<td>21.9</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15.5</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>10°</td>
<td>Leading</td>
<td>8</td>
<td>20.4</td>
<td>3.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18.3</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>45°</td>
<td>Leading</td>
<td>8</td>
<td>24.4</td>
<td>7.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>23.9</td>
<td>3.4</td>
</tr>
</tbody>
</table>

Incline tracing by proportional deflection clearly has an advantage over toggle printing by binary deflection. For these conditions the roughness was reduced by about a factor of two. An interesting point to note is that the leading edges are smoother than trailing edges during toggle printing, but trailing edges are smoother with incline tracing. This may be caused by the fact that droplet merging can "accidentally" smoothen profiles because of the locally high binder content. When the droplets are printed by proportional deflection, the more precise positioning control dominates.

The following plot shows the results graphically. The experimental results are superimposed over a prediction from an ideal geometric model, as in Figure 8.1. Binary deflection is treated as a 170-micron step (the line spacing), while proportional deflection is entered as 20-microns (the deflection step).
The toggle printing with 170-micron line spacing does follow the rectilinear model, but the incline tracing case does not. Error bars are shown at ± 2.2 microns, which is the standard deviation of the reference case. The discrepancy may be related to the proximity with the physical limit of smoothness. The idealized geometric model is not valid below about 15 microns, at which particle roughness becomes relevant. For the incline traced data, the roughness may become worse at less shallow angles because of a sub-optimal "over control" condition. Some profile anomalies from droplet control on non-shallow angles is evidenced in the photographs of the next section.
Regardless of the closeness-of-fit to analytical models, incline tracing does provide a substantial improvement in surface finish. This is clearly evidenced in the comparative values for profile $R_a$, as well as the composite result demonstrated in Figure 8.29.

8.5 Diffuse Line Termination

Sub-raster feature snapping (Section 8.3) adjusts line positions to properly follow geometric boundaries parallel to the fast axis. Incline tracing (Section 8.4) has proven to be a very valuable tool for improving the surface finish along shallow angles off the fast axis. The final profiles to consider for a complete picture are those perpendicular to the fast axis. Proportional deflection encounters a fundamental limitation on this profile because of droplet availability in each pass (Chapter 5). However, the profiles can still be smoothened by diffusing the ends of lines to close gaps and eliminate end effects. The following figure explains the basic principle. The technique is implemented only trailing edges of print lines because the leading edges are extremely difficult to control (Chapter 5).

![Diagram](image)

**Figure 8.46 Diffuse Line Termination**
Diffuse line termination becomes more valuable as the line spacing $\Delta y$ gets wider. Wider line spacing may be desired to manipulate the bulk saturation. However, wider line spacing also accentuates the gaps between consecutive lines. The figure above shows diffusion of the trailing edge into only two sub-levels, but more are certainly possible. Droplet availability (Chapter 5) presents the only fundamental restriction that there can only be one deflection position per position along the fast axis.

The deflection levels that would result in a uniform spacing between the centers of droplets along the perimeter are governed by the equation below. Each line of the deflection levels $j$ (from 1 to the number of levels $n$) is positioned away from the centerline $CL$ position of a line $i$. These positions depend only on the line spacing $\Delta y$. The derivation is listed as an appendix.

$$(y_j, y_{i_{CL}}) = \left[ \left( \frac{n-1}{2} + j \right) \frac{1}{n} - 1 \right] \cdot \Delta y \quad j \in \{1..n\} \quad \text{Equation 8.10}$$

For example, if there are $n = 2$ droplets used in the diffuse termination the first should be placed at $-\Delta y/4$ and the second at $+\Delta y/4$. If $n = 3$ the droplets are placed off the centerline by $-\Delta y/3$, 0, and $+\Delta y/3$. If the time independence hypothesis were entirely true, this geometric model would be appropriate. However, actual printing behavior may be affected by capillary forces and temporal effects and deserves experimental study. Still the geometric model provides a sound starting point.

A preliminary set of experimental test parts have been printed to examine the benefit of diffuse line termination. The technique has not yet been implemented in standard processing software,
but was made possible by Direct File Construction (see appendix). The figure below shows the test part design for printing single-layer samples.

![Test Part Design for Single-Layer Profiles](image)

**Figure 8.47 Test Part Design for Single-Layer Profiles**

The next figure shows an optically magnified photograph in which the individual lines are distinctly visible on the exit profile. The photo is taken from a top-view vantage point and lines are printed uni-directionally from left to right. The test parts were printed with integral frames for structural support and orientation. The fast axis speed was selected according to the droplet frequency such that it placed one droplet every 20 microns along the fast axis. A set of parts was also printed with 10-micron droplet spacing, but the binder content was too high to distinguish separate entities. Table 8.8 lists the experimental conditions.
Table 8.8 Experimental Conditions for Diffuse Line Termination

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Spacing</td>
<td>170 μm</td>
</tr>
<tr>
<td>Binder Material</td>
<td>Colloidal Silica 30%</td>
</tr>
<tr>
<td>Powder Material</td>
<td>Aluminum Oxide 28 μm</td>
</tr>
<tr>
<td>Powder Solid Fraction</td>
<td>35% estimated</td>
</tr>
<tr>
<td>Binder Flow Rate</td>
<td>0.06 ml/min</td>
</tr>
<tr>
<td>Droplet Frequency</td>
<td>46 kHz</td>
</tr>
<tr>
<td>Printhead Speed</td>
<td>0.92 m/s</td>
</tr>
<tr>
<td>Data File</td>
<td>19960416.3dp</td>
</tr>
</tbody>
</table>

Figure 8.48 Standard Line Termination 240 μm Line Spacing

This experiment used the last two droplets to attempt smoothening of the exit profile.

\[
(y_i) - (y_i)_{CL} = \left[ \left( \frac{n-1}{2} + j \right) \frac{1}{n} - 1 \right] \cdot \Delta y \quad j \in \{1..n\}
\]

Equation 8.10 suggests that the last two droplets of any given line should be offset by one quarter of the line spacing from the center position for geometric uniformity. This is equal to ± 60 microns for the 240-micron line spacing.
The following figure shows some evidence of the adjusted droplet positions, but the exit profile still shows roughness.

![Image of diffuse line termination with 240 µm line spacing and ±60 µm dither](image)

**Figure 8.49** Diffuse Line Termination, 240 µm Line Spacing with ±60 µm Dither

Apparently there is some temporal behavior in effect, in which the droplets preferentially merge with their "home" lines and do not completely stitch across gaps. The optimal condition is pursued by increasing the amount of deflection to better blend the exit profile. The next figure shows noticeably better results when using ±100 microns of deflection.
This preliminary investigation has proven that diffuse line termination is a viable method for smoothening perpendicular exit profiles on a layer-by-layer basis. Future development would implement the technique on complete parts to smoothen entire 3-D faces instead of just 2-D profiles.
9. Future Work

9.1 Three-Dimensional Extensions

The enhancements described in the last chapter have primarily focused on enhancements within each 2-D layer (except for using thinner layer spacing). Improvement efforts must address complete 3-D geometry to make the next major advances in surface quality, both in terms of dimensional accuracy and surface finish. Some ideas which have been considered are discussed below.

9.1.1 Incline-Specific Surface Offset

Chapter 8 presented a discussion of boundary offset for polygons in the two-dimensional plane. These offsets improve dimensional accuracy along the fast and slow axes, on a layer by layer basis. However, the following figure shows that this simple approach may not be optimal for following slopes that vary across layers, depending on the feature geometry. The diagrams use a sphere to represent an idealized printed feature, which may be a single-droplet primitive or a line in cross-section. Note that the slicing procedure gives an intersection where the layer position meets the CAD profile, but the point of control is the top center of the printed feature.
The best possible offset is one that causes the external surfaces of printed features to lie on the boundary of the original CAD model. It is important to recall that the position printed feature is constrained by the top surface of the powder layer and can not be adjusted in the vertical direction. The magnitude depends on the degree to which the intended boundary is inclined, as shown below.

The proper offset also depends on whether the surface is up-facing or down-facing, as demonstrated in the next figure. As the incline becomes more shallow, an interesting observation
is that the target offset decreases for up-facing surfaces, but the desired offset increases for down-facing surfaces.

![Diagram of up-facing and down-facing surfaces]

**Figure 9.3 Offset Requirements for Up-Facing and Down-Facing Surfaces**

The amount of offset may be specified in analytical form, assuming an idealized circular cross-section. The magnitude depends on the surface incline and the size of the feature, as well as the side (up-facing or down-facing) on which the exposed surface lies. Figure 9.4 marks key dimensions, and Equation 9.1 presents the analytical relationship for up-facing surfaces. Figure 9.5 and Equation 9.2 address down-facing surfaces. In each case, the angle $\theta$ is only defined for values less than 90 degrees. The circle is constrained to be tangent to both the CAD boundary and to the slice surface.
Figure 9.4 Analytic Geometry of Variable Boundary Offset for Up-Facing Surfaces

\[ d = R \tan \left( \frac{\theta}{2} \right) \]

Equation 9.1

Figure 9.5 Analytic Geometry of Variable Boundary Offset for Down Facing Surfaces

\[ d = \frac{R \left( 1 + \frac{1}{\cos \theta} \right)}{\tan \theta} \]

Equation 9.2

The next figure plots target offsets \( d \) as a function of incline \( \theta \) to the horizontal plane. The radius \( R \) of the printed feature is set arbitrarily to 100 microns. Clearly the two types of faces require
very different offset values for shallow inclines. For more vertical inclines the difference between up-facing and down-facing is less important, and the radius offset is adequate.

![Graph showing offset vs. incline](image)

**Figure 9.6 Target Offset Values for Boundary Offset vs. Incline to Horizontal**

The variable offset based on 3-D incline has not been implemented because the current 2-D polygon file format does not retain incline information for each edge. However, should the enhancement later become required, such orientation information is readily available from the directional normal vectors contained in the 3-D STL files. The slicing program would associate an angle with each edge of the resulting polygons.

### 9.1.2 Sub-Layer Scanning

The enhancement techniques in the 2-D plane make it not only possible, but also practical to apply drop placement information with fine resolution (about 10 microns) along both the fast and slow axes. The quality of the printed results are then as high as the machine performance and
process physics will allow. However, the richness of information along the vertical axis is far more coarse (typically over 150 microns). Therefore, high dimensional accuracy and fine feature resolution are difficult to achieve.

The benefit of simply printing with thinner layers was demonstrated in Chapter 8. Thinner layers implies a finer slicing grid, which enhances geometric information and improves surface finish. However, the spacing of layers is constrained by many factors including the ability to spread powder in thin distributions, the way in which binder permeates through the powder, and the ability to manage proper binder saturation levels. For a limited class of parts (e.g. ones that can tolerate a wide range of saturation), printing with thinner layers may be very useful. Furthermore, it may be possible to divide some parts into vertical regions, printing some at finer layer spacing than others [Suh].

This section presents an alternative strategy to improving vertical resolution. A key fact is that the slicing resolution does not have to match the printed layer resolution. It may be advantageous to examine more detailed geometric information during the slicing procedures, even if the layer spacing is much more coarse. The discussion presents a case with one sub-layer scan, but more sub-levels are certainly possible. The examples are further simplified by using rectangular block objects; freeform curved surfaces require more detailed examination.

First, the basic problem scenario is explained. The figure below shows a simplified cross-section of a CAD model for a rectangular block. Depending on how where the block lies in space, the layer slicing grid may intersect the block in different places. The block is not initialized to a reference height, and the grid is not initialized to the block itself. The relative alignment of the slicing grid and the block is considered to be random, because the block represents only one
piece of a larger production run, and the grid may have been initialized based on some other geometric object or feature in the build volume. The concept for adapting the slicing grid to the geometric features in the build volume is discussed in [Dolenc].

![Slicing Grid Diagram](image)

**Figure 9.7 Variations of Slicing Grid Alignment with Vertical Columns**

The figure below shows four separate CAD dimensions (solid outlines) which may differ by only a few microns. However, the particular slice planes (dashed lines) that intersect the columns differ significantly. The printing is dictated by the information in the slice intersections, so the printed results (shaded) may show disproportionate differences among the four cases below. The dimensional error of the printed parts may be as high as a full layer thickness, typically about 200 microns.
Figure 9.8 Variations of Printed Vertical Columns Based on Slicing Grid Alignment

The basic principle of sub-layer scanning is that slicing is performed at intermediate positions between those that correspond to actual printed layers. The intermediate scans perform a search for relevant geometric information between layers. Then the information associated with any particular layer may be adjusted, depending on features in the immediate proximity. The next figure illustrates how additional slice information may offer a more accurate way to realize the height and position of the intended CAD outline. The physical resolution is still fairly coarse, but the finer scanning resolution improves net accuracy.
Information from the intermediate slices may be used to decide whether or not to print a particular region in a given layer. Sub-layer scanning does not provide a perfect solution because the layers are still constrained by the surface positions of powder layers. The previous figure shows that even the adjusted printing does not match the CAD boundaries precisely.

Nevertheless, local dimensional discrepancies can be reduced to half a layer thickness, just by using the simple approach described above. The actual implementation of this principle faces challenges related to connectivity in the CAD model; however, relationships among surface facets can certainly be accomplished [Wozny].

9.1.3 Gap-Fill Layers

The standard method of printing treats all layers equally in terms of the completeness to which each is covered. Efforts have made to have the bulk saturation in every layer be the same. Managing a proper level of bulk saturation becomes very difficult as the layer spacing is reduced,
however, because some droplets must be sensibly discarded or diffused laterally. Upon reconsideration, this approach is not always necessary.

One idea that could easily be applied to simple CAD models is to print partial layers in between complete layers. These layers would be used only to fill surface gaps. Gaps along down-facing surfaces could be filled by locally printing extra binder. The super-saturation of the region would necessitate downward bleeding, thereby filling the gaps. The figure below illustrates this concept. The extra binder would be printed during a second pass over the same layer (layer 2 in the example) before spreading powder.

![Diagram of print layers](image)

**Figure 9.10 Gap-Fill Concept for Down-Facing Inclined Surfaces**

The approach for up-facing surfaces is slightly different, because a powder layer is necessary to position the top surface of a printed region. The next figure shows a possible sequence. After printing one layer under normal conditions, an intermediate powder layer is spread. Binder is only printed in gap regions (along the boundary), so that saturation in bulk regions does not become unacceptably high. Then the full second layer is printed. All layers are spaced at half-
height thickness. Up-facing surfaces are more difficult and time-consuming than down-facing surfaces because very thin powder layers must be spread.

![Diagram of target boundary and layering](image)

**Figure 9.11 Gap-Fill Concept for Up-Facing Inclined Surfaces**

The implementation of this idea requires more advanced software development to handle various exception cases. For example, a gap-fill layer should never be printed as the top-most layer in a part, because it would form a brim on the top surface. An important physical consideration is the ability to spread very thin layers of powder, especially if previous layers are still wet with binder. The concept of gap-fill layers has been attempted only on simple semi-cylinder test parts. Visual observations showed some improvement in smoothness, but no quantitative conclusions were established. The development of this concept was constrained primarily by time and resource constraints, but continued investigation is certainly recommended for future work.
9.2 Alternative Printing Styles

The term "printing style" is a loosely defined term that describes the way in which the digital information is actually transformed into binder-filled regions of material. It encompasses issues like line sequencing, print direction, and jet assignments. Chapters 6 explained the way in which a baseline printing style is encoded from software into specific machine instructions. Chapter 8 presented some enhancement techniques that modify the basic print style to achieve better surface quality.

All possible printing styles have certainly not been investigated. For example, the order in which lines are printed in each layer has always been sequential in the direction of the positive slow axis (back to front). This order can be alternated (back to front, then front to back) or the lines can even be randomized within each layer. This may offer improved uniformity of mechanical properties.

More fundamental changes to the machine could lead to an even wider range of possibilities. A vector-scanning printhead (arbitrary X-Y along polygon boundaries) may make it easier to trace boundaries in a continuous manner, instead of having to examine the boundary on a pass-by-pass basis. Each concept for a different print style carries a set of implications, which may involve trade-offs between surface finish, dimensional accuracy, computational intensity, machine design, and production rate. The following ideas focus on methods that would make process planning a more straight-forward task, without deviating far from the basic machine design. Basic characteristics of the current machine are continuous-jet technology with uniform droplet size, and raster scanning coverage of each layer.
9.2.1 Finishing Passes

The raster machine architecture contributes many strengths to 3D Printing. Among the most notable are high rate, low cost, and volume capacity. Every effort has been made to capitalize on the high rate virtue, by covering the print area with the fewest number of printhead passes. In most of the techniques presented in Chapter 8, the process planning software faced a significant challenge in making improvements with minimal sacrifice to production rate. In many cases, however, improvement in surface quality far outweigh the need for ultimate speed, and some compromise is necessary. This is the case for split-directional printing (Chapter 5), in which each layer is covered in twice as many passes as minimally necessary to span the powder bed.

Process planning for machining includes the separation of rough cuts and finishing cuts. Separate cuts (which may sometimes include a tool change) across the same vicinity of the workpiece result in a significant time penalty, but are used to achieve better surface finish. The process conditions such as feed rate and depth-of-cut between the rough cuts and finishing cuts are often quite different. The toolpath generation software can apply more stringent constraints on the finishing passes, while relaxing requirements for the rough cuts.

Three Dimensional Printing can also benefit from this approach. The next figure illustrates the basic concept. The bulk areas of polygons in each layer are printed under "rough" conditions. For example, the printhead speed may be relatively fast, the line spacing may be relatively wide, and the fill patterns may be relatively broad. Then the external boundaries are printed using "finishing" conditions. Examples may be slower printhead speed (to take advantage of fast axis resolution), more narrow line spacing, and finer deflection resolution.
Barriers that were encountered in the development of enhancement techniques in Chapter 8 might have been overcome more easily if rough passes and finishing passes were treated separately. For example, when implementing sub-raster scanning, much software overhead was required to properly code changes between interior and boundary segments in a single pass. There may be some issues regarding the uniformity of binder saturation, but initial experiments (Chapter 4) have shown that the process has some tolerance for locally high concentration of binder.

It is certainly not impossible to develop efficient techniques that perform printing with the least number of passes while properly managing surface detail. In fact this was achieved successfully for many cases encountered in Chapter 8. However, a clean separation between the relatively simple responsibility of filling material and the critical goal of high surface quality might be quite advantageous. This is especially true if the process parameters for each approach are under continual optimization -- and sometimes even in different directions (e.g. faster or slower printhead speed).
The fundamental concept of separating rough passes and finishing passes may be taken to various levels of advancement. More ambitious implementations may even open the possibility of different droplet sizes or different binder materials. Developers of the 3D Printing machine have designed the printhead with such flexibility in mind; different nozzles can be mounted on the same printhead. This could lead to benefits far beyond surface finish. A different material or binder concentration would allow local control over physical properties such as hardness or permeability.

9.2.2 Condensed Jet Spacing

The jet spacing of the current 3D Printing machine is fixed at 1/6 of an inch, or 4233 microns. This imposes some constraints with respect to line spacing and the space that must be covered by each jet during a pass (as mentioned in Chapter 7). The software data structures had to be designed such that the physical jet spacing could influence data search in each layer. This is because jet spacing is relatively wide compared to typical line width (200 microns). For example, if one print line was being encoded for jet #0 at slow axis position 10000, the next jet corresponding to that alignment would be at slow axis position 14233. In the next pass these positions might increment to positions 10200 and 14433, respectively. All the while it is necessary to note geometric features in the immediate vicinity to permit techniques like edge tracing and sub-raster feature snapping. The encoding software and data structures must thus be able to access geometric information with much flexibility.

It would simplify the data processing somewhat if geometric information in a layer could be processed in more continuous regions, and suggested by the right half of the figure below. The
distinction is analogous to the difference between a broom and a rake. With the packed jet spacing, various cross-checking routines could be performed completely within one region of the layer before moving onto the next. In effect the printing would be done in dense strips instead of a set of widely-spaced lines.

![Diagram of multi-jet coordination with tightly packed jet spacing](image)

**Figure 9.13 Multi-Jet Coordination with Tightly Packed Jet Spacing**

Packing the jets spatially can be accomplished by rotating the printhead or offsetting the deflection cells. Some options are listed below. Electronically-triggered delays could conveniently adjust for the jet separation along the fast axis. Offsetting is preferred to rotating because the deflection axis remains parallel to the y-axis.
The condensed jet spacing could simplify process planning requirements, but at this time the benefits do not warrant the necessary modifications to printhead design. However, the concept is still worth noting for future evolution of the 3D Printing machine. For example, in a production mode it may be desirable to use a printhead with 100 jets or more, and perhaps the "paintbrush" approach would offer more valued simplicity.

9.3 Comprehensive Process Planning

Process planning in the context of this thesis has focused specifically on droplet placement configurations to improve surface finish and dimensional accuracy. However, process planning in the general sense encompasses a wide variety of tasks, as described in Chapters 1 and 2. A natural extension of this thesis work would pursue a more comprehensive approach to process planning, including but not limited to the following:
• Prioritization to resolve choices multiple design criteria
• Material selection based on specific demands for part properties
• Part orientation to optimize build rate or to isolate specific curved surfaces
• Rule-based parameter selection to manage known information and decision making
• Integration with design rules that define geometric and functional constraints

[Müller] has also outlined concepts for a more complete process planning system, that even considers selection among different rapid prototyping techniques. Conventional process planning for the above tasks often requires a substantial amount of accumulated knowledge and well-integrated inferences. Advances in process planning for other manufacturing techniques have leaned toward expert systems applications [Kumara] to capture expertise in such areas. Three Dimensional Printing and other solid freeform fabrication techniques are inherently driven by symbolic information, and therefore are particularly open to great benefits from expert systems in the future.
10. Conclusions

Three Dimensional Printing is a manufacturing process that has been in development for only six years. Over this brief history, researchers have achieved much understanding of process science as well as advancement of machine capabilities. The contributing work of this thesis has been to extend the understanding of the physical process, specifically in terms of how well-chosen machine instructions can achieve the highest part quality. A brief review of the key points and conclusions from each chapter follows.

10.1 Process Characterization

Three Dimensional Printing is a constructive process that builds from a hierarchy of elements: primitives, lines, and layers. These building elements are created and placed according to distinctly different machine axes. Therefore, the surface finish and dimensional accuracy are governed by distinctly different considerations. The relevant factors have been identified and organized in Chapter 3, driving the options and strategies for improvement of part quality.

10.2 Process Modeling

Quantitative models have been developed to predict the geometry of resultant features (e.g. line width) based on controllable variables (e.g. binder dose). The models offer a way to analytically estimate printing parameters such as line spacing. Three Dimensional Printing is a sequential process (droplet by droplet) with potentially billions of stepwise events. Therefore, an
assumption of temporal independence was made to reduce the complexity of process planning. Experiments showed that the overall geometrical features were in agreement with the temporal independence assumption, although some counter-examples were noted (e.g. line pairing).

10.3 Machine Capability

The 3D Printing machine achieves tremendous flexibility by employing proportional deflection of binder droplets. Printing onto emulsion-coated substrates has been an extremely useful tool for studying deflection control without having to print onto powder. These experiments have shown that the deflection of printed segments is accurate to better than 2 microns, and that the deflection responds to its full potential in terms of frequency (tested at 64 kHz).

10.4 Datafile Processing

The fundamental steps for converting a 3-D surface model into machine instructions are slicing, rastering, and encoding. Robust file processing requires thorough consideration of a wide variety of geometric conditions. Separating the rastering and encoding steps has offered a clear distinction between geometric definitions and machine-specific operations.

10.5 Parameter Selection

There are several process parameters to be considered when planning droplet arrangement. The parameters span domains of software, machine hardware, and material behavior. Compromise among these parameters are often necessary to satisfy a broad range of objectives including surface finish, dimensional accuracy, feature resolution, production rate, and structural strength.
10.6 Surface Enhancement Techniques

Implementing the high-resolution, high-speed capabilities of proportional deflection in a raster architecture has made marked improvements in both surface finish and dimensional accuracy. These techniques are based on bridging an understanding of process physics with a detailed knowledge of machine performance. Boundary offset and sub-raster feature snapping make it possible to improve dimensional tolerances from about ±100 microns to about ±20 microns. Incline tracing for shallow inclines have reduced surface roughness from 45 microns to 25 microns ($R_a$).
Appendix A  Derivations

Appendix A1  Footprint Radius of a Spherical Section Cut by a Plane

The footprint radius $a_{\text{sphere}}$ formed by a spherical section cut by a plane can be determined as a function of the sectioned volume $V$ and base angle $\beta$. This derivation is completely general, but for 3D Printing it may be applied to a droplet of known volume printed onto a substrate with a known contact angle (Chapter 5). The figure below shows the 2-D cross-section of the layout.

![Figure A.1 Analytic Geometry of a Circular Arc Cut by a Plane](image)

The volume of a section of a sphere can be related to the radius $R$ of the sphere by analytic geometry and standard integration techniques. If $h$ is the distance from the top of the sphere to the plane that divides it, the volume of the spherical section is [Edwards]:

$$V = \frac{\pi}{3} h^2 (3R - h)$$

Equation A.1
By trigonometry, \( h = R (1 - \cos \beta) \) and \( R = \alpha_{\text{sphere}} / \sin \beta \). Therefore, the volume of the partial sphere can be expressed entirely in terms of the base angle \( \beta \) and the footprint radius \( \alpha_{\text{sphere}} \).

Algebraic rearrangement leaves the desired expression.

\[
V = \frac{\pi}{3} \left( \frac{\alpha_{\text{sphere}}}{\sin \beta} \right)^3 \left[ 3(1 - \cos \beta)^2 - (1 - \cos \beta)^3 \right] \\
\alpha_{\text{sphere}} = \left[ \frac{3V \sin^3 \beta}{\pi \left[ 3(1 - \cos \beta)^2 - (1 - \cos \beta)^3 \right]} \right]^{1/3}
\]

Equations A.2 and A.3
Appendix A2  Footprint Radius of a Cylindrical Section Cut by a Plane

The footprint radius $a_{cylinder}$ of a circular cylinder cut by a plane parallel to its longitudinal axis can be expressed in terms of the sectioned volume per unit length $V/L$ and the base angle $\beta$. This derivation is completely general, but for 3D Printing it may be applied to a line of known volume printed onto a substrate with a known contact angle. A 2-D cross-section perpendicular to the longitudinal axis appears as in Figure A.1.

The volume per unit length of the section above the plane can be determined by subtracting the area of the triangular region below the plane from the entire arc section of the cylinder swept over an angle $2\beta$. The area of both triangles together is $a \cdot R \cos \beta$. The area of the arc is the area of a complete circle $\pi R^2$ reduced by the fraction $(2\beta)/(2\pi)$. The radius is expressed in terms of the footprint parameter using $R = a / \sin \beta$ (the subscript is left off for conciseness). Algebraic manipulation rewrites the above equation with $a_{cylinder}$ as the dependent variable.

$$\frac{V}{L} = \left(\frac{2\beta}{2\pi}\right) \cdot \pi R^2 - aR \cos \beta = a^2 \left(\frac{\beta}{\sin^2 \beta} - \frac{\cos \beta}{\sin \beta}\right)$$

Equation A.4

$$a_{cylinder} = \left[\frac{V / L}{(\beta - \sin \beta \cos \beta) / \sin^2 \beta}\right]^{1/2}$$

Equation A.5
Appendix A3  Roughness Profile from an Analytic Circular Model

A profile determined by (uniform) circles can be characterized by two parameters, the radius of the circles and the spacing between them.

![Figure A.2 Roughness Profile Along Circle Boundaries](image)

The equation below gives a common analytic equation for a circle centered at the origin.

\[ x^2 + y^2 = r^2 \quad y = \pm \sqrt{r^2 - x^2} \]

Equation A.6

The roughness parameter \( R_a \) is defined as follows, in terms of the profile \( y \), the interval length \( L \), and the profile mean. The symmetry of the problem makes it possible to run the interval length from zero to half of the circle spacing, so \( L = a/2 \).

\[ R_a = \frac{1}{L} \int_0^L |y - \bar{y}|dx \]

Equation A.7
There are two particular issues in solving this integral. The first is that the mean must be first determined in terms of the profile. This draws upon one of the general applications of the integral, as expressed by the following equation.

\[ \bar{y} = \frac{1}{L} \int_0^L y \cdot dx \]

Equation A.8

Substituting the equation for a circle leaves a square-root term inside the integral. The solution from integral tables [Edwards] is as follows.

\[ \int \sqrt{r^2 - x^2} \, dx = \frac{x}{2} \sqrt{r^2 - x^2} + \frac{r^2}{2} \sin^{-1} \left( \frac{x}{r} \right) + C \]

Equation A.9

Therefore, the mean is expressed in terms of the problem geometry (r and L) as shown below.

\[ \bar{y} = \frac{1}{2} \sqrt{r^2 - L^2} + \frac{r^2}{2L} \sin^{-1} \left( \frac{L}{r} \right) \]

Equation A.10

Returning back to the original expression for \( R_o \), another issue is the handling of the absolute value. Part of the curve between zero and L necessarily occurs above the mean, and the remaining part lies below the mean. The point along the x-axis at which the curve crosses the mean will be called \( b \), and is known by entering the mean back into the equation for a circle.

\[ b = \sqrt{r^2 - \bar{y}^2} \]

Equation A.11
Equation A.8 is then split to account for the absolute value requirement.

\[ R_a = \frac{1}{L} \int_0^b (y - \bar{y})dx + \frac{1}{L} \int_b^L (\bar{y} - y)dx \]

Equation A.12

Each of these integral terms is solved by standard integration, using Equation A.9 as a template.

\[ R_a = \frac{1}{L} \left[ \frac{b}{2} \sqrt{r^2 - b^2} + \frac{r^2}{2} \sin^{-1} \left( \frac{b}{r} \right) - b\bar{y} \right] + \frac{1}{L} \left[ L\bar{y} - \left( \frac{L}{2} \sqrt{r^2 - L^2} + \frac{r^2}{2} \sin^{-1} \left( \frac{L}{r} \right) \right) \right] - \frac{1}{L} \left[ b\bar{y} - \left( \frac{b}{2} \sqrt{r^2 - b^2} + \frac{r^2}{2} \sin^{-1} \left( \frac{b}{r} \right) \right) \right] \]

Equation A.13

Finally, the roughness parameter \( R_a \) is written in terms of the circle spacing (using \( L \) instead of \( a \)) and the radius \( r \). The mean and \( x \)-intercept \( b \) are defined as above.

\[ R_a = \frac{1}{L} \left[ (L - 2b)\bar{y} + b\sqrt{r^2 - b^2} + r^2 \sin^{-1} \left( \frac{b}{r} \right) - \frac{L}{2} \sqrt{r^2 - L^2} - \frac{r^2}{2} \sin^{-1} \left( \frac{L}{r} \right) \right] \]

Equation A.14
Appendix A4  Uniform Spacing of Sub-Levels Between Print Lines

The spacing of sub-levels between print lines is determined by applying geometric constraints. The objective is to identify the positions for each sub-level analytically. The spacing values may be applied to printing techniques that seek uniform distribution, as is the case for diffuse line termination (Chapter 8). The following figure shows two consecutive lines at positions $y_i$ and $y_{i+1}$, separated by the line spacing $\Delta y$. Also shown are the $n$ sub-levels that flank each line.

\[ \Delta w = \frac{\Delta y}{n} \]

Equation A.15

Figure A.3 Uniformly Spaced Sub-Levels Between Print Lines

The objective is to determine the appropriate $y$-position of any sub-level $j$ of any line $i$. The first basic relation is that the spacing $\Delta w$ among sub-levels is equal to the line spacing $\Delta y$ divided by the number of levels $n$. In fact, for uniform spacing, this must be the spacing between any two consecutive sub-levels of any centerline $i$, as well as the spacing between the last sub-level of line $i$ and its successor $i+1$. These imposed constraints and the geometry in Figure A.3 above yield the following set of equations.
\[(y_{i,j}) = y_i + \left(\left(\frac{n-1}{2} + j\right) \frac{1}{n} - 1\right) \cdot \Delta y\] 

Equation A.19

Algebraic substitution of the above equations yields a final expression for any position \((y_{i,j})\) of the sub-level \(j\) associated with printhead pass \(i\), in terms of the line spacing \(\Delta y\) and the number of sub-levels \(n\).

For example, Equation A.19 will dictate that if there are \(n = 2\) sub-levels, the sub-levels should be set at \(\pm \Delta y/4\) from the center of each printhead pass position \(y_i\). If there are \(n = 3\) sub-levels, then there should be one level on center, and one level each at \(\pm \Delta y/3\) from the center of each printhead pass position \(y_i\).
Appendix B  Software Programs

In addition to the datafile processing programs described in Chapter 6, several utility programs have also been developed. These programs have been used to inspect model data and to make corrections when necessary. A Direct File Construction program (3dpConstruct) offers a convenient text-based (or spreadsheet) interface to specific machine instructions, for experiments that require a high degree of control. These programs are built upon object-oriented modules for manipulating basic components such as polygons, lines, and points. Several visual inspection programs have also been developed for verifying correctness of data manipulations. These programs offer the ability to view layer cross-sections at various zoom levels and to step through component elements (e.g. the different printhead passes). Text-based versions have also been developed by which the user can query selected information from the binary data files.

The following sections describe the functionality and usage of the core set of programs developed to enable data-level process planning for 3D Printing (except slicing, which was developed by others outside of this thesis work). The actual software code has been developed in the C programming language for UNIX-based operating systems (although many C++ design principles have been incorporated). The programs have been made cross-compatible for variations of UNIX, specifically IRIX 5.2 (Silicon Graphics, Inc.) and Ultrix 4.2 (Digital Equipment Corp.). Graphical interface programs were developed using Motif 1.2 (Open Software Foundation) and its subordinate toolkits (Xt, Xlib) for the X-Windows environment. All source code and executable files have been left in the care of the 3D Printing staff.
Appendix B1  raster

DESCRIPTION

Rasters an SLC polygonal slice file into an RST raster segment file. The input polygonal slice file must comply to format 19940315 and the output file complies to format 19960422.

USAGE

raster [options] <file.slc> <file.rst>

OPTIONS

-dx <num>  : raster with x-axis resolution of "num" microns, default 20.
-dy <num>  : raster with y-axis resolution of "num" microns, default 20.
-dz <num>  : raster with z-axis resolution of "num" microns, default 170.

LIMITATIONS

When creating a new output file, there is no check to see whether a file with the same name already exists.

FILES

raster.c
rstFile.h, rstFile.c
3dpFile.h, 3dpFile.c
Appendix B2  encode

DESCRIPTION

Encodes an RST raster segment file into a 3DP machine data file.
with specific instructions for the 3D Printing Alpha Machine.

The input file must comply to format 19960422,
and the output production file complies to format 19940811.

USAGE

encode [options] <file.rst> <file.3dp>

OPTIONS

-d <code> : set the direction to "1way" (default), "2way", or "split"
-g <code> : set the grid stacking to RECTANGULAR (default) or HEXAGONAL
-j <num> : Assign instructions to 'num' jets (default 8, range 1-8)
-p <file> : Apply fill patterns defined in 'file'
A pattern file is simply a text file with 256
deflection levels (in microns), separated one per line.
-s <num> : Set line spacing to 'num' microns (default 170)

LIMITATIONS

When creating a new output file, there is no check
to see whether a file with the same name already exists.

The pattern fill is currently limited to a single pattern for all jets.

FILES

encode.c
rstFile.h, rstFile.c
3dpFile.h, 3dpFile.c
Appendix B3  slcOffset

DESCRIPTION

Offsets the polygons of a .SLC file to compensate for the finite size of the primitive building elements.

The input polygonal slice file must comply to format 19941018. The program creates a new file of the same format. The input specifications are detailed in the file 'f19941018.txt'.

The input file should have the suffix '.slc'. with the suffix changed from '.slc*'.

USAGE

offset [options] <filename>.slc

OPTIONS

-a <num> : Filter vertex angles more narrow than 'num' radians.
-l <num> : Filter edge lengths shorter than 'num' microns.
-d <num> : Change the offset distance to 'num' microns.
(default 100)

LIMITATIONS

When creating a new output file, there is no check to see whether a file with the same name already exists.

If any required conditions for the input file format are violated, unpredictable errors can result.

FILES

slcOffset.c
Polygon.h, polygon.c
slcFile.h, slcFile.c
Appendix B4  stlView

DESCRIPTION

Shows image of an STL data file (in wireframe mode), using a menu-driven graphical user interface.

Both ASCII and binary forms of STL are supported; and the program automatically tries to decide which one is applicable.

It is recommended that the scrollbars begin at the bottom left corner.

The program assumes millimeter units.
STL files in inches will simply appear much smaller.

USAGE

stlView

OPTIONS

The graphical interface allows the user to select:

Vantage Point (one of three orthogonal views)
Viewing Frame (panning by buttons or mouse)
Zoom Level

Reference axes are also available, and are oriented as follows:

Top View: X-axis increases left, Y-axis increases up
Front View: X-axis increases left, Z-axis increases up
Side View: Z-axis increases left, Y-axis increases up

Grayed-out regions indicate negative quadrants of space.

LIMITATIONS

Warning messages and error handling are not very advanced. (For example, the program quits abruptly if a non-STL file is opened.)

It may be difficult to locate the part in the viewing space, especially if the STL file is in negative space.

FILES

stlView.c
stlFile.h, stlFile.c
byte.h, byte.c
Appendix B5  slcView

DESCRIPTION

Shows image of an polygonal slice (SLC) data file, using a menu-driven graphical user interface. The file format corresponds to format 19940315.

USAGE

slcView

OPTIONS

The graphical interface allows the user to scroll through layers, polygons, and vertices; and to adjust zoom level.

Panning may be done by clicking and dragging the mouse in the drawing area.

In addition to the graphical layer slider, the up and down arrows also navigate from layer to layer.

LIMITATIONS

FILES

  slcView.c
  slcFile.h, slcFile.c
  byte.h, byte.c
  polygon.h, polygon.c
  list.h, list.c
  line.h, line.c
  point.h, point.c
Appendix B6  rstToTiff

DESCRIPTION

'rstToTiff' exports a layer of an RST file as a 2-D TIFF image.
TIFF is a common graphics standard defined by Adobe Systems.
This program is in lieu of one a specific "rstView" program.

USAGE

rstToTiff [options] <file.rst> <file.tif>

OPTIONS

-1 <n>  : exports the cross-sectional image of layer number "n"
  default layer is layer 1

LIMITATIONS

The output file may be much larger than most common 2-D images.
1024x1024 is often considered high-resolution,
but the cross-sections of RST files for 3D Printing may be 30000x15000.
TIFF viewers may not be able to handle images with such large data size.
For simple visual inspection purposes it may be preferable to raster
at low resolution (e.g. 100 microns x 100 microns).

FILES

rstToTiff.c
slcFile.c, slcFile.h
byte.c, byte.h
Appendix B7  3dpView

DESCRIPTION

Shows image of a 3DP Alpha Machine (3DP) data file, using a menu-driven graphical user interface. The file format corresponds to format 19940811.

USAGE

3dpView

OPTIONS

The graphical interface allows the user to scroll through layers, passes and transitions; and to adjust zoom level.

Panning may be done by clicking and dragging the mouse in the drawing area.

In addition to the graphical layer slider, the up and down arrows also navigate from layer to layer when the focus is in the drawing window (click inside the window).

LIMITATIONS

FILES

3dpView.c
3dpFile.h, 3dpFile.c
byte.h, byte.c
list.h, list.c
Appendix B8  slcExpand

DESCRIPTION

'slcExpand' expands the contents of an encoded SLC datafile. The program interactively reports the numerical data for layers, polygons, and vertices.

The user is prompted for layer number, and then at a lower level for polygon number.

The input file must comply to format 19940315.

USAGE

   slcExpand <file.slc>

OPTIONS

   none

LIMITATIONS

FILES

   slcExpand.c
   slcFile.c, slcFile.h
   byte.c, byte.h
Appendix B9  rstExpand

DESCRIPTION

'rstExpand' expands the contents of an RST datafile.

The syntax "AxB" in a notation of run-length encoding, in which the value "A" runs for a length of "B" units (pixels).

The input text file must comply to format 19960422, as defined in 'format19960422.txt'.

USAGE

rstExpand <file.rst>

OPTIONS

none

LIMITATIONS

FILES

rstExpand.c
rstFile.c, rstFile.h
byte.c, byte.h
Appendix B10  3dpExpand

DESCRIPTION

'3dpExpand' expands the contents of an encoded 3DP datafile. The program interactively reports the numerical data for layers, passes, transitions, and patterns.

The user is prompted for layer number, and then at a lower level for pass number.

The input text file must comply to format 19940811, as defined in 'format19940811.txt'.

USAGE

3dpExpand <file.3dp>

OPTIONS

none

LIMITATIONS

FILES

3dpExpand.c
3dpFile.c, 3dpFile.h
byte.c, byte.h
Appendix B11  slcConstruct

DESCRIPTION

"slcConstruct" constructs an SLC datafile form a text file description. The program allows designers to specify layers, polygons, and vertices.

The input text file must comply to format 19951121, described below. The output production file complies to format 19940811. By convention the output file should have the suffix ".slc".

The (-x) and (-y) options allow replication of the entire file along the x and y axes, respectively. If neither option is used, there is only one instantiation of the file.

USAGE

    slcConstruct [options] <infile> <outfile>

OPTIONS

    -x <n> <dx> : instantiate 'n' number of times (including original) along the x-axis with an origin offset of 'dx' microns.

    -y <n> <dy> : instantiate 'n' number of times (including original) along the y-axis with an origin offset of 'dy' microns.

LIMITATIONS

    The replication options do not check whether the polygons overlap. The designer is responsible for choosing valid replication parameters.

    When creating an output file, there is no check to see whether a file with the same name already exists. An existing file may be inadvertently over-written.

FILES

    slcConstruct.c
    slcFile.c, slcFile.h
    byte.c, byte.h
FILE FORMAT 19951121

A construction file reads a text-file specification
for layers and polygons; and creates a 3DP data file.
Various construction elements ("building blocks") are first defined,
then used in combination to form a complete part.

All positional units are in microns,
and the file format is case sensitive (UPPERCASE/lowercase matters).

This format is recognized by a "signature" at the very beginning of the file.
The number "19951121" (text) must be the first entry in the file.

The file is constructed from a collection of layer definitions,
which are listed at the end of the file using the syntax below.
The layers are numbered sequentially and contain the z-position
in microns followed by the name of a layer definition.
This "construct paragraph" must be at the end of the file.
Each layer name must be a unique sequence of characters
beginning with a letter and containing no spaces.
The designer is responsible for choosing valid z-positions.

CONSTRUCT
{
  1   <z>   <layer name>
  2   <z>   <layer name>
    ...
}

Any layer definition used in the CONSTRUCT section
must be previously defined in the file, using the syntax below.
A layer definition consists of a numbered sequence of polygons
at specified fast axis (x) and slow axis (y) positions.
Each polygon name must be a unique sequence of characters
beginning with a letter and containing no spaces.
The designer is responsible for choosing valid coordinate positions.

DEFINE LAYER <layer name>
{
  1   <x>   <y>   <polygon name>
  2   <x>   <y>   <polygon name>
    ...
}

Similarly, each polygon used in a layer definition
must be previously defined in the file, using the syntax below.
A polygon definition consists of an ordered sequence of vertices
at specified (x,y) positions.
If the vertices are listed counter-clockwise,
the polygons are considered to be filled.
If the vertices are listed clockwise,
the polygons are considered to be empty (hole within another polygon).

DEFINE POLYGON <polygon name>
{
  1   <x>   <y>
  2   <x>   <y>
    ...
}

239
**Appendix B12  3dpConstruct**

**DESCRIPTION**

"3dpConstruct" constructs a 3DP datafile form a text file description. The program allows designers to specify patterns, layers, passes, and transitions used by the 3D Printing Alpha Machine.

The input text file must comply to format 19951228, described below. The output file complies to format 19940811. By convention the output file should have the suffix ".3dp".

**USAGE**

3dpConstruct [options] <infile> <outfile>

**OPTIONS**

- `-t <n>` : Copy the data for jet 0 to jet "n" in every pass. When this "twin" option is used, the original file may contain data for jet 0 only. The copies require no additional printhead passes.

- `-x <n> <dx>` : Instantiate 'n' number of times (including original) along the x-axis with an origin offset of 'dx' microns.

- `-y <n> <dy>` : Instantiate 'n' number of times (including original) along the y-axis with an origin offset of 'dy' microns. The copies are made with additional printhead passes.

**LIMITATIONS**

The replication options do not check whether transitions overlap. The designer is responsible for choosing valid replication parameters.

When creating an output file, there is no check to see whether a file with the same name already exists. An existing file may be inadvertently over-written.

Patterns for the entire file can only be defined at the beginning of the first layer, even though the machine control software is designed to accept re-definitions at any arbitrary layer.

There is a limit of 8 patterns per jet. If more than 8 patterns are requested, new patterns overwrite old ones.

**FILES**

3dpConstruct.c
3dpFile.c, 3dpFile.h
byte.c, byte.h
FILE FORMAT 19951228

A construction file reads a text-file specification for layers, passes, and transitions; and creates a 3DP data file. Various construction elements ("building blocks") are first defined, then used in combination to form a complete part.

All positional units are in microns, and the file format is case sensitive (UPPERCASE/lowercase matters).

This file format is recognized by a "signature" at the very beginning of the file. The number "19951228" (text) must be the first entry in the file.

The file is constructed from a collection of layer definitions, which are listed at the end of the file using the syntax below. The layers are numbered sequentially and contain the z-position in microns followed by the name of a layer definition. This "construct paragraph" must be at the end of the file. Each layer name must be a unique sequence of characters beginning with a letter and containing no spaces. The designer is responsible for choosing valid z-positions.

CONSTRUCT
{
  1  <z>  <layer name>
  2  <z>  <layer name>
  ;
}

Any layer definition used in the CONSTRUCT section must be previously defined in the file, using the layer-definition syntax below. A layer definition consists of a numbered sequence of print passes at specified slow axis (y) positions. Each pass name must be a unique sequence of characters beginning with a letter and containing no spaces. The designer is responsible for choosing valid y-positions.

DEFINE LAYER <layer name>
{
  1  <y>  <pass name>
  2  <y>  <pass name>
  ;
}

Similarly, each pass used in a layer definition must be previously defined in the file, using the pass-definition syntax below. A pass definition consists of a numbered sequence of transitions at specified fast axis (x) positions, for specified jets. The designer is responsible for choosing valid x-positions, jet numbers, and deflection values.

DEFINE PASS <pass name>
{
  1  <x>  <jet>  <value>
  2  <x>  <jet>  <value>
  ;
}
All of the transitions for a particular jet must be stated before stating the transitions for the next jet. The x-positions for any given jet should be in ascending order. The transition "value" may be a deflection magnitude (e.g. -100), the word "OFF", or the name of a pattern definition (described below). A deflection magnitude of zero means on-center printing. Each pattern name must be a unique sequence of characters beginning with a letter and containing no spaces.

The designer is responsible for specifying logical order with respect to the transitions. The machine necessarily makes forward and reverse passes (with respect to the x-axis). Odd-numbered passes move in the direction of increasing x, and even-numbered passes move in the direction of decreasing x. (Therefore an even-numbered pass would always begin with "OFF").

Each pattern requested in a pass definition must be previously defined in the file, using the pattern-definition syntax below. A pattern definition consists of a numbered sequence of transitions. There may be no more than 256 transitions per pattern. If there are less than 256 transitions, the transitions are simply copied to fill an array of 256 values. This array of 256 values is repeated as long as the pattern is active.

```
DEFINE PATTERN <pattern name>
{
  1   <value>
  2   <value>
  ...
}
```
Appendix C   File Format Specifications

This appendix lists file format definitions for the data representations used in 3D Printing and described in Chapter 6. These formats include the polygonal slices (SLC), raster segments (RST), and machine-encoded data (3DP). The STL surface facet representation is defined by 3D Systems, Inc., and is not presented in this appendix.

Key design issues for file formats include the following. The issues were addressed in so far as possible, balancing against expedience and level of expertise.

- Memory efficiency
- Accuracy and resolution
- Speed
- Portability to different platforms
- Robustness against flawed data
- Compatibility
- Extensibility
- Scalability: no arbitrary limits on data size
- Completeness

File input/output modules have been written for each of the file formats, with names stlFile.c, slcFile.c, rstFile.c, and 3dpFile.c. Compiling higher-level programs with these modules allow convenient use of general purpose functions like slcWriteVertex(x, y).
Appendix C1  SLC Polygonal Slice Format

A polygonal slice file contains a representation of a physical object. The file is organized in a sequence of two-dimensional layers. Each layer consists of simple closed polygons. Each polygon is represented as an ordered list of vertices. All positional information is expressed in micron units.

By convention, these files have the file name suffix ".slc". The file format was developed as an intermediate file processing stage for fabrication of CAD models by the 3D Printing process.

The binary data is enclosed by two key words, "DATA\n" and "END\n" (The \n represents a required newline character). File contents before "DATA" and after "END" are non-critical.

All data is written in (signed) 32-bit integers, in binary format. The range for any data value is from -2147483648 to +2147483647, although the file should never contain negative values. The four bytes of a 32-bit word are written from low to high (the file format was originally developed on a machine that wrote multiple bytes in this order by default).

As of April 1994, the most common source of a polygonal slice file has been the output of a slicing operation on a surface facet file from the Solider software package by Cubital, Ltd. As of the end of 1994, another source of the slice file has been the STL slicing program "sla2slc" by Phil Dench.

The first four bytes in the file must state the format ID, 19940315. This allows any reading program to immediately identify the format.

```
+--------------------------------------------------+
| File Format ID Number | +--------------------------------------------------+
```

The next several bytes may be used for comments, up to the character sequence "DATA\n". Thereafter begins the actual data describing part geometry.
The first two words are number of layers and layer spacing in microns. The layer spacing should not be used for vertical axis control, however. It is only provided for informational purposes. The z-Position stated for each layer should control the vertical axis.

+--------------------------------------------------------------+
| Number of Layers                                             |
+--------------------------------------------------------------+
| Layer Spacing                                                |
+--------------------------------------------------------------+

Every layer begins with three data words: layer index, vertical position, and number of polygons. The layer indices must occur sequentially, beginning from 1.

+--------------------------------------------------------------+
| Layer Index: 1, 2, 3, ...                                      |
+--------------------------------------------------------------+
| z-Position                                                   |
+--------------------------------------------------------------+
| Number of Polygons                                           |
+--------------------------------------------------------------+

Each polygon is headed by the number of vertices in the polygon. Then come the x and y coordinates of the vertices.

+--------------------------------------------------------------+
| Number of Vertices                                           |
+--------------------------------------------------------------+

The vertex information is a sequence of (x,y) coordinates. The ordering of the vertices is very important. If the polygon is traversed clockwise, it represents a filled region; if counter-clockwise, it represents an empty region. (This convention was adopted from the Cubital slicing program).

+--------------------------------------------------------------+
| x-Position                                                   |
+--------------------------------------------------------------+
| y-Position                                                   |
+--------------------------------------------------------------+
Appendix C2  RST Raster Segment Format

The raster file format is an intermediate representation of a CAD model, used in the file processing stages of 3D Printing. Essentially, it is a collection of 2-D images organized in layers. This offers a voxel (3-D pixel) representation of geometric data, providing a way to specify the state of each and every addressable point in the build volume.

The file format design is modeled closely after the TIFF file format (Adobe). The design of the TIFF file format should be studied to understand fully understand the organization of this RST file format. For reference:

Brown, C. Wayne and Shepherd, Barry J.  
Graphics File Formats: Reference and Guide  

The file contents are binary, and all entries are arranged in 4-byte words. This is the natural way to store 32-bit numbers on most machines. The bytes are ordered low to high.

The first three words in the file must be as follows. The file format signature is the number "19960422" (written in binary). The file consists of global parameters, such as x-axis resolution. The second word in the file identifies where to find the first parameter, which in turn chains to all other parameters. The third word in the file identifies where to find the layer index, which in turns lists the file locations of each layer in the file. Neither the first defined parameter nor the layer index need to be written at the beginning of the datafile, because the file positions are recorded here in the second and third words, respectively.

```
+------------------------------------|--|
| File Format Signature--------------|--|
+------------------------------------|--|
| File Position of First Parameter---|--|
+------------------------------------|--|
| File Position of Layer Index-------|--|
```

There must be at least one file parameter defined (see below). The maximum number of parameters is not limited, so that the file format becomes extensible as new requirements arise. Parameters can be edited independently, w/o rewriting the whole file. All parameters are listed as follows:

```
+------------------------------------|--|
| Parameter Code (arbitrary, but unique)|--|
+------------------------------------|--|
| Parameter Value----------------------|--|
+------------------------------------|--|
| File Position of Next Parameter (zero if last)|--|
```

247
The layer index is an index to the file positions of the start of data for each layer. It appears as follows:

```
<table>
<thead>
<tr>
<th>Number of Layers in the file</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Position of First Layer</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>File Position of Second Layer</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>etc...</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>File Position of Last Layer</td>
</tr>
</tbody>
</table>
```

The data for each layer may be organized in many different ways. Initially, the value8-length24 bit encoding below has been implemented. In this scheme, the first eight bits indicate a state, and the remaining 24 bits specify how long (in "pixels") the state runs. The end of a row is marked by a run with zero length. Every row should have the same number of total elements (pixels).

```
<table>
<thead>
<tr>
<th>Value</th>
<th>Run Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Run Length</td>
</tr>
<tr>
<td>etc...</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>0</td>
</tr>
</tbody>
</table>
```

Predefined values definitions are as follows (in hexadecimal notation):

- RST_OFF 0x00
- RST_ON_INTERIOR 0x80
- RST_ON_BOUNDARY 0xFF
PRE-DEFINED PARAMETERS ..........................................................

The initial set of pre-defined parameters for the RST format are listed below. New parameters may be added as the file content is extended with more options.

#define RST_HEIGHT 1 /* number of layers */
#define RST_WIDTH 2 /* rows per layer */
#define RST_LENGTH 3 /* elements per row */

#define RST_UNITS 4
#define RST_UNITS_UNKNOWN 0
#define RST_UNITS_MICRON 1

#define RST_RESOLUTION_Z 6
#define RST_RESOLUTION_Y 7
#define RST_RESOLUTION_X 8

#define RST_DATA_TYPE 9
#define RST_DATA_TYPE_UNKNOWN 0
#define RST_DATA_TYPE_BYTE 1

#define RST_COMPRESSION 10
#define RST_COMPRESSION_NONE 0
#define RST_COMPRESSION_8_24 1

#define RST_USER_ID 11
#define RST_USER_UNKNOWN 0
Appendix C3  3DP Machine Data Format

An alpha machine file contains a representation of physical objects to be fabricated by the 3D Printing process on the Alpha machine. By convention, such files have the suffix ".3dp".

The file is organized in an ordered sequence of layers. Each layer consists of a sequence of printhead passes. Each printhead pass consists of an ordered list of printhead transitions. Numerical data is stored in 32-bit (unsigned) integers and positions are expressed in micron units, unless otherwise specified.

The first four bytes in every .3dp file report the current file format. The format ID is an arbitrary (32-bit) number to check for compatibility. The format defined in this document is stated in the header of this file.

```
+----------------------------------------+------+
| File format ID                        |
+----------------------------------------+------+
```

Following the format ID is exactly 400 bytes allocated for comments. The machine control program should not depend on the comments; they are intended solely for the purpose of recording file history. So that the comments section can be easily previewed, each byte should be an ASCII text character.

```
+------------------------------------------------------------------+
| Comments (100 4-byte words = 400 bytes)                           |
+------------------------------------------------------------------+
```

The first block of critical information describes the overall geometry in the production file. The total number of printhead passes is provided for statistics only.

```
+----------------------------------------+------+
| Number of layers                       |
+----------------------------------------+------+
| Total number of printhead passes       |
+----------------------------------------+------+
```
The next six words report the spatial bounding box of the part.
These values are provided for automatic part placement in the powder bed.
The fast axis has a valid range from 0 to 300000,
the slow axis has a valid range from 0 to 150000,
and the vertical axis ranges from 0 to 300000.

<table>
<thead>
<tr>
<th>Fast Axis lower bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast Axis upper bound</td>
</tr>
<tr>
<td>Slow Axis lower bound</td>
</tr>
<tr>
<td>Slow Axis upper bound</td>
</tr>
<tr>
<td>Vertical Axis lower bound</td>
</tr>
<tr>
<td>Vertical Axis upper bound</td>
</tr>
</tbody>
</table>

A layer header begins each layer.
One forward & return cycle of the printhead counts as two passes.
The layer indices must be an ordered sequence 1, 2, 3, ...

| Layer index: 1, 2, 3, ...
| Vertical position in microns
| Number of printhead passes for this layer
| Number of pattern definitions |

If the number of pattern definitions at this layer is greater than zero,
the next data block will contain information for loading patterns.
This procedure is described at the end of this document.

After any pattern definitions (or if no patterns need to be defined),
the data for each of the printhead passes is listed.
In bi-directional printing, the pass index dictates printhead direction.
If it is an odd number, printing occurs in the positive fast axis direction;
if it is even, printing occurs in the negative fast axis direction.
The pass indices must be an ordered sequence 1, 2, 3, ...
Every pass must have at least two transitions.

| Pass index: 1, 2, 3, ...
| Slow axis position in microns
| Number of transitions for this pass |
Each transition is encoded into one 32-bit word, to promote direct download into memory chips.

<table>
<thead>
<tr>
<th>P</th>
<th>Fast axis position</th>
<th>#</th>
<th>T</th>
<th>deflection/pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>30</td>
<td>15</td>
<td>12</td>
<td>11</td>
</tr>
</tbody>
</table>

For most transitions, the bits are allocated in the following way (an exception is when a pattern fill is initiated):

0 - 11 droplet deflection from center in microns
12 transition flag
13 - 15 nozzle number, numbered 0-7
16 - 30 fast axis position in 10-micron units
31 pattern flag

Since negative numbers cannot be represented directly in binary (and since downstream interpretation of 2's complement is undesirable), the deflection magnitude has a constant offset of 2048, such that intended values between -2048 and 2047 are written as 12-bit numbers between 0 and 4095. Value 2048 corresponds to no deflection.

There are two condition flags, 'P' (bit 31) and 'T' (bit 12). The following table describes how the flags indicate possible conditions:

<table>
<thead>
<tr>
<th>P</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The transitions for any given row are sorted in increasing order, first on fast axis position, then on nozzle number. Each block of transition words terminates with a (32-bit) zero, which serves the purpose of data flow verification.

The relevant data ends when all of the layers have been read.
If the number of fill patterns is greater than zero
(noted at the beginning of any layer),
the file then lists pattern data in blocks of 256 words per pattern.
Each pattern block is headed by a pattern index, numbered 0-7,
and a nozzle number, 0-7.
The pattern index and nozzle number do not have to be sequential.

<table>
<thead>
<tr>
<th>Pattern index: 0, 1, 2, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nozzle number: 0, 1, 2, ...</td>
</tr>
</tbody>
</table>

The pattern index and nozzle number is followed by 256 data words,
in the same format as the transition word described above.

<table>
<thead>
<tr>
<th>P</th>
<th>Fast axis position</th>
<th>#</th>
<th>T</th>
<th>deflection/pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>31 30</td>
<td>15 12 11</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix D  Deflection Calibration

The file "vernier.3dp" represents a vernier test pattern for calibrating partial deflection on the 3D Printing Alpha machine. First a reference pattern is printed, consisting of 11 rows of paired lines segments. The rows are spaced uniformly (by an arbitrary distance of 3 mm). The gap between each pair lines is slightly more than 10 mm.

```
 _______  _______  
 _______  _______  
 _______  _______  
 _______  _______  
 _______  _______  
```

*Figure D.1 Deflection Calibration Reference Pattern*

Next, partially-deflected line segments are printed between the reference pairs. The line-up of the printhead is offset by 200 microns from where it was when printing the reference lines. Therefore, a deflection of 200 microns is needed to make a segment "line-up" with reference lines. Deflected segments are printed in the gap between the reference lines. The deflected segments have different deflection instructions, climbing from 100 microns to 300 microns in 20-micron increments. The middle (6th) row from the bottom is thus given an instruction for 200-micron deflection. The calibration condition is one in which the middle deflected row lines up exactly with the reference lines (the figure below shows a calibrated condition).
Figure D.2 Deflection Calibrated Condition

If the signal amplification is too weak, the "line-up" condition will occur early.

Figure D.3 Deflection Calibration Undershoot Condition

If the signal amplification is too strong, the "line-up" condition will occur late (shown on right).

Figure D.4 Deflection Calibration Overshoot Condition

Adjust Potentiometer #5 on the synchronization board to modify charge amplification. Counterclockwise rotation reduces amplification, clockwise adjustment increases it. It is expected that the file was written such that the deflection has a full range from -320 microns to +310 microns, corresponding to 10 microns of deflection per bit in the D/A conversion. All parts using this calibration must also be processed similarly.