DIGITAL MATERIALS:
VOXEL DESIGN, RAPID ASSEMBLY,
STRUCTURAL PROPERTIES, AND DESIGN
METHODS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Jonathan Daniel Hiller

May 2011

DIGITAL MATERIALS:

VOXEL DESIGN, RAPID ASSEMBLY, STRUCTURAL PROPERTIES, AND

DESIGN METHODS

Jonathan Daniel Hiller, Ph.D.

Cornell University 2011

The work presented here contains the first known comprehensive consideration of digital materials. Digital materials rely on a fundamentally new paradigm of manufacturing: Physical objects are composed of many discrete, aligned fundamental building units. Such an object is defined purely by the presence or absence of a physical voxel (3D pixel) at each defined location, and thus is fundamentally digital. *This implies that "perfect" objects can be physically fabricated with imperfect tools.* As a result, digital materials can be replicated over many generations without degradation. In contrast, existing manufacturing processes make use of electronic digital control systems to fabricate objects from a digital representation, but the physical objects they create are fundamentally continuous (or analog) in nature.

The specific contributions of this work fall into four categories: physical voxel design, rapid assembly of digital objects, structural properties thereof, and autonomous design methods. First, potential voxel designs were explored and analyzed for their suitability in a mass digital fabrication process. Microscale interlocking square tile voxels were fabricated and assembled to demonstrate the possibilities in high resolution digital materials. Second, two rapid assemblers were built to demonstrate both serial and parallel voxel deposition techniques. These were used to quickly assemble thousands of voxels into multi-material

freeform 3D shapes and show the possibilities of a massively parallel assembly process. Third, the precision and structural properties of objects made of many imperfect discrete units were explored. These experiments demonstrate the viability of precise large-scale multi material digital structures, as well as many inherent possibilities regarding tunable aggregate material properties. Lastly, design automation methods using evolutionary algorithms were explored to directly create blueprints for digital objects to meet high level functional goals. These methods were applied to demonstrate both functional static structures & mechanisms and dynamic locomoting soft robots.

BIOGRAPHICAL SKETCH

Jonathan Daniel Hiller was born in Everett, Washington on April 15, 1985 to Don and Julie Hiller and has one brother, David, two years his senior. His father is trained in electrical engineering and pursued a career in Lake Stevens, WA with Hewlett-Packard, which in later years became Agilent Technologies. Jonathan spent his childhood years running around in the woods and enjoying all manners of building and constructing increasingly complex things, from legos and model cars to toothpick bridges and forts. Through much trial and error and frustration, the intuition and skills of mechanical design began to be formed.

Jonathan attended public school in the Lake Stevens school district from primary through high school. His desire to be an engineer was clear to all from middle school onward, and directed his focus towards science and math. A CAD elective course in high school cemented mechanical engineering as his career direction of choice, although he was always fascinated by the power of programming. His senior year of high school he attended Everett Community College full time in order to stay challenged with advanced courses.

The following year he began his undergraduate career in Mechanical Engineering at the University of Washington, Seattle. There he found himself in a fertile environment that bridged his understanding of science with his passion for creating. Within a few weeks of arriving at UW he joined the Formula SAE race car team and learned Solidworks, finite element analysis software, and the use of machine tools like a child in a candy store. Eventually, advanced courses in finite element method, CNC machining, composites, acoustics, etc. gave him the tools and intuition he needed to become a more well-rounded engineer. However, he still preferred to develop intuition in these things and use the equations and formal analysis only when necessary. His senior project

tied together all these areas in designing and fabricating a carbon fiber acoustic guitar to sound like a conventional wooden guitar, with beautiful results.

Several research opportunities during his undergraduate years prepared Jonathan to define and explore complex research problems. These projects ranged from taking air quality measurements from a research airplane to designing and fabricating microfluidic valves with picoliters of displacement. His experience in the Genomation Lab at the UW taught him to appreciate the micro and nanoscale world. A summer internship at NASA's Jet Propulsion Lab in California opened his eyes to science that truly pushes the edge of human knowledge. A summer research exchange program in Germany taught him the finer points of precision engineering and provided an invaluable multi-cultural perspective on engineering.

A desire to prepare for a career of interesting, engaging, and new things drove Jonathan to immediately pursue graduate school after he complete his Bachelors in Mechanical Engineering. The search for a fertile environment at the intersection of computers and the physical world led him to pursue the doctoral program at Cornell University under Prof. Hod Lipson. The fruits of this work are presented here.

To my parents, for their support, patience, encouragement, and unwavering love. I would not be standing where I am today without them.

ACKNOWLEDGEMENTS

I would also like to extend my deepest gratitude to the following who have provided invaluable support during my time at Cornell:

To Hod Lipson for inspiration and guidance. For crafting an environment where I could learn and be productive with an astonishing lack of obstructions.

To Neil Gershenfeld for his original inspiration of "digital materials".

To current and former members of the Computational Synthesis Lab who have played an invaluable role in enabling the research I have accomplished as I implicitly and explicitly learned from them: Evan Malone, Victor Zykov, John Rieffel, Mike Schmidt.

To the aforementioned and others in the Computational Synthesis Lab who have created a wonderful, collaborative environment and played an invaluable role as a sounding board for ideas, thoughts, and directions: John Amend, Jeff Clune, David Hjelle, Jeff Lipton, Max Lobovsky, Rob MacCurdy, Jonas Neubert, Franz Niegel, Charlie Richter, Eric Schweikardt, Floris van Breugel.

To the friendly and helpful staff at Cornell University, in particular the staff of the Cornell Nanoscale Science and Technology Center as well as Marcia Sawyer who consistently goes above and beyond her call of duty.

To my family, both those of blood and in Spirit: My parents. The Doolings. The McCoys. The Dovis. Each have played a critical role in shaping my life and who I have become.

To my Lord and Saviour Jesus Christ, for being the meaning of it all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1   From Analog to Digital

The transition from analog to digital technologies has revolutionized many fields over the past century most notably communication and computation but physical objects have so far resisted a similar transition. Digital principles allow for perfect replication and zero noise despite using a noisy and inaccurate substrate. Though digital controllers govern the majority of manufacturing processes today, the fabricated objects are still inherently analog: Material can be added or removed anywhere, and every dimension has a finite error. Consequently, an analog fabrication system cannot make a part more accurate than its own positioning system, and performance degradation is inevitable in every subsequent replication. In contrast, many biological "fabrication" processes exploit digital assembly of fundamental building blocks (DNA, proteins, etc.), which repeatedly assemble into precise structures despite a very noisy environment and are able to reproduce without loss of accuracy over millions of generations. These principles extend to mechanical replication processes inspired by these biological processes [213], and may have profound implication to additive freeform fabrication as well.

This work explores a fundamentally new 3D freeform fabrication paradigm utilizing the efficient assembly of microscale units. Current additive manufacturing processes are the most direct analog equivalent to this. In comparison to these processes in which material is deposited or solidified in an inherent continuum, digital 3D printing imposes finite resolution: the size of a single unit.

1

Advantages of this 3D digital domain include high dimensional accuracy, perfect repeatability, deterministic error correction, and the inherent capability of low-temperature co-fabrication using a rich and diverse set of materials. These physical, self-aligning, fundamental units are hereafter referred to as "voxels", "volume elements", or simple "elements". Further definitions used throughout this text are as follows:

- **Voxel, volume element, element, micro brick:** A fundamental, physical, aligned building unit, constrained to a repeating unit of physical 3D space. Not necessarily cubic or space filling.

- **Digital material, digital object:** Matter composed of many such assembled voxels. Refers to the overall material or object, which is defined by the voxel topology, how the voxels are assembled, and the material each constituent voxel is constructed of.

- **Digital fabricator, rapid assembler:** An automated machine that efficiently assembles digital materials - a voxel printer. Note that although most conventional 3D printers use digital control and feedback systems, they are not capable of printing with physical voxels.

- **Analog object:** An object composed of continuous (non discretized) material. Includes the vast majority of objects made with currently established manufacturing processes.

- **Additive manufacturing, 3D printing, rapid prototyping:** Automated processes that build up analog objects layer by layer.

- **Freeform:** Having shape that incorporates complex curvature and/or arbitrary material distribution. Non-trivial to specify and create.

<div align="center">(a)   (b)</div>
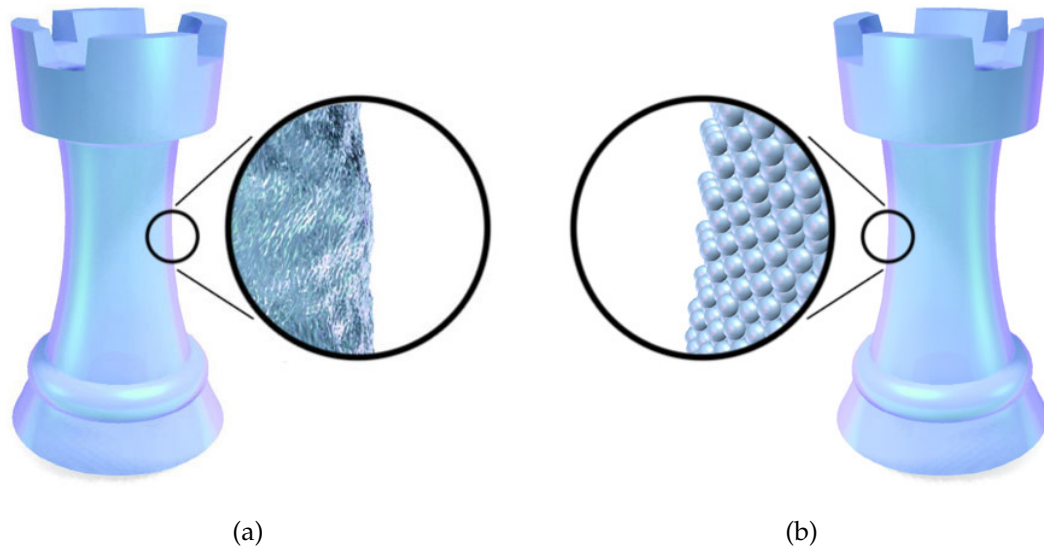
Figure 1.1: Traditional analog materials (a) are continuous in nature and every dimension has an associated finite error. Digital (discrete) materials are composed of fundamental, aligned building units (voxels) and the structure is defined perfectly by the presence or absence of a voxel at each location within the lattice.

The new paradigm of digital materials is illustrated in Figure 1.1. In the same way as an electronic digital signal is represented as a series of logical ones and zeros, a 3D object is represented completely by the presence or absence of a voxel at each location within its 3D lattice. When this abstract idea of digital materials becomes concrete as part of a physical voxel-based manufacturing process, a number of advantages are realized that roughly parallel those observed in the electronic digital domain.

Much like TTL (transistor-transistor logic) standardized the physical parameters (voltage and duration) of digital signals between semiconductor devices, there must be a standard or set of standards defining the interface between voxels in a digital material in order to facilitate widespread adoption. As with other digital technologies, a finite resolution must be chosen based on the needs of the

part. For example, to appear smooth to the eye at a reasonable distance voxels must be less than approximately 0.5mm. However, for other applications larger or smaller voxels may be more suitable, which points towards the necessity of a spectrum of well-defined voxel interfaces across multiple size scales.

1.2 Current State of the Art

Conventional additive manufacturing processes can fabricate 3D parts of varying precision, but typically of a single, homogeneous material [90, 211, 30]. Powder-based 3D printing processes use either a selectively deposited binder or scanning laser to melt and hold the raw powder material together. Because the parts are built in a full bed of powder of a single material, such processes cannot be easily adapted to create parts of multiple materials. Likewise, stereo lithography processes create the parts in a pool of a single resin type. Attempts have been made to flush and replace this bath between each layer with resins of varying solidified material properties, but at great expense and loss of efficiency.

Current fused deposition modeling (FDM) technology is capable of producing objects composed of multiple materials. However the range of materials that can be reliably melted and deposited has not led to widespread adoption of the inherent multi-material capabilities. Syringe-based paste extruding methods can co-fabricate objects with a greater difference in material properties, but in addition to a narrow range of rheological properties, the materials often have a limited working time and are limited in precision [116, 117]. Inkjet fabrication using multiple photo-curable materials has also been explored with great success, [18] and reliably creates high-quality parts with material ranging from
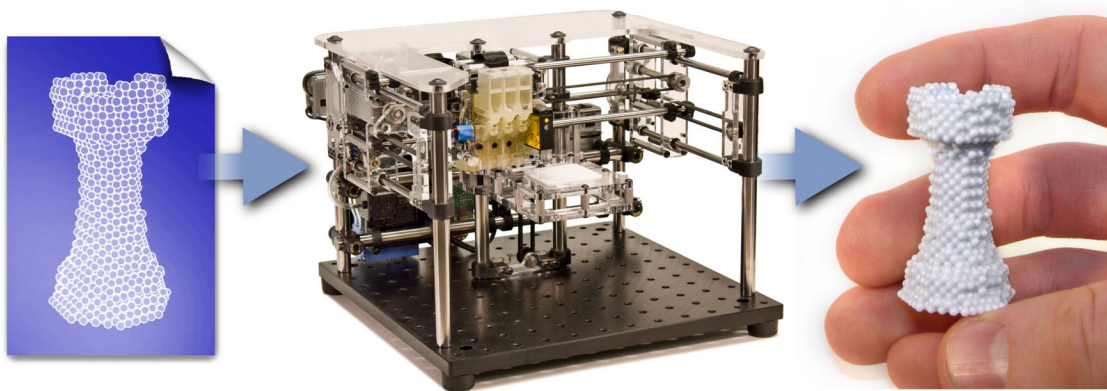
Figure 1.2: The principle of a digital manufacturing process, shown here using spherical voxels. A blueprint of a shape such as a rook is first specified (left), then a rapid assembler (center) combines voxels in the correct spatial orientation to make the physical object (right).

rubber to plastic. However, using inkjet technology places even more severe restrictions on the rheological properties of the deposited materials.

As a fundamentally different approach to 3D printing, Gershenfeld [57] recently proposed that thinking of fabrication as a digital (voxel based) rather than a continuous process can address these challenges (Figure 1.2). Although much work has been done in the area of voxels as data structures for computer graphics and CAD [88] and in additive manufacturing specifically [22, 115, 114, 104], here the challenges of creating a functional physical voxel printer and designing parts suited for such fabrication are addressed. The challenges and research associated with this are fundamentally different from the challenges and research taking place in the field of voxels as data structures.

Arbitrary-geometry (programmable) digital fabrication can take place as a top-down approach [144] or as a bottom-up approach [182]. In bottom-up manufacturing, sub-components arrange themselves into a more complex assembly.

In top-down manufacturing, an external fabricator directs assembly. There is currently a dichotomy between top-down and bottom-up manufacturing: Self assembly is advantageous for fabricating simple, regular, small scale structures, but does not scale to complex (non-regular) macroscopic structures [195]. Conversely, top-down manufacturing does not scale well to create objects with millions or billions of units.

Digital fabrication offers the opportunity to bridge this gap by allowing self assembly to be used for massively-parallel fabrication and alignment of relatively simple voxels, and then using top-down digital manufacturing to assemble complex macroscale structures out of the self-assembled voxels. For example, DNA self-assembly can be used to make spheres [113] or cubes [23], and other shapes [102]. These voxels, in turn, can be assembled in a top down way using a fabricator. Likewise, the self assembly of ordered lattices of voxels can be used to align lines or layers of voxels for selective deposition, reducing the continuous placement process to a digital selection process.

1.3   Physical Advantages of Digital Materials

Many advantages inherent to the digital domain have been observed in the field of digital electronics. Error correction and deterministic calculations rank high in the order of importance. These advantages translate into useful mechanical traits for a digital fabrication process, in addition to several others. These key advantages are listed below:

- **Accuracy:** By geometric design, the voxels will self align upon assembly so a fabricator need only place the voxel within a certain distance of its final position. Thus, the precision of the final part depends only on the tolerances of the voxels. This is analogous to a child with 1mm hand placement precision assembling LEGO^{TM} structures with $5\mu m$ precision. This results in an object more precise than the fabricator that created it.

- **Perfect repeatability:** Digital parts are perfectly repeatable with no loss of 3D information over subsequent replications.

- **Error scaling:** The overall precision of a digital object scales favorably as the number of voxels increases. Random dimensional errors of individual voxels tend to cancel out.

- **Multiple materials:** Since each voxel type is pre-manufactured independently, multiple materials may be combined in a single rapid assembly process. It is possible to combine materials such as high-melting point metals and low-melting point polymers whose initial processing properties are mutually incompatible.

- **Smart voxels:** Prefabricated voxels can also be pre-loaded with simple active components such as transistors, photovoltaics, microvalves, and other sensors and actuators thereby opening the door to fabrication of complex, functional integrated systems.

Disadvantages to digital manufacturing are analogous to the disadvantages of other digital technologies. Key disadvantages are:

- **A finite resolution:** Finite resolution leads to a loss of generality in the shape that can be fabricated. However, the resolution (voxel size) of digital materials may be chosen to fulfill the desired functionality. This is

analogous to choosing an appropriate bit-rate for digital music to sound continuous or determining a sufficiently small pixel size of a computer monitor to create smooth images.

- **Increased processing complexity:** Digital processing of material requires bit-by-bit addressing and therefore more complex machinery and slower processing time. This is again similar to digital signal processing that is generally more complex and significantly slower that analog signal processing.

1.4   Digital Manufacturing Technology

The process of transitioning from virtual voxels to physical voxels may be accomplished in several ways. The voxels regions may be fabricated in-situ, using either inkjetting technology to deposit liquid droplets that subsequently harden, or a DLP system to selectively harden pixels within a layer of homogeneous photo-curable material. In-situ fabricated voxels are often quick and economical to fabricate with, but they do not inherently retain their digital (discrete) nature upon instantiation, are limited to very specific subsets of materials, and do not display the precision characteristics presented here. The prefabricated voxels used in physical 3D voxel printing, although requiring a more involved fabrication process, remain discrete and may be fabricated of any solid material.

The technology to create and assemble large-scale parts made of digital materials is in place and needs only to be reduced to practice. The main technical hurdle is to deterministically assemble millions or billions of voxels. In order to accomplish this in a reasonable amount of time, the assembly process itself

must be massively parallel. There are varying degrees of parallelism to assemble digital materials. A serial assembly process would place one voxel at a time. A 1D parallel process would place a line of voxels simultaneously, a 2D parallel process would place an entire layer, and a 3D parallel process would form the entire digital part at once. Adding each dimension of parallelism results in vastly reduced assembly times as the process scales up to large numbers of voxels. Since no technology exists to do 3D parallel assembly, only the 2D and below are considered.

1.4.1   2D Voxel Manipulation

In order to manipulate an arbitrary 2D array of voxels, an $m \times n$ print head must be capable of attracting or repelling voxels at any of its individual locations. This requires individually addressable cells capable of producing a binary force (on or off). This force could be pneumatic, electrostatic, magnetic, surface tension, or any number of others depending on the scale and design of the voxels. The minimum number of control lines to individually address each cell scales favorably as the print head size increases using multiplexing, but this requires a force that can be autonomously maintained in either the on or off state when not being addressed. This rapidly leads to scaling problems when the number of voxels in a layer becomes large. Thus, in order for 2D parallel assembly process to scale favorably, the voxel holding cells should themselves be capable of parallel fabrication.

The problem of individually addressing arbitrary locations within a large 2D matrix has been reduced to common technology with the advent of the personal

computer. The output of a computer graphics card can address more than 20 million pixels at 16 bits and 60Hz, which is more than sufficient to process 20 megapixel layers at once. Assuming a cubic build space, this is sufficient to enable an 80 Gigavoxel digital fabricator. The challenge is turning the digital output into physical forces to manipulate voxels. By coupling the computer output to a monitor or projector, the optical images can be used as an input for massively parallel manipulation of 2D fields [27]. Optics can scale this output to any resolution and manipulate physical forces by selectively evaporating fluid or discharging photosensitive materials.

1.4.2   1D Voxel Manipulation

Manipulating an arbitrary 1D array of voxels is dimensionally easier than the 2D equivalent. Because a digital object is still printed layer-by-layer, this frees up a dimension to be utilized in the fabrication process. Because of the lower dimensionality, the ability to manufacture the deposition cells or modules becomes much less constrained. The extra dimension can be used to route mechanical and electrical information without the packaging difficulties of the 2D parallel processes. Individual cells can be more easily discretely addressed without necessarily resorting to multiplexing because there are far fewer of them. For instance, this allows electrostatic cells to be individually charged or discharged using traditional control circuitry.

However, individually picking up and placing each line of voxels is not necessarily the best way to approach a 1D parallel voxel printer. If the voxels are stored in a reservoir on the moving print head, voxels can be deposited on de-

mand where they are needed as the print head scans continuously across each layer, This greatly reduces the amount of movement and time involved in a given build. Depending on the scale of voxels used, the methods of selectively depositing the voxels may vary significantly. For millimeter scale voxels, gravity and mechanical valving are sufficient. At smaller scales, electric or magnetic fields can manipulate the voxels more efficiently. Regardless, continuously scanning 1D voxel manipulation holds promise for efficiently depositing voxels even as the number of voxels scales to millions or billions.

1.4.3   Serial Voxel Manipulation

The simplest way of building a digital object involves deterministically placing each voxel one voxel at a time. High performance pick-and-place robots with specialized grippers could easily be adapted to a digital fabrication task, but even if a voxel could be placed in a fraction of a second it would still take years to make high resolution digital objects. Placing one voxel at a time simply does not scale to building objects of millions or billions of components, and is therefore considered only as a baseline in this work.

1.5   The Digital Design Process

This new paradigm in physical manufacturing requires a new paradigm in the design process to synthesize digital objects. Although digital materials can certainly re-create conventionally manufactured objects, albeit at a finite resolution, the potential design space is exponentially larger. With the ability to place any

material and any function at any location in an object comes a staggeringly large design space.

The key to fully leveraging the capabilities of digital fabrication processes lies in design optimization and design automation algorithms. Design optimization is by far the easier of the two. Given an initial design with specified functions to optimize, an algorithm can easily explore incremental changes in the design and compare them to the original to determine if they are better or worse. The initial design may not even be necessary in some domains. For instance, in the case of a 3D circuit the high-level electrical schematic could be input and a 3D layout automatically generated. This would fulfill a similar function as current 2D PCB layout programs.

The design process becomes more interesting as design requirement in multiple physical domains come in to play. For instance, designing a robot to be made of digital materials requires the interaction of electrical and mechanical components. Not only does the electrical information need to be correctly processed and routed, there also must be a mechanical shape and actuator layout that enables motion in the physical world. Although either of these could be prescribed by the engineer and the other optimized, this does not scale well to many multiple domains with similarly inter-related functionality. Here, it is desirable to simply input a high level goal, such as a robot that locomotes, and let a design automation algorithm generate an optimal blueprint.

A critical piece of such a design automation algorithm is a virtual physical simulator that allows potential designs to be simulates with a high degree of correspondence between the simulated result and an eventual physical result. A whole new class of multi-physics simulators becomes necessary. Just like

current circuit design programs make use of well-established models for the behavior of components, accurate multi-physics models of each voxel type are needed to plug in to such a simulation framework. With this in place, computers can much more efficiently explore the huge design space enable by digital materials and help generate novel and efficient designs.

1.6 Potential Applications of Digital Materials

The ability to assemble an arbitrary structure out of a large number of voxels is a powerful tool. Besides fabricating complex and accurate geometries, in principle digital manufacturing allows for inclusion of any material that is rigid at the time of assembly. More generally, a voxel could be any fundamental element that can be picked up and laid down according to an electronic blueprint. It is here that digital manufacturing enables applications not possible with current additive manufacturing processes and introduces new possibilities to the microscale community as well. The flexibility of voxels as fabrication building blocks enables going beyond the creation of just passive materials with the inclusion of "smart voxels." Several potential areas of impact are listed here:

Electrical networks: Digital materials could be used to make extremely compact, integrated 3D electrical networks and micro-robots. With only conductive and non-conductive voxels, arbitrary 3D electrical networks can be made. Adding a small library of resistor, capacitor, transistor, and other electrical component voxels would enable compact custom 3D integrated circuits [60]. With logarithmic spacing of the values of these components, almost any desired value could be reached with a small number of components. By including piezoelec-

tric or shape memory alloy voxels for sensing and actuation, and solar or battery voxels for power, all the components to create microrobots [154] are in place.

Fluidic networks: A small library of voxels with microfluidic functionality could be developed to enable 3D integrated microfluidic circuits for chemical and biological uses. In fact, only two voxel types are needed to create arbitrary 3D fluidic networks (Figure 1.3). Compatible valving elements and sensing elements would allow not only quick fabrication of 3D microfluidics, but eliminate the high overhead and the difficulty of aligning individual layers in traditional microfabrication facilities.

Tissue engineering: Biological materials shaped into appropriate voxels could permit fabrication of heterogeneous tissue on demand for tissue engineering applications [43, 82, 126]. Small tissue spheroids could be incubated in near-optimal conditions with different types of cells from a patient. These voxels could then be rapidly assembled into the shape of the desired organ of body part including internal circulation channels, bio-markers, etc. and allowed to grow together to form a custom organ or tissue fragment.

Photonics: Digital materials may also benefit those at the forefront of photonics research. Currently, there are many simulations of 3D optical circuits that would usher in a new era of computation, but there is no way to readily produce them [105]. In general, optical circuits are constructed by arbitrary regular placement of high and low optical index elements within a larger matrix [185]. Even now, voxels (order $1000\mu m$) could be used to verify these properties with microwaves, and as the scale of voxels approaches the wavelength of visible light (order $0.5\mu m$), digital manufacturing will provide unprecedented ability to create optical circuits. (Figure 1.3)
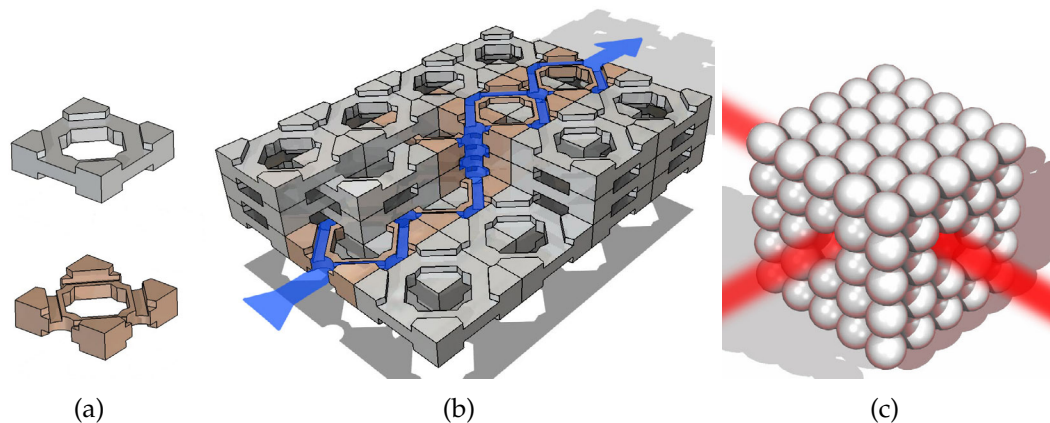
Figure 1.3: Examples of advanced digital structures illustrate possible future applications of digital materials. Two types of tiles could be used used for printable arbitrary 3D microfluidic fluidic networks (Left). Photonic crystals also are possible if the elements are a suitable scale.

1.7   Related Work

As a fundamentally new area of scientific exploration, there is very little established research in the area of digital materials. Several pre-existing systems such as LEGOs or modular robots could be considered a form of digital materials, but the scale, goals and aims are significantly different than those presented here.

The conceptual idea of digital materials originates with MIT visionary Neil Gershenfeld [57]. The idea of matter transitioning from analog to digital is an intriguing idea echoed by other technological visionaries such as Wil McCarthy [122]. Concepts have been published for such a digital fabricator and one specific voxel type known as a "GIK" (Great Invention Kit) was proposed and analyzed [144]. However, the concept of printing 3D objects from discrete, aligned fundamental building units has not yet been physically demonstrated except in the work presented here.

Leaving the idealogical definition of digital materials behind, there are also several existing processes which can create objects with voxels of multiple materials in specified 3D spatial patterns. Objet Geometries [135] is an Israeli additive manufacturing company that pioneered polyjet material deposition. This technology uses a plurality of inkjet heads to deposit photo-curable resins in 16 micron layers. In addition to support material, this process can co-fabricate hard and soft acrylic-based materials in any relative spatial orientation.

Recently, a startup company EoPlex began advertising a layered manufacturing process called High Volume Print Forming (HVPF) that can co-fabricate metals and ceramics [20]. It has not currently been demonstrated for freeform structures and in fact is more similar to semiconductor processing than additive manufacturing. Up to 8 different materials have been co-fabricated with specified microstructures. Applications include metamaterial antennas and electronics packaging.

These analog voxel processes result in objects that may be functionally similar to those created in digital fabrication processes, but do not incorporate the favorable error scaling, error correction, or recyclability of digital materials. Polyjet technology can make highly detailed models, but is limited only to photo curable materials which precludes functional elements and any non-acrylic based materials. Likewise for HVPF, each material in a given build must have a similar sintering temperature, which prohibits the ability to combine high and low melting point materials.

Surprisingly the artistic community has come the closest to automated fabrication of digital materials. In an art installation called "Geometric Death Frequency" by Frederico Diaz [40], 420,000 balls were assembled and glued by a

pair of industrial robot arms. Although these balls were not limited to a lattice, the majority were in fact ordered in this way. However, only small sections of the art installation were fabricated autonomously, and these were then transported and assembled on site to complete the structure. Although not officially published, it can be deduced that the robot arms would have had to run continuously 24 hours a day for at least two months to assemble all the necessary balls.

Very little work has been done studying the macroscale properties of many physical discrete, connected microscale voxels. The closest work lies in the realm of simulating atomic structures and polymers [19], but the interactions between molecules is fundamentally different than the interactions between discrete physical voxels. The individual interactions between GIKs have been characterized for parameters such as assembly angle and assembly/disassembly cycles [144], but have not been characterized for the purposes of large scale mechanical simulation.

In the area of design automation, general techniques have been widely used to optimize pre-existing parametric designs. In contrast, freeform design automation does not assume a pre-existing design. The automated design of static structures in 2D and 3D has been well explored using homogenization techniques [7, 12, 42]. These methods have been extended to 2D compliant structures [133], but are still based on optimizing the sum of a single local parameter common to each pixel of voxel.

There are many design automation problems that cannot be well solved using such techniques, such as designing a cantilever beam to deform in a specific shape or a soft locomoting robot as proposed here. These require evolutionary

algorithms which are also widely used for an astounding variety of problems. Although these have been previously applied to structural problems [180], these have been in the single material domain. there is no known work regarding 3D freeform multi-material design optimization.

1.8   A Vision of the Future

In the future, one can envision a standardized library of voxels that all have compatible geometry, in much the same way as transistor-transistor logic (TTL) standardized the interface between digital circuit components in the 1960s. These voxels will be mass produced in high volumes, so that millions of voxels may be purchased for several dollars. Voxels would be manufactured several different ways, depending on the geometry, material and size. For instance, interlocking voxels may be produced at a microscale using multilayer photolithography techniques. Using a 300mm wafer, a single wafer could produce more than 150 million $20\mu m$ voxels. Producing microscale spheres of many different materials is also well understood, and a wide variety are commercially available. However, the voxel manufacturing techniques are highly specialized processes. Thus, it will be most economical to mass produce the voxels at central facilities, then distribute them as a pseudo raw material for assembly on location. This concept of central manufacturing and distributed assembly is evident in products ranging from LEGOs^{TM} to modular structural components. As long as the function of each voxel is elementary, there will be a finite (and likely small) number of voxel types required to build arbitrarily complex objects.

The end user would order voxels of different materials and functions which

could be loaded into a digital fabricator sitting on a desktop. Then, plans for a glucose sensor, an educational micro-robot, a portable music player, or any number of other products could be downloaded from the internet and fabricated at will. When the object is no longer needed or a better blueprint becomes available, the object can be decomposed into its elemental voxels which can then be sorted and re-used. Advances to digital material technology will likely come in the form of increasingly complex and efficient smart voxels, with new interface standards being developed every few years to meet the needs of current technology.

The first steps in enabling this global economy of voxels are presented in this work.

Part I

Physical Voxel Designs and

Manufacturing

CHAPTER 2

VOXEL SHAPE AND DESIGN

2.1 Considerations for Voxel Designs

The abstract concept of digital materials begins to become concrete when considering the physical design of the voxels to be used. In this chapter favorable physical shape of the passive fundamental voxel is explored without consideration of functional "smart" voxels and their requisite information interfaces. The only conceptual constraints on the voxel shape is that they must occupy and fit together within a regular three-dimensional lattice. However, additional constraints are imposed to facilitate their construction and assembly in the physical world. In order to fabricate useful, robust devices in a massively parallel assembly operation, the shape voxels used in digital materials should maximize the following properties:

1. Passively self-align in $\mathbb{R}^3$ relative to each immediate neighbor upon addition to the assembly

2. Be invariant to rotation and flip to easily manipulate and align voxels in parallel

3. Rigidly connect to neighbors to form a structure that stays together

4. Fully tessellate and fill $\mathbb{R}^3$ space to allow for fabrication of fully dense solids

5. Mechanically robust for superior aggregate material properties, especially in tension

6. Simple geometry for cost-effective manufacturing in large quantities using a variety of methods

Of these properties, only passive self-alignment is critical to the fundamental nature of digital materials. Each voxel must be capable of passive self alignment in $\mathbb{R}^3$ relative to the voxels around it upon assembly. This ensures that a regular lattice with well-defined interfaces is preserved throughout the structure, enabling the discrete (digital) nature of the assembly process that would not be possible if continuous errors in position between adjacent bricks were possible.

A trade off exists between the number of redundant orientations of a voxel and the geometric complexity. If each voxel must be actively aligned to a single correct orientation before assembly, there will be a large penalty in assembly complexity and efficiency. This problem can be mitigated in one of two ways. First, alignment can be maintained throughout the voxel manufacturing, distributing, and assembly process. Although some voxel manufacturing processes may allow the rotation and orientation to be maintained throughout the fabrication and assembly process [208], there are many voxel fabrication processes where this would be prohibitive. For instance, when the substrate on which the voxels are manufactured on is costly and reusable, each micro voxel would need to be transferred to an intermediate substrate for distribution while maintaining absolute orientations.

Alternatively, to better meet the goals of a mass fabrication process the micro voxel geometry could be rotation and flip invariant. This allows the voxels to be manufactured, stored and transported without regard to absolute orientation, then passively aligned in parallel for assembly. Because any possible incorrect brick orientations are eliminated by design, the assembly process becomes

significantly more simple and robust.

In order to create usable objects, adjacent voxels must be rigidly connected. Physically interlocking bricks are generally desirable over gluing or otherwise adhering adjacent bricks for a several reasons. First, this decouples the material properties of the voxel from the assembly process. This allows voxels to be composed of any solid material, even materials that are mutually incompatible in any given fabrication process. Second, the process is reversible which allows for eventual re-use and recycling of the individual components.

The shape design of the individual voxels will have a large effect on the aggregate material properties of an object composed of many such voxels. Although a block of bricks made of a given material will never be as strong as a homogeneous block of the same material in the same dimensions, the differences between the two can be minimized by a directed choice of geometry. In particular, stress concentrations should be minimized in tension loading scenarios, which could otherwise be detrimental to the robustness of a digital object.

Additionally, it is desirable for the most demanding structural applications that the voxels fully tessellate 3D space such that voids are minimized. Among other favorable properties, this provides the maximum compressive strength, which should approach that of a homogeneous piece of the constituent material. With such a system, voids may still be introduced by leaving empty locations in the lattice in applications where weight savings or porosity is desirable. Well known artists such as M.C. Escher have explored many freeform 2D fully tessellating and space-filling shapes. While there are many more possibilities in 3D, the constraints on manufacturing and assembly complexity constrain the potential space of shapes significantly.

Perhaps the most important trade off in voxel design is between the geometric complexity of each voxel and the complexity of mechanically connecting adjacent voxel. On one hand, the geometry must be as simple as possible to facilitate ease of mass fabrication, yet the shapes must rigidly interlock in three dimensions to create structures. This is critical to a successful micro voxel architecture regarding the scaling of structures from the tens or thousands of voxels presented here to objects composed of millions, billions, or more micro bricks.

It quickly becomes apparent that not only will massively parallel assembly methods be required, but also massively parallel micro voxel manufacturing methods. Ideally, the geometry would be naturally suitable for fabrication processes such as micromachining [46], hot embossing [6, 86], micro casting[103], or additive layered manufacturing [53, 121]. All of these processes leverage reusable complexity, in which a single complex template with many individual brick templates is fabricated which can then used to easily and repeatedly fabricate many bricks in parallel. In order to facilitate all of these manufacturing processes, there should be no internal or undercut surfaces that would prevent having a single parting plane for a hypothetical molding process.

2.2   Classes of Physical Voxels

To begin the discussion of selecting a brick geometry, the space-filling requirement presented above are first broadly considered. There are a finite number of tessellation modes that fill three dimensional space. Of these, there are 3D tessellating shapes based on platonic solids and 2.5D extrusions of 2D shapes that also fill 3D space. Although cubes are the only 3D platonic solid that tiles

directly in euclidean space [38], truncating tetrahedra, and octahedra can also fill euclidean space. 2.5D tessellating shapes such as equilateral triangles, rectangles, and hexagons tessellate in 2D to form a complete layer, and then subsequent layers may be stacked to achieve a 3D lattice.

For both the 3D and 2.5D tiling shapes, geometry must be added at some or all of the interfaces to enable interlocking. If care is taken, this can be done with no loss of possible brick orientations, thereby maximizing the rotation and flip invariance. If the layers are suitably offset laterally from each other in the layer-based 2.5D schemes, a rigid structure can be created by interlocking only between bricks of adjacent layers, and not between adjacent bricks in the same layer. This allows a much simpler design while also allowing bricks within a given layer to be assembled in any order, even simultaneously in a parallel assembly process. Non space-filling designs may also be desirable in certain applications. Although largely a subset of the space-filling designs, this category is be considered separately.

2.2.1   2.5D Voxels

There are two basic shapes that fully tessellate $\mathbb{R}^2$: the equilateral triangle and the rectangle. Combining two equilateral triangles yields a diamond and three diamonds creates a hexagon. Many other irregular 2D tessellating shapes exist, such as those found in the art of MC Escher, but are not considered due to the lack of rotational symmetry which makes passive alignment difficult. The 2.5D voxels can create rigid 3D structures while interlocking only with the layers above and below, but each layer must be offset by a carefully chosen amount.
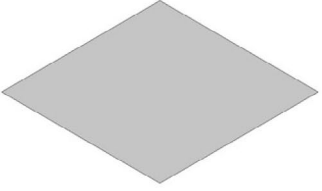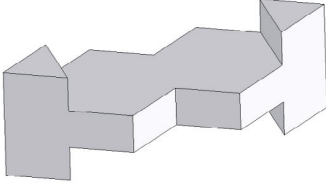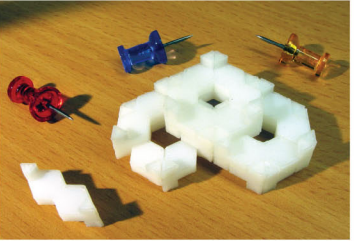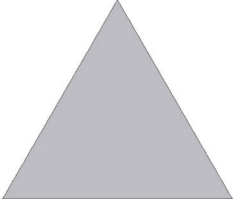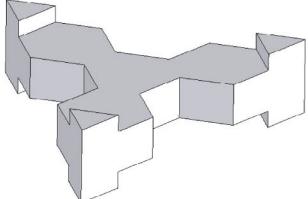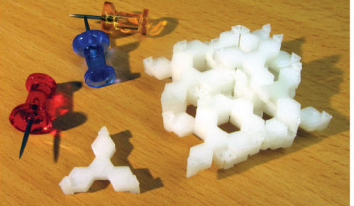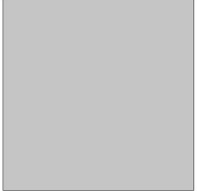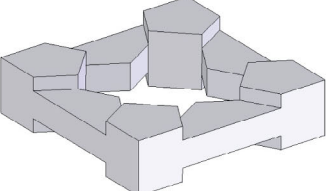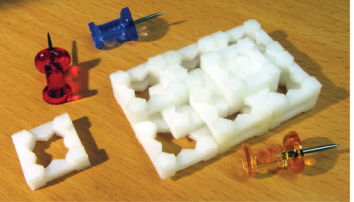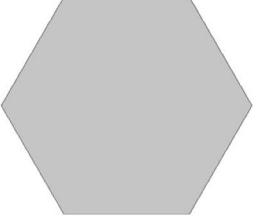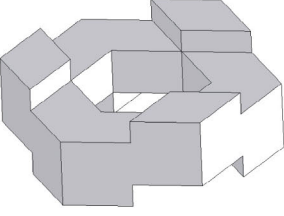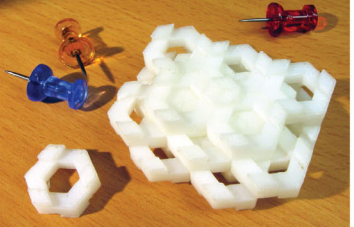
Basic Shape	Interlocking Geometry	Assembled Material

Figure 2.1: Interlocking, self aligning 2.5D voxel suitable for digital materials were designed and physically verified based on 2D diamond, triangle, square, and hexagon shapes. Thumb tacks are shown for scale.

Although 2.5D voxels only interlock in the vertical direction, once two layers are in place they form a rigid structure with positive constraint in the lateral directions. This allows the digital material to be stressed in tension in two directions and compression in all three without fear of disassembly. Tensile behavior in the vertical dimension will be dominated by the friction forces holding the tiles together. Additionally, since the individual voxels can be fabricated with three stacked and bonded layers, conventional multilayer photolithography [205] or other layer-bonding techniques can be used to make large numbers of microscale voxels in a batch process. The lack of in-layer interlocking allows an entire layer of voxels to be passively aligned and selectively placed at once, which is parallel in two dimensions. However, diamonds can tessellate $\mathbb{R}^2$ in two unique ways, and the hexagonal design has only 3 degrees of rotational symmetry out of 6 possible orientations, both of which may lead to problems with passive alignment.

2.2.2 3D Voxels

There are three distinct topologies that fully tessellate $\mathbb{R}^3$ and can physically interlock with no loss of possible orientations. These are the rectangular prism, a truncated tetrahedron and a truncated octahedron. Other more complex $\mathbb{R}^3$ tessellating solids, such as the rhombic dodecahedron exist, but are not considered here. The rectangular prism is by far the most familiar, and the cube will be considered as the most general case. Neither tetrahedra nor octahedra completely tessellate $\mathbb{R}^3$, but unique truncations of the corners yield this desired property, albeit with rather complex and unfamiliar shapes. In all three cases, the geometry can be designed to physically interlock into a rigid structure with no loss of
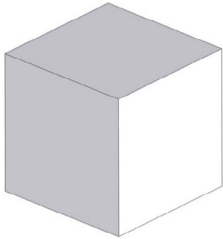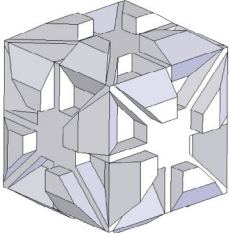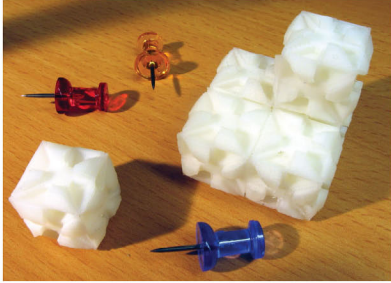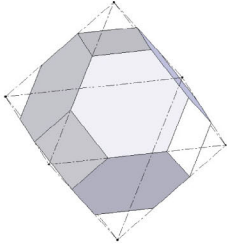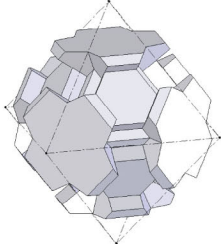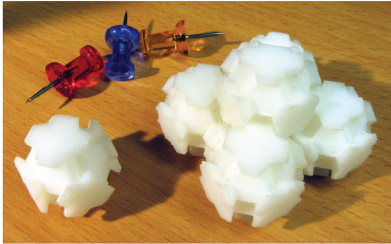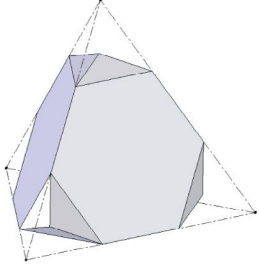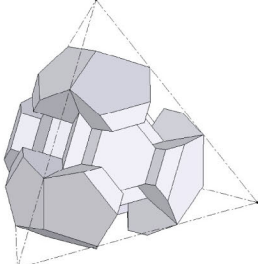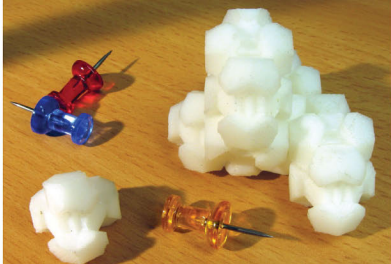
Basic Shape	Interlocking Geometry	Assembled Material

Figure 2.2: Interlocking, space-filling 3D voxel suitable for digital materials were designed an verified from rectangular prisms and truncated tetrahedral and octahedral 3D shapes. Thumb tacks are shown for scale.

possible orientations. However, the truncated octahedra has more interfaces (8) than orthogonal dimensions to assemble from (6), thereby limiting its ability to assemble in fully populated 3D lattices.

Voxels with 3D rotational symmetry interlock in three dimensions and thus have greater alignment redundancy. However, there are several aspects which make them impractical for large-scale digital manufacturing processes. First, by the very nature of interlocking in 3D, they must be actively assembled in

all three dimensions. This process would be a challenging for an automated machine compared to the one dimension of active assembly for the 2.5D voxels. Additionally, there currently is no cost effective way to manufacture large numbers of these voxels at the micro-scale with the necessary tolerances for the friction-fitting interfaces.

2.2.3   Non Space-Filling Voxels

In some applications, it is advantageous to use sparse materials. For instance, when weight or density is critical, greater compliance is needed, or fluids/gases must penetrate the structure. One solution is to place sacrificial voxels in a regular pattern throughout a structure which are later removed. If the lattice is too dense to allow efficient removal of the support material, leaving voxel voids during the fabrication process is compatible with all the geometries presented here, so long as they border a sufficient number of realized voxels.

Another technique to create sparse digital materials is to remove non-critical material from the 2.5D and 3D voxels. Thus, voids can be selectively introduced while preserving the geometric interlocking properties. This could be carried out such that the voids are isolated, (analogous to closed-cell foam) or interconnected to allow fluid flow in specific directions [128]. This opens up possibilities in the area of tissue engineering, where factors such as pore size and mechanical properties must be carefully controlled [69].

Here, several other exclusively non space-filling voxel architectures are presented as possible compromises between mechanical properties and ease of manufacturing individual voxels.

Spherical Voxels

Although close-packed spheres do not fill $\mathbb{R}^3$, they occupy a majority of the space (74% by volume). Spheres also have other practical advantages which cannot be ignored. The region of self alignment is the largest of any of the voxel shapes at approximately $1/2$ the diameter. Passive alignment of spherical voxels within each layer is trivial since rotation is irrelevant. However, an obvious drawback is the lack of geometric interlocking. This necessitates either adhesive applied during the build process or an additional post-processing step to bind the build materials and remove the sacrificial material, either mechanically, chemically, or thermally.

However, spherical voxels are especially attractive because they are relatively easy to manufacture in bulk and are readily available for a large range of materials such as steel, aluminum, copper, delrin, and acrylic. Commercial uses for these millimeter scale spheres include ball bearings and automated braille sign machines. Practically, this means that the manufacturing infrastructure is already in place to fabricate many spheres at a relatively low cost. Moreover, as higher resolution digital fabrication processes are considered, a number of techniques are currently available for fabricating high precision micro and nanoscale spheres [93].

"GIKs"

There are also voxel designs which are exclusively non space-filling, such as the GIKs (Great Invention Kit) (Figure 2.3) investigated by Popescu [144]. These are designed to geometrically interlock using 2D shapes which are simple to fabri-
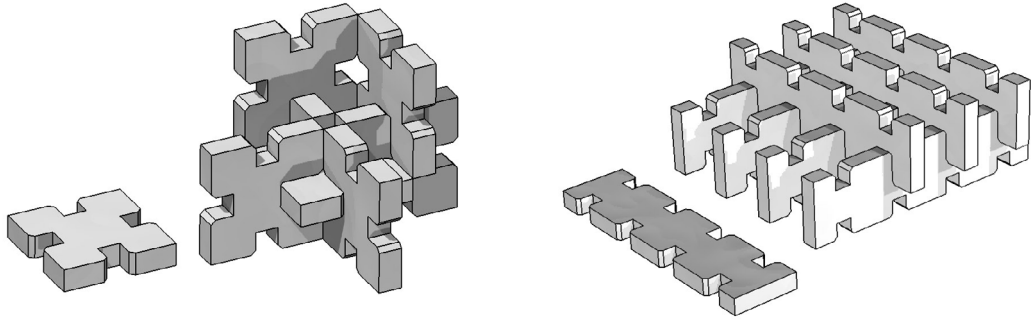
Figure 2.3: Illustration of sparse digital matter (GIKS) shows how simple 2D shapes can be assembled into 3D sparse structures (Adapted from [144]).

cate. However, in order to interlock purely 2D shapes, an extra dimension of complexity must incorporated into the assembly process. In this case, a hypothetical printing process must assemble GIKS in two separate orientations.

Two-Phase Digital Materials

Digital voxels may also consist of multiple subcomponents, allowing for more simple voxel fabrication and potentially richer design space at the expense of assembly effort. For example, all the basic 2D tessellating shapes presented in the 2.5D section can be kept as a strictly 2D layer and joined to adjacent layers with pin segments (Figure 2.4). Although two varieties of parts are now needed, the fabrication of each is proportionately less complex and the assembly process is potentially simpler as well.
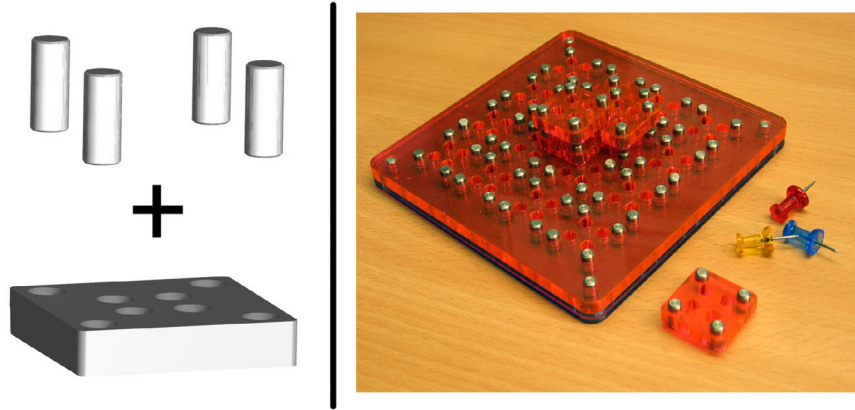
Figure 2.4: Multi-phase digital material consisting of 2D square tiles and cylindrical pins are functionally similar to the 2.5D square tile voxels but with potentially simpler fabrication. Thumbtacks are shown for scale.

2.3 Analysis of Potential Voxel Designs

It becomes clear that choosing a voxel design to pursue involves inevitable trade-offs between several relevant design variables and properties. Qualitative analysis was performed in the form of a weighted design matrix on these different voxel designs to summarize their strengths and weaknesses. They were evaluated on the basis of six different categories:

- **Self Alignment:** How large is the region of self alignment in comparison to the size of the voxel?

- **Rotation/flip invariance:** How difficult is it to passively align a layer of voxels for placement?

- **Interlocking:** Do the voxels physically interlock to form a 3D structure?

- **Space Filling:** How much of $\mathbb{R}^3$ can the voxels fill?

- **Manufacturability:** How easy is it to create millions of microscale voxels, based on current manufacturing technology?

Table 2.1: Qualitative analysis shows that spherical and square-tile based voxel designs are most suitable for a massively parallel top-down assembly process. Fully 3D designs are difficult to manufacture and assemble efficiently.

Class	Base	Self alignment (10)	Rotation/flip invariance (20)	Inter- locking (5)	Space Filling (5)	Manufactur- ability (25)	Assembly complexity (35)	Total (100)
2.5D	Diamond	Poor	Poor	Yes	>95%	Satisfactory	Poor	36
	Triangle	Poor	Satisfactory	Yes	100%	Satisfactory	Satisfactory	52
	Square	Poor	Good	Yes	100%	Good	Good	**76**
	Hexagon	Poor	Satisfactory	Yes	100%	Good	Good	70
3D	Cube	Poor	Good	Yes	100%	Poor	Poor	40
	Octahedra	Poor	Satisfactory	Yes	100%	Poor	Very Poor	27
	Tetrahedron	Poor	Satisfactory	Yes	100%	Poor	Very Poor	27
Spheres		Very Good	Very Good	No	79%	Very Good	Very Good	**94**
GIKs		Good	Good	Yes	<63%	Very Good	Poor	64
Multi-phase		Good	Good	Yes	100%	Good	Poor	61

- **Assembly complexity:** How many dimensions/degrees of freedom are needed for an automated digital fabricator to assemble objects?

Subjective categories were evaluated on the basis of Very Poor (0), Poor (2), Satisfactory (5), Good (8) and Very Good (10). Weights for each category were chosen according to intuition of how important each aspect will be in creating a useful digital fabricator. Spherical voxels had the highest utility (Table 2.1) primarily due to the ease of assembly, simple manufacturing process, and infinite rotation and flip invariance. Of the designs that interlock, square and hexagon tile-based designs are the most promising because they can be manufactured easily using existing multilayer fabrication processes and can easily be assembled layer by layer.

Of the voxel designs presented here, two of the most promising designs (highest utility) for a massively parallel assembly process were selected for continued experiments and analysis. Spherical voxels were selected for use in demonstrating the concepts of physical voxel fabrication for a number of reasons. First, every orientation is redundant, which eases the challenges of aligning voxels for assembly. Second, spheres can be purchased in bulk economically with very high accuracy which eliminates the need to fabricate thousands of specialized voxels. This comes at the expense of needing to bind the spheres together using glue or other methods to make an object which can be handled.

The 2.5D square tile was selected for simulations and experiments which required interlocking voxel shapes. In the future, physical voxel fabrication platforms will move to interlocking voxels as the additional design factors introduced by alignment are addressed. Although the square tiles only positively interlock within the horizontal plane, the geometry and friction of pressing them together in the vertical axis provides a sufficiently robust connection.

2.4 Optimization of Square Tile Shape

The geometry of the square tile design (Figure 2.5) was optimized to minimizes both the geometrical complexity of the tile and the internal stress concentrations as the tile is physically loaded. The chosen micro voxel design has eight redundantly identical orientations and no "partially correct" orientations that could cause problems in a passive alignment process. The geometry consists of a ring structure into which four feet of the four adjoining bricks in the next layer press in to. Each foot sticks up approximately 1/4 of the total height H (Figure 2.5(a)).
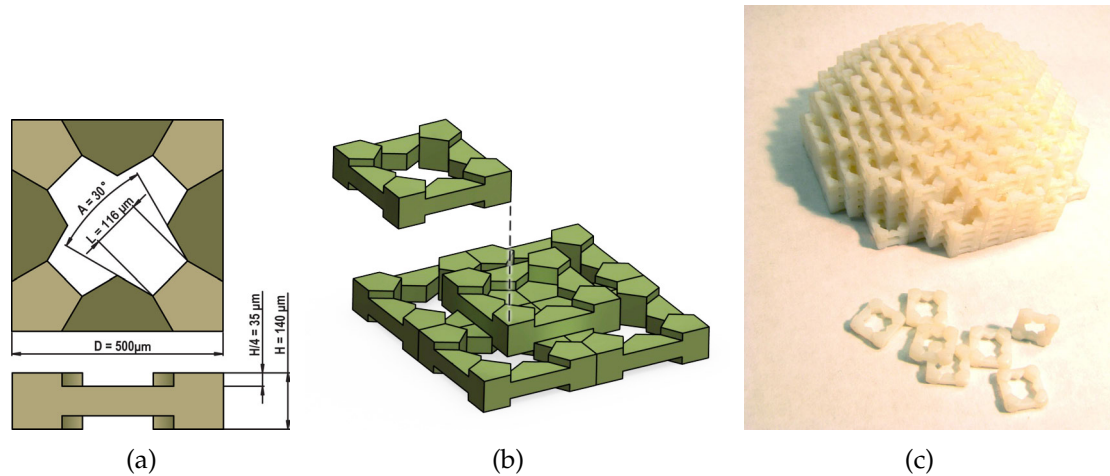
Figure 2.5: The selected square tile design (a) is shown with optimized parameters at $500\mu m$ sizing. The space-filling tiles interlock by offsetting each layer $1/2$ a tile dimension in X and Y (b). Millimeter-scale mock ups assembled into a hemisphere (c) demonstrate a large-scale assembly.

Thus, when feet are inserted into a given brick from both above and below, they touch in the middle, yielding a space filling design.

Given the constraint of filling 3D space, there were four parameters to vary: The tile edge dimension (D), the tile height (H), the tab width (L), the tab interface angle (A). All other dimensions are derived from these to satisfy the space-filling criterion. For different experiments, D and H are prescribed based on the desired scale of the voxels. L and A are optimized once here and these values are scaled and used for all other sizes.

2.4.1 Compactness vs. Strength

The other two free parameters dictate the size and shape of the tab feet, and by extension the size and shape of the ring geometry. The tab width L was selected

to be D/4 as a starting point. Then the voxel geometry was analyzed to determine the optimal tab interface angle A. There are many arbitrary metrics that could be used to decide the best geometry. Here, a combination of two metrics was chosen to capture information about the geometric complexity of the shape along with an estimate how strong a structure composed of such micro voxels would be.

The first metric was a measure of the compactness of a 3D shape. Here compactness is defined as the total surface area of the brick divided by the total volume of the brick. In this case the total volume of the brick was constant due to the space filling constraints. Essentially, varying the tab dimensions adds and subtracts equal quantities of material at different locations of the brick shape. Bricks with higher surface area are less compact, and thus less desirable.

The second metric for each candidate brick geometry was obtained by performing finite element analysis to approximate the tile's behavior in a lattice of tiles under tensile load. The axial tension scenario was selected as a worst-case scenario because the space filling nature of the bricks will give them favorable compressive properties as compared to tensile properties. The tiles were loaded as shown in Figure 2.6(a), which is a good approximation of the forces a tile will undergo within a structure under axial tension. One interface surface was fixed while the other had unit force applied in the outwards direction. A linear elastic simulation was used with a mesh of approximately 230,000 elements. A sample output of the simulation is shown in Figure 2.6(b). The maximum stress was output and normalized by the stiffness of the material used by the simulation to obtain a dimensionless qualitative indicator of the strength of each geometry. Due to the variation of meshes between geometries and the sensitivity of finite

<div style="text-align:center">(a)   (b)</div>
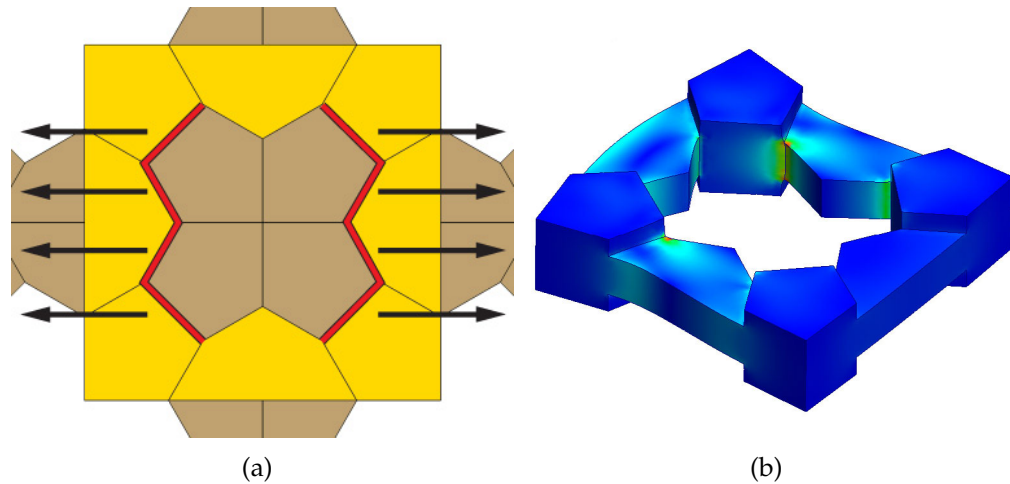
Figure 2.6: The voxels were loaded in simulation in a manner consistent with the forces an individual tile would undergo when a larger structure is subjected to an axial tension load (a). An example output of the finite element simulation shows the exaggerated deformation and stress concentrations resulting from the loading.

element simulations to the mesh structure at sharp corners of a stress concentration, a line of best fit was calculated to the data using the least squares method with respect to the tab interface angle.

2.4.2 Final Design

To obtain a single metric from the complexity and strength metrics, each unitless metric was normalized, then the two values were added to obtain the final geometry metric. The results of this analysis are shown in Figure 2.7 for tab angles varying from $0°$ to $90°$ (Figure 2.7(a)-2.7(d)) The optimal design at the minimum of the curve was found to be very close to $30°$ with favorable overall properties. This geometry was used in all subsequent chapters of this work involving interlocking square tile voxels.
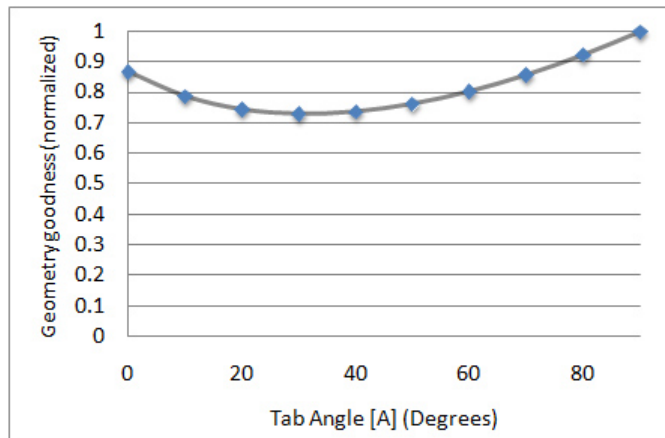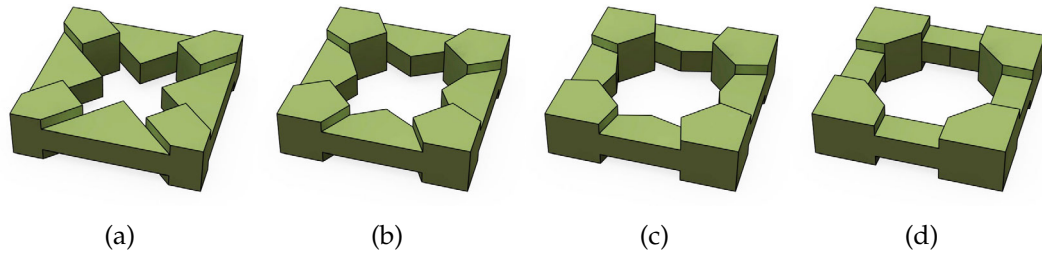
Figure 2.7: The interface angle of the tabs was optimized based on a com-
bination of minimizing the total surface area of the tile and
minimizing the internal stress when stacked microtiles are put
under lateral load. Tab angles of 0° (a), 30° (b), 60° (c), and 90°
(d) illustrate the range of geometries under consideration. The
optimal angle was determined to be 30°, which corresponds to
a relatively compact shape that performs well under mechani-
cal loading.

CHAPTER 3

VOXEL FABRICATION METHODS

3.1 Overview

Physical voxels were used at a variety of scales both to demonstrate large-scale voxel fabrication and to demonstrate parallel fabrication methods and assembly of interlocking voxels. For large scale voxel fabrication, both metal and plastic spherical voxels were procured in large quantities. For voxel design verification, voxel shapes were 3D printed at a millimeter scale that are suitable for human manipulation. To demonstrate the possibilities of fabricating and assembling interlocking voxels at a micro scale, a small number of voxels were micro fabricated and assembled. The details of fabricating or procuring each of these are outlined in this chapter.

3.1.1 Spherical Voxels

Spheres are relatively easy to fabricate at a wide range of scales with a high degree of accuracy. For the proof-of-concept experiments in this work, spheres on the order of 1-2mm diameter were used. For both plastic and metallic balls, the process to make them has been well explored and optimized. Plastic balls can simply be molded to shape very accurately in large quantities using automated injection molding equipment. Typically there is some variation in this process, so the finished spheres are sorted for different tolerance ranges.

The process for making metallic balls at a 1-2mm scale is slightly more in-

volved but still relatively efficient. The major steps are as follows:

1. Cutting: A continuous rod or wire of the desired metal is cut to length so that each section contains approximately the correct amount of material for a single ball.

2. Heading: The blanks are then fed through a header which stamps each piece into a sphere shape.

3. Flashing: The extra material in the form of the flashing where the two halves of the heading process come together is removed.

4. Heat Treating: For steels and other heat-treatable metals, the balls are heat treated to harden the ball for grinding.

5. Grinding: Each ball is then ground to approximately correct dimensions using appropriately spaced grinding surfaces.

6. Lapping: This final shaping and polishing of the ball determines the tolerance and surface finish of each ball.

7. Inspection: The balls are run through a tapered roller or inspected optically to verify the desired shape and tolerance demands.

Because of the overhead of the equipment involved, spheres used in this work were obtained in bulk directly from specialized manufacturers. Because of the economies of scale, an order of 50,000 1/16″ stainless steel balls with a tolerance of +/- 0.001″ cost less than two cents each. Similar acrylic plastic balls were even less expensive, coming in at a fraction of a cent. These 1/16″ spheres were used for all physical experiments involving spherical voxels.

3.1.2 3D Printed Voxels

In order to verify fit and assembly of the more complex voxel shapes, many designs were 3D printed to verify assembly. Two methods were used. A Stratasys Dimension SST 768 fused deposition modeling (FDM) system was able to fabricate voxels at around a scale of 1cm. This process uses an abs-like thermoplastic which makes durable, robust voxels that can be assembled and disassembled many times.

Smaller and higher resolution voxels were printed using Objet Geometries' polyjet technology. This involved an Objet Eden 260V or an Objet Connex 500, depending on timing. The material used in the polyjet process is currently less robust than the FDM thermoplastic, but the resolution and speed are much better. This enables fast fabrication of a few hundred of voxels several millimeters wide in a matter of minutes. These were used to assemble structures of several hundred voxels to verify pseudo-continuous surface approximations.

3.1.3 Microfabricated Voxels

In order to achieve truly functional and smooth-looking digital objects, the size of the voxels must decrease from macro to micro scale. Also, in order to create enough voxels to print a macro scale object from micro scale voxels, the method of fabrication the voxels must be parallelizable to create tens of thousands of voxels at once. Here parallel fabrication of $500\mu m$ square micro voxels are detailed using photolithography techniques suitable for subsequent 3D assembly.

The final voxel geometry was fabricated using two different fabrication tech-

niques: Silicon tiles were micro machined out of silicon and SU-8 polymer tiles were built up in a three layer additive process. The resulting bricks were characterized, and proof-of-concept structures comprising 10 bricks were assembled to demonstrate physical interlocking and compatibility between the two materials.

Modular architectures comprising functional components with standard interfaces have been recognized as key to the growth in complexity in microelectronic systems [146, 61]. Mechanical systems and components, however, have largely resisted standardization, thus hindering the ability to quickly assemble complex systems out of standardized components. One challenge to realizing three-dimensional modular microsystem architectures is the identification of a basic mechanical building block. Once such a module is designed, it can be fabricated of various materials in independent processes. Both passive and functionalized bricks can be simulated, fabricated and assembled, spanning a large design space of potential integrated systems.

A number of systems have explored the use of modular interlocking components [166]. While these systems are inherently serial in their assembly, they demonstrate the feasibility of manipulating and assembling interlocking shapes at this scale [209]. However, the components used in these systems are not geared towards assembly of general 3D integrated structures. For example, they are porous, a property that is well suited for their use in tissue scaffolds, but could negatively affect the mechanical properties.

The tile edge dimension was selected to be $500\mu m$ based on the desired scale of the bricks. $500\mu m$ tiles are small enough that a structure composed of many such tiles would appear smooth to the human eye, yet large enough to be ma-

nipulated by hand when necessary. The tile height of $130\mu m$ was selected as a compromise between the two manufacturing processes that were pursued to fabricate the bricks. The polymer bricks were built up from layers of SU-8 photoresist in a additive process. Given the relative difficulty of spinning very accurate and thick layers of SU-8 and the processing difficultly of aligning a large number of separate SU-8 layers, the geometry was kept as thin as possible. On the other hand, the silicon bricks were created with a subtractive process. Since the initial thickness of the wafer dictated the height of the bricks, there were constraints on the thinnest possible wafer that could be reasonably handled. The $130\mu m$ thickness was an appropriate compromise between manipulating silicon wafers that were thick enough to handle while not necessitating unreasonably thick SU-8 layers. A $5\mu m$ offset was included between all mating surfaces to mitigate concerns that the brittle tiles would fracture upon assembly.

3.2 Micro Voxel Manufacturing Process

3.2.1 Polymer Bricks

Standard photolithography techniques were used to create MicroChem SU-8 polymer micro bricks. SU-8 is an ideal negative epoxy based photoresist for this purpose for three reasons. First, high aspect ratio structures are routinely fabricated with excellent accuracy [111]. Second, it has a relatively large thickness allowance [9] and is available in a wide range of viscosities. Lastly, it is known for being robust and physically strong. The 2.5D design of the microtiles enabled the SU-8 tiles to be created with 3 layers of separately spun and exposed
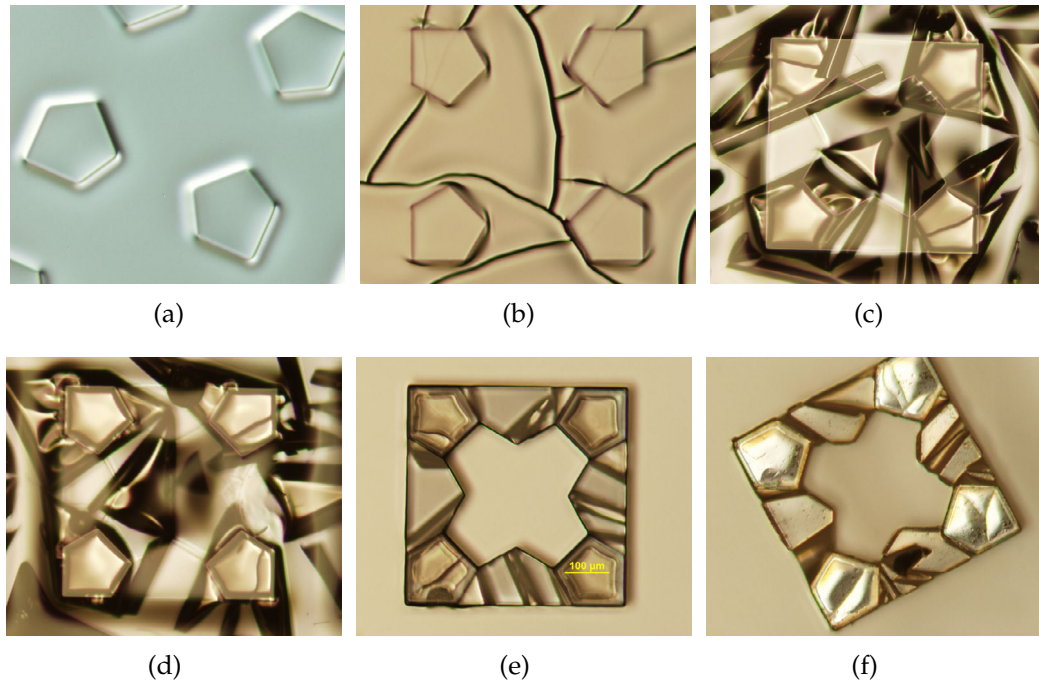
Figure 3.1: The process of creating SU-8 micro bricks. The first layer of SU-8 was deposited over a sacrificial layer, exposed, and baked (a). A chrome layer masked the first SU-8 layer from being further cross linked in later steps (b). Then the second (c) and third layer (d) of SU-8 were spun, exposed, and baked. The top layers of SU-8 were then developed away and the tiles were placed in a chrome etch to remove excess regions of the chrome layer (e). Then the final layer of SU-8 was developed away and the sacrificial layer was etched away, leaving free tiles.

SU-8. Overexposure of underlying SU-8 was minimized by masking the first layer with a vacuum deposited metal layer. Process details are detailed below and in Figure 3.1.

Before any layers of SU-8 were deposited, alignment marks were etched on the bottom side of the wafer to provide a reference to keep subsequent layer in proper alignment. SC 1827 photoresist was spun for 30 seconds at 2000 rpm on the underside of the wafer, prebaked for 1 min at 115C and passively cooled to room temperature. A generic alignment mark photo mask was used to expose

the wafer for 15 seconds, then the wafer was developed normally. Twelve loops in a Uniaxis plasma etcher resulted in alignment marks etched approximately 4.8 micrometers deep, which was sufficient to be visible in subsequent alignment steps. The wafer was then stripped clean of the remaining photoresist.

In order to facilitate the removal of the SU-8 tiles from the wafer substrate, a sacrificial layer of MicroChem Omnicoat was first deposited. To ensure reliable release of the finished bricks, it was found that this layer should be no less than 17nm thick. After the layer was spun, the wafer was placed on a hotplate for 1 minute at 200° to harden, then passively cooled to room temperature.

SU-8 2020 was selected for the thinner bottom and top layers in order to achieve the desired $35\mu m$ layer thickness. SU-8 2050 was used for the thicker $70\mu m$ center layer. For the $30\mu m$ layers, the wafer was subjected to a spin process involving a closed bowl spin of 500 rpm for 10 seconds followed by 3000 rpm for 30 seconds, with accelerations of 1000 rev/s. The sequence was completed with an uncovered continuous spin of 1000 rpm to facilitate the edge bead removal. During this final stage, a SU-8 remover solvent was applied for 15 seconds at a distance of 10mm from the edge of the wafer followed by a sweeping application starting outside the edge of the wafer and then moving inward until 8 mm from the edge of the wafer. The wafer was then allowed to continue spinning uncovered for 60 sec to enable uniform solvent evaporation.
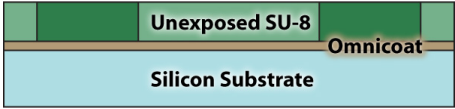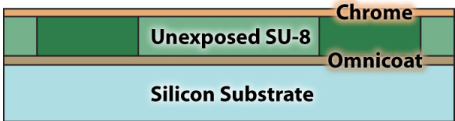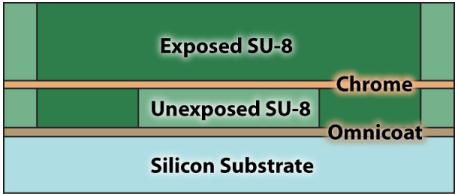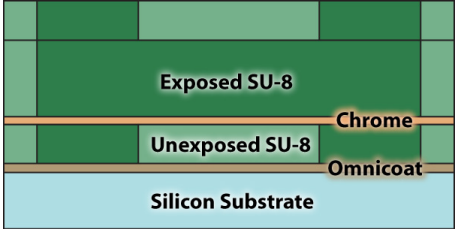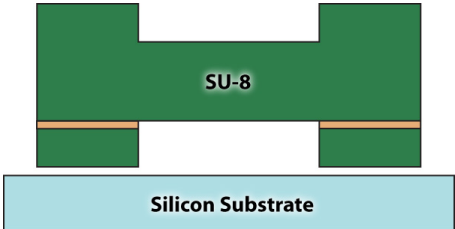
The wafer was then soft-baked and exposed with a bottom side soft contact alignment. A filter was used to eliminate light below 350 nm wavelength to prevent flawed exposure and rough edges of the SU-8. The wafer was then placed on the hotplate for a post bake. A longer hotter post-bake (30 minutes at 95C) is required to help cross link the exposed SU-8 and remove as much remaining

solvent in the SU-8 as possible. This is crucial to provide a stable substrate for the remaining process steps, since the unexposed SU-8 must remain in place to provide a flat substrate for the next layers.

Because the tiles have overhanging regions by necessity of design, care must be taken when exposing subsequent layers that no UV light can reach this first layer and cross-link any unexposed regions of SU-8. To this end a UV-opaque metallic layer was deposited between the first and second SU-8 layers. Chrome was selected for this task. A thermal evaporation of 100 nm of chrome was applied at 4 angstroms per second and 10 kV. This relatively slow chrome deposition rate yielded more favorable chrome layers than faster depositions.

The second and third layers of SU-8 were done in a similar manner to the first layer of SU-8, with appropriate adjustments to the spin schedule to attain a $70 \mu m$ center layer with the SU-8 2050. Once all layers were in place and exposed, the SU-8 was developed to remove the superficial supporting material. After rinsing the wafer with IPA, it was placed in a chrome etch bath for 90 seconds to remove the now-revealed chrome layer. Next the wafer was dried and placed in another SU-8 developer bath to develop the bottom layer. The wafer was then rinsed with IPA and air dried to remove the unexposed SU-8 of the initial layer. To finally release the bricks from the substrate, the wafer is placed in a MF 321 developer (TMAH) bath for approximately 30 minutes until the bricks are released from the Omnicoat layer. The bricks were then strained out of solution and set aside to dry for assembly. Complete SU-8 tiles are shown in Figure 3.2. A pictorial process table is shown in Table 3.1.

Table 3.1: SU-8 brick fabrication process diagrams. This multilayer additive process makes use of the SU-8 photopolymer and selective masking to create the tile geometry.

Step	Description	Diagram
1	Substrate patterned with sacrificial release layer	Omnicoat / Silicon Substrate
2	First layer of SU-8 is spun and patterned	Unexposed SU-8 / Omnicoat / Silicon Substrate
3	Masking chrome layer is deposited	Chrome / Unexposed SU-8 / Omnicoat / Silicon Substrate
4	Second layer of SU-8 is deposited and patterned	Exposed SU-8 / Chrome / Unexposed SU-8 / Omnicoat / Silicon Substrate
5	Third layer of SU-8 is deposited and patterned	Exposed SU-8 / Chrome / Unexposed SU-8 / Omnicoat / Silicon Substrate
6	SU-8, exposed chrome, and release layer are developed away	SU-8 / Silicon Substrate
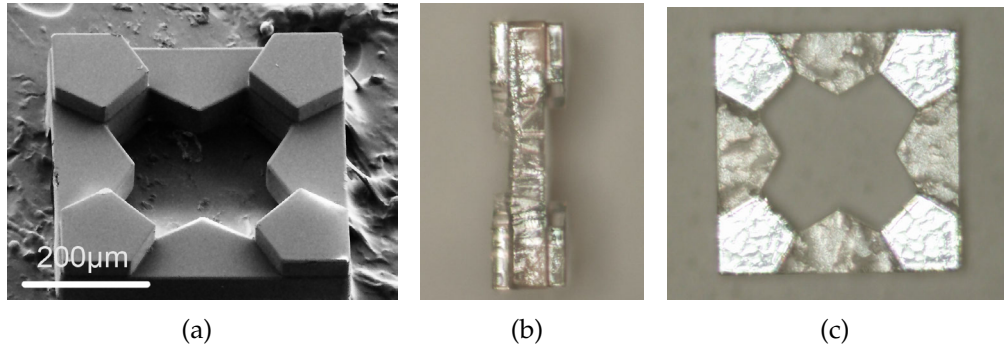
Figure 3.2: Completed SU-8 micro bricks. The SEM image (a) shows excellent feature accuracy. Side (b) and top (c) views show the individual layers of SU-8 clearly, as well as the finished aspect ratio of the tiles.

3.2.2 Silicon Bricks

Although silicon is a traditional material for microfabrication, fabricating free bricks out of silicon had its own challenges. Given that the target brick thickness was approximately $130\mu m$ thick, the silicon voxels were fabricated by two-sided etching of a thin silicon wafer (Figure 3.3). Normal silicon wafers were first commercially ground to the correct thickness. Each side of the brick underwent two etching steps to reach the center plane of the wafer and impart the correct geometry.

To do a two-step silicon etch, a layer of SiO_2 was first deposited on one side of the wafer. This SiO_2 was then patterned to mask the area corresponding to the feet of the bricks. This was accomplished by masking with a positive photoresist and then reactive ion etching the surface to selectively remove the SiO_2. Then another layer of photoresist was deposited over the patterned SiO_2. This layer was patterned and developed to leave the negative shape of the middle layer of the brick exposed. At this point the exposed regions of the wafer were etched down $35\mu m$ with an inductively coupled plasma etcher utilizing
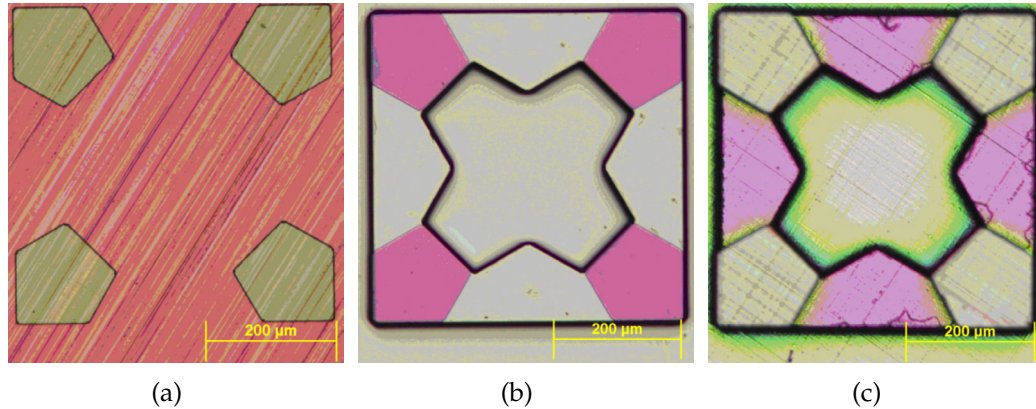
Figure 3.3: The process of creating silicon micro bricks. A silicon wafer pre-ground to the correct total brick thickness had SiO$_2$ deposited and etched to mask the second silicon etching process (a). Then another layer of photoresist was patterned and the first silicon etching stage was completed, etching away the center and outside of each tile(b). Then the most recent layer of photoresist was stripped and another silicon etching stage was completed (c), leaving the feet higher than the rest. A retaining membrane was then deposited on this side to hold the bricks in place while the process is completed from the opposite side.

the Bosch process for deep silicon etching. Then the remaining layer of photoresist was stripped off, leaving only the previously patterned SiO$_2$ layer. Another $35\mu m$ etch was done so that areas exposed for the first etch were now approximately $65\mu m$ deep, corresponding to the approximate mid-plane of the wafer.

Before the same process could be completed from the opposite side to complete the brick geometry, a sacrificial membrane was deposited on the recently-etched side to hold the bricks in place during the final silicon etching step from the other side. A combined layer of SiO$_2$ and nitride was deposited using plasma enhanced chemical vapor deposition (PECVD). The wafer was then flipped and the two-layer etching process was repeated. The final etching step was allowed to continue until silicon wafer was etched completely through in the exposed regions. The membrane holding the bricks together was then dis-

Table 3.2: Silicon brick fabrication process diagrams. This multilayer subtractive process creates the tile geometry by etching away regions of a thin silicon wafer.

Step	Description	Diagram
1	Silicon patterned with SiO_2	
2	Photoresist deposited and patterned	
3	First silicon etching step is performed	
4	Photoresist is stripped and second silicon etching step is performed	
5	Retaining layer of SiO_2 and nitride is deposited	
6	Process is repeated from the opposite side	

solved to leave the free bricks. A pictorial process table is shown in Table 3.2.

Table 3.3: Summary of measured tile dimensions of silicon and SU-8 micro-fabricated voxels. Dimensions and standard errors are based on measurements of ten tiles.

	Silicon		SU-8	
	Dimension	StdErr	Dimension	StdErr
Tile Width	$497.08\mu m$	$0.14\mu m$	$500.75\mu m$	$0.13\mu m$
Total Thickness	$113.39\mu m$	$0.21\mu m$	$153.65\mu m$	$0.53\mu m$
Center Layer Thickness	$56.35\mu m$	$0.18\mu m$	$75.53\mu m$	$0.34\mu m$

3.3 Micro Voxel Shape Analysis

3.3.1 Polymer Brick Results

Overall the SU-8 tiles displayed excellent feature accuracy in the horizontal plane. However, the SU-8 tiles ended up slightly thicker than the target dimension, although not detrimentally so. In the horizontal plane, the tiles were $0.75\mu m$ large (Table 3.3). This is likely due to mild bleeding of the UV at the edged of the mask, which cross-linked slightly more than the nominal amount of SU-8. The most difficult to control parameter in the SU-8 process was spinning an accurate thickness of the SU-8 layers. Based on the steps outlined earlier, the tiles ended up being $153\mu m$ thick on average, which was slightly thicker than the target height of $140\mu m$. However, the relative thicknesses of the layers were very close to the desirable ratios, with the center layer being 49.2% of the total thickness (nominally 50%).

The main processing challenge of the SU-8 brick fabrication process was ob-

taining a suitable UV masking layer after the first SU-8 layer. This masking layer needed to be thick enough to successfully block light from penetrating and curing the SU-8 below, while not being so thick as to be brittle and crack. This in turn would also allow curing of the underlying SU-8 beneath the cracks. Even with the successful 100nm chrome layer, this masking layer was observed to have a few minor ruptures (Figure 3.1(b)) after cooling from deposition. Subsequent step in the fabrication process introduced dramatic wrinkling of the chrome layer, likely from the thermal stresses of baking and cooling the SU-8. However, UV bleeding was still kept to a minimum, as evidenced by lack of unwanted cured SU-8 as illustrated in Figure 3.2(b).

Another option to mask the lower layer of SU-8 was to replace the chrome in the process with a thermally deposited aluminum layer. This yielded very smooth and effective masking layers that were not prone to cracking or wrinkling. However, aluminum is more reactive than chrome to the SU-8 developing solution. Developing the SU-8 led to unwanted dissolving of the aluminum that holds the bottom of the feet to the rest of the tile. With these significant incuts between the first and second layers of SU-8, the tiles with aluminum did not hold together well enough to assemble.

3.3.2 Silicon Brick Results

In contrast to the SU-8 tiles, the silicon tiles ended up smaller than the target dimensions. In the horizontal plane, the tiles were $2.92 \mu m$ small (Table 3.3). This can likely be attributed to the plasma etching mildly undercutting the masked region. However, the wafers that were used in these experiments were on aver-
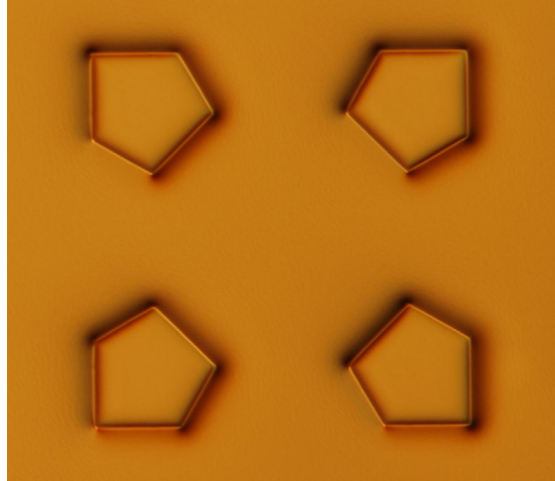
Figure 3.4: Using aluminum instead of chrome (compare directly to Figure 3.1(b)) in the SU-8 brick fabrication yielded less cracking, but aluminum reacted significantly with SU-8 developer which led to incutting of the tiles at the interface between the aluminum and the SU-8 in subsequent steps.

age only $113\mu m$ thick, which gave the silicon bricks a significantly different aspect ratio than the SU-8 tiles. However the relative thicknesses of the feet with respect to the thickness of the center layer was 49.7%, which was very close to the desired nominal 50%.

The challenges of fabrication the silicon bricks revolved around the use of very thin and fragile silicon wafers. Besides needing extra care in all stages of processing, the correct proportions of SiO_2 and nitride for the retaining layer were critical. The retaining layer had to be tensile, so that as the final microns of silicon were etched away, the tiles would hold in the correct position. However, a tensile layer on one side of a very thin silicon wafer introduces unwanted stresses that would repeatedly shatter the wafer in the final stages of etching. To address this, the membrane was tuned to be only very slightly tensile by correctly proportioning SiO_2 and nitride.

3.4 Verification of Micro Brick assembly

In order to demonstrate the tiles interlocking in 3D, a proof of concept structure was assembled by hand from SU-8 bricks. The structure contains 9 tiles in three layers, and demonstrated the effectiveness of purely friction-based adhesion at the $500\mu m$ scale. An SEM image is shown in Figure 3.5(a), with a tenth tile unattached for reference. At $500\mu m$, the tiles can be manipulated by tweezers, although to assemble any significant number of tiles will necessitate further development of an automated system. Even though the SU-8 tiles were slightly on the large side, the surfaces were compliant enough to allow for assembly without undue breakage of the micro tiles.

The silicon bricks, however, proved relatively fragile when pressed together, even though slightly undersized. This can be attributed to the relatively uncontrolled forces of assembling them by hand with tweezers. However, the more robust SU-8 tiles had enough compliance to assemble a structure that combined SU-8 and silicon tiles (Figure 3.6). Even though the aspect ratios were significantly different, they were still compatible and objects with a small number of layers could be assembled without issue. However, moving to objects with more layers will necessitate similar aspect ratios between tiles.

3.5 Future of Micro Voxels

The key to scaling up the complexity of micro systems lies in modularizing material and function, which requires standardizing the interface between components. In this way, each module type can be efficiently fabricated in indepen-
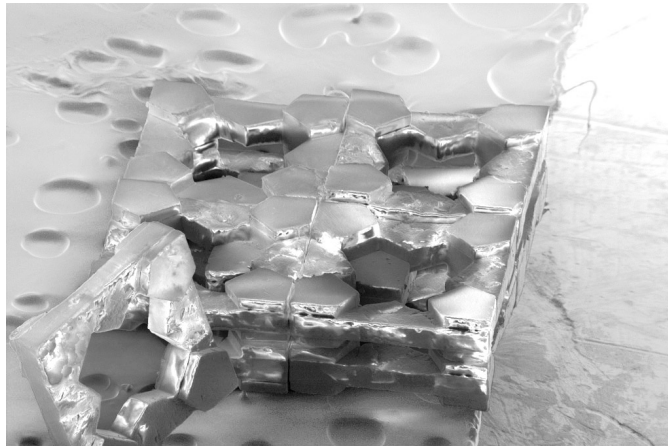
(a)
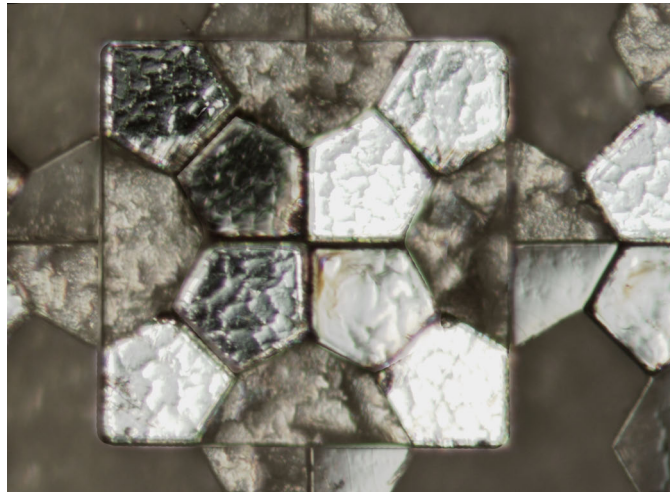
(b)

Figure 3.5: An assembly of $500\mu m$ tiles demonstrate a 3D interlocking structure. An SEM image shows the whole structure (a), while optical image shows the feet of the tiles in the lower layer inserted into the ring of the top voxel(b).

dent, optimized processes then combined into a functional hybrid system. This enables materials and functions that would otherwise be mutually incompatible to be combined in a single integrated system. The first step in this process is to define a suitable mechanical module that is both robust and suitable for a massively parallel 3D assembly process.

Figure 3.6: An 11 brick two layer multi-material structure was assembled by hand. Although the silicon bricks were too fragile to assemble on their own, combining them with the polymer bricks yielded a stable assembly. The dark tiles are silicon, while the translucent tiles are SU-8.

The mechanical architecture presented here fulfills these requirements. Components have simple 2.5D geometric shapes that could be fabricated in bulk by a wide variety of common parallel microscale fabrication techniques. The bricks physically interlock, yet can be assembled in a layer-by-layer process without complex 3D alignment or manipulation. The bricks also maximize mechanical properties by utilizing a space-filling design and minimizing internal stress under tension.

There are many potential applications for 3D multi-material micro brick systems. Even with just two bricks of differing material stiffnesses there are many possibilities for creating structures with novel properties. Once modularity is
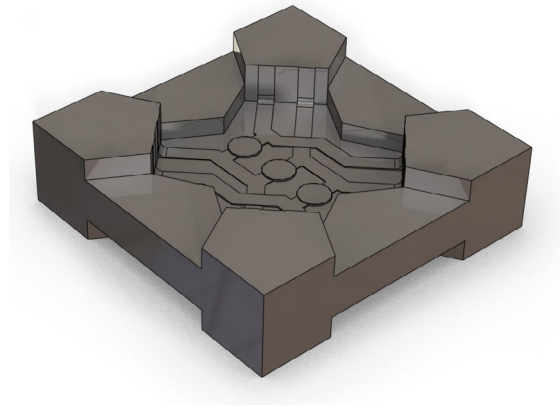
Figure 3.7: Future micro bricks will contain specific functionality. The internal circuitry is not structural and therefore not considered in this analysis.

enabled by a well-defined interfaces between components, there are many ways to accelerate the design and assembly process. In the future, one can envision hierarchical assembly processes where individual components are assembled functional macro-structures which are then assembled into larger systems. Alternatively, more complex sub-systems (e.g. a micro controller component) could be prefabricated in a larger package such as a $3 \times 3 \times 3$ cube that is compatible with the smaller components.

In the distant future, a library of compatible bricks will span not only the range of properties of materials currently available to engineers, but also incorporate specific functionalities that would enable 3D electro-mechanical objects to be fabricated quickly and cheaply from micro voxels (Figure 3.7). Because of the reversible nature of the mechanical assembly, these objects could be just as easily decomposed back into individual tiles for re-use in other objects. In this way, the advantages of a modular architecture can be leveraged in the physical world to enable reconfigurable 3D functional macroscale structures composed of microscale components.

Part II

Rapid Assembly Methods

CHAPTER 4

SERIAL DEPOSITION METHODS

4.1   Serial and Parallel Printing

The main challenge of fabricating digital objects is determining how to physically program a digital object from physical bits. The fundamental challenge regards how to individually address each physical location within a potentially massive lattice. Historically, this problem has been well-addressed in two dimensional space by computer printers. A brief summary provides the background for the 3D rapid assemblers developed here.

There are two fundamental ways to address a large number of discrete locations (i.e. pixels in two dimensions) quickly. Both of these were present early in computer printer development in the 1950's. The first method, referred to as *serial* here, has a single continuously scanning print head that can very quickly address locations as it scans over them. Early implementations of serial printers include dot-matrix printers, where a solenoid would actuate a pin-shaped hammer to press a small region of ink onto the paper. This method usually involved several pins in a row perpendicular to the scanning direction to reduce the number of scans that needed to be completed, making these processes approach a 1D parallel process. Several decades later, inkjet printers began to gain popularity. While still a serial printing process, discrete drops of ink were jetted from orifices using thermal or piezoelectric forces. The print head still scans continuously though, and ink drops are fired only when the corresponding pixel is needed. Such systems began to be known as "drop on demand" inkjet printers.

A second fundamentally different printing process was also developed around the same time that is inherently *parallel*. The process of xerography had been around since the 1930's. A charged plate of photoconductive material is statically charged, then selectively discharged using a photo mask. Toner particles are then dusted in and stick only to the charged regions. The selectively patterned toner is then transferred to the new page. In the 1950's, xerography was adapted for use with a scanning laser to create the first laser printers. Although most laser printers currently make use of continuous rollers, the xerography process is inherently parallel in that an entire 2D sheet can be processed in one cycle.

Both the serial and parallel printing paradigms have scaled relatively equally in their ability to address millions of pixels in a matter of seconds for application in traditional paper printers. In the traditional analog 3D printing world, both concepts have also demonstrated to scale well to create complex, detailed 3D objects. Some of the first rapid prototyping machines used the parallel principle of layered object manufacturing, in which entire layers were cut from paper and stacked to create a 3D object. Several 3D printing processes also make use of serial inkjet heads to selectively deposit either a binder onto powder or the build material itself layer by layer to create 3D objects. Thus both the serial and the parallel paradigm are expected to be useful for large-scale digital manufacturing to eventually print billions or trillions of discrete units quickly.

Proof of concept hardware was fabricated to demonstrate both serial and parallel assembly of digital structures from 1/16″ spheres of multiple materials. In this chapter serial assembly is explored using the VoxJet research platform and in the next chapter parallel assembly is explored using the VoxLayer re-

search platform. It is important to note that a serial process can be extended to a 1D parallel process by the inclusion of a line of deposition sites, but this still is short of a full 2D parallel process.

4.2 Voxjet Research Platform

The Voxjet printer used here is shown in Figure 4.1. The theory of operation is very similar to that of a 3D inkjet printer, only discrete units are jetted instead of amorphous drops of ink. This USB-controlled platform is capable of simultaneously printing with three different materials and binding them together as they are placed. Additionally, a laser distance sensor enables closed loop feedback to perform error correction on the printed part. The print head scans continuously in a similar manner to an inkjet printer, and can deposit spheres of any given material in a continuous line at approximately 10Hz. When taking into account the time between rows, motion system, feedback, etc., the printer builds at a net rate of approximately 2-4 voxels per second, depending on the geometry. Thus, a 1000 element structure takes on the order of 5 minutes to build.

In order to address the challenge of scalability, all the alignment and placing of the spheres is handled mechanically by modular deposition modules (Figure 4.2). To load digital material in the module, spheres of the desired material are simply poured into the reservoir at the top of the voxel deposition module. In the first stage of the deposition process, the randomly aligned spheres are ordered into a single one-dimensional buffer. Here, the buffer contains a maximum of approximately twenty spheres which is sufficient to smooth out any random variances in the rate of refilling the buffer under continuous voxel
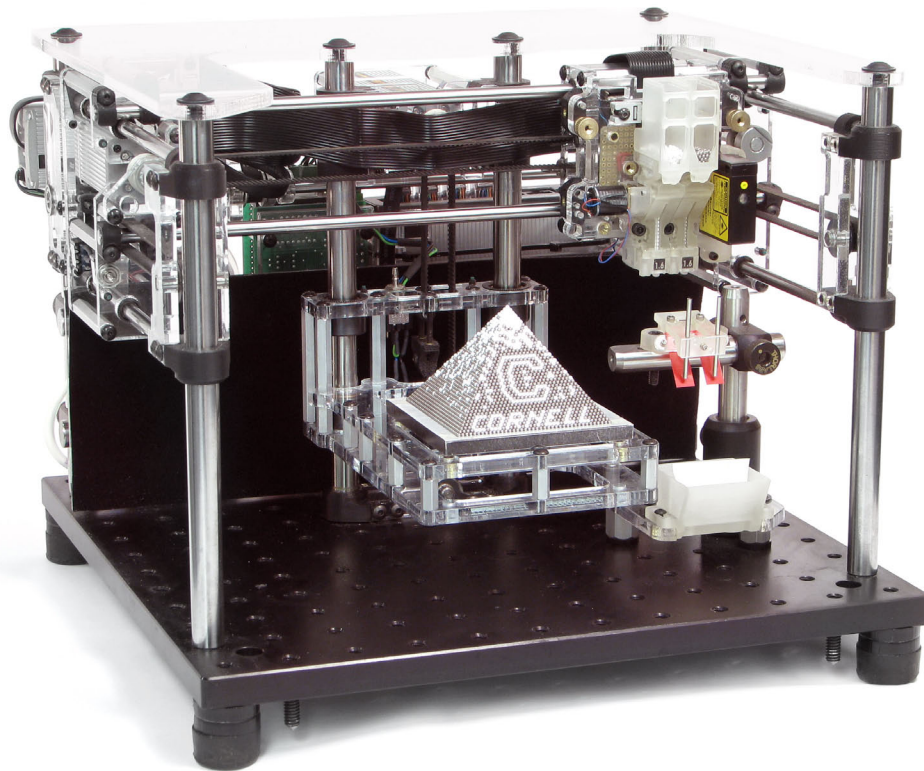
Figure 4.1: The VoxJet research platform was used to quickly and autonomously build multi-material structures made of thousands of 1.5mm spherical voxels. The system optionally deposits binder to hold the structure together and includes error correction via closed loop deposition feedback.

deposition.

The buffer is loaded by a continuously rotating paddle wheel at the bottom of the reservoir. Each paddle captures an individual spheres and drops it into the gravity-fed buffer tube. Special care was taken in designing the geometry such that the paddle wheel does not jam with spheres and that spheres are never forced downwards into the buffer. The paddle wheel is thus able to rotate continuously even when the buffer is full, such as in between row depositions. This
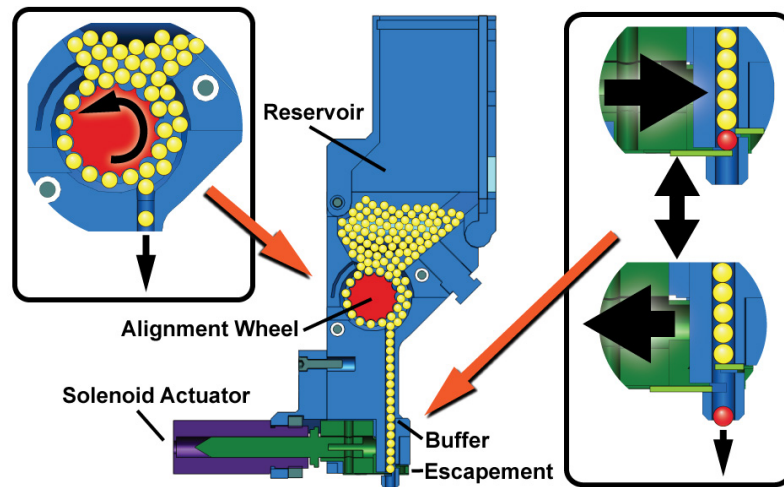
Figure 4.2: A cutaway diagram of the material deposition modules. Loose spheres are poured into the reservoir, which are aligned into a one-dimensional buffer by the continuously rotating alignment wheel. A solenoid actuated escapement mechanism deposits precisely one sphere with each cycle, and can reliable deposit up to ten spheres per second.

eliminates the need for costly feedback in this alignment process. The deposition modules were designed modularly such several may be stacked side-by side, and a single micro gear motor attached to one extreme provides the rotation for the alignment wheel of all attached modules.

4.3   Motion System Design

A three-axis positioning system was developed specifically to meet the needs for serial voxel deposition. Requirements included a print head that could scan across a build area in the X (transverse) direction at a minimum of 2 inches per second and positioning precision in the Y and Z directions of better than 0.001 inches. Another design goal that greatly aided the development process

was relying solely on off-the-shelf hardware components, 3D printed components, and laser cut pieces. These requirements allowed the VoxJet platform to undergo rapid design iterations because re-fabricating redesigned parts was a largely autonomous affair.

The most difficult choices revolved around which motion system to use and how to control it. The modular Snapmotor system [87] was selected as an appropriate level of modularity with enough flexibility to use it as the sole interface between a host computer and the VoxJet hardware. Integrated motors, encoders, and control circuit modules were selected with appropriate gear ratios to meet the speed and precision requirements. All motors plug into a central Snap Hub using standard RJ-45 cables that transmit both power and data. The Snap Hub is capable of storing and running programs locally, but here was configured for pass-through commands coming directly from the host PC over a standard USB connection.

In addition to motors for the three axis, a fourth general input-output module was used to send and receive information specific to the deposition head. Digital outputs control the actuation of each solenoid, a pulse-width modulated (PWM) motor controller turns the alignment motor, and an analog input channel reads the laser range finder. Programmatic access to all functions was provided by the vendor in the form of C++ source code.

Although low-level control of the motor positioning and feedback loop is handled locally at each motor module, the control loop of the build process was run on the host PC. Commands are sent to the network to initiate each movement and the host program continuously monitors the progress of movement to determine when each movement is complete. Communication with the net-

work is sufficiently fast to communicate in near-real time.

4.4 Deposition Head Design

The mechanical deposition of individual voxels is controlled by a variation of an escapement mechanism (Figure 4.2) actuated by a solenoid. With each cycle of the solenoid, the buffer is advanced under the force of gravity by exactly one voxel. When the solenoid activates, one tab enters the buffer stream right above the bottom sphere, holding the rest of spheres from advancing. The same motion also withdraws the tab holding the bottom sphere from falling, allowing this single sphere to drop. When the solenoid is deactivated, it returns to the original position, allowing the buffer to advance by one by the force of gravity. Thus, exactly one voxel falls out the bottom of the buffer, at a very predictable delay from the actuation of the solenoid. At a rate of 8Hz, the error rate of of this deposition mechanism is less than 0.01%, which is negligible for the current VoxJet implementation.

The deposition head is also capable of autonomously gluing the spheres during the build process (Figure 4.3). The application of the binder is accomplished using the standard method of a pressurized reservoir of binder controlled by a solenoid valve. For these experiments a polyvinyl acetate binder in aqueous solution is used. Advantages of this system include a relatively quick drying time, mechanical robustness, and the ability to reverse the bonding upon submersion in water. Regular white multi-purpose glue was further watered down at a ratio of 1:1 (glue to water) to yield a solution with low enough viscosity to jet. This was jetted at approximately 10psi (using a regulated diaphragm pump)
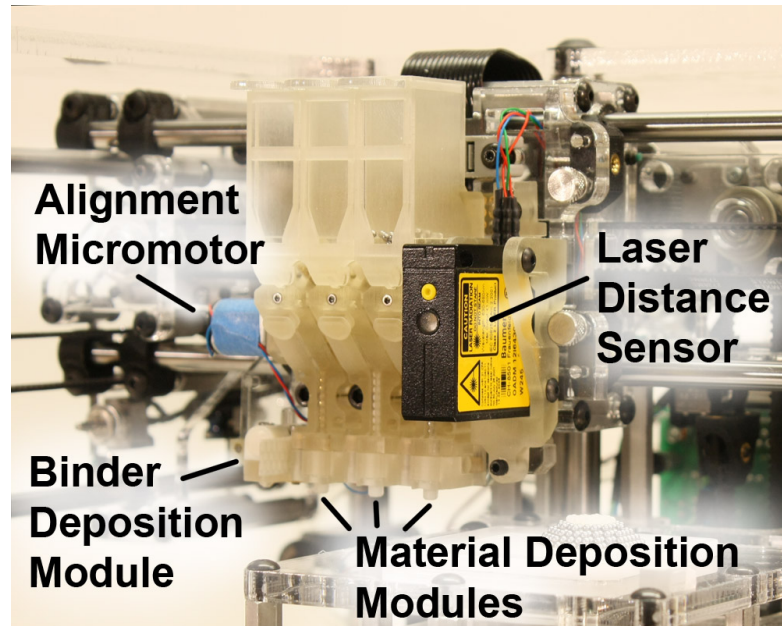
Figure 4.3: The deposition head consists of a binder depositing system, three separate material deposition modules, and a laser distance sensor. As the print head scans from right to left, binder is deposited appropriately, followed by spheres of any available material. The laser sensor verifies the placement of each sphere.

through a nozzle 0.020″ in diameter and solenoid actuation duration of about 17 ms.

Front and side views of the print head clearly show the major components of the deposition modules (Figure 4.4). Transistor-based solenoid driver circuits are housed on the circuit board to enable logic-level high and lows to be converted into 12 volt potentials with sufficient current available to actuate each solenoid. The feeders themselves are printed in 8 different pieces using an Objet Connex500 polyjet machine. Numerous design iterations perfected the exact interior dimensions needed specifically for manipulating and depositing 1.5mm (1/16″) balls with high reliability.

<center>(a)   (b)</center>
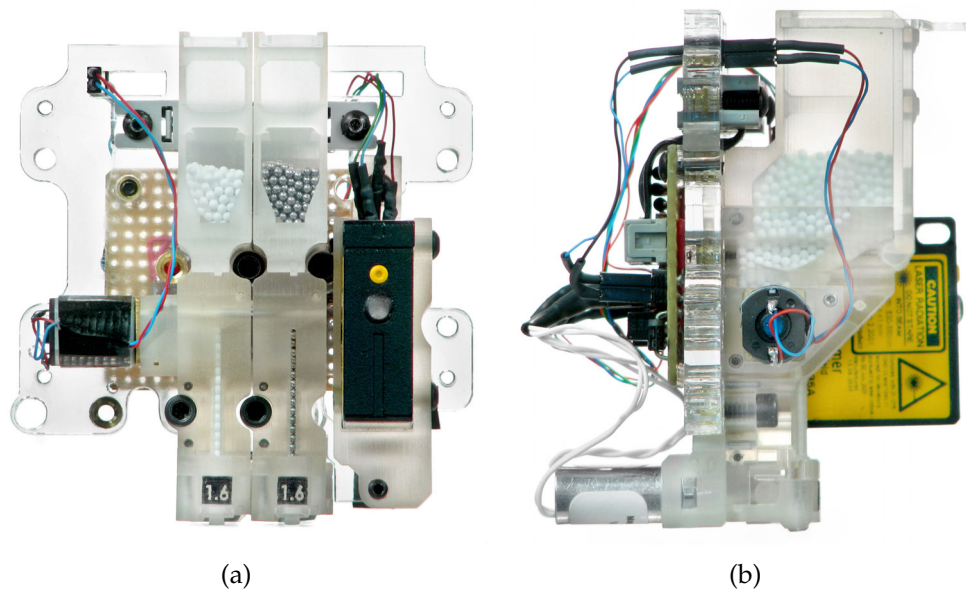
Figure 4.4: Front (a) and side(b) views of the serial deposition head show the major components, such as the solenoids used for actuation. The glue module is removed for clarity.

4.5 Voxel Deposition Feedback

Closed-loop feedback of the voxel deposition process is accomplished using a Baumer OADM-12 laser distance sensor, which enables error correction on the physical structure. With a theoretical measurement precision of two microns and a measurement update every millisecond, this compact module easily met the applicable design requirements. The output of the laser is an analog current that must be processed to obtain the appropriate distance the laser is currently reading. The output current is passed to ground through a 1k resistor to generate the proportional voltage across the resistor which can be input to the I/O board via an analog input. Because this input had only 8-bit resolution (256 total input values), the output of the laser was scaled to make full use of the output range over only 0.1" of distance.

The accuracy of laser-based range sensors inherently depends on the surface properties of the materials they are measuring. A perfect measured material would be an opaque, perfectly diffuse white surface with no specular reflection. As materials deviate from this ideal the accuracy of the reported distance decreases, sometimes dramatically if reflections (internal or external) come in to play.

Although the absolute measurements of sphere heights were nowhere close to the theoretical accuracy of the laser sensor, the laser sensor worked sufficiently well for all materials of spheres to determine robustly if a sphere was present or absent (Figure 4.5). Materials tested include those with diffuse surfaces and highly reflective surfaces. Even the presences of clear acrylic spheres are sensed correctly, making this a robust, material independent solution for feedback. However, the sensor does not differentiate between materials, so errors involving misplacement of material could pass undetected. The inclusion of machine vision to the feedback process would allow differentiation between materials in a future revision.

The inclusion of feedback allowed the robust printing of structures with many thousands of elements. These results are included in the following chapters.
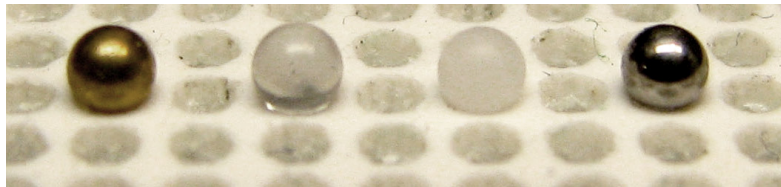
(a)

(b)

Figure 4.5: 1/16″ inch balls of four materials (brass, transparent acrylic, delrin, and steel) were placed side-by-side and scanned with the non-contact laser sensor (a). The scans (b) show significant artifacts that depend on the specularity, opacity, and diffuse properties of each material, but are sufficient to determine the presence or absence of a ball.

CHAPTER 5

PARALLEL DEPOSITION METHODS

5.1   Motivation for Massively Parallel Deposition

Although serial and 1D parallel methods can efficiently deposit many elements quickly, 2D parallel methods hold promise for scaling to structures of many more elements. Two methods for parallel assembly of components into arbitrary 3D lattice configurations are described here. An electrostatic method for selectively manipulating the voxels was explored initially. The results for the desired scale of 1.5mm spheres were inconclusive. The balls could be physically picked up in selective regions of a print head, but not reliably. An alternative opto-fluidic approach proved much more robust at these size scales and was used to demonstrate 3D multi-material voxel assembly.

Self assembly is often considered a massively parallel assembly process. Recent bottom-up self assembly techniques [195, 197, 100] offer some ability to spontaneously assemble materials in parallel guided by local interactions between components; however, self-assembled processes can be difficult to control and are generally limited to regular, semi-periodic or random structures [16, 185, 33]. Attempts have been made to reconcile different modes of parallel assembly [32, 153, 210] such as hierarchical [64, 78, 199, 109], directed [106, 75, 15], and templated [190, 26, 203, 76] self assembly, but are difficult to scale easily to 3D structures of millions of components in arbitrary configurations.

Alternatively, recent additive manufacturing technologies based on selective inkjetting also enable top-down fabrication of precise, complex geometries

with control over the internal material distribution [74], but they cannot handle prefabricated components. As a result, they are limited to a small set of homogeneous materials with mutually-compatible rheological and/or photo-curable properties [18, 81, 176]. Because the objects created in these processes are fundamentally continuous, all the advantages of the digital material paradigm are not realized.

The alternative process presented here focuses on a top-down 2D parallel process in which large numbers of components can be selectively assembled in a reasonable amount of time. Inspiration is derived from optical manipulation techniques used in a typical laser-printer. A laser printer addresses an entire layer at once by charging a photoconductive surface with a static electric charge, then selectively discharging it using a scanning laser [44]. When considering discrete voxels, the self alignment properties of the voxels are used to organize (self assemble) an entire, homogeneous layer of voxels. Once complete, the relative location of each voxel within this 2D matrix in the pre-assembly area is known. This allows an indexed print head to come into contact with a single voxel at each cell location within its matrix, which it can either attract or not using a selective electrostatic charge as used in xerography.

While electrostatic principles work well to manipulate toner particles (10 microns), they do not scale well to millimeter scale discrete components. Instead, optical energy can be used to control fluidic capillary effects that are well suited for those scales. A wetted surface can easily pick up small objects, much like a wet hand will pick up grains of sand. The level of wetting can be used to determine the location and strength of adhesion. The wetting is controlled by selectively drying a wetted surface using a scanning laser or projector.

This effect was demonstrated by depositing spheres of a variety of materials into a 3D lattice. The self-aligning nature of the spheres allows creation of highly-precise and repeatable objects. In these demonstrations, spherical 1.5mm diameter voxels were used in 55-unit triangular trays, but this process is extendable to larger layers, smaller voxels, and to any number of materials.

The general process used to fabricate a two material lattice of sphere using parallel deposition methods is illustrated in Figure 5.1. The target 3D configuration was first expressed in a blueprint made up of a series of binary bitmaps that correspond to successive layers of the physical materials to be printed. Each bit in the bitmap represents the existence or absence of a specific physical voxel type in a specific layer in the target object (Figure 5.1a). Prefabricated voxels of the appropriate types were then dispensed into the material feeder trays (Figure 5.1b).

Each layer of the matrix was then printed and stacked in turn, as follows: Voxels of each material were self-aligned using gravity and vibration in separate trays (Figure 5.1c). Once a uniform layer of a given material has been self-aligned into a perfect 2D lattice (Figure 5.1c), The correct cells on the print head are activated to selectively pick up voxels specified by the desired electronic bitmap (Figure 5.1d). Each layer of each material is then sequentially deposited on a build stage (Figure 5.1e) to form the desired multi-material 3D object (Figure 5.1f). The voxels are then post-processed to achieve the desired results. Desired voxels can be bound together to create a robust freeform 3D object and any sacrificial voxels can be removed (Figure 5.1g). In all the examples shown here, only two materials are used, but this process is fundamentally compatible with any number.
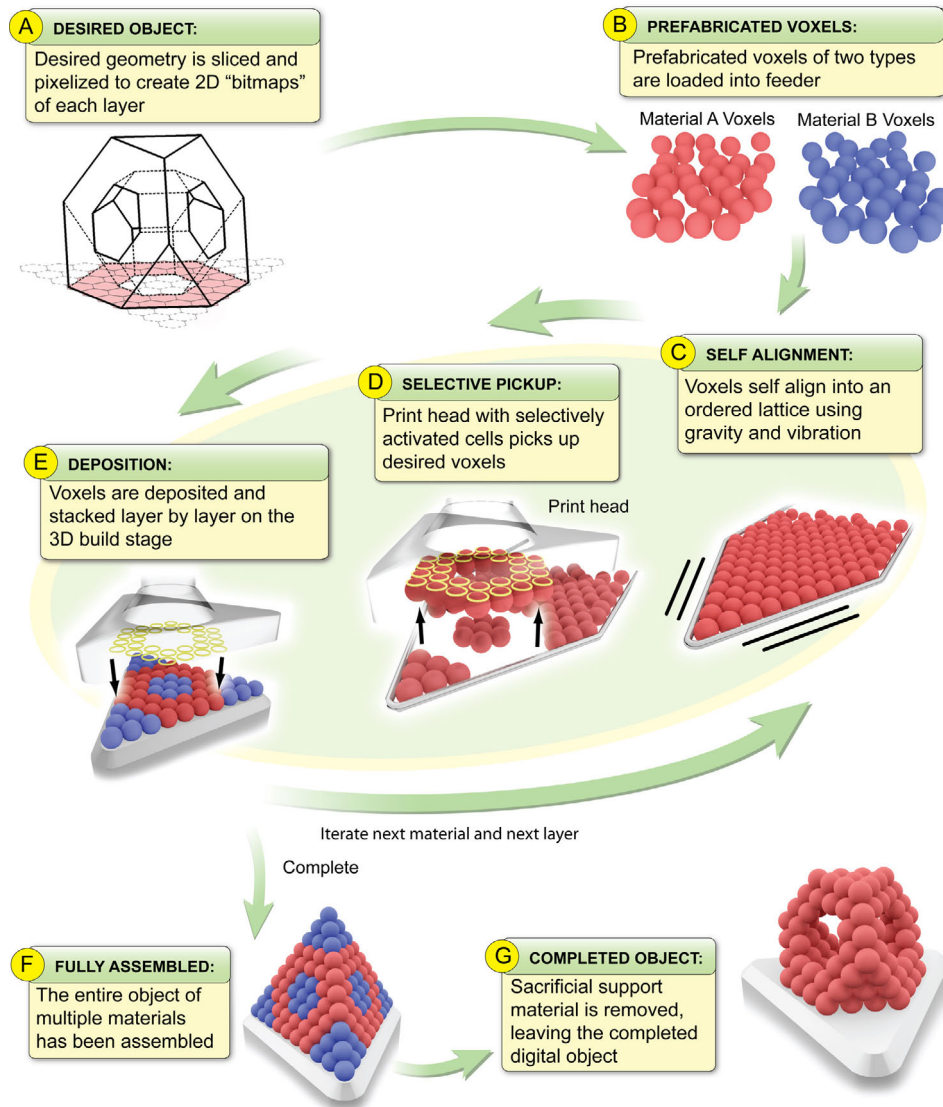
Figure 5.1: The process of parallel digital fabrication using spherical voxels: (A) A target blueprint is converted into a series of bitmaps describing the existence or absence of a particular voxel type in each layer of the object. (B) Prefabricated voxels of multiple materials are poured into feeders and (C) self align into an ordered lattice. (D) A selectively activated deposition tool picks up an entire layer of voxels based and deposits them on the build stage (E). Steps C through E are repeated for each material of each layer. Once the entire object is assembled (F), sacrificial support material is removed to create freeform geometry (G).

5.2   VoxLayer Research Platform

5.2.1   Overview

To demonstrate the concepts of parallel voxel manipulation described above, a prototype instrument (VoxLayer) was developed (Figure 5.2 and 5.3). The machine as shown here is equipped for opto-fluidic experiments, but can be re-equipped for electrostatic experiments. This research platform was implemented to selectively place 1.5mm spheres in a workspace of 220 voxels, arranged in a pyramid with 10 voxels to an edge. Modular stations allowed the machine to be easily adapted to the different voxel manipulation methods presented here. This provided the automated framework to achieve repeatable experimentation results.

The VoxLayer deposition print head (Figure 5.2a) moves to a series of stations (Figure 5.2b-e). Selective drying takes place at the exposure station (Figure 5.2a) from a pattern of light generated by the projector (Figure 5.2j) and scaled through a series of lenses (Figure 5.2i). A camera underneath monitors and controls the drying process in real time for opto-fluidic experiments. At the activation station, (Figure 5.2b) the entire surface of the print head is immersed in a solution of water and detergent before selective drying takes place. For electrostatic experiments a corona wire statically charges the print head surface instead.

Raw materials (spheres) are poured into the hoppers in the back of the material feeders, (Figure 5.2c) and each of the two feeders inclines approximately 3 degrees. Pager motors generating mechanical vibration are embedded in each

feeder. Activating these motors settles the spheres into the positions of lowest energy, corresponding to a close packed hexagonal lattice. A camera mounted beneath each feeder monitors the position of each sphere using machine vision techniques to verify the integrity of the 2D lattice.

The inspection station (Figure 5.2d) also contains a camera underneath which inspects the deposition head before and after depositing the spheres on the build stage (Figure 5.2e). This allows indirect verification of spheres that were actually deposited. The build stage simply holds the part being assembled, and moves down on the Z axis degree of freedom (Figure 5.2h) as subsequent layers are stacked up. Control electronics (Figure 5.2f) provide a USB interface to the master computer that controls the slave micro controllers in charge of the motion systems and camera image acquisition.

5.2.2   Motion Control Implementation

For the large-displacement degrees of freedom of the VoxLayer platform (the main X axis and the build stage Z axis), stepper motors were used, paired to MForce Microdrive Plus micro controllers. The X axis motor, a NEMA 17 mount 100-pole 2 phase motor is powered at 24 volts and is attached to a 12″ travel XSlide linear stage from Velmex corporation. The lead screw pitch was selected such that the X axis moves 0.002″ per motor step. The Z axis build stage motor is a NEMA 11 mount 100 pole two phase stepper motor also driven at 24 volts, with a fine pitched lead screw for positioning accuracy of approximately 0.0001″ per step.

For the lesser degrees of freedom, hobby servo motors were used for their
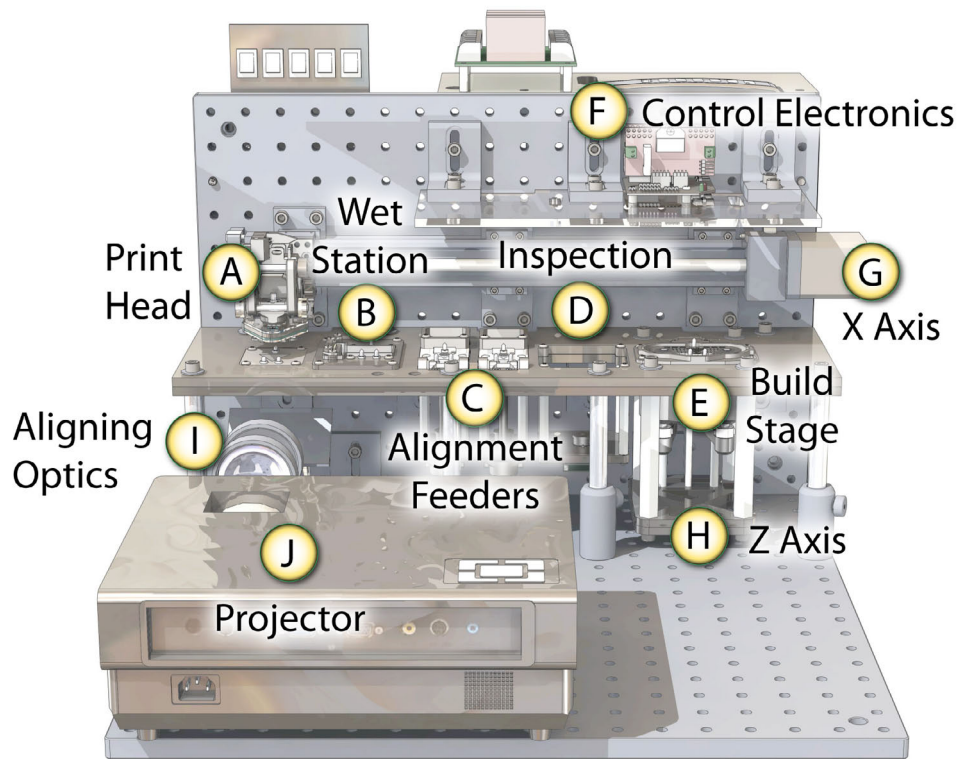
Figure 5.2: A schematic diagram of the parallel rapid assembler shows all major components. The print head (A) moves back and forth on the X axis of motion (G). A projector (J) shining through focusing optics (I) dries the print head selectively. A wet station (B) allows the entire head to be wetted while alignment feeders (C) align and hold layers of voxels for pickup. After an inspection station (D) each layer is deposited on the build stage (E) which moves along the Z axis of motion (H). Control electronics (F) direct the entire process.

simplicity and modularity. A Brainstem GP 1.0 microcontroller module with 8 independent servo channels was used. These controlled the movement of the deposition head up and down (for moving between stations) and rotating the deposition head through a range of 240 degrees. This rotation, accompanied by a mildly eccentric center of rotation, allowed the print head to deposit the layer of spheres at one of three offsets, depending on current layer offset. Servo mo-
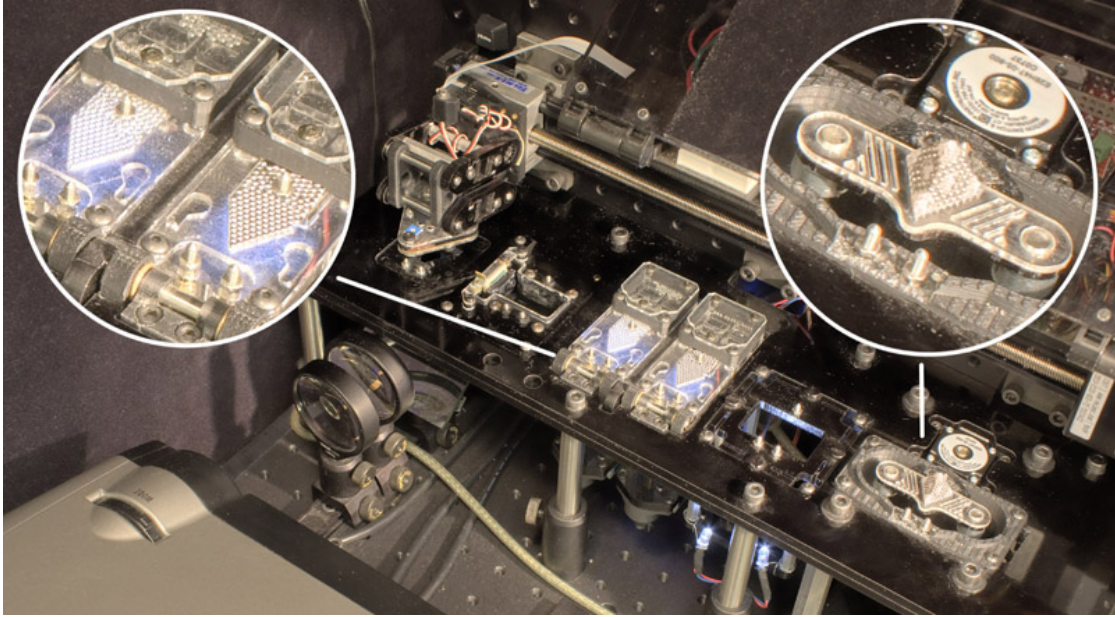
Figure 5.3: The implementation of the parallel rapid assembler shows a steel and acrylic object built at the maximum workspace of 220 voxels (right). The alignment feeders (left) hold aligned layers ready for selective pick up.

tors also controlled the elevation of the material feeders for the settling process. The pager motors used to vibrate the spheres into position were controlled by an add-on board to the Brainstem system, a Brainstem Moto 1.0 module. The motors were controlled via pulse-width modulation of a 5 volt input signal.

5.2.3 Machine Vision Details

At several steps in each layer deposition process, machine vision was used to monitor and verify important information regarding the build. During the self-alignment of voxels in the feeders, any errors (such as a void within the lattice or a dislocation) are characterized, and can be accounted for as necessary either by resetting the alignment process or by repeating the layer deposition for

error correction. Additional machine vision steps directly before and after deposition verify which voxels were actually deposited, and this information is used in a closed-loop algorithm to account for errors. OEM USB cameras with a native resolution of 1280x1024 were used for acquiring images in real time with LED-based structured lighting implemented as necessary. All machine vision algorithms used here rely on image subtraction routines to harvest useful information from the raw acquired images.

5.2.4 Voxel Layer Self Alignment

The key to the massively parallel assembly process is the ability to selectively pick up voxels at arbitrary locations within a pre-aligned layer. In order to accomplish this, an entire layer of voxels is self aligned using gravity and vibration. Figure 5.4 shows frames from a video taken from above capturing the self alignment of the spheres. This process is highly repeatable and suitable for unsupervised automation.

This process is effective with the voxel feeder spring-loaded to an inclination of approximately 2.5° (Figure 5.5). The voxel feeder was designed to hold a large number of voxels, and continually replenish the active layer as voxels are removed by the print head. An additional servo motor controlled the inclination of the voxel feeders for the alignment process. After alignment, the feeders were returned to near-level before the print head arrived.

To ensure a perfect layer every time, the self alignment of the voxels in the feeders is monitored in closed loop by machine vision. Initially the user performs a one-time calibration routine. First, a frame of the empty voxel feeder
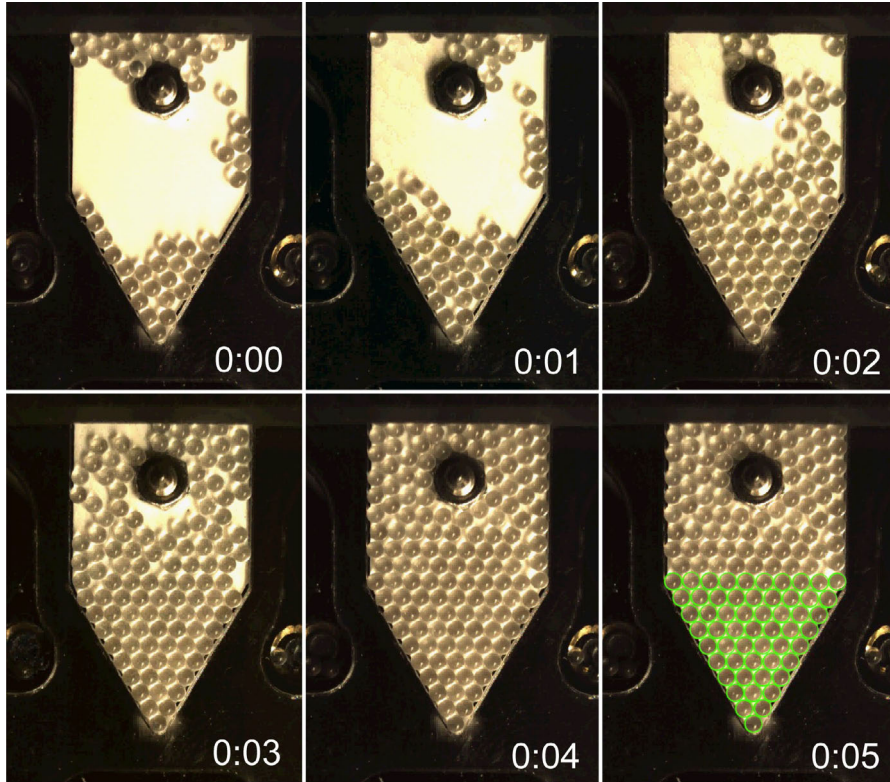
Figure 5.4: Frames from a video showing a layer of voxels self aligning in the feeder. The last frame shows the verification of the lattice within the triangular region of interest.
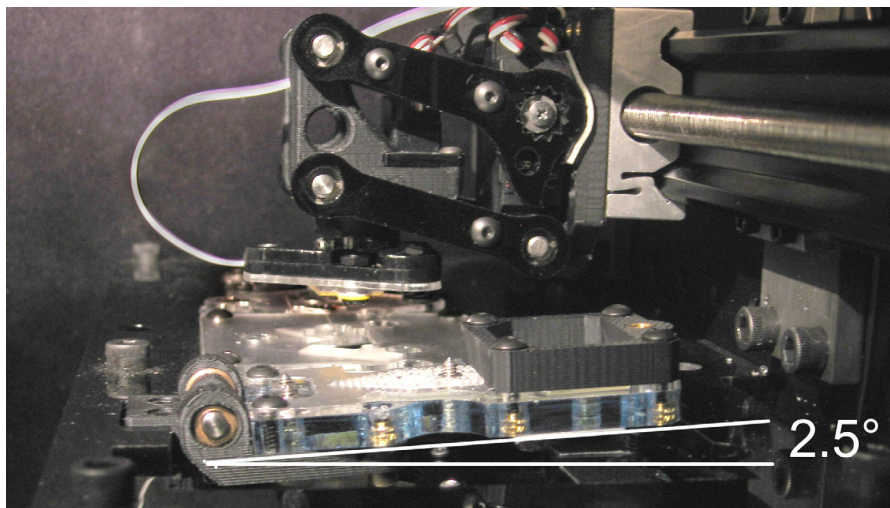
Figure 5.5: Photo of the experimental setup illustrating the inclination angle of the voxel feeder to align the spheres. The print head is also clearly visible in its upward (raised) position.

is recorded. For all subsequent images captured, this image will be subtracted. This has the effect of removing any artifacts in the field of view. Next, the user manually verifies an aligned layer of spheres, and identifies the corner spheres. A lattice of nominal positions in the camera coordinate system is thus calculated. The algorithm calculates the diameter of each sphere, then averages pixel by pixel each sphere in the field of view to create a master single sphere image for later comparison. This must be performed for every material with varying surface reflectance properties.

During a 3D build, the location of spheres within the field of view must be verified before every layer deposition step. A series of seeker points within the field of view is defined on a grid with significantly tighter resolution than the lattice of spheres. For each of these seeker points, the master image of the single sphere is subtracted pixel by pixel to obtain an error. The four surrounding cardinal locations are also tested and if any of these has a lower error than the original location, the algorithm repeats the process at this new point. This continues until all the original seeker locations have minimized their error. Each of these is then thresholded to determine whether it is in a sub-optimal local error well, or is actually the location of a sphere. It is then trivial to examine each possible location in the lattice to verify if a sphere is present within a certain tolerance.

With this method in place, the self alignment process can then be carried out in closed loop. The feeder is inclined and vibrated, then a frame is captured and analyzed. When all spheres are within some threshold of optimal lattice positions, the algorithm exits. Otherwise the algorithm chooses to either continue vibrating, or to reset the feeder by emptying all the spheres back into the
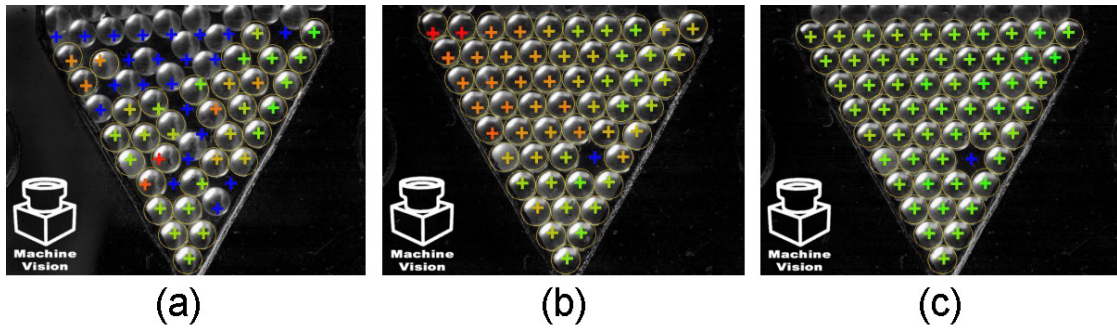
Figure 5.6: In the self alignment process, machine vision is critical to ensure that each sphere has assumed a distinct position within the lattice. Errors are characterized perfectly, and can be accounted for in software or the entire process retried. The original un-aligned spheres (a) are gradually ordered (b) into a perfect lattice (c).

hopper. Several representative frames from the alignment process are shown in Figure 5.6.

5.2.5 Voxel Deposition Feedback

The last machine vision step verifies which spheres are physically held and deposited by the deposition head. By differencing the spheres present both before and after the deposition step, it can be inferred which ones were deposited. Again, a reference image of the print head is obtained with no voxels present. After defining the possible location of spheres (based on user input on the reference image), the area of the image around each of these points is subtracted from each image captured for analysis. Again, the threshold of error representing a sphere present vs. not present is determined and applied to determine which positions contain spheres. Two machine vision views of the print head holding patterns of spheres are shown in Figure 5.7.
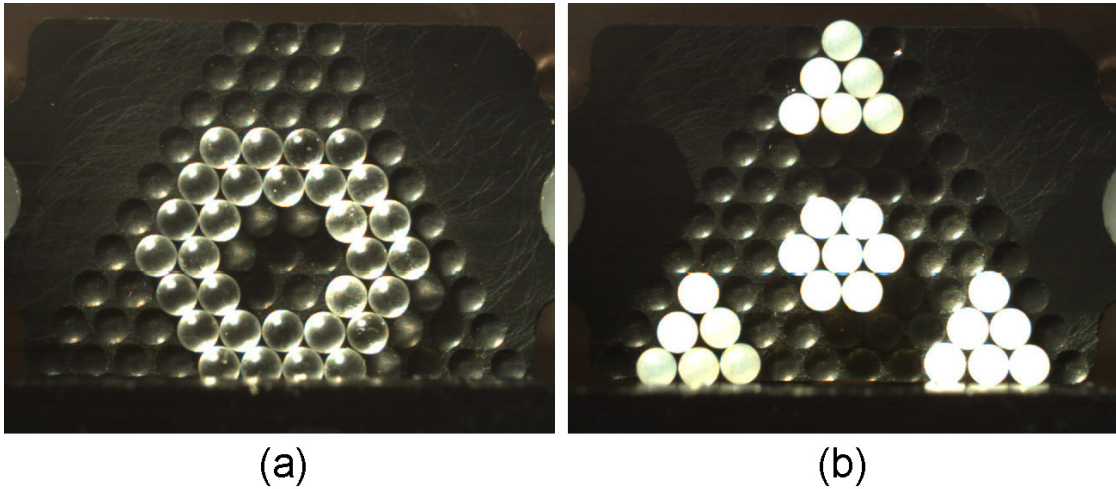
Figure 5.7: Machine vision is used to verify the pattern of voxels that is picked up for a robust, closed loop deposition process. Pictures are captured from below with print head holding spheres.

5.3 Electrostatic Parallel Methods

5.3.1 Concepts

Electrostatic forces are routinely used to robustly manipulate toner particles over millions of pixels in laser printers. Here the use of electrostatic forces is considered to manipulate discrete voxels. The basic concept of electrostatic particle manipulation is shown in Figure 5.8. Because opposite charges attract, voxels will be attracted to a print head with opposite effective charge. This situation can be generated in a number of ways. In the case illustrated here, grounded conductive spheres will be attracted to charged cells on a print head by charge separation. A dielectric layer preserves the electric charge difference while maintaining close proximity. A voxel cradle layer holds each sphere in a discrete position relative to the print head.
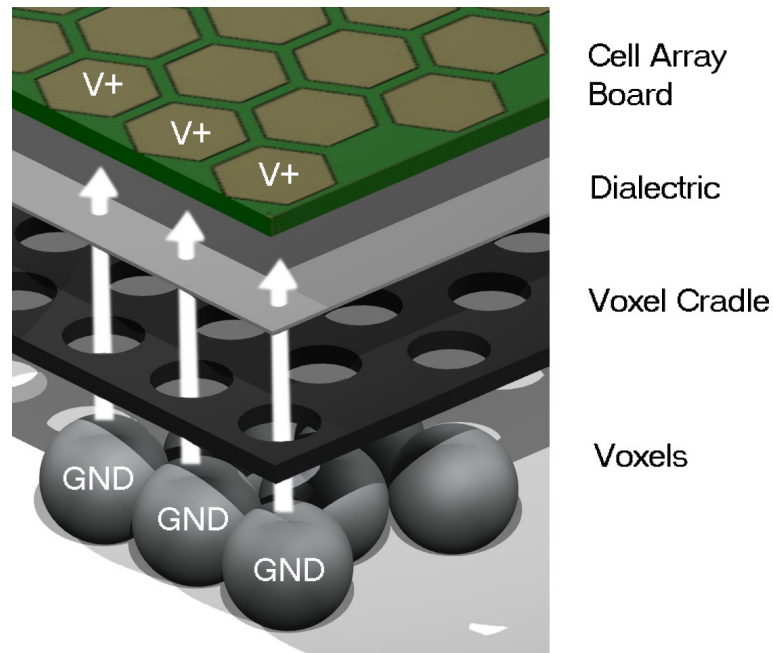
Figure 5.8: The basic scheme of electrostatic voxel attraction for conductive spheres. Charged cells with a dielectric layer over them induce charge separation in conductive spheres, which promotes attractive electrostatic forces.

For dielectric spheres the situation is even simpler. Because charge is essentially not mobile on the surface of dielectric spheres, a dielectric layer can simply be statically charged and spheres will be attracted to it. If the voxels are physically constrained by a cradle layer, only the small contact area will discharge and the rest of the static charge will provide adhesive force. If the dielectric surface is a photo conductive material with a ground plane behind it, charge can be selectively bled off via optical addressing, as is used in xerography and laser printers.

Relative scales are important to consider in order to effectively use electrostatic forces to manipulate voxels. (Figure 5.9) Here, 1.5mm spherical voxels made of acrylic are considered, although the analysis and results hold in principle for any non-conductive material. The analysis of voxels made of conductive
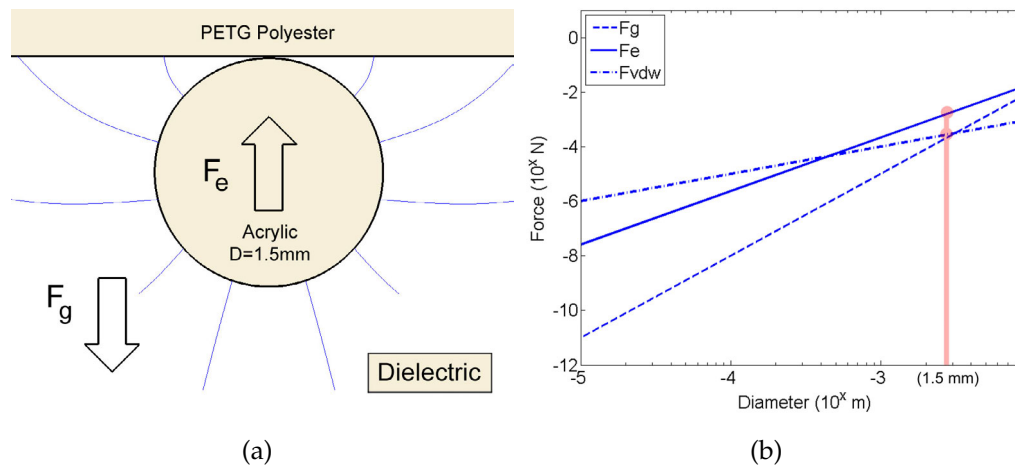
Figure 5.9: Electrostatic force F_e must be significantly greater than gravitational force F_g to effectively manipulate a dielectric sphere (a) which is the case for 1.5mm acrylic spheres (b). Van der Waals force F_{vdw} may also aid attraction. (Figure (b) adapted from Arai et al, 1996)

material is similar but not presented here. At the millimeter scale, the magnitude of electrostatic forces on a sphere can be more than an order of magnitude greater than the gravitational force. As the size of voxels shrinks, gravitational forces become less significant with respect to electrostatic forces.

An electrostatic voxel deposition print head has been developed to selectively pick up voxels from a pre-aligned layer from above, lift them out, and deposit them on a build stage (Figure 5.10). For these experiments, a layer consists of 55 voxels arranged within an equilateral triangular area, and the print head has the identical pattern of cells that can attract a single voxel each. The print head consists of several parts: a voxel cell array which maintains the relative positions of each voxel, the separation layer which may either insulate or conduct charge to the ground plane, the ground plane, and the rigid body for mounting the head the motion system. An insulative separation layer made of PETG polyester was used here.
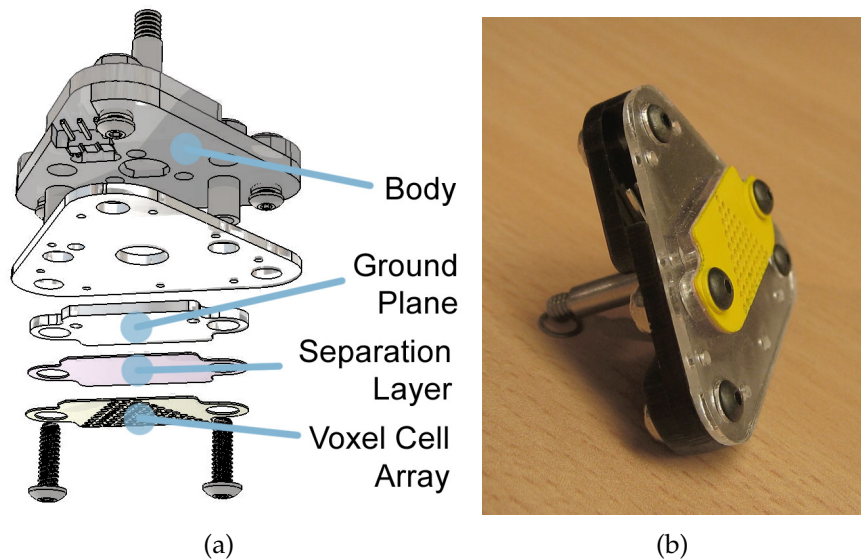
Figure 5.10: An exploded view of the electrostatic print head (a) and an assembled view (b) show the layers used to pick up voxels using electrostatic forces.

5.3.2 Electrostatic Sphere Attraction

For this analysis, an insulating separation layer thick enough to reduce any effects of the mobile electrons in the ground plane is considered. The problem of attractive force between a sphere (conducting or dielectric) and a plane (conducting or dielectric) has been well studied [8, 178]. In the case of these experiments, a dielectric voxel should be attracted to a semi-infinite dielectric plane [201].

To successfully lift a sphere, the attractive electrostatic attraction force between the sphere and the print head must be greater than the force that gravity and the print head acceleration can exert. Each 1.5mm diameter acrylic sphere weighs approximately 2.4mg. Under gravity and 1G opposing acceleration of the print head, $40\mu N$ of electrostatic force would be necessary to maintain contact with the print head. The attractive force between a sphere and a semi-

infinite dielectric ground plane can be calculated approximately by solving the appropriate equation for the given conditions [201].

$$F = \int (\rho E_z - \frac{1}{2}E^2 \bigtriangledown \varepsilon)dv \qquad (5.1)$$

An electric field intensity of 72Kv/m was used for the calculations, coinciding with values measure during experimentation. Representative dielectric constants of 3.2 for the acrylic spheres, and 2.6 for the PETG separation layer were assumed, and the maximum surface charge density on the spheres in air of $26.5\mu C/m^2$ was used. For the 1.5mm diameter spheres, the subsequent calculations yielded approximately 25μN of electrostatic force, which is less than the required 40μN. However, this does not account for the nesting effect of the voxels within the cells of the print head, which has the effect of increasing the electrostatic attraction force due to more surface area of the sphere in close proximity to the head. Additionally, van der Waals forces may also contribute a measurable amount of attractive force. Combined, these effects could theoretically provide the necessary force to explain the observed experimental behavior. More detailed simulation will be necessary to confirm this. However, it is evident that spheres much larger than 1.5mm would be not well suited to electrostatic manipulation, whereas scaling the spheres down only increases the relative magnitude of the electrostatic force as compared to gravity.

5.3.3 Electrostatic Charging Process

The VoxLayer rapid assembler was equipped with a corona charging station and an electric field meter to perform the electrostatic experiments (Figure 5.11). The
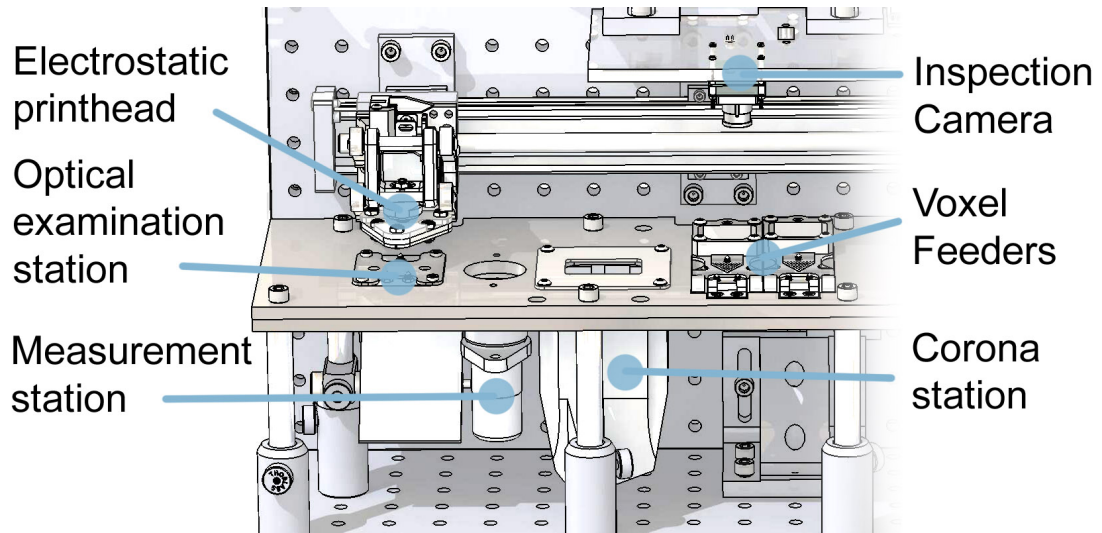
Figure 5.11: Schematic of Voxlayer as equipped for electrostatic sphere pickup experiments, including a corona wire charging station and an electrostatic field meter.

flow of electrostatic voxel manipulation for the digital fabrication process is as follows:

1. **Electrostatic charging:** The print head moves to the static charge station (Figure 5.12a) and a corona wire housed below emits a stream of ions to impart a static charge on the print head. To obtain a selective charge, a physical barrier was placed between the desired cells and the corona emitter.

2. **Layer self alignment:** Meanwhile, an entire layer of voxels is self aligned in the voxel feeder (Figure 5.12b) using gravity and vibration.

3. **Selective parallel voxel pickup:** The electrostatic print head moves over the appropriate voxel feeder and presses down uniformly on all voxels (Figure 5.12c). When it lifts up, the voxels corresponding to the charged cells are carried along.
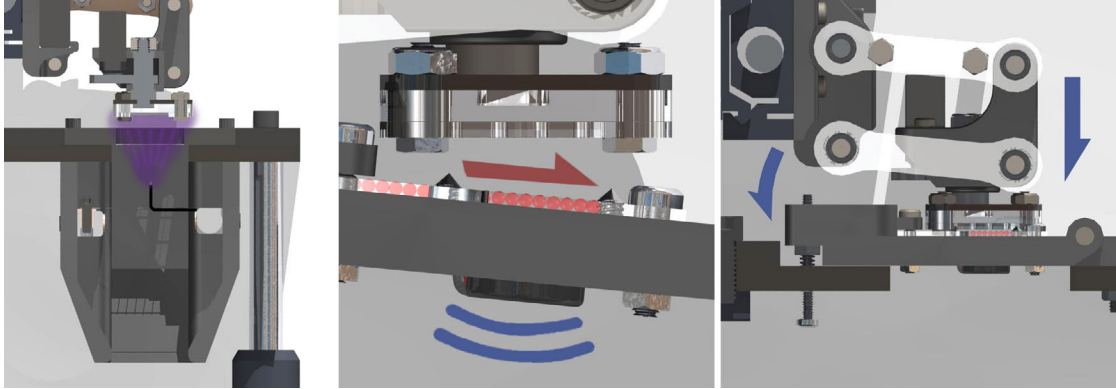
Figure 5.12: Illustrations of voxel manipulation steps in the electrostatic parallel process. (a) Charging the print head at the corona station, (b) aligning a layer of voxels in the feeder, and (c) picking up a layer of voxels.

This process requires a series of stations that the electrostatic print head must move between in order to complete the layer deposition process (Figure 5.11). The corona station houses the high voltage corona discharge system, and is insulated from its surrounding by an ABS plastic casing approximately 3mm thick. Another station houses a voxel feeder, where the voxels are picked up. Cone-headed indexing pins with adjustable height are present at each critical station that precisely align with indexing holes on the head as it presses down, ensuring accurate positioning.

A separate station houses a static electricity meter (Ultrastable Stable Voltmeter, Alphalab Inc., 0-20Kv range). Although not in the normal build sequence, this station was used extensively to obtain data on the relative charge of the print head by measuring the electric field emitted by it. When measuring a non-conductive surface, this particular sensor outputs a number proportional to the emitted electric field. Since the electrostatic print head does not fill the entire sensing field of view, a ground plane was introduced to mask any unwanted

measurements. The electric field due to the print head was then calculated by dividing the total sensor measurement by the percent area of the print head. All subsequent data on the charge of the print head is presented in terms of the emitted electric field.

In the first step of the layer deposition process, a corona wire is used to impart a static charge on the print head surface. Corona discharge occurs when the strength of an electric field increases to the point where the surrounding air is ionized, but an arc does not yet form. To obtain this in practice, a bare conductor in the shape of a needle is charged to high voltage, pointing upwards towards the print head. The ions that form at the end of the needle are then attracted to a ground plane, drawing them up towards the print head. Although other methods exist to obtain a high static charge, such as triboelectric effects, corona discharge was selected to obtain a uniform static charge on a wide variety of materials.

A 35Kv DC voltage source using a Cockcroft Walton voltage multiplier (capacitor ladder) was used to power the corona wire with a maximum current of 1.5mA. In these experiments, the open circuit (no load) voltage of the ion emitter tip was varied between 6 and 9 Kilovolts. This alone is enough to ionize the surrounding air, but a flow of ions is set up in the presence of the ground plane housed in the print head, causing a net migration of charge through the air. If a non-conducting surface such as the separation layer is placed in this ionic wind, a static charge will be imparted on it. As the voltage per unit distance rises, the air begins to break down and conduct increasingly more charge until the point of arcing, which is to be avoided.

A series of experiments were conducted to characterize the charging process of the electrostatic print head. Variables considered include the voltage of the corona wire, the distance between the emitter tip and the print head, and the time of charging. Other variables that were not considered include the geometry of the corona emitter tip, the thickness of the separation layer, and the ground plane resistance, which were held constant for this analysis. A positive relationship was observed between the emitter tip distance and resulting static charge on the print head, as shown in Figure 5.13 for a variety of corona voltages. This distance was measured from the tip of the emitter to the bottom plane of the print head. The exposure time was held constant at one second. This relationship can be explained by noting that as the corona tip approaches the print head, the voltage per distance rises. This allows the air to break down and conduct a greater amount, thus lowering the difference in potential, resulting in less charge on the head.

A positive relationship is also observed between the time the corona wire is emitting and the amount of static charge resulting on the head. (Figure 5.13) This is plotted on a logarithmic scale with exposure time varying from 25ms to 5 seconds. However, the maximum charge levels off very quickly as exposure time increases, and there are no significant gains in the measured electric field of the head after approximately one second of exposure.

5.3.4 Selective Electrostatic Parallel Voxel Pickup

To demonstrate selective parallel voxel pickup, full layer pickup was demonstrated using the electrostatic print head. Figure 5.14 shows three frames from
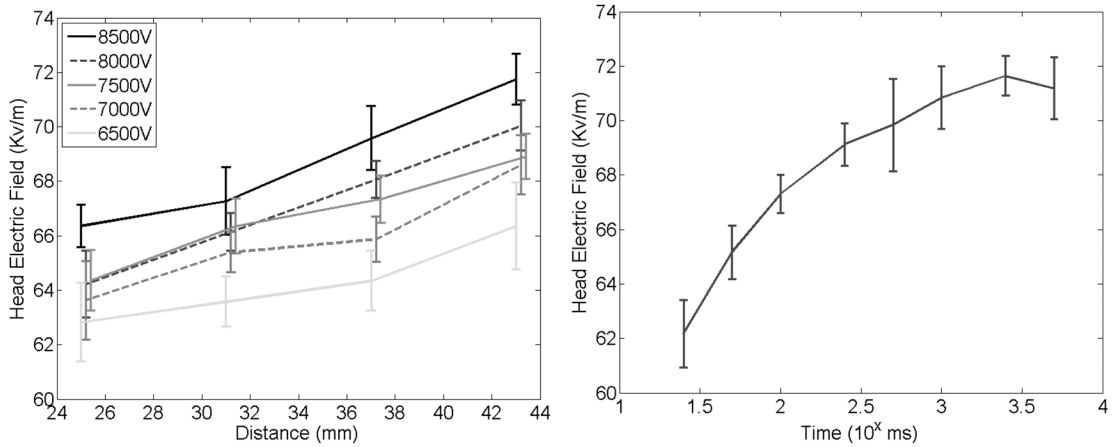
Figure 5.13: (a) Print head charge as a function of corona emitter distance for several corona voltages (left). (b) Print head charge as a function of corona exposure time.

a movie as the print head moves over, presses down, and lifts up with the majority (all but one) of the spheres. The camera was located below the voxel feeder pointing upwards, and the spheres are seen through its transparent bottom. This trial used a contoured print head interface consisting of five laser-cut layers with contours that fit each sphere in order to increase the surface area of the sphere-head interface.

However, not every sphere that was initially lifted from feeder stayed attached as the head moved away. This can be attributed to relatively harsh accelerations and vibrations from the stepper-motor driven X stage, as well as the static charge bleeding off to the surrounding air in the presence of humidity. For a relative humidity of 45%, a representative electrostatic discharge curve is shown in Figure 5.15. The experimental time constant is approximately 13 seconds. Given the time to place a layer is less than 3 seconds, and the most extreme forces are felt at the beginning of this, this effect can be accounted for.
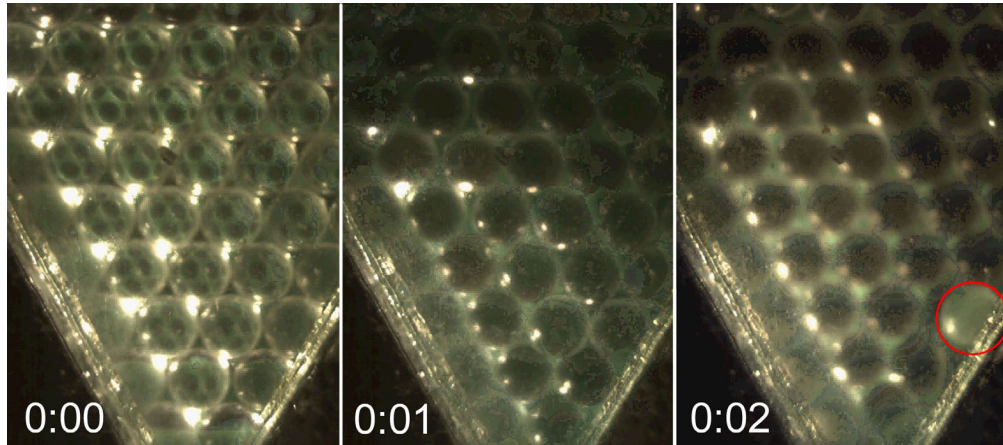
Figure 5.14: Electrostatic voxel pickup using 1.5mm acrylic spheres, viewed from below. Print head moves over aligned voxels (left), presses down (center) and lifts up all voxels except one (right, circled). The out-of-focus spheres have been lifted up from the feeder away from the camera.
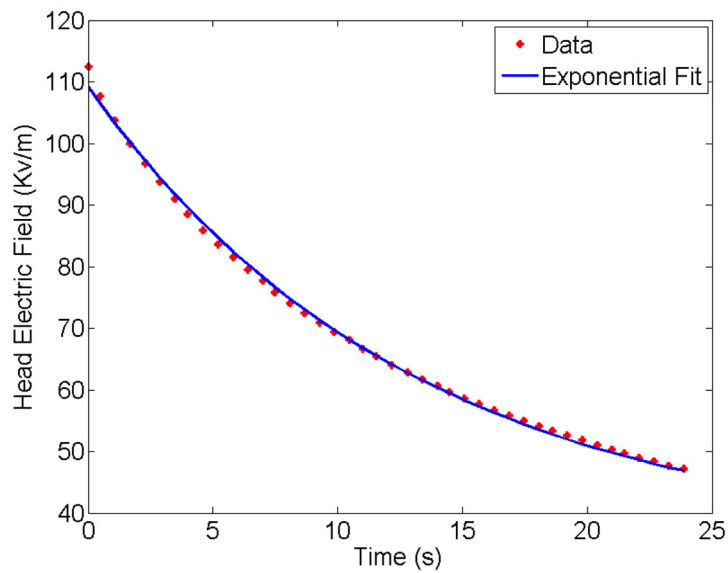
Figure 5.15: Relative charge loss of print head as a function of time. The experimental time constant is 13 seconds, which is sufficient to place a layer.
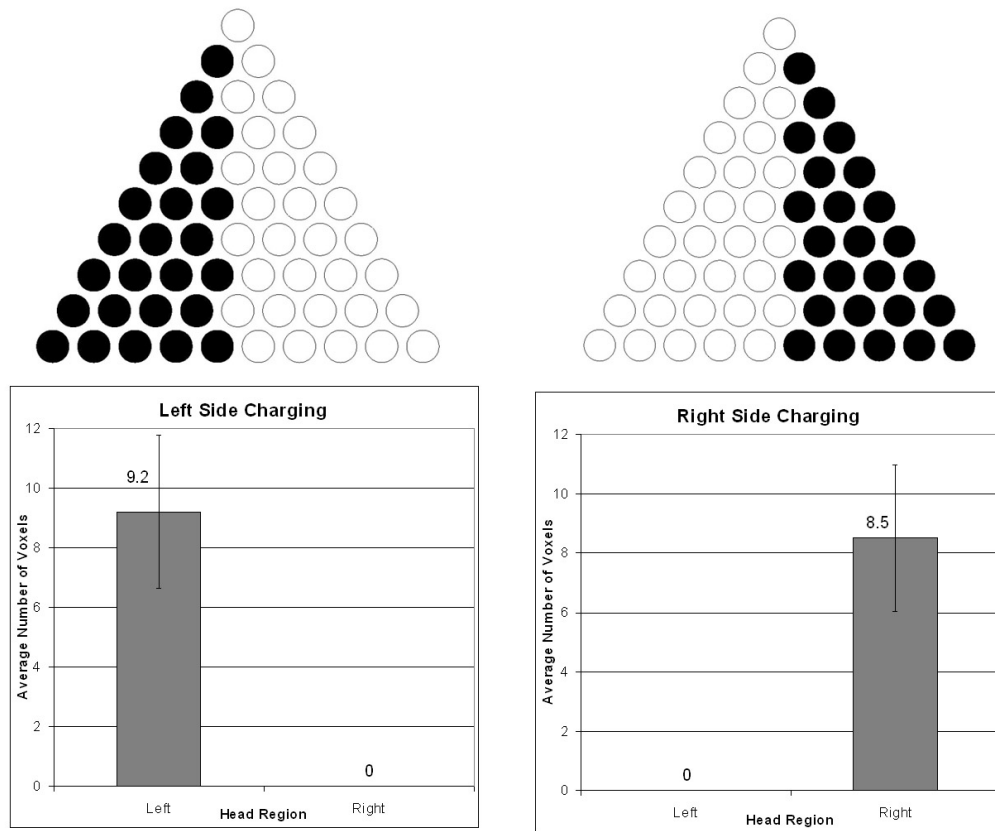
Figure 5.16: Results of voxel pickup experiment demonstrate selective manipulation of 1.5mm acrylic spheres using solely electrostatic forces. Spheres were never erroneously picked up, but only 35% of the desired spheres were picked up on average.

Next, custom masks were made for the corona charging station to physically block the charge from selected cells. A separate mask was made to block each half of the 55 cell triangle. Ten trial runs were done for each case. The number of voxels suspended in place on each half of the print head was recorded after it moved away from the feeder station (Figure 5.16). It was observed that a sphere was never erroneously deposited from an uncharged cell. Within the charged region, the number of remaining spheres was consistently less than the total possible 25 cell locations, although still a significant percentage.

93

5.4 Opto-Capillary Parallel Methods

In order to more robustly manipulate spherical voxels at the millimeter scale, fluidic capillary forces were explored as an alternative to electrostatic forces. Instead of electrostatic forces holding each voxel to the print head, a discrete drop of water was utilized. Using this principle, automated parallel assembly of 220 1.5mm spheres into a variety of multi-material 3D structures was demonstrated. This technique is expected to scale well to the assembly of large numbers of elementary building blocks into complex integrated 3D structures.

5.4.1 Selective Drying

First, a flat print head containing a pattern of divots corresponding to lattice nodes was uniformly wetted (Figure 5.17a) using a solution of water mixed with alconox laboratory detergent. The solution was combined at the recommended concentration of 10 grams per liter to form a non-hazardous, non volatile solution with excellent wetting properties. This solution was chosen to enable favorable wetting characteristics. The desired bitmap of voxels was then transformed into a black and white image of a pattern of dots that each coincide with a cell on the print head.

Selective drying was carried out using a DLP-based office projector with several modifications and an external optical system. The base projector used is an Optoma EP-719, rated at 2000 ANSI lumens, a 2500:1 contrast ratio, and a native resolution of 1024x768 pixels. Modifications were carried out with two goals: (1) increase throughput of infrared wavelengths for quicker drying times and
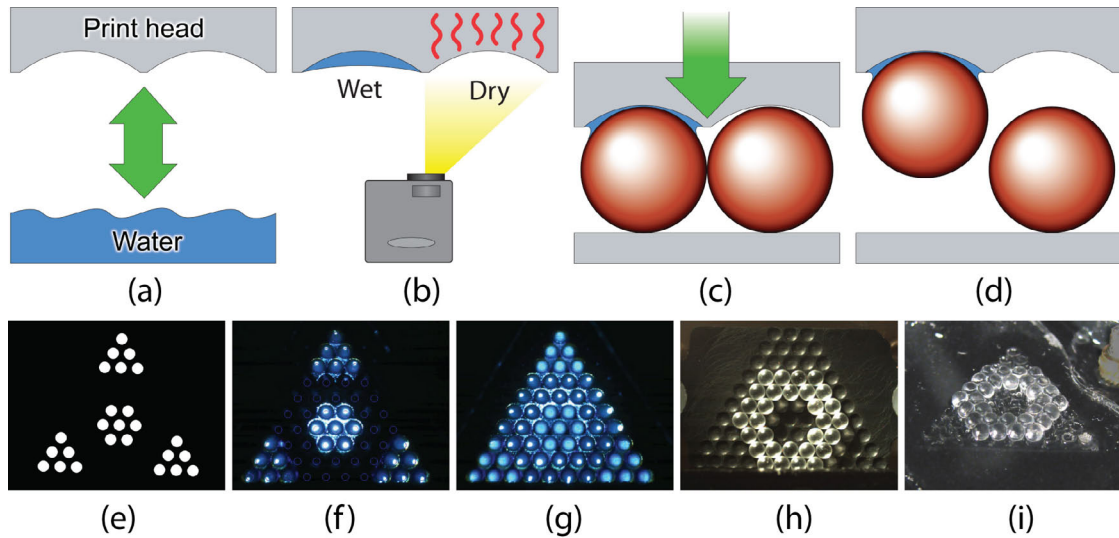
Figure 5.17: Method of selective pickup for opto-capillary parallel voxel deposition. The entire deposition head is uniformly wetted (a). A projector selectively illuminates and dries certain cells (b), and the deposition head presses onto the pre-aligned voxels (c). Voxels corresponding to the active (wet) cells are lifted and carried to the build stage (d). In the actual implementation, a pattern of dots is specified (e), drying the corresponding cells on the print head (f) leaving the others wet, or active (g). The voxels are lifted (h) and placed on the build stage (j).

(2) project an undistorted image on the area of the print head (approximately 1/2″ wide). To meet the first goal, the color wheel was removed, along with the infrared filter directly in front of the arc lamp. The light pipe mirrors were also changed to polished aluminum to handle the additional heat load. To meet the second goal, the primary (projection) optic assembly was removed. Using two 50mm diameter plano-convex lenses external to the projector, an image of 1/2″ width was clearly projected at a linear distance of approximately 9 inches from the projector. A 45° angle mirror reflected the image onto the bottom of the downward-facing deposition head. The desired pattern of cells dries in approximately one minute, leaving the remainder of the cells wet, or activated (Figure 5.17b).

To pick up the voxels, the selectively-wetted print head presses down onto the aligned spheres (Figure 5.17c). The water in the active divots wets around the perimeter of its respective sphere, holding it in place by the capillary effects of surface tension. The deposition head then lifts and carries only these selected spheres to the build stage (Figure 5.17d). A liquid polyvinyl acetate binder was used to bind the structure together during the build process. The layer of binder is spread on the existing printed object, and the current layer is deposited. Each sphere falls into the interstitial region of the three spheres below it and is held by the adhesive properties of the binder as the deposition head moves away.

5.4.2   Real Time Selective Drying Feedback

Several uncontrolled parameters necessitated real-time feedback on the the drying process in order to robustly obtain the desired wetted pattern. For instance, the output of the projector is not constant across the field of view, and the quantity of water in each cell is not precisely controlled. If a given cell is heated for too long after it has dried, it can also begin to dry the cells around it, which is undesirable. For these reasons, a closed loop drying process was implemented.

Machine vision was used to monitor and actively control the drying process. To calibrate this machine vision view, at the beginning of a build sequence the user is asked to verify that the surface of the deposition head is dry and free of visible contaminants. Then, a series of 5 reference images is captured and averaged (to reduce pixel noise). The user also manually identifies the outer divots of the triangular pattern. A virtual pattern of divot positions is thus indexed and stored.
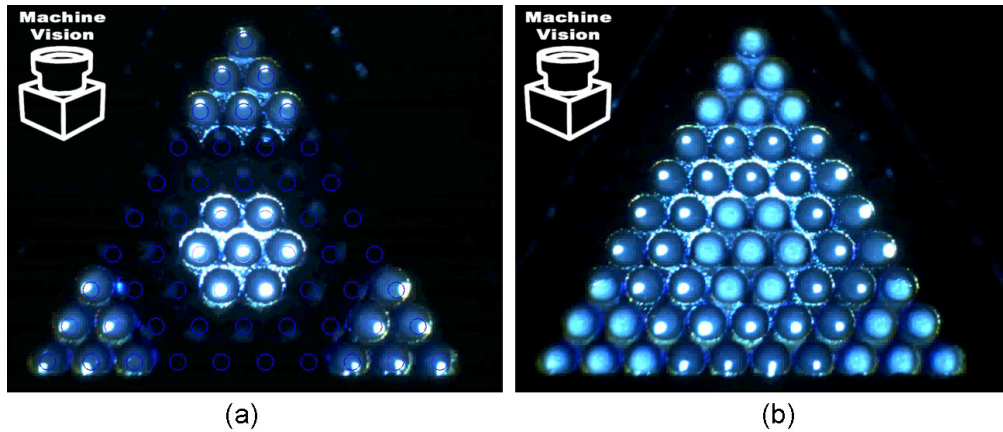
Figure 5.18: During selective drying process, a pattern of dots illuminates certain cells on the print head (a), drying them in approximately 60 seconds (b). Machine vision monitors the transition from wet to dry and adjusts the exposure of each cell independently.

After the print head has been wetted, the drying process begins. After the desired dots have been turned on, the machine vision algorithm continually captures images. A region of approximately the size of each divot around each position is subtracted pixel-wise from the reference dry image. Because of the specular difference between water and the milled ABS plastic deposition head surface, a large average pixel error is observed when the cell is wet, gradually reducing until it falls below a threshold that signifies that a cell is dry. Each cell is monitored independently and switched off once it is dry. Representative machine vision views are shown in Figure 5.18. An additional machine learning stage was also incorporated to learn the delay in turning on the light for each cell to account for the varying dry times. A series of 3-5 drying experiments is conducted during a pre-calibration routine. After each iteration, the delay of each cell is adjusted based on how long it takes to dry compared to the average.

Using the closed loop selective drying, the print head could repeatably be

patterned with any arbitrary pattern of wet cells. The results of the VoxLayer fabrication process are outlined in the following chapter.

5.5   Conclusions and Future Directions

The feasibility of selectively manipulating 1.5mm spherical voxels using both electrostatic forces and capillary forces has been demonstrated here. However, at this scale capillary forces performed much better and more reliably than electrostatics. All aspects of these experiments are designed to be parallel in nature and scale effortlessly to layers with arbitrarily large numbers of voxels. Thus, this research forms the technical foundation for a digital 3D printer capable of creating multi-material structures with millions of voxels.

Future work for the electrostatic process involves refining the experimental conditions and parameters such that individual voxels can be addressed and manipulated robustly. The separation layer will be switched to photosensitive material. This would allow the entire print head to be uniformly charged, then moved to an optics station where unwanted voxel locations would be exposed to light from a scanning laser or computer projector. This allows the static charge to bleed to the ground plane behind, allowing for any arbitrary layer of voxels to be picked up. The photosensitive material can also be uniformly illuminated in order to release the voxels once they have been stacked on the 3D assembly stage.

In order to scale the opto-capillary voxel deposition process, either the size of the voxels must decrease or the power of the projector must increase. In the current iteration, the time for each layer to dry accounts for the vast majority

of the time it takes to perform a build. However, an increase in heating rate could increase the sensitivity of drying out neighboring cells. Thus, it would be desirable to maximize the thermal resistance between each cell on the print head. The use of a high-powered scanning laser would also be suitable for fast selective drying.

These parallel methods are unlike any existing processes in that the advantages of both self-assembly and top-down assembly are combined in a single process. The ability to selectively place an entire 2D layer of discrete, aligned elements has not been demonstrated previously in literature, and has significant implications regarding the scaling of such processes to many voxels.

CHAPTER 6

DIGITAL OBJECTS: FABRICATED, POST-PROCESSED, AND RECYCLED

6.1 Rapidly Assembled Digital Objects

While the efficient arrangement of voxels into a 3D lattice is the primary techni-
cal challenge of fabricating digital objects, the question still remains of how to
form useful, freeform shapes and the implications on the total life cycle of digi-
tal parts. If the voxels interlock, post-processing is needed only to remove any
support voxels that were in place to facilitate fabrication. However, the physical
platforms demonstrated in the previous chapters make use of spherical voxels
which do not interlock.

In this chapter several digitally fabricated objects of different materials are
shown as examples of the VoxLayer and VoxJet platform capabilities. Both the
serial/1D parallel, and 2D parallel processes were compared on their ability to
scale to objects with many millions of voxels. Two post-processing methods
were explored to enable freeform plastic and steel objects. Finally, the recycling
potential of digital materials is explored to leverage the inherent discrete nature
of such objects.

6.1.1 VoxLayer

Using the parallel 3D voxel printer a variety of structures were fabricated us-
ing different combinations of metallic and non-metallic materials (Figure 6.1).
These include a simple electrical network constructed of copper and nylon vox-

els. Freeform geometries were also created from stainless steel, using acrylic spheres as a sacrificial support material to create overhanging regions in steel voxels. Upon completion, the printed structure was sintered to burn out the acrylic and bind the steel voxels together into a robust, freeform part. This actual implementation of parallel voxel fabrication has a maximum workspace of only 220 voxels, so the resulting objects are relatively low resolution.

6.1.2 VoxJet

The serial 3D voxel printer was used to fabricate much larger and complex digital objects with as many as 22,140 voxels. The size is limited primarily by the dimensions of the egg crate base that the spheres are deposited on. The primary materials chosen to be utilized in the VoxJet fabricator were opaque white acrylic as build material and stainless steel as support.

Due to the constrained sources of procuring tens of thousands of 1/16" spheres, the difference in mean diameter of these two materials of spheres was around 0.001". In large builds, this is not a problem if the two materials are well distributed spatially. However, this is not usually the case when constructing arbitrary geometrical shapes where the build material spheres are grouped into a common shape. This leads to non-planar layers as the height of the build increases, disrupting the accuracy of the deposition system. To solve this problem, the regions of material used as support can be interlaced with spheres of both materials to keep each layer approximately planar. For example the knight in Figure 6.2 shows the intended object surrounded by support spheres, but much of the support region is filled with alternating build and support material.
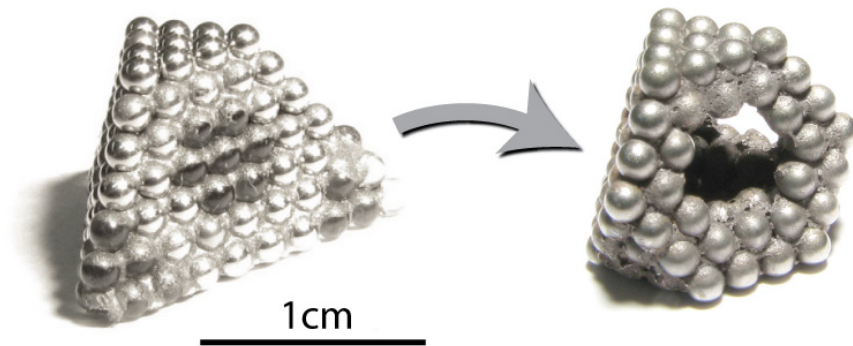
Figure 6.1: Digital object of various materials demonstrate the flexibility of the materials that can be utilized by the VoxLayer parallel rapid assembler. Examples include a freeform stainless steel structure, a brass and steel structure, and a simple electrical network from copper and nylon voxels. The current parallel implementation is limited to a pyramid envelope with a base of 10 voxels.
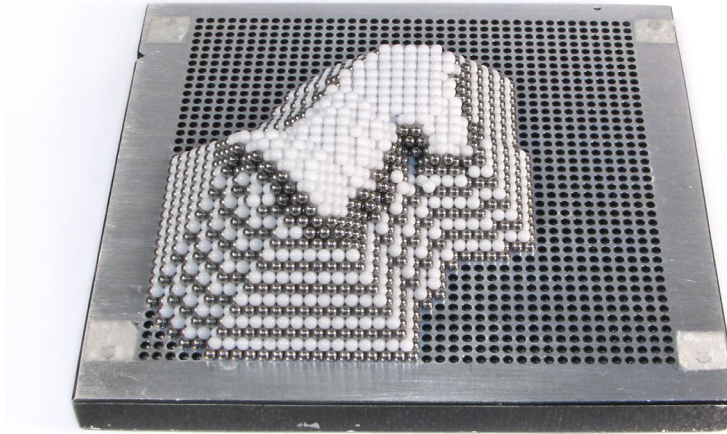
Figure 6.2: In this knight build, supporting regions are mixed with both build and support spheres to keep layers planar in the face of slightly differing mean diameters of the metal and plastic spheres.

Several examples of objects built with the VoxJet platform are shown in Figure 6.3. With a base size of 40 voxels square the maximum workspace involves 22,140 individually placed spheres. Figure 6.3(a) shows a pyramid fully utilizing the build space with an internal material gradient transitioning from steel on the bottom to acrylic on the top. To illustrate the deterministic placement of each voxel, a logo was imposed on the front surface. Two types of chess pieces were also fabricated. Each piece has a steel core enclosed in a plastic boundary that serve to weight the pieces in a pleasing manner. The knight is composed of 1441 voxels and required 5700 voxels to print including support. The rook is composed of 2400 voxels and required 10,006 units to construct including support.

(a)

(b)

(c)

(d)
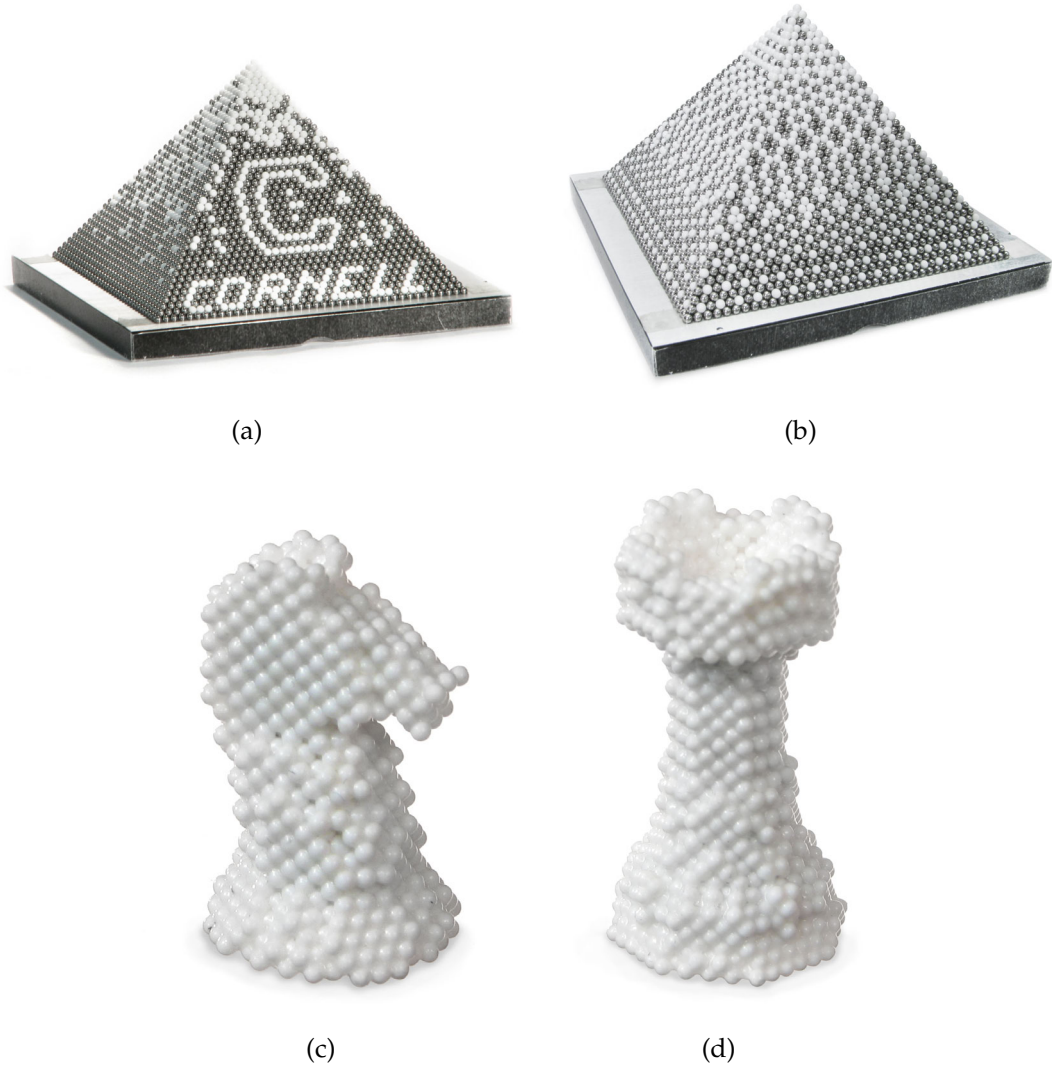
Figure 6.3: Objects fabricated with the serial Voxjet rapid assembler include two 22,140 voxel pyramids with internal material gradient (a & b*). Knight and rook chess pieces (c & d) required 5,700 and 10,000 voxels respectively to construct including necessary support material.

*Manual assistance used during this specific debugging build.

6.2 Scaling Laws of Digital Fabrication Processes

The performance characteristics of the serial and parallel implementations are shown in Table 6.1. As a comparison, a high performance pick-and-place benchmark system that can place one element per second was considered. The serial process was configured to deposit approximately 8 elements per second when depositing in a continuous line. However, when accounting for re-positioning the deposition head and error correction, the average deposition rate is around 2.5 elements per second. The parallel system places one material of one layer of up to 55 elements in approximately 285 seconds. Therefore it takes 90 minutes to complete a full two-material 10-layer build. The length of time needed to complete a single layer is independent of the number of elements in a given layer but rather is limited by the time needed for the drying process.

A fundamental characteristic that distinguishes rapid assembly processes from traditional pick-and place assemblers is their ability to scale favorably to large numbers of building blocks. At small numbers of modules, a fast pick-and place method may be superior. But as the number of elements grows, both the serial and parallel methods scale favorably with regard to the time to place each element. The serial method can be adapted to deposit an entire line of material simultaneously. The parallel method naturally scales still more favorably because increasing the number of elements per layer does not increase the time per layer. The number of building blocks per layer can scale to large sizes and resolutions limited only by the current optical addressing technology, making the selective deposition of 1000×1000 building blocks in a single pass within reach. All three methods can be further accelerated by adding multiple print heads or assembly arms working in parallel, and using hierarchical approaches.

Table 6.1: Assembly rate scaling characteristics for rapid assembly implementations. As implemented, the serial assembly approach is much faster per element, but the parallel approach has a lower error rate and scales more favorably to objects with many more elements.

	Fast pick-and-place (baseline)	Serial implementation	Parallel implementation
Minimum time/element	1s	0.13s	5.16s
Average time/element	1s	0.43s*	27.47s*
Error rate	N/A	1.57%	0.91%
Scalability (as function of number of elements n)	$O(n)$	$O(n^{2/3})$	$O(n^{1/3})$
Expected time to assemble one billion building blocks in a 1000-sided cube[†]	30 years	42 hours[‡]	80 hours[§]

[*] Includes error correction time

[†] Excludes potential error correction time

[‡] Assumes identical deposition and feed rates as the demonstrated implementation with a deposition module per row

[§] Assumes identical time per layer as demonstrated implementation

Although self assembly methods can form entire 3D structures more or less in parallel, it is impossible to specify a material at each location in the lattice. All other previously demonstrated methods of creating large-scale lattices otherwise rely on serial pick-and-place approaches that do not scale well, and are therefore unsuitable for a mass digital fabrication process.

6.3 Post-Processing Techniques

Because spherical voxels do not physically interlock, a variety of post-processing techniques were explored to realize robust physical objects. There are many such techniques that can be used with various combinations of build materials. Of course, glue can simply be added between layers. But without selectively placing the glue, it is difficult to separate the finished object from the support spheres that must be placed around it. To address this challenge two methods for creating freeform digital objects were demonstrated. The first makes use of acrylic spheres as a build material and stainless steel as a support material without any binder during the build process. The second method uses stainless steel as the build material and acrylic as the support material. A binder is used equally between all spheres of both materials in this case.

6.3.1 Vapor Fusing for Acrylic Voxels

The first method uses a simple chemical fusing process to form structures from acrylic spheres with metal spheres as intermediate support. In this process, no liquid binder is necessary during the build process. However, the structures were periodically wetted with water during the build process to obtain temporary surface tension adhesion forces for robustness.

Once the build is complete, the structure is thoroughly dried in a passive drying oven at 65° C overnight to evaporate any remaining water. The structure is then placed in an enclosed chamber with an open petri dish of acetone. Here, the internal volume of the container was approximately 1300 cubic cen-

timeters and 2cc acetone was sufficient to maintain vapor saturation within the enclosure. The volatile vapors infuse the acrylic spheres to soften them slightly, allowing touching spheres to bond. After 100 minutes, the interfaces between touching spheres are bonded without significantly deforming the overall geometry. Following 30 minutes in fresh air to allow the latent acetone to evaporate, the acrylic returns to its rigid state. The bonded plastic part is then removed from the metal support spheres which are unaffected by the acrylic vapors. Although the resulting parts are nominally able to be handled in this state, they are relatively fragile and therefore are infused with glue for increased robustness.

This process allows for freeform plastic parts of arbitrary geometry to be fabricated without needing a separate binder during the print process. This simplifies the necessary complexity of the digital fabricator significantly. The vapor fusing process utilizes materials that are extremely inexpensive and commonly available. Multi-material objects can be realized in certain situation. For instance, non-bonding steel spheres can be contained in the interior of a plastic part to increase the total mass.

6.3.2   Sintering Process for Stainless Steel Voxels

The second process requires more infrastructure, but can form robust freeform stainless steel structures. Here, the stainless steel (alloy 302) spheres were used for the desired final object and acrylic spheres were used for sacrificial support material. A binder was prepared using polyvinyl acetate (white glue) and stainless steel 302 powder with particle size less than 44 microns. Powder was gradually mixed in until the point just before it began to solidify the mixture.

The digital object is then built as normal using this prepared binder between each layer. After a long drying period (24 hours or more) the object was inserted into a tube furnace for burnout and sintering. The entire operation was performed under a flow of inert gas to reduce charring from the burnout. The temperature profile is detailed in Table 6.2. A slow ramp up to 200° C ensures that moisture is evaporated and begins to soften the acrylic. After holding this temperature for an hour, another slow ramp takes the temperature up to 550° C. At this point, the acrylic is burned out but the binder is still holding the part together. After holding for an hour here, the temperature is ramped up to a final sintering temperature of 1380° C, at which the stainless steel powder in the binder sinters the voxels together. This elevated temperature is held for one hour as well, then gradually brought back down to room temperature.

This process results in parts that also retain their discrete nature, as the spheres themselves do not melt or deform noticeable. The parts are also mechanically robust and able to be handled and dropped repeatedly. Although the stainless steel powder in the binder does not ever fully fill in the voids between the spheres, it tends to form a semi-solid region around the interfaces between the spheres.

6.4 Recycling Digital Materials

These post-processing techniques tend to be irreversible, but this is not necessary. If the bond between voxels are reversible, or if mechanically interlocking voxels are utilized, the resulting structure can always be decomposed into fundamental elements which can then be re-used in subsequent builds.

Table 6.2: Parameters for sintering freeform stainless steel structures while simultaneously burning out acrylic support material.

Step	Ramp (deg C/hr)	Temperature (deg C)	Hold (hrs)
1	500°	200°	1
2	500°	550°	1
3	500°	1380°	1
4	300°	30°	(Done)

In the current continuous material paradigm, recycling is often a costly and inefficient process, particularly for objects composed of multiple integrated materials. The new paradigm of digital (discrete) matter enables any number of materials to be printed together in any configuration. Coupled with a compatible voxel sorting process, voxels can be easily re-used. Here, multiple generations of freeform fabricated objects were fabricated using the same physical materials. This opens the door to a flexible desktop fabrication process in which 3D multi-material objects are fully recyclable and re-usable with minimal infrastructure.

6.4.1 Challenges of Conventional Multi-Material Recycling

With the advent of multi-material additive manufacturing (AM) processes and the recent push for sustainability among developed nations, the need for recycling of additively fabricated 3D parts has become a critical need for future research [14]. The AM industry is growing steadily [198], but as AM parts see widespread adoption the potential impact has not been thoroughly considered

[204]. Here, the focus is specifically on recycling end-use additively manu-
factured objects, not the full life cycle analysis of additive manufactured parts
[85, 41, 70].

The vast majority of parts created with additive manufacturing processes
today are not recycled. This is primarily for two reasons. First, in many AM
processes the material in use undergoes irreversible changes such as polymer-
ization or infiltration with binder which makes further re-use of the material
intractably inefficient. Second, AM parts made of material which could theo-
retically be recycled, such as those using thermoplastic or metals, are not recy-
cled due to lack of volume, incentives and infrastructure. To recycle these parts
would require an infrastructure involving an energy-intensive melting stage,
along with specialized equipment to return the material to its original form,
such as extruded thermoplastic for fused deposition modeling (FDM) or pow-
der of an appropriate size distribution for laser sintering. Although technically
feasible, this is not common in practice, mainly due to the low volume of AM
parts and the lack of economic incentive.

The challenges of recycling multi-material objects extend far beyond the
realm of additive manufacturing. Vast amounts of electronics waste are labori-
ously and hazardously decomposed by hand in developing countries [67]. More
generally, many manufactured parts consisting primarily of recyclable mate-
rial cannot be recycled because separating the materials is often prohibitively
expensive [168]. A manufacturing process which addresses these challenges
could have significant impact on how waste is handled. Additionally, remote
situations such as space exploration would benefit from completely recyclable
materials [116, 151].

The results presented here should not be confused with the process of recycling material involved in an AM process but unused in the final part. For instance, it is common practice to reuse the leftover resin in a stereolithography build, and it is also possible to recycle the unused bed of powder in power-based processes [139]. The challenges in re-using and recycling these unused materials are wholly different than those needed to recycle the material used to make the actual end-use part addressed here.

6.4.2   Reusing and Recycling Digital Materials

Using the VoxJet research platform configured to deposit a water-soluble binder between spheres, fully recyclable digital materials are demonstrated here. First, two dissimilar materials were chosen to print with, in this case steel and delrin were chosen. It should be noted that this combination of materials could not be printed together using any other process. A two-material dome shape was designed for printing, containing approximately 400 voxels (Figure 6.4(a) & 6.4(b)).

In the first phase, the center region of the geometry was selected to be delrin, while the periphery region was steel voxels. The structure was printed in approximately three minutes, and allowed to dry for 4 hours at room temperature, to allow the interior of the structure to dry. After this, the finished structure was removed from the base-plate. Next the voxels were reclaimed by dissolving the binder (Figure 6.4(c) & 6.4(d)). The entire structure was placed in water and allowed to soak for several minutes. With the help of mechanical agitation, the spheres were separated from each other in solution, then dried. This process

was repeated with clean water a second time to ensure minimal glue residue left on the voxels.

In the case of steel and delrin spheres, separating them from a common pile is trivial with the help of a low-intensity magnet (Figure 6.4(e)). High intensity magnets were avoided to avoid significant magnetization of the steel spheres, which could interfere in the subsequent alignment and deposition process. The remaining spheres in the reservoir for each material were emptied to ensure that the recycled spheres were in fact the ones being printed, then each constituent material was poured directly back into its respective material reservoir (Figure 6.4(f)).

The inverse shape was specified, with steel spheres now in the center region of the geometry and delrin spheres on the peripheral region. This object was then physically printed in a similar amount of time, using the same voxels as the first object (Figure 6.4(g) & 6.4(h)). This demonstrates completed recycling of both materials, discounting the trace amounts of binder which was lost in the process.

6.4.3   Implications

Here recycling has been demonstrated of two dissimilar materials which can easily be sorted using a magnet once the structure is decomposed. The challenges in sorting the voxels for re-use becomes more difficult when three or more materials are used, especially when those material have similar material properties. However, this is not likely to be a significant obstacle to moving forward. Passive sorting based on density will be particularly effective, using
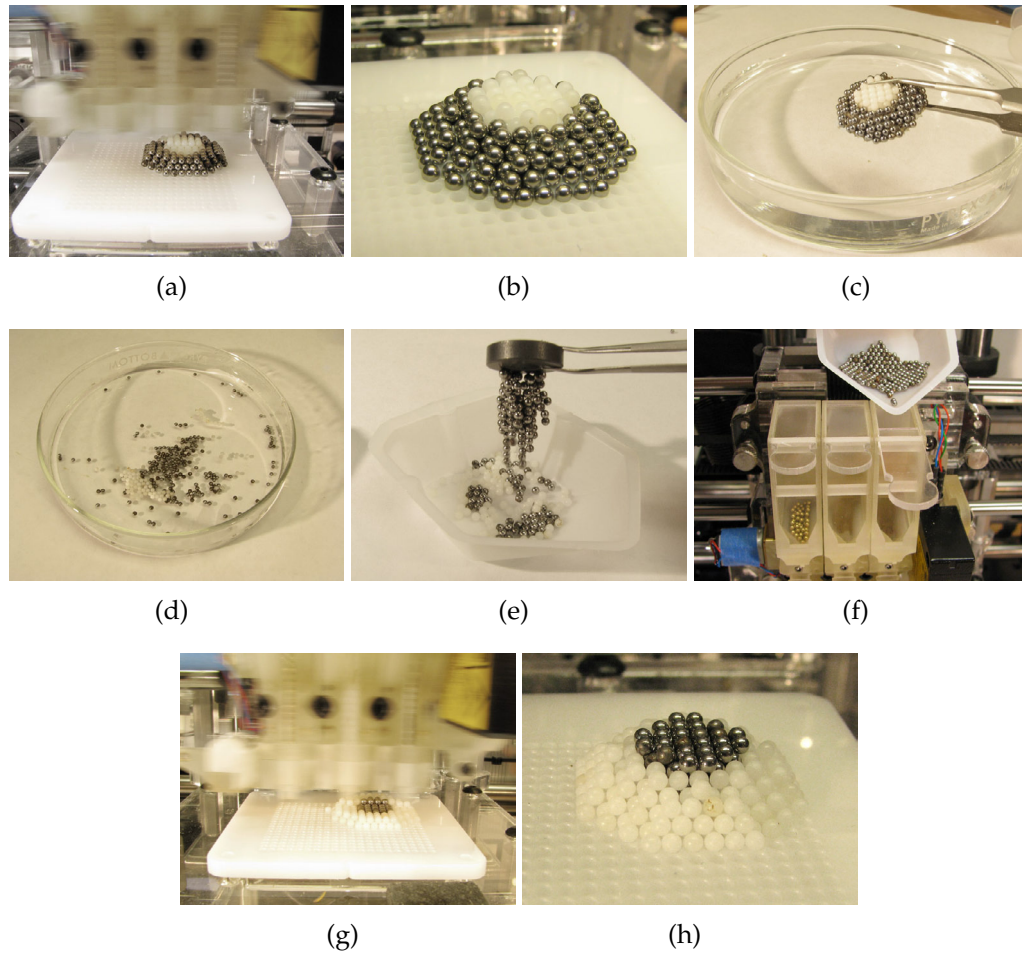
Figure 6.4: The process of recycling printed material: The initial object is printed consisting of both steel and delrin voxels (a and b). The finished object is removed from the base plate (c) and dropped into water to dissolve the bonds holding the voxels together (d). After mechanical agitation and drying, the spheres are sorted using a magnet to retrieve the steel spheres (e). The reclaimed raw materials are directly poured back into the deposition modules (f) and another object is printed (g) with the materials reversed (h).

114

vibrations, graded bins, etc., although the process becomes more difficult if two materials with very similar density are used. Efficient active sorting systems also exist which are capable of sorting objects based on optical properties, and one could certainly devise sorting systems based on other material properties such as electrical conductivity, permittivity, dielectric strength, thermal effects, permeability, acoustic properties, etc.

The complete recycling of the materials used in a multi-material, additive manufacturing process has been demonstrated here. The process presented scales to arbitrarily large numbers of voxels in an object, arbitrarily small voxels, and an arbitrarily high number of materials, arranged in any spatial combination, including dense interleaving. Thus, digitally fabricated objects show promise in being environmentally friendly and generating essentially no waste in their total life cycle.

Part III

Physical and Structural Properties

CHAPTER 7

PRECISION ANALYSIS OF DIGITAL MATERIALS WITH IMPERFECT VOXELS

Because voxels in a digital material must self-align upon being assembled, the overall assembly accuracy of digital material is determined solely by the precision of the individual voxels, not by the fabrication process. Thus, it is necessary to determine how the error of the overall structure scales with respect to manufacturing errors in the individual voxels.

Because manufacturing voxels at very high precision comes with a cost, it is unrealistic to assume voxels will have near-perfect geometry. To better understand how dimensional errors propagate through the structure, extensive numerical simulation was performed assembling virtual digital structures of spheres and tiles with varying dimensional error distributions. The resulting precision of the aggregate structures was measured, and this process repeated with different errors a statistically significant number of times. For the simple case of spherical voxels, the simulation results were verified both analytically and experimentally. The simulation was then adapted to make predictions about the more complex case of the square tile voxel geometry.

There is no known previous work that attempts to quantify the error scaling laws of the precision of a voxel-based object. This is in large part due to the lack of previous demonstrations of voxel-based digital fabrication with prefabricated voxels. Although the summing of random variable is well understood, this has not been explored for interlocking digital structures with random errors in the individual voxels.

7.1 Relaxation Simulation

To determine how the accuracy of large assemblies of randomly varying spheres scales, a simulator was implemented here to measure the deviation of heights of virtual stacked pyramids of spheres. The simulation was carried out using a relaxation algorithm, and the spheres were settled under gravity (Figure 7.1). Each sphere was modeled as a point mass with as many degrees of freedom as the current simulation required (I.E. each sphere in the 3D simulation had three translational degrees of freedom). Each voxel was initialized with a diameter varying randomly within the specified tolerance according to a uniform random distribution.

The interaction between spheres was modeled as a bi-linear force vs. penetration curve. Two adjacent spheres were overlapping if

$$|P_{v2} - P_{v1}| < r_{v1} + r_{v2} \tag{7.1}$$

where P_{v2} and P_{v1} are the positions of the voxel in question and r_{v1} and r_{v2} are their respective radii. In this overlapping condition, a repelling force proportional to the overlap was applied resisting the penetration. If two spheres were adjacent in the lattice but not overlapping, a small binding force with a constant of proportionality 0.1% of the penetration case is added to restrain the spheres from flying apart if over penetration (or numerical instability) occurs.

After the total force on each voxel from all interactions was calculated, the positions were updated proportional to the magnitude and direction of this force. A sufficiently small time step was chosen to avoid numerical instabil-
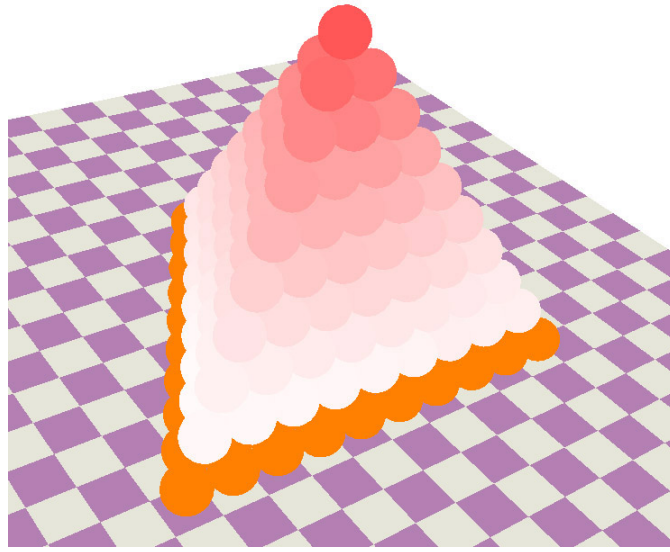
Figure 7.1: A pyramid of spheres with varying diameters is settled in simulation under gravity. White represents settled spheres, red represents spheres in motion.

ity due to the non-linear nature of collision interactions. Each simulation was considered complete when the total motion of the structure was below an arbitrarily low threshold.

7.2 Precision of Sphere Structures

7.2.1 Analytical Analysis

First, the theoretical basis was derived for how error scales using spherical voxels. Spheres are an ideal case because the interaction between voxels is simple and easily modeled. In this analysis, the random dimensional errors [34] are assumed to be small relative to the size of the spheres, such that they stay on a recognizable lattice and the angle of interaction between spheres changes in-

significantly. In the 1D case (a line of spheres), the dimensional error can be calculated analytically. The mean μ and variance σ^2 of a single uniform distribution are

$$\mu = \frac{1}{2}\left(\left(d + \frac{\varepsilon}{2}\right) + \left(d + \frac{\varepsilon}{2}\right)\right) = d \qquad (7.2)$$

and

$$\sigma^2 = \frac{1}{12}\varepsilon^2 \qquad (7.3)$$

where d is the diameter of the spheres which varies uniformly within a given tolerance range ε. Given constant nominal diameters and tolerances, summing N uniform distributions results in

$$\mu_N = Nd \qquad (7.4)$$

and

$$\sigma_N^2 = \frac{N}{12}\varepsilon^2 \qquad (7.5)$$

where μ_N and σ_N^2 are the total mean and variation. Thus, stacking a 1D line of voxels gives a dimensional standard deviation of

$$\sigma_{1D} = \varepsilon\sqrt{\frac{N - 0.5}{12}} + c \qquad (7.6)$$

where σ_{1D} is the standard deviation of length, $(N - 0.5)$ is the effective number of voxel layers which (along with a small constant correction c) accounts for

the physical constraints of the first voxel, and ε is the overall tolerance ($\pm\varepsilon/2$) of each spherical voxel. For higher dimensions, a geometric lattice calculation yields

$$\sigma_{2D} = \sqrt{\frac{3}{4}}\sigma_{1D} \tag{7.7}$$

and

$$\sigma_{3D} = \sqrt{\frac{2}{3}}\sigma_{1D} \tag{7.8}$$

In the 2D case, spheres are assumed to be on a hexagonal grid. Patterning on a square grid would result in largely decoupling the interactions between dimensions, so the 1D case would hold true in both dimensions. In the 3D case the spheres are in a hexagonal close-packed (HCP) lattice structure. Other lattice structures would result slightly varying coefficients based on the difference of interaction angle between spheres in the 3rd dimension. However, all these equations can be consolidated into a single worst-case equation when put in terms of the overall dimension L by multiplying the number of layers N by the normal distance between them. The resulting equation holds for all 3 dimensions if the number of spheres is large:

$$\sigma_L \approx \varepsilon\sqrt{\frac{L}{12d}} \tag{7.9}$$

These theoretical models suggest that the error in the dimensions of a printed object grows more slowly than in proportion to its size. Specifically, stacking a pyramid of n layers of voxels, each with a random diametrical standard devi-

ation σ_v would result in a structure with a theoretical dimensional precision ε that grows as $\varepsilon \approx \sigma_v\sqrt{n/14.7}$. This sub-linear error scaling is due to the fact that voxel imprecisions tend to cancel each other out. The unintuitive implication is that a printed structure can be more precise that the printer that printed it.

7.2.2   Simulation Results

In the 1D and 2D cases the relaxation algorithm described above was carried out in Matlab, but the 3D case was implemented in C++ for superior speed. The vertical error of the topmost voxel from its nominal position was recorded and normalized by the tolerance range of the voxels. Voxels with nominal diameters of 1mm varying uniformly randomly within a given tolerance of 0.002mm were generated and stacked in a 1D vertical line or a 2D/3D pyramid, yielding an ε/d ratio of 0.2%. This was repeated for voxels stacked to varying heights with 1000 runs per height. The standard deviation of vertical positions for each height was output and plotted in Figure 7.2.

The data for the one and two dimensional cases confirm the analytical analysis. The three-dimensional case illustrates increased error canceling above 10 voxels that was not included in the analytical analysis, and is fit more accurately with a lower exponent (cube root). This is because there are more paths for random errors to average out. This favorable property is not necessarily valid for structures that are thin relative to the size of the voxels. However, due to the limitations of layered manufacturing processes, a sacrificial material must be included in the printing process. These extra voxels (in the form of a stable pyramid) not only support the thin structure, but also impart their precision,
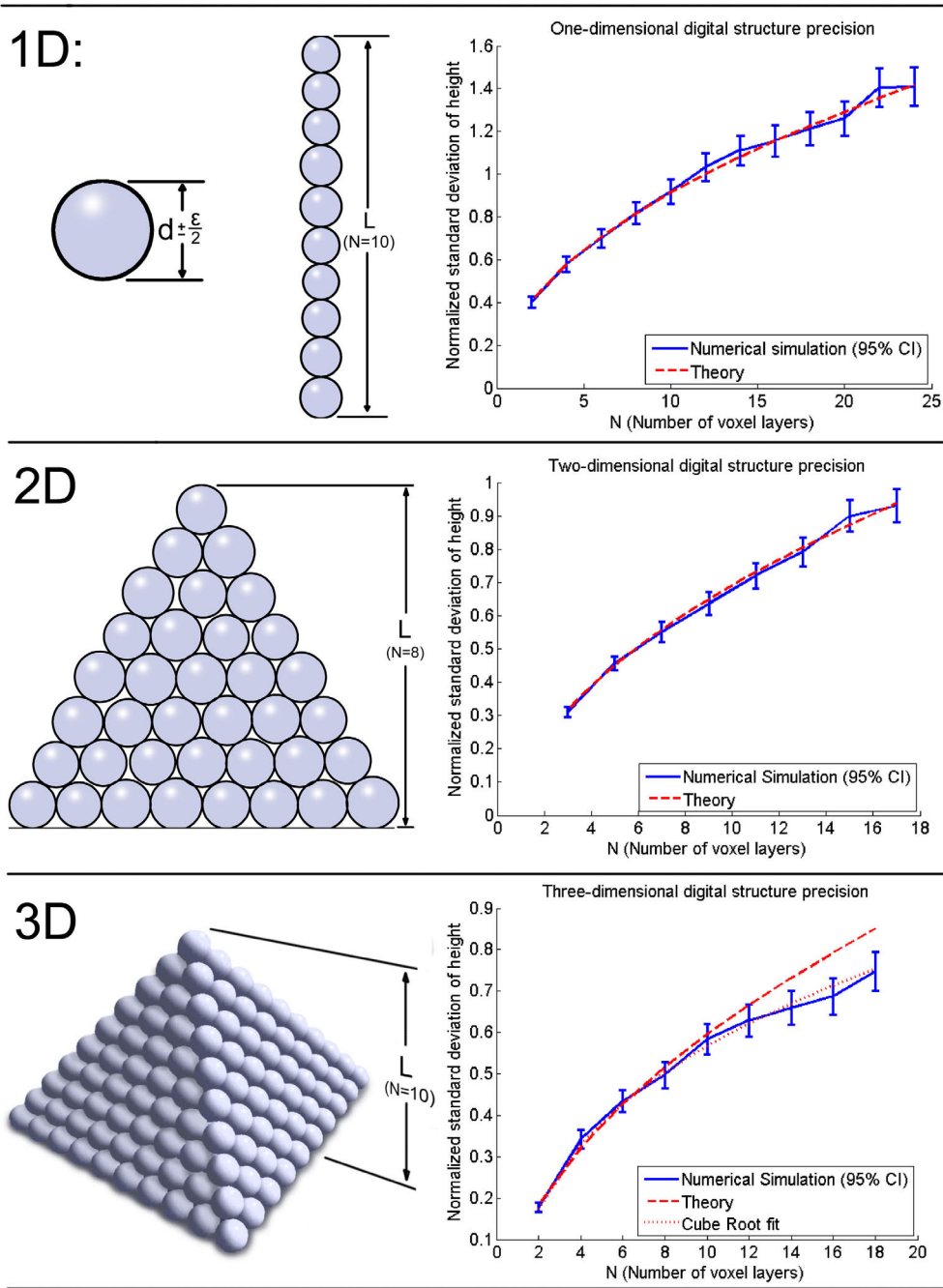
Figure 7.2: The error of multi-dimensional digital structure composed of spheres illustrates sub-linear growth. This implies random errors in the individual voxels tend to cancel out rather than amplify.

effectively eliminating this issue. The implications of this for a practical sphere printer are good. For example, using $100\mu m$ spheres with tolerance $\varepsilon = 5\mu m$ will allow fabrication of a 10 cm-scale 3D part with $100\mu m$ resolution and dimensions with a $45\mu m$ standard deviation.

7.2.3   Physical Sphere Measurements

In order to verify the predictions regarding sphere accuracy, 15 independent pyramids were stacked of acrylic voxels with heights of 4, 6, 8, and 10 layers. The height of each pyramid was measured using an indexed optical method (Figure 7.3). Pyramids were created and measured within a form with a tetrahedral void. This inverse structure was truncated at the top, such that the top sphere emerged slightly above the level of the form. This enabled many independent pyramids to be created and measured using the same set of spheres repeatedly. This method also negates any influence the binder (glue) may have on the results, which was not modeled in the simulations.

To create a pyramid, the form was turned over, and spheres were poured in and vibrated until they formed a perfect lattice. Then, an indexed bottom plate (with holes at the location of the bottom spheres) was slid over the opening and the entire assembly flipped over, so that the spheres were resting on each other and not influenced by the form. This was then carefully placed under an optical microscope and pushed against three stops for precise, repeatable kinematic positioning. By including a 45 degree first surface mirror, the microscope view was effectively horizontal, looking across the top of the top sphere. A fixed feature (relative to the microscope) was included in the frame with a reference
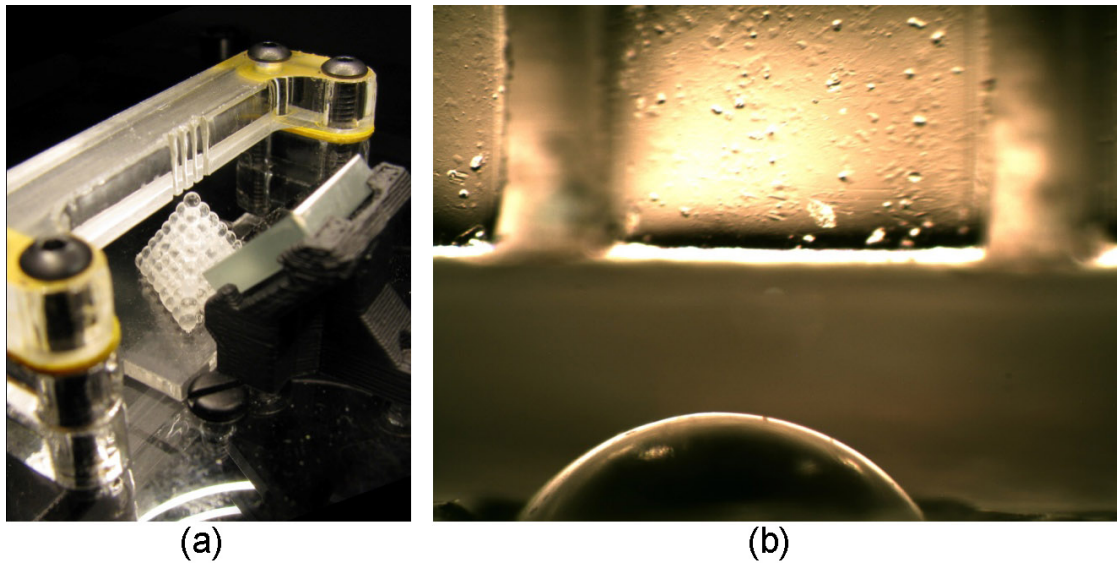
Figure 7.3: Physical pyramids of spheres were stacked and measure using a non-contact indexed optical method (a). A sample view (b) shows the indexed (fixed) feature above and the top of the pyramid of spheres below, which varied in height from trial to trial.

dimension (Figure 7.3b). A 3.2 megapixel digital camera was mounted on the microscope for an effective resolution of 1 micron per pixel. A full resolution frame was captured for each pyramid. Then, each frame was analyzed by measuring the distance from the top of the sphere to the fixed feature. This yields a relative measure of height for each pyramid, from which the standard deviation in height may be calculated for the entire set.

In order to compare the measured precision results to the computer simulations, the standard deviation of pyramid heights was normalized by the standard deviation of the diameter of the individual spheres. In order to accurately determine the dimensions of the spheres, 50 spheres were selected at random as representative of the entire set. Images of each were captured under an optical microscope with a 3.2 megapixel microscope camera for an approximate pixel resolution of $0.7\mu m$. Structured lighting was set up such that the edge
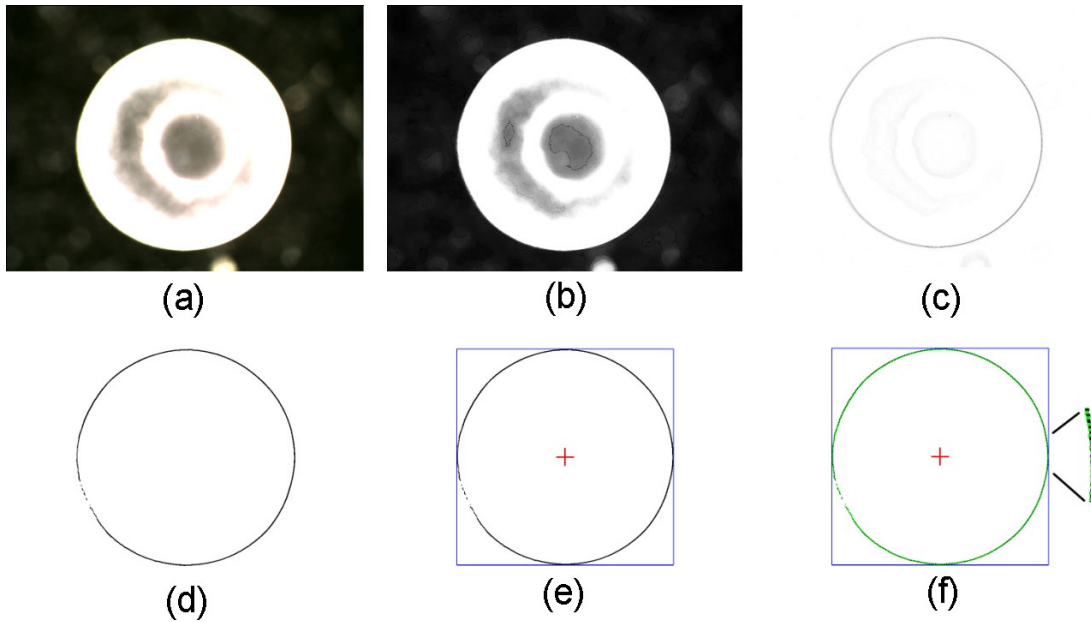
Figure 7.4: The diameter of 50 acrylic spheres was characterized using machine vision and image processing. A high-contrast photo of a sphere is taken (a) and converted to black and white (b). A local edge finding filter is applied (c) and then thresholded to highlight the perimeter of the sphere. The center point and bounding box are calculated, then iterated to reduce sensitivity to noise and other undesired parameters. Finally, the perimeter is segmented at 360 points and used for analysis.

of each sphere was well lit in contrast to a black background, to make edge identification easy (Figure 7.4). The image was converted to black and white for computational simplicity and a Gaussian blurring filter was applied with a half-mean half width radius of 1.7 pixels. Then, an edge finding filter was applied by calculating the intensity gradient in the X and Y direction and finding the euclidean norm. This process was especially robust at finding the edge of the sphere because the limited depth of field of the microscope resulted in everything else in the image being out of focus, and thus not making it through the edge finding filter.

Next the image was thresholded to create a binary image and remove any

low-intensity artifacts from the edge-finding. The center and bounding box of the resulting image was computed by sweeping in from each edge of the image until a significant (10 or more) number of black pixels were encountered in any given row or column. Then, to characterize the radius of the circle, rays were sent out from the center of the circle every degree (360 rays). For each ray, the pixels were successively considered from the center outward. Every time the series switch from white pixels to black or vice versa, the radius was recorded. Upon reaching the edge of the image, the largest continuous region of black pixels was assumed to be the edge of the sphere, and the average radius of these pixel was recorded. In order to reduce the effect of outlying points or other image noise, any points greater than three standard deviations from the mean were discarded as outliers. This process was iterated a few times (typically 4) to ensure the true edge of the sphere was being measured in all cases.

The resulting quantitative error distribution of the actual pyramids was less precise than predicted by the models (Figure 7.5). However, the qualitative sub-linear error scaling the grows with approximately the square root of the number of layers was observed. It is likely that the simulation was optimistic in its assumption of perfectly spherical voxels in which only radius varied.

7.3   Precision of Square Tile Structures

7.3.1   Extensions to the Relaxation Simulator

The existing 3D relaxation simulator was adapted for the square tile-based voxel design. Each tile was modeled as four separate points (Figure 7.6) connected
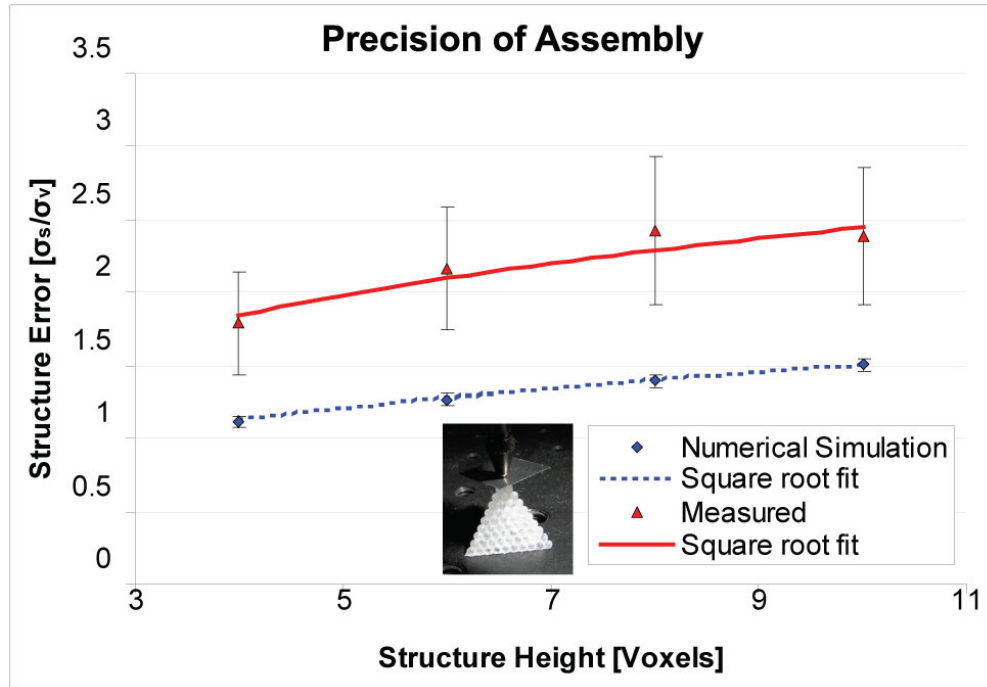
Figure 7.5: Precision of voxel-based materials scales favorably as a function of the size of structure. Model and data illustrates sublinear error scaling. The term σ_s represents the standard deviation in the structure height, and σ_v represents the standard deviation in the diameter of a single voxel. Error bars are based on 15 independent structures for experiments and 100 independent trials for numerical simulation.

rigidly. Both the dimensions of the tiles and the locations of the out-of-plane joints were varied uniformly within the tolerance range ε, representing the accuracy of the voxel manufacturing process. In the numerical simulation of the tiles, several assumptions were made. First, the tile errors were kept small relative to the size of the tiles. Thus, the ratio of the error ε to the tile size d (ε/d) was less than 1%. This allowed the small angle assumption to be made in calculating forces involved. Interactions with adjacent tiles in plane were modeled as a contact force (pushing, but no pulling), and X/Y interactions with the tiles directly above and below were modeled with a small dead zone around equilibrium to represent inaccuracies in the fit. This dead zone was sized such that based on
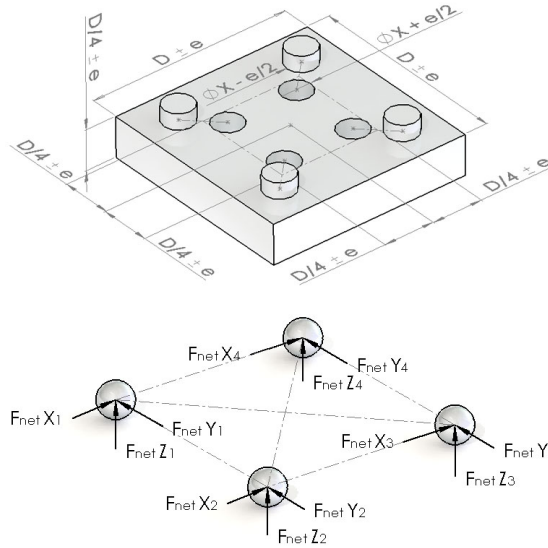
Figure 7.6: The dimensions, errors, and forces involved in large-scale tile numerical simulation are illustrated. This tile model was used to predict the precision scaling of structures composed of many such tiles.

the manufacturing tolerances of the tiles, the largest possible peg would always fit in the smallest possible hole.

An example of a relaxation of a large tile structure relaxation visualization is shown in Figure 7.7. Cubes of simulated tile voxels were generated with dimensions $N \times N \times N$, filling the space between (0,0,0) and (N-1,N-1,N-1). Each tile was modeled with uniform random errors and the entire structure was relaxed. These structures were loosely constrained along the X, Y, and Z planes, and for each trial the location of the tile at (N,N,N) was recorded, representing the overall size of the digital-material cube. For each dimension of cube, 1000 trials were completed to obtain good statistical significance. The standard deviation of these trials are plotted in Figure 7.8 with varying N. The optimized simulation ran for about 20 hours on four CPU cores simultaneously to obtain these results.
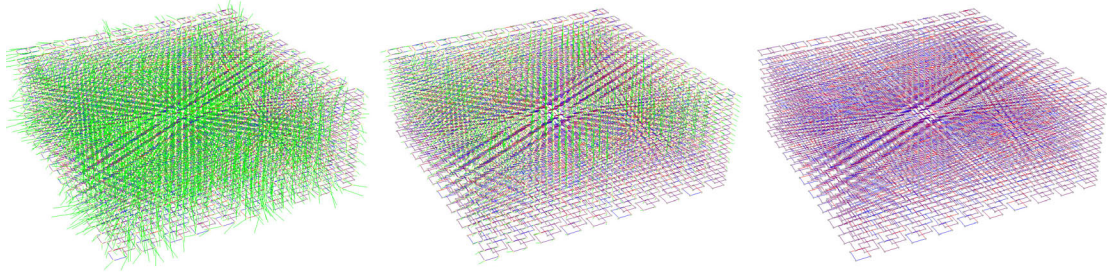
Figure 7.7: Screen shots from one relaxation iteration for a $10 \times 10 \times 20$ tile-voxel based structure. Yellow nodes with green force vectors represents unrelaxed areas, whereas red and blue links represents the residual tension and compression forces (respectively) within the tiles after relaxation.

7.3.2 Simulation Results

As observed with the spherical voxels, the error scales with a power law. In the X and Y dimensions, this exponent is much lower (about 0.1) than observed in the Z dimension and with spherical voxels (corresponding exponent of about 0.5) (Figure 7.8). Thus, if a multilayer lithography process can create 1mm tiles with a precision of $1\mu m$, a one cubic meter ($1000 \times 1000 \times 1000$ units) structure would have dimensional error of less than $2\mu m$ in the X and Y directions, and an error of about $30\mu m$ in the Z direction.

Additional simulations were carried out to determine how the overall error of the structure varies with the error of the individual tiles over three orders of magnitude of errors. Cubes of tile voxels with $10 \times 10 \times 10$ elements were generated and relaxed with unit tile size and voxel precisions ranging from 10^{-5} to 10^{-2}. For each precision level, 1000 runs were completed and the standard deviation of voxel (10,10,10) was recorded and normalized by the current voxel precision. As long as the error of the tiles is small ($< 1\%$ of the size), the relationship is a direct linear correlation with unit slope (Figure 7.9). Thus, a voxel
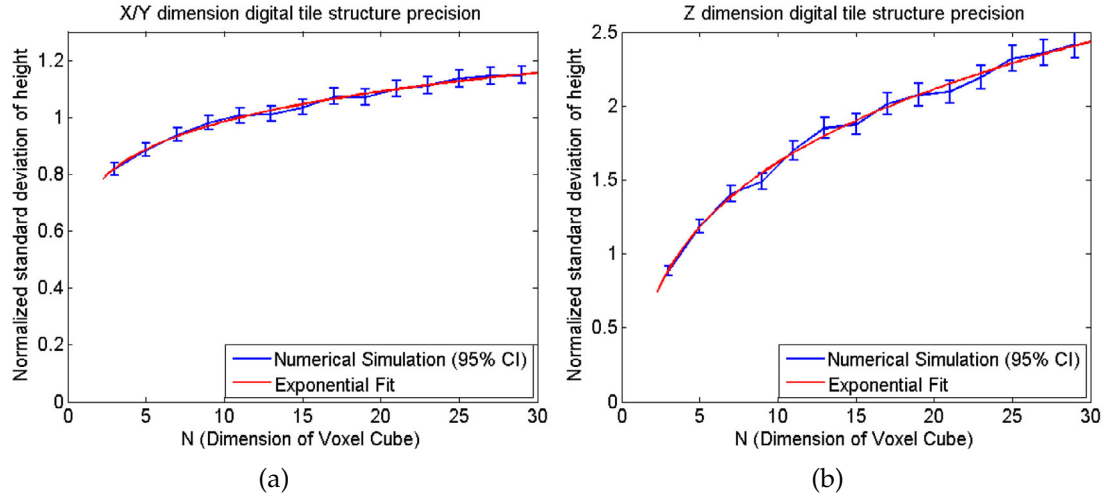
Figure 7.8: Digital structure precision for square tile structures. Total error scales with a power law in respect to the size of the structure. In the X/Y direction (a), active force cancellation leads to a very low exponent ($x^{0.1}$), whereas in the Z direction (b), similar results to spheres are observed ($x^{0.5}$)

manufacturing process that improves voxel precision by 50% will increase the precision of an aggregate digital structure by 50%.

7.4 Extension to Other Voxel Geometries

Based on the results of the sphere and square tile simulations, there are two classes of voxel interfaces that govern how error in a digital object scales: contact-only force transmission where voxels can only push against each other and bi-directional force transmission where voxels can push or pull against each other. Contact only force such as spheres and the out-of-plane direction of tiles results in precision that scales roughly with the square root of the number of layers. Bidirectional force interfaces such as the in-plane direction of the interlocking tiles result in precision that scales by a power law with the exponent being sig-
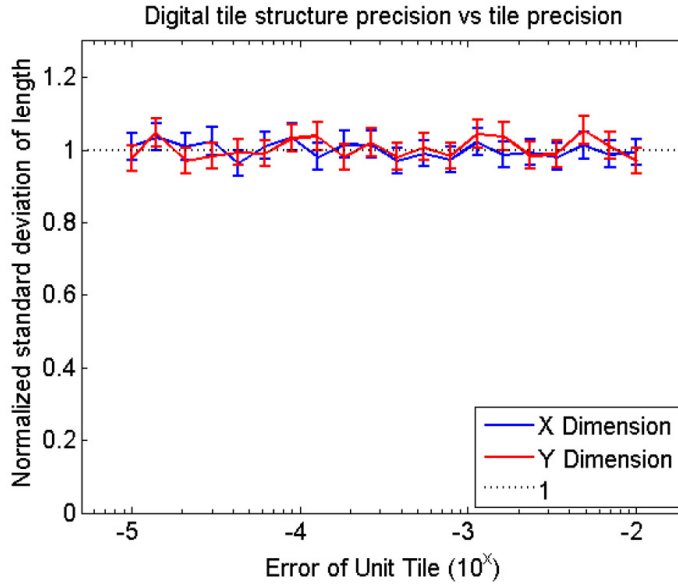
Figure 7.9: The precision of a digital structure varies directly with the precision of the tiles (ε). This implies that doubling voxel precision will double the aggregate digital object precision.

nificantly less (0.1 in this simulation). Thus, it is possible make useful predictions regarding how error scales in any of the other voxel designs based on which class of interactions they experience.

CHAPTER 8

TUNABLE HETEROGENEOUS MATERIAL PROPERTIES

8.1   Heterogeneous Digital Materials

A fundamental property of digital materials is that they can be fabricated with a specific material at each location within the defining lattice. Each material can have varying properties to create a heterogeneous composite digital material. This enables a whole new design space of heterogeneous materials with highly tunable properties. If the structures are large relative to the size of the individual voxels, the discrete effects of the voxels are minimal and the structure behaves as a pseudo-continuous composite material.

The range of material properties attainable using these methods are explored and simulated to determine how the properties of digital materials can be tuned for a wide range of applications. By varying the precision, geometry, and material of the individual voxels, continuous control over the density, elastic modulus, CTE, ductility, and failure mode of the material is obtained. Also, the effects of several hierarchical voxel "microstructures" are demonstrated, resulting in interesting properties such as auxetic (negative Poissons ratio) materials. This implies that digital materials can exhibit widely varying properties in a single desktop fabrication process.

8.2   Non-Linear Tile Lookup Simulation

In order to better understand the properties of voxel-based materials, comprehensive simulation framework was developed for the previously explored square tile geometry (Figure 8.1(a)). This allows efficient simulation of the physical properties of structures composed of thousands of voxels of multiple materials. This framework incorporates non-linear interaction between interlocking voxels, taking into the specific geometry of the interfaces and including surface contact and plastic material yielding effects. Also, this simulation supports first order modeling of random variation in tile geometry resulting from non-ideal manufacturing processes and thermal effects.

Random half toning between two materials is supported (Figure 8.1(b)), as well as user-definable voxel sub-structures (microstructure) that can be tiled throughout a region. Although all following results are based on the square tile geometry, the principles demonstrated apply to all voxel geometries that physically interlock.

A relaxation algorithm [107] was implemented in order to calculate the rest position of each voxel under random dimensional variation and/or prescribed force or displacement to the structure. The entire simulation was written in C++ for superior speed, flexibility, and the ability to use real-time OpenGL 3D visualization of the structure. Dynamic time stepping was implemented to ensure quick convergence while maintaining computational stability. Running on a single core of a desktop computer, a relaxation cycle of 1000 tiles ($10 \times 10 \times 10$ block) composed of 4500 individual interactions takes approximately 3ms per time step and scales as O(n) where n represents the number of tiles in the sim-

<div align="center">(a)   (b)</div>
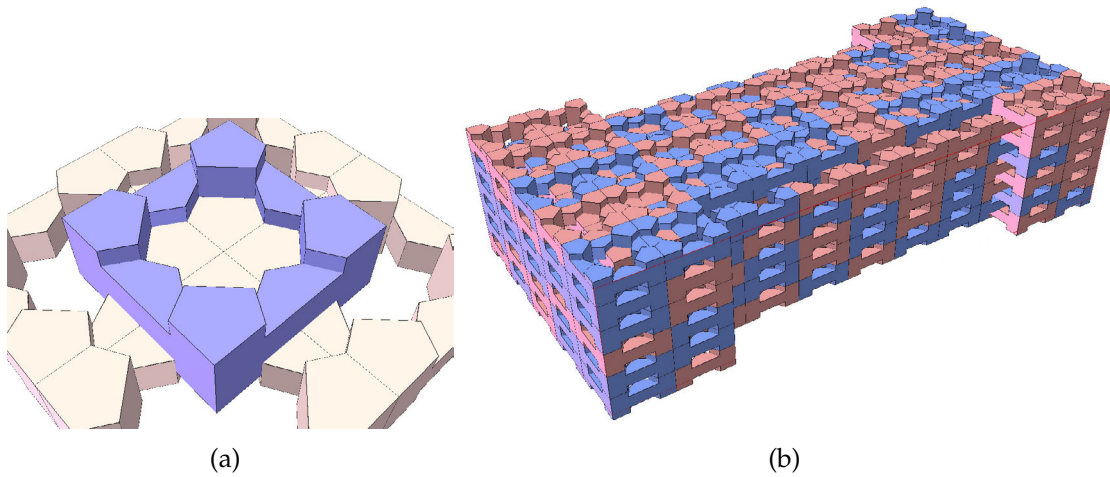
Figure 8.1: 2.5D interlocking square voxels (a) are considered for this analysis due to their ease of fabrication at many scales/materials. Virtual tensile tests were carried out on large blocks of voxels of multiple materials (b)

ulation. Convergence is arbitrarily defined to be when average tile movement is less than 10 picometers for a given time step. The aforementioned example takes about 1100 time steps to converge, or 4 seconds on a single CPU. This allows us to feasibly compute a statistically significant number of trials for the work presented here.

In order to accurately model the force between each voxel and its neighbors, a finite element model was developed in COMSOL Multiphysics for each type of tile interaction (Figure 8.2). In the case of square tiles, there are only two modes in which adjacent tiles interact: an edge-to-edge interaction between tiles in the same layer and an corner-to-corner interaction between tiles in adjacent layers. A separate COMSOL simulation was developed for each taking advantage of symmetry to reduce the requisite mesh size. For the edge contacts, only half of each tile was modeled and for the corner interactions, only one quarter of each tile was needed.

The materials model in the COMSOL simulation was set to be elasto-plastic, and the interaction between the tiles was modeled as a non-linear contact force. Several assumptions were made: First, individual tile interactions were considered independently within a given tile. This does not pose a problem as long as the material has not yielded, but past that point it provides only an approximation of the tile failure. Failure was take to occur as soon as the maximum stress in the tile reached the ultimate tensile stress of the material. This corresponds to the worst case condition of brittle fracture, and in reality higher deformations are likely attainable. Also, displacements and rotations out of plane were disregarded, since tile displacements of the tiles were kept small.

With both tiles of a contact pair modeled in COMSOL, one tile was grounded (about its symmetry plane(s) and a prescribed displacement was applied to the other. The model was run using a non-linear solver and the resulting reaction forces and maximum stress was recorded. This was repeated for varying X, Y, and θ_z displacements to fill out a parameter-space. This data was recorded in a lookup table for importing into the main simulation program. Because the time to retrieve a value in an indexed lookup table does not change with the size of the table, the parameter-space can be filled out to arbitrary accuracy without affecting the simulation time of the relaxation simulator.

Because of the design of the tiles, there is inherent anisotropy in the Z direction compared to the X and Y. In this direction there is no physical interlocking, just friction that binds the tiles together. Thus, the physical behavior in the Z direction is less interesting, and not addressed in the following results.
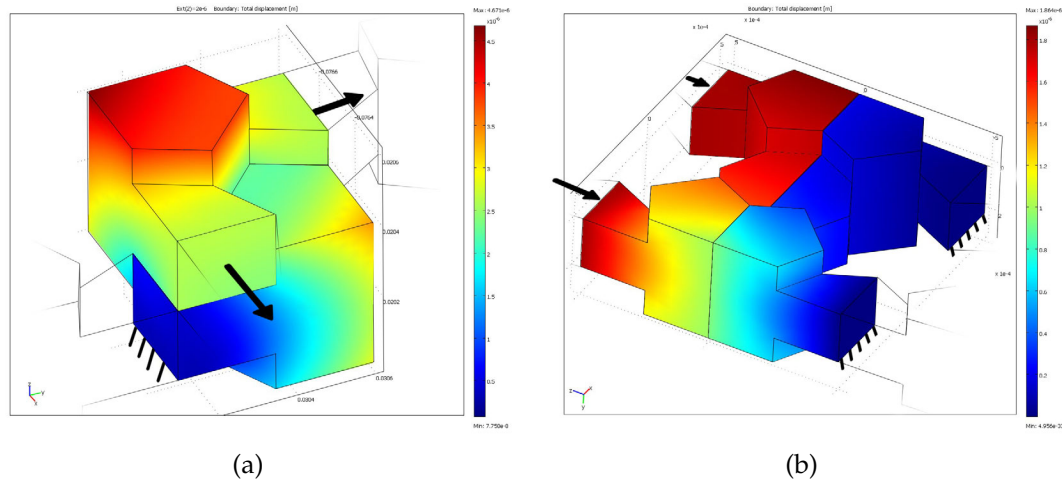
Figure 8.2: Output of COMSOL finite element simulations for the two types of interactions between tiles: corner to corner (a) and edge to edge (b). Non-linear interactions were characterized for a range of displacements and imported into a large-scale simulation with thousands of tiles.

8.3 Finite Element Models of Tile Interaction

The results of single tile-to-tile interactions from the finite element modeling are shown in Figure 8.3 for edge-to-edge and Figure 8.4 for corner-to-corner interactions of aluminum tiles. The maximum stress plots clearly show the non-linear material model yielding at 276MPa as the tiles are forced into greater interference. The limits of the parameter-space were determined by the displacement necessary to cause the material to reach its ultimate stress.

8.4 Virtual Tensile Tests

Virtual tensile tests were conducted on structures of approximately 500 tiles, in the configuration shown in Figure 8.1. As is normal practice in physical ten-
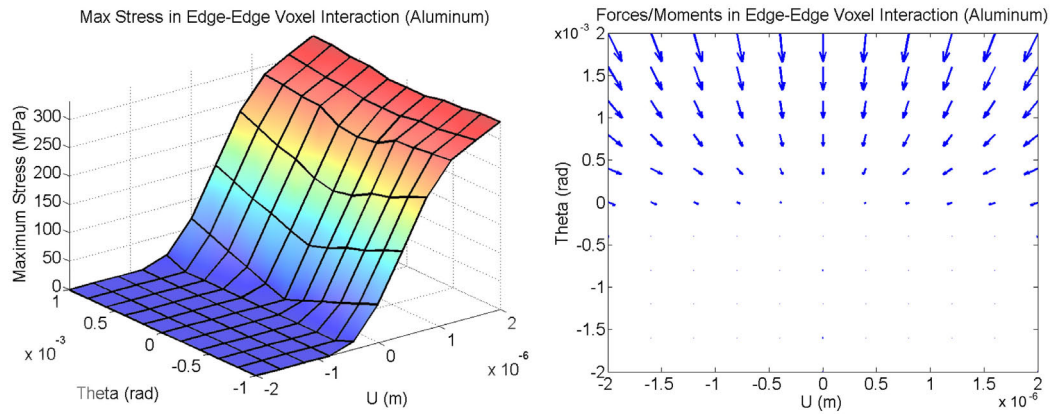
Figure 8.3: Results of finite element simulations. The stress (left) and resultant forces (right) in edge-to edge tile interactions for a range of tile positions and rotations is shown for aluminum tiles. This information is stored in a lookup table for quick access.
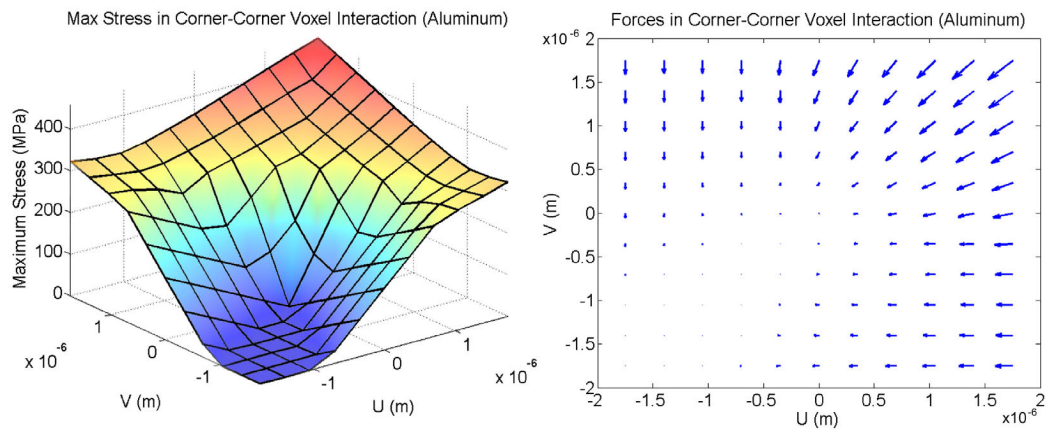
Figure 8.4: Stress (left) and resultant force (right) for corner-to-corner tile interactions. Each data point represents a separate finite element model solution.

sile tests, the specimen included a slight reduction in area in the center to avoid breakage from at the prescribed boundary conditions at the ends. The tiles at the extreme ends of the structure were fixed via stiff springs to ground. Multiple relaxation iterations were then carried out, incrementing the elongation slightly each time. The structure was fully relaxed between each iteration, corresponding to a "slow" tensile test in which dynamic effects are minimal. In the case of one or more joints failing, the relaxation was repeated with the broken joint(s) rendered useless until no more joints broke, in order to allow failure to propagate realistically in brittle samples.

8.4.1 Single Material

As designed, the tiles presented here are space-filling. It is therefore reasonable to compare the material properties of a digital material specimen of a single material to an equivalent sample of bulk (analog) material, which would have equivalent net densities. In the case of aluminum 6061, a bulk elastic modulus of 70GPa was used. In the virtual tensile test, a structure of solid aluminum tiles had an aggregate elastic modulus of 4.75GPa. This corresponds to about a 15x reduction in stiffness. Other materials, such as acrylic, showed similar reductions (10-15x) in overall stiffness.

Even using a single material, the bulk material properties and failure mode of a digital structure can be tuned over a wide range by simply varying the precision of the manufacturing process used to manufacture the tiles. In simulation, this was modeled as a uniformly random variation in the dimension of the tiles over a given range. As the errors of the tile geometry increased, the elas-

tic modulus continuously decreased and the brittle failure mode became more ductile (Figure 8.5). Also, strain to failure increased dramatically, although maximum stress decreased. Additionally, as the voxel error increased, an "uptake" region was introduced in the tensile test, where the slack was being taken out of the structure. This feature can be controlled by the offsets of the tile dimension from nominal, for instance by making all the outside dimensions of the tiles slightly larger.

This brittle-to-ductile behavior is explained by observing the number of failed joints in the structure. When there is greater variation in the voxel dimensions, individual tiles begin to fail before the structure has reached a critical stress. The rest of the structure can absorb the additional force released by the broken voxel, and the number of failing voxels (and thus the failure point of the material) can be predicted. The initial broken joints in the ductile ($10\mu m$) case were due to high errors that caused failure upon the original assembly of the structure.

8.4.2   Material Half-toning

The properties of digital materials may be tuned even more extensively by introducing hybrid systems of two or more materials. Because a digital fabrication process is inherently compatible with any number of materials in a single build process, these materials may be easily combined to create hybrid composite materials. The simplest way to do this is to randomly halftone a percentage of two materials to obtain the desired proportions. Much research has gone into half-toning for creating black and white images from gray scale, but for a pseudo-homogeneous material, the problem becomes easier. For these simula-
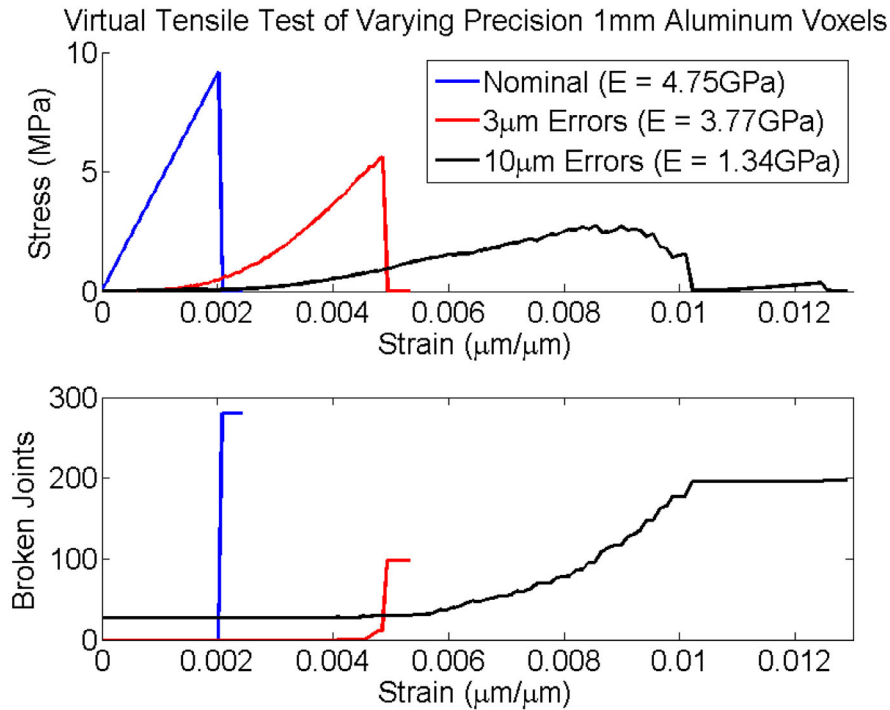
Figure 8.5: Tensile tests (top) on single material voxel structures with vary-
ing precision of the voxel geometries demonstrates a wide
range of elastic moduli (E) and failure modes. The number of
broken connections (bottom) shows the difference between a
brittle and ductile fracture.

tions, a desired percentage of material was selected. Given the number of voxel
in the region to be half-toned, the total number of voxels of each material was
calculated and voxels were randomly selected for each material such that the
totals added up to the correct desired proportion.

By using this process, bulk material properties of a digital structure may be
continuously varied between the properties of each single material. Some prop-
erties, such as density, can be calculated analytically based on the percentage
and density of the two component materials. For materials of density ρ_A and
ρ_B, with a percentage α of material A, the total density ρ_{Total} can be calculated
as:

$$\rho_{Total} = \alpha\rho_A + (1-\alpha)\rho_B \tag{8.1}$$

This can be used for design purposes to selected the desired density of a composite material. Other material properties, such as elastic modulus, also continuously vary over the range of percentages, but non-linear effects and interactions between tiles make an analytical solution impractical for random half toning. The results of simulated tensile tests of the full range of percentages varying between acrylic and aluminum are shown in Figure 8.6. The elastic modulus varies exponentially between the two extremes of homogeneous material. Since the half toning is carried out randomly, 50 tensile tests were run for each material percentage, giving an approximation of the variation in stiffness to be expected.

In addition to tensile tests, thermal expansion tests were carried out to demonstrate the tunable coefficient of thermal expansion (CTE) of digital materials. In the same manner as density and elastic modulus, the CTE of a material may be tuned anywhere between the two aggregate materials, depending on their relative volume fractions. This is particularly interesting when one of the materials has a negative CTE, implying that it contracts upon heating. Although not common, these materials exists both in nature and in select engineering materials. One of the most common of the latter is zirconium tungstate ($Zr(WO_4)_2$) (Sleight, 1998), a ceramic that exhibits this property over a wide range of temperatures. The dimension of hybrid structures of zirconium tungstate and aluminum over a range of temperatures is shown in Figure 8.7. In this case, a combination of approximately 17% aluminum and 85% zirconium tungstate yields virtually no thermal expansion or contraction at the temperature varies.
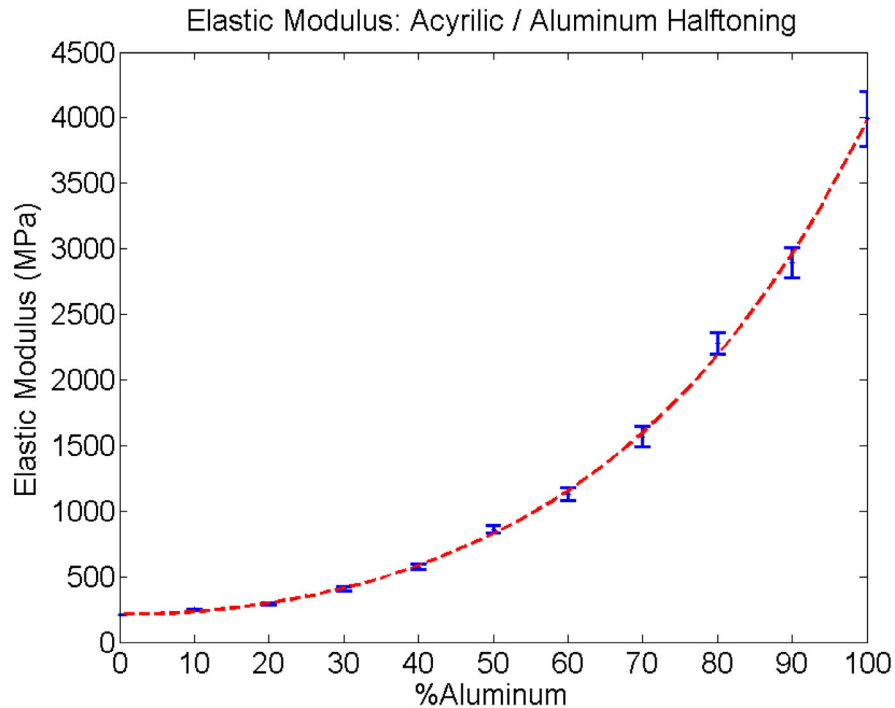
Figure 8.6: The elastic modulus of two-material digital structures varies continuously between the each constituent material property as the ratio of half-toning is varied. Fully acrylic structures are on the left and fully aluminum structures are on the right. The intermediate relationship is best fit with an exponential curve, shown as a dotted line.

8.4.3 Microstructure

The use of a single material or random half-tone blending yields approximately isotropic material properties in the horizontal directions of digital material composed of the square tile design. In some cases, it is desirable to introduce more anisotropy to optimize material properties in specific directions. To address this, the idea of microstructure on a voxel level is introduced. By defining a voxel sub-structure that encompasses certain properties, this "super voxel" can be tiled throughout a structure creating a hierarchical material that reflects the properties of the sub-structure. One of the most obvious applications of this is
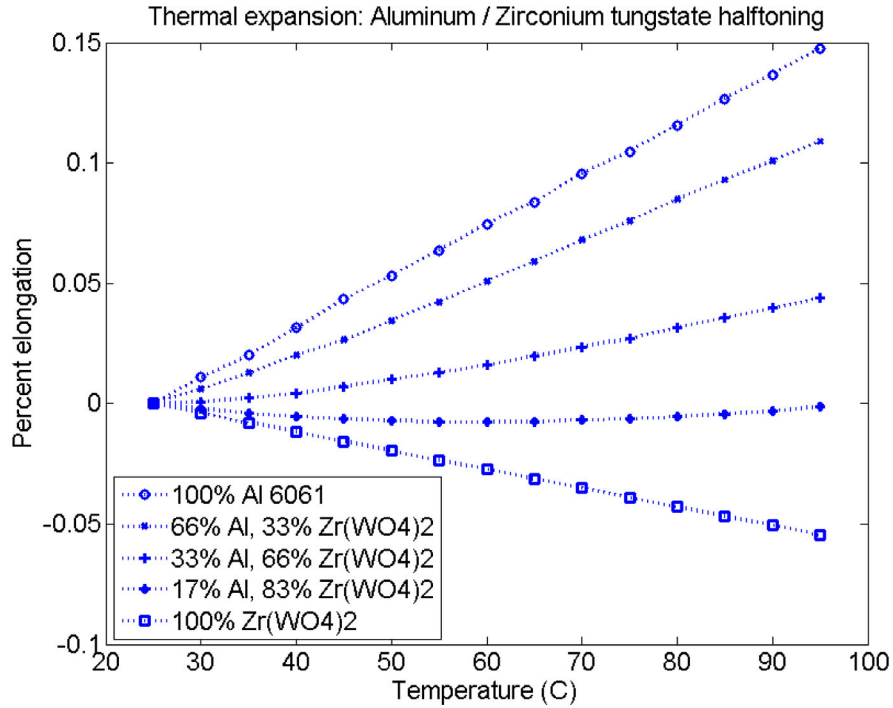
Figure 8.7: The thermal expansion/contraction of a digital structure made of different ratios of aluminum and zirconium tungstate demonstrates the ability to choose the desired CTE based on the percentage of each material.

to tune the stiffness of a hybrid material in specific direction. This can be accomplished, for instance, by aligning the stiffer voxels in lines along the direction of elongation to stiffen the material.

The stiffness of a tensile specimen composed of various $2 \times 2 \times 2$ microstructures is compared to the random half-toning case in Figure 8.8. In all cases, 50% of the voxels are aluminum and 50% are acrylic. In these tests, the tensile specimen consisted of greater than 2000 voxels, in order to ensure observation of bulk material behavior and avoid direct effects of the microstructure. Of the microstructures presented here, random, mesh, layers, and dither are pseudo-isotropic in X and Y, whereas longitudinal and transverse are at the extreme of anisotropy.

144
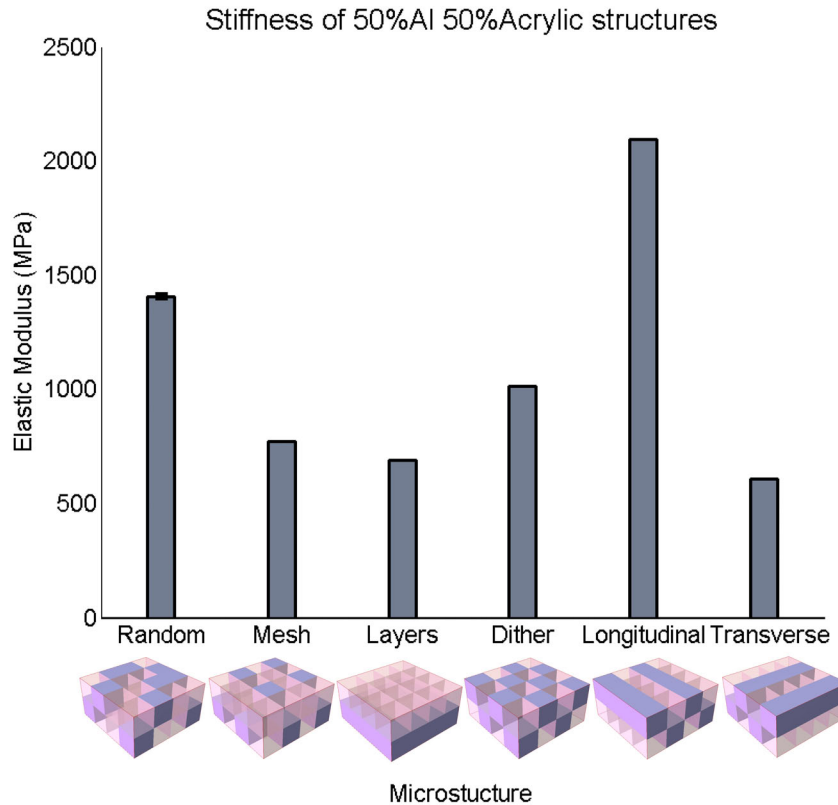
Figure 8.8: The properties of multi-material digital matter depend heavily on the voxel microstructure. Here, stiffness varies by over a factor of three for the same volume fraction of aluminum and acrylic voxels. Although visualized as cubes for clarity, the simulation used the interlocking square tile geometry.

From these tests, it is apparent that the properties of composite digital materials depend heavily on the microstructure. Even considering only the pseudo-isotropic microstructures, the stiffness of the material varies by nearly a factor of two. The difference in stiffness in the extreme anisotropy case is greater than a factor of 3 in the X/Y directions. This provides yet another aspect of control over the physical properties of digital materials. For instance, this could be used to tune the stiffness of a material while keeping the density constant.

The idea of voxel microstructure may be extended to yield even more inter-

esting properties of digital materials. For instance, auxetic materials (negative Poissons ratio) can be created. This special class of materials expands laterally when extended axially, and is useful in a wide variety of applications due to its high energy absorption and fracture resistance [48]. Most auxetic materials that have been created are low density and based on foam [98] or honeycomb structures, including polymer and metallic foams [55]. Using digital material, in simulation a dense auxetic material is demonstrated here made of aluminum and acrylic voxels, exhibiting a specific microstructure [47] of aluminum within acrylic. In theory, any combination of stiff and flexible materials would work. Using a $6 \times 8 \times 2$ voxel base unit consisting of 68% aluminum, 48% acrylic, and only 8% voids (although the voids could be reduced further), a Poissons ratio of -0.63 was attained (Figure 8.9). A similar structure with the same mass-fraction of materials and voids randomly distributed had a more mundane Poissons ration of 0.11. Because the material is highly non-linear, instantaneous poisons ratio was used and plotted as a function of strain [167].

The results clearly show the effect of the auxetic voxel microstructure. The Poissons ratio of the random pattern of voxels drops off only slightly from 0.11 until failure. However, the Poissons ratio of the auxetic microstructure begins highly negative and increases rapidly as the strain increases. This highlights the fact that the material does not expand linearly. However, this behavior is consistent with similar plots of previously created auxetic materials [98].

146
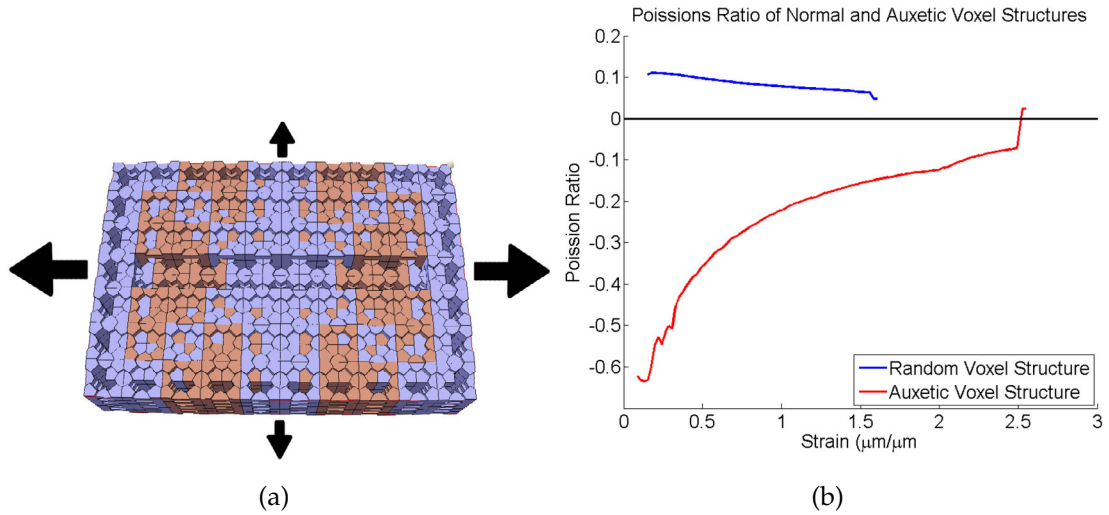
Figure 8.9: Digital material with a negative Poissons ratio can be created by tiling the structure shown at left. The resulting Poissons ratio as a function of strain is shown at right for material using both this structure and material with the same volume fractions of material ordered randomly.

8.5   Implications for Tunable Materials

These simulations demonstrate the basis of tunable materials properties of digital materials. A wide range of density, stiffness, CTE, and failure modes may be obtained by varying voxel manufacturing precision, the percentage of randomly distributed materials, and the voxel microstructure. By varying only the precision of voxels, the stiffness and failure mode of a single material structure can be controlled. By introducing a second material and randomly half-toning a percentage of voxels between them, the material properties may be tuned anywhere between the respective properties of the two materials. Additionally, the inclusion of hierarchical voxel substructures (microstructure) allows further tuning of properties such as stiffness, and allows novel behavior such as negative Poissons ratio to be obtained with dense combinations of common materials.

147

Although the material properties of a single material lattices [170] have been explored, the possibilities enabled by multiple materials opens a whole new area of research. Co-continuous composite materials [191] have been explored, and the stiffness, strength, and energy dissipation explored, but only for random distributions of materials. Although materials negative Poisson's ratio materials have been demonstrated [98, 55], they have previously been limited to sparse foam-like structures, in contrast to the nearly dense structures demonstrated here with digital materials.

Based on these results, one can expect expect digital materials to provide unprecedented control over all aspects of a bulk material. Although only mechanical stresses displacements were output for these simulation results, the multiphysics nature of the COMSOL finite element model allows for the inclusion of friction, heat transfer, fluid flow, etc. between voxels in the finite element model for future verification. In addition, the use of 3 or more materials will facilitate the tuning of multiple material properties simultaneously. The end result is the possibility of a desktop manufacturing system that can produce continually varying material properties ranging over several orders of magnitude.

CHAPTER 9

SIMULATORS FOR RAPID PHYSICAL PREDICTION

9.1   Overview of Physical Simulators

Because of the exponentially increasing design space enabled by multi-material freeform manufacturing methods including digital manufacturing, design automation will play an increasing role in the design and optimization of structures that fully take advantage of these capabilities. Most design automation algorithms such as homogenization and evolutionary algorithms depend on many, many physical simulations. Therefore a balance must be struck between calculating accurate results while minimizing CPU cycles.

Two different simulation frameworks were implemented to efficiently simulate the mechanical behavior of multi-material voxel structures. Each is useful for a subset of design problems. The first uses the linear direct stiffness method to solve for the static displacement of a structure subject to applied forces and mechanically grounded (fixed) regions. This method is limited to problems involving small displacements with linear elastic materials only. Because the entire system is formulated and solved as a system of linear equations, any non-linearities must be linearized before the solution can be computed. However, the advantage is that the linear direct stiffness method is very computationally efficient if solved with an appropriately optimized solver.

The second simulation framework performs time-domain physical simulation of voxel objects. Each voxel has a location and mass, and the positions are iteratively updated according to the current forces and internal stresses of the

object. This has the advantage of being able to simulate the time response of objects including the dynamics. Additionally, because forces are recalculated at each step, non-linear materials, friction, large deformation, etc. can be accurately simulated. However these additional capabilities come at the price of computational expense, although care was take to keep the simulation computationally efficient.

In these simulation frameworks, both static and dynamic properties are quantitatively very close to the analytical solutions for simple textbook scenarios. Features such as collision detection are in place for the dynamic soft body simulation to avoid the great inaccuracy of self intersection, but are not meant to draw scientific conclusions about the interaction between two soft bodies. By carefully budgeting CPU cycles, each simulator can accurately model physical properties while not wasting undue time on negligible effects.

9.1.1 Finite Element vs. Mass-Spring Methods

Finite element analysis (FEA) is a well established method of simulating the mechanical behavior of objects. Advantages include the ability to solve a system with irregularly spaced discretized mesh elements. A stiffness matrix is composed containing information about the connectivity of the entire mesh and the local material properties at each node. This system can only be efficiently solved if the underlying equations are linear. Thus, deformations that change the geometry significantly require periodic re-meshing [200]. Other non-linearities such as friction and advanced material models require additional levels of iteration to solve.

Mass-spring methods are widely used for deformable bodies, especially in dynamic simulations for computer graphics [130]. Advantages include relative simplicity and handling large deformations and other non-linearities with ease. An object is decomposed into discrete point masses connected by springs. Thus, the entire system forms a system of ordinary differential equations (ODEs) that can be integrated directly to solve for the behavior of the system. This makes these particle-based physical simulations very computationally efficient at the expense of accuracy.

9.1.2   Freeform Mesh vs. Voxels on a Lattice

There are a number of trade-offs associated with choosing either a freeform mesh or a lattice of voxels to dynamically simulate a heterogeneous object. Both FEA methods and many existing gaming physics simulators use a freeform mesh to discretize a 3D object for simulation. By allowing the vertices to lie at any position within the object, there is greater control over the local detail of the simulation. Specifically, this allows objects to be meshed based on the desired accuracy in a given region or dynamically re-meshed based on the current regions of interest in a deformed shape [131]. Care must be taken when forming the mesh such that the aspect ratio of each element does not vary significantly in order to preserve accuracy. However the advantages of freeform meshes quickly diminish as materials of different stiffnesses and properties are interspersed within the object. This constrains the mesh generation process and can potentially create very large and inefficient meshes, such as the case of a dithering between two materials.
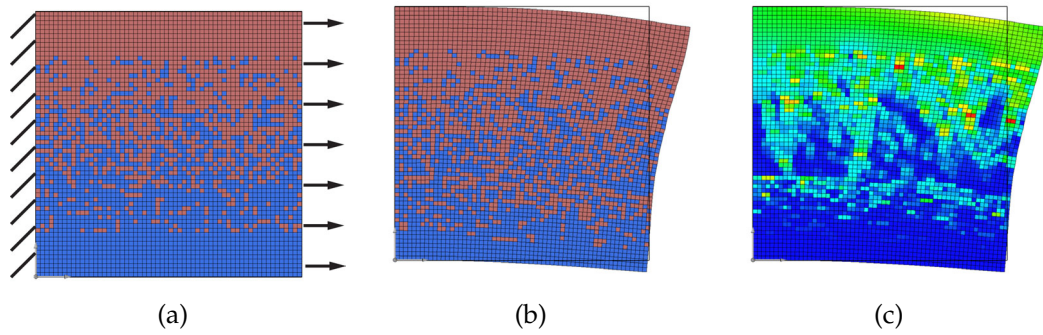
Figure 9.1: Advantages of a lattice-based voxel simulation include the native ability to simulate objects with multiple interspersed materials of varying properties (a). Here, the blue material is 100 times stiffer than the red, leading to higher deformation along the top (b). Internal deformation (strain) is also shown (c).

Limiting the discretized elements in a simulation to voxels has a number of favorable advantages. This approach enables efficient computation of the force of each constituent element, since they begin on a principle axis with identical lengths. Additionally, the stiffness of each linking beam can be pre-computed based on the stiffnesses of each constituent voxel so that each individual voxel can have a unique stiffness without altering the efficiency of the simulator (Figure 9.1). This allows heterogeneous materials to be simulated with the same computational complexity as homogeneous materials. Additionally, using a voxel lattice eliminates the possibility of ill-formed meshes [163]. However voxel lattices are at a disadvantage to freeform meshes when large regions of homogeneous material are present or very fine local details must be simulated. Because the goal here is to simulate voxel-based digital structures, the choice of a voxel lattice mesh was natural.
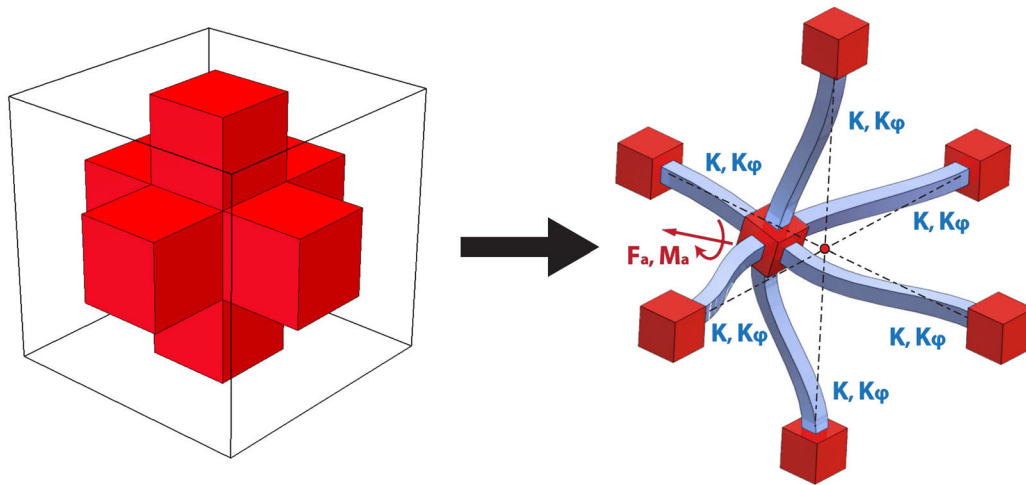
Figure 9.2: Each voxel is modeled as a lattice point with mass and rotational inertia as appropriate (red). Voxels are connected by beam elements (blue) with appropriate translational and rotational stiffnesses leading to realistic deformation under applied forces and moments.

9.2 Linear Direct Stiffness Method for Voxels

The linear direct stiffness method has been well studied for many decades, and thus the underlying methods are not covered in detail here. When implemented for voxels, each voxel was modeled with all six translational and rotational degrees of freedom and has an associated stiffness and Poissons ratio. In order to capture information about both translation and rotation of the voxels relative to each other, a constant cross-section beam element was used to connect adjacent voxels in the lattice. Beam elements resist both translation and rotation by exerting biaxial bending, transverse shear, and axial stretching forces in response to appropriate displacements (Figure 9.2). Here the standard linear Bernoulli-Euler beam theory is utilized.

9.2.1   Simulation Beam Elements

Since the solid objects are represented by a network of beams connecting nodes, the physical parameters for these beams must be calculated. Because the geometry is constrained to voxels, the length l of the beam was taken to be the distance between the voxels, and the cross-sectional area of the beam A was l^2. The standard formula was used to calculate the bending moment of inertia (I), given that in this case both the base b and height h equal the lattice dimension l.

$$I = \frac{bh^3}{12} = \frac{l^4}{12} \tag{9.1}$$

The torsion constant (J) was approximated by the polar moment of inertia of a rectangular cross-section beam, and calculated as

$$J = \frac{bh(bb + hh)}{12} = \frac{l^4}{6} \tag{9.2}$$

Using the standard Hermitian cubic shape functions for beam elements the stiffness matrix was determined for a beam element with 12 degrees of freedom: Three translational and three rotational degrees of freedom for each endpoint of the beam. This can be assembled into a stiffness matrix. The result for a beam element oriented in the positive X direction is as follows:

$$
\begin{vmatrix}
F_{x_1} \\
F_{y_1} \\
F_{z_1} \\
M_{\theta x_1} \\
M_{\theta y_1} \\
M_{\theta z_1} \\
F_{x_2} \\
F_{y_2} \\
F_{z_2} \\
M_{\theta x_2} \\
M_{\theta y_2} \\
M_{\theta z_2}
\end{vmatrix}
= [K]
\begin{vmatrix}
X_1 \\
Y_1 \\
Z_1 \\
\theta_{x_1} \\
\theta_{y_1} \\
\theta_{y_1} \\
X_2 \\
Y_2 \\
Z_2 \\
\theta_{x_2} \\
\theta_{y_2} \\
\theta_{z_2}
\end{vmatrix}
\tag{9.3}
$$

where the element stiffness matrix $[K]$ is

$$
[K] =
\begin{vmatrix}
a_1 & 0 & 0 & 0 & 0 & 0 & -a_1 & 0 & 0 & 0 & 0 & 0 \\
 & b_1 & 0 & 0 & 0 & b_2 & 0 & -b_1 & 0 & 0 & 0 & b_2 \\
 & & b_1 & 0 & -b_2 & 0 & 0 & 0 & -b_1 & 0 & -b_2 & 0 \\
 & & & a_2 & 0 & 0 & 0 & 0 & 0 & -a_2 & 0 & 0 \\
 & & & & 2b_3 & 0 & 0 & 0 & b_2 & 0 & b_3 & 0 \\
 & & & & & 2b_3 & 0 & -b_2 & 0 & 0 & 0 & b_3 \\
 & & & & & & a_1 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & b_1 & 0 & 0 & 0 & -b_2 \\
 & & & & & & & & b_1 & 0 & b_2 & 0 \\
 & & & & & & & & & a_2 & 0 & 0 \\
 & (sym) & & & & & & & & & 2b_3 & 0 \\
 & & & & & & & & & & & 2b_3
\end{vmatrix}
\tag{9.4}
$$

where

$$
a_1 = \frac{E_c A}{l} \tag{9.5}
$$

$$
a_2 = \frac{G_c J}{l} \tag{9.6}
$$

$$
b_1 = \frac{12 E_c I}{l^3} \tag{9.7}
$$

$$
b_2 = \frac{6 E_c I}{l^2} \tag{9.8}
$$

$$
b_3 = \frac{2 E_c I}{l} \tag{9.9}
$$

In order to solve the entire system, each element stiffness matrix was added to a global $6n \times 6n$ stiffness matrix $[K]$, where n is the total number of voxels in the simulation. This stiffness matrix relates the relative displacements D_n and

155

angles θ_n of each voxel to the applied forces F_n and moments M_n. This forms a matrix equation

$$\begin{bmatrix} F_n \\ M_n \end{bmatrix} = \begin{bmatrix} K \end{bmatrix} \begin{bmatrix} D_n \\ \theta_n \end{bmatrix} \tag{9.10}$$

To solve for the resulting displacements and angles, this system of equations must be solved. Fortunately, the stiffness matrix is extremely sparse, which allows for structures with tens of thousands of degrees of freedom to be solved in less than a second on an ordinary desktop computer. This sparse stiffness matrix problem was solved using the highly optimized PARDISO solver [158] to yield the resulting displacements of each voxel of the structure under load.

9.2.2   Compositing Adjacent Dissimilar Materials

In calculating the stiffness matrix, the Bernoulli-Euler beam theory requires the material to be elastic and isotropic. When two adjacent voxels are composed of the same material, the elastic modulus and stiffness of this material are used in the equivalent beam connection. However, when the materials have differing properties, an appropriate composite property must be calculated. To this end the composite stiffness of a bond between two dissimilar materials is approximated by:

$$E_c = \frac{2E_1 E_2}{E_1 + E_2} \tag{9.11}$$

where E_c is the composite elastic modulus and E_1 and E_2 are the two con-

stituent stiffnesses. Since the elastic modulus directly corresponds to the spring constant of each bond, this is analogous to combining two springs of half the length in series with dissimilar stiffnesses. The composite shear modulus are calculated in a similar manner:

$$G_c = \frac{2G_1G_2}{G_1 + G_2} \tag{9.12}$$

where G_c is the composite shear modulus and G_1 and G_2 are the two constituent shear moduli. Because Poisson's ratio relates the elastic modulus to shear modulus, it follows that the composite Poisson's ratio μ_c is calculated by:

$$\mu_c = \frac{E_c}{2G_c} - 1 \tag{9.13}$$

This method allows composite heterogeneous digital materials to be simulated using the standard direct stiffness method. Dithered and gradient materials with voxels of different mechanical properties can be accommodated in a computationally efficient manner.

9.3   Voxelyze: A Dynamic Soft Body Voxel Simulator

In some cases, non-linearities in the simulation are necessary to accurately predict relevant behavior. For instance, non-linear material models, large deformations, or time-domains behavior involving collisions and friction all prohibit the sole use of a linear simulation. In order to efficiently evaluate non-linear composite materials and amorphous soft robot morphologies with volumetric

actuation, a soft body simulator was developed ab initio in C++. The main features of the Voxelyze simulation are:

1. Speed: With thousands of time steps per evaluation, and thousands of evaluations per evolutionary run, the feasibility of autonomous design of amorphous robots and functional materials depends on having an efficient simulation.

2. Dynamics: Full dynamics modeling with variable damping allows for realistic, 2nd order momentum effects in all translational and rotational degrees of freedom.

3. Large deformation: Shapes can be bent and twisted far past any linear small angle approximations without re-voxelizing.

4. Multi-material: Any number of materials can be combined in any internal material distribution, each with varying stiffnesses and densities.

5. Friction: Nonlinear friction is incorporated with a static/dynamic friction model.

6. Collision detection and handling: Self intersection is calculated and enforced. With large deformation comes the need to avoid an object penetrating itself.

9.3.1   Background on Soft Body Simulators

There are many established methods and implementations for simulating the dynamics of deformable soft bodies [130]. Considering the large deformations
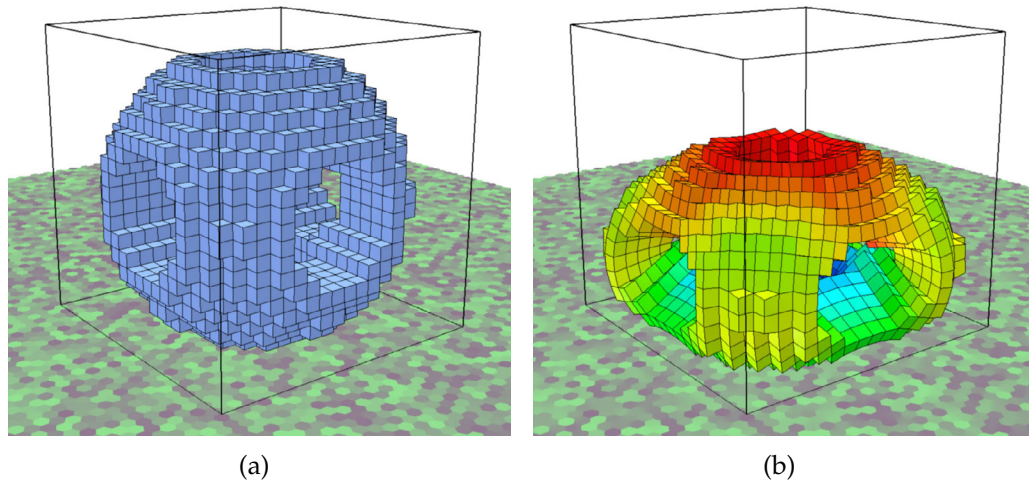
Figure 9.3: Two frames of a hollow ball bouncing under gravity illustrate an undeformed ball just before impact (a) and the color-coded displacement of the highly deformed ball at the moment of highest non-linear deformation (b).

and relatively low stiffness of the materials involved, the physically-based dynamics are often significant and must be modeled (Figure 9.3). Much of the development in simulating soft bodies has been driven by the computer graphics community. Many of the well established physics engines provide support for dynamic deformable bodies, whether 1D rope, 2D cloth, or 3D "Jello" [141, 140].

The goal of these simulations is generally to create realistic visual effects in real time at the expense of accuracy [179, 45, 129, 50]. For instance, lattice shape matching [150] creates visually appealing rubber effects very efficiently and is unconditionally stable. However, the underlying methods are geometrically based, which limits their direct application to quantitative engineering analysis problems. Other simulators are derived from more physically based principles [84], but their performance at predicting real-world behaviors is unverified. Deformable body simulators have also been developed specifically for real-time surgery simulation [36, 200]. These simulators address challenges such as mod-

ifying the geometry dynamically to simulate incisions, but due to the variance of biological materials it is also difficult to verify quantitative accuracy.

There is also a large body of work regarding the simulation of compliant mechanisms and design thereof [107, 137]. Existing efforts focus on small displacements [133] or discrete thin beam members that can flex significantly [156, 73]. In the simulation framework presented here, an entire freeform 3D shape can be non-linearly deformed, leading to many novel possibilities for soft mechanisms that cannot be simulated efficiently using current techniques and tools.

9.3.2 Heterogeneous Deformable Body Core Simulator

A voxel-based mass-spring lattice was chosen to best simulate the dynamics of highly deformable heterogeneous materials. Several measures were incorporated to mitigate the lack of quantitative accuracy normally associated with the discrete particle-based simulations. Each lattice point was modeled with six degrees of freedom. In addition to the traditional three translational degrees of freedom, all three rotational degrees of freedom are stored and updates as part of the state.

It follows that both mass and moment of inertia are stored for each voxel. Instead of using simple extension springs to connect adjacent points, beam elements were used that resist lateral shearing and rotation in all axes in addition to extension. By setting the properties of the beam equal to the equivalent size and stiffness of the bulk material connecting two voxels, a good approximation of the aggregate bulk material behavior is obtained. The details of these beams elements are identical to those used for the linear direct stiffness method.

Each time iteration in the dynamic simulation consists of two steps: (1) calculating all internal forces, then (2) updating all positions. In order to preserve the proper dynamics of the system, positions were updated synchronously so that the order of calculation is irrelevant. It is important to note that even though the same linear Bernoulli-Euler beam theory is used, because of the iterative nature of the simulation non-linear deformations becomes possible. However, the linear nature of the discrete elements implies that even though the physics engine presented here is capable of modeling large aggregate non-linear deformations, the accuracy drops off as the angle between any two adjacent voxels becomes too large for a reasonable small angle approximation.

Unlike the direct stiffness method, a global stiffness matrix is never formed. Instead the force on each voxel from each bond is calculated and summed to obtain the total force on each voxel. In order to reduce the computational complexity, the transformations to rotate each individual element from its initial orientation into the positive X direction were precomputed. Due to the initial voxel lattice constraints, all elements are initially located on principal axes. This makes the transformation into the X-direction stiffness matrix computationally trivial. As the structure deforms over the course of a simulation, an additional transformation was calculated and applied to each element to translate the position and angle of the first voxel to zero. Therefore X_1, Y_1, Z_1, θ_{x_1}, θ_{y_1}, and θ_{z_1} all become zero and drop out of the calculation. The large matrix calculation of the linear direct stiffness case is reduced to:

$$F_{x_1} = -a_1 X_2 \tag{9.14}$$

$$F_{y_1} = -b_1 Y_2 + b_2 \theta_{z_2} \tag{9.15}$$

$$F_{z_1} = -b_1 Z_2 + b_2 \theta_{y_2} \tag{9.16}$$

$$M_{\theta_{x_1}} = -a_2 \theta_{x_2} \tag{9.17}$$

$$M_{\theta_{y_1}} = b_2 Z_2 + b_3 \theta_{y_2} \tag{9.18}$$

$$M_{\theta_{z_1}} = -b_2 Y_2 + b_3 \theta_{z_2} \tag{9.19}$$

$$F_{x_2} = -F_{x_1} \tag{9.20}$$

$$F_{y_2} = -F_{y_1} \tag{9.21}$$

$$F_{z_2} = -F_{z_1} \tag{9.22}$$

$$M_{\theta_{x_2}} = a_2 \theta_{x_2} \tag{9.23}$$

$$M_{\theta_{y_2}} = b_2 Z_2 + 2b_3 \theta_{y_2} \tag{9.24}$$

$$M_{\theta_{z_2}} = -b_2 Y_2 + 2b_3 \theta_{z_2} \tag{9.25}$$

The resulting forces and moments are then transformed back to the current orientation of the bond using the inverse of the transform calculated to arrange them in the positive X axis. The forces for all the bonds are calculated separately, then total forces (F_t) and moments (M_t) on each voxel are summed according to the number of bonds n are connected to it.

$$F_t = \sum_{b=1}^{b=n} \vec{F_b} \tag{9.26}$$

$$M_t = \sum_{b=1}^{b=n} \vec{M_b} \tag{9.27}$$

162

Integration

Because momentum plays a key role in all dynamic simulations, two integrations are necessary to update the position realistically. For this physics engine, double Euler integration was used. Although there are more accurate integration methods, such as the Runge-Kutta (RK4) method, Euler was chosen because the massive number of discrete voxels and non-linear effects such as stick-slip friction are not well suited for the predictive steps of the RK4 integration scheme. The state of each voxel was represented by three dimensional position ($\vec{D}$) and rotation ($\vec{\theta}$) vectors and three dimensional linear ($\vec{P}$) and angular momentum ($\vec{\phi}$) vectors. In order to advance the simulation from time t_n to time $t_{n+1} = t + dt$,

$$\vec{P}_{t_{n+1}} = \vec{P}_{t_n} + F_t dt \tag{9.28}$$

$$\vec{D}_{t_{n+1}} = \vec{D}_{t_n} + \frac{\vec{P}_{t_{n+1}}}{m} dt \tag{9.29}$$

$$\vec{\phi}_{t_{n+1}} = \vec{\phi}_{t_n} + M_t dt \tag{9.30}$$

$$\vec{\theta}_{t_{n+1}} = \vec{\theta}_{t_n} + \frac{\vec{\phi}_{t_{n+1}}}{I} dt \tag{9.31}$$

where m is the mass of the voxel and I is the rotational inertia.

Choosing the Time Step

A critical aspect of implementing a robust physics simulation driven by Euler integration is to choose a suitably small time step to prevent numerical instability. However, in order to be computationally efficient, the time step should not

be unduly small. Fortunately, it is trivial both conceptually and computation-
ally to determine the longest stable time step at each iteration of the simulation.
In an oscillating system, the simulation will be stable if

$$dt < \frac{1}{2\pi\omega_{0_m}} \tag{9.32}$$

Because each bond between voxels is essentially a mass-spring-damper sys-
tem, ω_{0_m} is simply the maximum natural frequency of any bond in the system.
The stiffness of each bond was divided by the minimum mass of either voxel
connected to it to calculate the maximum natural frequency of each bond ac-
cording to

$$\omega_{0_{max}} = \sqrt{\frac{k_b}{m_m}} \tag{9.33}$$

where k_b is the stiffness of the bond and m_m is the minimum of either mass
connected by this bond.

Damping

Once an optimal time step has been chosen, it is necessary to implement damp-
ing into the system to avoid the accumulation of numerical error as well as to
enable realistically damped material properties. Because one application goal
of this simulation involves unconstrained motion of soft bodies, damping must
be included at the local interaction between voxels, not just applying a force
proportional to each voxel's global velocity which would also damp rigid body
motion.

The local damping between adjacent voxels ensures that modal resonances at the scale of a single voxel do not accumulate. For each bond between two voxels, a force was applied to each voxel opposing the relative velocity between them. However, because rotational degrees of freedom allow this bond to be spinning, both angular and translational velocities must be correctly accommodated to make sure rigid body motion is not being damped. For each bond, first the average position, velocity, and angular velocity were calculated. Then the velocity of the second voxel relative to the first ($\vec{V}_{2\to1}$) is calculated according to

$$\vec{V}_{2\to1} = (\vec{V}_2 - \vec{V}_a) + (\vec{D}_2 - \vec{D}_a) \times \vec{\omega}_a \tag{9.34}$$

where $\vec{V}_2$ is the velocity of the second voxel, $\vec{V}_a$ is the average velocity of two voxels, $\vec{D}_2$ is the position of the second voxel, $\vec{D}_a$ is the average position, and $\vec{\omega}_a$ is the average angular velocity. This is in effect subtracting out the rigid motion components of the relative velocity such that they are not damped. Then for each each voxel the damping force is calculated according to the standard linear damping formula:

$$F_d = 2\zeta\sqrt{mk}V_r \tag{9.35}$$

where F_d is the damping force to be applied to the voxels with mass m attached to a bond with stiffness k and a relative velocity V_r. The damping ratio ζ is normally selected to be 1, corresponding to critical damping. Likewise, angular velocities are also damped according to

$$M_d = 2\zeta\sqrt{Ik_\phi}\omega_r \tag{9.36}$$

where M_d is the damping force to be applied to the voxels with a rotational moment of inertia I attached to a bond with rotational stiffness k and a relative angular velocity ω_r. The rotational damping ratio ζ is also normally selected to be unity. However, even though each bond is critically damped locally, the structure as a whole is still quite underdamped. So, each voxel was also variably damped in a similar manner relative to ground according to the situation at hand.

9.3.3   Collisions

Gravity, Floor and Friction Model

In order to properly simulate freely moving soft bodies, gravity is necessary. As the force is summed on each voxel, the mass of the voxel times the acceleration of gravity was subtracted from the vertical component of force. In conjunction with gravity, a floor was implemented to for objects to rest on. Because the maximum simulation time step that can be taken is limited by the maximum stiffness between any two connected masses, the effective normal stiffness of the floor on any voxels in contact with it cannot be infinitely high.

In order to keep the simulation as efficient as possible, the stiffness of each voxel contacting the floor was the stiffness of the floor in that location. Although this allows significant floor penetration in some cases, the qualitative behavior is appropriate. Potential collisions on the resulting overlap with the floor are trivial to detect by simply comparing the vertical position of each voxel to the ground plane, after accounting for the current size of the voxel.

Although a standard linear friction model would provide a relatively realistic simulation, much more interesting and realistic behavior can be observed using a Coulomb friction model. This implies that a voxel at rest with the floor will resist any motion until

$$|F_l| > \mu_s F_n \tag{9.37}$$

where F_l is the horizontal force parallel to the ground, μ_s is the coefficient of static friction between the voxel and the ground plane and F_n is the normal force pressing this voxel into the plane of the ground. A boolean flag is set indicating to the simulation that this voxel should not move laterally, but can still move in the direction normal to ground such that it can be unweighted and then moved laterally. Once the static friction threshold has been exceeded at any given time step, the voxel is allowed to begin motion in the appropriate lateral direction by clearing the boolean static friction flag. The voxel is allowed to move in all three dimensions, but a friction force is applied opposing the lateral direction of motion according to:

$$|F_l| = \mu_d F_n \tag{9.38}$$

where μ_d is the dynamic coefficient of friction. In order to properly detect when a voxel has stopped lateral motion, a minimum motion threshold must be set. Otherwise the voxel will never re-enter the static friction state until the velocity is less than the precision of a floating point variable. In order to detect a stopping voxel, especially one that would change direction and incorrectly bypass the effects of static friction, a voxel is artificially halted if

$$V_l \leq \frac{F_n \mu_d dt}{m} \tag{9.39}$$

where m is the mass of the voxel in question. Because the force of friction ($F_n \mu_d$) is always directly opposed to the voxels lateral velocity (V_l), the voxel is stopped if the projected change in velocity would change the direction of the surface velocity, which would involve the momentary stopping of the voxel. Collisions are also damped normal to the direction of contact with a user variable damping ratio ranging from zero (no damping) to 1 (critical damping).

Self Collision Detection and Handling

Collision detection between voxels must be implemented carefully to avoid creating a major computational bottleneck. Especially in simulations with many independently moving particles, the $O(n^2)$ process of checking every particle against every other to detect collisions is prohibitively expensive in CPU cycles. In a voxel simulation such as this there are many methods to make collision detection more efficient. Since large deformations and multiple bodies are possible, one cannot simply exclude collision detecting between voxels that are connected in a continuous region of material. However, one can immediately assume that voxels on the interior of an object my be disregarded for any collisions, assuming that collisions are handled in such a way that overlaps cannot penetrate the outer shell. Upon import into the simulation, a list of surface voxels is precomputed, since this information will never change.

The next step is to build a list of voxel pairs that are within a collision horizon (reasonable range) of each other. Voxels on this shortlist should be compared

at every time step for potential overlap. The collision horizon was chosen to be a distance equivalent to two voxels. However, it is undesirable to watch for potential collisions between voxels that are immediately adjacent and connected in the lattice, since the internal forces between them already resist penetration. To account for this, a list of voxels within a 3D manhattan distance of 3 in the lattice is precompiled upon import into the simulation for each voxel. Once complete, any potential collision interactions can be compared against this list in linear time to exclude the computational overhead of calculating spurious collision interactions.

Finally, the list of potential voxel collision pairs must be updated often enough that out-of-range voxels beyond the collision horizon do not have a chance to penetrate before being recognized as potential collisions. To accomplish this, a global maximum motion variable is initialized in the simulation. Each time step, the magnitude of the maximum velocity of any voxel in the simulation is added to this variable. While the maximum motion is less than half the collision horizon, it is guaranteed that no voxels can overlap into a collision. This is extremely conservative, but also computationally trivial to compute.

9.3.4 Volumetric Actuation

For convenience, volumetric actuation will be referred to in the context of materials with a non-zero coefficient of thermal expansion (CTE) in conjunction with a changing "temperature" control variable. However, volumetric actuation may be physically achieved in a variety of ways, so there is no reason to assume that the results presented here are applicable only to temperature changes. There is

also no reason that different materials within the simulation couldn't expand or contract out of sync to multiple independent control variables. This would be analogous to having multiple "temperatures" that only affect certain materials. But in the following discussion, only a single temperature control variable will be referred to for simplicity.

With the soft body relaxation engine in place, such volumetric actuation is implemented by simple changing the nominal rest length between adjacent voxels when computing the elastic force between them. If the elastic force (F_E) between two voxels is normally calculated according to

$$\vec{F}_E = K \left(\vec{P}_2 - \vec{P}_1 - \vec{D}_{N_{P_1 \to P_2}} \right) \tag{9.40}$$

to add in the effects of volumetric actuation,

$$\vec{D}_{N_{P_1 \to P_2}} \mid_{T=T_c} = (1 + \frac{\alpha_1 + \alpha_2}{2}(T_c - T_r))\vec{D}_{NP_1 \to P_2} \mid_{T=T_r} \tag{9.41}$$

where $\vec{D}_{NP_1 \to P_2} \mid_{T=T_c}$ is the modified rest distance based on the current temperature, α_1 and α_2 are the coefficients of thermal expansion of the bond's constituent materials, T_c is the current temperature and T_r is the reference temperature, at which there is no temperature-based expansion or contraction. An example of the freeform, large-displacement nature of this volumetric actuation is shown in Figure 9.4 with the yellow material changing volume based on a global temperature variable.
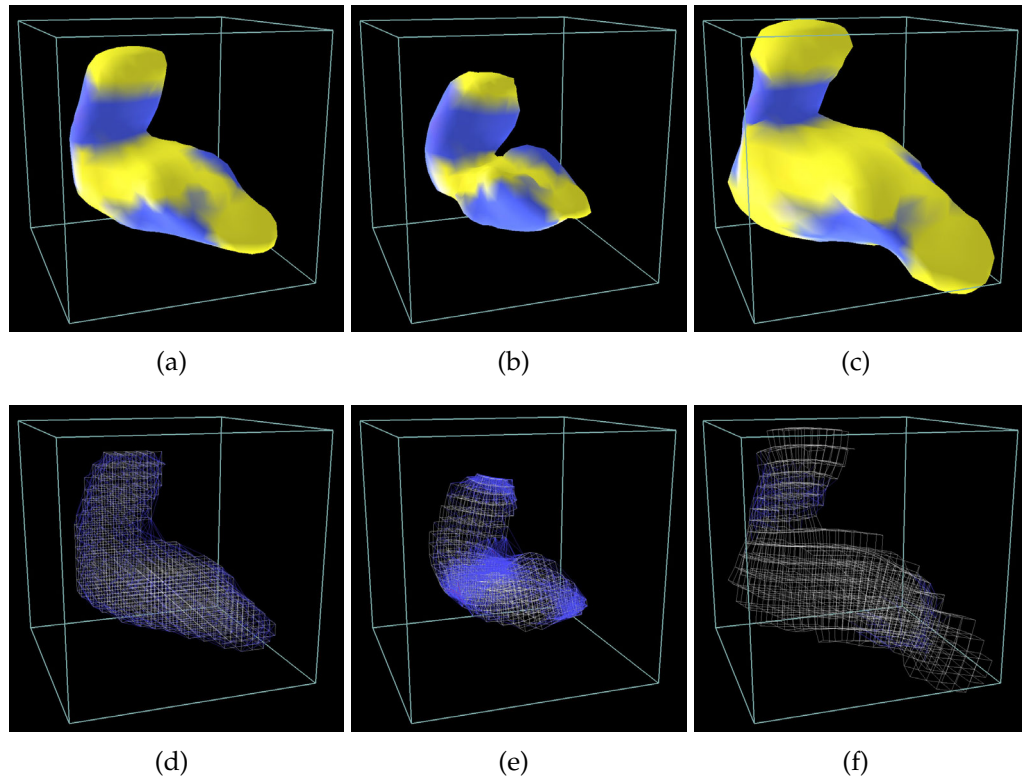
Figure 9.4: A randomly generated 3D object (a) is imported into the soft-body simulation, which voxelizes the object at an appropriate level of detail. (d) Potential self-intersection collisions are shown as blue lines. As the temperature varies, the volume of the yellow material shrinks (b & e) and swells (c & f) accordingly. The volume of the blue material remains constant.

9.4 Voxelyze Simulation Performance

Several parameters were explored to characterize the performance of the soft body simulator. All results presented here assume the simulation is run on a single worker thread of an Intel Core i7 CPU at 2.67GHz. As implemented, the simulation proved very computationally efficient. For a reasonable size object of 4000 voxels, 122 complete simulation iterations were completed per second, or approximately 500,000 voxel calculations per second. As the number of voxels of increases in the object, the total voxels calculated per second decreases, but
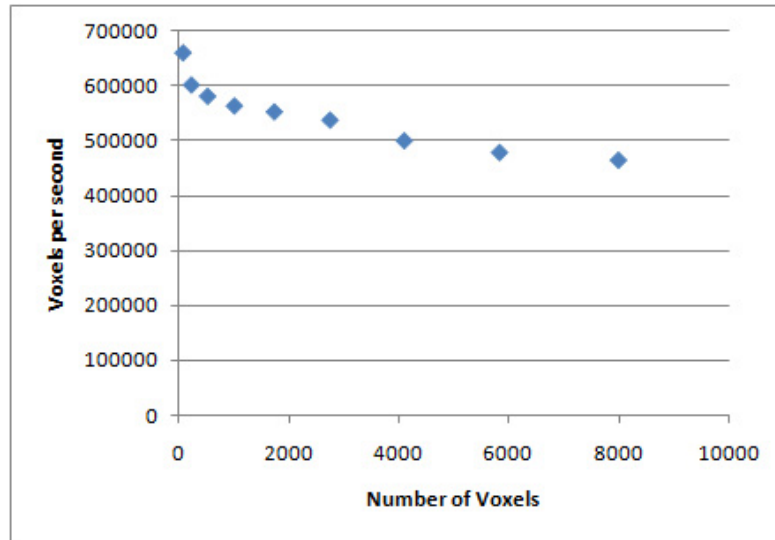
Figure 9.5: The computational speed per voxel drops off slightly as the number of voxels in the dynamic Voxelyze simulation increases.

not dramatically. The simulation speed per voxel for cubic blocks of various numbers of elements are shown in Figure 9.5.

9.4.1 Effects of Local Bond Damping

By applying a combination of damping both to the individual bonds and to the voxels relative to ground, the solution converges quickly to steady state with very little numerical jitter. The addition local damping does not significantly affect the convergence speed, but allows the solution to converge to a residual static error approximately 7 orders of magnitude lower (Figure 9.6). By suppressing jitter in this manner, both static and dynamic solutions are much less susceptible to numerical instability.
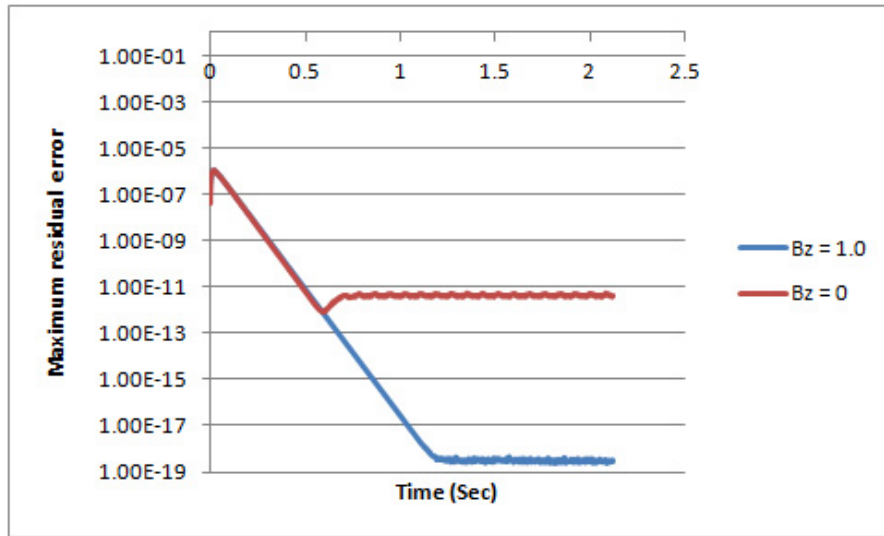
Figure 9.6: Damping individual bonds critically ($B_z = 1.0$) lowers the noise floor by approximately 7 orders of magnitude compared to the undamped case ($B_z = 0$).

9.4.2 Speedup of Self Collision Schemes

Different combinations of self collision detection methods were directly compared using the test geometry shown in Figure 9.7. All materials in this setup were defined with a stiffness of 1MPa. The red and blue materials each were assigned to have a thermal expansion coefficient with magnitude of 0.02, although one was positive and one negative. The temperature of the environment was then sinusoidally varied with an amplitude of 30 degrees, which corresponds to a 60% expansion and contraction of the red and blue materials 180 degrees out of phase. This sets up a periodic collision between the extremities that are repeatedly entering and exiting the assigned collision horizon.

The results of the collision experiments are shown in Table 9.1. The case of comparing the distance from all voxels to all other voxels in the structure at every time step (All+Every) was included as a baseline. Comparing only surface
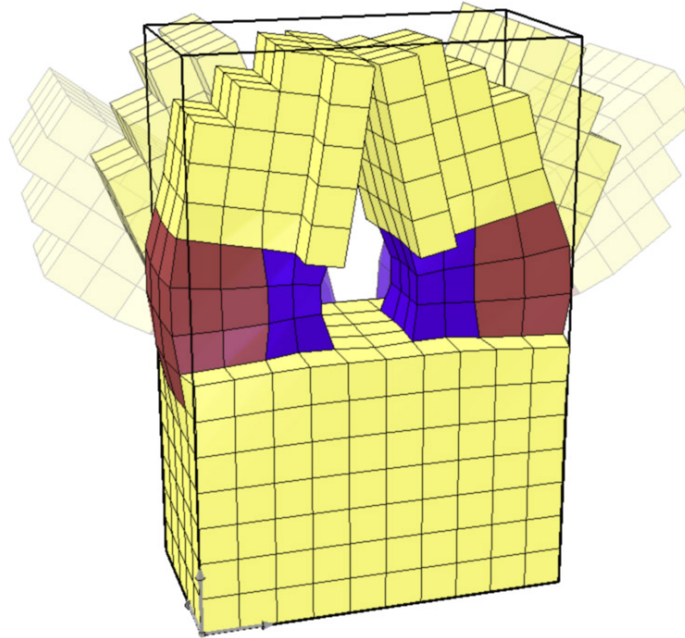
Figure 9.7: An arbitrary clapper setup to test net iteration rates with different collision types. The red and blue materials change volume sinusoidally 180 degrees out of phase to provide the actuation.

voxels to each other at every time step (Surf+Every) resulted in a very minor speedup. This is expected when examining the chosen geometry because the majority of voxels in the structure are surface voxels. However, large gains in speed are realized when incorporating the collision horizon. Even when comparing all voxels to all voxels whenever recalculation is needed (All+Horizon), the simulation as a whole speeds up almost 6x. Again, minor acceleration is realized in this case when considering only surface voxels with the collision horizon (Surf+Horizon).

It should be stressed that these results are merely representative. In cases with larger numbers of voxels, the bottleneck of the entire simulation is in collision detection. In this case, comparing all voxels to all other voxels is $O(N^2)$ (where N is the number of voxels) which dominates the $O(N)$ scaling of calcu-

Table 9.1: Iteration rates for various collisions detection and handling schemes of the dynamic voxel simulator (Higher is better). The introduction of a collision horizon (Horizon) vastly speeds up the simulation whether comparing all voxels to each other (All) or just those on the surface (Surf).

Geometry	Rate (Iter/sec)
All+Every	140.4
Surf+Every	140.6
All+Horizon	834.4
Surf+Horizon	835.4

lating forces and updating positions. In this case, precompiling a list of surface voxels and only using them in collision detection can reduce collision detection down to approximately $O(N^{1.5})$, although this depends on the geometry. Of course, if the geometry is thin such that all voxels are on the surface, no speedup will be observed.

Additionally, the incorporating the collision horizon has extremely variable speedup up the simulation. In the extreme case of voxels moving very fast, no speedup is observed, since the collision horizon may be exceeded every time step. However, this is only possible in extremely fast rigid body motion collisions, for which this simulation is not intended. On the opposite end of the spectrum, if the object is stationary or moving slowly, the collision horizon will take many, many time steps to be exceeded, so almost no collision detection calculations will be needed, resulting in dramatic acceleration of the simulation.

9.4.3   Demonstrations of Volumetrically Actuated Mechanisms

Several demonstration scenes were created to illustrate the simulator in action. In the first scene, (Figure 9.8a-c) an actuated beam kicks a ball into a bowling pin. The beam has a stiffness 10 times great than the ball, which in turn has a stiffness 10 times greater than the bowling pin. The frequency of the red and blue volumetric actuation was selected such that the beam would swing in resonance. The second scene (Figure 9.8d-f) shows a 2D layer of voxels falling and interacting with a fixed sphere as cloth would. In the third scene, a quadruped with periodic leg actuation walks forward using the nonlinearities of the surface friction with the floor as well as the side-to-side resonance of the head swinging back and forth. These illustrations demonstrate not just the dynamics and large deformation of capabilities of the simulation, but also the use of volumetric actuation and the robust yet efficient collision system.

9.5   Static Deflection Validation

In order to ensure that the physics engines were performing properly, the static behavior of cantilever beams in simulation were compared to analytical solutions. To verify the static behavior of the simulation, beam deflections of both thin and think cantilever beams were computed and compared for the linear direct stiffness method, the dynamic time simulation method, and (in the case of the thin beam) to the analytical solution. The results are outlined in Table 9.2. For the thin beams, $20 \times 1 \times 1$ voxels were used with a physical size of 1mm each, for a total beam size of 20mm long by 1mm thick. A material stiffness of 1.0MPa was specified. The force at the end of the beam was selected to be 0.03mN so
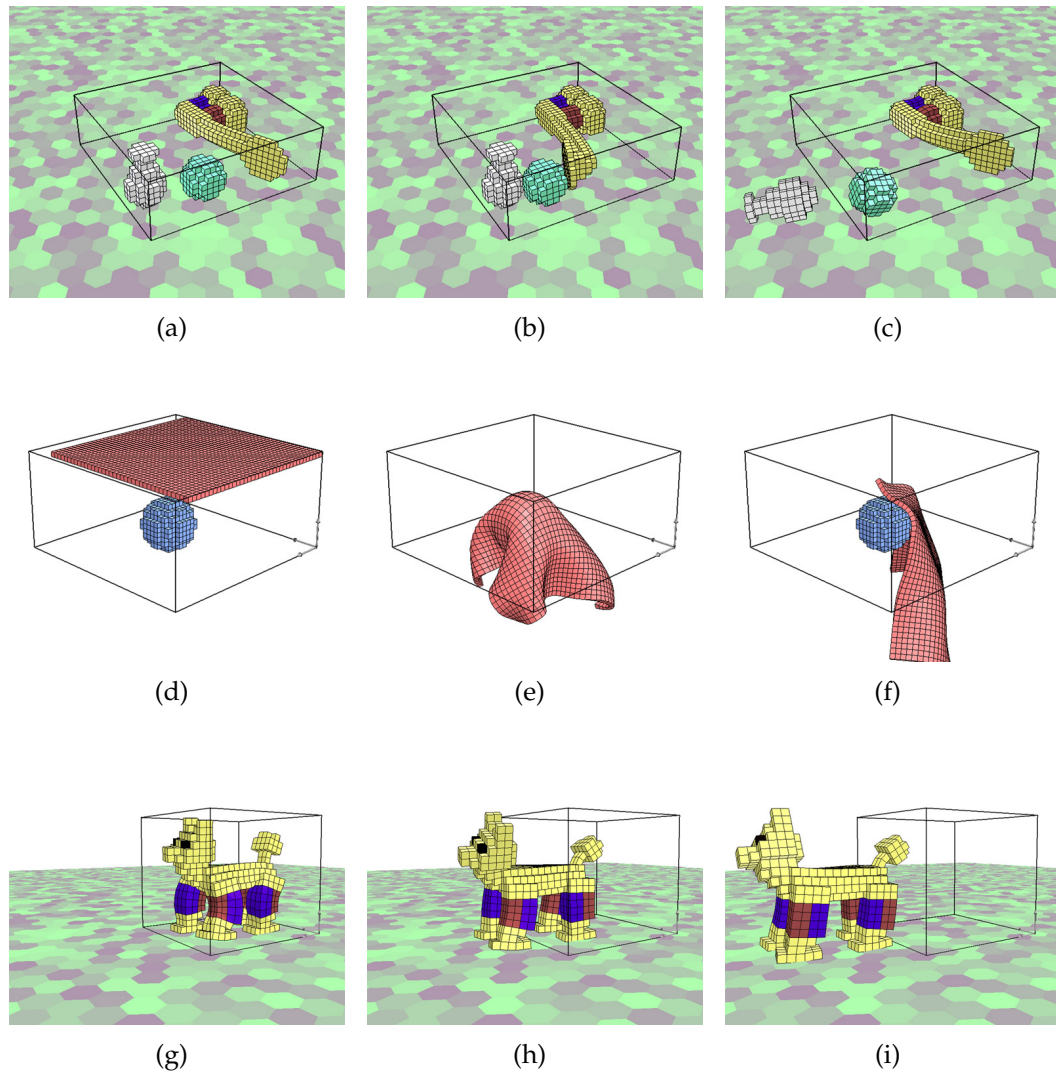
Figure 9.8: Frames from demonstration scenes show a volumetrically actuated flexible beam swung in resonance to kick a soft ball into an even softer pin (a-c), a 2D layer of voxels falling under gravity and interacting with a fixed sphere (d-f), and a locomoting quadruped (g-i).

Table 9.2: Comparison of maximum displacements of thin and thick cantilever beams demonstrate good agreement.

Geometry	Mass/spring	Direct stiffness	Analytical
Thin cantilever beam	0.822mm	0.823mm	0.823mm
Thick cantilever beam	0.538mm	0.546mm	N/A

that the displacement would be small (less than a voxel-height). This ensures that small angle approximations of the analytical solution are valid. The thick beams were modeled as $10 \times 5 \times 5$ blocks of 1mm voxels with the same stiffness. In this case 0.1N of force was applied at the free end to achieve a non-negligible displacement.

The non-linear mass/spring method presented here results in slightly smaller displacements than the direct stiffness method and the analytical solutions (Table 9.2). This difference is negligible in the thin beam case. The difference is more pronounced in the thick beam case. This is likely because the deformation (Figure 9.9) is large enough that the change in geometry in the relaxation method factors in to the results. Therefore it is likely that in this case the linear methods slightly over-predicts the deflection.

9.6 Dynamic Response Validation

To verify the dynamic properties of the Voxelyze simulation, a thin cantilever beam of the same dimensions and properties as used in the static deflection verification section was excited with an impulse force at the free end. Damping was turned off, except a trace amount of local bond rotational damping ($\zeta =$

(a)	(b)
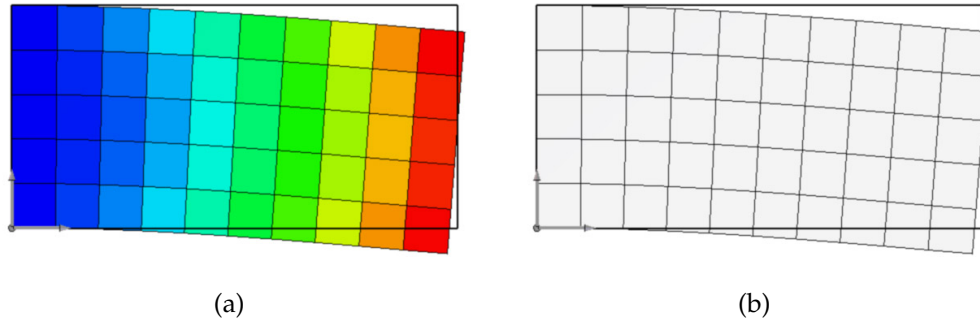
Figure 9.9: The deflection of a thick cantilever beam as calculated by the direct stiffness method (finite element analysis) (a), and the mass / spring method (b).

0.01) to maintain numerical stability. 20,000 data points were collected of the z position of the voxel at the free end of the beam, corresponding to 0.13 seconds of physical time, or about 60 oscillations of the lowest-frequency fundamental mode. The frequency characteristics of this data were then plotted with the analytically calculated natural frequencies of a thin cantilever beam. (Figure 9.10) The analytical natural frequencies were calculated according to

$$\omega_n = K_n \sqrt{\frac{EI}{\bar{m}L^4}} \tag{9.42}$$

where ω_n is the natural frequency of mode n in radians per second, K_n is the standard scaling factor for the mode in question, and $\bar{m}$ is the mass per unit length of the beam [187]. These values are overlaid on Figure 9.10 and the simulated and analytical natural frequencies are tabulated in Table 9.3. The simulation predicts natural frequencies that are slightly lower than those predicted by beam theory. This is likely because the 20x1 aspect ratio of the simulated beam is not quite an ideal thin beam, and because there is a small amount of damping in the simulation which would tends towards under-predicting natu-

Table 9.3: The modal frequencies of a simulated thin beam and analytically calculated results agree well, with the dynamic mass/spring simulation under-predicting slightly due to damping and having a finite thickness.

Mode	Analytical	Mass/spring
1	389Hz	404Hz
2	2428Hz	2531Hz
3	6749Hz	7087Hz
4	13070Hz	13890Hz
5	21260Hz	22960Hz
6	31180Hz	34290Hz

ral frequencies.

9.7   Open Source Implementation of Voxelyze

All code and documentation for the voxelyze dynamic simulator is freely available at www.voxcad.com, including a standalone GUI for editing and simulating objects in a real-time interactive environment. This simulation opens the door to the design automation of a wide variety of non-linear physical structures and mechanisms that were not possible with previous soft-body physics simulation packages. By leveraging the crowd-sourcing of the open software world, interested developers can contribute not only to the core of the simulator, but also apply it to many interesting applications.
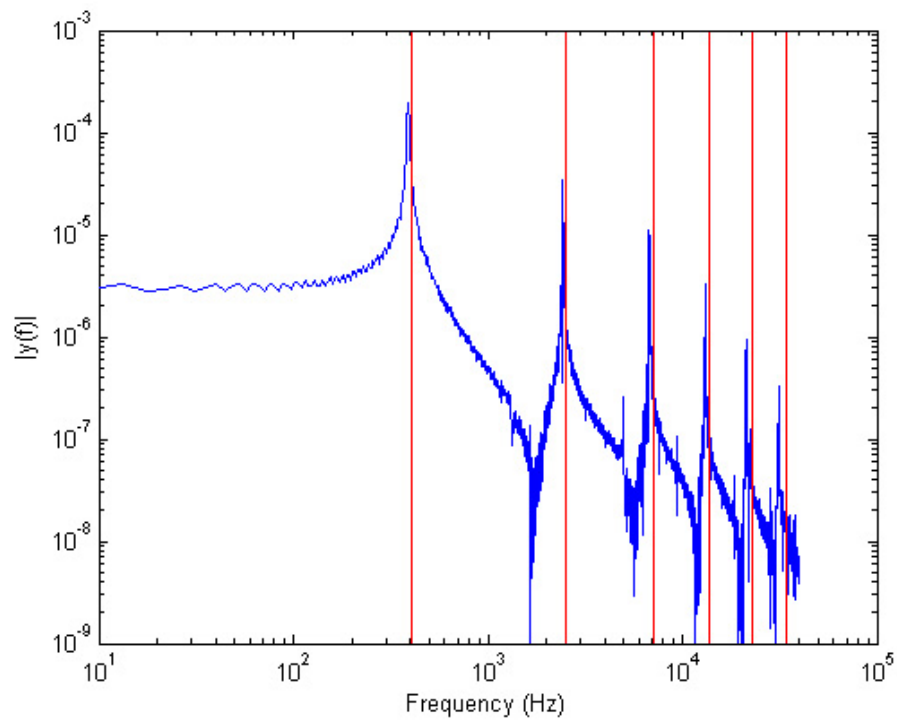
Figure 9.10: The dynamic frequency response of a simulated cantilever beam (blue) with low damping clearly shows modal resonances that agree well with analytically calculated values (red overlay lines) which serves to validate the dynamic behavior of the simulation.

181

Part IV

Design Methods

CHAPTER 10

MANUAL DESIGN METHODS

10.1   Voxel Design Process

The process of designing and representing objects in the digital voxel domain
has been well explored for single material shapes. Sophisticated software tools
exist for freeform sculpting density fields which are then meshed using the
marching cubes algorithm [110]. However, all these tools are focused on mod-
eling and generating meshes for rendering purposes. Here the design process
is explored for creating and simulating objects specifically for multi-material
voxel-based printing. The main challenges here lie in how to efficiently specify
and create models with multiple interspersed materials with specific micro and
macro structure.

Much work has been done with voxel data structures for purposes of visual-
ization [194, 68], rendering [101], storage [13], manipulation [56], and compres-
sion [196] of 3D shapes. Unlike surface meshes or boundary representations
which contain only information about the *surface* of an object, voxel models in-
herently contain information about *volume* of an object. Voxels can be used for
volumetric modeling in a manner that simulates how manipulates a lump of
clay in the real world [56]. This ensures a topologically correct model and re-
duces uncertainly about normals and other issues with triangle meshes.

A front-end GUI (graphical user interface) and editor was created for edit-
ing multi-material voxel objects and analyzing them using the both the static
and dynamic voxel simulation engines presented earlier (Figure 10.1). This
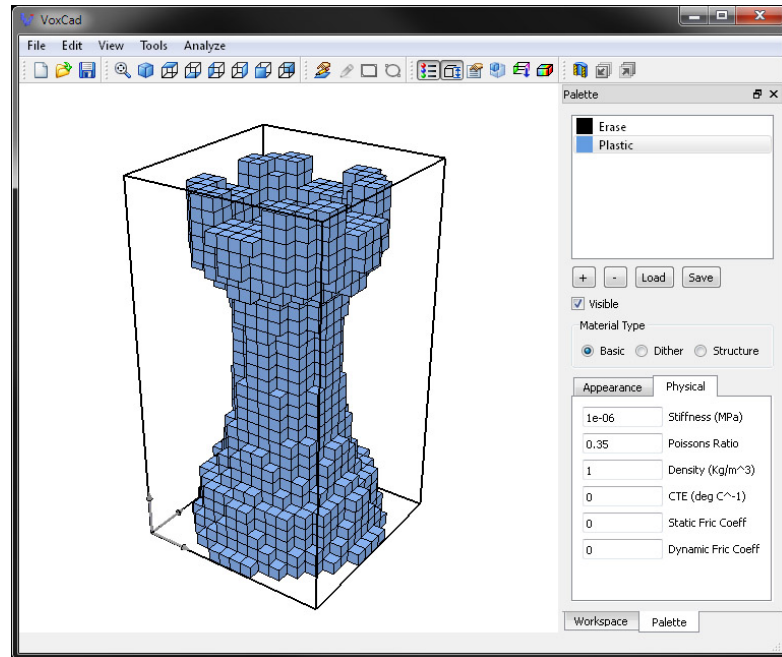
Figure 10.1: The basic user interface of VoxCad uses a main 2D/3D view to view and edit voxel objects. Toolbars to the right control various parameters.

project was written in C++ using the QT open-source cross platform user interface libraries and open sourced under the name "VoxCad" [65]. The simulation engines were open-sourced under the name "Voxelyze" [66] in platform-independent C++. This will allow developers of independent applications to make use of these voxel analysis tools and incorporate them into their own programs.

10.2 Basic User Interface

The goal of this research was not to compete with existing established software tools in modeling the form of objects. To that end, only basic voxel manipulation functions were included. In order to best specify both the external in internal

184
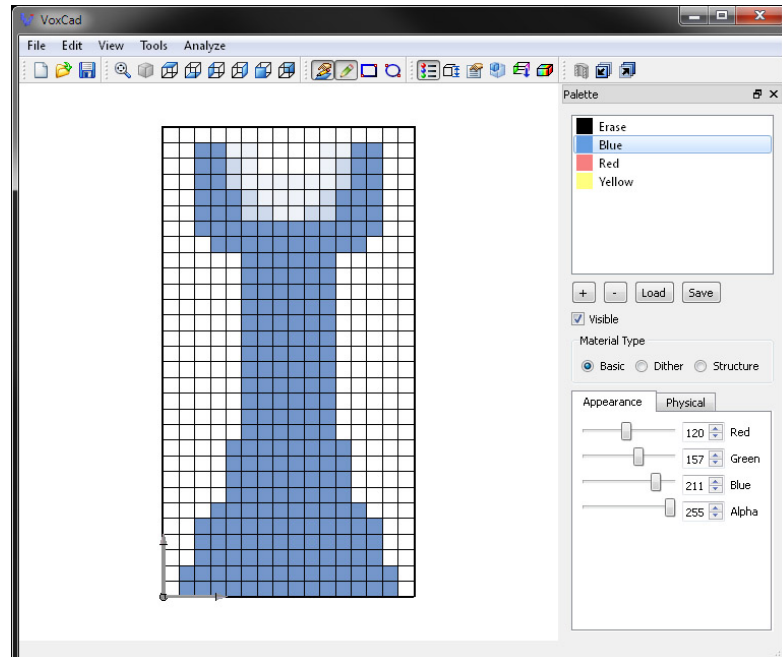
Figure 10.2: The main method of editing voxel objects involves creating a slice plane, then using drawing tools to add or remove voxels in this plane.

voxel bitmap, the editing of voxel object revolves around the idea of a slice plane. This is one way to resolve the ambiguity of resolving the inherently 2D input on a computer (i.e. screen coordinates) into sensible 3D input.

When in editing mode, the user can select which of the six cartesian planes (front, back, right, left, top, bottom) to edit using. An editing plane is then defined parallel to this plane which can be moved back and forth to intersect and edit different layers of the model. All voxels within this layer are displayed on the screen in a 2D view, with any potentially occluding voxels hidden and voxels in layers behind the current layer faded out to reduce ambiguity.

Once the desired slice plane is defined, the user has a choice of paint-style tools to place voxels of appropriate materials in the desired locations. A pen tool draws a line of voxels of the current material following the mouse cursor.

The box tool and the ellipse tool create filled regions of their respective shapes based on the starting and stopping positions of a mouse click and drag.

In most cases, it is inefficient to specify an entire volume of an object by selecting each voxel individually. If multiple layers are identical, they can simply be copied and pasted. However, to make more sophisticated models in the VoxCad interface, 3D brushes are critical. In the 3D brush mode, the user selects a primitive such as a box, cylinder, or sphere. A series of sliders allow this primitive to be scaled and positioned within the current voxel workspace. Once positioned, the user selects the desired material and applies the brush. All voxel locations contained within the primitive shape are set to this material or erased if desired. The brush can then be adjusted and the process repeated as many times as necessary.

This greatly facilitates the creation of regular, geometric objects. In cases where more sophisticated 3D shapes are needed quickly, a pre-existing surface mesh may be imported as a 3D brush. This allows users to quickly leverage existing CAD designs and import them into VoxCad at the desired resolution and even create multiple instances for more complex models (Figure 10.3).

10.3   Lattice Description

Existing voxel modeling programs make use of cubic lattice structures for cubic voxels. However, in physical voxel manufacturing the fundamental volume elements are not limited to cubes. In fact existing processes where physical voxels are specified such as Objet's polyjet technology and the discrete voxel deposition tools presented in previous chapters do not make use of cubic lattices.
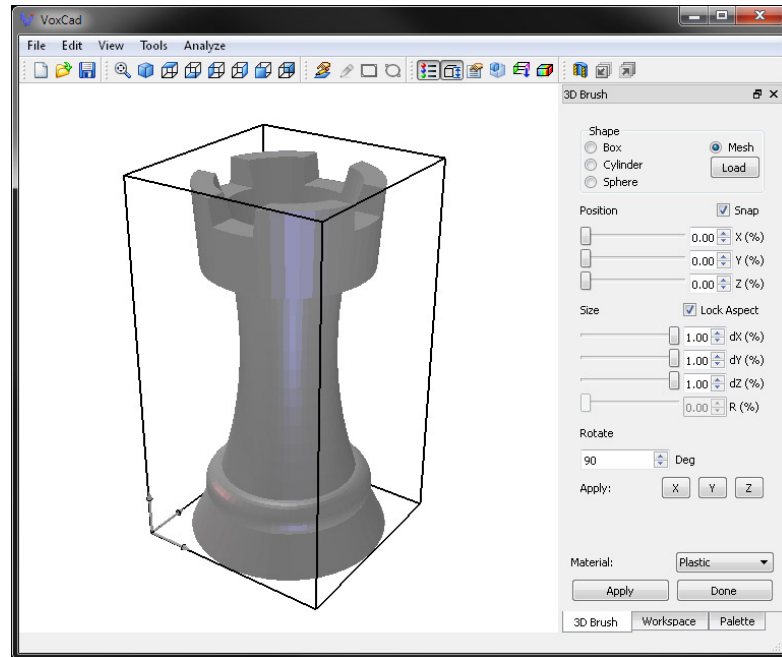
Figure 10.3: An efficient way to edit voxels in 3D involves the use of a 3D brush. Here, a rook-shaped brush has been imported from a mesh file. Applying this brush results in the rook shown in Figure 10.1.

This brings up the question of how to best represent an arbitrary lattice in 3D. In the case of rectangular prismatic but non-cubic voxels, the distances between adjacent voxels can simply be scaled appropriately. However, other lattices such as face-centered cubic and hexagonal close packing cannot be so easily incorporated. Here, the assumption is made that objects are to be made in a layered manufacturing process. Thus, voxel lattice positions should be largely grouped into planar layers.

To address arbitrary lattices, idea of line and layer offsets are used. To create arbitrary layer patterns, line offsets and dimension adjustments are used. Starting with the lattice location at the origin of the part (index: 0,0,0), the position of each voxel within a given layer is shifted by a line offset in X and Y multiplied by its y and x index, respectively. The line offset are stored as a percentage of the
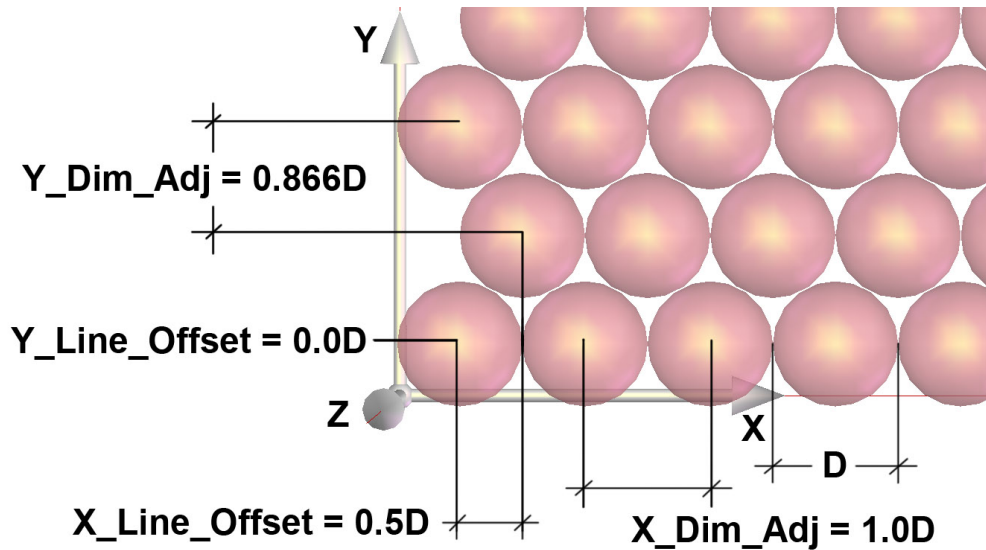
Figure 10.4: The in-layer parameters for a hexagonal lattice are shown for spherical voxels shapes. The diameter D is equivalent to the global lattice dimension l in this case.

base voxel lattice dimension l. When calculating the offset of a given voxel, off-sets should not be greater than the lattice dimension to avoid non-rectangular layer workspaces. The modulo operator is used to calculate the correct offset within the range of zero to the lattice dimension. Additionally, the offset of each row and column within the layer can be adjusted from its nominal value of 1.0 to shift the voxels closer together of further apart. Figure 10.4 illustrates the in-layer parameters needed to define a hexagonal grid of spheres.

Offsets between layers are addressed similarly. The offset of all voxels within a layer for each successive layer is shifted by the layer offsets in X and Y. The X and Y offsets again should not be greater than the inter-voxel spacing. All offsets (both line and layer) are calculated and summed before the modulo operator is applied. Figure 10.5 illustrates the additional parameters needed to define a 3D hexagonal close-packed lattice structure. For any voxel, the real-valued position at the index (i, j, k) can be easily computed. The X position can be found by:
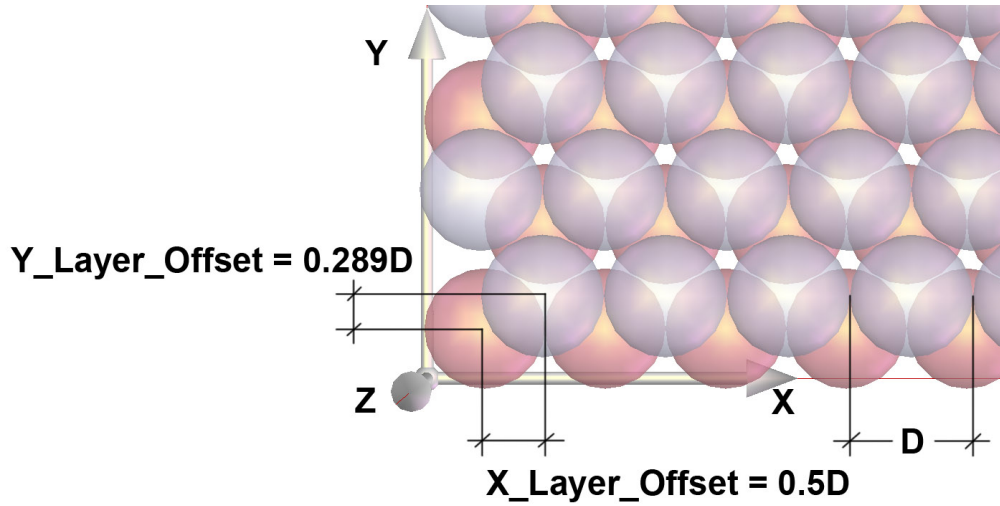
Figure 10.5: The inter-layer parameters for a hexagonal close-packed lattice are shown for spherical voxels shapes. The purple spheres rest in the layer above the red spheres.

$$X(i, j, k) = (i + (O_{nx}j + O_{rx}k) \% 1.0)S_x l \qquad (10.1)$$

where O_{nx} is the X line offset parameter, O_{rx} is the X layer offset, and S_x is the X dimension adjustment factor. Likewise the Y position is found by

$$Y(i, j, k) = (j + (O_{ny}i + O_{ry}k) \% 1.0)S_y l \qquad (10.2)$$

with analogous parameter descriptions. The Z position is simply

$$Z(k) = kS_z l \qquad (10.3)$$

where S_z is similarly the Z dimension adjustment factor.

This description of lattice geometry is paired with the user choice whether

189

Table 10.1: Lattice specification parameters for common lattice types.

Lattice	Cubic	2:1:4 Aspect Rectangular	Face Centered Cubic (FCC)	Hexagonal Close Packed (HCP)
X Scale	1.0	1.0	1.0	1.0
Y Scale	1.0	0.5	1.0	$\frac{\sqrt{3}}{2}$
Z Scale	1.0	2.0	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{\sqrt{3}}$
X Line Offset	0.0	0.0	0.0	0.5
Y Line Offset	0.0	0.0	0.0	0.0
X Layer Offset	0.0	0.0	0.5	0.5
Y Layer Offset	0.0	0.0	0.5	$\frac{1}{\sqrt{12}}$

to draw the voxels as rectangular prisms, cylinders, or ellipsoids. Depending on the lattice in effect, these primitives may need to be scaled differently than the lattice dimension scaling. For example, spheres in hexagonal close packed arrangement are scaled closer together in two of three axes, yet should still be drawn as true spheres to retain visual contact at all interfaces. To this end, separate scaling factors are included for visualization purposes.

This method of representing lattices allows all the main lattice types to be represented with these parameters. A summary of the appropriate values for each variable are shown in Table 10.1 for several common lattice types (Figure 10.6).
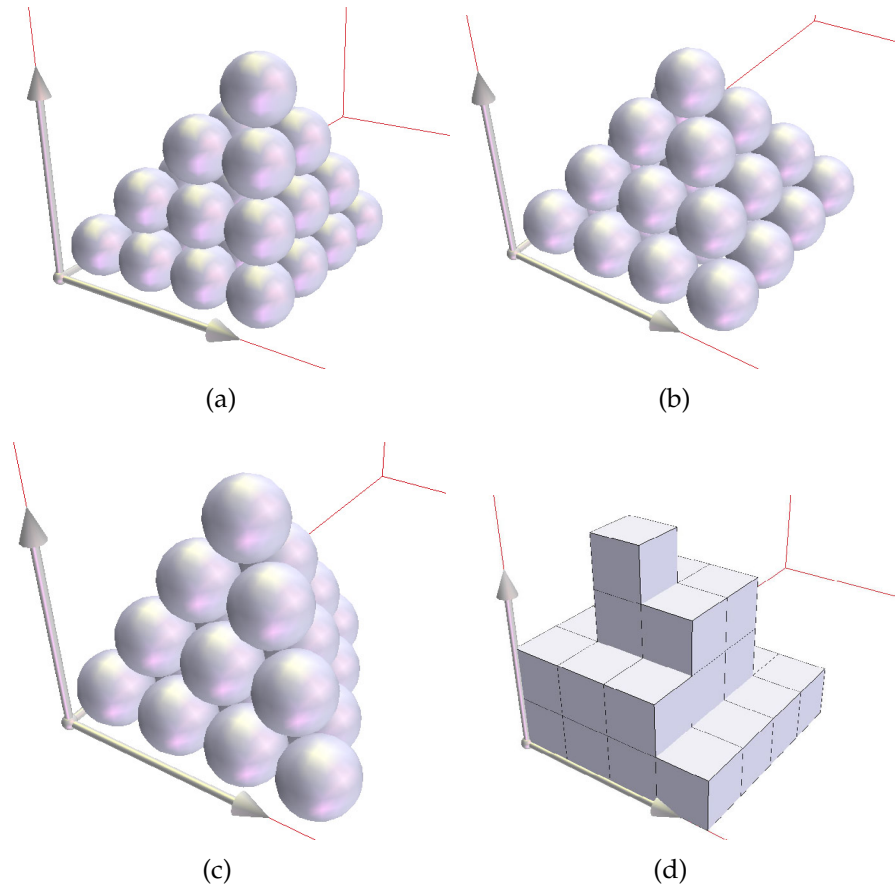
Figure 10.6: The lattice specification parameters allow for a variety common lattice types such as cubic (a), face-centered cubic (b), and hexagonal close packed (c). The voxel display shape is independent of the lattice, as shown by the display of cubes instead of spheres for a cubic lattice (d).

10.4 Material Palette

The appropriate handling of multiple interspersed materials is paramount to the success of creating and simulating useful digital objects. This revolves around the setup and use of a material palette. The material palette is a user-created list of all materials that are used in a particular object. Each material, including a blank (empty) material has a discrete index, and the digital object is defined at each location in the lattice with a material index. Each material has associated

properties. For visualization, a material is associated with a color and transparency. However, each material also has associated material properties that are used when physically simulating an object. These include stiffness, Poisson's ratio, density, coefficient of thermal expansion, friction coefficients, etc. These material properties are then used for their respective regions in simulation space. Up to 64 materials can be defined, which far surpasses the capabilities of current manufacturing processes.

Materials in the palette may be defined as a basic material, as described above, or alternatively as a composite material. There or two types of composite materials: dithered and structures. Each of these meta-materials can be painted in the 3D workspace just like a basic material. However, an algorithm iteratively propagates through the composite material tree for each voxel until it reaches a base material to display and use the properties from. Each composite material is described in more detail below.

10.4.1 Dithered Materials

With a limited set of discrete colors or materials, intermediate pseudo materials with intermediate properties can be easily obtained by randomly mixing voxels of the constituent materials in a prescribed proportion. If the resolution is high enough, these composite materials approach the continuity of basic materials but with highly tunable properties anywhere between the two constituent materials as detailed in Chapter 8.

To this end, a dithered material can be defined in the material palette that references two other materials. A user selectable proportion determines to how

much of each material ends up instantiated. The actual placement of each material is determined by a pseudo-random spatial mapping (PRSM) algorithm. If the algorithm simply picked a random number each time it evaluated the geometry, the resulting distribution of materials would be constantly changing, even if a file was simply saved and then re-opened. For the sake of repeatability, this is undesirable.

The PRSM method takes a real-valued input for X, Y, and Z and outputs a pseudo-random number based on these three parameters. Subsequent calls to a PRSM function with the same X, Y, and Z parameters will always yield the same random number. This allows a random sampling to be deterministically defined for consistent material placement.

To create random distributions of more than two materials, the user can create dithering of dithered materials. For instance, if four materials are to be evenly distributed, a composite U would be created with an even ratio of materials A and B. Then a composite V would be created with an even ratio of material C and D. Finally, a composite material W would be created with an even ratio of material U and V.

10.4.2   Defining Microstructure

In some cases, more control over internal material distribution is desired than simply random proportioning between materials. Here the idea of creating voxel microstructure is introduced. To create a composite material with micro structure, a unit cell is defined in units of voxels. A separate editor can then be opened up with a list of available materials that would not result in infinite
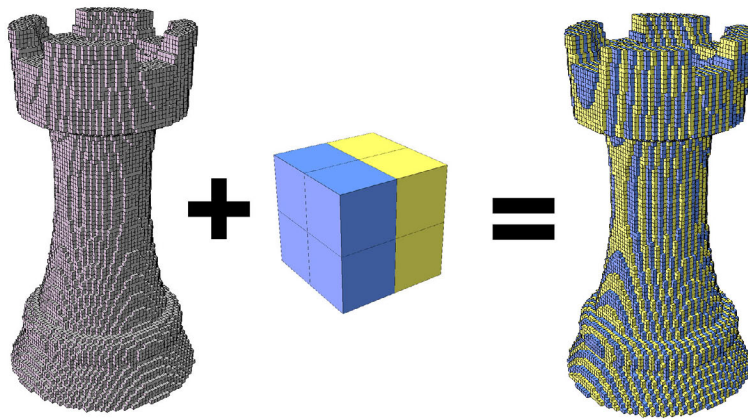
Figure 10.7: Microstructures allow a repeating pattern of materials to be patterned throughout a structure. Changing the microstructure pattern automatically updates the macro structure.

recursion when resolving composite materials. The user then designs the unit cell with the desired pattern. All voxels specified to be a given composite microstructure reference this structure to determine the correct voxel to evaluate for drawing and simulation (Figure 10.7).

As with dithered materials, microstructures can be defined hierarchically, with no constraints on the size of any repeating unit cell (Figure 10.8). As expected, dithered materials can also reference structure materials and vice versa, as long as no circular references are created that would lead to infinite recursion.

10.5 Running Simulations

In many cases, certain regions of the workspace must be constrained as fixed, constrained with a specified displacement, or have a force applied to them. Using the same interface as the 3D brush tool, a list of such regions can be defined that make use of rectangular prismatic, cylindrical, spherical, or arbitrary mesh
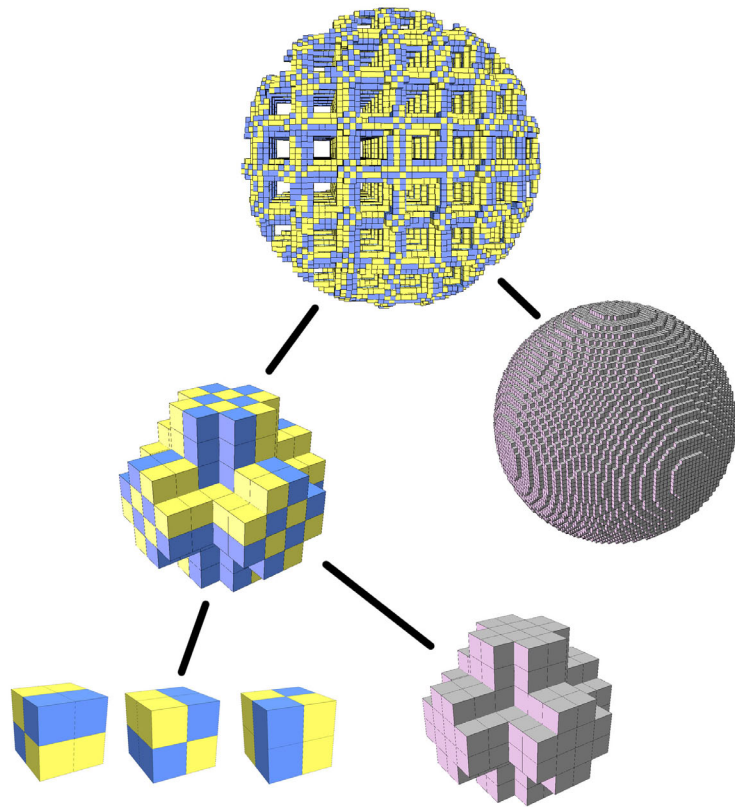
Figure 10.8: Microstructures can be defined hierarchically to create powerful, fully editable micro and macro structures. Here, three simple microstructures are applied to a simple intersection of cylinders. This intermediate microstructure can then be applied to a macroscopic geometry.

surfaces. Each regions can be defined as fixed to ground (immovable) with or without a prescribed displacement, or to have an external force applied to them. There are no practical limits to the number of such regions that can be created. This allows a user to set up a relevant loading scenario to explore the reaction of the structure to mechanical or thermal inputs in order to make hypothesis about its corresponding behavior in the physical world. Figure 10.9 illustrates an example loading scenario.

There are several GUI options to perform mechanical analysis using both the
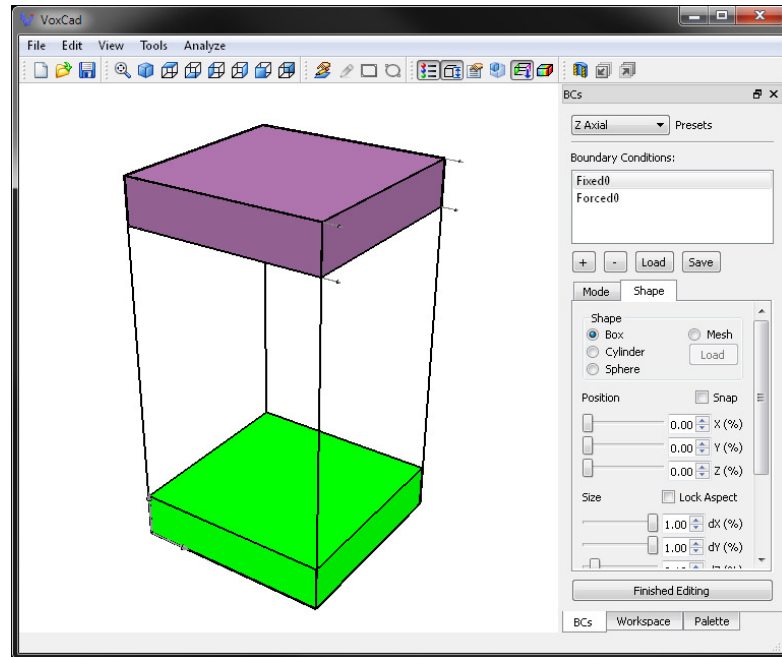
Figure 10.9: Boundary conditions can be set to model how a voxel object should be fixed or forced in simulation. Voxels in green regions are fixed to ground and voxels in purple regions undergo the specified applied force(s).

static and dynamic solvers discussed earlier. A basic static analysis performs the finite element analysis on the structure, and outputs a color-coded visualization of the deformed structure according to the magnitude of displacement, internal stress, or internal strain. This magnitude can be filtered to show only the X, Y, or Z component, or the 3D maximum. In order to form a solvable system, at least one voxel must be fixed to ground. In order to obtain any sort of displacement, at least one other connected voxel must have a non-zero prescribed displacement or applied force.

To explore the dynamic behavior of the structure, there are two options. The first is a simple physics sandbox, which allows the user to interactively push and pull on the structure in addition to the specified boundary conditions (Figure 10.10). Additional parameters can be included such as gravity, damping

196

coefficients, collision detection, or the inclusion of a floor. Free-moving and rotating objects can be obtained by including no fixed or forced regions. Complete stress-strain tests can also be performed by automated stepping of a displaced region in relation to a fixed region. With small steps and time between them for the structure to settle, a good approximation of a slow tensile test can be achieved.
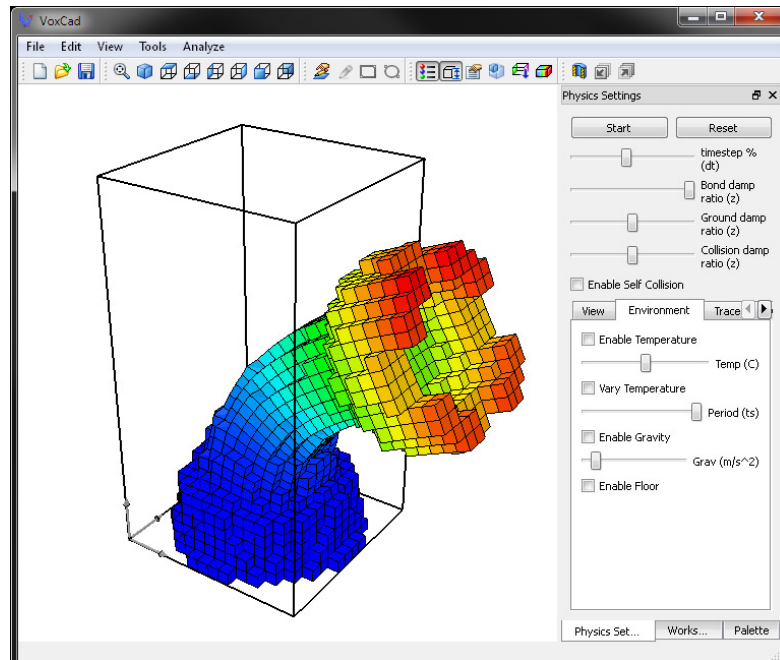
(a)
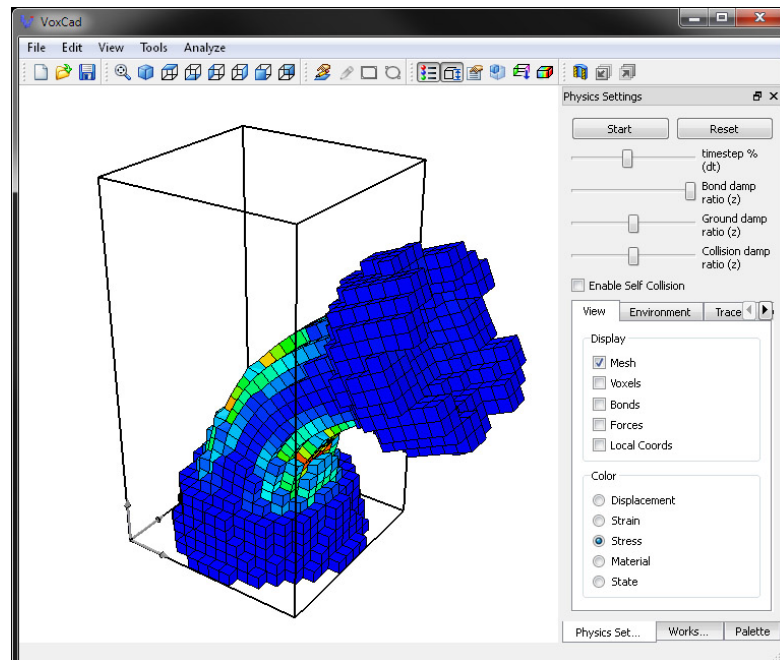
(b)

Figure 10.10: A rook shape undergoing physical analysis in the sand-
box using the large-deformation dynamic voxel simulator.
Color-coded displacement (a), internal stress (b) and other
relevant results can be visualized.

CHAPTER 11

REPRESENTATIONS FOR EVOLVING FREEFORM MULTI-MATERIAL

SHAPES

11.1   Desirable Representation Features

In order to efficiently leverage the power of evolutionary design automation algorithms, the algorithm must have a suitable means of creating and manipulating 3D geometry. There are many possible methods and paradigms for representing three dimensional freeform shapes for an evolutionary algorithm. It is desirable for the representation used within a evolutionary algorithm to define a desired sub-space of sensible objects using a minimum number of parameters [92]. Kicinger et al summarizes various graph structures, generative encodings, and constrained bit-wise encodings that have been proposed to address these challenges [94].

Here representations suitable for freeform, multi-material shapes are considered. Freeform implies that there are minimal constraints on the shape or topology of the object. Surfaces are not limited to flat planes or circular profiles. Instead, complex 3D curvatures are possible unbounded by orientation, radius, or topology. Such shapes have the fewest constraints, yet can naturally be built voxel-by-voxel in a digital fabrication process. The desirable properties for potential representations are enumerated below:

1. **Multiple materials:** The representation should naturally generate objects of multiple materials without imposing constraints on the geometrical internal material distribution. Ideally, a representation should work equally

199

well for single material objects as well as objects with several materials.

2. **Topology independent:** The representation should not be limited to a given topology. Different problem domains may be best solved with very different shapes, including multiple interspersed regions of different materials. To enumerate and optimize all possible topologies would be prohibitive. Thus a representation should be able to seamlessly adapt to different solution topologies.

3. **Smooth shapes:** The general scale of allowable features should be limited to equal or greater than the scale of the voxels that it will eventually be simulated or fabricated with. This implies that a representation produces smooth shapes at some arbitrary level of detail. Otherwise, irrelevant information will be taking up space and optimization effort in the representation.

4. **Minimum parameters:** The ability of an evolutionary algorithm to solve a given problem depends largely on the number of free parameters. The dimension of the problem to be solved is on the order of the number of parameters in the representation. Due to the curse of dimensionality, the number of parameters should be minimized.

5. **Evolvable:** This is the most abstract requirement, but generally specifies that a small change in the candidate representation should result in a small change of the represented shape. Otherwise, the evolutionary algorithm would approach a random search algorithm.

There is no known research exploring freeform multi-material representations for use in evolutionary algorithms. Indeed, in most structural optimization problems solved with genetic algorithms, a design is pre-conceived and

simply parametrically optimized by the algorithm. The methods presented here allow for unconstrained evolution of multi-material shapes and topologies with smooth, rounded, organic shapes.

11.2   Level Set Methods

Here, a level-set class of representations was used to best fulfill these requirements. 3D level set methods create a four-dimensional landscape, which is then thresholded to create a three dimensional solid [161, 193]. In order to achieve smooth, rounded objects, this 4D landscape must be smoothly varying without sharp changes. A convenient analogy is to view the genotype as specifying a 3D density field, to which a threshold is applied. All the volume at a higher density is instantiated as part of the solid, whereas the rest is interpreted as empty space. The iso-surface can also be computed for visualization purposes using the marching cubes algorithm.

The level-set concept is versatile and useful for evolving shapes for several reasons. First, there is complete freedom in the topology of the object. Also, a continuous evolution path between different topologies exists since a phenotypes topology is derived, not prescribed. Moreover, multiple materials can be seamlessly interspersed throughout the volume. A density field for each material is generated. Then the boundary of the volume is determined by thresholding the sum of the density fields of each material at each location. The material with the highest density at each location within the lattice is instantiated at that location. Alternatively, mixtures of materials could be described by blending materials in ratios proportional to their respective density fields.

(a)	(b)
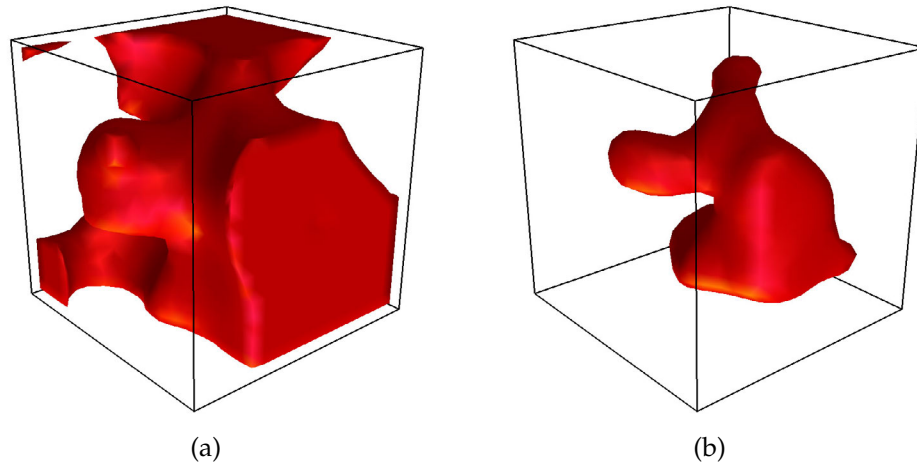
Figure 11.1: Unpadded results of thresholding a randomly varying density field intersect the boundaries of the domain, leading to flat faces (a). Incorporating a soft weighting filter to the edges of the domain results in a smooth, fully contained version of the same object (b).

In some problem domains it is desirable to generate objects that are fully contained within the workspace without intersecting the boundary. These boundary intersections lead to flat faces that interrupt the continuous nature of the objects generated. A soft padding was incorporated by artificially reducing the weights of the cells near the edge according to the proximity of each location to the boundary in each axis. The results of the soft boundary padding filter are shown in Figure 11.1.

Here, three independent representations are explored that create 3D density fields: (a) the Gaussian Mixtures representation [138], (b) The Discrete Cosine Transform (DCT) representation, and (c) the Compositional Pattern Producing Network (CPPN) representation [171]. Each of these was chosen to create smooth shapes of multiple materials. Each representation is also open ended in that it has the ability to dynamically increase the complexity of the resulting objects at the expense of the number of evolved parameters.

11.3 Gaussian Mixtures (GMX)

The Gaussian mixtures representation[138] is a very direct way to specify a 3D density field. A list of Gaussian points within the 3D workspace is maintained to represent the 3D density field. Each point has an associated density and Gaussian falloff range. To generate the instantiated 3D density field, the relative density at each location in the workspace is calculated based on the linear sum of contributions from all Gaussian points. An illustration of Gaussian mixtures in two dimensions is shown in Figure 11.2. A single point (A) results in a circular section. Two points (B), one positive and one negative, creates a non-uniform shape. Twenty randomly generated points (C) generate interesting smooth 2D shapes. For the results presented here, this same concept was used in 3D. Here only uniform 3D falloff for the Gaussian points was used, although representing each distribution using a covariance matrix would lead to more complex shapes with more free parameters per point.

In order to extend the Gaussian mixtures representation to multiple materials, a material index for each Gaussian point is maintained. For each possible material in the simulation, the corresponding Gaussian points are rendered to a separate density field. Then the final heterogeneous object is determined by the relative density of each material at each location. The total density of all materials is thresholded to obtain the shape. Within the shape, the material with the highest density at each location is then instantiated.

Mutating the Gaussian Mixtures representation involves making small changes to the location, density, and falloff (radius) of a Gaussian point, and occasionally adding or removing points. Crossing over two individuals is accomplished by

| (a) 1 Gaussian point | (b) 2 Gaussian points | (c) 20 Gaussian points |
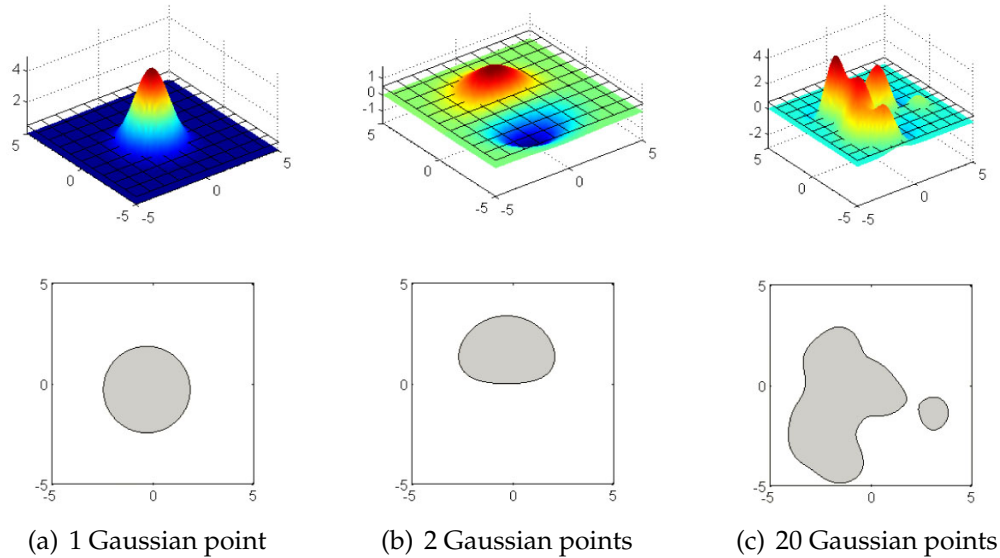
Figure 11.2: The Gaussian Mixtures level set method is illustrated here in two dimensions. A single Gaussian density point thresholded at an arbitrary level (the grid plane) yields a circular region. Adding a second point, here with negative density, results in a non-uniform shape. Twenty points yields complex, smooth freeform shapes. This concept easily extends to three dimensions.

initializing a random plane that intersects the volume of the workspace. Points from one side of the plane are taken from one parent, while points from the other side of the plane are taken from the second parent.

For each new design automation problem, a workspace size is defined along with a single user-defined parameter to determine the appropriate minimum feature size. This parameter corresponds to the minimum falloff standard deviation size of the Gaussian points. When initializing a new population with random values, each new Gaussian point is initialized with a standard deviation between this value and half of the workspace size. Smaller radius Gaussian points are more likely to be generated than larger ones in proportion to their surface area. Each point is also initialized at a uniformly random position within

204
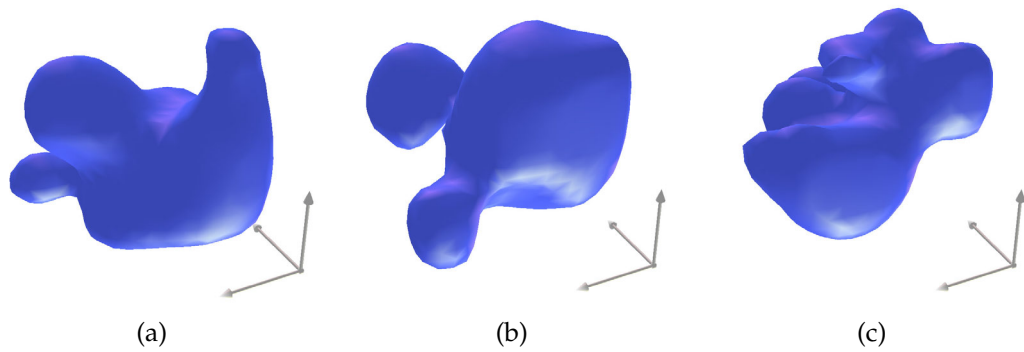
Figure 11.3: Randomly generated single material shapes using the Gaussian mixtures illustrate the propensity to naturally create smooth, rounded, interesting objects.

the workspace of a random material index. The number of points to be generated is determined by the volume of the workspace divided by the square of the specified minimum Gaussian point standard deviation. This results in an appropriate number of points that create interesting shapes but are not massively redundant. Several randomly generated Gaussian mixtures shapes are shown in Figure 11.3.

11.4 Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is a pseudo-generative encoding which naturally generates smooth, freeform 2D and 3D shapes at an arbitrary level of complexity with a minimal number of parameters. The genotype consists of a series of frequency amplitudes at harmonic multiples, and the phenotype is generated by applying the inverse discrete Fourier transform to this series. This representation was explored for its relative simplicity of implementation and evolvability. By representing a three-dimensional object as a 3D matrix of

frequency amplitude components, a number of advantages are realized. These include the ability to render the genotype at any resolution, complete dimensional independence, and naturally available symmetry.

There are a number of possible variations of the inverse discrete Fourier transform that could be utilized. Each implies a different set of boundary conditions. Each boundary within the domain can be even or odd at the left and right boundaries and symmetric about either its endpoint, or a point halfway to the adjacent point. This generates a total of 16 possibilities. Here symmetric endpoint boundary conditions are assumed, which is commonly referred to as simply the discrete cosine transform (DCT).

Multidimensional Fourier transforms such as the DCT also have the favorable property of linear independence. For instance, a 2D spatial object is constructed by applying the inverse DCT independently to each row, then to each column of a 2D matrix of frequency components. The result will be the same regardless of order. This extends into the 3rd dimension as well. The discrete cosine transform also compresses the most meaningful information about an arbitrary geometry into the lowest order frequency coefficients. For this reason, it is has been widely utilized in jpeg image compression. In 3D, this enables complex freeform (smooth) geometries to be represented with relatively few low-order coefficients.

Here the phenotype is a 3D matrix of frequency amplitudes, ranging from -1 to 1. To convert each phenotype to a genotype, the inverse DCT is applied to each row of each dimension of this matrix, converting from the frequency domain to the spatial domain. Thus, each element in the frequency matrix scales a harmonic density field, where the number of modes in each of the three dimen-
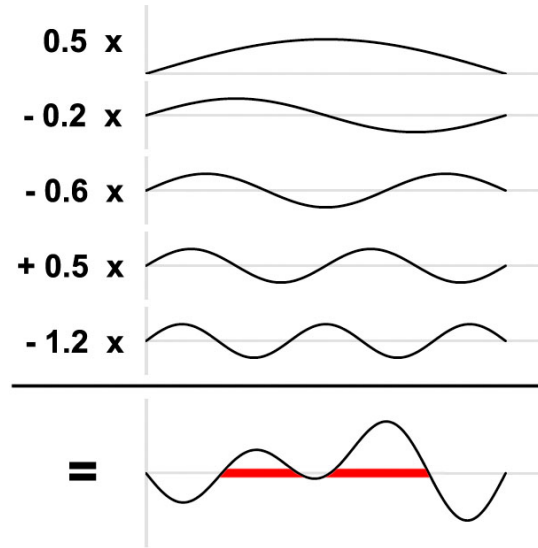
Figure 11.4: The inverse discrete Fourier transform representation sums weighted sinusoids, then thresholds them at zero as shown in this 1D example. The weights are the only evolved parameters.

sions corresponds to its X, Y, and Z indices in the frequency matrix. A simple 1D example is shown in Figure 11.4.

The first element of the DCT representation matrix scales the fundamental harmonic, the second element scales the second harmonic, and so on. These weighted harmonic functions are then summed to create a density field, which is thresholded at zero. In the 1D case, this results in a "freeform" 1D line segment, as shown in red in Figure 11.4. By extension, a 3D matrix of frequency components results in a freeform 3D solid.

The usefulness of the DCT representation for creating smooth, amorphous shapes is realized when the evolved frequency amplitude matrix is smaller than the rendered matrix of voxels in the spatial domain. Before the inverse DCT is applied, the frequency amplitude matrix of the genotype is simply padded with zeros to match the dimension of the number of voxels in the phenotype.

Thus, smooth freeform shapes are naturally created. The minimum feature size relative to the workspace is defined by size of the frequency matrix relative to the spatial matrix.

When initializing a DCT phenotype, the dimensions of the geometrical envelope (workspace) are first defined. A single complexity metric is defined to control the trade off between level of detail and number of values in the genotype. This metric is interpreted as the number of frequency components in the longest dimension of the workspace. For example, if the complexity metric is chosen as 10, the longest dimension of the workspace will have 10 frequency components. If the workspace is not cubic, the number of frequency components in the narrower dimensions is reduced to maintain approximately uniform features at the smallest scale. Otherwise, features would naturally be elongated in proportion to the aspect ratio of the workspace.

For example, a $40 \times 12 \times 12$ beam workspace would result in scaling the number of frequency components by the dimensions of the beam. For a complexity metric of 10, the resulting frequency matrix will be $10 \times 3 \times 3$ elements. If symmetry is enforced, the number of components in each respective dimension should be reduced by another factor of two. In this case, a symmetry along the Y plane would result in only 1.5 (rounded up to 2) frequency components in this direction. Thus a complexity metric of 10 for this example results in evolving only 60 values ($10 \times 2 \times 3$ frequency components) to represent a 5760 voxel structure, which makes this problem computationally feasible. The trade off is that sharp edges and flat surfaces are harder to evolve. Because the size ratio between the physical and frequency matrices is 4:1, features smaller than approximately 4 voxels are also more difficult to represent, and thus less likely to result.

When starting off, a 3D matrix of frequency components is randomly initialized for each material present in the current experiment. These values range from -1.0 to 1.0 in dimensionless units. The range is arbitrary, and can be linearly scaled without effect because the threshold is set to zero. If all components of the frequency matrix were randomly varying within the same interval, the higher order terms would dominate, and a randomly generated object would essentially just be high frequency 3D noise. Thus, before rendering the phenotype, the higher order coefficients of the frequency matrices are reduced. This amount of reduction corresponds to the square root of the 3D manhattan distance from the constant coefficient (zeroth order frequency) to maintain a sensible overall structure with suitably complex topology.

To rendering the phenotype to a genotype, each frequency matrix is padded with zeros to equal the dimension of voxels in the geometrical envelope. The inverse DCT is applied sequentially for each material in each dimension. This results in a 3D density field for every material. Each voxel within the geometrical envelope is instantiated as the material with the highest density at that location.

When evolving freeform amorphous morphologies using the DCT representation, mutation involves making small changes (up to 5%) in amplitude of these frequency components. However, one disadvantage of the DCT representation is that each small change to the phenotype results in a small change throughout the entire rendered structure. The means that isolated local changes to a certain part of the evolved shape require coordinated changes in the frequency components, which is undesirable. Several randomly generated DCT shapes are shown in Figure 11.5.
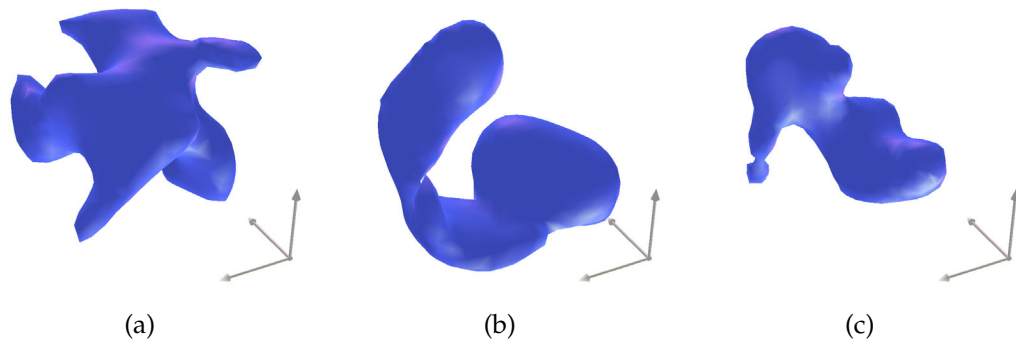
Figure 11.5: Randomly generated shapes using the discrete cosine transform representation illustrate the freeform, smooth shapes it naturally creates.

11.5 Compositional Pattern Producing Network (CPPN)

Compositional Pattern Producing Networks (CPPNs) [171] have been demonstrated to be useful for evolving two dimensional density fields (often interpreted as grey-scale images). Here, the third dimension is included to produce 3D density fields to threshold into amorphous morphologies. A CPPN is similar in concept to an artificial neural network (ANN), except that more geometrically-useful transfer functions are used instead of just sigmoids. A network of nodes (each containing a function) are connected by weighted paths. In order to create 3D amorphous morphologies, three coordinates (X, Y, and Z) that represent the position of a point in 3D space are used as inputs. The network has a single output, which represents the resulting density at that point. By iterating through X, Y, and Z, the full 3D density field is obtained.

Unlike ANNs, a variety of activation functions are used in a CPPN. Activation functions used here include traditional sigmoids and gaussians, as well as sinusoids and the absolute value function for inducing repetition and symmetry, respectively. For each node (function), several parameters were evolved.
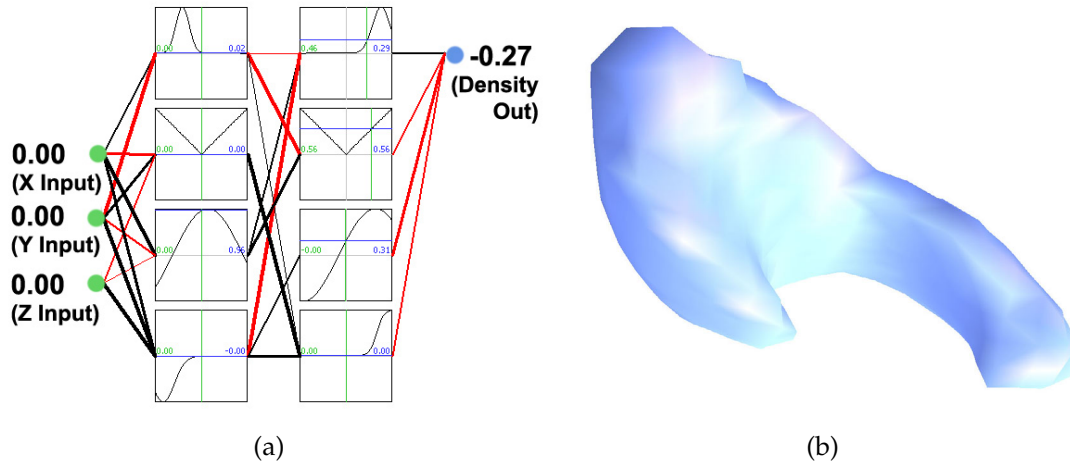
Figure 11.6: The Compositional Pattern Producing Network (CPPN) representation evolves a network of functions with three inputs (X, Y, and Z) and one output, which is the density at that location. The node functions and connecting weights (negative shown red, positive shown black) are evolved. After sweeping the inputs and thresholding, the network (a) produces a 3D freeform shape (b).

These include the function type, offsets, and scaling. Additionally, a complexity measure was implemented to control minimum feature sizes, such that features were not being lost at a sub-voxel scale. Weights between nodes were also subject to evolution. An example CPPN and the resulting geometry is shown in Figure 11.6.

A CPPN has many evolvable parameters. Given a network that is m layers deep with n nodes per layer, there are a total of $m \times n$ nodes. Each node has an assigned activation function and three parameters that describe it (offset and scaling along the X axis, and scaling along the Y axis). There are also as many as $(m - 1) \times n^2$ real-value weighted connections between the nodes, which can be either active or inactive. All these parameters are eligible for small changes upon mutation. However, the mutation rate is chosen such that on average only one of these values (in total) is adjusted. In the crossover operation, a
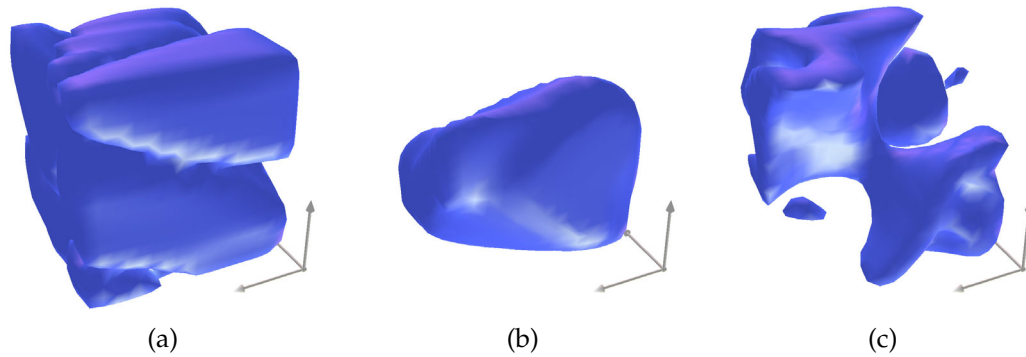
Figure 11.7: Randomly generated shapes using the CPPN representation illustrate the types of shapes it naturally creates.

rectangular region of nodes is selected from one parent, and the rest of the nodes are taken from the second parent. This region is chosen such that all nodes are equally likely to be in the selected region, not favoring the center nodes.

Here, fixed size networks were used without dynamic growth, which may have hampered the performance of the CPPN methods. The user defined the number of layers and the number of nodes per layer. Typically, two layers of 10 nodes each produced relatively interesting 3D shapes. A separate network was created for each material to be used in the simulation. Several randomly generated CPPN shapes are shown in Figure 11.5 for a single material.

CHAPTER 12

EVOLVING STATIC STRUCTURES AND MECHANISMS

12.1   Background and Motivation

Multi-material 3D-printing technologies including digital fabrication permit the freeform fabrication of complex spatial arrangements of materials in arbitrary geometries. This technology has opened the door to a large mechanical design space with many novel yet non-intuitive possibilities. This space is not easily searched using conventional topological optimization methods such as homogenization [7]. Here an evolutionary design process for static three-dimensional multi-material structures is proposed that explores this design space and designs substructures tailored for custom functionalities. The use of this algorithm is demonstrated for the design of 3D shapes, 3D non-uniform beams and 3D compliant actuators.

In the engineering world, the method of fabrication dictates the mode of design. Traditional machine tools naturally create single material parts with straight lines, flat surfaces, and circular profiles. Thus, the current engineering design paradigm is feature-based modeling, in which two dimensional sketches are extruded, revolved, swept, and patterned to create three-dimensional parts of a single, homogeneous material. With the advent of manufacturing techniques which easily create freeform (smooth) geometry and arbitrary internal material distributions, the constraints of the current computer aided design (CAD) paradigm become crippling [62].

Despite the advent of powerful computing, human input is currently still
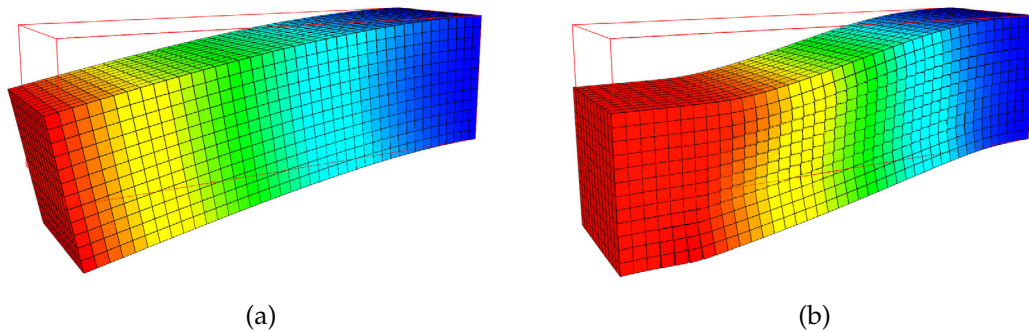
Figure 12.1: The direct stiffness method was used for physical evaluation of multi-material cantilever beams, here showing homogeneous (a) and evolved (b) deformations

crucial to transform a set of abstract functional goals into an instantiated design. A major challenge in automating the design process lies in the development of robust algorithms to address the spatial and physical constraints of a traditional feature-based manufacturing process, such as machining [118]. However, the advent of efficient freeform fabrication methods such as digital manufacturing allows any 3D shape and topology to be fabricated without penalty, removing many complex constraints from the automated design process.

The use of evolutionary algorithms is demonstrated here to design the three dimensional material distribution of a shapes and cantilever beams to meet high-level functional goals not easily accomplished using current CAD tools. These goals include interdependent competing objectives that cannot be directly solved for. The algorithm makes use of the linear direct stiffness to evaluate the physical performance of the shapes (Figure 12.1). The solutions presented here are suitable for freeform fabrication.

214

12.1.1 Relevant Freeform Fabrication Technologies

There are many fabrication technologies capable of creating smooth, freeform objects. Subtractive processes such as computer numerical controlled (CNC) milling selectively remove material from a homogeneous block. Depending on the complexity of the machining system, overhangs and concave features may be possible but internal cavities and complicated topologies often require prohibitive tooling and complexity. In contrast, additive manufacturing processes selectively build up a freeform object layer by layer, placing temporary supports where needed to create any geometry and topology. A variety of available commercial systems are capable of fabricating parts out of materials ranging from photo resins to stainless steel [5]. However, additive fabrication technology is largely limited to creating parts of a single, homogeneous material.

The recent advent of multi-material freeform fabrication machines promises to exponentially expand the design-space for physical 3D parts. Currently, technology exists commercially that allows 3D objects to be seamlessly fabricated from two or more materials ranging from soft rubber to rigid polymers [117, 135]. Digital manufacturing techniques can easily place voxels of soft and hard material in any spatial combination. These technologies allow complete freedom over the internal distribution of materials, including the ability to create repeating microstructures. However, CAD tools which can utilize these unique capabilities are lagging behind the fabrication technology itself.

12.1.2 Topology Optimization

The ideal design process would autonomously create an optimal blueprint purely from the functional goals of the desired part and the available materials. Much work has been done in this field [39, 51] regarding the topology or shape optimization of single material structures, with the goal of maximizing stiffness per weight. The most established method in this field is homogenization, as originally demonstrated by Bendsoe and Kikuchi [7]. This iterative process varies the effective stiffness of each cell within a 2D or 3D matrix according to its strain energy, and optimizes the structure subject to constraints on total volume and minimizing strain energy. Variations on this method have yielded results that maximize deflection for applications such as a simple gripper structure, or even utilize two materials to emulate an actuated structure [17, 133, 164].

However, homogenization results are generally limited to optimizing overall deflection or force. As complexity is added to the design problem, such as with competing objectives involving multiple materials or specifying a desired deformed shape, homogenization approaches become unwieldy or even completely impractical.

Despite their generality, evolutionary algorithms have not found universal use in topological optimization for several reasons. First, the homogenization method greatly outperforms evolutionary algorithms for the single objective structural optimization problems that are often addressed in literature. Secondly, the success of evolutionary algorithms depends strongly on the encoding (genotype) used to represent the physical object (phenotype). In early attempts at using evolutionary algorithms to solve topological optimization problems, every individual pixel or voxel was represented explicitly in the genotype
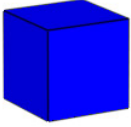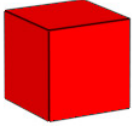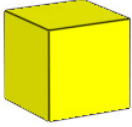
[83, 91] as a bit stream. In addition to the challenge of making crossover and mutation non-destructive, this method scales poorly to large structures.

12.2 Evolutionary Algorithm Specifics

Several cases were chosen to illustrate the use of evolutionary algorithms in designing structures with high-level functionality. First, single material structures were evolved to compare the results of the genetic algorithm directly to homogenization methods. Next, the deflected shape of the top surface of a cantilever beam comprising stiff and flexible materials was considered. These results were verified in experiment using a multi-material 3D printer to fabricate the resulting material distributions. A third, lighter material was then introduced along with an additional objective of minimizing structural weight.

Lastly, an actuator that maximizes the two-dimensional deflection of a beam tip subject to applied forces in the mutually orthogonal direction to the desired deflection was considered. The two material single objective configuration and the three material multi-objective configurations were both considered in this case as well. The physical parameters of the materials used in all these experiments are outlined in Table 12.1. These material properties are similar to those which can be printed with existing multi-material freeform fabrication processes.

217

Table 12.1: Summary of experimental materials for static beam deflection experiments

Identifier	Stiff	Flexible	Lightweight
Color	Blue	Red	Yellow
Element Shape			
Stiffness	15 MPa*	0.15 MPa	1.5 MPa
Poisson's ratio	0.3	0.3	0.3
Density	1.2 g/cm^3	1.2 g/cm^3	0.12 g/cm^3

*Several experiments use 150MPa, as noted in text

12.2.1 Algorithm Parameters

The evolutionary algorithm used here is outlined in Figure 12.2. Table 12.2 summarizes the details of the evolutionary algorithm. Of the representations outlined in the previous chapter, the discrete cosine transform (DCT) method was used here. The method of selecting which individuals of the population to crossover has implications regarding the necessary population size and diversity of that population. When two individuals are selected for crossover, each frequency component of the offspring is randomly chosen from either parent. The parent which contributed the greatest number of frequency components is stored as the "most similar" parent for the selection process.

To mutate an individual, random variations were introduced into the frequency amplitudes of the genotype. These variations were kept small, such that any value could not change more than 10% for any given mutation. No bounds were placed on the range of values, allowing the algorithm to essentially re-scale

Table 12.2: Summary of evolutionary algorithm setup for static beam deflection experiments

Population Size	25-50
Selection method	Deterministic crowding
P(Mutation)	0.2
Solution Encoding	DCT generative

all values instead of saturating at arbitrary limits. In the DCT representation, low frequency components are weighted higher than the high frequency components. The mutation process preserves this general scaling, while allowing complete freedom for beneficial high frequency components to become more significant. For all experiments presented here, deterministic crowding selection was used with a population of 25-50 individuals and 20% probability of mutation. In this method, each individual was paired with a random mate and crossover was performed. If the child was more fit than its most similar parent, the child replaces the parent in the population. Otherwise, the offspring is disregarded.

12.2.2 Leveraging Resolution Independent Representations

As in most evolutionary algorithms, the evaluation step is the most computationally expensive part of the algorithm. Reasonable effort was made to optimize the code. It was written in C++ for low computational overhead, and the optimized PARDISO solver library was used for the heavy matrix calculations. This combination made the evolution of $40 \times 12 \times 12$ voxel structures with approximately 35,000 degrees of freedom computationally feasible on a single

1. Initialize

Frequency matrix initialized with random values

(Individual X)

$$\begin{bmatrix} -7 & 1 & 0 \\ 5 & -9 & -3 \\ -4 & -1 & 8 \end{bmatrix}$$

2. Render

(Individual X)

$$\begin{bmatrix} -7 & 1 & 0 \\ 5 & -9 & -3 \\ -4 & -1 & 8 \end{bmatrix}$$

Inverse DCT

Inverse discrete cosine transform performed on each frequency matrix to generate geometry

3. Evaluate

Each object is analyzed using direct stiffness method to determine fitness

4. Crossover

The best individuals are combined to create a new generation

(Individual X)

$$\begin{bmatrix} -7 & 1 & 0 \\ 5 & -9 & -3 \\ -4 & -1 & 8 \end{bmatrix}$$

(Individual Y)

$$\begin{bmatrix} 5 & -4 & 9 \\ -1 & 1 & -7 \\ 2 & 6 & -5 \end{bmatrix}$$

(Individual Z)

$$\begin{bmatrix} 5 & 1 & 0 \\ -1 & -9 & -3 \\ 2 & 6 & 8 \end{bmatrix}$$

5. Mutate

Small random variations are made to the frequency matrix of each individual

(Individual Z)

$$\begin{bmatrix} 4 & 1 & 0 \\ -1 & -9 & -5 \\ 2 & 6 & 8 \end{bmatrix}$$

Repeats for many generations

6. Optimized

The best individual after many generations is output

Figure 12.2: Flowchart of the multi-material evolutionary algorithm used for design automation. The discrete cosine transform encoding allows freeform objects to be encoded with a minimal number of parameters.

Average Fitness: Slope Discontinuity Experiment

Resolution doubles in evaluation
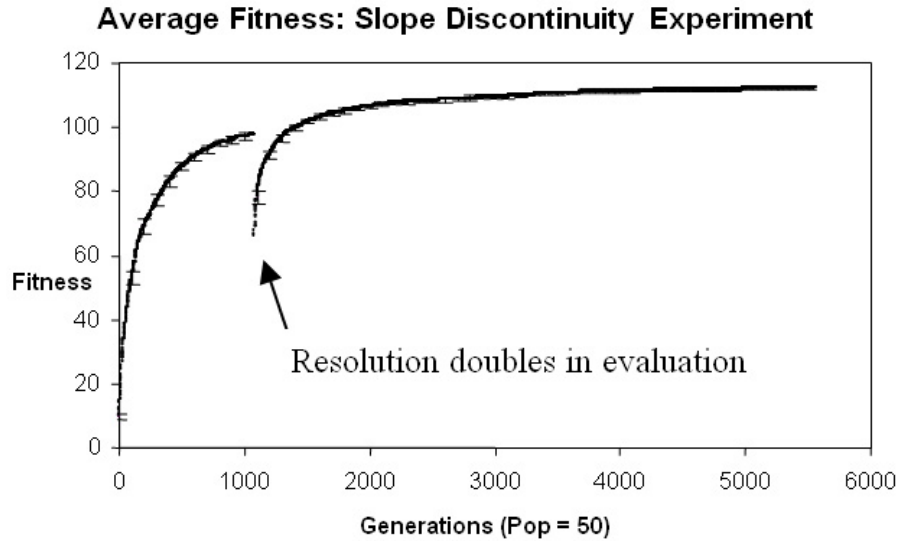
Figure 12.3: A representative fitness plot illustrates the effect of evolving with a low resolution phenotype before switching to a higher resolution population. The temporary drop in fitness upon switching resolution is quickly recovered.

desktop computer. Each evaluation takes less than four seconds on a single processor, dual core desktop computer at 3.2 GHz.

The resolution independence of the DCT encoding was also exploited to speed up the evolution process. First, an entire population was evolved using a domain of one half the dimensions (12.5% of the total voxels) by simply sampling the inverse Fourier transform at half the density. These evaluate in a fraction of a second, resulting in an order of magnitude increase in speed over the full resolution evaluation. Once fitness begins to level off (less than 10% increase in the last half of the evaluations), the evaluation is switch to full resolution. (Figure 12.3) Fitness is usually reduced directly after the transition, but quickly recovers and continues to progress.

221

<div style="text-align:center">(a)   (b)</div>
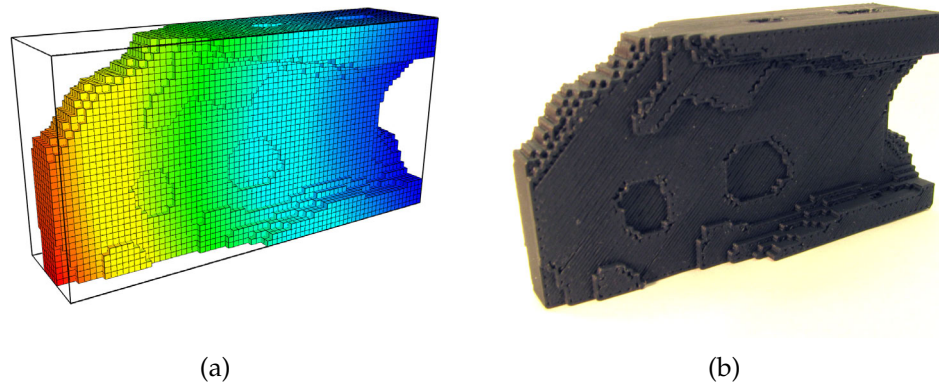
Figure 12.4: A cantilever beam was autonomously designed to maximize stiffness and minimize weight (a) using the proposed algorithm. The physical printed beam is shown in (b)

12.3 Freeform Shape Optimization

Given any set of loading and constraint conditions, the algorithm is successful in creating and optimizing a stiffest shape relative to the desired proportion of voxels present. The case of a cantilever beam is shown in Figure 12.4. Here, the population size was set at 25 and the domain used while evolving was $24 \times 6 \times 12$ voxels. The algorithm was allowed to run for approximately 3500 generations, although the gains past 1200 generations were negligible. In the images shown, the resulting genome was rendered at twice the resolution as it was evolved at, demonstrating further use of the resolution independence of the DCT representation.

A variety of other boundary conditions were applied to various sizes of domains to generate solutions with different static mechanical functionality (Figure 12.5). In some cases a negative region of density was enforced, such as with the crane example (Figure 12.5(b)) underneath the location of a downward applied force. In all cases a total desired percentage of material was specified

(typically 30-40%), which was balanced with minimizing the maximum displacement of the forced region(s) of voxels. The solutions were allowed to run overnight or until suitably interesting results were obtained.

12.4 Deflected Beam Profile Optimization

12.4.1 Two Material

As a further exploration into the use of this genetic algorithm for design automation, it was applied to a problem that cannot be easily solved using homogenization methods. Given a cantilever beam, a variety of desired deflected shapes were selected for the top surface of the beam. These profiles departed in varying degrees from the normal third-order polynomial profile. In order to observe interesting behavior, two materials of varying stiffness were considered in a domain of $40 \times 12 \times 12$mm. Each voxel was a 1mm cube, and symmetry along the vertical plane parallel to the major axis of the beam was enforced.

A total of four different beam profile shapes were autonomously designed and physically tested in addition to verifying the deflected shape of the homogeneous beams. In approximate increasing order of difficulty, these include a straight profile, a circular arc shaped profile, a discontinuous slope profile, and a fourth order polynomial profile that incorporated negative (upward) curvature.

In order to facilitate the emergence of the correct qualitative shapes, differences in quantitative deflection of competing designs were not penalized. In
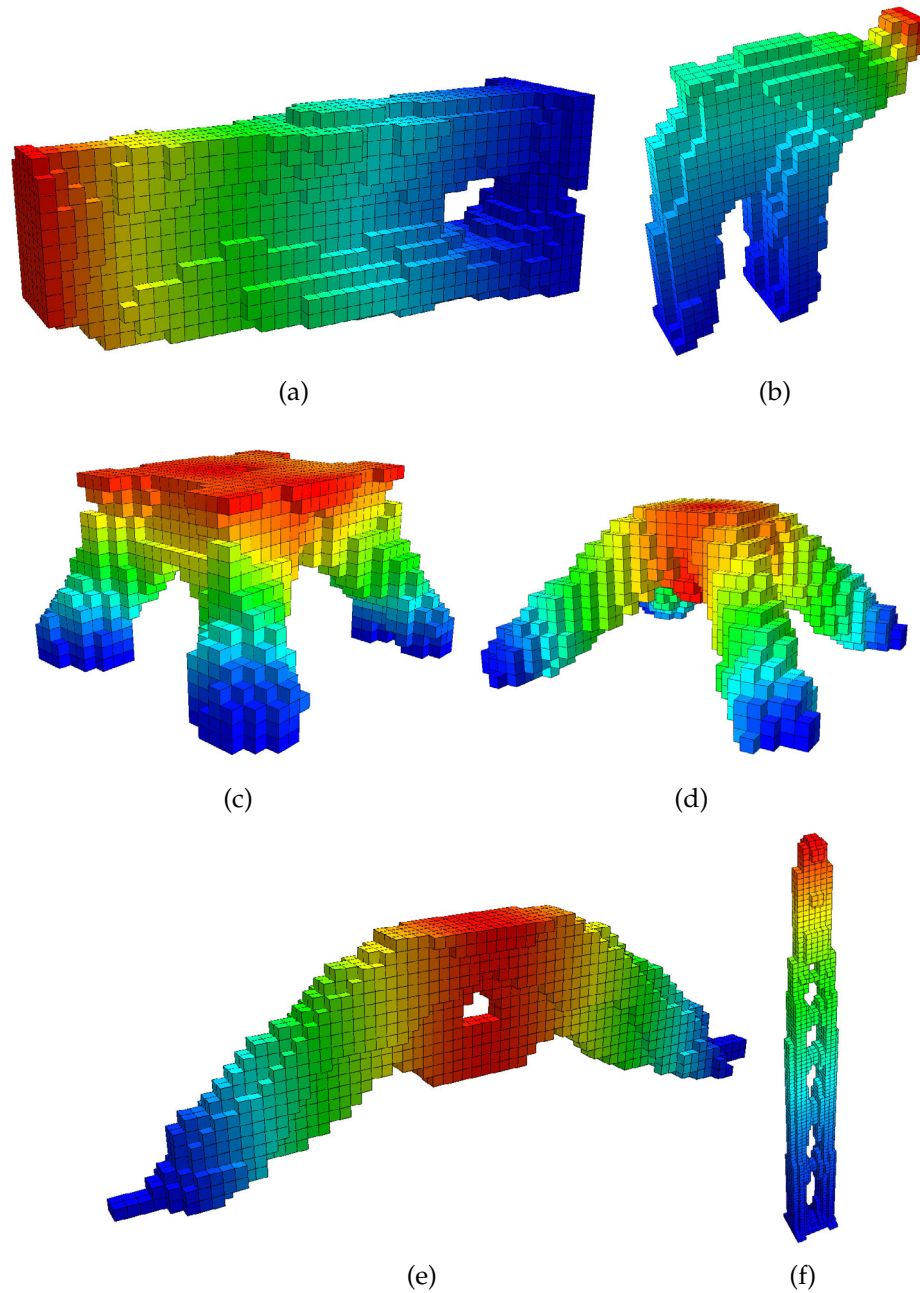
Figure 12.5: Examples of optimized 3D single material shapes to meet a variety of tasks are shown. Such tasks include a cantilever beam (a), a crane with enforce negative region (b), a table (c), a table support (d), a bridge (e) and a tower to resist side loading (f).

Table 12.3: Number of generations and evaluations for static two material beam design automation results.

Desired Shape	# Generations	# Evaluations
Straight	22900	1.1×10^6
Circular Arc	10700	5.3×10^5
Discontinuous Slope	5600	2.8×10^5
4th Order Polynomial	4700	2.4×10^5

order for the design algorithm to evaluate each design, the beam was loaded and simulated to estimate the deflected shape. Both desired and simulated profiles were normalized so that the free end had unit deflection. Then points along top surface of each deflected beam were recorded and compared to the desired profile. The workspace of the beam was selected to be $12 \times 12 \times 40$ (5760 total) voxels. All solutions were allowed to progress for at least 200,000 evaluations (Table 12.3).

Physical Verification

In order to verify the performance of the design automation in the physical world, each beam was autonomously fabricated in 3D using an Objet Connex 500 printer. First, a homogeneous beam was printed and tested to verify the experimental procedure (12.6) with favorable results. Because the shapes were only being compared qualitatively, the absolute stiffness of the physical materials did not have to be exact. However, the ratio of stiffnesses of the two materials was important to replicate the designed shapes in the physical world. A combination of Objet's VeroGrey (hard) and TangoPlus (soft) were used. Before printing, additional geometry was appended to each shape to facilitate physical
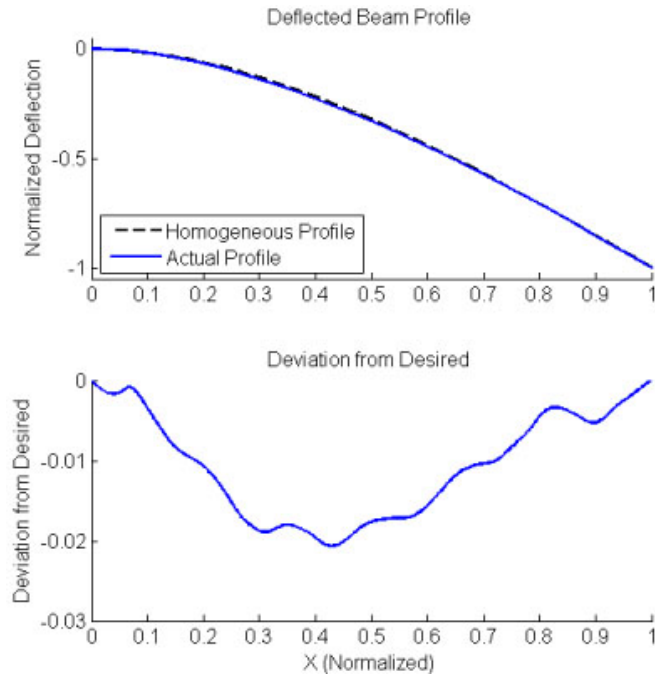
Figure 12.6: A test beam of homogeneous material distribution was fabricated to verify the experimental setup. The top plot shows the shape of the beam fixed on the left and pulled down on the right. The deviation (homogeneous - actual) is shown in the bottom plot. RMS error in deviation from the expected profile about 1.25%.

testing. This included a base block to be clamped to ground and loops to enable force to be evenly applied at the free end.

Each design was printed out and tested. In order to compare the actual resulting physical beams with the expected results, image analysis was used to record the deflected profile of each beam. The test beam was firmly clamped at the fixed end while enough downward force was applied to the free end manually to achieve a non-trivial deflection. An image was then captured of the beam from the side with the lighting and background set up to facilitate accurate edge detection.

The process of extracting the profile of the beam from each image is shown

(a) Original Image

(b) Edge Find

(c) Threshold

(d) Segment
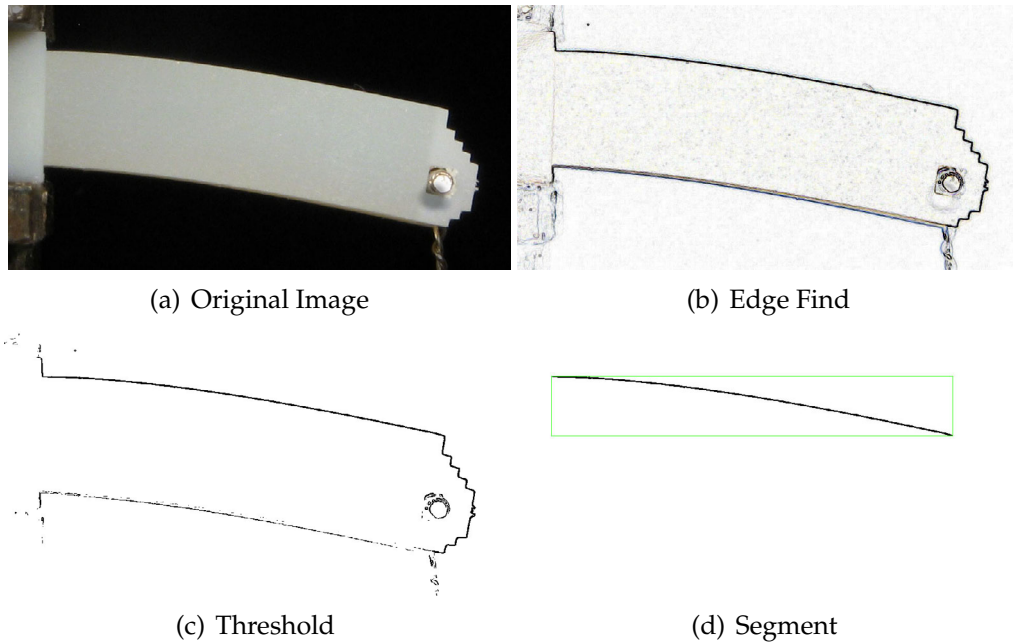
Figure 12.7: To extract the profile of a deflected beam, an image was captured of each beam in its deflected state (a). An edge-finding filter was applied (b) followed by a thresholding filter to create a black and while image (c). Pixels not associated with the top surface of the beam were then manually erased to allow the profile to be detected and segmented with a Matlab script (d).

in Figure 12.7. First, an edge-finding filter was applied in Photoshop. The result was then thresholded into a black and white image. Extraneous image noise not connected to the profile of the beam was eliminated. A Matlab script then determined the average height of the black pixels for each column of the image to create a 1D function. This function was then smoothed using the LOWESS (locally weighted scatter plot smoothing) method with a linear fit over a span of 101 pixels in the X direction. This sufficiently smooths out the effects of image noise while retaining the actual shape of the beam. The resulting function is then normalized and compared to the simulated and desired function and the RMS error between each is calculated.

The design automation algorithm was able to match the simulated shape with a high degree of accuracy to each desired profile. The resulting geometries and deflections are shown in Figures 12.8 and 12.9) for the simulated and tested beams. Also, plots showing the deflected shape, the desired deflected shape, and the actual deflected shape compared to the deflection of a nominal, homogeneous single material beam are shown. The deviation of the simulated and tested beams from the desired profile are also plotted. A summary of the errors is given in table 12.4.

A straight profile was considered as a simple geometry which ideally needs only a single region of each material for the ideal solution. The result (Figure 12.8(a)) shows the expected distribution of flexible material near the grounded edge and stiff material comprising the rest of the beam. Deviation from the nominal profile was very low, at three orders of magnitude less than the dimension of a single voxel. The tested behavior was also in good agreement, with deviations less than 1% maximum or 0.5% RMS error from the specified profile.

A circular (arc) deflected profile was also considered. (Figure 12.8(b)) The solution to this problem is somewhat more complex than for the simple straight line but just as effective, with very low deviation from the desired shape in simulation. In reality, the resulting shape was clearly an arc, but with slightly different curvature that yielded deviations of around 3% maximum or 1.6% RMS from nominal, which is still quite reasonable. This change in curvature could be attributed to a small inconsistency in the proportional stiffness of the hard and soft materials that were physically printed as compared to the values used in simulation.

Two cases of more extreme specified profiles were also considered. In the
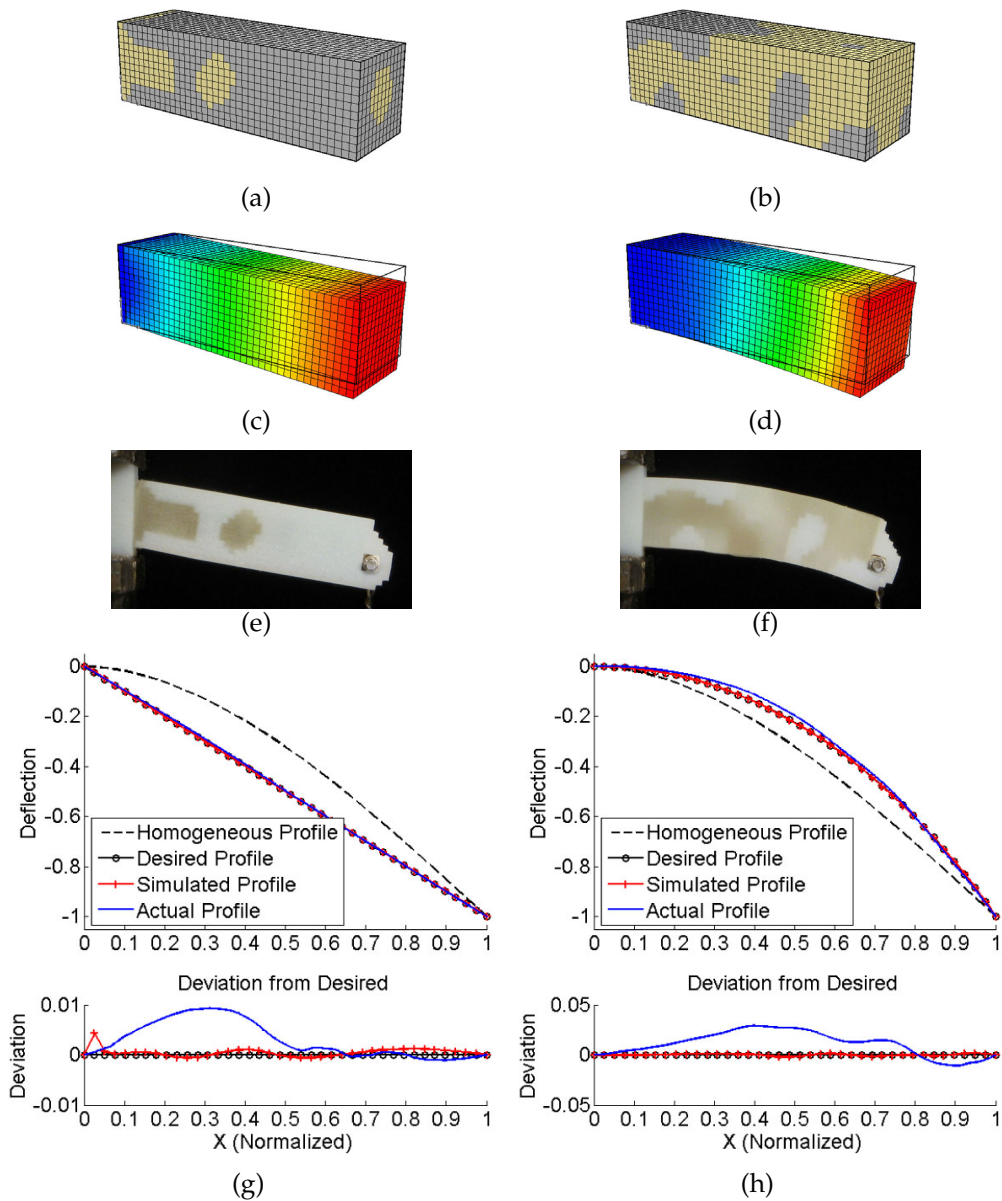
Figure 12.8: Autonomously designed material distributions of static cantilever beams deflect in non-traditional top-surface profiles such as a straight line (left column) and circular arc (right column). The generated material distributions (a-b) and the corresponding simulated deflected profiles (c-d) closely match the input functions. Actual beams were autonomously fabricated and tested (e-f). The results for both simulation and reality are compared with the desired shape and nominal shape (g-h). The gray material is stiff.

Table 12.4: RMS error in deviation of the simulated and actual deflected beams from the desired profile shape.

Desired Shape	RMS Simulation Error	RMS Physical Error
Nominal Homogeneous	-	1.25%
Straight	0.10%	0.47%
Circular Arc	0.10%	1.64%
Discontinuous Slope	0.86%	2.36%
4th Order Polynomial	0.90%	7.62%

first case, a discontinuous slope was applied by combining two linear functions piece-wise (Figure 12.9(a)). The algorithm returned a favorable result in simulation, with the largest deviations occurring right around the discontinuity, as expected. This was still well bounded at two orders of magnitude less than the dimension of a voxel. The actual beam showed similar qualitative error from the nominal profile, but greater worst case absolute error on the order of 8% and RMS error of around 2%.

Lastly, a 4th order polynomial profile involving both positive and negative curvature that matches the fixed zero slope boundary condition of the cantilever beam was specified (Figure 12.9(b)). Even after optimization, there is a small deviation at the free end in simulation, but this is likely because it is physically impossible to perfectly match a zero-slope boundary condition at the end of deflected cantilever beam. A stiffness ratio of 1000:1 between the stiff and flexible materials was used in this case to enable a wider variety of solutions. Deviation in reality was much worse, but still bounded at less than 15% maximum error or approximately 7% RMS error.
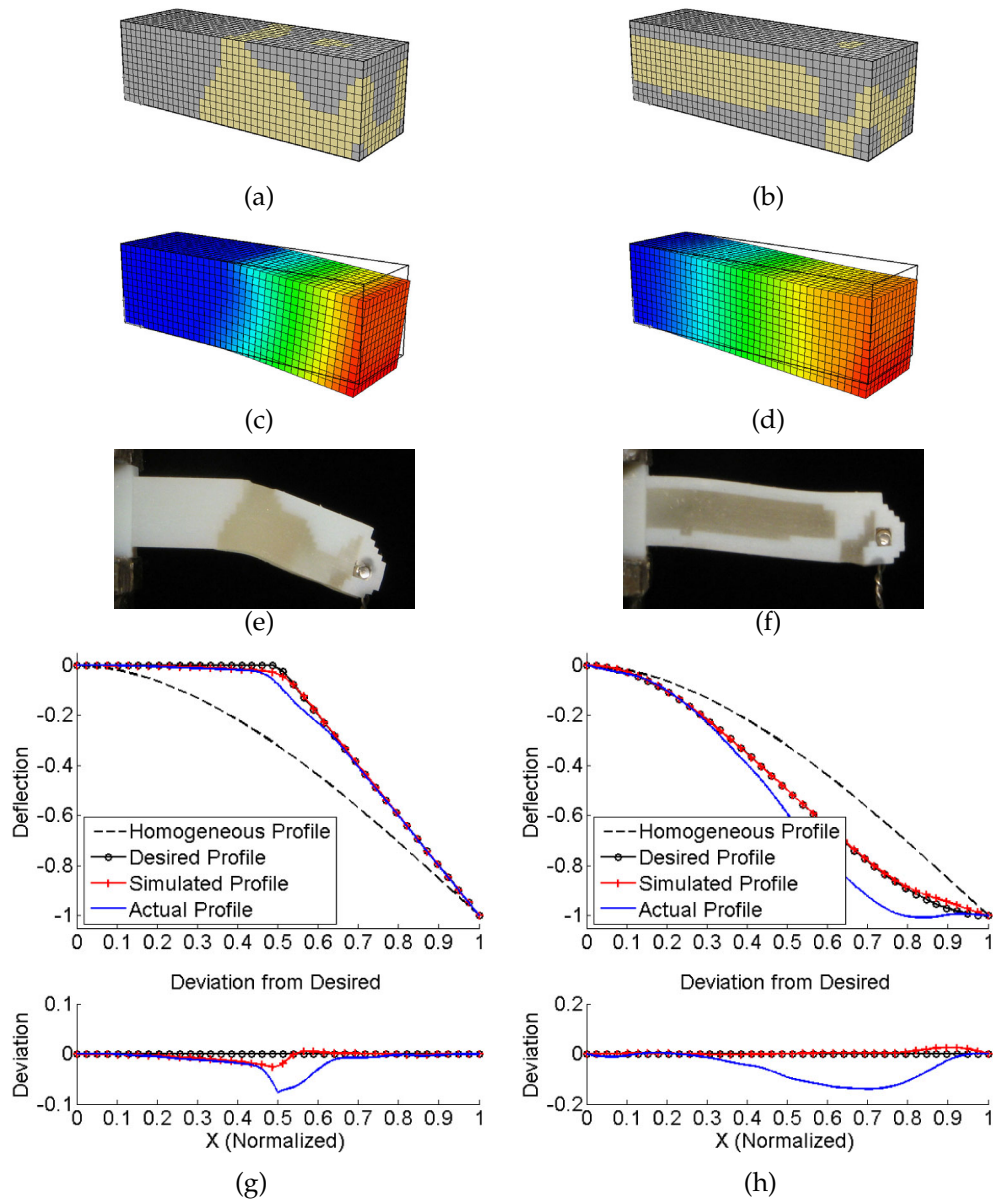
Figure 12.9: Discontinuous slope (left column) and 4th oder polynomial (right column) profiles were also explored. The generated material distributions (a-b) and the corresponding simulated deflected profiles (c-d) closely match the input functions. Actual beams were autonomously fabricated and tested (e-f). The results for both simulation and reality are compared with the desired shape and nominal shape (g-h). The gray material is stiff.

Table 12.5: Summary of algorithm results for three material deflected pro-
file experiments and associated beam deflection errors and
weights.

Profile	Straight	Circular Arc	Slope Discontinuity
RMS Error	0.74%	0.16%	1.36%
Generations	15000	14300	13500
Evaluations	7.5×10^5	7.1×10^5	6.7×10^5
Normalized Weight	0.556	0.447	0.502

12.4.2 Three Material Multi-Objective

An additional criterion of minimizing weight was introduced to the previous
case to add complexity to the optimization problem. The stiff and flexible ma-
terials explored in the previous experiments were preserved, and interpreted to
be of equal density. A third, much lighter material was introduced, with an in-
termediate stiffness. This creates a situation where the ideal solution will very
likely make use of all three materials. The fitness used in the previous experi-
ments was rewarded by up to 50% of its value according to the total mass of the
beam. Results of this multi-objective optimization are shown in Figure 12.10.

Three of the desired profiles from the previous experiments were selected
for multi-objective optimization. The linear deflection case resulted in a slightly
worse fit than the single objective case, but based on the properties of the mate-
rials presented in Table 12.1, has a density of nearly half that of the two material
solution. The circular arc case (Figure 12.10) was slightly worse in matching
the desired profile, and yet was less than half the density of the single objective
case. Finally, in the case of the discontinuous slope, the profile fit metric was
only slightly worse, while the resulting beam was very nearly half the density.
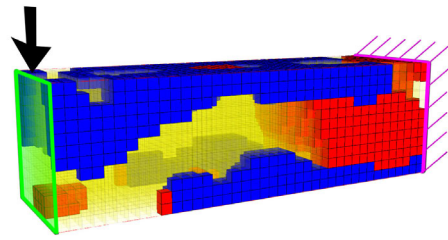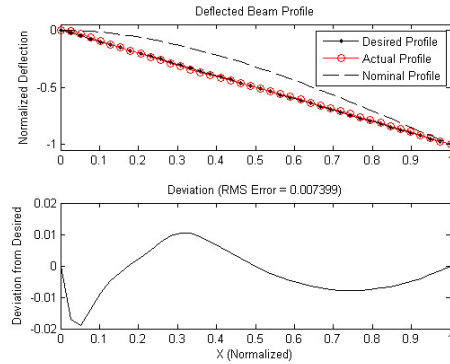
(a)   (b)
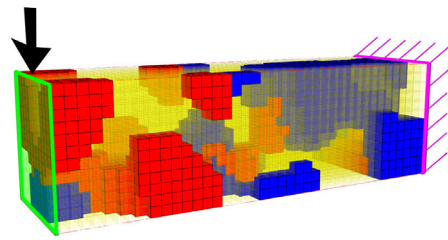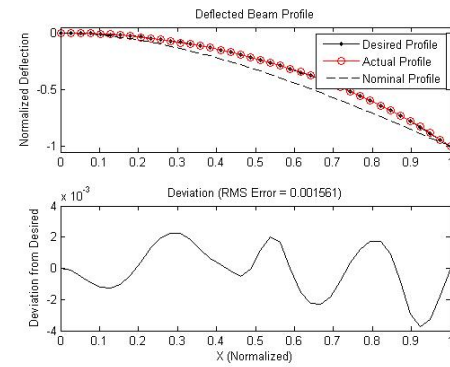
(c)   (d)
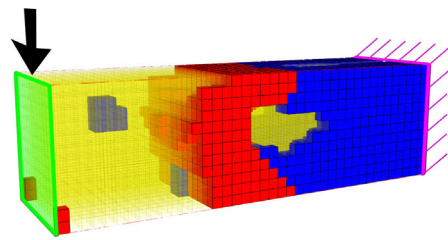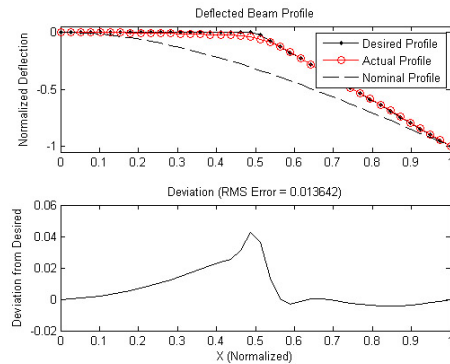
(e)   (f)

Figure 12.10: Results for the deflected shapes of three material beams are similar to the two material results, while additionally optimizing for the lightest structure. The straight profile (a-b), circular arc (c-d), and discontinuous slope (e-f) cases were considered here. The low density material (yellow, shown translucent) has an intermediate stiffness between the stiff blue and flexible red materials

233

12.4.3 Continuous Material

The optimization of multiple, gradated materials presents a more interesting problem which is not addressed in the previous results. In these experiments, a qualitative material model was used to generalize the ability of additive manufacturing processes which simultaneously print high stiffness and low stiffness materials. A two-material system was assumed, where both stiff and flexible materials can be placed and combined in any combination at both the macroscopic and microscopic level. In effect, the material property can vary continuously between the two extremes. The resulting material property of each location is calculated from the density output by the discrete cosine transform, according to a 4th order exponential weighting. The resulting geometries are shown in Figure 12.11, along with plots showing the desired deflected shape overlaid onto the actual, evolved deflected shape.

The algorithm was also successful in meeting these high-level goals in the pseudo-continuous material distribution domain. The straight profile (Figure 12.11 a,b) was again considered as a simple geometry which needs only simple regions of stiff and flexible material. The result shows the expected distribution of flexible material near the grounded edge and stiff material composing the rest of the beam. It further improves by accounting for the zero slope boundary condition imposed by the fixed end of the cantilever beam. A discontinuous slope (sharp bend) was also considered again, with similarly successful results (Figure 12.11 c,d). Also, the difficult 4th order polynomial profile involving both positive and negative curvature was tried (Figure 12.11 e,f). Remarkably, the algorithm found a suitable solution to this problem, which is in a fundamentally different material distribution domain than the other solutions.

(a)

(b)

(c)

(d)

(e)

(f)
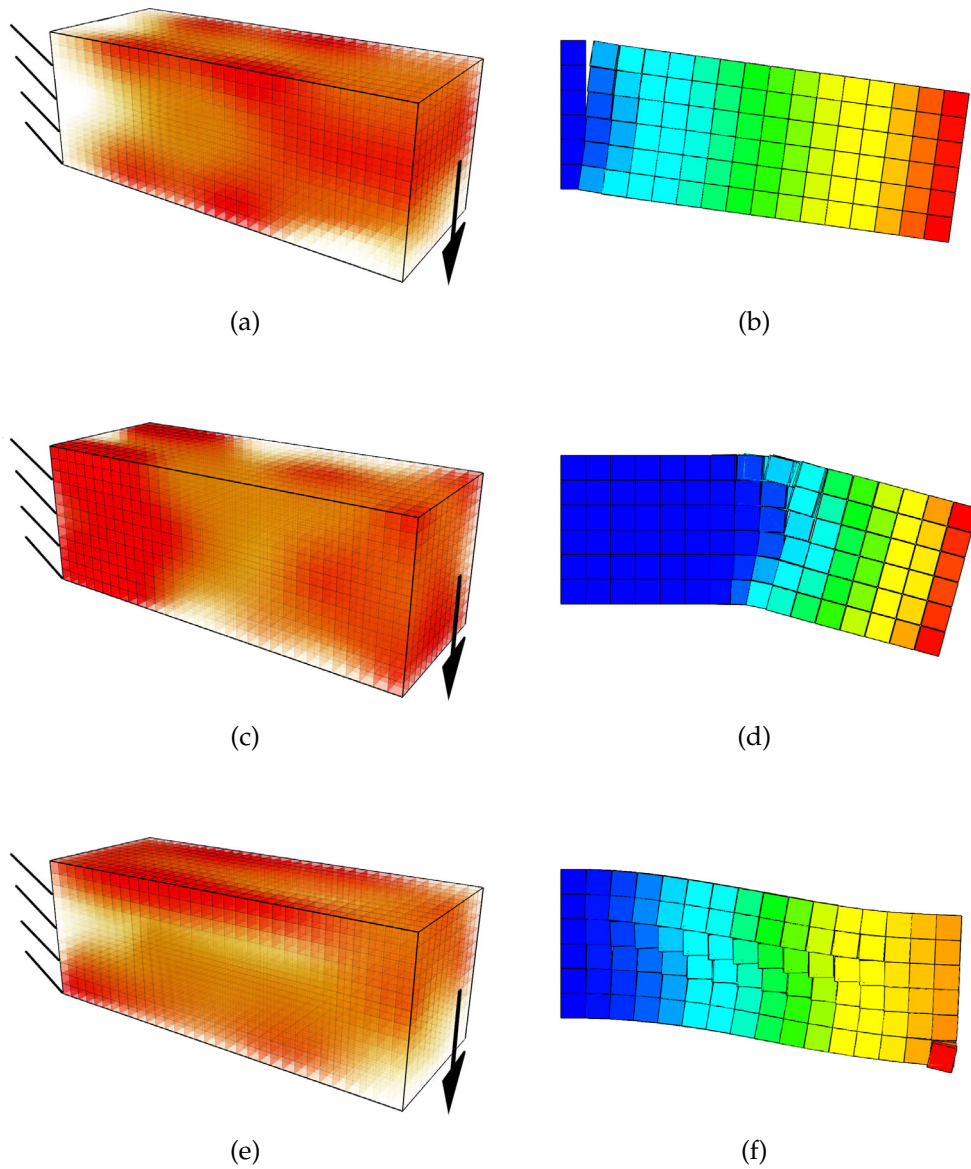
Figure 12.11: Evolved geometries (a,c,e) show stiff regions of material as solid red transitioning to transparent yellow for lower stiffness. The results for the deflected shape of each evolved cantilever beam (b,d,f respectively) demonstrate the ability to control the deflected profile of the cantilever beam to a high degree of accuracy, including slope discontinuities and non-intuitive upward curvatures.

235

12.5 Two Force Actuator Optimization

12.5.1 Discrete Material

The optimization of a planar actuator was also considered. The loading conditions are shown in Figure 12.12. Two input forces are applied midway down the beam parallel to the major axis of the beam. These forces range from -0.1 to 0.1 newtons, and the output is the displacement of the tip of the beam in the two orthogonal directions to the major axis of the beam. Fitness was defined as the reachable area of the center of the tip of the beam. The domain of these problems was constrained to $20 \times 6 \times 6$mm, with 1mm cubic voxels. No symmetry constraints were imposed in order lessen constraints on design space available to potential solutions. In a manner similar to the previous experiments, an additional objective of minimizing weight was also considered subsequently. The resulting geometry and tip tracing areas are shown in Figure 12.13 for the single objective (two material) and multi objective (three material) cases.

In both solutions, the algorithm came up with very similar solutions that make intuitive sense upon examination. A single row of stiff material connects the grounded face of the beam to the mid-plane of the beam where the forces are applied, while the rest of the beam is filled in with the most flexible material. Regions of stiff material (or the lighter, semi-stiff material) connect this strand to the two locations of applied force. The free half of the beam essentially has no forces acting on it. Thus, in the single objective case, this region was effectively ignored, leading to a mostly random material distribution within it. However, the addition of the density objective provided incentive for this unused area to become strictly light material, which accounts for nearly all the difference in the
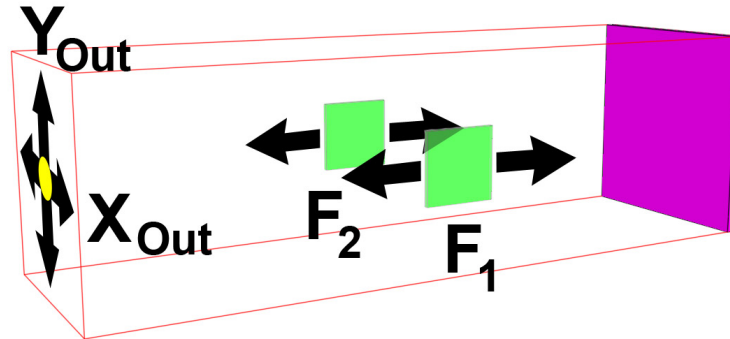
Figure 12.12: The loading conditions for the actuator optimization present a high-level, non-intuitive problem to be solved by the design automation algorithm.

Table 12.6: A summary of the multi-material actuator experiments shows similar absolute performance of single and multi-objective cases. A larger reachable area is better.

Scenario	Single Objective	Multi-Objective
Reachable Area	10.14mm^2	10.08mm^2
Generations	5000	19500
Evaluations	2.5×10^5	9.8×10^5
Normalized Weight	1.0	0.604

density of the three-material beam, which is approximately 60% of the mass of the two-material beam.

The resulting tip-deflection plots for the two are extremely similar, although the two-material solution is approximately 0.6% better. This demonstrates the
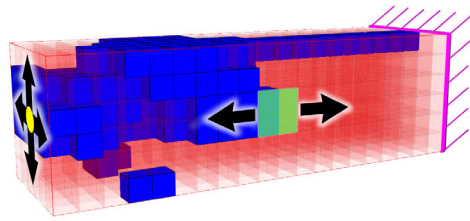
(a)
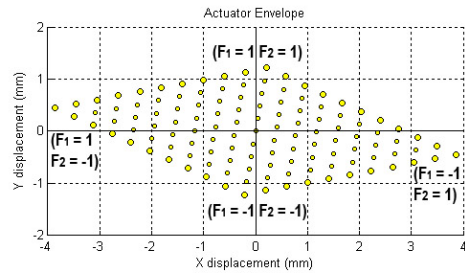
(b)
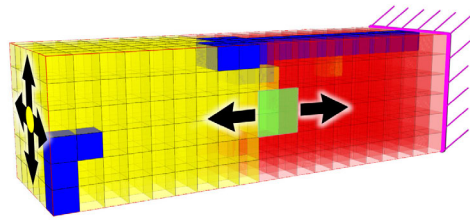
(c)
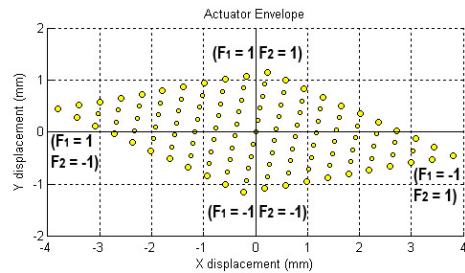
(d)

Figure 12.13: Evolved beam actuators are shown that maximize the potential swept area of the tip of the beam. Both single objective and multi-objective weight-optimized results show very similar topology and behavior.

ability of the algorithm to repeatably solve a given high-level structural problem in a similar optimal manner, even under varying objective pressures.

In general, the results presented here have been left to run for several thousand generations, with a population size of 50 for a total of several hundred thousand evaluations. This corresponds roughly to running on a quad core CPU overnight. It should be noted that good results were obtained with a much lower number of evaluations, but the best results (after running overnight or longer) are presented here.

12.5.2 Continuous Material

The optimization of a similar planar actuator with continuous material distribution was also considered. In the best solution (Figure 12.14), the algorithm converged on a similar functional solution to the discrete material cases. A thin row of stiff material connects the grounded face of the beam to the mid-plane of the beam where the forces are applied. Regions of stiff material connect this strand to the two locations of applied force. The outer half of the beam essentially has no forces acting on it, so this region was again effectively ignored, leading to a mostly random material distribution within it.

12.6   Case Study: Bracket Material Distribution

To further demonstrate the usefulness and flexibility of this algorithm, a case study is considered with a pre-existing bracket design. With the advent of multi-material fabrication, an engineer would hypothetically like to optimize the internal material distribution to maximize strength with respect to weight. Here, the stiff material is proportionately heavier than the flexible material. The desired geometry and resulting material distribution is shown in Figure 12.15.

In order to adapt the algorithm to this type of optimization, the desire bracket geometry is first voxelized at the resolution it is to be simulated at. This voxelized envelope is then inverted and used as a mask on all generated genotypes to remove all voxels not inside the bracket volume immediately before simulating. Then the algorithm proceeds as before using the continuous material interpolation. The algorithm was successful in optimizing the strength to weight ra-

(a)

(b)

(c)
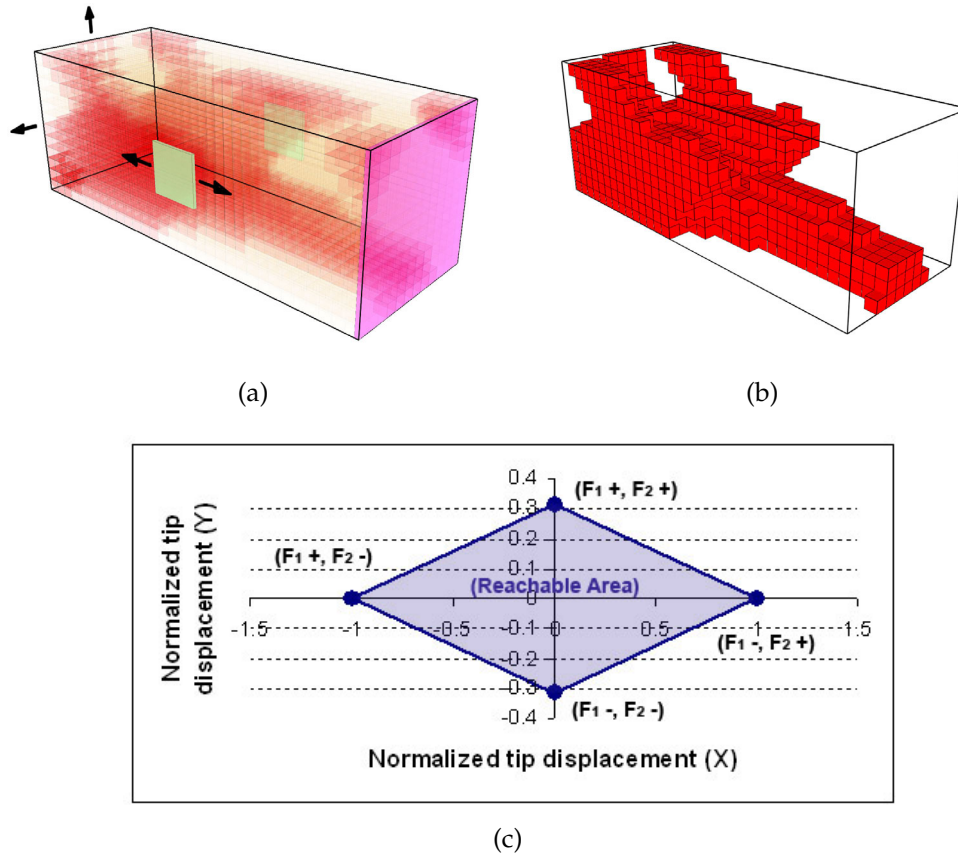
Figure 12.14: The design automation algorithm is able to solve the high-level, non-intuitive compliant actuator problem in the continuous material domain as well. The evolved geometry (a) (arbitrarily thresholded for clarity in (b)) achieves a large reachable area (c). Here, red represents stiff material and transparent yellow is flexible.

tio, and in this case generated a material distribution that makes intuitive sense upon examination. Stiff material connects the grounded regions to the forced region, and the majority of the rest of the bracket is lightweight material.
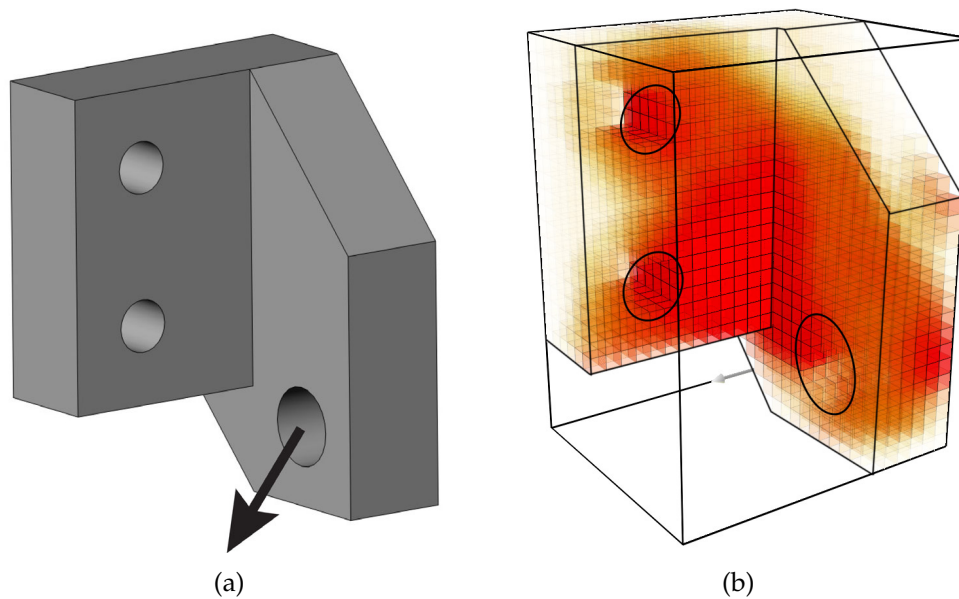
Figure 12.15: The design automation algorithm optimizes the internal material distribution of a pre-designed bracket (a), in order to maximize stiffness and minimize weight. The results are shown in (b), where red represents stiff, dense material transitioning to transparent yellow, which represents flexible, lightweight material.

12.7 Conclusions

Evolutionary algorithms are suitable for designing the complex multi-material objects that have recently become possible to fabricate using additive manufacturing techniques. The lack of existing software design tools to fully take advantage of the capabilities of these fabrication processes enables evolutionary algorithms to fill a new niche in the mechanical design space. Instead of designing an object using traditional CAD programs, evolutionary algorithms allow an engineer to simply set high-level goals to be fulfilled and the blueprint is autonomously generated. Likewise, existing geometries can be optimized for multi-material printing.

The design automation algorithm presented here is successful at optimizing the geometry of single material beams and other shapes. Additionally, by specifying the desired deflected shape of a multi-material beam, or defining vague goals of maximizing deflection area of a compliant actuator, a problem that cannot be easily addressed using current state-of-the-art topological optimization tools can be solved. Additionally, by introducing multiple objectives such as minimizing weight in addition meeting the previous criteria, the flexibility of evolutionary algorithms to easily adapt to competing objectives of very different problems is demonstrated. These results open the door to robust, high-level design tools for complex design problems that can fully utilize the capabilities of multi-material additive fabrication techniques.

CHAPTER 13

EVOLVING DYNAMIC SOFT ROBOTS

13.1 Background and Motivation

13.1.1 Soft Robotics

The field of soft robots is an emerging field which hopes to address the high cost and poor robustness of current state-of-the-art robotics with inexpensive, soft, robust robots. Robots traditionally are composed of rigid links connected by discrete single degree of freedom rotary or linear actuators. Such robots can be very precise and are invaluable and ubiquitous in structured, well-known environments such as assembly lines. Although this precision is desirable in may cases, the downside is a lack of robustness. In order to avoid damage, the kinematics of these machines must be modeled deterministically, then carefully applied to do path planning and avoid collisions.

However, in uncertain and potentially harsh environments, the complexity, precision, and fragility of traditional robots can be a hindrance. Although robots have long been inspired by various aspects of biological systems, a new paradigm in robotics has recently emerged to replicate their robustness and resilience. These "soft" robots trade deterministic control for probabilistic models, but gain robustness [149]. Several actuation methods have been explored for soft muscles and robots such as jamming [127], electroactive polymers [4], pneumatics [148], and shape memory alloy (SMA). These all place constraints on how the internal locomotion forces can be applied as well as the resulting

geometry. In addition, soft actuators have been well explored as fitted to robots with rigid kinematics. Although these gain some advantages of fully soft robots, the rigid kinematics still lie within the traditional robotics paradigm.

Here an isometric volume changing material is incorporated for truly distributed actuation without geometric constraints. This allows the design automation algorithm complete freedom over the method and magnitude of locomotion based purely on material distributions. Thus, the function of an evolved soft robot is inextricably paired with its morphology without imposing additional forces at arbitrary locations. Since the complexity of how locomotion is achieved is embedded in the shape, a very simple control scheme can be used. Here, a simple global sinusoidal variation in volume of the designated actuation material is imposed.

13.1.2   Evolutionary Robotics

With a this new paradigm of freeform soft robots, traditional design methods no longer are effective. Traditional computer aided design (CAD) tools are typically inappropriate for designing amorphous machines with continuous morphology and actuation. Such tools rely on feature-based modeling approaches that work well for well-defined geometric primitives made of a single material. However, the lack of constraints on the shape and material distribution of soft robots indicate that existing CAD programs would be ineffective in their ability to fully take advantage of the design space offered. This implies that a new class of design automation algorithms is in order. In order to address such a large and complex search space, evolutionary algorithms are again a natural choice.

The field of evolutionary robotics has explored methods for generating interesting and functional robot morphologies [134]. Ever since the early work in evolving virtual [165] and physical [108] creatures, many examples have been published of evolved walking, [142] running [212, 72], and swimming [77] robots. These simulations all use rigid-body simulations of discrete components connected by rotational or linear joints. Many interesting biological forms of locomotion, however, are not modeled well by rigid links and joints such as the earthworm [147] and the amoeba [120]. More recent work on morphogenetic robotics explores the development of more complex morphologies using many rigid links, but these bodies are still inherently discrete and relatively sparsely connected [71, 11].

In this chapter, the challenges of representing evolvable multi-material freeform shapes and evaluation (simulation) of the resulting soft bodies are discussed. All three genotypic representations presented earlier (GMX, DCT, CPPN) are explored to generate the material distribution in the soft robot phenotype. Soft body simulation of the robot is accomplished using the dynamic simulation discussed earlier to model the dynamics of the resulting amorphous machines under the actuation material expansion, gravity forces, and non-linear ground friction. These results open the door to a new design space that more closely mimics the freeform, amorphous nature of biological systems.

13.1.3   Freeform Fabrication of Soft Actuators

As discussed earlier, fabricating arbitrary soft and hard structures is well within the means of current additive manufacturing processes. However, a signifi-

cant missing link in soft robots becoming ubiquitous is the ability to directly print generic volumetric actuators. Many examples are present in literature of additively manufactured robots with actuators added after fabrication. However, these are limited to traditional rotational or linear actuators, [142] which would severely limit the generality and methods of actuation of an amorphous robot. Here an ideal volumetric actuator is explored in simulation, in which a given material expands isometrically (equally in all dimensions). A useful analogy is evolving robots with materials of varying thermal expansion coefficients (CTE), then "actuating" the robot by globally or locally varying the "temperature". Thus materials with a simulated CTE of zero will not change volume, whereas materials with a non-zero CTE will swell or contract isometrically as the temperature changes.

In these experiments, the temperature is assumed to vary sinusoidally over time, and slowly enough that actuation across the entire structure occurs simultaneously without heat diffusion effects. The period and amplitude of this temperature variation determine how dynamic the movement of the robot is. More complex actuation patterns including evolved brains will be examined in the future.

In order to physically demonstrate the design presented here, the external environment is modulated in order to achieve selective volumetric expansion and contraction of autonomously fabricated regions. By varying the pneumatic pressure of the environment, regions of the soft robot that are pneumatically isolated from the environment experience a change in relative pressure. Because the surrounding material is soft, this results in a change of volume. Regions of the robot that are pneumatically connected to the environment can

Table 13.1: Summary of evolutionary algorithm parameters used for evolving locomoting amorphous robots.

Parameter	Value
Population Size	50
Selection method	Deterministic crowding
P(Mutation)	0.2
Solution Encoding	Gaussian Points

quickly equalize to the changing external pressure, so no volume change is observed. This allows structures of reasonable size to be actuated at a millisecond timescale, as opposed to several minutes per actuation for thermally actuated robots.

13.2 Evolutionary Algorithm Parameters

The parameters of the evolutionary algorithm used here are similar to those used in the static analysis. The only major difference lies in the evaluation step. Each evaluation of an amorphous machine was broken into two segments. The relaxation segment settles the object under gravity and friction, allowing it to come to rest in a neutral position. In the movement segment, temperature oscillations begin. After 10 complete temperature cycles, the magnitude of change in position of the center of mass during the movement segment in the positive X direction is returned as the fitness for a given individual. Table 13.1 summarizes the details of the evolutionary algorithm.

The solutions presented here were each evolved on a single quad-core desk-

top computer. Each solution was voxelized into a $20 \times 20 \times 20$ workspace, which provided suitably accurate resolution while remaining computationally feasible. At a rate of approximately 3-15 seconds per evaluation (depending on number of instantiated voxels), 20,000 evaluations in a 24 hour day was typical. Small population sizes work well with the deterministic crowding method, which allowed for an increase in the number of generations explored in a fixed computational time.

Several parameters of the simulation were chosen to be of interest for exploring further. The first is the level of dynamic response. This term refers to the importance of the momentum term of the robot. An object with a high level of dynamic response could be a very dense, soft, rubbery object actuated near resonance, where movements can be significantly out of phase with the actuation. Conversely, an object with a low level of dynamic response would be light and stiff (or actuated very slowly), such that the static movement dominates the momentum terms. Several combinations of static and dynamic friction were also explored, ranging from realistic values to exaggerated stick-slip scenarios. For the bulk of experiments, the coefficient of dynamic friction was 0.3 and the coefficient of static friction was 1.0.

13.3 Evolved Soft Robots

13.3.1 Two Material Locomoting Robots

The evolved behaviors of the amorphous robots generally took advantage of a combination of dynamics and non-linear friction to make forward progress.

248

Two modes of movement in the desired direction were generally observed: Several robots made significant progress by maximizing the distance traveled as they fell and flopped over. This movement was often aided by the actuation cycles gradually tipping the robot over, but this method of movement does not count as true locomotion because it cannot be sustained over an indefinite distance. The more successful mode of movement involved scooting, in which the robots expanded and contracted in specific ways, making and breaking contact friction selectively to make forward progress. Several observed solutions made use of a combination of flopping and scooting.

In the first experiments, a palette of two equal stiffness and density materials was used. The first material (shown in blue) had a CTE of zero, signifying that it was not actuated. The second material (shown in yellow) had an arbitrary CTE of 0.01. The temperature was varied sinusoidally globally with an amplitude of 30 degrees, leading to a $\pm30\%$ change in volume of the actuated material. The period of oscillation was 500 time steps.

Comparison of Representations

The three representations under consideration were all run three times for a total of nine evolutionary runs. Figure 13.1 displays the average and standard error of the best solution of each of the three runs. The GMX representation outperformed the other representations consistently. This may be a result of locality that preserves geometric novelties in the crossover process and can make small changes to specific areas of the robot through mutation.

The DCT representation fell behind and had a very large standard error,
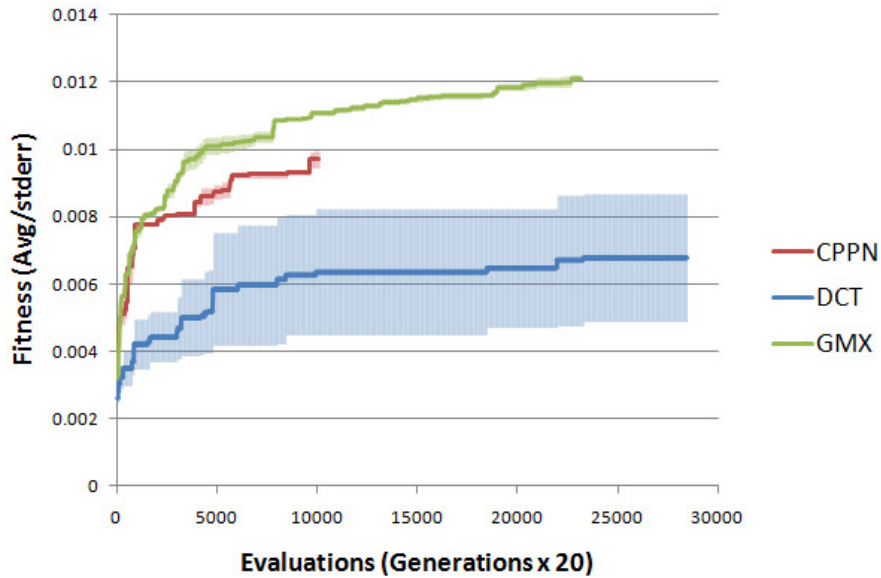
Figure 13.1: Three independent runs were completed for each of the three representations. The average and standard error of the three best solutions are plotted for each. The GMX representation outperformed the others.

which means that the evolutionary algorithm was not able to consistently find good solutions. This is likely because each mutation in the genome has a global effect across the entire structure, a characteristic that couples the mutations and prohibits small, subtle changes.

The CPPN representation, as implemented, was not well suited to evolving freeform amorphous morphologies. Mutations drove the solution toward filling the entire workspace with material, a trend that significantly slowed the simulation down (since more elements needed to be simulated) and also led to fewer interesting geometries. Resulting amorphous robots generated by each representation are shown in Figure 13.2.

(a)

(b)

(c)
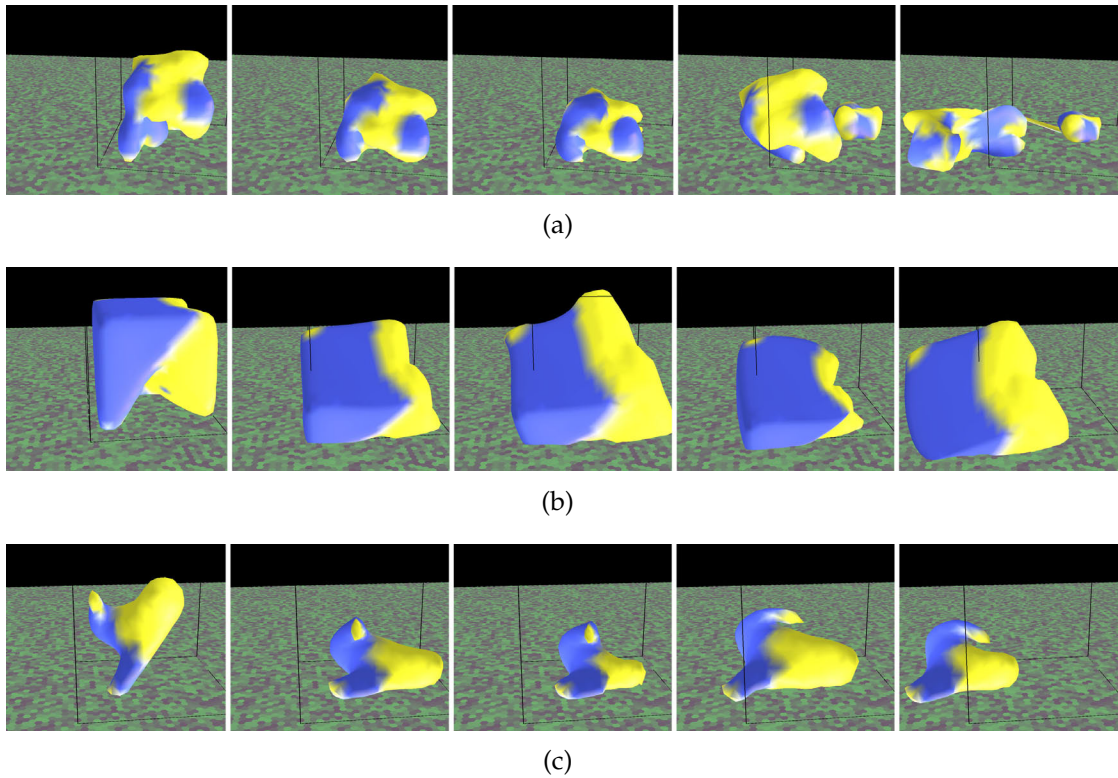
Figure 13.2: Evolved robots for the DCT (a), CPPN (b), and GMX (c) representations demonstrate successful locomotion. The blue material is passive, while the yellow material changes volume sinusoidally. The first frames for each show the initialized shape, the second frames show the settled result under gravity, and the following frames are snapshots of its motion. Direction of motion is to the left.

13.3.2 Evolved Three Material Soft Robots

A second volumetrically actuated material was introduced to explore the possibilities enabled by multiple actuation modes. The second actuator material was specified with the same stiffness and density as the others, but with 90 degree phase lag in actuation. It was hypothesized that this would enable different modes of locomotion, such as continuous rolling or more interesting gaits. However, only the more primitive locomotion modes of flopping (Figure 13.3(a)) and scooting (Figure 13.3(b)) were observed.

(a)

(b)

(c)
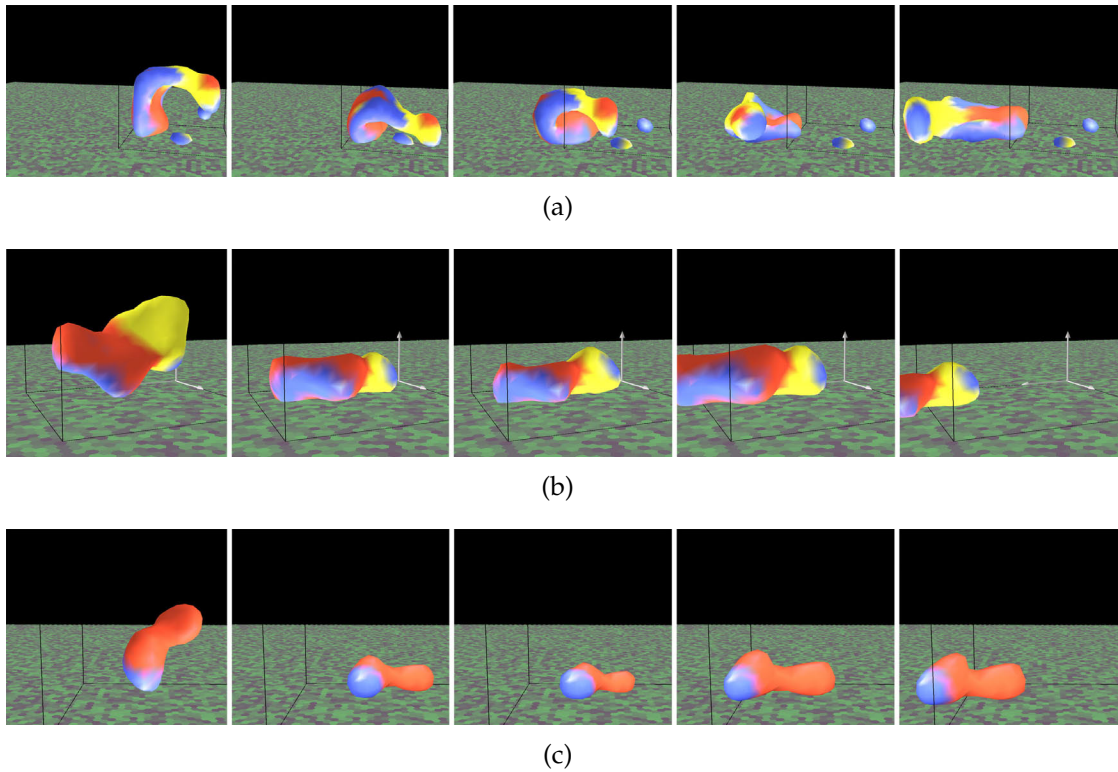
Figure 13.3: Evolved three material amorphous morphologies showing flopping (a), scooting (b), and dynamic bouncing (c) behaviors. The yellow and red materials are both sinusoidally actuated, but 90 degrees out of phase. Direction of motion is to the left.

By varying the actuation speed of the temperature fluctuations and the internal material damping, the importance of momentum effects in the amorphous morphologies was adjusted to dominate the motion. The best solution of the dynamic runs ended up using only one actuator material, as shown in Figure 13.3(c). Based on the size and mass of the optimized object, however, the dynamics were strongly exploited to bounce forward.

Different parameters for friction bias the solution towards different modes of locomotion. Experiments were run with very low dynamic friction and high static friction (0.1/5.0) and with dynamic and static friction values that were

very close (0.4/0.5). The solutions with very high static friction tended to exhibit flopping/rolling over behavior, such as in Figure 13.3(a), since the force to overcome static friction was extremely high. However, solutions with moderate static and dynamic friction tended towards the scooting locomotion, such as in Figure 13.3(b).

13.4 Physical Implementation of a Locomoting Soft Robot

13.4.1 Simulation Parameters for Fabrication

To physically fabricate a soft robot demonstrating the effectiveness of the design automation algorithm in the real world, realistic parameters must be input into the simulation. The primary concern was replicating the dynamic properties, which are a function of mass, actuation speed, and actuation magnitude. Based on the materials to be used in fabrication of the volumetric actuator, the volume change was selected to be $\pm 20\%$. The relative speed (period) of this oscillation determines to what degree the dynamics of the structure play a role in the movement. Fast oscillations around the effective resonance of the shape will result in physical motion substantially out of phase with the input. Here the oscillation period was selected to be significantly slower that any resonances so that the dynamic response of the structure does not dominate the motion.

A resolution of $20 \times 20 \times 20$ (8000 possible) voxels was selected as a reasonable sized domain to work within. Each initial random solution was initialized with 88 randomly generated Gaussian points. Although mutation operators occasionally add or subtract points from this total, the final solutions generally
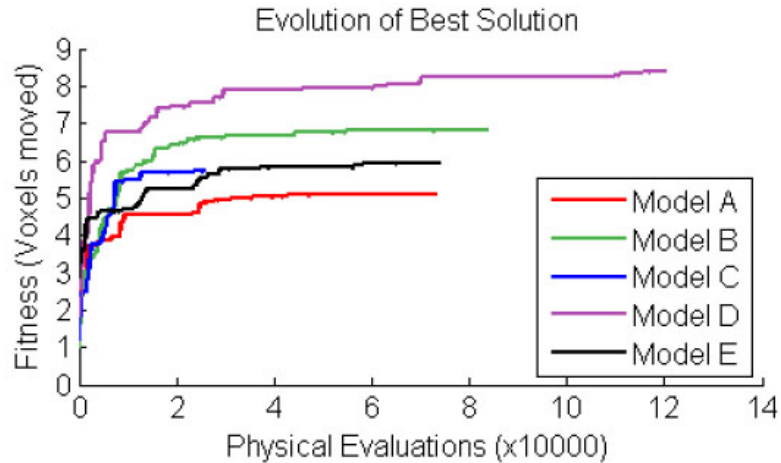
Figure 13.4: The best results of 5 independent evolutionary algorithm runs designing freeform soft robots. The fitness over time increases dramatically at first but then levels off as expected.

had around 100 total points. The evolution history of five separate runs of the evolutionary algorithm are shown in Figure 13.4. The solutions were run an arbitrary amount of time on a single desktop computer, ranging from overnight to a couple days, corresponding to tens of thousands of individual physics simulations.

All five solutions are shown in Figure 13.5. The fundamental mode of locomotion is similar for all of them, as evidenced by very similar traces of position of the center of mass over time (Figure 13.6). All examples exhibit a scooting motion where the expanding phase of the actuator pushes the front of the blob forward while the contraction phase resets the back of the robot to repeat this motion.

254
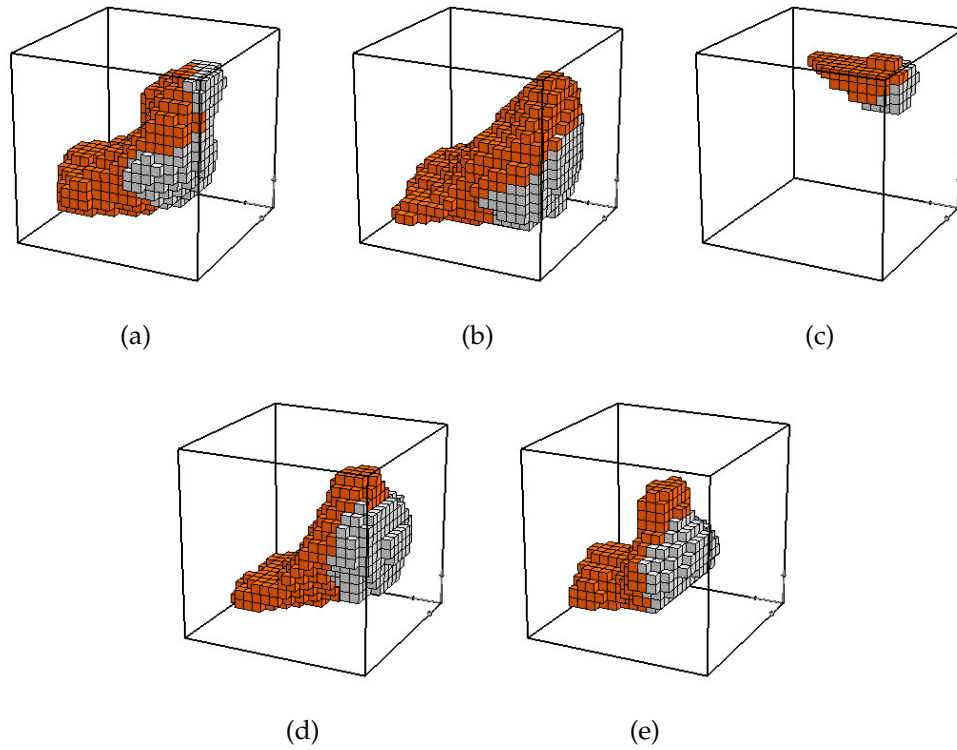
Figure 13.5: The resulting freeform robots of the five evolution runs all converged to similar morphologies and locomotion modes for the physical parameters assigned in these experiments. The orange material actuates volumetrically while the white material is passive. Intended direction of motion is to the right.
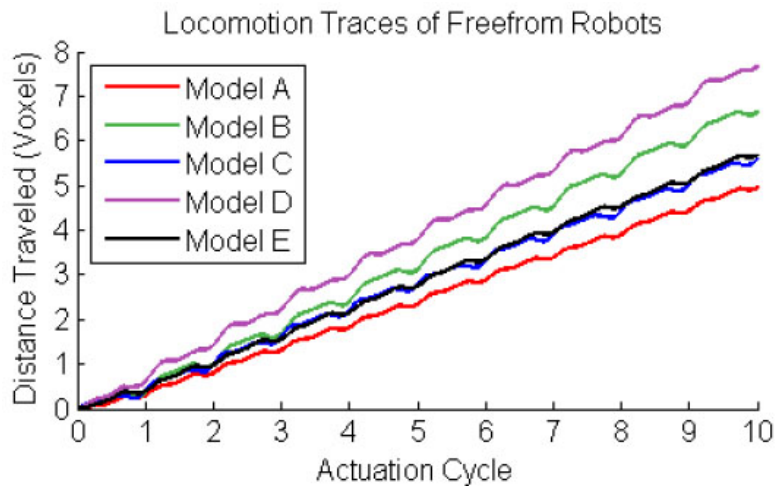
Figure 13.6: The movement of the best evolved soft robot at the end of each run show very similar locomotion pattern.
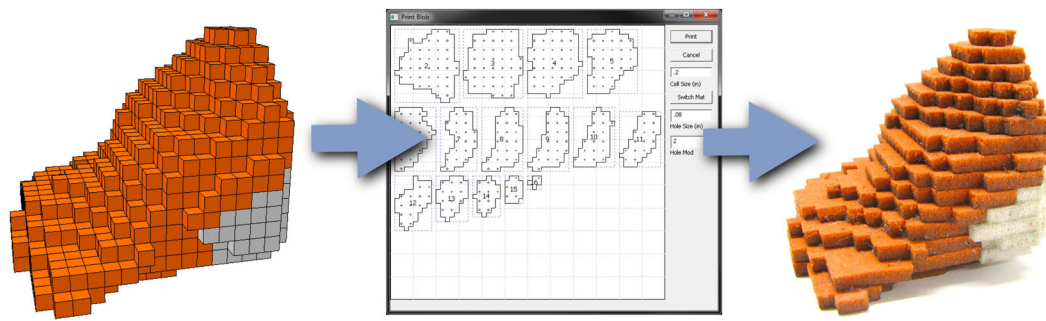
Figure 13.7: The final simulated soft robot was decomposed into layers with alignment holes to be automatically fabricated using a laser cutter. The resulting layers were manually stacked onto alignment pins to assembly the robot.

13.4.2 Autonomous Fabrication

Of the five designs, model B was arbitrarily selected for fabrication. A custom layered manufacturing process was developed to use pre-existing open-cell and closed-cell foam rubbers to fabricate the robots. A routine was written to send the profile of each voxel layer to a laser cutter to be autonomously cut out with alignment holes between layers. The individual layers were then assembled onto alignment pins with glue interspersed between layers to finish the 3D structure. Although this process is not yet fully autonomous, human involvement is minimal and no particular skill is needed to stack and assemble the fabricated layers (Figure 13.7).

The robot was chosen to be 55mm in length. Based on the the simulated size, density, and actuation period of the evolved solutions, this size gave a similar scale of relative dynamic internal forces when actuated with a period of approximately 10 seconds. The effective dynamic actuation force F_d for each case was calculated according to:

$$F_d = m|a| \tag{13.1}$$

where m is the mass of the total robot and $|a|$ is the magnitude of the acceleration assuming sinusoidal actuation. Given an actuator displacement d that is proportional to the size of the robot and an actuation frequency f, an effective position P can be modeled as

$$P(t) = dsin(ft) \tag{13.2}$$

which when differentiated twice yields

$$A(t) = df^2sin(ft) \tag{13.3}$$

where A is the effective acceleration. Thus, the magnitude of acceleration $|A|$ is

$$|A| = df^2 \tag{13.4}$$

so that

$$F_d = mdf^2 \tag{13.5}$$

This effective dynamic actuation force is not quantitatively meaningful, but for a given geometry of soft robot it provides a qualitative measure of the similarity of behavior under sinusoidal actuation. For the parameters used here, the
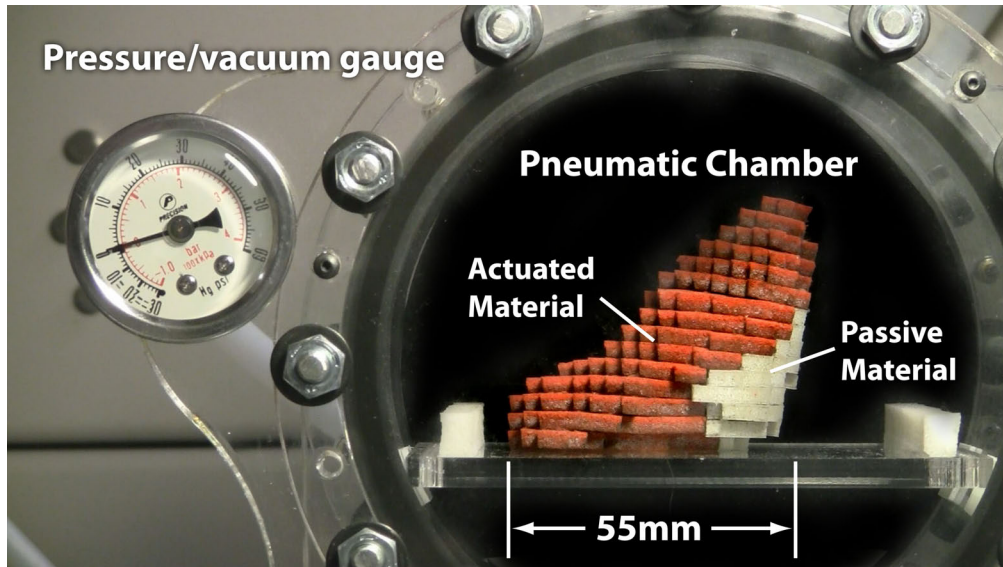
Figure 13.8: The soft robot was placed in a pneumatic pressure/vacuum chamber to demonstrate locomotion. Selective volumetric actuation was attained using closed cell (orange) foam rubber, which changes volume as the air pressure changes periodically. Open cell foam rubber (white) does not change volume.

simulation soft robot had a F_d value of $6.15\mu N$ and the actual robot had a value of $6.09\mu N$ (See Table 13.2). This is more than close enough to assume similar qualitative behavior would be observed between simulated and actual robots.

The physical robot was placed in a pressure and vacuum chamber to verify the locomotion (Figure 13.8). The trace of the position of the physical robot at the end of each actuation cycle is plotted with the simulation results (Figure 13.9). In each case, the distance traveled is normalized by the size of the voxels. The physical robot displays robust locomotion, and progresses at a very similar rate as the simulation predicts.

Table 13.2: Comparison of simulated and physical blob robot physical parameters and performance.

	Simulated	Physical
Size (mm)	18	55
Weight (grams)	1.675×10^{-3}	12.18
Actuation Period (s)	0.07	10
Effective Dynamic Actuation Force (μN)	**6.15**	**6.09**
Speed (% Body length/actuation cycle)	3.38%	3.96%
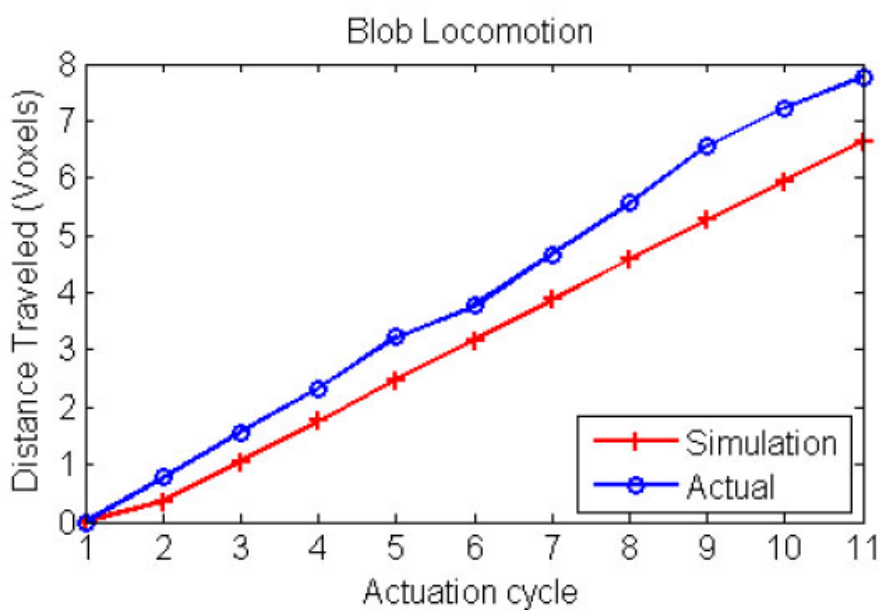
Figure 13.9: The movement of the simulated and actual model B soft robot show similar velocities, suggesting that the physics simulator is an accurate predictor of real-world performance.

13.5 Conclusions

For decades of mechanisms and robotics, the paradigm of discrete rigid links and single degree of freedom joints has prevailed. Here the effectiveness of evolutionary algorithms paired with a Gaussian mixture representation has demonstrated the capability of autonomously designing freeform shapes and material distributions to achieve high-level functionality in the physical world. The same algorithm can be applied to very different mechanical design problems by simply changing the function that evaluates how good a potential design is. This is easily extended to exploring multiple, competing objectives by incorporating a Pareto front of solutions.

In the future one can envision directly manufacturing pneumatically actuated soft robots with additive manufacturing techniques. By utilizing soft, elastic material, discrete hollow cells could be fabricated. Those which are sealed off from the exterior environment would act as actuators and others which are open to the environment would be passive. By incorporating a source of pressure within the elastic freeform robots connected to the sealed actuation regions, the designs presented here would work without adaptation in our everyday environment where atmospheric pressure is constant. This would also allow more complex actuation patterns such as out-of-phase actuation of different regions of the soft robot. With such distributed actuation, morphological computation could be explored for the purpose of controlling gaits.

Soft robots are also suitable for fabricated by digital manufacturing processes, which could alleviate many of the limitations of traditional additive manufacturing techniques. For instance, to replicate the results here, only two

voxel are necessary. Both could be made of soft, hollow rubber cubes or spheres. One material would be sealed, the other would have holes, allowing the pressure to equalize with the environment. By arranging the sealed voxels in the actuated regions and the permeable ones in the non-actuated regions, similar locomoting behavior is expected to be observed.

With the unconstrained design space enabled by multi-material additive manufacturing techniques, a whole new design space of soft, amorphous robots is possible. There are still challenges that remain regarding how to effectively design functional objects with discrete components and how to best actuate a truly amorphous blob robot. Volumetric actuation provides a viable method for actuating soft robots, and offers many advantages for design automation because locomotion is dictated purely by material distributions. However, more work is needed to develop self contained and powered volumetrically actuating materials. As these challenges are addressed, robots will be better equipped to operate robustly in highly uncertain and dangerous environments.

APPENDIX A

STL 2.0: A PROPOSAL FOR A UNIVERSAL MULTI-MATERIAL ADDITIVE MANUFACTURING FILE FORMAT

Disclaimer: A subsequent version of the proposed AMF file format is under development and consideration for inclusion in a set of standards developed by the ASTM F-42 committee on Additive Manufacturing. The information presented here does not represent the actual ASTM AMF standard.

A.1 Introduction

The de-facto standard STL file format has served the rapid prototyping community for over two decades. However, it falls short with the advent of new technological developments such as the ability to handle multiple and graded materials, and the ability to specify volumetric digital patterns and surface colors. A variety of requirements were studied for additive fabrication technologies and a new compact XML-based file format is proposed to meet the needs of the entire additive manufacturing community.

The new Additive Manufacturing File (AMF) format allows the resolution-independent specification of geometry and material properties. Regions may be defined geometrically either using a triangle mesh, using functional representations, or through a voxel bitmap. Each region is associated with a material, which may be defined as a base (single) material or hierarchically by a combination of other materials, either functionally (enabling smooth gradients) or voxel-wise (for arbitrary microstructure). Files can be self-contained or refer to

external or online material libraries. With a simple conversion, the AMF file format is both forward and backwards compatible with the current standard STL format, and the flexibility of the XML structure enables additional features to be adopted as needed by CAD programs and future additive manufacturing processes. Code and examples are publicly available.

Additive manufacturing technology is quickly evolving from producing primarily single-material, homogeneous shapes to producing multi-material geometries in full color with functionally graded materials and microstructures. Already, several vendor-specific file formats have been introduced to overcome the lack of a standard interchange file format that contains these features. This results in little or no compatibility between files for machines from different vendors, such as the ObjDF format for Objet's Connex^{TM} printers [135] and ZCorp's color ZPR format [207]. Here, a framework is proposed for a simple, intuitive file format to address the severe limitations of the industry standard single-material STL file format with regards to the additive manufacturing technology of the future.

There is a trade off between the generality of a file format, and its usefulness for a specific purpose [124]. Thus, features designed to meet the needs of one community may hinder the usefulness of a file format for other uses. In order to be successful across the field of additive manufacturing, a file format should address the following concerns:

1. **Technology independent:** A file format should describe an object in a general way such that any machine can build it to the best of its ability [80]. A suitable file format should be resolution and layer-thickness independent, and must not contain information specific to any one manufacturing

process or technique. This does not negate the inclusion of properties that only certain advanced machines support (color, multiple materials, etc.), but they must be defined in such away to avoid exclusivity.

2. **Simplicity:** A file format should be easy to implement and understand. It should be read and debugged in a simple ASCII text viewer to encourage rapid adoption. However, this is at odds with the size requirements of the file format. Also, there is no place for redundancy in a file format, and no identical information should be stored in multiple places.

3. **Scalability:** As the parts to print increase in size and complexity, the file format should scale well. This includes being able to handle large arrays of identical objects, complex repeated internal features (e.g. meshes), and multiple components arranged in the optimal packing for printing.

4. **Future compatibility:** In order to remain useful in a rapidly changing industry, a file format must be easily extensible while remaining compatible with earlier versions and technologies. This allows new features to be added as advances in technology warrant, while still working flawlessly for simple homogeneous geometries on the oldest hardware.

Additionally, there are several specific aspects of an additive manufacturing interchange format that are addressed in the proposed format to meet the demands of the current generation of research and hardware. These pressing aspects include:

- Multiple/graded materials

- Colors and surface textures

- Hierarchical microstructure and mesostructure

A.2 Background

For the last two decades, the STL file format has been the industry standard for transferring information between design programs and the software specific to a given additive manufacturing hardware [95, 89, 63]. An STL file contains information only about a surface mesh, and has no provisions for representing color, texture, material, etc, although several extensions have been proposed (but not widely accepted) [29, 173]. The surface is represented by an unordered list of triangles, and each triangle is defined by 12 floating point numbers. A 3D surface normal is defined, followed by three coordinates that define the vertices of the triangle in three dimensions. Already, this contains redundant information, since the surface normal can be calculated from the order and location of the three vertices. By default, the right-hand rule is used to define the direction of the normal based on the order the points are encoded. Since each triangle is represented separately, each vertex must be written repeatedly for every triangle that shares that vertex (three or more times). This introduces leaks, where small rounding errors result in vertices that do not precisely line up, which make subsequent slicing algorithms ineffective without an intermediate "welding" step to (hopefully) repair these defects.

Also, the STL file has no provisions for defining the physical units intended. Even though pre-processing software can make an educated guess between inches or mm depending on the build size of the machine, there is still unnecessary ambiguity. An additional point of confusion regarding the STL file is that in fact there are two separate file formats that may be used: binary and ASCII. The ASCII version exists to make the format human readable, but the binary version is often used by mature programs to minimize storage space. A summary of the

Table A.1: Advantages and disadvantages of the current STL format used universally in additive manufacturing processes.

Advantages	Disadvantages
Simple	Geometry leaks
Sequential memory access*	No specified units
Portable	Unnecessary redundancy
	Incompatible with color, multiple materials, etc.
	Poor scalability
	Lacks auxiliary information

*Does not require large amounts of RAM, critical in '80s

advantages and disadvantages of the current STL file format is shown in Table A.1.

Many 3D file formats have been used over the years for a wide variety of purposes. Several of these have been proposed for use within the Additive Manufacturing (AM) community to replace the STL. However, none has gained traction [145, 136, 152] for two main reasons. First, up until recently the AM end-user community has not needed functionality past what the STL offered. Secondly, the more general file formats in existence include many features that are irrelevant for the AM field, and do not include many features that would be useful, which causes unnecessary coding and complexity.

Several file formats that have been proposed for the additive manufacturing community are summarized below:

- X3D (VRML): (Virtual Reality Modeling Language) This mesh-based file format was intended to allow 3D content to be viewed over the web. As such, it includes information about a 3D surface and its color, but also in-

formation that is not relevant for AM, such as transparency, animations, lights, sounds, and embedded navigation URLs. Other disadvantages include no provisions for defining multiple materials within a given mesh or arbitrary microstructure.

- STEP: This format is a general-use solid model representation, using extruded and swept solids, wireframe, boolean primitives modeling, and many other modeling paradigms to represent a 3D object [58]. As such it is unnecessarily complex and difficult to implement for the needs of the AM community.

- PLY: This format was intended to store and view data from 3D scanners. It uses polygon meshes and can include information about texture and color. However, like other purely mesh-based formats it does not define materials or microstructure volumetrically.

- SAT: The ACIS SAT format is widely used for boundary-representation (B-Rep) objects in CAD packages. However, the entire format revolves around its internal topological data structure, which makes it difficult to understand and unsuitable for an exchange format.

- OBJ: This meshed file format is simple, compact, widely accepted in the 3D modeling community, and can map textures easily. However, it lacks the ability to define materials or microstructure volumetrically.

- DXF: Although the DXF format allows the definition of 3D triangle meshes and solids, it was originally intended for 2D drawings and remains best suited for such.

- 3DS: Another triangular mesh-based format with color and texture information. It is limited to 65536 vertices and polygons, and contains much

information not necessary for the AM industry, such as lighting and animation info.

- SLC: The SLC format represents individual 2D slices of a 3D object as contours representing internal and external boundaries. However, since a specific Z-slice distance is assumed, this format is not suitable for a cross-platform interchange format.

A.3 Proposed AMF Format

In order to address the lack of a suitable file format for the additive manufacturing industry, the following framework is proposed for a simple, flexible, extensible file format.

First, the information should be stored in standard XML format. Using this widely accepted data format opens the door to a rich host of tools for creating, viewing, manipulating, and storing AMF files. However, the beauty of XML is that it accomplishes this without alienating programmers who wish to code low-level native parsing/storing routines. XML is human readable, which makes debugging errors in the file possible. Unlike many of the current file formats which specify completely separate ASCII and binary file formats, the AMF format will be store entirely in ASCII XML, then compressed if desired in a post-processing step using highly optimized standardized compression routines. This allows significantly smaller file sizes without maintaining multiple parallel file specifications.

Another significant advantage of XML is its inherent flexibility. Missing or extra parameters do not present a problem for a parser as long as the document

conforms to the XML standard. Practically, this allows new features to be added without needing to update old versions of the parser, such as in legacy software.

A.3.1 Top Level Tags

There are four top level tags in the AMF file, of which only a single object tag is required for a fully functional AMF file that encompasses the usefulness of the STL format.

<Object> The object tag defines a region or regions of material, each of which are associated with a material ID for printing.

<Constellation> The constellation tag hierarchically combines objects and other constellations into a relative pattern for printing. If no constellation tags are specified, each object tag will be imported with no relative position data.

<Palette> The palette tag defines one or more named materials for printing with an associated material ID. If no palette tag is included, a single default material is assumed.

<Print> The print tag specifies which constellations and/or objects to print, and is necessary only if multiple constellations or objects introduce ambiguity as to how many of each to print.

```
<?xml version="1.0"?>
<AMF>
  <Object PrintID = "0" units = "mm">
    <Mesh>
      <Vertices>
        <Vertex VertexID="0">
          <VertexLocation x="0" y="1.332" z="3.715"/>
        </Vertex>
        <Vertex VertexID="1">
          <VertexLocation x="0" y="1.269" z="3.715"/>
        </Vertex>
        ...
      </Vertices>

      <Region FillMaterialID = "0">
        <Triangle V1 = "0" V2 = "1" V3 = "3"/>
        <Triangle V1 = "0" V2 = "1" V3 = "4"/>
        ...
      </Region>
    </Mesh>
  </Object>
</AMF>
```

Figure A.1: The AMF file can use a simple list of vertices and triangles to define a mesh and replicate the functionality of an STL file, but without leaks. The XML-compliant format makes storing, parsing, and reading the file easy.

A.3.2 Examples

Basic STL Equivalent

The first example presents a simple case of an AMF file that contains all the functionality of an STL file (Figure A.1). Note that in the sample code, the ellipses (...) denotes a continuing, similar list of tags.

The opening <AMF> tag is necessary to denote the file type, as well as fulfill the requirement that all XML files have a single root element. The top level <Object> tag contains two sub-tags: <Vertices> and <Region>. The required <Vertices> tag lists all vertices that are used in this object, and assigns a unique

vertex ID to each. The sub-tag <VertexLocation> gives the position of the point in 3D space. After the vertex information, at least one <Region> tag must be included, using the <Triangle> tag to define triangles by the right-hand rule (vertices listed in counter-clockwise order as viewed from the outside) from the indices of the defined vertices. The problem of STL leaks is solved by the fact that common vertices of triangles reference the same <Vertex> tag.

Multiple-material STL equivalent

One of the most critical limitations of the current STL format is the lack of support for multiple materials. With the AMF format, this minor extension introduces the <Palette> tag. Here, any number of materials may be defined by name and associated with a material ID. Other relevant attributes may also be added to each material. Then, within the <mesh> tags, additional <Region> tags can be added that reference different material indices. Since the vertex list is shared, no leaks are introduced at the boundaries between materials (Figure A.2).

Gradated material example

The advantages of the material palette come from "meta" materials that can be defined using other materials. These "meta" materials may be defined functionally, enabling arbitrarily gradients of two or more materials to be defined as a single material (Figure A.3). When defining functions, the only variables that should be used are "x", "y" and "z", representing the respective spatial coordinates. A list of acceptable operations is given in Table A.2.

```
<?xml version="1.0"?>
<AMF>
  <Palette>
    <Material MaterialID = "0">
      <Name>StiffMaterial</Name>
    </Material>
    <Material MaterialID = "1">
      <Name>FlexibleMaterial</Name>
    </Material>
  </Palette>

  <Object PrintID = "0" units = "mm">
    <Mesh>
      <Vertices>
        ...
      </Vertices>

      <Region FillMaterialID = "0">
        ...
      </Region>
      <Region FillMaterialID = "1">
        <Triangle V1 = "5" V2 = "6" V3 = "7"/>
        <Triangle V1 = "5" V2 = "7" V3 = "9"/>
        ...
      </Region>
    </Mesh>
  </Object>
</AMF>
```

Figure A.2: With the addition of the palette tag, multiple materials may be easily defined and assigned to different regions. A common vertex list ensures no leaks between materials.

The computational process to determine the relative concentrations of materials at any given sample point is quite easy. The equation tag for each material present is evaluated to a single number by plugging in the desired 3D spatial coordinates. All values are assumed to be zero where not defined, and are also floored at zero, so that it is possible to have regions that are fully a single material. Then the values for each material are added and the actual concentration of each is calculated as its proportion of the sum. Any equation using an undefined or self-referencing material ID is ignored.

Table A.2: Operators used in the functional representation of material gradients and geometry in the AMF format.

Precedence	Operator	Description
1	()	Parentheses block
2	^	Power
3	$*$	Multiply
3	/	Divide
3	%	Modulus
4	+	Add
4	-	Subtract
5	=	Equal
5	$<, <=$	Less than (or equal to)
5	$>, >=$	Greater than (or equal to)
6	&	Intersection (Logical AND)
6	—	Union (Logical OR)
6	\	Difference (Logical XOR)
6	$\sim$	Negation (Logical NOT)

Mesostructure example

Meta-materials may also be defined as a tiled combination of materials and empty space to create micro and meso structures that are repeated throughout the region. These may be defined functionally, by a mesh or a voxel bitmap. This allows complex internal structure to be defined once, then tiled for efficient use of storage space. The region is tiled at its envelope dimension, or in the case of the functional representation the modulus operator may be used to create periodic structures (Figure A.4).

```
<?xml version="1.0"?>
<AMF>
  <Palette>
    <Material MaterialID = "0">
      <Name>StiffMaterial</Name>
    </Material>
    <Material MaterialID = "1">
      <Name>FlexibleMaterial</Name>
    </Material>
    <Material MaterialID = "2">
      <Name>GradientMaterial</Name>
      <Equation UseMaterialID = "0">0.30*X</Equation>
      <Equation UseMaterialID = "1">0.30*(1−X)</Equation>
    </Material>
  </Palette>

  <Object PrintID = "0" units = "mm">
    ...
  </Object>
</AMF>
```

Figure A.3: A "meta" material is defined in the material palette as a functional combination of two previously defined materials. This enables smooth gradients and arbitrary 3D material distributions within an object.

Colored surface example

It is desirable to be able to define surface properties of the geometry, such as color, that do not necessarily need to penetrate into the 3D body. Thus, a <Color> tag can be introduced at either the <Object> level, the <Region> level, or the <Vertex> level, in increasing order of precedence. Thus, the entire object or region can be made one color with a single tag, but certain vertices can be recolored individually. All RGB values must be real values ranging from 0 to 1. Also, it is possible to map images onto the surface of the part. This is accomplished by adding <ColorMap> or <ColorFile> tags with unique map ID numbers. Then vertices in the vertex list are associated with the desired map ID and pixel location within the image. This way, any triangle in which all three vertices have a <VertexMap> tag with the same ID will have that associated triangle

```
<?xml version="1.0"?>
<AMF>
  <Palette>
    <Material MaterialID = "0">
      <Name>StiffMaterial</Name>
    </Material>
    <Material MaterialID = "1">
      <Name>FlexibleMaterial</Name>
    </Material>
    <Material MaterialID = "2">
      <Name>MesoStructureMaterial</Name>
      <Lattice>
        ...
      </Lattice>
      <Structure>
        <X_Voxels>10</X_Voxels>
        <Y_Voxels>10</Y_Voxels>
        <Z_Voxels>10</Z_Voxels>
        <Data Layer = "1">0000110000...</Data>
        <Data Layer = "2">0001111000...</Data>
        ...
      </Structure>
    </Material>
  </Palette>

  <Object PrintID = "0" units = "mm">
    ...
  </Object>
</AMF>
```
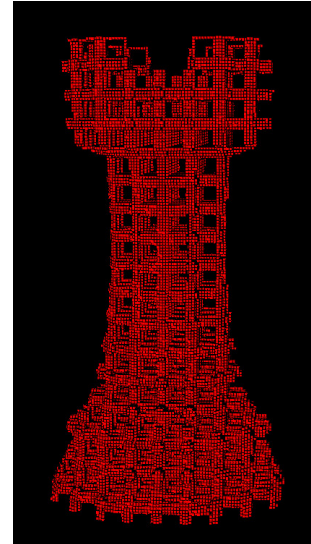

Figure A.4: Mesostructure is defined by a voxel bitmap here using a simple repeated mesh pattern. Any tiling size and material combination may be used.

of the image mapped onto it. Vertices may have more than one <VertexMap> tag, which can also refer to material distribution on the surface (with an associated depth) or a physical texture (bump map) that is to be mapped onto the surface. In the case of ambiguity (I.E. multiple color maps) the lowest map ID takes precedence.

```xml
<?xml version="1.0"?>
<AMF>
  <Object PrintID = "0" units = "mm">
    <Mesh>
      <ColorFile MapID="0">
        <File>Logo.bmp</File>
      </ColorFile>
      <Vertices>
        <Vertex VertexID="0">
          <VertexLocation x="0" y="1.332" z="3.715"/>
          <VertexMap UseMapID="0" MapXPixel="65"
              MapYPixel="87"/>
        </Vertex>
        <Vertex VertexID="1">
          <VertexLocation x="0" y="1.269" z="3.715"/>
          <VertexMap UseMapID="0" MapXPixel="64" MapYPixel
              ="87"/>
        </Vertex>
        <Vertex VertexID="2">
          <VertexLocation x="0" y="1.310" z="3.587"/>
          <VertexMap UseMapID="0" MapXPixel="32" MapYPixel
              ="10"/>
        </Vertex>
        . . .
      </Vertices>

      <Region FillMaterialID = "0">
        <Color R = "0" G = "0" B = "0.5"/>
        <Triangle V1 = "0" V2 = "1" V3 = "2"/>
        <Triangle V1 = "0" V2 = "1" V3 = "4"/>
        . . .
      </Region>
    </Mesh>
  </Object>
</AMF>
```


Figure A.5: Surface properties such as color and texture may be added with several additional tags. Individual regions or vertices may be colored, or a bitmap can be mapped onto the 3D surface.

CSG (Computational Solid Geometry) Example

In addition to defining regions by a mesh, there are cases in which it is desirable
to define regions via computational solid geometry, or a functional representa-
tion. From a complexity standpoint, it is not desirable to replicate all the features
of the advanced CAD engines such as ACIS in the file format. However, regions
may be defined using the $<$FRep$>$ tag as an equation involving the three vari-
able "X", "Y", and "Z", along with any number of unions, intersections, differ-
ences, and negations. By using binary operators "less than" and "greater than",
complex regions may defined. By convention, wherever the equation evaluates
to "true", geometry is present and wherever it is false, no geometry is present. If
color is to be added, separate continuous $<$RedEquation$>$, $<$GreenEquation$>$,
and $<$BlueEquation$>$ tags should be added. The additional CDATA text in the
file is a requirement of XML when storing arbitrary text, such as an equation.

```
<?xml version="1.0"?>
<AMF>
  <Object PrintID = "0" units = "mm">
    <FRep MaterialID = "0">
      <GeometryEquation>
        <![CDATA[X^2+Y^2+Z^2-4 <= 0 & Z >= 0]]>
      </GeometryEquation>
    </FRep>
  </Object>
</AMF>
```

Additional Capabilities and Extensions

In addition to the <Mesh> and the <FRep> tags, a region can also be defined by the <Voxels> tag. Within the <Voxels> tag, a <Lattice> tag sets up the dimensions and packing type of the voxels, and a <Structure> tag defines a three dimensional matrix of material IDs to define the geometry and materials. The <Voxels> tag can also be used for a material in the palette to define complex, repeating microstructures of multiple materials.

Materials within the palette may refer to external material library files using the <File> tag. This allows vendors to maintain a database of existing materials, including useful material data, and for an AMF file to specify a specific material to be built with. Also, objects can include a <tolerance> to define the build tolerances that are required.

A.3.3   Performance

One disadvantage of the XML format is that the human readability comes at the cost of file size. In order to compare the AMF file sizes to the standard STL files, a sample mesh geometry of a rook was created with 3680 triangles. This geometry was saved as both ASCII and binary STL files. As expected, the binary STL exhibited a much smaller file size, or about 24% of the ASCII STL version (Figure A.6). The XML text version of the AMF file was already 44% smaller than the ASCII STL file, and after applying standard compression routines, the AMF file was approximately 25% smaller than the binary STL. Compressing the binary STL yielded a file that was still 48% larger than the compressed AMF. Different types of compression routines may also be used for

File Size of STL vs AMF File Formats

(a)   (b)
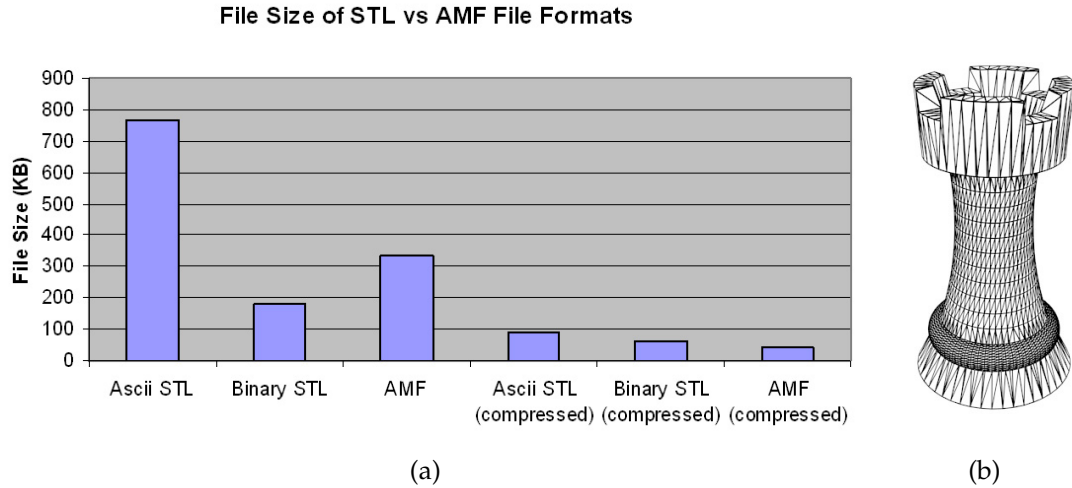
Figure A.6: The size of files generated for both STL and AMF formats are shown in (a) for a rook geometry with 3680 triangles (b). The ASCII readable AMF file is 44% smaller than the equivalent ASCII STL, and after compression the AMF is 240% smaller than the binary STL and 67% smaller than the similarly compressed binary STL.

the XML [132], including those in which elements can be efficiently extracted without decompressing the entire file.

A.4   Conclusion

Here a replacement for the STL file format is proposed for use in additive manufacturing applications. The shortcomings of the STL file are addressed, namely leaks, lack of multi-material support, and a lack of provisions for surface data. This is accomplished with a flexible, extensible XML-based format. This allows the file format to be extensible while maintaining compatibility with legacy applications. The AMF format is easily forwards and backwards compatible with STL files with a simple conversion, which will allow adoption driven by need

and not by mandate as additive manufacturing hardware becomes more versatile. Additionally, the AMF format is easy to understand and human readable (before standard compression routines) which will allow for easy debugging and adoption by software developers. Ultimately, the success of a file format depends on its adoption by both manufacturers and users, and its acceptance as a formal standard.

Up-to-date information and open source code for an implementation of a parser and viewer for AMF files is available at http://ccsl.mae.cornell.edu/AMF.

Major Contributions

Chapter 1: Introduction

- **Explored the theoretical differences in dimensionality of different rapid assembly processes.** Previous work explores the complexity of assembling two discrete parts by hand [59], but there are no other known explorations of top-down elemental assembly methods for large numbers of components.

Chapter 2: Voxel Shape and Design

- **Designated desirable requirements for voxels in a generic digital fabrication process.** Conceptual work to date has not addressed the specifics of the physical voxel design and simply referred to elements as featureless microbricks [61]. Previous efforts towards deterministically assembling physical voxels have assumed a predetermined voxel type such as spheres [126], "GIKS" [144], or other shapes [208] based solely on a specific voxel type, manufacturing process or application.

- **Designed and characterized 2.5D and 3D interlocking voxel shapes of all generic topologies and demonstrated 3D assembly of each.** Previous work has explored regular tiling 3D solids [38], but without consideration for interlocking geometry. Many isolated examples exist of interlocking space filling 3D shapes such as cubes [213] or toy bricks, but no previous comprehensive demonstration of interlocking 3D topologies exists.

- **Evaluated potential voxel designs using qualitative analysis to determine the most promising directions.** Other work in discrete elemental assembly methods has assumed a voxel geometry [144]. There are no other known evaluations or summaries of potential shapes for 3D voxel assembly.

Chapter 3: Voxel Fabrication Methods

- **Fabricated and demonstrated first known space-filling 3D assembled micro brick structures from multiple materials.** Previous work has demonstrated the process of multilayer microfabrication using SU-8 [121, 53, 202, 160] and separately the 3D etching of silicon [188, 119], but not for compatible interlocking shapes. Also, 3D non space-filling microscale SU-8 elements have been constructed and roboticly assembled [208, 209] for tissue scaffolding, but only one material has been demonstrated. Two dimensional multi-material structures have also been assembled in fluid from the bottom up using fluidic forces [31, 181], but these techniques do not scale well to large numbers of components.

Chapter 4: Serial Deposition Methods

- **Developed the first known serial digital fabricator in the form of the VoxJet research platform hardware and software.** 3D drop-on-demand systems are currently available that can fabricate analog freeform 3D structures [135, 18] from liquids and photopolymers using inkjet technology but are limited to a limited set of materials with specific rheological properties. Preliminary conceptual work has been done on an assembler for use

with interlocking "GIK" shaped voxels [144] but without demonstrated functionality.

- **Developed gravity-fed voxel deposition module for accurately depositing 1/16" spherical voxels in a 3D lattice at up to 10Hz.** The deposition of 1/16" spheres for making braille signs is widely used commercially [177], but these tools cannot be adapted to create 3D lattices due to necessary spring-loaded downward pressure that would destroy previously deposited layers. Assembly rates are also estimated to be a much slower 2-4Hz.

- **Demonstrated the first state-perfect error correction on the fabrication of a physical object.** Although current additive manufacturing processes use low level closed loop feedback for parameters like positioning and temperature, they are open loop with respect to the geometry they create [5]. Geometric feedback has been demonstrated for a continuous domain wax drop-on-demand system [35], but finite errors still remain due to the analog medium.

Chapter 5: Parallel Deposition Methods

- **Developed the first known parallel digital fabricator in the form of the VoxLayer research platform hardware and software.** Parallel additive manufacturing systems using maskless lithography [125, 21] have been demonstrated for fabricating 3D objects one layer at a time [176, 112, 24] from liquid photopolymer, but are limited to a single material at a time. Although passive self assembly has been used to orient and position many discrete components in parallel [10], this is a fundamentally different challenge than assembling arbitrary 2D arrays of components into a 3D object.

- **Demonstrated fundamental principles of parallel electrostatic manipulation of voxels.** Although well established principles of xerography have long been used to transfer 2D patterns of toner particles [157] using electrostatic forces, this principle has not been demonstrated for deterministically manipulating discrete components. Even ubiquitous laser printer technology attracts many non-aligned toner particle so to each discrete pixel on the 2D page.

- **Demonstrated first known parallel opto-fluidic selective manipulation of prefabricated millimeter scale elements in air.** Several optical methods have been developed to move microscale particles in fluid. Optical tweezers [96, 192, 37] are widely used and massively parallel methods have been developed that can move elements in parallel[27], but these are limited to particles no larger than tens of microns in a fluid carrier and are not easily adaptable to assembling 3D structures.

Chapter 6: Digital Objects: Fabricated, Post-Processed, and Recycled

- **Printed the first known multi-material digital objects, up to 22 kilovoxels at an average of three voxels per second.** Robotic pick and place systems have assembled as many as 420,000 spheres of one type into a lattice for artistic purposes [40], but at a rate of approximately of one element every 25 seconds. Likewise, LEGO^{TM} structures have been manually assembled with upwards of three million pieces of different types, but at the expense of tens of thousands of person-hours.

- **Developed post-processing techniques to obtain freeform stainless steel and acrylic digital structures.** Many post-processing methods exist for rapid manufacturing processes such as infusing with molten brass or wax

[5]. However, these require previous removal of support material which would be prohibitive in an unbound voxel manufacturing process.

- **Demonstrated the first known minimal-energy recycling of a multi-material metal and plastic object.** Conventional composite materials such as these can only be recycled by manually separating each material, then re-processing each raw material by energy-intensive melting and purification [168, 169, 155]. Interlocking brick toys such as LEGO^{TM} blocks can also be reused, but are manufactured of only a single material. Other examples of re-usable functional modules [213, 206] typically are limited to tens of modules.

Chapter 7: Precision Analysis of Digital Material with Imperfect Voxels

- **Demonstrated in theory, numerical simulation, and experiment the favorable error scaling properties of digital materials.** Summing uniform distributions is widely known to result in standard deviations that scale slower than linearly. There is no other known work quantifying this for the dimensional accuracy of interlocking and non-interlocking lattices of elements.

Chapter 8: Tunable Heterogeneous Material Properties

- **Performed detailed non-linear finite element analysis to characterize individual interactions between interlocking square tile voxels.** Interactions between other interlocking shapes have been characterized for assembly and disassembly [144], but without applications to large scale physical simulation.

- **Developed a non-linear relaxation simulator to efficiently predict the structural properties of structures composed of many individual tiles based on the characterized individual interactions.** Relaxation algorithms are widely used for everything from spatial localization [54] to simulating mechanisms [107]. Lattice-based relaxation algorithms have been used to simulate ion transfer [97] or polymers [19], but without consideration of the geometry and imprecisions of the individual elements.

- **Demonstrated in simulation the extreme mechanical tunability of digital materials based on factors such as voxel precision, material mixing, and voxel microstructure.** Single material lattices [170] and co-continuous composite materials [191] have been explored for the stiffness, strength, and energy dissipation of analog materials. Materials with negative Poisson's ratio have been demonstrated previously [98, 55], but have been limited to sparse foam-like structures unlike the nearly dense structures demonstrated here with digital materials.

Chapter 9: Simulators for Rapid Physical Prediction

- **Developed finite element software package optimized for simulating multi-material voxel structures.** Voxel finite element packages abound especially in biomedical engineering [1] and multi-material linear elastic models can be formulated in ANSYS [162]. These use general 8-noded brick elements with more degrees of freedom instead of the simpler beam elements used here.

- **Developed Voxelyze: the first known dynamic voxel simulation package that quantitatively models the statics and dynamics of multi-material voxel structures with large deformations.** Many soft body simulators are

used computer graphics [130, 141, 140], but are limited to a single material. Existing multi-material voxel simulators assume small linear displacements [162] and do not include nonlinear interactions such as friction and collision detection without additional levels of iteration.

- **Evaluated both simulators with respect to accepted theory in order to validate their static and dynamic performance.** The majority of soft-body simulators for computer graphics do not utilize physically based principles [150] or are not validated quantitatively with respect to static or dynamic performance [179, 45, 129, 50]. Finite element methods do provide quantitative results, but must be iterated to achieve non-linear deformation and other such effects [200].

Chapter 10: Manual Design Methods

- **Developed and open-sourced VoxCAD: a voxel editing and simulating program for general use.** Other voxel-modeling software is available for volumetric modeling [143] or voxel-by voxel editing [49], but none provide simulation abilities or are open source.

Chapter 11: Representations for Evolving Freeform Multi-material Shapes

- **Developed and demonstrated the discrete cosine transform (DCT) and Gaussian mixtures (GMX) generic representations for evolving freeform 3D multi-material shapes.** The discrete cosine transform is widely used for 2D image compression [25] using fixed size blocks, and keyed variants are used for video compression [52]. True 3D DCT encodings have been explored for video where the third dimension is time [99], but not for three

spatial dimensions as here. Gaussian mixtures have been used for a variety of purposes including mapping color to surfaces [123] in two dimensions and making abstract n-dimensional classification models [184, 186]. However, their use in specifying 3D shapes has not been previously explored.

- **Demonstrated the first known use of Compositional Pattern-Producing Networks (CPPNs) to generate 3D freeform shapes.** CPPNs [171] have been used to produces patterns for weighting neural networks [172] and creating images [159]. Simultaneous to the results presented here, CPPNs have also been used to indirectly grow structures [3], as opposed to the direct specification of volume utilized here.

Chapter 12: Evolving Static Structures and Mechanisms

- **Demonstrated the first known generic 3D freeform design automation for multi-material voxel based objects to meet high level multi-objective functional goals.** Iterative homogenization techniques [7] have been widely adapted for freeform 3D structural problems [12, 42], but are not suitable for high-level functional goals. Homogenization has also been explored in 2D for compliant structures [133], but without specified deformed shape or other high level objectives. Evolutionary algorithms have been used for problems such as single material truss optimization [180] and other more complex problems, but not for multi-material freeform domains. Design automation has been demonstrated in 3D for clothing [189], but only in parametric space, not a true freeform search space.

- **Verified non-intuitive predicted multi-material beam deflections in the physical world.** Although beams with varying heterogeneous material

288

distributions have been studied [2], the heterogeneity is random and does not significantly affect the deflected shape of the beams as here.

Chapter 13: Evolving Dynamic Soft Robots

- **Demonstrated the first known fully-soft autonomously designed locomoting robot functioning in the physical world.** All previous demonstrated soft robots are design manually. Some also utilize rigid components and discrete actuators [174, 175, 183]. Fully soft robots have been developed that makes use of jamming [127] or rubber pneumatic actuators [79], but are tethered to complex pressure distribution systems.

- **Demonstrated the utility of designing and fabricating soft robots with ideal volumetric actuation material.** Actuation involving the linear transport of fluid volume has been well explored [4], but not with isometric 3D volume changing. Volumetric actuation has been explored for a pneumatic soft robot, but without demonstrated locomotion [79].

- **Evaluated several 3D freeform representations for their suitability in design automation.** There are no other known direct comparisons of 3D freeform representations for genetic algorithms.

Appendix A: STL 2.0: A Proposal for a Universal Multi-material Additive Manufacturing File Format

- **Developed an extensible file format framework to meet the needs of the current and future capabilities of additive manufacturing processes.** The existing STL file [95, 89, 63] is not suitable for the needs of the industry. Other existing file formats such as VRML, STEP, SAT, OBJ, etc., do not

289

allow for interconnected regions of multiple materials or other features necessary for the additive manufacturing community.

- **Demonstrated the performance of this file format with respect to the current standard STL file format.** Although XML formats are typically viewed as oversize and inefficient for data [28], the performance is suitably fast for the needs of the additive manufacturing community.

Contributions of Others

This section lists where others directly contributed to the results presented in this thesis. This list does not include indirect contributions through prior work that is cited in the thesis.

Chapter 3: Voxel Fabrication Methods

- Collaborated with summer undergraduate researchers Joseph Heil, Britta Lundberg, and Joseph Miller to fabricate microbricks from multiple materials and develop the process.

BIBLIOGRAPHY

[1] Taiji Adachi, Ken ichi Tsubota, Yoshihiro Tomita, and Scott J. Hollister. Trabecular surface remodeling simulation for cancellous bone using microstructural voxel finite element models. *Journal of Biomechanical Engineering*, 123(5):403–409, 2001.

[2] Eli Altus. Statistical modeling of heterogeneous micro-beams. *International Journal of Solids and Structures*, 38(34-35):5915 – 5934, 2001.

[3] Joshua E. Auerbach and Josh C. Bongard. Evolving cppns to grow three-dimensional physical structures. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 627–634, New York, NY, USA, 2010. ACM.

[4] Yoseph Bar-Cohen. *Electroactive Polymer (EAP) Actuators as Artificial Muscles - Reality, Potential and Challenges*, volume PM98. SPIE Press, 2001.

[5] Joseph J Beaman, Harris L Marcus, David L Bourell, Joel W Barlow, Richard H Crawford, and Kevin P McAlea. *Solid Freeform Fabrication: A New Direction in Manufacturing*. Kluwer Academic Publishers, 1997.

[6] Holger Becker and Ulf Heim. Hot embossing as a method for the fabrication of polymer high aspect ratio structures. *Sensors and Actuators A: Physical*, 83(1-3):130 – 135, 2000.

[7] Martin Philip Bendsoe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Comput. Methods Appl. Mech. Eng.*, 71(2):197–224, 1988.

[8] J S Berry and I G Higginbotham. Electrostatic forces on a conducting sphere due to a charged, insulating plane. *Journal of Physics A: Mathematical and General*, 8(11):1842, 1975.

[9] A. Bertsch, H. Lorenz, and P. Renaud. 3d microfabrication by combining microstereolithography and thick resist uv lithography. *Sensors and Actuators A: Physical*, 73(1-2):14 – 23, 1999.

[10] K.-F. Bohringer, K. Goldberg, M. Cohn, R. Howe, and A. Pisano. Parallel microassembly with electrostatic force fields. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1204 –1211, may 1998.

[11] Josh Bongard. *Incremental Approaches to the Combined Evolution of a Robot's Body and Brain*. PhD thesis, University of Zurich, 2003.

[12] T. Borrvall and J. Petersson. Large-scale topology optimization in 3d using parallel computing. *Computer Methods in Applied Mechanics and Engineering*, 190:6201–6229, 2001.

[13] Mario Botsch, Andreas Wiratanaya, and Leif Kobbelt. Efficient high quality rendering of point sampled geometry. In *Proceedings of the 13th Eurographics workshop on Rendering*, EGRW '02, pages 53–64, Aire-la-Ville, Switzerland, 2002. Eurographics Association.

[14] David L. Bourell, Ming C. Leu, and David W. Rosen. *Roadmap for Additive Manufacturing: Identifying the Future of Freeform Processing*. The University of Texas at Austin Laboratory for Freeform Fabrication, Austin, TX, 2009.

[15] Ned Bowden, Andreas Terfort, Jeff Carbeck, and George M. Whitesides. Self-assembly of mesoscale objects into ordered two-dimensional arrays. *Science*, 276(5310):233–235, 1997.

[16] Tricia L. Breen, Joe Tien, Scott R. J. Oliver, Tanja Hadzic, and George M. Whitesides. Design and self-assembly of open, regular, 3d mesostructures. *Science*, 284(5416):948–951, 1999.

[17] Markus J. Buehler, Bernhard Bettig, and Gordon G. Parker. Topology optimization of smart structures using a homogenization approach. *Journal of Intelligent Material Systems and Structures*, 15(8):655–667, 2004.

[18] Paul Calvert. Inkjet printing for materials and devices. *Chemistry of Materials*, 13(10):3299–3305, 2001.

[19] I. Carmesin and Kurt Kremer. The bond fluctuation method: a new effective algorithm for the dynamics of polymers in all spatial dimensions. *Macromolecules*, 21(9):2819–2823, 1988.

[20] Arthur L. Chait. Custom designed microstructures using metamaterials, Jan 2010.

[21] Kin Foong Chan, Zhiqiang Feng, Ren Yang, Akihito Ishikawa, and Wenhui Mei. High-resolution maskless lithography. *J. Microlithogr. Microfabrication, Microsyst.*, 2:331, 2003.

[22] V. Chandru, S. Manohar, and C. E. Prakash. Voxel-based modeling for layered manufacturing. *Computer Graphics and Applications, IEEE*, 15(6):42–47, 1995.

[23] J H Chen and N C Seeman. Synthesis from dna of a molecule with the connectivity of a cube. *Nature*, 350(6319):631–633, 1991.

[24] Weiwu Chen, Soshu Kirihara, and Yoshinari Miyamoto. Fabrication of three-dimensional micro photonic crystals of resin-incorporating tio2 particles and their terahertz wave properties. *Journal of the American Ceramic Society*, 90(1):92–96, 2007.

[25] Wen-Hsiung Chen and W. Pratt. Scene adaptive coder. *Communications, IEEE Transactions on*, 32(3):225 – 232, March 1984.

[26] Joy Y. Cheng, Anne M. Mayes, and Caroline A. Ross. Nanostructure engineering by templated self-assembly of block copolymers. *Nat Mater*, 3(11):823–828, 2004.

[27] Pei Yu Chiou, Aaron T. Ohta, and Ming C. Wu. Massively parallel manipulation of single cells and microparticles using optical images. *Nature*, 436(7049):370–372, 2005.

[28] Kenneth Chiu, Tharaka Devadithya, Wei Lu, and Aleksander Slominski. A binary xml for scientific applications. *e-Science and Grid Computing, International Conference on*, 0:336–343, 2005.

[29] W. K. Chiu and S. T. Tan. Multiple material objects: from cad representation to data format for rapid prototyping. *Computer-Aided Design*, 32:707–717, 2000.

[30] C. K. Chua, S. M. Chou, and T. S. Wong. A study of the state-of-the-art rapid prototyping technologies. *The International Journal of Advanced Manufacturing Technology*, 14:146–152, 1998.

[31] Su Eun Chung, Wook Park, Sunghwan Shin, Seung Ah Lee, and Sunghoon Kwon. Guided and fluidic self-assembly of microstructures using railed microfluidic channels. *Nature Materials*, 7:581–587, 2008.

[32] Sung-Wook Chung, DavidS Ginger, MarkW Morales, Zhengfan Zhang, Venkat Chandrasekhar, MarkA Ratner, and ChadA Mirkin. Top-down

meets bottom-up: Dip-pen nanolithography and dna-directed assembly of nanoscale electrical circuits. *Small*, 1(1):64–69, 2005.

[33] Arkadiusz Chworos, Isil Severcan, Alexey Y. Koyfman, Patrick Weinkam, Emin Oroudjev, Helen G. Hansma, and Luc Jaeger. Building programmable jigsaw puzzles with rna. *Science*, 306(5704):2068–2072, 2004.

[34] R. L. Coble. Effects of particle-size distribution in initial-stage sintering. *Journal of the American Ceramic Society*, 56(9):461–466, 1973.

[35] Daniel Cohen and Hod Lipson. Geometric feedback control of discrete-deposition sff systems. *Rapid Prototyping Journal*, 16:377–393, 2010.

[36] S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *Visualization and Computer Graphics, IEEE Transactions on*, 5(1):62 –73, Jan-Mar 1999.

[37] Jennifer E. Curtis, Brian A. Koss, and David G. Grier. Dynamic holographic optical tweezers. *Optics Communications*, 207:169 – 175, 2002.

[38] O. Delgado Friedrichs and D. H. Huson. Tiling space by platonic solids, i. *Discrete and Computational Geometry*, 21(2):299–315, 1999.

[39] A. Diaz and R. Lipton. Optimal material layout for 3d elastic structures. *Structural and Multidisciplinary Optimization*, 13(1):60–64, 1997.

[40] Frederico Diaz. Geometric death frequency art installation, November 2010.

[41] Aleksandra Drizo and Joseph Pegna. Environmental impacts of rapid prototyping: an overview of research to date. *Rapid Prototyping Journal*, 12:64–71, 2006.

[42] J. Du and N. Olhoff. Topological optimization of continuum structures with design-dependent surface loading - part ii: algorithm and examples for 3d problems. *Structural and Multidisciplinary Optimization*, 27:166–177, 2004.

[43] Yanan Du, Edward Lo, Shamsher Ali, and Ali Khademhosseini. Directed assembly of cell-laden microgels for fabrication of 3d tissue constructs. *Proceedings of the National Academy of Sciences*, 105(28):9522–9527, 2008.

[44] C. B. Duke, J. Noolandi, and T. Thieret. The surface science of xerography. *Surface Science*, 500:1005–1023, 2002.

[45] Bernhard Eberhardt, Andreas Weber, and Wolfgang Strasser. A fast, flexible, particle-system model for cloth draping. *IEEE Computer Graphics and Applications*, 16:52–59, 1996.

[46] M. Elwenspoek and H. V. Jansen. *Silicon Micromachining*. Cambridge University Press, 2004.

[47] K E Evans. Tensile network microstructures exhibiting negative poisson's ratios. *Journal of Physics D: Applied Physics*, 22(12):1870, 1989.

[48] K. E. Evans and A. Alderson. Auxetic materials: Functional materials and structures from lateral thinking! *Advanced Materials*, 12(9):617–628, 2000.

[49] Everygraph. Voxel3d, voxel based modeling software. www.everygraph.com/voxel3d, 2011.

[50] P. Faloutsos, M. Van de Panne, and D. Terzopoulos. Dynamic free-form deformations for animation synthesis. *Visualization and Computer Graphics, IEEE Transactions on*, 3(3):201 –214, jul-sep 1997.

[51] P. Fernandes, J. M. Guedes, and H. Rodrigues. Topology optimization of three-dimensional linear elastic structures with a constraint on perimeter. *Computers and Structures*, 73(6):583–594, 1999.

[52] R. Forchheimer and T. Kronander. Image coding-from waveforms in animation. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(12):2008 –2023, dec 1989.

[53] I G Foulds and M Parameswaran. A planar self-sacrificial multilayer su-8-based mems process utilizing a uv-blocking layer for the creation of freely moving parts. *Journal of Micromechanics and Microengineering*, 16(10):2109, 2006.

[54] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localization and mapping. *Robotics, IEEE Transactions on*, 21(2):196 – 207, april 2005.

[55] E. A. Friis, R. S. Lakes, and J. B. Park. Negative poisson's ratio polymeric and metallic foams. *Journal of Materials Science*, 23:4406–4414, 1988.

[56] Tinsley A. Galyean and John F. Hughes. Sculpting: an interactive volumetric modeling technique. *SIGGRAPH Comput. Graph.*, 25:267–274, July 1991.

[57] Neil Gershenfeld. *FAB: The Coming Revolution on Your Desktop–From Personal Computers to Personal Fabrication*. Basic Books, New York, 2005.

[58] C. R. Gilman and S. J. Rock. The use of step to integrate design and solid freeform fabrication. In *Solid Freeform Fabrication Symposium*, pages 213–220, Austin, TX, 1995.

[59] Michael Goldwasser, Jean Latombe, and Rajeev Motwani. Complexity measures for assembly sequences. Technical report, Stanford University, Stanford, CA, USA, 1995.

[60] David H. Gracias, Joe Tien, Tricia L. Breen, Carey Hsu, and George M. Whitesides. Forming electrical networks in three dimensions by self-assembly. *Science*, 289(5482):1170–1172, 2000.

[61] Adolfo Gutierrez. Mems/mst fabrication technology based on micro-bricks: A strategy for industry growth. *MST News*, 1:4–8, 1999.

[62] R. Hague, I. Campbell, and P. Dickens. Implications on design of rapid manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 217(1):25–30, 2003.

[63] R. J. M. Hague and P. E. Reeves. *Rapid Prototyping, Tooling and Manufacturing*. Rapra Technology Ltd, 2000.

[64] Yu He, Tao Ye, Min Su, Chuan Zhang, Alexander E. Ribbe, Wen Jiang, and Chengde Mao. Hierarchical self-assembly of dna into symmetric supramolecular polyhedra. *Nature*, 452(7184):198–201, 2008.

[65] Jonathan Hiller. Voxcad open source project, 2011. http://sourceforge.net/projects/voxcad.

[66] Jonathan Hiller. Voxelyze open source project, 2011. http://sourceforge.net/projects/voxelyze.

[67] Lorenz M. Hilty. Electronic waste–an emerging risk? *Environmental Impact Assessment Review*, 25(5):431–435, 2005.

[68] Karl Heinz Hohne, Michael Bomans, Andreas Pommert, Martin Riemer, Carsten Schiers, Ulf Tiede, and Gunnar Wiebecke. 3d visualization of tomographic volume data using the generalized voxel model. *The Visual Computer*, 6:28–36, 1990.

[69] S.J. Hollister. Porous scaffold design for tissue engineering. *Nature Materials*, 4:518–524, jul 2005.

[70] N. Hopkinson, Y. Gao, and D. McAfee. Design for environment analyses applied to rapid manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 220(10):1363–1372, 2006.

[71] G. S. Hornby, H. Lipson, and J. B. Pollack. Evolution of generative design systems for modular physical robots. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 4146–4151 vol.4, 2001.

[72] G. S. Hornby, S. Takamura, T. Yamamoto, and M. Fujita. Autonomous evolution of dynamic gaits with two quadruped robots. *Robotics, IEEE Transactions on*, 21(3):402–410, 2005.

[73] L. L. Howell, A. Midha, and T. W. Norton. Evaluation of equivalent spring stiffness for use in a pseudo-rigid-body model of large-deflection compliant mechanisms. *Journal of Mechanical Design*, 118(1):126–131, 1996.

[74] Daniel Huang, Frank Liao, Steven Molesa, David Redinger, and Vivek Subramanian. Plastic-compatible low resistance printable gold nanoparticle conductors for flexible electronics. *Journal of The Electrochemical Society*, 150(7):G412–G417, 2003.

[75] Yu Huang, Xiangfeng Duan, Qingqiao Wei, and Charles M. Lieber. Directed assembly of one-dimensional nanostructures into functional networks. *Science*, 291(5504):630–633, 2001.

[76] Antti-Pekka Hynninen, Job H. J. Thijssen, Esther C. M. Vermolen, Marjolein Dijkstra, and Alfons van Blaaderen. Self-assembly route for photonic crystals with a bandgap in the visible region. *Nature Materials*, 6(3):202–205, 2007.

[77] Auke Jan Ijspeert and Jrme Kodjabachian. Evolution and development of a central pattern generator for the swimming of a lamprey. *Artificial Life*, 5(3):247–269, 1999.

[78] Olli Ikkala and Gerrit ten Brinke. Functional materials based on self-assembly of polymeric supramolecules. *Science*, 295(5564):2407–2409, 2002.

[79] Filip Ilievski, Aaron D. Mazzeo, Robert F. Shepherd, Xin Chen, and George M. Whitesides. Soft robotics for chemists. *Angewandte Chemie*, 2011.

[80] Gan G. K. Jacob, Chua Chee Kai, and Tong Mei. Development of a new rapid prototyping interface. *Computers in Industry*, 39(1):61–70, 1999.

[81] Paul Jacobs. *Rapid Prototyping and Manufacturing: Fundamentals of StereoLithography*. Society of Manufacturing Engineers, 1992.

[82] Karoly Jakab, Adrian Neagu, Vladimir Mironov, Roger R. Markwald, and Gabor Forgacs. Engineering biological structures of prescribed shape using self-assembling multicellular systems. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2864–2869, 2004.

[83] Mark J. Jakiela, Colin Chapman, James Duda, Adenike Adewuya, and Kazuhiro Saitou. Continuum structural topology design with genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4):339–356, 2000.

[84] Doug L. James and Dinesh K. Pai. Artdefo: accurate real time deformable objects. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 65–72, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[85] Helmut Jansen and Frank-Lothar Krause. *Life-Cycle Modelling for Innovative Products and Processes*. Chapman and Hall, Berlin, 1995.

[86] R. W. Jaszewski, H. Schift, J. Gobrecht, and P. Smith. Hot embossing in polymers as a direct way to pattern resist. *Microelectronic Engineering*, 41-42:575 – 578, 1998. International Conference on Micro- and Nanofarbication.

[87] LLC Jeffrey Kerr. Snap motors, 2011. http://jrkerr.com/snapmotors/.

[88] G. J. Jense. Voxel-based methods for cad. *Computer-Aided Design*, 21(8):528 – 533, 1989.

[89] Kevin K. Jurrens. Standards for the rapid prototyping industry. *Rapid Prototyping Journal*, 5:169–178, 1999.

[90] Vinay Kadekar, Weiya Fang, and Frank Liou. Deposition technologies for micromanufacturing: A review. *Journal of Manufacturing Science and Engineering*, 126(4):787–795, 2004.

[91] Couro Kane. Topological optimum design using genetic algorithms. *Control and Cybernetics*, 25(5):1059–1088, 1996.

[92] Couro Kane and Marc Schoenauer. Genetic operators for two-dimentional shape optimization. In *Artificial Evolution*, pages 355–369. 1996.

[93] Haruma Kawaguchi. Functional polymer microspheres. *Progress in Polymer Science*, 25(8):1171 – 1210, 2000.

[94] Rafal Kicinger, Tomasz Arciszewski, and Kenneth De Jong. Evolutionary computation and structural design: A survey of the state-of-the-art. *Computers and Structures*, 83(23-24):1943–1978, 2005.

[95] Vinod Kumar and Debasish Dutta. An assessment of data formats for layered manufacturing. *Advances in Engineering Software*, 28(3):151–164, 1997.

[96] SC Kuo and MP Sheetz. Force of single kinesin molecules measured with optical tweezers. *Science*, 260(5105):232–234, 1993.

[97] Maria G. Kurnikova, Rob D. Coalson, Peter Graf, and Abraham Nitzan. A lattice relaxation algorithm for three-dimensional poisson-nernst-planck theory with application to ion transport through the gramicidin a channel. *Biophysical Journal*, 76(2):642 – 656, 1999.

[98] R. Lakes. Foam structures with a negative poisson's ratio. *Science*, 235:1038–1040, feb 1987.

[99] M. C. Lee, Raymond K. W. Chan, and Donald A. Adjeroh. Quantization of 3d-dct coefficients and scan order for video compression. *Journal of Visual Communication and Image Representation*, 8(4):405 – 422, 1997.

[100] Jean-Marie Lehn. Toward self-organization and complex matter. *Science*, 295(5564):2400–2403, 2002.

[101] Marc Levoy. Efficient ray tracing of volume data. *ACM Trans. Graph.*, 9:245–261, July 1990.

[102] C.M. Liddell and C.J. Summers. Monodispersed zns dimers, trimers, and tetramers for lower symmetry photonic crystal lattices. *Advanced Materials*, 15(20):1715–1719, 2003.

[103] Li-Anne Liew, Wenge Zhang, Victor M. Bright, Linan An, Martin L. Dunn, and Rishi Raj. Fabrication of sicn ceramic mems using injectable polymer-precursor technique. *Sensors and Actuators A: Physical*, 89(1-2):64 – 70, 2001.

[104] F. Lin, H. S. Seah, Z. Wu, and D. Ma. Voxelization and fabrication of freeform models. *Virtual and Physical Prototyping*, 2(2):65 – 73, 2007.

[105] S. Y. Lin, J. G. Fleming, D. L. Hetherington, B. K. Smith, R. Biswas, K. M. Ho, M. M. Sigalas, W. Zubrzycki, S. R. Kurtz, and J. Bur. A three-dimensional photonic crystal operating at infrared wavelengths. *Nature*, 394:251–253, jul 1998.

[106] Yao Lin, Alexander Boker, Jinbo He, Kevin Sill, Hongqi Xiang, Clarissa Abetz, Xuefa Li, Jin Wang, Todd Emrick, Su Long, Qian Wang, Anna Balazs, and Thomas P. Russell. Self-directed self-assembly of nanoparticle/-copolymer mixtures. *Nature*, 434(7029):55–59, 2005.

[107] Hod Lipson. A relaxation method for simulating the kinematics of compound nonlinear mechanisms. *Journal of Mechanical Design*, 128(4):719–728, 2006.

[108] Hod Lipson and Jordan B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978, 2000.

[109] Ward A. Lopes and Heinrich M. Jaeger. Hierarchical self-assembly of metal nanostructures on diblock copolymer scaffolds. *Nature*, 414(6865):735–738, 2001.

[110] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21:163–169, August 1987.

[111] H. Lorenz, M. Despont, N. Fahrni, J. Brugger, P. Vettiger, and P. Renaud. High-aspect-ratio, ultrathick, negative-tone near-uv photoresist and its

applications for mems. *Sensors and Actuators A: Physical*, 64(1):33 – 39, 1998. Tenth IEEE International Workshop on Micro Electro Mechanical Systems.

[112] Yi Lu, Gazell Mapili, Gerry Suhali, Shaochen Chen, and Krishnendu Roy. A digital micro-mirror device-based system for the microfabrication of complex, spatially patterned tissue engineering scaffolds. *Journal of Biomedical Materials Research Part A*, 77A(2):396–405, 2006.

[113] Dan Luo, Kim Woodrow-Mumford, Nadya Belcheva, and W.Mark Saltzman. Controlled dna delivery systems. *Pharmaceutical Research*, 16:1300–1308, 1999.

[114] D. Ma, F. Lin, and C. K. Chua. Rapid prototyping applications in medicine. part 2: Stl file generation and case studies. *The International Journal of Advanced Manufacturing Technology*, 18:118–127, 2001.

[115] D. Ma, F. Lin, and C.K. Chua. Rapid prototyping applications in medicine. part 1: Nurbs-based volume modelling. *The International Journal of Advanced Manufacturing Technology*, 18:103–117, 2001.

[116] Evan Malone and Hod Lipson. Freeform fabrication of complete devices: Compact manufacturing for human and robotic exploration. In *AIAA Space 2006*, San Jose, CA, 2006.

[117] Evan Malone, Kian Rasa, Daniel Cohen, Todd Isaacson, Hilary Lashley, and Hod Lipson. Freeform fabrication of zinc-air batteries and electromechanical assemblies. *Rapid Prototyping Journal*, 10(1):58–69, 2004.

[118] Martti Mantyla, Dana Nau, and Jami Shah. Challenges in feature-based manufacturing research. *Commun. ACM*, 39(2):77–85, 1996.

[119] F. Marty, L. Rousseau, B. Saadany, B. Mercier, O. Franais, Y. Mita, and T. Bourouina. Advanced etching of silicon based on deep reactive ion etching for silicon high aspect ratio microstructures and three-dimensional micro- and nanostructures. *Microelectronics Journal*, 36(7):673 – 677, 2005. European Micro and Nano Systems - EMN 2004.

[120] S. O. Mast. Structure, movement, locomotion, and stimulation in amoeba. *Journal of Morphology*, 41(2):347–425, 1926.

[121] Alvaro Mata, Aaron J Fleischman, and Shuvo Roy. Fabrication of multi-

layer su-8 microstructures. *Journal of Micromechanics and Microengineering*, 16(2):276, 2006.

[122] Wil McCarthy. *Hacking Matter: Levitating Chairs, Quantum Mirages, and the Infinite Weirdness of Programmable Atoms*. Basic Books, New York, 2004.

[123] Stephen J. Mckenna, Shaogang Gong, and Yogesh Raja. Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition*, 31(12):1883 – 1892, 1998.

[124] S. McMains, J. Smith, and C. Sequin. The evolution of a layered manufacturing interchange format, Sept 29-Oct 2 2002.

[125] Rajesh Menon, Amil Patel, Dario Gil, and Henry I. Smith. Maskless lithography. *Materials Today*, 8(2):26 – 33, 2005.

[126] Vladimir Mironov, Richard P. Visconti, Vladimir Kasyanov, Gabor Forgacs, Christopher J. Drake, and Roger R. Markwald. Organ printing: Tissue spheroids as building blocks. *Biomaterials*, 30(12):2164–2174, 2009.

[127] Annan Mozeika, Erik Steltz, and Heinrich M. Jaeger. The first steps of a robot based on jamming skin enabled locomotion, October 11-15 2009.

[128] Lewis Mullen, Robin C. Stamp, Wesley K. Brooks, Eric Jones, and Christopher J. Sutcliffe. Selective laser melting: A regular unit cell approach for the manufacture of porous, titanium, bone in-growth constructs, suitable for orthopedic applications. *Journal of Biomedical Materials Research Part B: Applied Biomaterials*, 89B(2):325–334, 2009.

[129] Matthias Muller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '02, pages 49–54, New York, NY, USA, 2002. ACM.

[130] Andrew Nealen, Matthias Mller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836, 2006.

[131] Matthieu Nesme, Franois Faure, and Yohan Payan. Hierarchical multi-resolution finite element model for soft body simulation. In Matthias Harders and Gbor Szkely, editors, *Biomedical Simulation*, volume 4072 of

Lecture Notes in Computer Science, pages 40–47. Springer Berlin / Heidelberg, 2006.

[132] Wilfred Ng, Wai-Yeung Lam, and James Cheng. Comparative analysis of xml compression technologies. *World Wide Web*, 9(1):5–33, 2006.

[133] Shinji Nishiwaki, Mary I. Frecker, Seungjae Min, and Noboru Kikuchi. Topology optimization of compliant mechanisms using the homogenization method. *International Journal for Numerical Methods in Engineering*, 42(3):535–559, 1998.

[134] Stefano Nolfi and Dario Floreano. Synthesis of autonomous robots through evolution. *Trends in Cognitive Sciences*, 6(1):31–37, 2002.

[135] Objet. Objet geometries inc, 2010. http://www.objet.com.

[136] L. Patil, D. Dutta, A. D. Bhatt, K. Jurrens, K. Lyons, M. J. Pratt, and R. D. Sriram. A proposed standards-based approach for representing heterogeneous objects for layered manufacturing. *Rapid Prototyping Journal*, 8:134–146, 2002.

[137] Claus B. W. Pedersen, Thomas Buhl, and Ole Sigmund. Topology synthesis of large-displacement compliant mechanisms. *International Journal for Numerical Methods in Engineering*, 50(12):2683–2705, 2001.

[138] F. Pernkopf and D. Bouchaffra. Genetic-based em algorithm for learning gaussian mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1344–1348, 2005.

[139] D. T. Pham and R. S. Gault. A comparison of rapid prototyping technologies. *International Journal of Machine Tools and Manufacture*, 38(10-11):1257–1287, 1998.

[140] Bullet Physics. Bullet physics: Project summary, 2011. http://code.google.com/p/bullet/.

[141] PhysX. Physx features: Soft bodies, 2011. http://developer.nvidia.com/object/physx_features.html.

[142] Jordan B. Pollack, Hod Lipson, Gregory Hornby, and Pablo Funes. Three generations of automatically designed robots. *Artificial Life*, 7(3):215–223, 2001.

[143] Vladimir Popelnukh. 3d coat. www.3d-coat.com/voxel-sculpting/, 2011.

[144] George A. Popescu, Tushar Mahale, and Neil Gershenfeld. Digital material for digital printing, 2006.

[145] M. J. Pratt, A. D. Bhatt, D. Dutta, K. W. Lyons, L. Patil, and R. D. Sriram. Progress towards an international standard for data transfer in rapid prototyping and layered manufacturing. *Computer-Aided Design*, 34(14):1111–1121, 2002.

[146] Erik Puik and Leo van Moergestel. Agile multi-parallel micro manufacturing using a grid of equiplets. In Svetan Ratchev, editor, *Precision Assembly Technologies and Systems*, volume 315 of *IFIP Advances in Information and Communication Technology*, pages 271–282. Springer Boston, 2010.

[147] K. J. Quillin. Kinematic scaling of locomotion by hydrostatic animals: ontogeny of peristaltic crawling by the earthworm lumbricus terrestris. *J Exp Biol*, 202(6):661–674, 1999.

[148] John Rieffel, Frank Saunders, Shilpa Nadimpalli, Harvey Zhou, Soha Hassoun, Jason Rife, and Barry Trimmer. Evolving soft robotic locomotion in physx. In *GECCO '09*. ACM, 2009.

[149] John Rieffel, Barry Trimmer, and Hod Lipson. Mechanism as mind: What tensegrities and caterpillars can teach us about soft robotics. In *ALIFE XI*, pages 506–512, 2008.

[150] Alec R. Rivers and Doug L. James. Fastlsm: fast lattice shape matching for robust real-time deformation. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[151] Karen Taminger Robert, Robert A. Hafley, and Dennis L. Dicus. Solid freeform fabrication: An enabling technology for future space missions. In *2002 International Conference on Metal Powder Deposition for Rapid Manufacturing*, pages 51–60, San Antonio, TX, 2002.

[152] Stephen J. Rock and Michael J. Wozny. A flexible file format for solid freeform fabrication, 1991.

[153] Paul W. K. Rothemund. Folding dna to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.

[154] Daniela Rus and Marsette Vona. Crystalline robots: Self-reconfiguration with compressible unit modules. *Autonomous Robots*, 10(1):107–124, 2001.

[155] Sodhi M. S., Young J., and Knight W. A. Modelling material separation processes in bulk recycling. *International Journal of Production Research*, 37:2239–2252(14), 1999.

[156] A. Saxena and G. K. Ananthasuresh. Topology synthesis of compliant mechanisms for nonlinear force-deflection and curved path specifications. *Journal of Mechanical Design*, 123(1):33–42, 2001.

[157] R. M. Schaffert and C. D. Oughton. Xerography: A new principle of photography and graphic reproduction. *J. Opt. Soc. Am.*, 38(12):991–998, Dec 1948.

[158] Olaf Schenk and Klaus Gartner. Solving unsymmetric sparse systems of linear equations with pardiso. *Future Generation Computer Systems*, 20(3):475–487, 2004.

[159] Jimmy Secretan, Nicholas Beato, David B. D Ambrosio, Adelein Rodriguez, Adam Campbell, and Kenneth O. Stanley. Picbreeder: evolving pictures collaboratively online. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1759–1768, New York, NY, USA, 2008. ACM.

[160] V. Seidemann, J. Rabe, M. Feldmann, and S. Buttgenbach. Su8-micromechanical structures with in situ fabricated movable parts. *Microsystem Technologies*, 8:348–350, 2002.

[161] J. A. Sethian and A. Wiegmann. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics*, 163:489–528, 2000.

[162] Sandra J. Shefelbine, Ulrich Simon, Lutz Claes, Andreas Gold, Yankel Gabet, Itai Bab, Ralph Mller, and Peter Augat. Prediction of fracture callus mechanical properties using micro-ct images and voxel-based finite element analysis. *Bone*, 36(3):480 – 488, 2005.

[163] Mark S. Shephard and Marcel K. Georges. Automatic three-dimensional mesh generation by the finite octree technique. *International Journal for Numerical Methods in Engineering*, pages 709–749, 1991.

[164] O. Sigmund and S. Torquato. Design of smart composite materials using topology optimization. *Smart Materials and Structures*, 8:365–379, 1999.

[165] Karl Sims. Evolving virtual creatures. In *SIGGRAPH*, pages 15–22. ACM, July 24-29 1994.

[166] G. Skidmore, M. Ellis, A. Geisberger, K. Tsui, R. Saini, T. Huang, and J. Randall. Parallel assembly of microsystems using si micro electro mechanical systems. *Microelectronic Engineering*, 67-68:445 – 452, 2003. Proceedings of the 28th International Conference on Micro- and Nano-Engineering.

[167] C. Smith, R. Wootton, and K. Evans. Interpretation of experimental data for poisson's ratio of highly nonlinear materials. *Experimental Mechanics*, 39:356–362, 1999.

[168] M. Sodhi and W. A. Knight. Product design for disassembly and bulk recycling. *CIRP Annals - Manufacturing Technology*, 47(1):115–118, 1998.

[169] Manbir S. Sodhi and Bryan Reimer. Models for recycling electronics end-of-life products. *OR Spectrum*, 23:97–115, 2001.

[170] Jurgen Stampfl, Heinz E. Pettermann, and Mathias H. Luxner. Three-dimensional open cell structures: Evaluation and fabrication by additive manufacturing. In Fernando A. A. Lasagni and Andrs F. F. Lasagni, editors, *Fabrication and Characterization in the Micro-Nano Range*, volume 10 of *Advanced Structured Materials*, pages 95–117. Springer Berlin Heidelberg, 2011.

[171] Kenneth Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162, 2007.

[172] Kenneth O. Stanley. Generative and developmental systems. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, GECCO '09, pages 3335–3354, New York, NY, USA, 2009. ACM.

[173] I. Stroud and P. C. Xirouchakis. Stl and extensions. *Advances in Engineering Software*, 31(2):83–95, 2000.

[174] Y. Sugiyama and S. Hirai. Crawling and jumping of deformable soft

robot. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 3276 – 3281 vol.4, sept.-2 oct. 2004.

[175] Yuuta Sugiyama and Shinichi Hirai. Crawling and jumping by a deformable robot. *The International Journal of Robotics Research*, 25(5-6):603–620, 2006.

[176] C. Sun, N. Fang, D. M. Wu, and X. Zhang. Projection microstereolithography using digital micro-mirror dynamic mask. *Sensors and Actuators A: Physical*, 121(1):113–120, 2005.

[177] Jim Tapscott. Braille oz description. http://braille-oz.com.au/, 2011.

[178] B. Techaumnat and T. Takuma. Analysis of the electric field and force in an arrangement of a conducting sphere and a plane electrode with a dielectric barrier. *Dielectrics and Electrical Insulation, IEEE Transactions on*, 13(2):336 – 344, april 2006.

[179] M. Teschner, B. Heidelberger, M. Muller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Computer Graphics International, 2004. Proceedings*, pages 312 –319, june 2004.

[180] Vedat Togan and Ayse T. Daloglu. Optimization of 3d trusses with adaptive approach in genetic algorithms. *Engineering Structures*, 28(7):1019 – 1027, 2006.

[181] M. T. Tolley, A. Baisch, M. Krishnan, D. Erickson, and H. Lipson. Interfacing methods for fluidically-assembled microcomponents. In *IEEE International Conference on Micro Electro Mechanical Systems*, pages 1073–1076, Tucson AZ, 2008.

[182] M.T. Tolley, M. Kalontarov, J. Neubert, D. Erickson, and H. Lipson. Stochastic modular robotic systems: A study of fluidic assembly strategies. *Robotics, IEEE Transactions on*, 26(3):518 –530, june 2010.

[183] Barry A. Trimmer, Ann E. Takesian, Brian M. Sweet, Chris B. Rogers, Daniel C. Hake, and Daniel J. Rogers. Caterpillar locomotion: A new model for softbodied climbing and burrowing robots, 2006.

[184] J. J. Verbeek, N. Vlassis, and B. Krose. Efficient greedy learning of gaussian mixture models. *Neural Computation*, 15(2):469–485, 2003.

[185] Yurii A. Vlasov, Xiang-Zheng Bo, James C. Sturm, and David J. Norris. On-chip natural assembly of silicon photonic bandgap crystals. *Nature*, 414(6861):289–293, 2001.

[186] Ba-Ngu Vo and Wing-Kin Ma. The gaussian mixture probability hypothesis density filter. *Signal Processing, IEEE Transactions on*, 54(11):4091 –4104, nov. 2006.

[187] E. Volterra and E. C. Zachmanoglou. *Dynamics of Vibrations*. Charles E. Merrill Books, Inc., 1965.

[188] C. M. Waits, B. Morgan, M. Kastantin, and R. Ghodssi. Microfabrication of 3d silicon mems structures using gray-scale lithography and deep reactive ion etching. *Sensors and Actuators A: Physical*, 119(1):245 – 253, 2005.

[189] Charlie C.L. Wang, Yu Wang, and Matthew M.F. Yuen. Design automation for customized apparel products. *Computer-Aided Design*, 37(7):675 – 691, 2005.

[190] Dayang Wang and Helmuth Mhwald. Template-directed colloidal self-assembly the route to top-down nanochemical engineering. *Journal of Materials Chemistry*, 14:459–468, 2004.

[191] Lifeng Wang, Jacky Lau, Edwin L. Thomas, and Mary C. Boyce. Co-continuous composite materials for stiffness, strength, and energy dissipation. *Advanced Materials*, 23(13):1524–1529, 2011.

[192] M.D. Wang, H. Yin, R. Landick, J. Gelles, and S.M. Block. Stretching dna with optical tweezers. *Biophysical Journal*, 72(3):1335 – 1346, 1997.

[193] Michael Yu Wang, Xiaoming Wang, and Dongming Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1-2):227–246, 2003.

[194] Robert E. Webber. Ray tracing voxel data via biquadratic local surface interpolation. *The Visual Computer*, 6:8–15, 1990.

[195] George M. Whitesides and Bartosz Grzybowski. Self-assembly at all scales. *Science*, 295(5564):2418–2421, 2002.

[196] Jane Wilhelms and Allen Van Gelder. Multi-dimensional trees for controlled volume rendering and compression. In *Proceedings of the 1994 sym-*

posium on Volume visualization, VVS '94, pages 27–34, New York, NY, USA, 1994. ACM.

[197] Erik Winfree, Furong Liu, Lisa A. Wenzler, and Nadrian C. Seeman. Design and self-assembly of two-dimensional dna crystals. *Nature*, 394(6693):539–544, 1998.

[198] Terry Wohlers. *Wohlers Report 2008*. Wohlers Associates, Ft. Collins, CO, 2008.

[199] Hongkai Wu, Venkat R. Thalladi, Sue Whitesides, and George M. Whitesides. Using hierarchical self-assembly to form three-dimensional lattices of spheres. *Journal of the American Chemical Society*, 124(48):14495–14502, 2002.

[200] Xunlei Wu, Michael S. Downes, Tolga Goktekin, and Frank Tendick. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. *Computer Graphics Forum*, 2001.

[201] Cai Xiaoping and Li Jingyi. Electrostatic problem of the system consisting of a dielectric sphere and a semi-infinite dielectric. In *Industry Applications Society Annual Meeting, 1988., Conference Record of the 1988 IEEE*, pages 1733 –1734 vol.2, oct 1988.

[202] Baojian Xu, Yi-Kuen Lee, Qinghui Jin, Jianlong Zhao, and Chih-Ming Ho. Multilayer su-8 based microdispenser for microarray assay. *Sensors and Actuators A: Physical*, 132(2):714 – 725, 2006.

[203] Hao Yan, Sung Ha Park, Gleb Finkelstein, John H. Reif, and Thomas H. LaBean. Dna-templated self-assembly of protein arrays and highly conductive nanowires. *Science*, 301(5641):1882–1884, 2003.

[204] Luo Yanchun, Ji Zhiming, M. C. Leu, and R. Caudill. Environmental performance analysis of solid freedom fabrication processes. In *Electronics and the Environment, 1999. ISEE -1999. Proceedings of the 1999 IEEE International Symposium on*, pages 1–6, 1999.

[205] Peng Yao, Garrett J. Schneider, Binglin Miao, Janusz Murakowski, Dennis W. Prather, Eric D. Wetzel, and Daniel J. OBrien. Multilayer three-dimensional photolithography with traditional planar method. *Applied Physics Letters*, 85(17):3920 –3922, oct 2004.

[206] Mark Yim. Locomotion with a unit-modular reconfigurable robot. Technical report, Stanford University, Stanford, CA, USA, 1994.

[207] ZCorporation. Zedit software, 2009.

[208] Han Zhang, E. Burdet, A.N. Poo, and D.W. Hutmacher. Microassembly fabrication of tissue engineering scaffolds with customized design. *Automation Science and Engineering, IEEE Transactions on*, 5(3):446 –456, july 2008.

[209] Han Zhang, Dietmar W. Hutmacher, Franck Chollet, Aun Neow Poo, and Etienne Burdet. Microrobotics and mems-based fabrication techniques for scaffold-based tissue engineering. *Macromolecular Bioscience*, 5(6):477–489, 2005.

[210] Shuguang Zhang. Fabrication of novel biomaterials through molecular self-assembly. *Nat Biotech*, 21(10):1171–1178, 2003.

[211] X. Zhang, X. N. Jiang, and C. Sun. Micro-stereolithography of polymeric and ceramic microstructures. *Sensors and Actuators A: Physical*, 77(2):149 – 156, 1999.

[212] V. Zykov, J. Bongard, and H. Lipson. Evolving dynamic gaits on a physical robot, 2004.

[213] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson. Evolved and designed self-reproducing modular robotics. *Robotics, IEEE Transactions on*, 23(2):308 –319, april 2007.