

# Augmented Manual Fabrication Methods for 2D Tool Positioning and 3D Sculpting

by

Alec Rivers

B.A., Cornell University (2007)  
S.M., Massachusetts Institute of Technology (2010)

Submitted to the Department of  
Electrical Engineering and Computer Science  
in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy in Computer Science and Engineering  
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

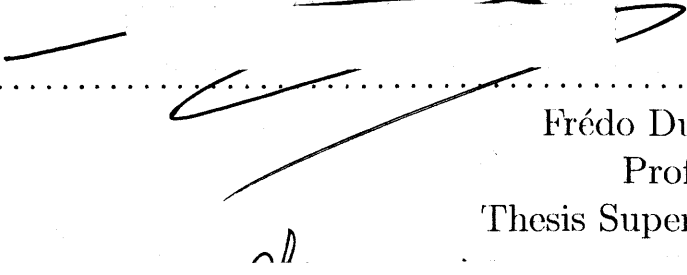
February 2013

© Massachusetts Institute of Technology 2013. All rights reserved.


Author .....

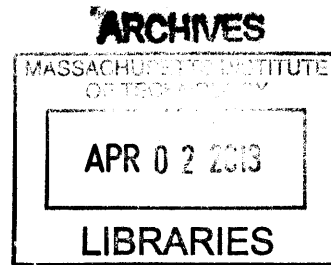
Department of  
Electrical Engineering and Computer Science  
November 30th, 2012

Certified by .....

  
Frédo Durand  
Professor  
Thesis Supervisor

Accepted by .....

  
Leslie Kolodziejski  
Chairman, Department Committee on Graduate Students





# Augmented Manual Fabrication Methods for 2D Tool Positioning and 3D Sculpting

by

Alec Rivers

B.A., Cornell University (2007)

S.M., Massachusetts Institute of Technology (2010)

Submitted to the Department of  
Electrical Engineering and Computer Science  
on November 30th, 2012, in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy in Computer Science and Engineering

## Abstract

Augmented manual fabrication involves using digital technology to assist a user engaged in a manual fabrication task. Methods in this space aim to combine the abilities of a human operator, such as motion planning and large-range mechanical manipulation, with technological capabilities that compensate for the operator's areas of weakness, such as precise 3D sensing, manipulation of complex shape data, and millimeter-scale actuation.

This thesis presents two new augmented manual fabrication methods. The first is a method for helping a sculptor create an object that precisely matches the shape of a digital 3D model. In this approach, a projector-camera pair is used to scan a sculpture in progress, and the resulting scan data is compared to the target 3D model. The system then computes the changes necessary to bring the physical sculpture closer to the target 3D shape, and projects guidance directly onto the sculpture that indicates where and how the sculpture should be changed, such as by adding or removing material. We describe multiple types of guidance that can be used to direct the sculptor, as well as several related applications of this technique.

The second method described in this thesis is a means of precisely positioning a hand-held tool on a sheet of material using a hybrid digital-manual approach. An operator is responsible for manually moving a frame containing the tool to the approximate neighborhood of the desired position. The device then detects the frame's position and uses digitally-controlled actuators to move the tool within the frame to the exact target position. By doing this in a real time feedback loop, a tool can be smoothly moved along a digitally-specified 2D path, allowing many types of digital fabrication

over an unlimited range using an inexpensive handheld tool.

Thesis Supervisor: Frédo Durand

Title: Professor

## Acknowledgments

I'd like to thank my co-authors on the papers in this thesis, Ilan Moyer (Position-Correcting Tools for 2D Digital Fabrication) and Andrew Adams (Sculpting by Numbers), as well as my thesis committee members Professors Wojciech Matusik and Bill Freeman. I would like to thank Professors Doug James of Cornell and Takeo Igarashi of the University of Tokyo for starting me on the path to academia. I'd like to thank the graphics groups of MIT and the University of Tokyo. I'd like to thank my family and friends for their encouragement and my cat Zeke for his moral support. Most of all I'd like to thank Professor Frédo Durand, my adviser, for being amazingly helpful and supportive.



# Contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Sculpting by Numbers . . . . .	14
1.2	Position-Correcting Tools . . . . .	15
<b>2</b>	<b>Background</b>	<b>17</b>
2.1	Design . . . . .	17
2.2	Guidance . . . . .	20
2.2.1	Spatially-Augmented Reality . . . . .	22
2.3	Actuation . . . . .	23
2.4	Hybrid methods in surgical applications . . . . .	25
<b>3</b>	<b>Sculpting by numbers</b>	<b>29</b>
3.1	Introduction . . . . .	29
3.2	Related work . . . . .	30
3.3	System design . . . . .	31
3.3.1	Scanning . . . . .	31
3.3.2	Guidance . . . . .	35
3.4	Applications . . . . .	36
3.4.1	Creating an object that matches a 3D model . . . . .	36
3.4.2	Sculpting from a physical example . . . . .	38
3.4.3	Stop-motion animation . . . . .	39
3.4.4	User testing and limitations . . . . .	40
3.5	Conclusion . . . . .	41

3.6	Hindsight . . . . .	42
<b>4</b>	<b>Position-correcting tools</b>	<b>47</b>
4.1	Introduction . . . . .	47
4.2	Related Work . . . . .	48
4.3	Localization . . . . .	49
4.3.1	High-contrast markers . . . . .	50
4.3.2	Image processing . . . . .	51
4.3.3	Building a map . . . . .	52
4.3.4	Localization using the map . . . . .	53
4.4	Actuation . . . . .	53
4.4.1	Actuation system . . . . .	54
4.4.2	Following a plan . . . . .	55
4.5	Using the tool . . . . .	59
4.5.1	User interface . . . . .	60
4.6	Results . . . . .	60
4.7	Conclusion and future work . . . . .	62
4.8	Hindsight . . . . .	62
<b>5</b>	<b>Conclusion</b>	<b>65</b>



# List of Figures

- 3-1 We assist users in creating physical objects that match digital 3D models. Given a target 3D model **(a)**, we project different forms of guidance onto a work in progress **(b)** that indicate how it must be deformed to match the target model. As the user follows this guidance, the physical object’s shape approaches that of the target **(c)**. With our system, unskilled users are able to produce accurate physical replicas of complex 3D models. Here, we recreate the Stanford bunny model (courtesy of the Stanford Computer Graphics Laboratory) out of polymer clay. . . . . 29
- 3-2 **Sculpting guidance:** A user is given feedback during sculpting by two forms of guidance projected onto the model, which they can toggle between. Guidance is computed by comparing the target depth map **(a)** to the scanned depth map of the current shape **(b)**. Depth guidance **(c)** illustrates the difference between the two maps and is helpful for matching the position of the surface in absolute 3D coordinates. Edge guidance **(d)** illustrates the second derivative of the target model (i.e. ridges and valleys), and is useful for recreating fine details once the model is close to the target shape. A smoothly-varying range of colors is used: in depth mode green indicates where the surface should be moved closer to the scanner and red farther, while in edge mode green indicates a positive second derivative (a ridge) and red negative (a valley). In either mode, blue indicates background, so any part of the material colored blue should be removed entirely. . . . . 32

3-3	The head of Michaelangelo’s David, in the original marble (photograph by Steve Hanna) <b>(a)</b> and in a 6” tall version made in modeling clay with our system <b>(b)</b> using data from the digital Michelangelo project [45]. The clay model was created by novice sculptors. . . . .	37
3-4	<b>Sculpting from a physical example:</b> With a simple extension, we can use a scan of an existing physical object as a target for sculpting. For example, without any prior cake-decorating experience, we remade this toy car as a cake. We are free to make a replica at a different scale and out of different materials: the car has been scaled-up and made out of madeira cake, frosting, marzipan, cookies, and candy. . . . .	38
3-5	<b>Stop-motion animation:</b> Our approach can be used to help create stop-motion animation. We demonstrate an end-to-end system, in which a 2.5D video (left and center columns) of a real person walking was captured with a Kinect and used to create a claymation sequence (right column). . . . .	39
3-6	<b>User testing results:</b> After 10 minutes of training and practice, users were asked to sculpt the Stanford bunny out of polymer clay both freehand (top row), and also using our system (bottom row). 30 minutes were allotted for each task. The odd columns first used the system and then sculpted freehand, while the even columns sculpted freehand and then used our system. For all users but the artistically-inclined column 3, the system greatly improved their ability to create a correctly-proportioned bunny. . . . .	41

4-1	<p><b>Overview: (a):</b> A position-correcting tool. The device consists of a frame and a tool (in this case a router) mounted within that frame. The frame is positioned manually by the user. A camera on the frame (top right in the figure) is used to determine the frame's location. The device can adjust the position of the tool within the frame to correct for error in the user's coarse positioning. <b>(b):</b> To follow a complex path, the user need only move the frame in a rough approximation of the path. In this example, the dotted blue line shows the path that the tool would take if its position were not adjusted; the black line is its actual path. <b>(c):</b> An example of a shape cut out of wood using this tool. . . . .</p>	47
4-2	<p><b>Map:</b> A scanned map with a plan registered to it. The red dotted line indicates a path that a user could conceivably follow to cut out the shape. . . . .</p>	50
4-3	<p><b>Markers:</b> A sequence of markers, with values 1000 to 1006, such as would be printed on a strip of tape. In our current implementation, markers are printed at a size of roughly 0.8" × 0.4". This is small relative to the area of the material the camera can see at once (roughly 8" × 6"). . . . .</p>	51
4-4	<p><b>Positioning linkage: (a):</b> Our linkage converts the rotary motion of two shafts (filled circles) into translation of a tool (cross) mounted on a sliding stage. This is achieved using eccentrics (shaded circles), which are circular disks rotating about the off-center shafts to produce linear displacement in fitted collars. To properly constrain the degrees of freedom of the stage, one eccentric is directly connected to the stage while the other is connected via an additional hinge. <b>(b):</b> A photo of the actual linkage. . . . .</p>	54

4-5	<b>Freeform motion paths:</b> Each box illustrates a case in which a different path (described below) is used, due to the higher-preference paths being infeasible. In each box, the cross is the current position of the tool, the circle is the range of the positioning system, the green dot is the target position, and the green path is the selected path. . . . .	56
4-6	<b>User interface:</b> This display shows the shapes of the plan (blue polygons); the path that the tool is actually following, which is those shapes offset by the tool's radius (dotted line); the tool's current position (cross); the area cut by the tool (shaded area); and the range of the tool's position correction (black circle). As long as the user keeps the tool path within the correction range, the tool should be able to follow the plan. . . . .	57
4-7	<b>Results:</b> Several shapes cut out from various materials (clockwise from top left: hardwood, hardwood, paperboard, polycarbonate plastic, plywood, and sheet metal). . . . .	59
4-8	<b>Range:</b> A full-size vinyl cutout of a human silhouette (5'6" tall), with original. . . . .	61
4-9	<b>Fine details:</b> With a vinyl cutter, the resolution of features is not limited by the width of the bit. Here, we show a 6"-wide sticker with fine details. . . . .	61
4-10	<b>Accuracy:</b> A scan of a plotted pattern (6" wide) shown with the design that was used to create it (red). The inset shows an expansion of the area of worst error, with the addition of the line fit to the scan for analysis (green). Note that even here the error is only on the order of the width of the pen. . . . .	63

# Chapter 1

## Introduction

Digital fabrication devices such as CNC routers and 3D printers have made it possible to fully automatically manufacture objects from a digital plan. They are incredibly precise, and can machine extremely complex shapes as easily as simple ones. However, they remain large and expensive. Much of this cost goes to pay for mechanical components that reproduce physical manipulations, such as large-range positioning, that a human is able to perform as a matter of course. This thesis explores the space of augmented manual fabrication: methods and devices to help a user manually create a physical object. Augmented manual fabrication seeks to take the advantages of a computer's precision and use them to augment, rather than replace, manual fabrication, allowing hybrid methods of creating things that combine the accuracy of an automatic method with the flexibility and availability of manual crafting.

The key advantage of such a hybrid approach is that it leverages, rather than discards, the mechanical ability of a human operator. As a physical device, the human body is extremely capable. Most people find it easy to manipulate tools in complex and finely-controlled ways, over an effectively unlimited working volume. Humans are also excellent at planning physical motions to accomplish a task. However, they cannot sense absolute 3D positions, cannot remember exact 3D shapes, and the precision with which they manipulate objects, while high, is not perfect.

Digital devices, meanwhile, have an almost exactly complementary set of strengths and weaknesses. Even the most advanced and expensive robots have difficulty moving in a non-controlled environment and grasping and manipulating tools and materials, but computers and digital scanners are trivially capable of manipulating large amounts of shape data, accurately scanning a 3D scene, and driving actuators to perform incredibly fine adjustments.

There are therefore multiple fruitful areas for computers to aid humans in manually crafting objects. A computer’s ability to manipulate large amounts of shape data can be used to help design the object to be fabricated and plan the fabrication steps. A scanner’s ability to accurately perceive a scene can be used to help guide a user during fabrication to make adjustments in exactly the right 3D location. And a digitally-driven actuator can be used to augment or finely adjust the mechanical actuation performed by the user to improve accuracy.

In this thesis, we present two approaches that lie within this space. In the first, sculpting by numbers, we present a method that augments only with guidance, to help a user accurately create a 3D shape that matches a target digital model. In the second, position-correcting tools, we present a method that augments with both guidance and partial actuation, to help a user accurately guide a tool along a 2D path for digital fabrication.

## 1.1 Sculpting by Numbers

Most people find it challenging to sculpt, carve or manually form a precise shape. We argue that this is not usually because they lack manual dexterity – the average person is able to perform very precise manipulations – but rather because they lack precise 3D information, and cannot figure out what needs to be done to modify a work in progress in order to reach a goal shape. An analogy can be made to the task of reproducing a 2D painting: when given outlines that need only be filled in, as in a child’s coloring book or a paint-by-numbers kit, even an unskilled user can

accurately reproduce a complex painting; the challenge lies not in placing paint on the canvas but in knowing where to place it. We therefore present Sculpting by Numbers, a method to provide analogous guidance for the case of creating 3D objects, which assists a user in making an object that precisely matches the shape of a target 3D model.

We employ a spatially-augmented reality approach (see e.g. Raskar et al. [60] or Bimber and Raskar [7] for an overview of spatially-augmented reality), in which visual feedback illustrates the discrepancy between a work in progress and a target 3D shape. This approach was first proposed by Skeels and Rehg [69]. In this approach, a projector-camera pair is used to scan the object being created using structured light. The scanned shape is compared with the target 3D model, and the projector then annotates the object with colors that indicate how the object ought to be changed to match the target. The user follows this guidance to adjust the object and rescans when necessary, bringing the object closer to the target shape over time.

Our proposed method provides guidance that illustrates depth disparities between the current work and the target, similar to the approach of Skeels and Rehg [69], as well as an additional form of guidance, which we call edge guidance, which aids in reproducing high-frequency surface details. We demonstrate an application to reproducing an existing physical object at a different scale or using different materials. We further show how a scene can be repeatedly deformed to match a sequence of target 3D models for the purpose of stop-motion animation. Starting from a depth video recorded with a Kinect, we produce claymation with physically correct dynamics. Finally, we present results of user testing with novice sculptors.

## 1.2 Position-Correcting Tools

Guiding a user performing manual fabrication can greatly improve accuracy relative to unguided fabrication, but not beyond the user’s physical ability to maneuver a tool. For many functional (as opposed to aesthetic) applications, a level of accuracy higher

than can be achieved manually, even with guidance, is often required. Typically, the solution is to handle actuation fully automatically. However, automatic positioning mechanisms, as described above, are cumbersome and expensive, especially for larger-range devices.

We aim to reduce the cost of digital fabrication for the domain of 2D shapes while simultaneously removing constraints on range. Our central idea is to use a hybrid approach to positioning where a human provides range while a tool with a cheap short-range position-adjustment enables precision. Given an input 2D digital plan such as the outline of a shape, the user manually moves a frame containing a tool in a rough approximation of the desired plan. The frame tracks its location and can adjust the position of the tool within the frame over a small range to correct the human's coarse positioning, keeping the tool exactly on the plan. A variety of tools can be positioned in this manner, including but not limited to a router (which spins a sharp bit to cut through wood, plastic, or sheet metal in an omnidirectional manner) to cut shapes, a vinyl cutter to make signs, and a pen to plot designs.

In this approach, the core challenges are localization (determining the current position of the tool) and actuation (correcting the tool's position). For localization, we use computer vision and special markers placed on the material. For actuation, we present a two-axis linkage that can adjust the position of the tool within the frame. We also describe an interface for guiding the user using a screen on the frame, which illustrates the tool's current position relative to the plan. We demonstrate an example of a device, measuring roughly 13" x 10" x 9", that uses our approach and can be fitted with a router or a vinyl cutter, and show results that can be achieved with these tools when they are positioned with our computer-augmented approach.



# Chapter 2

## Background

Digital augmentation of manual fabrication already occurs in a variety of contexts. Here, we place into context three broad areas in which augmentation can occur: design, in which software tools can help the user create, analyze and fully specify the plan for the object to be fabricated; guidance, in which feedback or instructions presented during the fabrication can help the user execute the plan; and actuation, in which some or all of the actual physical manipulation can be done by a machine.

### 2.1 Design

Designing an object to be manufactured involves many steps. The appearance and functionality of the object must be conceived, and the mechanical parts that constitute the object must be designed. The parts' designs must be made such that they are amenable to the type of manufacturing (e.g. injection molding) that will be used, and often need to be analyzed to determine where stresses and fractures could occur. Finally, the parts must be fully specified with allowable tolerances to enable their manufacture.

In the digital age, much of this process has moved to the computer. CAD modeling programs such as SolidWorks, Pro/Engineer, and CATIA [71, 59, 10] provide end-

to-end systems for designing complex mechanical objects with moving parts. These systems are explicitly geared towards designing objects that will ultimately be manufactured (as opposed to used only in a digital context, such as for animation), and as such directly incorporate mechanical engineering tools. For example, a CAD system will typically include built-in physics simulators, both to predict the gross-scale mechanical motions of the parts, and to perform stress analysis such as with finite-element methods [14].

These tools, as well as more general-purpose 3D modelers, remain difficult and time-consuming to use, due largely to the difficulty of specifying a complex 3D shape. A large variety of approaches have been proposed to make this task easier. On the hardware side, 3D input devices [8] and haptic interfaces that make it possible to “feel” a shape as it is being modeled [25] seek to recreate some of the ease of making a shape with one’s hands. On the software side, a large variety of interfaces have been proposed to make 3D modeling more intuitive and natural. Zeleznik et al. [81] proposed a modeling system in which the input was 2D sketches drawn into a 3D view. These sketched shapes were interpreted using techniques from gesture recognition and 3D reconstruction, and enabled the user to perform modeling operations quickly and easily. Igarashi et al. [33] presented a sketch-based system for creating organic 3D models in which modeling operations were even higher-level, and allowed the user to create a 3D shape almost as if sketching a 2D drawing. From these foundational works came a large body of sketch-based modeling approaches for both specifying both mechanical shapes [81, 20, 36, 61, 70] and organic shapes [33, 32, 37, 55, 64, 38, 54].

Recently, much work has focused on developing software methods that integrate fabrication considerations directly into the design process. Some of these methods dovetail with specific fabrication technologies such as 3D printers, and seek to ensure successful fabrication of objects with desired properties. Bickel et al. [6] propose a method for automatically generating a 3D-printable pattern of multiple materials such that the resulting printed object will match desired deformation characteristics. Alexa et al. [4] demonstrate how to generate 3D-printable relief surfaces that match user-

specified appearances under directed lighting. Hasan et al. [24] and Dong et al. [19] demonstrate systems for fabricating objects with desired subsurface scattering. Stava et al. [72] propose an automatic method for increasing the structural stability of 3D printable objects by analyzing a 3D model to be printed and selectively hollowing, thickening, and adding struts to elements to increase stability.

Other methods seek to design or reformulate existing designs into a format such that they can be fabricated using alternative methods that are less general but have desirable characteristics such as ease or low cost. Two recent works propose methods to generate models of sliding planar slices that approximate target 3D models [28, 51]. Lau et al. [43] propose a method to convert 3D furniture models into fabricatable planar parts.

A recent trend in computer graphics methods for fabrication has been to design interfaces in which fabrication constraints are included in the modeling interface, so that objects designed with the system are guaranteed to have straightforward fabrication. A prototypical example is an interface by Mori and Igarashi [53] for designing stuffed animals. In this task, the stuffed animal has a 3D shape that is determined by a collection of 2D pieces of cloth and the interactions between them and the stuffing. An interface in which the user modeled the 3D shape could easily result in 3D surfaces that were difficult or impossible to create out of cloth pieces and stuffing, while a modeling system in which the user directly designed the shapes of the cloth pieces would require that the user understand the complex mapping between the 2D cloth shapes and the 3D shape that will result. They therefore propose an interface in which the 2D cloth shapes and the 3D model that will result from these shapes (determined by a rapid physics simulation) are displayed simultaneously, and the user can perform editing operations on either and see the results on the other. In this way the user can easily achieve a desired 3D shape while ensuring that it remains fabricatable in practice.

Other works have proposed interfaces for fabrication-conscious design of other types of objects. Saul et al. [63] proposed a method for designing chairs using a sketch-based

that limits input to shapes that can be easily fabricated out of planar parts. Kilian et al. [40] propose a method for designing shapes that can be produced by folding a single sheet of material. Lau et al. [42] propose an augmented reality interface for designing 3D shapes in which real-world objects can be incorporated as primitives.

## 2.2 Guidance

Once the object to be manufactured has been fully specified, it must be made. If the construction is being done manually, the builder will need guidance to successfully execute the plan.

The simplest forms of guidance in manufacturing involve simply making the specification as clear and easy to understand as possible. Traditionally, this was achieved by technical draftsmen creating blueprints, using a long-developed visual language that maximizes the clarity of specification for manufactured parts. More recently, the field of computer visualization has sought to encode visual design principles in algorithms that can automatically generate intuitive static or interactive specifications. Automatic visualization techniques have been applied to a wide variety of types of data, ranging from scientific simulation data to medical scans (See Viola et al. [77] for a survey). Visualization specifically of mechanical assemblies and fabrication instructions has also received attention (see e.g. [22, 65]). Agrawala et al. [2] attempted to empirically determine cognitive design principles for designing effective assembly instructions, and then developed an automatic system that implemented those principles, combining the task of generating the plan with the challenge of visualizing it. This work very directly augments manual fabrication with automatic guidance. Li et al. proposed related approaches to automatically generate interactive cutaway visualizations of complex shapes [48] and exploded view diagrams of mechanical assemblies [47].

The visualization of the specification may also take place during fabrication itself. A large body of work has been developed on using augmented reality [5] to provide guid-

ance during the performance of procedural tasks such as assembly. These typically provide guidance to a user using a head-mounted display that overlays see-through instructions in the context of the user's field of view, an intuitive format that makes it easier to understand the 3D spatial relationships of complex assemblies. In an early work, Webster et al. [78] proposed using augmented reality for assembly, inspection, and renovation, providing features such as displaying otherwise-invisible information such as the location of rebar in a concrete wall, and guiding the user through complex multi-stage assembly processes. Caudell and Mizell [11] proposed applications to wiring layout and composite cloth layup in aircraft manufacture.

Guidance presented to the user during fabrication need not display only information collected in advance such as the plan for the object. Rather, the user can also be guided based on data being collected during fabrication, potentially derived from changing properties of the scene. In a rudimentary form, any mechanical sensor, from a level to a stud-finder, can be thought of as a device for augmenting fabrication using guidance based on data that the user cannot detect unaided. Our augmented reality approach for sculpting uses a more advanced structured light scanner to collect data about the scene that is used to determine guidance that aids the user.

Recent augmented reality systems also display guidance based on the changing properties of the scene itself. A prototypical system in this area is the ARMAR system of Henderson and Feiner [26, 27]. They demonstrate an augmented reality system to aid in the task of the assembly of a motor in the course of maintenance. They track multiple parts of the motor as well as the user's head, and generate a see-through interface on a head-mounted display that describes the current task to be performed (such as aligning two parts or tightening a bolt), shows the location of the salient components, illustrates the motion required of the user, and provides real-time feedback as the parts are adjusted. By tracking the parts in real time and providing feedback to the user on their positions, the user can be helped to make precise adjustments.

### 2.2.1 Spatially-Augmented Reality

While many augmented reality devices rely on a head-mounted display to overlay information on the user's view of the scene, another approach for displaying data in context is spatially-augmented reality (see e.g. Raskar et al. [60] or Bimber and Raskar [7]). In SAR approaches, a projector is used to directly annotate the scene itself with an image or interactive interface. These approaches have been used, for example, to turn virtually any flat surface into a monitor by projecting video onto it, or to project an animated face onto a static robot head.

The primary advantage of SAR approaches is that they eliminate the need for the user or users to wear special equipment such as head-mounted displays. However, this approach is also less flexible than head-mounted displays. Virtual objects cannot be created in mid-air, as they can with a head-mounted display, and instead must lie on the surfaces of the physical scene. Furthermore, not all surfaces will work well: certain surfaces such as reflective metals will not show the projected image clearly. Finally, irregular surfaces will cause the projected image to appear warped.

The problem of irregular-shaped surfaces can be largely resolved by scanning the scene using a calibrated camera in conjunction with the projector to perform structured-light scanning. With the resulting information about the shape of the scene, the desired image can be pre-warped such that it appears undistorted when projected onto the known irregular surface.

The augmented sculpting approach presented in this thesis is an example of a spatially-augmented reality method. In this application, the scene must in any case be scanned by a structured-light scanner to determine the feedback that will be shown to the sculptor, so presenting the feedback using the same hardware is economical. Furthermore, any surface on which projected images will not be easily visible, such as glossy surfaces, will also be difficult or impossible to scan using structured light, meaning that the necessary information cannot be gathered in the first place and the limitation of the display is a moot issue.

## 2.3 Actuation

Once the part has been designed, and the user is fully aware of how the design must be executed, the object still must be physically made. Devices to help apply the necessary mechanical actuation are, of course, the oldest form of augmented manual fabrication, as every handheld tool is an example of physical augmentation.

The advent of digital fabrication technologies, such as 3D printers, laser cutters, and CNC machines, has dramatically expanded the range of ways that a computer can aid in the mechanical aspects of fabrication. CNC, or computer numerical control, technology simply refers to machine tools that can be controlled directly by a computer. This technology was originally applied to milling machines for the purpose of cutting complex shapes out of metal for aircraft parts. Subsequent development has seen numerical control applied to a much wider range of tools for a much wider range of applications. With these devices, it has become possible to fully automatically produce 2D and 3D objects.

Digital fabrication devices can be broadly categorized into subtractive and additive approaches. Subtractive devices start with a block of material larger than the target target shape, and automatically maneuver a tool, such as a router, water-jet cutter, or lathe, to selectively remove material, leaving an object matching the target shape. By contrast, additive devices, such as 3D printers, build up an object by depositing (or in some cases selectively hardening) material in precise shapes, typically forming one thin layer at a time and stacking these layers together to form a desired 3D shape.

Digital fabrication devices are extremely precise, can be quite fast, and have made it possible to easily manufacture very complex shapes directly from a digital plan. These strengths have made them extremely appealing in certain industrial contexts. They have revolutionized low-volume manufacturing by effectively removing the tooling costs (such as creating a die or mold) that can otherwise dominate production costs for short runs, and further simplify low-volume manufacturing by working directly from a CAD specification. This has made them attractive for tasks such as

rapid prototyping and producing one-off objects such as scientific apparatuses or replacement parts. They are also used in some high-volume assembly cases where the ability to quickly change the design is essential, such as certain consumer electronics devices. Finally, they are used in some contexts where the ability to machine a shape out of a single block of material, without the artifacts and difficulties imposed by casting, is necessary.

The current designs of CNC devices, however, have drawbacks that have limited wider adoption. CNC devices are typically quite expensive, though recent efforts in the DIY community have made low-priced entry-level tools available [30, 39, 49, 21, 66]. They also typically must surround the material being worked on, meaning that a given device can only produce objects smaller than itself, limiting the working volume. They also suffer from fabrication artifacts, such as “stepping” artifacts in objects generated by 3D printers that reveal the different layers out of which the object was made. Finally, they are limited in the range of materials they can work with, especially for additive devices, and multi-material fabrication is particularly challenging.

A wide range of additional applications could become viable were these restrictions to be eliminated. Most immediately, there is a large class of potential consumers of digital fabrication technologies that cannot afford the cost or space requirements of a digital fabrication device. These include small-scale industries but also individual consumers. A large amount of enthusiasm for CNC devices has recently been generated by the DIY / “maker” communities, who see uses for CNC technology in home improvement, art, and personalized consumer goods. Current industrial consumers of CNC technologies could also see additional applications and improved utility if cost and / or space requirements were reduced. CNC devices are also typically not portable, meaning the job must be brought to the machine; were it possible to do the reverse, applications such as in-situ repair or renovation would become possible.

Many of the drawbacks of CNC devices arise from the nature of the mechanical systems used to position the tool. The most typical system is a computer-controlled 3-axis stage or gantry. As described, such a system can only move a tool within its



own volume, meaning that to work on a large object, a large device is required, which imposes space requirements. Furthermore, as the stage gets larger, it becomes increasingly difficult (and consequently expensive) to maintain accurate positioning over the entire range. As an illustration, a 2'  $\times$  1.5' ShopBot CNC mill costs approximately \$6,000, while a 5'  $\times$  8' ShopBot mill costs over \$20,000 [68].

This thesis presents an alternative, hybrid method for tool positioning for digital fabrication, in which the user provides large-scale, low-accuracy positioning of a tool, and the tool is digitally actuated over only a small range to correct that positioning. This type of hybrid approach to actuation is comparatively less common, but aims to address the limitations of both manual crafting and current digital fabrication technologies by combining elements of each in a hybrid approach.

## 2.4 Hybrid methods in surgical applications

Hybrid manual-automatic methods have been particularly well developed in their applications to surgery. A variety of systems have been developed that range from providing only guidance and sensing to partially or fully automating the mechanical act of surgery.

Methods that provide only guidance typically involve registering and integrating sets of scans of a patient, registering these scans onto the patient during surgery, and presenting the scanned data to the surgeon during the operation. Each of these represents a research field in its own right, and requires a combination of techniques, including advanced image registration methods to integrate data from completely different sensor modalities; 3D scanning technology to detect the shape of the patient's skin during surgery to provide a reference onto which data can be registered; and augmented-reality methods that display the information to the surgeon in an intuitive manner, such as on a head-mounted display. An overview of the field in general can be found in Delingette et al. [16].

Of closer relevance to the task of digitally-augmented manual fabrication are the variety of robotic methods that have been developed for assisting the surgeon with the mechanical task of performing the surgery itself. The conceptually simplest form of robot-assisted surgery is telesurgery, which was an outgrowth of tele-control for robots such as in the space program [29]. In this approach, the surgeon uses a remote console to control a robot (typically consisting of one or more robotic arms) to perform the surgery. One clear advantage of this is that the surgeon need not be colocated with the patient, though in practice this application has been dogged by latency issues. Other key advantages are that the motions of the surgeon can be recreated at a smaller scale, e.g. for microsurgery [41], or the robotic arms performing the surgery may be smaller than a surgeon's hands, allowing for smaller incisions to be made into the patient [35].

On the other end of the scale of automation is supervisory-controlled systems, in which the robot performs the surgery autonomously (though with human supervision). In these applications, a surgeon carefully plans the motions to be performed in advance, but the actual implementation is entirely automatic. This type of approach is particularly suited to applications where the surgery to be performed is straightforward conceptually but difficult mechanically. For example, a robot may be used to cut away a precise half sphere into a bone to make a site for a mechanical joint to be placed.

Finally, there exist approaches in which the tool is controlled by both the human and a robot, called shared-control robotic surgery systems. Such a system typically involves motions being planned and executed by a human, but modulated by a machine. For example, in the case of a surgery performed near critical tissues, the surgeon may mark certain regions of the 3D working volume as forbidden; then, when manipulating a tool, a robotic component will prevent the tool from entering those regions [50]. Or, the robot may act to smooth out the motions of the surgeon, to eliminate tremors [74].

These methods all provide insight into potential areas for human-machine collabora-

tion in physical tasks. Researchers developing these methods have thought carefully about the relative abilities of humans versus machines in planning, sensing, and mechanical actuation, and drawn conclusions as to what challenges are most amenable to hybrid solutions [31]. Interestingly, these methods have focused almost entirely on combining human judgment with robotic manipulation. By comparison, in this thesis, we propose methods in which the human operator is used largely as a mechanical actuator, and the automatic component performs the sensing and motion planning.

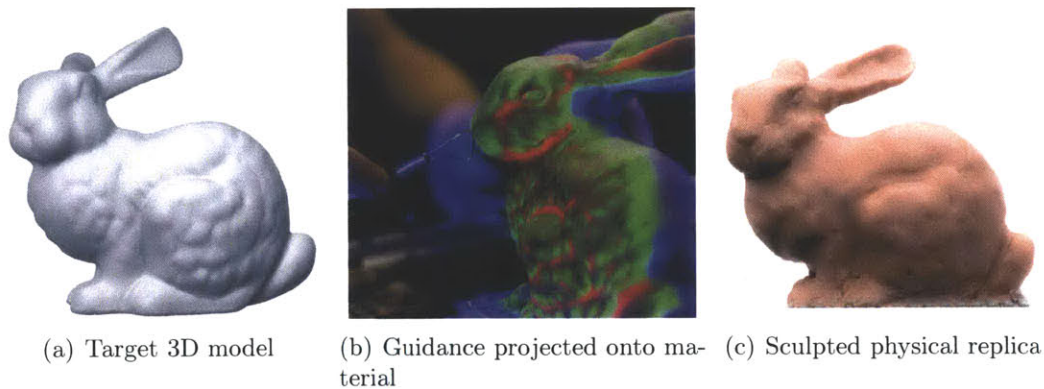


# Chapter 3

## Sculpting by numbers

### 3.1 Introduction

In this chapter, we present a method for augmenting manual fabrication by sculpting in which the only augmentation is in the form of guidance projected onto the object. We use a projector/camera pair to scan a work in progress, and use that to project



**Figure 3-1:** *We assist users in creating physical objects that match digital 3D models. Given a target 3D model (a), we project different forms of guidance onto a work in progress (b) that indicate how it must be deformed to match the target model. As the user follows this guidance, the physical object’s shape approaches that of the target (c). With our system, unskilled users are able to produce accurate physical replicas of complex 3D models. Here, we recreate the Stanford bunny model (courtesy of the Stanford Computer Graphics Laboratory) out of polymer clay.*

guidance onto the object itself that indicates how the surface needs to change in order to make the shape match a target 3D model. Thus, in this case, the computer makes up for a person’s inability to accurately detect absolute 3D positions, but leaves all physical manipulation to the user. When provided with just this missing ability, even an unskilled user is able create an accurate physical replica of a digital 3D model.

## 3.2 Related work

As described in Section 2, recent work in computer graphics has focused on software techniques that enable alternative methods for fabrication, and our technique falls into this category.

Our approach shares some similarities with previous hybrid manual-automatic methods that have been used for 2D design, again to provide automatic guidance to a user who is performing the task. Painters have used photographs or a camera obscura as references [13, 57]. Dixon et al. [18] proposed a hybrid method for the task of sketching a human face from a reference photograph in which the user is guided by automatic feedback. The guidance included high-level sketching principles for each part of the face as well as specific feedback on how to adjust the freehand drawings to more closely match the shape of the reference image. ShadowDraw [44] proposed a more open-ended system in which a user is allowed to draw freely while a “shadow image” of suggestive contours is updated and displayed behind the sketch to illustrate possible sketch completions.

Our approach to guided sculpting involves scanning the work in progress and projecting guidance that annotates the physical object. 3D scanning has been an active area of research for many years, with a variety of optical methods being proposed; see Chen et al. [12] for a survey. One of the oldest and most robust methods involves projecting a sequence of Gray codes and imaging the result to retrieve depth information [34, 9, 62]. This method is simple to implement, robust, and able to capture a full depth field quickly.

### 3.3 System design

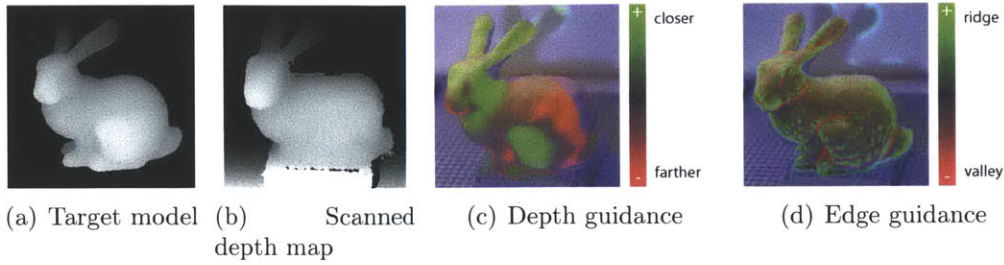
The core operation in guided sculpting is creating a physical object that matches a target 3D model. This is done by projecting different forms of guidance onto the object that illustrate how it must be changed to match a target shape. All adjustments to the model are done manually, which naturally supports a wide range of materials and shapes. The guidance makes it possible to achieve complex and precise shapes.

This approach requires two main technical components: a method for scanning a 3D shape at high spatial and depth resolution, and a projector registered to the same coordinate system to project guidance onto the material. We achieve both goals with the same hardware by using a projector-camera pair to perform structured light scanning with Gray codes (e.g. [9]). While faster scanning methods exist, e.g. the Primesense/Microsoft Kinect, they typically have much lower spatial resolution, which is more of an issue to us than refresh rate. We require the user to explicitly request each new scan, which has two advantages: the guidance does not change unexpectedly, and the user’s hands are not scanned and confused with the target object.

#### 3.3.1 Scanning

The input to the scanning process is a target 3D model, which may be a mesh or a point cloud. It is virtually positioned within the working volume, and a block of raw material (e.g. clay) is placed at roughly the same real position.

Whenever the user initiates a scan, we project a sequence of Gray codes onto the scene and photograph the scene illuminated by each code. These images give us a correspondence from camera coordinates to projector coordinates. We filter out invalid or implausible correspondences, and then invert this map to compute a correspondence from projector coordinates to camera coordinates. From this we triangulate to compute a depth map from the projector’s perspective. We also compute a depth map of



**Figure 3-2: Sculpting guidance:** *A user is given feedback during sculpting by two forms of guidance projected onto the model, which they can toggle between. Guidance is computed by comparing the target depth map (a) to the scanned depth map of the current shape (b). Depth guidance (c) illustrates the difference between the two maps and is helpful for matching the position of the surface in absolute 3D coordinates. Edge guidance (d) illustrates the second derivative of the target model (i.e. ridges and valleys), and is useful for recreating fine details once the model is close to the target shape. A smoothly-varying range of colors is used: in depth mode green indicates where the surface should be moved closer to the scanner and red farther, while in edge mode green indicates a positive second derivative (a ridge) and red negative (a valley). In either mode, blue indicates background, so any part of the material colored blue should be removed entirely.*

the target model rendered from the same perspective. Finally, we compare the depth maps to compute the guidance that we project onto the object. We describe each of these operations in detail below.

**Obtaining pixel correspondences** Gray-code-based structured light scanning operates by projecting a sequence of black and white horizontal and vertical bars onto a scene and recording the resulting images. By examining the illumination of a single camera pixel across all projected patterns, one can determine the X and Y coordinate of the projector pixel that was imaged by that camera pixel.

While the basic approach is straightforward, a number of factors can create incorrect correspondences. First, some camera pixels may image parts of the scene that are not illuminated by the projector at all. These can be detected and excluded by ignoring pixels that do not vary substantially between images in which the projector displays entirely white and entirely black.

More difficult to detect are cases in which a camera pixel records specular highlights or



reflections of light cast by the projector. These camera pixels will vary appropriately between white and black projector images, and a corresponding projector pixel can be computed for them; however, this invalid correspondence will corrupt the depth map at that projector pixel. We are able to reject the majority of these correspondences with a consistency check. Given the homographies for the camera and projector, we can determine the world-space ray that corresponds to each camera and projector pixel. These rays ought to intersect within the working volume in the case of a true correspondence. If the rays do not approximately intersect, as may be the case if the camera is seeing a reflection, we reject the correspondence. Put another way, the scan in X, the scan in Y, and the epipolar constraint combine to overconstrain the pixel correspondence, which lets us reject invalid pixels. This constraint also helps eliminate spurious correspondences detected due to noise.

A final source of inaccuracy can arise when narrow alternating black and white columns in a fine-scale Gray code pattern are blurred into an average gray due to the combination of camera and projector defocus. In our case, the working volume is often close to the camera and projector, so their depths of field are relatively shallow. Loss of the finest Gray code patterns results in losing the least-significant bits of the projector-camera correspondence.

When coarsely shaping an object this loss of resolution does not matter, but towards the end of the sculpting process it becomes important. We therefore allow the user to perform a slightly more time-consuming “precise” scan inspired by laser plane sweep scanning (e.g. [15]). In a simple implementation, we could project a single white row or column that sweeps across the entire projector image, and for every camera pixel simply record for which code the illumination was highest. This method is robust to defocus, but slow. We instead project the Gray code patterns as usual to compute a correspondence that may be incorrect in the least significant bits. We then sweep one-pixel-wide vertical or horizontal lines that recur every 8 rows or columns across all 8 positions, and record the position of maximum illumination for every camera pixel to correct the three least-significant bits of the correspondence. This precise

scan takes about 7 seconds, and gives us a depth resolution of under 1mm at the center of the working volume.

**Computing depth maps** As we will be displaying guidance using the projector, we need to compute the depth map from the projector’s point of view rather than the camera’s. To achieve this, we must resample the correspondence into the projector’s point of view.

We do so by splatting: each valid camera pixel contributes to a  $7 \times 7$  kernel with exponential falloff about the corresponding projector pixel. The contributions are summed in homogeneous coordinates and then normalized to produce a camera coordinate for some subset of projector pixels. The kernel size chosen is enough to interpolate across small holes in the depth map, which arise from projector pixels not seen by any camera pixel.

Given these resampled correspondences and the homographies for both camera and projector, we can triangulate to compute depth values for each projector pixel. Meanwhile, the depth map for the target model is computed by rendering it from the projector’s point of view using the projector’s calibrated homography as the OpenGL projection matrix. These depth maps are then compared to compute the guidance that we project onto the material.

**Hardware setup** We use a 3M MP160 projector running at  $800 \times 600$  at 60Hz and a Nokia N9 camera-phone using the FCam API [1] generating raw data at  $1632 \times 1232$  at 13Hz (75ms exposure time). The FCam API makes it easier for us to synchronize the camera with the projector, and the raw stream prevents us from losing resolution to demosaicing. Gray codes do not rely on color or spatial coherence, so we can treat each pixel independently and ignore the Bayer mosaic. The separation between camera and projector is 5”, and they placed roughly 2’ from the center of the work area. We compute world-space homographies for the camera and projector by scanning a known planar calibration target placed at two known depths in the work volume.

### 3.3.2 Guidance

We provide two forms of guidance, which the user may toggle between depending on the context. Each form is useful for recreating a different aspect of the model’s appearance.

**Depth guidance** is used to achieve the correct surface position in absolute 3D space. In this mode, projector pixels are colored based on the difference between the scanned depth at that pixel and the target depth (Figure 3-2 (c)). Green indicates areas where the user must bring the surface closer to the projector and red indicates areas where the user must move the surface farther away. Pixels that do not intersect the target model are colored blue. The range of the scale is user-tunable. The user can also probe the exact depth error in millimeters at a particular point by mousing over that pixel on the laptop controlling the projector.

**Edge guidance** depends only on the target 3D model, and is used to help the user match the high-frequency surface details of the target 3D model. In this mode, projector pixels are colored according to the second derivative of the target depth map (i.e. ridges and valleys, as shown in Figure 3-2 (d)). This color follows the same scale as the depth feedback mode, and its range can be similarly tuned.

**Rotation.** The method described so far assists a user in achieving the correct shape from a single point of view. However, a complete model must be correct from all sides. The challenge is to rotate the target model and the partial sculpture by the same amount about the same axis of rotation so the user can continue without invalidating work already done. While automatic methods (e.g. ICP [82]) could be employed to deal with this, we handle this more simply by sculpting atop a Lego stage, which can be accurately rotated by 90 degrees around a known axis.

## 3.4 Applications

Using our system in practice involves an interplay of computer guidance and human intuition, in which both the precision of the computer and the intuition of the artist are required to create an object with the correct proportions and aesthetic qualities.

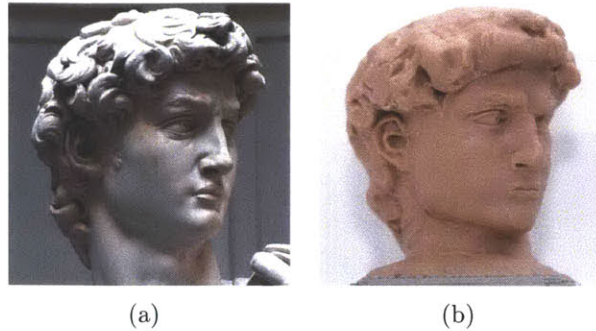
In this process, the computer’s ability to accurately detect absolute 3D shape in world coordinates means that the guidance can be relied upon to achieve the correct relative positions and sizes of features of the model, which can otherwise be extremely challenging for a novice sculptor. However, the process also relies on the user’s understanding of the shape. One simple reason is that the guidance must be interpreted and internalized, as it will both be blocked by the user’s hands and tools, and is slowly invalidated as the user works on the model between scans. As users become familiar with the system, they quickly become able to perceive how a shape must be deformed and judge when to initiate a rescan.

A less obvious but important way in which we benefit from having an artist (even if a novice one) “in the loop” is that the user can make unprompted changes to the model to achieve a desired aesthetic appearance. Thus, a user may interpret the projected guidance liberally, and perform freehand adjustments to recreate salient details. This can both speed the creation of major features and enable the reproduction of fine-detail surface features that are too small to be individually highlighted and reproduced using the system.

We describe how this process plays out for the basic case of making a shape that matches a target 3D model, and go on to explore additional applications.

### 3.4.1 Creating an object that matches a 3D model

We found that use of our system fell into a fairly consistent pattern. A typical modeling session will begin in depth guidance mode. Using this the user will establish the coarse shape and silhouette of the model. As the model approaches the target



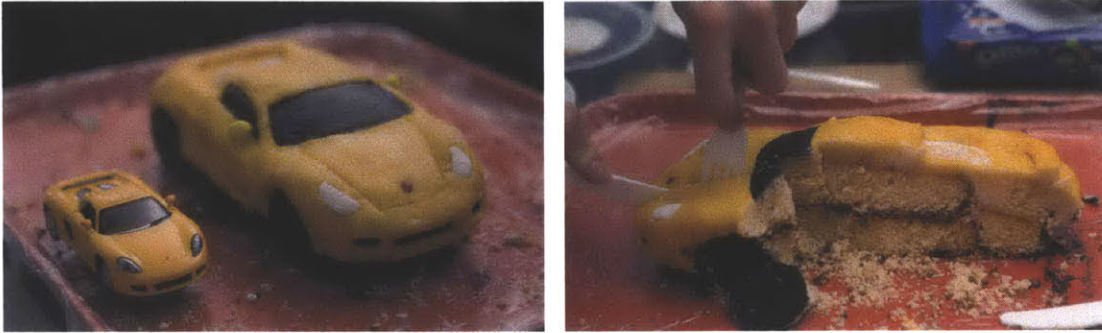
**Figure 3-3:** *The head of Michaelangelo’s David, in the original marble (photograph by Steve Hanna) (a) and in a 6” tall version made in modeling clay with our system (b) using data from the digital Michelanglo project [45]. The clay model was created by novice sculptors.*

shape, the user will make successively smaller adjustments to the object, shortening the red-to-green guidance range as necessary to highlight the necessary adjustments. Eventually, the user will have a shape that matches the target 3D model in absolute coordinates to the limit of the user’s ability to model and the system’s ability to scan.

Even after the model has been made to match the target 3D shape as closely as possible in depth guidance mode, the surface may still lack high-frequency surface details, such as wrinkles or the rough surface of hair. These details are often essential to the aesthetic appearance of a model, but are difficult to replicate with simple depth feedback: they may be finer than the scan can resolve, or may be hidden by low-frequency depth discrepancies that, although larger, can be ignored. Therefore, users will typically finish a model by switching to edge guidance mode. In this mode, they can see the locations of ridges, valleys, and other high-frequency details projected onto the surface, and use those as a starting point for an artistic pass on the surface.

The user will usually want to shape an object from all sides. Rather than forming the model perfectly from one side and then rotating, it is generally easier to form the material to match the target model roughly from all sides first, and then make successively higher-detail passes from each direction. This minimizes the problem of alterations to one side affecting already-completed regions of another.

Results of novice sculptors using our system can be seen in Figures 3-1 and 3-3. We



**Figure 3-4: Sculpting from a physical example:** *With a simple extension, we can use a scan of an existing physical object as a target for sculpting. For example, without any prior cake-decorating experience, we remade this toy car as a cake. We are free to make a replica at a different scale and out of different materials: the car has been scaled-up and made out of madeira cake, frosting, marzipan, cookies, and candy.*

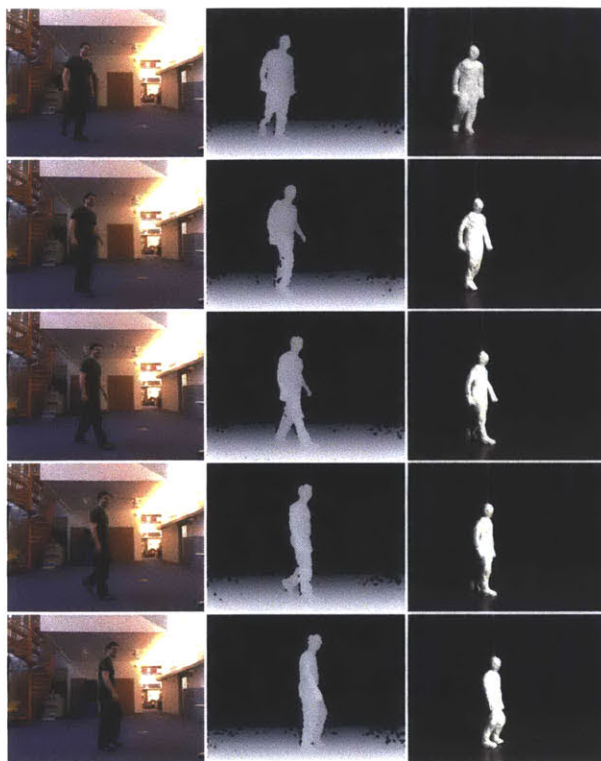
found that our system enabled unskilled sculptors to achieve far better results than they would otherwise be capable of, while giving them a sense of ownership of the resulting creations that would be absent with an automatically-produced object.

One unexpected possibility allowed by our approach is collaborative sculpting. Because the computer guidance provides a common goal to work towards, it is possible for multiple users to work on a sculpture at the same time. This is both time-saving and highly enjoyable.

### 3.4.2 Sculpting from a physical example

Our approach can also be used with a simple extension to allow the user to create an object not just from a digital 3D mesh, but also from an already-existing physical object. To do this, the user places an object in front of the projector-camera pair and scans it using the method above, rotating it and scanning it from all four directions. The recovered point clouds are saved. The model is then removed and the stored point clouds are used as the target shape for modeling as above.

Because we use the point cloud recovered by the scanning algorithm directly, we are able to sidestep the difficulties and edge cases of attempting to reconstruct a full 3D



**Figure 3-5: Stop-motion animation:** *Our approach can be used to help create stop-motion animation. We demonstrate an end-to-end system, in which a 2.5D video (left and center columns) of a real person walking was captured with a Kinect and used to create a claymation sequence (right column).*

mesh from a scan. In addition, we allow the user to change the scale or the materials of the reproduction relative to the original.

To illustrate the flexibility of this process, we scanned a toy car and used our system to recreate it at a scaling factor of  $2.5\times$  as a cake consisting of a mixture of edible materials (Figure 3-4). A timelapse of the process can be seen in the video that accompanies this paper. The completed cake was eaten and enjoyed.

### 3.4.3 Stop-motion animation

Our approach also adapts well to being used as a guide for stop-motion animation. By loading a series of targets, instead of just one, and allowing the user to flip between these targets, it becomes possible to generate a series of snapshots of a physical object

that correspond to a desired 3D animation.

While a skilled sculptor can already achieve what our system allows in its basic form, this application provides a capability that is otherwise extremely difficult even for expert artists. This is the ability to deform a physical object into a sequence of shapes such that the animation formed by concatenating images of those shapes appears physically correct. Put another way, while a skilled artist may be able to make a plausible reproduction of a single pose of a human in the act of running, even that skilled artist may have trouble determining how much to adjust the model to make it correspond to the pose of that runner exactly 1/30th of a second later.

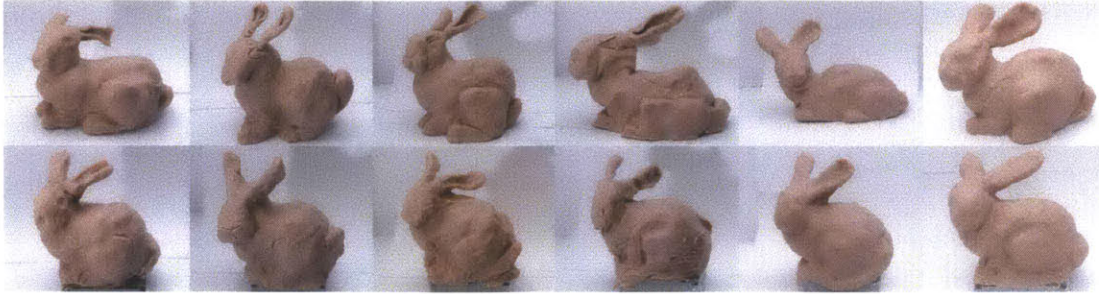
To test this, we built an end-to-end system in which a performance is captured by a Kinect and used to make a claymotion sequence (Figure 3-5). As with reconstructing an existing physical object, we have the advantage that the point cloud derived from the Kinect can be used directly for modeling, without needing to generate an explicit 3D mesh for each frame. Modeling proceeds as normal, with the addition of the ability to toggle between frames and to take a snapshot of the scene to save as a frame of the animation. We show thumbnails from the resulting animation in Figure 3-5, and the full animation can be viewed in our accompanying video.

#### **3.4.4 User testing and limitations**

We felt that our approach made it possible for users with no prior experience sculpting to create aesthetically-pleasing 3D sculptures. To test this, we asked six first-time users to reproduce a 3D model of the Stanford bunny out of modeling clay. Each user was asked to make one attempt freehand, referring to a rendering (Figure 3-1(a)), and one attempt using our system, after a 10 minute training period. Users were given 30 minutes for each attempt. To save time, we requested that only the front of the bunny need look correct. Half of the users started with the freehand attempt, and half with our method. The results are shown in Figure 3-6.

Users commented that the system greatly improved their ability to achieve the correct





**Figure 3-6: User testing results:** *After 10 minutes of training and practice, users were asked to sculpt the Stanford bunny out of polymer clay both freehand (top row), and also using our system (bottom row). 30 minutes were allotted for each task. The odd columns first used the system and then sculpted freehand, while the even columns sculpted freehand and then used our system. For all users but the artistically-inclined column 3, the system greatly improved their ability to create a correctly-proportioned bunny.*

proportions. The dynamic of the sculpting process was also changed. When sculpting freehand, users typically felt finished after 30 minutes; they acknowledged that the proportions weren't quite right, but also that they weren't sure what to do to fix them. Using our method was slower, but always provided a clear direction in which to proceed; after 30 minutes with our system, users felt that they could have continued to productively improve the model.

Sculptures made with our method were generally better proportioned than the free-hand sculptures. Only one user did not benefit from the system (shown in Figure 3-6 column 3). This user's freehand model was very good. While able to make continual progress using our method, this user was not able to achieve the same quality in the same amount of time. When a sculptor has enough skill to innately know how to adjust a model, automatic guidance just slows them down.

### 3.5 Conclusion

We have demonstrated a new approach to making a physical object that matches a target 3D shape in which an unskilled user is assisted by guidance projected onto the object as it is being made. We have found that our system greatly increases

what novice sculptors are capable of (e.g. Figures 3-1, 3-3, 3-4, and 3-5). In addition to producing physical replicas of a 3D model, our approach can be used to enable reproduction of existing physical objects, potentially at a different scale and out of different materials. We further demonstrate how our approach can be used to retarget real-world performance capture data to generate stop-motion animation with physically correct dynamics.

We believe that the combination of human actuation and situated feedback from computer-accurate measurements has great potential. In future work, we particularly wish to explore applications to the task of assembling rigid parts to accurately match a target configuration, which may be useful in industrial contexts.

## 3.6 Hindsight

The motivating idea behind this project was that accurate 3D shape perception was the key element limiting the accuracy and complexity achievable by manual sculpting. We believe that the results presented here largely bear out this conjecture. However, we learned from our experience that the mechanical part of the process – i.e., making controlled, local adjustments to a work in progress – was a significant factor as well. We had believed that if the user had perfect guidance, then achieving arbitrarily accurate motions would be possible, as it is with painting by numbers; however, sculpting imposes mechanical difficulties beyond those of painting, and these difficulties reduced the accuracy achievable in our system. Most notably, in painting, one can assume that the painting will remain static when you are not working on it, and making an adjustment (i.e. adding paint) will only affect the local area of the extra paint you add. In sculpting, this is not the case: any time you touch the material, you will be affecting the entire sculpture, e.g. by twisting it or pushing it. Thanks to the volume conservation, pushing in the surface in one area will push the surface out everywhere else, and so on.

These mechanical challenges, rather than any problems with the interface or inter-

preting guidance, ended up being the main factor increasing the amount of time it took to achieve a desired level of accuracy. Often, the guidance would instruct the user to make a local change, and the user would try to make that change; then, when the object was scanned again, the user would find that in fact that change had subtly warped the entire sculpture, necessitating further adjustments all over. This was particularly an issue with the claymation application, as the clay character was not rigidly fixed to the table and would move whenever touched.

In practice, many of these mechanical challenges were overcome (or mitigated) with mechanical solutions. We found that putting a solid bar in the middle of a sculpture greatly helped stop the unwanted deformation and allowed the user to use the system in the intended way, making the experience closer to painting by numbers. Using firmer materials also helped. While we never attempted this, sculpting a rigid material, such as wood (e.g. by sanding it), may not suffer from these difficulties at all.

For future work, we would like to explore a potential algorithmic way around these issues. We observed that much of the unwanted deformation consisted of very low-frequency global warps. These low-frequency errors did not actually detract from the aesthetic quality of the sculpture, but nonetheless dominated the correction guidance generated by our system, overwhelming high-frequency detail guidance that did in fact affect the aesthetics. These low warps therefore had to be corrected before details could be worked on. The edge guidance mode described above was, of course, not affected by this, but in some cases one wished to have depth guidance to help with details. We are therefore interested in the idea of adaptively warping the target model to the scanned point cloud before guidance is computed, so that aesthetically-important high-frequency differences in depth can be highlighted and corrected, even in the presence of variable low-frequency deformations of the shape.

In implementing the system as described, the main technical challenge was to calibrate the camera and projector relative to each other and to the scene. We initially researched existing solutions to this problem. We found that the predominant (or at

least most publicized) method was to move a fixed planar pattern to a number of random locations and orientations, and at each position to detect the pattern with the camera and then project a pattern onto a white area on the same plane adjacent to the pattern and detect that. By combining detected patterns from a variety of positions and orientations, the position of the projector relative to the camera could be determined. (Calibrating to the scene would be handled separately.)

This approach was a natural extension of the popular method for calibrating a camera in which a fixed pattern is imaged in a number of unfixed locations. However, what worked well for a camera calibration did not work as well for projector-camera calibration, due to a variety of reasons. First, for the camera, the only thing being computed is the camera intrinsics, which tend to remain constant (though refocusing can change them), and therefore one can expect to do the calibration only once. In this context, the moving known pattern approach makes sense, as the work of moving the pattern to many positions is more than offset by the time saved in not needing to create a complex calibration rig. However, in the case of the projector-camera pair, the calibration must be re-run every time the relationship between the projector and camera is changed, which can happen very easily such as when either is bumped accidentally. Furthermore, it is difficult and tedious to maneuver a board with a printed pattern and a white area such that the printed pattern can be seen by the camera and the projected pattern falls only within the white area, which also must be visible to the camera. Running this type of calibration for a projector-camera pair proved too time-consuming and frustrating. Alternatives in which both patterns can occupy the same space on the board, such as by using different color channels or projecting recognizable QR codes, were insufficiently robust in the publicly-available implementations, and still difficult to use. At that point, it became well worth it to expend a modest amount of effort to develop a calibration rig that could hold a known pattern at two known locations. With this rig in place, calibration became trivially easy, to the extent that we typically recalibrated at the beginning of every modeling session (in case either projector or camera had been inadvertently moved).

While the applications we presented here tended to be artistic, we see significant potential for industrial and practical applications to this technology. A large amount of fabrication is still done using hand tools, even when digital plans are being used, and in these cases it is a real challenge to meet tolerance requirements relative to those 3D specifications. For example, a metalworker may be bend a sheet of metal in an English wheel (a sheet metalworking device) to match a 3D specification, as it is far faster and cheaper than using a CNC mill to machine a stamping die. In cases such as these, we believe that our approach, in which analysis is automatic and guidance is projected in an intuitive format directly onto the part itself, could be very helpful to guide the machinist. Assembly tasks, in which a collection of rigid parts must be precisely aligned relative to one another, may also be a useful application domain for this system.

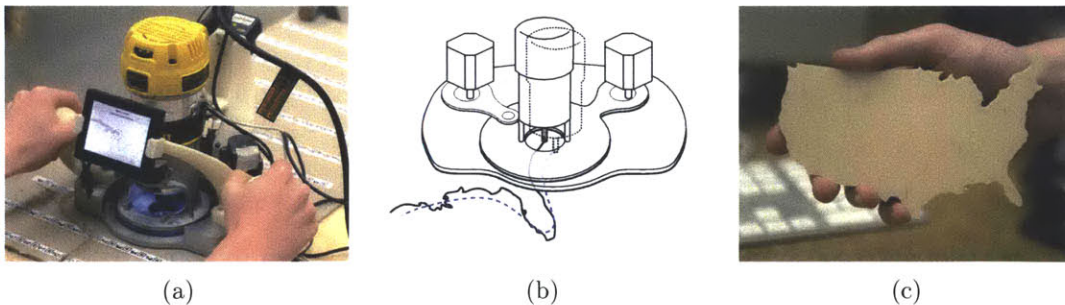


# Chapter 4

## Position-correcting tools

### 4.1 Introduction

Our second method for augmented manual fabrication provides both guidance and partial actuation to aid in the task of precisely moving a tool along an arbitrary 2D path. This type of precise motion is typically accomplished fully automatically us-



**Figure 4-1: Overview:** (a): A position-correcting tool. The device consists of a frame and a tool (in this case a router) mounted within that frame. The frame is positioned manually by the user. A camera on the frame (top right in the figure) is used to determine the frame's location. The device can adjust the position of the tool within the frame to correct for error in the user's coarse positioning. (b): To follow a complex path, the user need only move the frame in a rough approximation of the path. In this example, the dotted blue line shows the path that the tool would take if its position were not adjusted; the black line is its actual path. (c): An example of a shape cut out of wood using this tool.

ing a computer-controlled multi-axis stage. With that approach, one can only create objects smaller than the positioning stage, and large stages can be quite expensive. In our approach, a user coarsely positions a frame containing the tool in an approximation of the desired path, while the device tracks the frame’s location and adjusts the position of the tool within the frame to correct the user’s positioning error in real time. Because the automatic positioning need only cover the range of the human’s positioning error, this frame can be small and inexpensive, and because the human has unlimited range, such a frame can be used to precisely position tools over an unlimited range.

## 4.2 Related Work

Our computer-augmented positioning approach removes the limitation on range of traditional gantry-based positioning technologies. To do so, it relies on accurately detecting the position of the frame in real time. A variety of approaches to real-time localization have been employed over the years, from global-scale GPS [23] to local-scale systems based on radio and ultrasonic signals [58]; an overview is given in a survey by Welch and Foxlin [79].

Our approach to localization is based on computer vision. Computer vision has been widely used for position tracking in the context of motion capture (see Moeslund et al. [52] for a survey). These setups typically use stationary cameras tracking a moving object, though recently Shiratori et al. [67] proposed a system in which cameras are placed on the human and track the environment. In our approach, the camera is on the tool and tracks the material over which it moves, first stitching frames together to make a map of the material (see Zitova and Flusser [83] and Szeliski [73] for surveys of image registration and stitching techniques) and then using that map to perform localization. This approach has been used before, with some differences, in a recent new peripheral, LG’s LSM-100 scanner mouse [46, 80], which is a mouse that can scan a document it is passed over. Our implementation differs from theirs in that we use



only a camera (no optical mice), capture a wider area of the material in each frame, and use high-contrast markers placed on the material to allow capture of untextured materials.

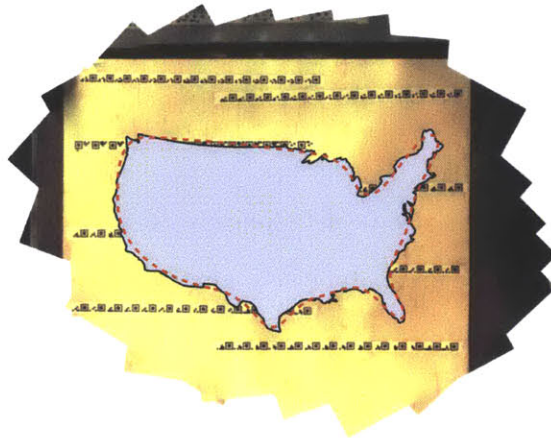
Computer vision has previously been applied to CNC manufacturing, for example to monitor for mistakes [3], or to precisely align a tool path to a piece of material [75]. These approaches, however, do not re-imagine the fundamental approach or form factor of a table-based, fully automatically-positioned CNC device.

### 4.3 Localization

To keep the tool on the plan as closely as possible, the tool must detect its current position accurately, robustly, and with low latency.

We considered a variety of localization systems, eventually settling on a simple computer vision-based approach, in which a camera on the frame of the device tracks high-contrast markers placed in an arbitrary pattern on the material. A map of the material (Figure 4-2) is first built by passing the device back and forth over the material to be cut; then, images from the camera are compared to this map to determine the device's location. This approach was chosen for a variety of reasons: it can achieve very high accuracy; it always remains calibrated to the material, as the markers are on the material itself (as opposed to external beacons, which can become uncalibrated); it does not require excessive setup; the hardware required is relatively inexpensive; and it can be implemented using standard computer vision techniques. Building the map is fast and easy.

We considered using the camera to track just motion, as in an optical mouse, but this approach would be subject to drift. An alternative would be to draw the plan on the material itself, e.g. with a pencil, and then track that, but that would require additional work on the part of the user and could introduce error.



**Figure 4-2: Map:** *A scanned map with a plan registered to it. The red dotted line indicates a path that a user could conceivably follow to cut out the shape.*

### 4.3.1 High-contrast markers

We leverage specially-printed tape marked with high-contrast patterns to make it possible to track materials that have no visual features of their own (such as sheet metal or plastic) and to increase robustness under varying lighting conditions. This tape is applied before map-making, in any pattern so long as some tape is visible from every position that the device will move to, and can be removed when the job is complete. The tape consists of many Quick Response (QR) code-like markers [17] in a row, each consisting of an easily-detectable box-within-box pattern we call an “anchor” and a 2D barcode that associates a unique number with the anchor (see Figure 4-3). As long as four of these markers are visible at any time (which is typically the case even if only a single piece of tape is visible), the device is able to locate itself. The redundancy of the markers means that it does not matter if some are occluded (e.g. by sawdust) or obliterated by the tool itself. Note that these markers function just as features – their positions are not assumed before mapping, and they need not be laid out in any specific pattern.



**Figure 4-3: Markers:** *A sequence of markers, with values 1000 to 1006, such as would be printed on a strip of tape. In our current implementation, markers are printed at a size of roughly  $0.8'' \times 0.4''$ . This is small relative to the area of the material the camera can see at once (roughly  $8'' \times 6''$ ).*

### 4.3.2 Image processing

The core operations used during locating and building a map are detecting markers in an image and registering one set of markers onto another.

**Detecting markers** To detect markers, the frame is first binarized using the Otsu method [56], and then rectified to a top-down orthographic view based on a one-time calibration of the camera relative to the flat plane on which the tool sits. Anchors are extracted using a standard approach to QR code reading: first, horizontal scanlines are searched for runs of alternating pixel colors matching the ratio of 1:1:3:1:1, as will always be found at an anchor. Locations that match this pattern are checked for the same pattern vertically. Locations that match horizontally and vertically are floodfilled to confirm the box-within-box pattern. Once anchors have been extracted, each anchor is experimentally matched with the nearest anchor, and the area in between is parsed as a barcode. Barcode orientation is disambiguated by having the first bit of the 2D barcode always be 1 and the last bit always be 0. If the parsed barcode does not match this pattern, the next-nearest anchor is tried. If neither matches, the anchor is discarded. If the pattern is matched, the barcode's value is associated with the first anchor and that anchor's position is added to the list of detected markers.

**Matching sets of markers** To match two sets of markers, we find all pairs of two markers, one from each set, that share the same ID. If there are at least four such pairs, we run RANSAC [Fischler and Bowles, 1981] to find the Euclidean transformation

that maps the positions of the pairs' markers in the first set to the corresponding positions in the second set, with a tolerance of 5 millimeters. While only two pairs are sufficient to determine a Euclidean transformation, we set the minimum number of inliers to four to prevent false positives. The resulting least-squares transformation matrix establishes a relative position constraint between the sets.

### 4.3.3 Building a map

Mapping is done by stitching together video frames into a 2D mosaic (Figure 4-2) as the user passes the device back and forth over the material. To reduce memory loads, we retain only frames that overlap with the previously retained frame by less than 75%, as computed after determining the frame's position relative to all other frames as described below.

We use a simple method to stitch images together. Each frame's current position and orientation in the map is expressed as a  $3 \times 3$  transformation matrix. The first frame is assigned the identity matrix. When a new frame is received, a match is attempted with all previous frames. Every successful match generates one relative position constraint, also expressed as a  $3 \times 3$  matrix. The new frame's transformation matrix is obtained by multiplying each matched frame's transformation matrix with the corresponding constraint matrix, and averaging the resulting matrices. The new frame's transformation matrix is then orthonormalized.

Because frames' positions are constrained only relative to overlapping frames, the transformation between two non-overlapping frames depends on the relative constraints of some number of in-between frames. This transformation is less accurate as the number of intermediate connections increases. This may cause discontinuities where the sequence of frames loops back on itself. We smooth out these discontinuities by iteratively reapplying relative position constraints for all frames once all images have been acquired, until the system converges to a stable configuration, similar to bundle adjustment [76]. In each iteration of smoothing, each frame computes a new

transformation matrix by multiplying each connected frame's transformation matrix from the previous iteration with the corresponding constraint matrix, and averaging the resulting matrices across all constraints. As before, the new transformation matrix is then orthonormalized. This process always converges in practice.

Once the map is complete, a super-list of markers for the entire map is generated from the markers in input images by averaging the map-space positions of markers that share the same ID. This global list of known positions for each marker ID is used to localize new images when the device is in use.

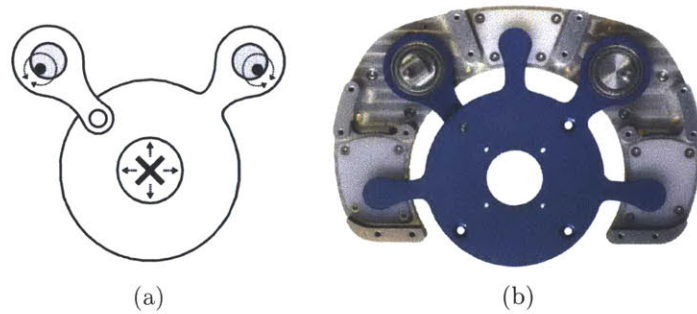
When preparing to cut a shape, the user registers a shape onto this 2D map. Having the map of the material makes it trivial to visually align the plan with features of the material. With a traditional CNC machine, this would typically require careful calibration.

#### **4.3.4 Localization using the map**

Once the map has been created, registering a new image to the map is straightforward. Markers are detected as above and matched to the global list of markers using the same RANSAC algorithm as above. An image from the camera can be registered to a map in  $\sim 4$  milliseconds total on a standard laptop. Although localization exhibits strong time coherence, thanks to the low cost of processing we can afford to solve the system from scratch at every frame.

### **4.4 Actuation**

Once the location of the frame is known, the tool must be repositioned within the frame to keep it on the plan. This task can be broken down into the control challenge of determining where within the frame to move (as there are usually many possible positions that lie on the plan) and the mechanical engineering challenge of building an accurate, responsive, and low-cost position-adjusting actuation system.



**Figure 4-4: Positioning linkage:** (a): *Our linkage converts the rotary motion of two shafts (filled circles) into translation of a tool (cross) mounted on a sliding stage. This is achieved using eccentrics (shaded circles), which are circular disks rotating about the off-center shafts to produce linear displacement in fitted collars. To properly constrain the degrees of freedom of the stage, one eccentric is directly connected to the stage while the other is connected via an additional hinge.* (b): *A photo of the actual linkage.*

The range of our positioning linkage was determined by first attempting to move the frame along a 2D plan as closely as possible by hand. We found that when provided with accurate location information relative to the plan a user can keep the tool within  $1/8''$  of the plan, even when cutting wood. Having accurate location information allows for greater precision than normal freehand positioning. To allow for a safety margin and increase ease of use, we doubled this value to arrive at the goal of being able to correct errors up to  $1/4''$  (i.e. having a range circle with a  $1/2''$  diameter).

#### 4.4.1 Actuation system

The actuation system need only support a small range of motion, as it need only correct the coarse positioning done by the human. This affords the possibility of using a very different design for the positioning system than the multi-axis stage employed by traditional rapid prototyping machines.

Our major mechanical departure from traditional rapid prototyping machines is that we use eccentrics, rather than linear stages, to convert the rotational motion of the motors into linear motion. Eccentrics are circular disks rotating around an off-center

shaft. As they are rotated, they produce linear motion in a collar wrapped around the disk. Eccentrics are able to maintain the same low-backlash accuracy of a precision linear stage while being much cheaper. For this, they sacrifice range. However, a linear displacement range of 1/2" is well within the capabilities of an eccentric.

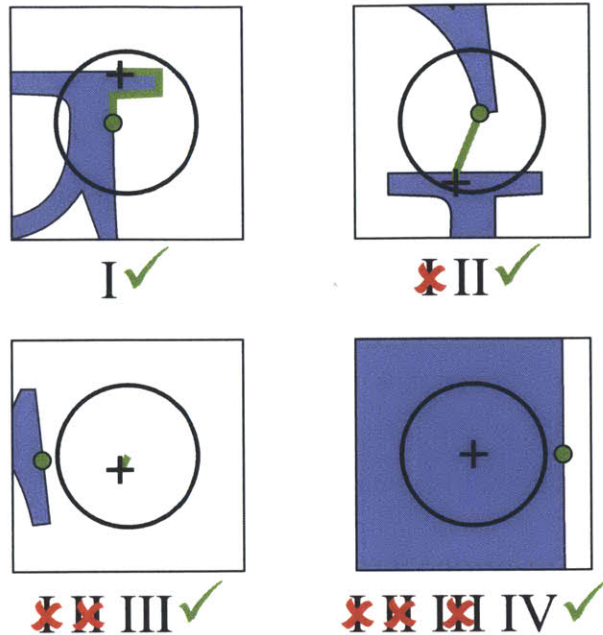
Our design (Figure 4-4) consists of two eccentrics mounted to the frame and connected to a stage that can slide on its base. The eccentrics are rotated by stepper motors, and by rotating them the stage can be moved within the frame. To properly constrain the stage, one eccentric is connected directly to the stage by a ball bearing coupling, while the other is connected both by a coupling and a hinge.

This linkage design results in a nonlinear relationship between eccentric orientation and stage position: as the tool approaches the edge of its range, increasingly large rotations are necessary to move the stage a fixed amount. We limit stage displacement to ~95% of the maximum range to cut out the positions with extreme nonlinearity. This linkage design also permits backdriving, in that forces acting on the tool can cause the cams to rotate away from their target positions; however, we found that the stepper motors we use (62 oz-in holding torque) are sufficiently powerful to preclude backdriving, even in the presence of significant material forces.

#### **4.4.2 Following a plan**

As the user moves the frame, the device must ensure that the tool stays on the plan. To do this, the path that is to be followed must be first computed (which may not be the same as the plan); then, every frame, given the frame's position, the tool's position within the frame, and the plan, the device must determine where to move the tool within the frame.

For the applications we focus on – routing and vinyl cutting – the user generally wishes to cut a shape out of a piece of material. This means that there will be some areas of the material that are outside the target shape, and which may be cut freely (which we call “exterior material”), while other areas lie inside the target shape and



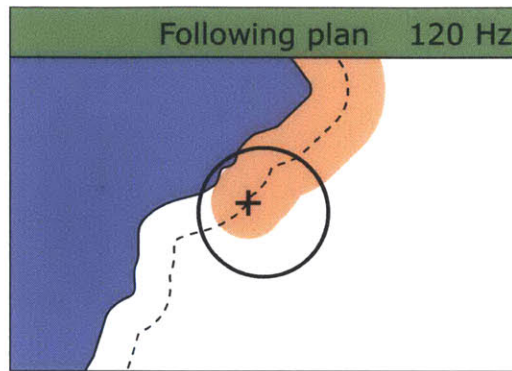
**Figure 4-5: Freeform motion paths:** *Each box illustrates a case in which a different path (described below) is used, due to the higher-preference paths being infeasible. In each box, the cross is the current position of the tool, the circle is the range of the positioning system, the green dot is the target position, and the green path is the selected path.*

must not be cut (“interior material”). To allow for this distinction, we define a plan as consisting of polygons, which have defined interiors, rather than simply as paths.

In applications such as vinyl cutting, the tool should follow the border of the interior material as closely as possible. When routing, however, the size of the cutting bit must be taken into account, and the tool should move along a path offset from the interior material by the radius of the bit, to leave the actual cut shape as close as possible to the specified plan. We provide an option to set the diameter of the cutting bit and offset the plan polygons accordingly.

We propose two different strategies for moving the tool to keep it on the plan, and will show how each of these is appropriate for a different class of applications.





**Figure 4-6: User interface:** *This display shows the shapes of the plan (blue polygons); the path that the tool is actually following, which is those shapes offset by the tool's radius (dotted line); the tool's current position (cross); the area cut by the tool (shaded area); and the range of the tool's position correction (black circle). As long as the user keeps the tool path within the correction range, the tool should be able to follow the plan.*

### Constant-speed motion

In the simpler strategy, the tool is moved through the material at as close to a constant rate as possible. This strategy is useful for applications such as routing, in which the material may offer resistance if the tool is moved too quickly and may burn if the tool is moved too slowly.

In this approach, the user decides only what polygon to follow and when to start motion. Thereafter, the software drives the tool around that polygon at a constant rate; the rate is a setting the user can tune (0.2" per second by default). While the tool is moving, the user moves the frame to keep the tool near the center of its range, ensuring that the tool can continue its constant-speed motion without reaching the end of its range. If the tool does reach the end of its range, it must stop until the user catches up.

### Freeform motion

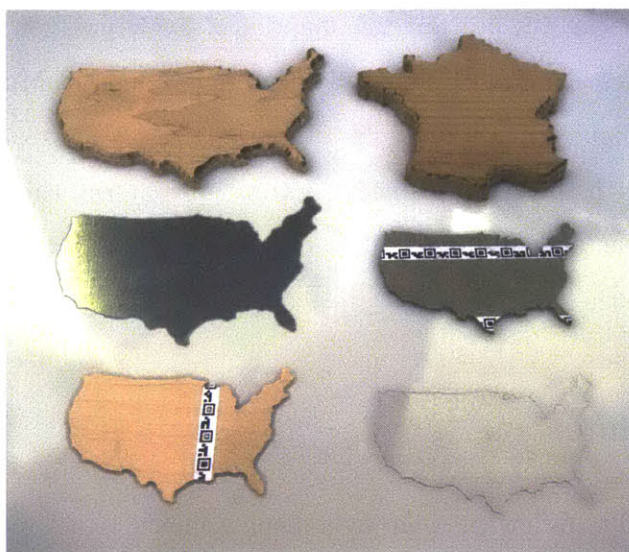
In the second strategy, the user moves the frame around the plan freely, and the device tries to keep the tool at the point on the plan that most "makes sense" given

the user's positioning of the frame. This approach is suitable to applications such as plotting or vinyl cutting in which there is negligible material resistance and no need to move at a constant rate. It can be used to cut shapes much faster than the constant-speed strategy, as the user can slow down only when necessary, such as at sharp turns. In this mode, the amount of error in the tool's positioning is related not to the overall speed of the device, but rather to the speed at which the device is moving away from the plan; roughly speaking, the user should not diverge from the plan faster than 0.2" per second.

The point that the tool is moved to is, generally speaking, the closest point on the border of a plan polygon to the center of the tool's range. However, several considerations make the determination of the path to get to this point complicated. First, the tool should never move through interior material, even if the shortest path from its current position to the target position goes through it. Second, the tool should seek to follow the border of the interior material even when a shorter direct route is possible through exterior material, to avoid skipping over features of the plan.

We aim to account for these considerations while also maximizing the predictability of the tool's motion. We propose a simple strategy in which four possible paths are computed each frame, ranking from most desirable to least desirable, and the most desirable path that is feasible is followed. All seek to move the tool to the target position, which is the closest point on the border of a plan polygon to the center of the tool's range, or to the center of the tool's range itself if the target position is not reachable. These paths, illustrated in Figure 4-5, are:

- I. The path that goes from the tool's position to the nearest point on the border of a polygon, and then walks along the border of that polygon to the target position in whichever direction is shorter. This path is infeasible if it leaves the tool's range or if the target position is on the border of a polygon other than the polygon closest to the tool's position.
- II. The path that goes from the tool's position to the nearest exterior material (if it is in the interior material) and then in a straight line to the target position.



**Figure 4-7: Results:** *Several shapes cut out from various materials (clockwise from top left: hardwood, hardwood, paperboard, polycarbonate plastic, plywood, and sheet metal).*

This path is infeasible if the nearest exterior material is outside the range or the straight line segment passes through interior material.

III. The path that goes from the tool's position to the nearest exterior material (if it is in the interior material) and then in a straight line to the center of the tool's range, stopping whenever interior material is encountered. This path is infeasible if the nearest exterior material lies outside the range of the tool.

IV. Don't move. This path is always feasible.

## 4.5 Using the tool

The use of the device proceeds as follows: the user places marker tape on the material; the user scans the material; the user registers a plan onto the scanned map of the material; the user uses the device to follow the plan. The user roughly follows the shape of the plan, and the positioning linkage moves the tool to keep it exactly on the plan. In principle, the tool can follow any 2D path. In the application of routing,

this means that it can cut out any 2D shape in a single pass, or more complex 2.5D (heightmap) shapes using multiple passes at different depths. Multiple passes can be taken with or without adjusting the plan between passes. For example, we have used multiple passes with the same plan to cut through material thicker than can be cut in a single pass; we have also used different plans to engrave text to a shallow depth on a piece that is then cut out.

### 4.5.1 User interface

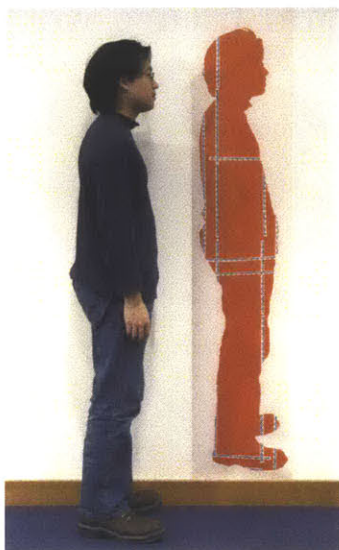
The user is helped to follow the plan by a display shown on a screen on the tool. This display shows the position of the tool relative to the plan (see Figure 4-6). In theory, the user's task is to keep the center of the router's motion range as close to the plan as possible. In practice, the user may deviate by as much as the radius of the router's adjustment range.

We found that this simple interface was able to effectively guide the user in every case we tested. It was easy to follow complex shapes, and thanks to showing the area already cut by the tool, it was possible to interrupt cutting and then return to the correct place.

## 4.6 Results

We built a device (Figure 4-1) that implements the position-correcting system described above. The aluminum base was cut on a CNC mill, while the plastic parts were made with a 3D printer. The device that we built can be mounted with a router or vinyl cutter, and can follow any 2D plan. Figures 4-1 and 4-7 show shapes cut out of wood, plastic, paperboard, and sheet metal. Figure 4-8 demonstrates the range of the device with a full-size vinyl cutout of a human silhouette. Figure 4-9 shows an example of a cut shape with high-resolution details.

We empirically tested the fidelity of shape reproduction by plotting a complex pattern



**Figure 4-8: Range:** *A full-size vinyl cutout of a human silhouette (5'6" tall), with original.*



**Figure 4-9: Fine details:** *With a vinyl cutter, the resolution of features is not limited by the width of the bit. Here, we show a 6"-wide sticker with fine details.*

with a pen mounted as the tool, scanning the result, and measuring deviation from the digital plan (Figure 4-10). The shape was plotted 6" wide. We fitted a curve to the scanned plot, aligned the plan to that curve, and measured deviation from evenly-sampled points along the drawn shape curve to the nearest point on the plan. The average error was 0.009", with a maximum error of 0.023". The error was small enough that the aligned design always fell within the width of the pen stroke.

## 4.7 Conclusion and future work

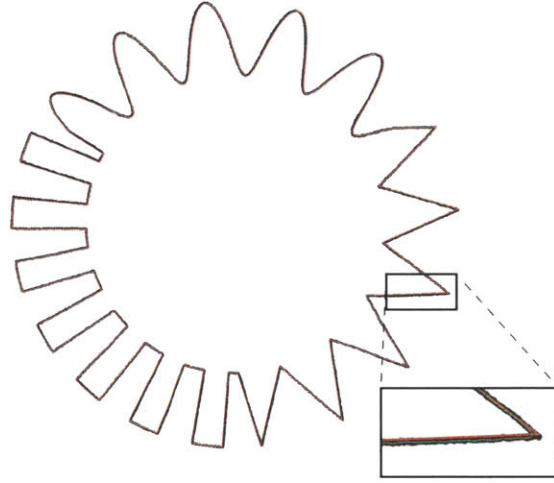
We have demonstrated a computer-augmented positioning system that avoids the cost-versus-range tension that currently affects rapid prototyping devices, and shown a tool using this approach that combines the unlimited range of a human operator with the accuracy of a computerized positioning system. This device incorporates a computer vision-based system for localization and a specially designed low-cost linkage that can be used to adjust the position of a tool within the device's frame. We have shown how this device can be used with a router and a vinyl cutter to accurately fabricate objects from digital plans.

In future work, we would like to explore applying this type of computer-augmented positioning to a variety of other tools and device form factors. It may also be useful to add automatic Z-axis control to the tool for certain applications, such as when the plan consists of many disconnected subparts. We also wish to explore the possibility of capturing the map using a separate camera looking over the work area, as this would speed up mapmaking and may be useful in cases where many sheets of material are cut in the same physical workspace.

## 4.8 Hindsight

As with *Sculpting by Numbers*, one of the lessons we learned in this project was to respect the mechanical limitations of the user. In this case, we first designed a prototype that was theoretically as capable as our ultimate design, and observed that it was far less useful in practice due to the mechanical difficulties of operating it.

In that prototype, the router could be automatically adjusted along only one axis, which ran horizontally across the device. It was the responsibility of the user to keep that axis roughly perpendicular to the line that was being cut. While that seemed straightforward conceptually, it proved too taxing on the user to both translate and rotate the device during use. Instead, the user often left the adjustment axis nearly



**Figure 4-10: Accuracy:** A scan of a plotted pattern (6" wide) shown with the design that was used to create it (red). The inset shows an expansion of the area of worst error, with the addition of the line fit to the scan for analysis (green). Note that even here the error is only on the order of the width of the pen.

parallel to the cut line; this diminished the ability of the device to correct and hurt accuracy. In our second prototype, we resolved this by supporting position adjustment over two axes. This eliminated the need to rotate the device, greatly increasing its usability and accuracy.

Other implementation issues remain. In our prototype, the camera's field of view was unfortunately small, which meant that tape had to be spaced quite closely on the material, and many passes had to be made to scan a large piece of material. This issue will likely be fixed in a future version by simply using a wider lens.

Another mechanical issue we encountered was slip-stick behavior of the device on the material. As the user attempted to move the tool in a smooth motion, the device would stick to one location due to static friction, only to suddenly overcome the friction and jerk to a new location. This slip-stick issue was significantly ameliorated by the addition of the constant-speed strategy as outlined above. Previously, we had been using the freeform motion strategy, and this had the effect that when the device jerked forward, the device would attempt to move the tool through a significant amount of material to catch up, adding significant material forces that made it hard to

control. In the constant-speed strategy, by comparison, when the tool jerked forward, the device moved the tool backwards, and minimizing the amount of extra material forces occurring at one time.

We also encountered difficulties in software. Our initial plan was to track the material solely based on visible features of the material such as wood grain. This, it turned out, was impractical given the current state of computer vision. In particular, variations in lighting conditions made it very difficult to precisely align two images of the material. In addition, the time required to extract complex features from each image was significant, making it difficult to maintain a sufficiently high framerate. Switching to the tape, in addition to making it possible to work with featureless materials such as sheet metal or plastic, largely resolved these issues, by providing sharp, easily-detectable features that could be resolved precisely even in varying lighting conditions.

Going forward, we see a wide range of possible applications of position-augmented devices for digital fabrication. We consider the device proposed here as fundamentally consisting of three separate components: a localization device for devices moving over sheet material, which we think of as a “tool GPS” system; the position-adjustment linkage that allows us to programmatically adjust the position of a tool over a short range; and the tool itself, in this case a router or a vinyl cutter. We would like to develop both of the first two independently, and believe that the tool GPS device may find use even in the absence of an attached position-adjustment system.



# Chapter 5

## Conclusion

In this thesis, we have presented two novel methods for augmented manual fabrication. These contributions each aim to make up for some inability on the part of a user by supplementing the user's abilities with the abilities of a digital device. One method compensates for a user's inability to detect absolute 3D positions with a system that accurately scans a working volume and conveys guidance based on that information back to the user. The second compensates both for a user's inability to detect absolute position and a user's mechanical inability to make fine adjustments to a tool's position with a combined guidance and partial actuation system.

We believe that augmented manual fabrication methods such as these fill a real gap in the space of fabrication solutions. While fully automatic and fully manual methods tend to each excel in different areas, such as automatic tools' high accuracy versus manual tools' unlimited range, augmented manual methods tend to occupy a middle ground across a variety of characteristics: cost, precision, range, portability, space requirements, and speed. We believe that this is an excellent position to occupy for a variety of applications, such as home users who desire accuracy but do not have the space or resources to dedicate to a CNC device.

This thesis has presented augmented manual tools for the tasks of 3D sculpting and 2D routing and vinyl cutting. We believe there is potential for augmented methods

to be usefully applied to a far wider range of tasks, encompassing every type of manufacturing that is currently done manually. These future methods may develop not only improved methods but also new targets for what in the fabrication process is subject to augmentation.

The methods presented in this thesis were primarily concerned with helping a user move a tool to the correct 3D position. We believe that far more tasks stand to benefit from hybrid manual-automatic solutions. As just one example, we would be interested in exploring technologies that help plan and guide the user through the steps of a fabrication process for complex jobs. To illustrate, one could imagine a semiautomatic means for determining in what order of steps to construct a house, coupled with a device such as the ARMAR system [26, 27] to help go through those steps.

Applications to new domains may require the development of better enabling technologies. In particular, we would like to explore tailored solutions for accurately detecting the 3D position of an object across a large range. While this is a general problem of broad concern that has long resisted solution, we believe that it may be possible to solve it in the domain of augmented manual fabrication by taking advantage of both the active participation of the user (e.g. to place beacons at fixed locations) and the knowledge of the shape being constructed (e.g. to localize by recognizing landmarks from the partially-completed plan).

# Bibliography

- [1] Andrew Adams, Mark Horowitz, Sung Hee Park, Natasha Gelfand, Jongmin Baek, Wojciech Matusik, Marc Levoy, David E. Jacobs, Jennifer Dolson, Marius Tico, Kari Pulli, Eino-Ville Talvala, Boris Ajdin, Daniel Vaquero, and Hendrik P. A. Lensch. The Frankencamera. *ACM Trans. Graph.*, 29(4):1, July 2010.
- [2] Maneesh Agrawala, Doantam Phan, Julie Heiser, John Haymaker, Jeff Klingner, Pat Hanrahan, and Barbara Tversky. Designing effective step-by-step assembly instructions. In *ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03*, volume 22, page 828, New York, New York, USA, July 2003. ACM Press.
- [3] G.A.H. Al-Kindi, R.M. Baul, and K.F. Gill. Vision-controlled CNC machines. *Computing and Control Engineering Journal*, 4(2):92–96, 1993.
- [4] Marc Alexa and Wojciech Matusik. Reliefs as images. *ACM Trans. Graph.*, 29(4):60:1–60:7, July 2010.
- [5] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, August 1997.
- [6] Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph.*, 29(4):63:1–63:10, July 2010.
- [7] Oliver Bimber and Ramesh Raskar. *Spatial augmented reality: merging real and virtual worlds*. A. K. Peters, Ltd., Natick, MA, USA, 2005.

- [8] Doug A Bowman, Ernst Kruijff, Joseph J LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.
- [9] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:470 – 480, 1998.
- [10] CATIA. Dassault Systemes. 2009.
- [11] T P Caudell and D W Mizell. Augmented reality: an application of heads-up display technology to manual manufacturing processes. *Proceedings of the TwentyFifth Hawaii International Conference on System Sciences*, 2(1-4):659–669 vol.2, 1992.
- [12] Frank Chen, Gordon M. Brown, and Mumin Song. Overview of three-dimensional shape measurement using optical methods. *Optical Engineering*, 39(10), 2000.
- [13] V D Coke. *The painter and the photograph: from Delacroix to Warhol*. University of New Mexico Press, 1964.
- [14] Robert Davis Cook. Finite Element Modeling for Stress Analysis. December 1994.
- [15] B. Curless and M. Levoy. Better optical triangulation through spacetime analysis. *Proceedings of the Fifth International Conference on Computer Vision*, pages 987–994, June 1995.
- [16] H Delingette, X Pennec, L Soler, J Marescaux, and N Ayache. Computational Models for Image-Guided Robot-Assisted and Simulated Medical Interventions, 2006.
- [17] Denso-Wave Incorporated. QR Code Specification. <http://www.denso-wave.com/qrcode/index-e.html>.
- [18] Daniel Dixon, Manoj Prasad, and Tracy Hammond. iCanDraw: using sketch recognition and corrective feedback to assist a user in drawing human faces. In

*Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 897–906, New York, NY, USA, 2010. ACM.

- [19] Yue Dong, Jiaping Wang, Fabio Pellacini, Xin Tong, and Baining Guo. Fabricating spatially-varying subsurface scattering. *ACM Trans. Graph.*, 29(4):62:1–62:10, July 2010.
- [20] Julie Dorsey, Songhua Xu, Gabe Smedresman, Holly Rushmeier, and Leonard McMillan. The Mental Canvas: A Tool for Conceptual Architectural Design and Analysis. *PG*, 2007.
- [21] Brook Drumm. Printrbot. <http://www.printrbot.com/>, 2011.
- [22] Steven Feiner. Apex: An Experiment in the Automated Creation of Pictorial Explanations. *IEEE Comput. Graph. Appl.*, 5(11):29–37, November 1985.
- [23] I.A. Getting. Perspective/navigation-The Global Positioning System. *IEEE Spectrum*, 30(12):36–38, 43–47, 1993.
- [24] Miloš Hašan, Martin Fuchs, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. Physical reproduction of materials with specified subsurface scattering. *ACM Trans. Graph.*, 29(4):61:1–61:10, July 2010.
- [25] Vincent Hayward, Oliver R. Astley, Manuel Cruz-Hernandez, Danny Grant, and Gabriel Robles-De-La-Torre. Haptic interfaces and devices. *Sensor Review*, 24(1):16–29, 2004.
- [26] Steven Henderson and Steven Feiner. Exploring the Benefits of Augmented Reality Documentation for Maintenance and Repair. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1355–1368, October 2011.
- [27] Steven J Henderson and Steven K Feiner. Augmented reality in the psychomotor phase of a procedural task. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 191–200, Washington, DC, USA, 2011. IEEE Computer Society.

- [28] Kristian Hildebrand, Bernd Bickel, and Marc Alexa. crdbrd: Shape Fabrication by Sliding Planar Slices. In *Computer Graphics Forum (Eurographics 2012)*, volume 31, 2012.
- [29] Gerd Hirzinger, Gerhard Grunwald, Bernhard Brunner, and J Heindl. A Sensor-based Telerobotic System for the Space Robot Experiment ROTEX. In *The 2nd International Symposium on Experimental Robotics II*, pages 222–238, London, UK, UK, 1993. Springer-Verlag.
- [30] Taylor Hokanson and Chris Reilly. DIYLILCNC. <http://diylilcnc.org/>.
- [31] R D Howe and Y Matsuoka. Robotics for surgery. *Annual Review of Biomedical Engineering*, 1(1):211–40, 1999.
- [32] Takeo Igarashi and John F. Hughes. A suggestive interface for 3D drawing. *Symposium on User Interface Software and Technology*, 2001.
- [33] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: A Sketching Interface for 3D Freeform Design. *International Conference on Computer Graphics and Interactive Techniques*, 1999.
- [34] S. Inokuchi, K. Sato, and F. Matsuda. Range-imaging for 3D object recognition. *ICPR*, pages 806–808, 1984.
- [35] Intuitive Surgical Inc. The Da Vinci Surgical System.
- [36] Levent Burak Kara and Kenji Shimada. Sketch-Based 3D-Shape Creation for Industrial Styling Design. *IEEE Comput. Graph. Appl.*, 27:60–71, 2007.
- [37] O Karpenko, J F Hughes, and R Raskar. Free-form sketching with variational implicit surfaces. *Computer Graphics Forum*, 21:585–594, 2002.
- [38] Olga A Karpenko and John F Hughes. SmoothSketch: 3D free-form shapes from complex sketches. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 589–598, New York, NY, USA, 2006. ACM.

- [39] Shawn Kelly. Bluemax CNC. <http://www.bluemaxcnc.com/Gantry-Router.html>.
- [40] Martin Kilian, Simon Flöry, Zhonggui Chen, Niloy J. Mitra, Alla Sheffer, and Helmut Pottmann. Curved folding. *ACM Trans. Graph.*, 27(3):75:1–75:9, August 2008.
- [41] D. Kragic, D Kragic, P Marayong, M Li, A. M Okamura, and G. D Hager. Human-machine collaborative systems for microsurgical applications. *International Journal of Robotics Research*, 24:731–741, 2005.
- [42] Manfred Lau, Masaki Hirose, Akira Ohgawara, Jun Mitani, and Takeo Igarashi. Situated modeling: a shape-stamping interface with tangible primitives. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, pages 275–282, New York, NY, USA, 2012. ACM.
- [43] Manfred Lau, Akira Ohgawara, Jun Mitani, and Takeo Igarashi. Converting 3D furniture models to fabricatable parts and connectors. *ACM Trans. Graph.*, 30(4):85:1–85:6, August 2011.
- [44] Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. ShadowDraw. *ACM Trans. Graph.*, 30(4):27:1–27:10, July 2011.
- [45] Marc Levoy, Jeremy Ginsberg, Jonathan Shade, Duane Fulk, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, and James Davis. The digital Michelangelo project. In *Proceedings of the 27th annual conference on computer graphics and interactive techniques*, pages 131–144, New York, New York, USA, July 2000. ACM Press.
- [46] LG. LSM-100. <http://www.lg.com/ae/it-products/external-hard-disk/LG-LSM-100.jsp>.
- [47] Wilmot Li, Maneesh Agrawala, Brian Curless, and David Salesin. Automated generation of interactive 3D exploded view diagrams. *ACM Trans. Graph.*, 27(3):101:1–101:7, August 2008.

- [48] Wilmot Li, Lincoln Ritter, Maneesh Agrawala, Brian Curless, and David Salesin. Interactive cutaway illustrations of complex 3D models. *ACM Trans. Graph.*, 26(3), July 2007.
- [49] MakerBot Industries. MakerBot. <http://www.makerbot.com/>.
- [50] Mako Surgical. RIO Robotic Arm Interactive System.
- [51] James McCrae, Karan Singh, and Niloy J. Mitra. Slices: a shape-proxy based on planar sections. *ACM Transactions on Graphics (TOG)*, 30(6):168, December 2011.
- [52] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, November 2006.
- [53] Yuki Mori and Takeo Igarashi. Plushie: an interactive design system for plush toys. *ACM Trans. Graph.*, 26(3):45:1–45:7, August 2007.
- [54] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fiber-Mesh: designing freeform surfaces with 3D curves. *ACM Transactions on Graphics (TOG)*, 26(3), 2007.
- [55] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. A sketch-based interface for detail-preserving mesh editing. *ACM Trans. Graph.*, 24:1142–1147, 2005.
- [56] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.
- [57] Pennsylvania Academy of the Fine Arts, S Danly, and C Leibold. *Eakins and the photograph: works by Thomas Eakins and his circle in the collection of the Pennsylvania Academy of the Fine Arts*. Published for the Pennsylvania Academy of the Fine Arts by the Smithsonian Institution Press, 1994.
- [58] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *Proceedings of the 6th Annual International Confer-*



- ence on *Mobile Computing and Networking*, MobiCom '00, pages 32–43. ACM Press, August 2000.
- [59] Pro/ENGINEER. Parametri Technology Corporation. 2009.
- [60] Ramesh Raskar, Greg Welch, and Henry Fuchs. Spatially augmented reality. In *In First IEEE Workshop on Augmented Reality (IWAR98)*, pages 11–20, 1998.
- [61] Alec Rivers, Fredo Durand, and Takeo Igarashi. A Comparison of 3D Modeling with Silhouettes with Google SketchUp. Technical report, Massachusetts Institute of Technology, 2010.
- [62] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3D model acquisition. *ACM Trans. Graph.*, 21(3):438–446, July 2002.
- [63] Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. SketchChair: an all-in-one chair design system for end users. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '11, pages 73–80. ACM Press, January 2011.
- [64] R Schmidt, B Wyvill, M C Sousa, and J A Jorge. ShapeShop: Sketch-Based Solid Modeling with BlobTrees, 2005.
- [65] Dorée Duncan Seligmann and Steven Feiner. Automated generation of intent-based 3D Illustrations. *SIGGRAPH Comput. Graph.*, 25(4):123–132, July 1991.
- [66] Ed Sells, Zach Smith, Sebastien Bailard, Adrian Bowyer, and Vik Olliver. RepRap: The Replicating Rapid Prototyper-maximizing customizability by breeding the means of production. *Handbook of Research in Mass Customization and Personalization*, 1:568–580, 2009.
- [67] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K. Hodgins. Motion capture from body-mounted cameras. *ACM Trans. Graph.*, 30(4):31:1–31:10, July 2011.
- [68] ShopBot Tools. ShopBot. <http://www.shopbottools.com/>.

- [69] Christopher Skeels and James Rehg. ShapeShift: a projector-guided sculpture system (Poster). *UIST 2007 Poster Session*, 2007.
- [70] SketchUp. Google. 2009.
- [71] SolidWorks. Dassault Systemes. 2009.
- [72] O. Stava, J. Vanek, B. Benes, N. Carr, and R. Mech. Stress relief: improving structural strength of 3D printable objects. *ACM Trans. Graph.*, 31(4), 2012.
- [73] Richard Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, January 2006.
- [74] Russell Taylor, Aaron Barnes, Rajesh Kumar, Puneet Gupta, Zhengxian Wang, Pat Jensen, Louis Whitcomb, Eugene DeJuan, Dan Stoianovici, Louis Kavoussi, and Pat Jensen Louis Whitcomb. A Steady-Hand Robotic System for Microsurgical Augmentation. *International Journal of Robotics Research*, 18:1201–1210, 1999.
- [75] Techno CNC Router Systems. TechnoVision.
- [76] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment - A modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice, ICCV '99*, pages 298–372. Springer-Verlag, 2000.
- [77] Ivan Viola, Mario Costa Sousa, David Ebert, Bernhard Preim, Bruce Gooch, Bill Andrews, and Christian Tietjen. Eurographics tutorial on illustrative visualization for science and medicine, 2006.
- [78] Anthony Webster, Steven Feiner, Blair Macintyre, William Massie, and Theodore Krueger. Augmented Reality in Architectural Construction, Inspection, and Renovation. In *In Proc. ASCE Third Congress on Computing in Civil Engineering*, pages 913–919, 1996.
- [79] Greg Welch and Eric Foxlin. Motion tracking: no silver bullet, but a respectable arsenal. *IEEE Comput. Graph. Appl.*, 22(6):24–38, November 2002.

- [80] Martin Georg Zahnert, Erik Fonseca, and Alexander Ilic. Handheld Scanner with High Image Quality. U.S. Patent US 2010/0296140 A1, 2010.
- [81] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. SKETCH: An Interface for Sketching 3D Scenes. *International Conference on Computer Graphics and Interactive Techniques*, 1996.
- [82] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, October 1994.
- [83] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.