IMPROVING SCULPTING VIA CROWDSOURCING

by

Matthew K. Sabol, M.S.
A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Master of Science
Computer Science

Committee:

Dr. Yotam Gingold, Thesis Director

Dr. Jim X. Chen, Committee Member

Dr. Zoran Duric, Committee Member

Dr. Sanjeev Setia, Chairman, Department of Volgenau School of Engineering

Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering

Date: May 2, 2014

Spring Semester 2014
George Mason University
Fairfax, VA

Improving Sculpting via Crowdsourcing

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

By

Matthew K. Sabol, M.S.
Bachelor of Science
George Mason University, 2004

Director: Dr. Yotam Gingold, Professor
Department of Volgenau School of Engineering

Spring Semester 2014
George Mason University
Fairfax, VA

Copyright © 2014 by Matthew K. Sabol, M.S.
All Rights Reserved

Dedication

This is dedicated to my loving wife Soo Mi, my two wonderful children Micah and Eva, and my wonderful parents.

Acknowledgments

I would like to thank the many friends, relatives, and supporters who have made this happen. My loving wife, Soo Mi, assisted me in my research. My son and daughter helped me with model analysis. Dr. Yotam Gingold was of invaluable help through the instructing, coding, experiment, and analysis process.

Table of Contents

Page

List of Figures . . . vii

Abstract . . . ix

1 Introduction . . . 1

1.1 Contributions . . . 2

1.2 Overview . . . 2

2 Related Work . . . 4

2.1 Modeling Tools . . . 4

2.2 Aggregation/Weighted Average . . . 6

2.3 Visualization . . . 6

3 Collecting the Models . . . 8

3.1 Modeling Tool . . . 8

3.1.1 Training . . . 8

3.1.2 Availability . . . 10

3.1.3 Flexibility . . . 11

3.1.4 Scalability . . . 12

3.1.5 Extensibility . . . 13

3.2 Default Voxel and Rotation Action Differences . . . 14

4 Analyzing Models . . . 16

4.1 Median Analysis . . . 16

4.2 Weighted Average . . . 17

4.2.1 Crowdsource Defined Weights . . . 17

4.2.2 Generated Weights . . . 18

4.2.3 Aligning Shapes . . . 18

5 Infrastructure . . . 21

5.1 UI/UX . . . 21

5.2 Crowdsourcing . . . 24

5.3 Technology Stack . . . 25

6 Preliminary Experiments . . . 26

6.1 Comparing Additive and Subtractive Sculpting 26
6.2 Comparing Automatic and Manual Rotation 30
6.3 Scaling with Complexity 35
7 Perceptual Evaluation 38
7.1 Experiment 38
7.2 Complexity 41
7.3 Additive vs Subtractive 47
7.4 Automatic vs Manual Rotation 50
7.5 A Final Look at the Aggregate Models 54
8 Future Work 56
A An Appendix 59
Bibliography 62

List of Figures

Figure Page

2.1 Voxels [1] 4

2.2 Structured Annotations [2] 5

2.3 Various options were considered for Digital Clay 5

4.1 Subject averages to determine isosurface. 17

4.2 Two sets of points representing two shapes can be aligned using Principal Component Analysis [3], even without a correspondence between pairs of points. 19

5.1 Voxel Sculpting Interface 21

6.1 Pear Aggregate Marching Cubes 27

6.2 Interfaces for Additive and Subtractive 28

6.3 An arrow was added to the image on the upper right and to the canvas to indicate direction of the sculpture to avoid confusion and provide a more accurate consensus. 31

6.4 Hits by Gender 32

6.5 Rotate Hits by Rating Sorted by Ascending Complexity 33

6.6 The models of complexity ranged from a simple pear to a more complex Standford Bunny. 35

6.7 Hits by Rating Sorted by Ascending Complexity 35

7.1 Example Likert Survey 39

7.2 The results for all experiments run suggests that the Isosurface aggregates, including pca, for the weighted and unweighted were chosen by crowdsourcing to show better resemblance to the source. 40

7.3 No Complexity 42

7.4 Low Complexity 43

7.5 Mid Complexity 44

7.6 High Complexity 45

7.7 High Complexity Best Aggregate Isosurface 46

7.8 The results for the Pear auto additive suggests that the Isosurface aggregates, including PCA, for the weighted and unweighted were chosen by crowdsourcing to have better resemblance to the pear. 47
7.9 Auto-Rotate Additive Voxel vs Isosurface vs Isosurface PCA 48
7.10 Pear Aggregate Marching Cubes 49
7.11 Pear Individual Contribution Voxels 51
7.12 Voxel Highest Rated Consensus Models 52
7.13 Isosurface Highest Rated Consensus Models 53
7.14 Consensus Models Boxplots 55
8.1 Iterative Refinement applied to a car 56
8.2 An aggregate knight in SculptGL [4] 57
A.1 Pawn Auto Rotate Voxels 59
A.2 Knight Auto Rotate Voxels 59
A.3 Knight Manual Consensus Iso 59
A.4 Car Auto Rotate Voxels 60
A.5 Flower Manual Rotate Voxels 60
A.6 Aggregate Car with PCA Iso with rating of 3 or higher and weight better than 1 61

Abstract

IMPROVING SCULPTING VIA CROWDSOURCING

Matthew K. Sabol, M.S.

George Mason University, 2014

Thesis Director: Dr. Yotam Gingold

Crowdsourcing and the wisdom of crowds [5] have established that a large number of low quality inputs can be reliably aggregated to yield a higher quality output. This thesis examines whether the wisdom of crowds can be extended to sculptures or three-dimensional models. To do so, we discuss the design of a 3D modeling tool requiring no training and make use of it to crowdsource the creation of hundreds of sculptures from nearly a thousand individuals. We introduce several techniques for aggregating low quality 3D models. Finally, we performed a perceptual study to evaluate the quality of input and aggregate models. We report our findings related to additive versus subtractive modeling, automatic versus manual rotation, and our various aggregation techniques.

Chapter 1: Introduction

Curves, surfaces, and images are at the core of computer graphics and all applications of digital geometry processing. Currently, approaches for creating and editing curves, surfaces, and images are limited by the expertise of the participant. We explore an approach to overcome this limitation based on crowdsourcing and the wisdom of crowds [5], through which a crowd can often surpass the skills of an individual.

A fundamental task in computer graphics and computer-aided design is the creation of the 2D and 3D shapes, made up of curves and surfaces. Curves and surfaces are the basic building blocks of drawings, paintings, and sculptures. The quality of a shape is limited by the abilities of its creator, either at the time of initial creation or during later editing. This is particularly apparent as a result of the growth in participant-generated (amateur) content. Most people can draw, paint, or sculpt at a basic level, but expert performance is difficult to achieve. Even individuals with some expertise find greater expertise is difficult to achieve. For example, everyone can draw, and yet drawing well (precisely or "beautifully") requires a great deal of skill and practice [6]. At the same time, individuals with motor disabilities, whether resulting from injury, stroke, Parkinson's disease, or otherwise may find that performance at even a basic level is challenging or impossible to achieve.

This thesis examines whether the wisdom of crowds can be extended to sculptures or three-dimensional models. Current practice, while informed by real-world observations and insights [2, 6–12], often follows the traditional, non-computer mediated process of shape creation and produces output limited by the skill of the individual.

1.1 Contributions

Our work is the first large-scale evaluation of aggregate 3D models that we know of. We crowdsourced the creation of hundreds of sculptures from 802 unique individuals. Our specific contributions are

1. Algorithms for robustly aggregating low-quality 3D models to compute a consensus average (Section 4.2.1).

2. An investigation of design decisions for a naive model system, namely additive versus subtractive modeling and manual versus automatic rotation.

3. A perceptual study evaluating the design decisions and aggregation algorithms.

1.2 Overview

Consensus model generation require several processes for acquisition, adjustment, and appearance. Acquisition is the process to which data is gathered. It is important to then analyze and identify the best tool, establish optimal participant interface techniques, and determine methods for gathering quality models. Adjustment is the process to which acquired models are weighed and aligned. Appearance is the process in how adjusted models are displayed.

Acquisition. We wish to know if it is possible for untrained individuals to make quality 3D models without training through crowdsourcing. Multiple methods of sculpting are available and for untrained participants which is thoroughly described in Section 3.1. Section 5.1 presents our interface for performing this task via crowdsourcing, using naive participants over the Internet. We find that they generally can, up to limits in the complexity of the shapes (Section 6.3).

The history of sculpting has provided multiple methods of creating 3D art including chiseling, molding, and rotating (i.e. clay pots). We experimentally evaluate methods for gathering 3D model from untrained individuals (Section 3.2). We wish to know if we get

better results from building (additive) or chiseling (subtractive), which we discuss in Section 6.1. Section 6.2 evaluates whether to automatically rotate the 3D model while it is being created. We present the results of a perceptual study evaluating the quality of all models and their aggregates (Chapter 7).

Adjustment. We explored several techniques for aggregation. A simple binary technique is discussed in Section 4.1 that looks at the median of contributed models. We developed weighted averaging techniques based on human evaluators (Section 4.2.1) and system generated weights (Section 4.2.2). To correct models that are oriented in different directions and to appropriately adjust the overall scale of models, Section 4.2.3 presets an alignment technique based on Principal Component Analysis.

Appearance. Certain aggregate techniques are be best displayed in a simple binary appearance while others use a more complex appearance that benefits from the small changes between models. We describe the design of our perceptual study for rating individual models and aggregates in Section 7.1. Results related to model complexity are presented in Section 7.2.

Chapter 2: Related Work

2.1 Modeling Tools

A variety of approaches to 3D modeling have been proposed in the literature.

Voxels.

Mr. Doob supports a simple editor tool [1] using the ThreeJS framework to showcase the creation of art using voxels within a browser. This tool provides an interface based on simplistic controls: a simple point and click method to create voxels, shift click to remove, keystrokes 0-9 to define color and mouse drag to rotate. Additionally, basic option of "Share" to share a link to a friend, "Save" to save a png image, and "Clear" to remove all voxels are provided. This tool was used as the initial building blocks of the first experiments conducted with several alterations to improve the usability and participant experience. Further improvements required a complete refactor/redevelopment of the framework to provide better rotation control, form capture, abstraction of different images and textures, direction indication, a panel to enable add and remove rather than click and shift-click, a panel that allows coloring options, and "experiment-appropriate" instructions. The modified editor also needed support to return and save results to Amazon's Mechanical Turk and to the analysis database. The voxel analysis tool used to analyze the aggregate was based on the content produced by the modified voxel editor tool.

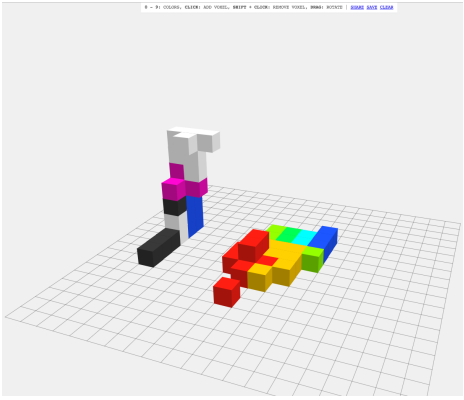


Figure 2.1: Voxels [1]

2D to 3D.

Teddy 1999 [13] with an improved Smooth Teddy in 2007 that allowed for a participant

to free-form stroke a 2D drawing and the system creates a 3D model with various limitations. We found this approach to be easy to use but did not have the scalability for more complex models.

Structured Annotations for 2D-to-3D Modeling 2009 [2] allows for the participant to annotate a 2D drawing in order to create a 3D model. This is a huge improvement to scalability and provides an excellent participant experience with increased usability. This tool would be an excellent candidate for future work.

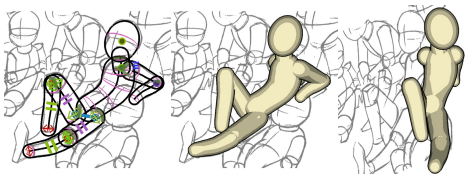
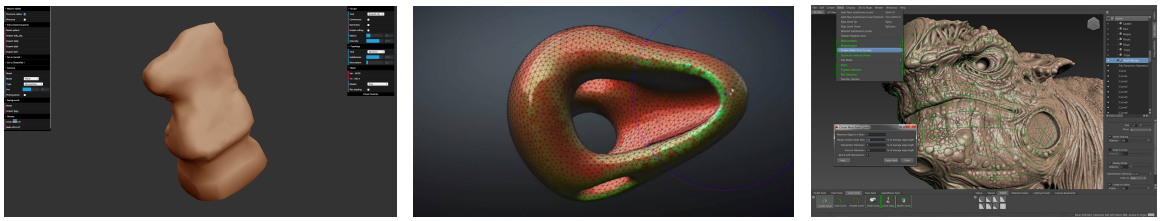


Figure 2.2: Structured Annotations [2]

Geosemantic Snapping for Sketch-Based Modeling 2013 [14] allows for a sketch with a drag and drop approach to object creation. Though this is very simple for a crowdsource community to use in a relative short amount of time, the averaging would not lend itself well to using similar primitives to a simple model. More complex models may prove the averaging techniques to be useful but would be harder for a participant to accomplish in a short amount of time.

3D Sculpting / Digital Clay.



(a) SculptGL [4] (b) Freestyle [15] (c) Mudbox [16]

Figure 2.3: Various options were considered for Digital Clay

Mudbox [16] initially created in 2007 is an excellent tool but it was not used in these experiments because it requires more time and training than the other tools. SculptGL [4], a variation based on Mudbox, was considered because of the availability as it is WebGL, however, the training requirement was a great concern to the goal of this project.

Freestyle: Sculpting Meshes with Self-Adaptive Topology [15] is a further evolved version

of Mudbox that maintains volume as the digital clay is molded. This is not yet available for consumption and also requires training for use.

Polygon Modeling Tool.

Blender [17] was considered when thinking of Polygon Modeling Tools since it is free to download for participants, however, since it wasn't a controlled environment that could be used easily for this experiment, it was thrown out.

Maya [18] is a very well known tool used in the industry for 3D modeling, however, given the costs and the amount of training required in addition for issues found with Blender, we thought this would be a very poor choice for this experiment. If we were looking at modeling experts' average, it may be something to consider.

Proprietary Tool.

Spore monster creator [19] was one of the free tools that was made for easy use for the game Spore. However, because there isn't enough variance in the tool to get ans average in the true sense of points and polygons, this was not chosen.

2.2 Aggregation/Weighted Average

We propose an algorithm that is very similar to *Weiszfeld's algorithm* for computing the geometric median of a collection of points [20, 21]. Here we can apply various weights based on this foundation.

One of the weighted averaging techniques we followed closely was the Bayesian Truth Serum [22] in the way we looked at the difference from the average and individual submissions to further weight the average systemically.

2.3 Visualization

Various visualizations were used in the analysis of the averages as well as weights of the results of the crowdsource experiments.

Marching Cubes [23], Marching Tetrehedra [24], and Constrained Elastic Nets [25] are used in the analysis of aggregated point cloud results through surface reconstruction.

Poisson Reconstruction [26] was considered, but a real-time engine with provided voxel type grid points offered several unnecessary steps. For future work this may be necessary.

Depixelizing Pixel Art [27] was considered, especially for groups and colors with similar traits for a 3D implementation. Since the same colors are used throughout, there isn't a need to create a Depixelizing Voxel Art rendition.

Chapter 3: Collecting the Models

3.1 Modeling Tool

A proper analysis of alternatives to select the best modeling tool for crowdsource sculpting would build it's foundation on several key ideals. The modeling tool should...

1. Require as little training as possible.
2. Be available to a diverse audience, ignoring skill set as well as other diversity considerations.
3. Be flexible as to what kind of models can be created.
4. Be able to scale in complexity.
5. Be extensible, meaning it can export a usable file format for consumption.

These ideals are not requirements in the true sense, as some are not as quantitatively measurable or simply difficult to achieve.The correct tool attributes many of these ideals as possible.

3.1.1 Training

The modeling tool should require as little training as possible. Crowdsourcing requires an audience of participants that are capable of producing results. If the participants cannot produce the results, then the analysis has noise in the data set. In order to reduce data noise, the participant needs to have training or a tool that is intuitive enough where the participant can simply start producing quality work with the tool. In experiments conducted, one of the main goals was to have the participants spend little time creating a

model. This rules out any sort of training outside instructions that can be given or small tutorials provided with the tool itself. It's also understood that computer knowledge in general is required.

Voxels. Blocks are something that many humans are comfortable with even at an early age. Children are provided blocks at a toddler age to begin building and playing with. Many young children are familiar with LEGOs and enjoy the simplistic art provided by cubes. Recently, Minecraft held captive the minds of many children, teenagers, and adults where all objects in the game are made of or can be built with blocks. Simply allowing a participant to point and click to add or remove a block plays to the experience many participants already have. This makes it an excellent choice for a sculpting tool.

2D to 3D. Drawing has defined human expression since the dawn of man, capturing ideas and concepts of the heart. Children, without training, know how to draw stick figures and shapes. Through growth, they learn to add color, shading, lines of perception, and much more. Computer scientists have sought to take advantage of drawn art for the purpose of converting it to it's corresponding 3D model. If there was a way to read the art or convert it directly to a 3D model, this could reduce training. The current tools that provide this typically require a small amount of training or perhaps a set of tutorials to bootstrap an unskilled participant.

3D Sculpting / Digital Clay. As humans created more tools to help them with daily activities, clay became a prominent medium for simple structured tools. Being able to use hands to mold, bend, flatten, push and pull became common to the human experience. Having something like this also allows for detailed work on these models, giving more control to the participant. There are digital tools that allow for this sculpting approach and are used in the professional 3D modeling world. However, this tool requires much more time to accomplish similar tasks done with other tools for simple models and requires a level of training to properly understand/use unlike the physical manifestation of clay.

Polygon Modeling Tool. Polygon modeling tools are used for many production ready applications supported by many software companies. The tools typically require a degree

in art with a specialization in computer graphics or self taught knowledge. The main issue is that the amount of time required to attain the knowledge to use one of these tools effectively would prove to be problematic. Even experts in the field may not be able to produce certain models in 10-20 minutes of time. The amount of training and time it would take to accomplish the task would make it difficult to choose polygon modeling tools for crowdsource purposes.

Proprietary Tool. There are several tools that companies provide to help create monsters or levels for their games. These tools are some of the most intuitive products as it is aimed to get non-graphic artists to create attractive 3D models. Most participants would not spend more than 10-20 minutes to produce a nice sculpture and many of these tools are free with the expectation they are used for the sold game. The ease of use and little training required to use the tool makes proprietary tools attractive.

3.1.2 Availability

The modeling tool should be available to a diverse audience. The modeling tool should be available to a wide variety of participants and not for specific medium. In this section we discuss the different media available to participants for 3D modeling.

Computer Software. The majority of 3D modeling tools require software to be installed or run from scripts on the host operating system. The problem with requiring participants to run software separately on their client computer is that it's hard to setup appropriate controls for crowdsourced submission and the community willing to install, run, and submit data based on the software would require a higher skill set and cost valuable time. Additionally, the percent of people willing to do it would reduce this number further. In a tightly controlled environment such as a company or a classroom where participants could be forced to install software after accepting a hit, this approach may work well but would sacrifice diversity.

The Modern Browser. Unlike other 3d modeling tools, HTML5 canvas or WebGL based editors do not need a special installation because it can be run directly from any

modern browser. This means the potential audience is vastly larger than that of other 3d modeling tools that require a special download. This also allows us to have the same interface setup as an experiment control; allowing us to set a time limit, validate the model work, and ensure a similar participant experience.

3.1.3 Flexibility

The modeling tool should be flexible as to what kind of models can be created.

Voxels. Since voxel based editing is mostly point generation, certain kinds of model shapes don't necessarily give an advantage to the workflow. Symmetrical and asymmetrical objects with different curves or features can be handled by a voxel editor but look jagged and require a great deal of experience of using pixels or voxels to accomplish. The higher resolution resulting in more voxels produces a better curve with increased effort. This is similar to other tools, higher complexity requires more effort.

2D to 3D. Unlike voxels, 2D to 3D modeling excels at curves. Most of the examples provided by the various tools that use this method also use characters or objects with rounded ends. Symmetrical models have a huge advantage in this paradigm as bending the object equally on two sides is made simple. Asymmetrical objects can be broken into symmetrical parts and handled relatively easily. Since the tool is based on lines and not voxels, the resolution handling is made easy, without concern for a pixelated or voxelated look. It is difficult to create square edged shapes and sharp corners with this tool. Additionally, high detail work can be accomplished with much increased effort and skill.

3D Sculpting / Digital Clay. This tool excels at creating detailed work of existing shapes and models. Typically the default shape is a sphere, allowing the user to pull and push portions of the sphere to create an object. This will lend itself to create objects with rounded edges as well as only simple modifications of the existing object. Digital Clay's method does not deal well with any task that requires components or parts, square or sharp edges, and with strict proportions.

Polygon Modeling Tool. Professional polygon editors provide the most flexibility as

to what kind of models can be created. These tools were designed to allow expert graphic modelers to design any models feasible with modern computer graphics today. Companies provide a great deal of resources into the production that goes into these tools and the products that result from these tools. The issue is training and cost for expert designers to develop relatively simple models. These tools and render high detail work, smooth edges, sharp edges, and much more.

Proprietary Tool.

3.1.4 Scalability

The modeling tool should be able to scale in complexity.

Voxels. The biggest problem with a voxel tool is that the effort/complexity ratio increases exponentially. When more voxels and parts to the model are required, a significant amount of time and skill are required to gauge the resulting model properly. Initially the task seems feasible but after some time of work, it is obvious that the work is tedious and seems to produce worse results. There are ways to mitigate this, such as iterative refinement or simply using a big general picture with the voxel editor and exporting it to another tool that is optimal for detailing.

2D to 3D. 2D to 3D tools are excellent at creating general shapes/sizes from a 2D source. Some of these tools have greatly improved detail work. When the source becomes more complex, for most types of objects, this method is generally pretty good. Unfortunately, the detail work does end up being cumbersome and generating asymmetrical objects with sharp edges becomes difficult with frequency.

3D Sculpting / Digital Clay. Digital clay tools scale very well with detailed complexity. They have a significant amount of control with regards to adding detailed work to existing models. Many professional artists use a digital clay type tool to add the detail work after generating general objects in another tool. Digital clay, however, does not scale well with large, multifaceted objects as it is burdensome to add large new parts with specific proportions. As more of these features exist, the complexity of the model creation worsens

significantly.

Polygon Modeling Tool. Polygon modeling tools are probably the best in scalability as any object can be made utilizing built in tools. There are great sets of tools that help with proportion, details, shading, symmetry, and exporting usable data for various purposes. The main issue here is that these tools require a great deal of expertise to use initially. Once these skills are acquired, complex model creation scales well.

Proprietary Tool. This is a very broad set of tools, some that scale incredibly well with complexity and some that only create a certain set of models specific to the corresponding game. Generally proprietary tools scale in complexity very well in the context for which they were made.

3.1.5 Extensibility

The modeling tool should be extensible. The modeling tool should allow for other technologies to consume it so that the content generated is useful. In many cases, extensibility specifically describes the code within software. Though that is an accurate definition, the goal here is to make sure that other software can reuse the consensus models.

Non-Proprietary Tools. Depending on the different types of available tools, most open source or purchasable software intended for modeling allows exports of universally accepted file types. Included file types are .fbx, .obj, and others. Since most software accepts obj as an import type, this was the desirable option.

Proprietary Tools. The biggest problem with proprietary tools is that they generate a file format whose purpose is meant for their software. The Spore Creature Creator and Starcraft II editor, for example, uses a file format that specifically is for their corresponding video game. Additionally, when the game is dated, the file formats used will be further obsolete. Some purchasable software also uses it's own file types. Included in these file types are .Max and .Blend. Proprietary file types makes it difficult to load data for aggregate purposes and the data structure may not make it easy to perform aggregates in general.

3.2 Default Voxel and Rotation Action Differences

This section is used to discuss the various default actions and rotation type used in the experiment. Rotating manually and rotating automatically gave two different results. Starting with a default of adding blocks or a default of blocks existing requiring participants to remove also yielded very different results.

Default Rotation: Automatic vs Manual.

Manual, as described int he UI/UX section, provides participant full control of the rotation with no automatic rotation after the introductory tween rotation. This provided a more standard 3D modeling tool behavior, giving the 3D designer full control of their view. Some of the models were amazing from the manual rotation options, specifically the Stanford Bunny and the car. This initially lead the experiment to be removing bad outliers to focus on the quality models.

Automatic as described in the UI/UX section, provides a participant control over the constantly rotating object. For the radially symmetric objects, this seemed to be the best approach as the participant could use the tool like a pottery table. With apprehension, the stairs experiment was run with automatic rotate. Surprisingly, this had better results overall. After looking through the variations of different experiments run, it was the conclusion that inexperienced participants understand the 3D nature of the object better with a constantly rotating interface than with a static one. In each individual model, the results typically have more mistakes and minor issues. As a collective, the results are better based on the participant rating and the system generated weights. The goal of the thesis was to see if there was a way to generate more beautiful models from a calculated weight or average of different human work on the same task. The weighted average result definitely was stronger with an automatic rotation.

Default Action: Additive vs Subtractive

Additive was the initial experiment run as the tool we heavily modified, Voxels [1], used a blank canvas with a tree, clouds, and some other small objects to demo the functionality in additive mode. Following this approach, we provided the participant with a blank canvas

and asked the participant to add voxels to create a model of the image. The results varied in size, location, dimensions, and many simply created random objects. This became highly problematic since the tool wasn't accurate at explaining what was expected, not necessarily in lack of words but lack of intuition. Additionally, it was difficult to average models directly as there could be orphan voxel groups or models with holes.

Subtractive, from a UI/UX perspective, seemed to be the best approach. The rational behind this is that it gives a subconscious dimensional boundaries, suggesting that the participant should only manipulate what is available in the existing block. The results from multiple participants results in aligned models with a generally correct rotation and scale. Subtractive mode's preset block fundamentally gives a suggestion of expectations to the participant. From the results of the experiments and aggregation, the conclusion is that subtractive makes the experiment easier on the participant end and reduces noise on for aggregate computations.

Chapter 4: Analyzing Models

4.1 Median (Binary) Analysis Model

The binary or median analysis of the data model displays a voxel when the chosen threshold $\bar{T}$ indicates that at least that percentage of participant provided models have a voxel at location **vp** for all models i in experiment n. This would mean that it is also true that threshold T is greater than the percentage of voxels missing at location **vm** for all models i in experiment n. This is described in equation 4.1 below.

$$\sum_{i=1}^{n} \frac{\mathbf{vm}_i}{n} < \bar{T} <= \sum_{i=1}^{n} \frac{\mathbf{vp}_i}{n} \tag{4.1}$$

This effectively allows the analysis of each experiment to see a full range of thresholds between 0 and 1. The optimal range for T that displayed a more realistic model using binary analysis was typically around (0.5,0.8).

More simply, $U_i e 0, 1$

Another variation of this analysis was Opacity Analysis where the opacity α was the average to a power of ρ, chosen by the user to tweak the display. This equation for voxel v averaged for all models in an experiment is described in equation 4.2 below.

$$\alpha = \left(\sum_{i=1}^{n} \frac{\mathbf{vm}_i}{n} \right)^{\rho} \tag{4.2}$$

A simple change to the median analysis of the data model allows us to view the add and remove actions as the median rather than only the results. Being able to see the actions

allows us to see where the users were likely to add and remove, giving us a more accurate view of what each participant collectively did. This is described in equation 4.3 below.

$$\bar{T} <= \begin{cases} \sum_{i=1}^{n} \frac{\mathbf{vp}_i}{n} & \text{if p not in default block} \\ 1 - \sum_{i=1}^{n} \frac{\mathbf{vp}_i}{n} & \text{if p in default block} \end{cases} \tag{4.3}$$

4.2 Weighted Average

4.2.1 Crowdsource Defined Weights

We use various techniques to define weights based on crowdsource feedback. This is a very effective way of starting analysis of "beauty" or "likeness" to the source image.

Binary involves a way for the wisdom of crowds to determine a very rigid way in which models will be included and which will be excluded in the aggregate. In a sense, it allows for the "throwing out" of poor individual contributions where the participant simply quit the task early or tried to beat the crowdsource system to get paid without having to build a model based on the source image. If the value is 1, then the subject believes that the model is any where from a bad representation of the source image to an excellent representation. The results of running a binary crowdsourced defined weight rendered in a positive direction. The average sculpture started to look more like the source image.

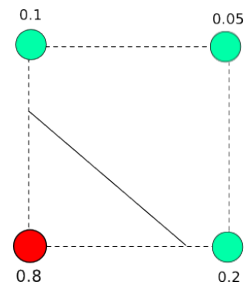


Figure 4.1: Subject averages to determine isosurface.

Likert surveys in the past have been very effective at producing unbiased selection of comparative subjects rendering quality ranked results. Collecting these results using amazon turk provides a qualitative ranking of each model, allowing us to determine a more objective approach to a crowdsource provided weight technique. More is discussed in the Experiments Conducted section. In this section it is important to know that these weights are a good candidate for aggregation. In this system, the subject chooses 1 to 5. The average is then weighted into the Isofunction to which a point is decided to be either inside or outside the

object in order to add further detail to the aggregate model.

4.2.2 Generated Weights

Given a set of aligned models, we use an algorithm closely related to Weiszfeld's algorithm for computing the geometric median of a collection of points [20, 21]. We iteratively compute a weighted average of the aligned shapes and then computes a new weight for each crowdsourced model based on its agreement with the current weighted average. Stated formally, let $U_j(x, y, z)$ represent a collection of models, where U_j is a binary function that maps a point in space to 1 if the point is inside the shape created by crowd participant j, and 0 otherwise. With uniform initial weights for each model $w_j^0 = 1$, the weighted average F^i is defined as

$$F^i(x, y, z) = \frac{\sum_{j=1}^{N} w_j^i U_j(x, y, z)}{\sum_{j=1}^{N} w_j^i}$$

where N is the number of models. For the next iteration,

$$\frac{1}{w_j^{i+1}} = 1 + \iiint \|U_j(x, y, z) - F^i(x, y, z)\|^2 \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z$$

which gives a model a larger weight the closer it is to last iteration's average. This process decreases the weight of outliers in each iteration, and stabilizes to a consensus shape. As opposed to Weiszfeld's algorithm, we penalize according to the square of the distance from the previous iteration's weighted average, and we add a regularization term to account for models which are exactly equal to the previous iteration's weighted average.

4.2.3 Aligning Shapes

Let X and Y be two sets of 3D points representing the shapes created by two participants (Figure 4.2). In the ideal setting, we would have a correspondence between every pair of points $x_i \in X$ and $y_i \in Y$, such as the tip of a screwdriver. Given such a correspondence, we

could solve for the similarity transform that aligns the sets of points by (1) translating and scaling the points to lie within the unit cube and (2) solving for the rotation R that minimizes the so-called Procrustes distance [28] $\sum_i \|x_i - Ry_i\|^2$ via singular-value decomposition (SVD) [29]. With aligned points, we could compute simple coordinate-wise aggregation of the data, such as the average or median.

However, X and Y may not have the same number of points, and the two users may not have chosen to represent the same aspects of the shape. For example, some users may not have created the wheels of a car. The challenge, therefore, is to align data from different individuals without an a priori correspondence.

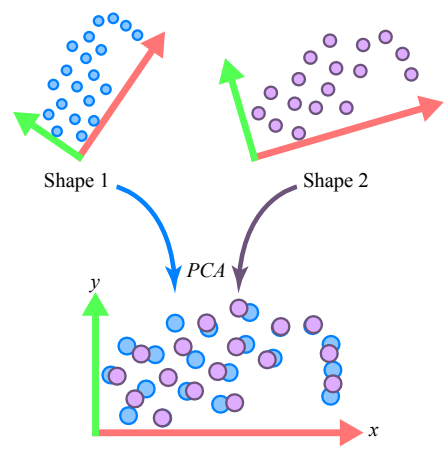


Figure 4.2: Two sets of points representing two shapes can be aligned using Principal Component Analysis [3], even without a correspondence between pairs of points.

One family of techniques for computing automatic shape correspondences are designed for smooth shapes [30–35]. However, we require a consensus shape even after the first iteration of crowdsourcing, at which point we have a collection of extremely coarse shapes.

For the initial coarse registration step, we use to use a data agnostic alignment technique based on Principal Component Analysis [3]. This technique is based on the insight that the directions of maximum variation for a set of points (X or Y from the example above) likely represent similar intrinsic axes of the shapes. The directions of maximum variation can be computed by (1) translating X such that its center of mass is located at the origin—let X' be the translated X—and (2) finding the eigenvectors of $X'^T X'$ sorted according to corresponding eigenvalue. The eigenvectors are guaranteed to be orthogonal to one another. We can align a model by rotating it such that its eigenvectors are aligned with the x, y, z directions and scaling it to fit within the unit cube.[1]

[1]Note that we do not know the sign of the eigenvectors. There are $2^3 = 8$ combinations of positive and negative eigenvectors, but only two are rotations, while the rest are undesirable reflections.

This PCA-based alignment approach produces a similarity transformation that is stable and parameter-free. It is well-suited to our scenario in which we with to mutually align a number of shapes, since it registers each shape to a global coordinate frame. However, we also experiment with non-rigid point set registration techniques [36–38]. These techniques solve for an alignment between two point sets in which the points need not be transformed by the same translation, rotation, and scale. Rather, they generalize to more general transformations which may stretch or deform the models in order to achieve a better alignment.

Chapter 5: Infrastructure

5.1 User Interface / User Experience

When UI/UX is described, it is defined literally as User Interface or User Experience. Many use UI/UX to describe all things related to the GUI, however, this section describes changes made based on the psychology, design decisions, and usability pertaining to the true definition of UI/UX.

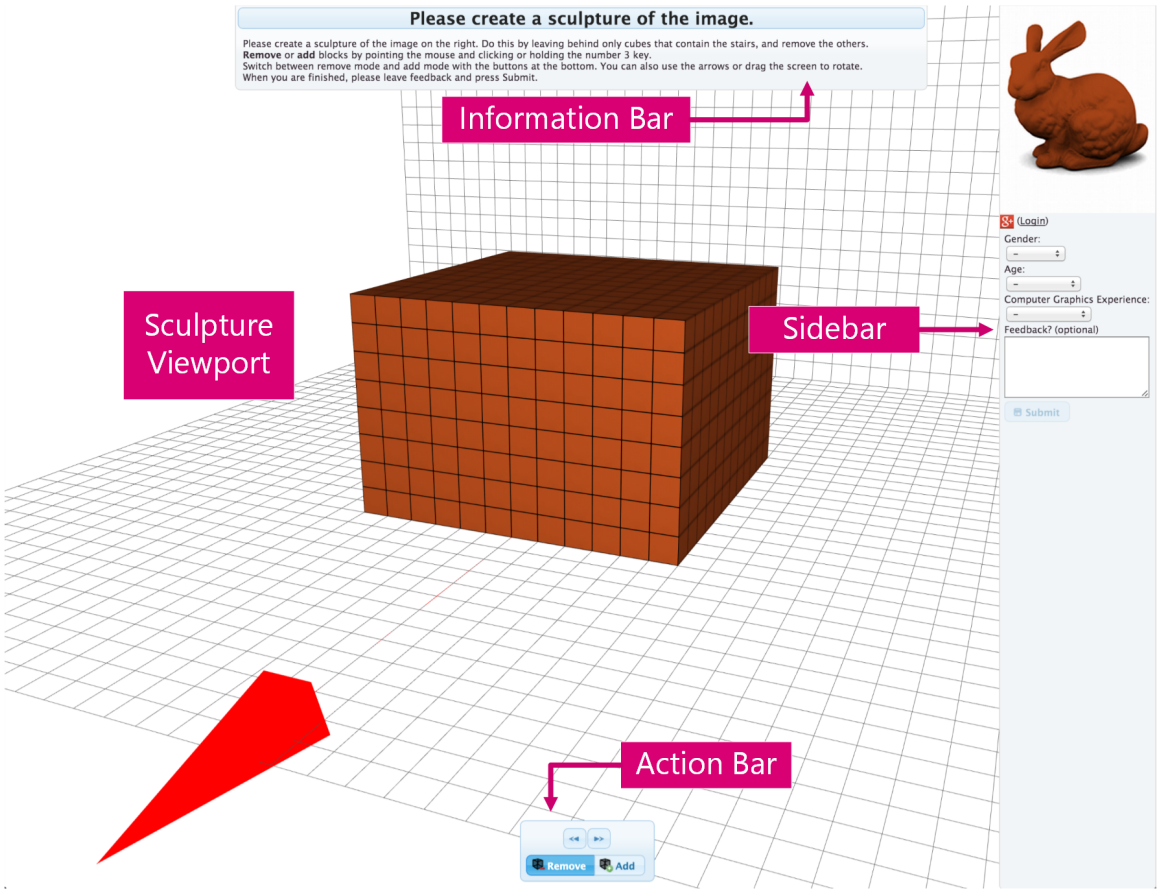


Figure 5.1: Voxel Sculpting Interface

Design Explanation. With any good design, the goal or information of the application needs to be clear, obvious, and intuitive. The most important information of the application is the sculpture. Accordingly, it is the center of the application and takes up the entire window viewport. The sculpture either has an existing default block for the participant to remove or starts with a blank canvas for the participant to built upon. To guide the participant, there are additional supporting structures in this viewport. One of these structures is the grid. The grid lies below the sculpture to show where voxels can be added. In other experiments, there is a back grid to help assist participants define a perceived boundary.

The information bar is on the top of the screen with explanation and instructions to guide the participant. The top of this bar has big picture instruction "Please create a sculpture of the image.". Below the instructions expound: "Please create a sculpture of the image on the right. Do this by leaving behind only cubes that contain the (source name), and remove the others". Below lie more detailed instructions, explaining how to use the action bar and how to submit using the sidebar.

The action bar is directly below with supporting tooltips to explain functionality. Within the action bar, the number of actions are kept to a simple minimum as to reduce confusion. The action bar also changes depending on which experiment is being run. For an automatic rotation experiment, the first three buttons that display as a left arrow, pause/play, and right. The left arrow changes the rotation speed in the left direction, the play/pause temporarily pauses for 10 seconds or continue the rotation, and the right arrow changes the rotation speed in the right direction. Beneath are the remove and add buttons which toggle between remove and add mode. In remove or add mode, when the participant clicks, the action either adds or removes accordingly. Additionally, in add or remove mode, the mouse over changes to guide the participant as to where the new voxel is added or which voxel is removed. For manual rotation, the participant is able to drag to rotate, scroll to zoom, use the left and right arrows to rotate at 90 degree increments using a quadratic tween, and the play/pause button is removed as there isn't a constant rotation.

The side bar is on the right side of the screen. This portion of the screen contains the image that the participant needs to sculpt. Below are participant information including gender, age, and computer graphic experience optional questions. Below is feedback, giving the participant a way to report issues or problems while sculpting the model. Finally is the submit button, which sends information to the server for analysis.

Sculpture Viewport Design Decisions. Several changes were made in the design of the main viewport. Initially the zoom was further out, making the model or grid rather small. After receiving feedback about the difficulty of using the tool with it zoomed out, the decision was made to have the default zoom and position adjusted to make it fit the viewport more appropriately. The next decision that was made was the number of ticks on the grid and the default block size. For each model, the ease of creation based on a certain number of voxels differed. For expected simpler models, less ticks were used and for more complex models, more ticks were used. This helped keep a consistent size of the models. In the same way, the default voxels amount and dimensions changed based on the expected the result, to help guide the participant to create similar sized models. Finally, an arrow was added to help explain the direction expected for directionally sensitive models. Each of these design changes provided better results.

Action Bar Design Decisions. Several changes were made over the course of experiments on the way the action bar worked as well as the placement of the action bar. Initially, the action bar was part of the sidebar. This caused confusion as some buttons effected the main sculpture while others were data entry and submission based. The decision was made to place the action bar directly below the sculpture and reduce the number of options provided to simplify the process. This resulted in better results.

Information and Sidebar Design Decisions. A few changes were made to the information and sidebar. The information bar initially had very minimal information and the sidebar had more detailed information below the image. This proved to be confusing as the information was partially on the top and partially below the image. Additionally, as experiments continued, it was unknown if the UI was only effective to certain groups

or if it was indeed diverse as intended. To help answer these questions, gender, age, and experience were added.

5.2 Crowdsourcing

Amazon Turk and Google OpenID were implemented to retrieve data from participants to apply to a central database. Amazon Turk was used for a large majority of the data. Each hit was setup using a python script. Each hit was also copied to a separate folder so that analysis of the hit in context to the exact tool/experience was preserved. The data was placed in a separate set of folders for each unique hit prior to being loaded to a database. Amazon Turk was the main engine driving the data that we consumed for each experiment. Ensuring the quality and that quality can exist from Amazon Turk was critical. Matthew Crump, John McDonnell, and Todd Gureckis invested substantial research to determine the fidelity of the system for cognitive behavioral experiments [39]. Additionally there are ways to improve the quality of workers for a hit that we used. We ensured that workers had accomplished at least 20 hits and had a 95 percent approval rating. There were more techniques we used as found in recent research [40].

5.3 Technology Stack

MySQL 5.1.67	
Python 2.6.6	
numpy	Numeric Python used for matrix algebra, vector calculations, and various mathematical functions
csv	CSV library for parsing through data coming back from Mechanical Turk Hits
MysqlDB	Library for connecting to the MySQL Database.
PHP 5.3.3	
JavaScript	
jQuery 1.9	JavaScript library to simplify selecting and manipulating DOM elements.
jQuery UI	Interface tools for widgets, buttons, tabs, accordions, and more.
ThreeJS	Provides WebGL support for creating scenes, objects, animation, shaders, and more
flotcharts	Used for plotting charts.
jQuery Knob	An interface widget that replaces a slider for a knob style ui.
Underscore	Used for object and array aggregation and manipulation.
RequireJS	Loads JavaScript and modules asynchronously. This specifically was used for organization and performance.
Tween	Provides an api to control ease in/out animations within JavaScript.
NumericJS	Used for eigenvalues, matrix algebra, and vector calculations.
ImagePicker	Creates a radio button style using images

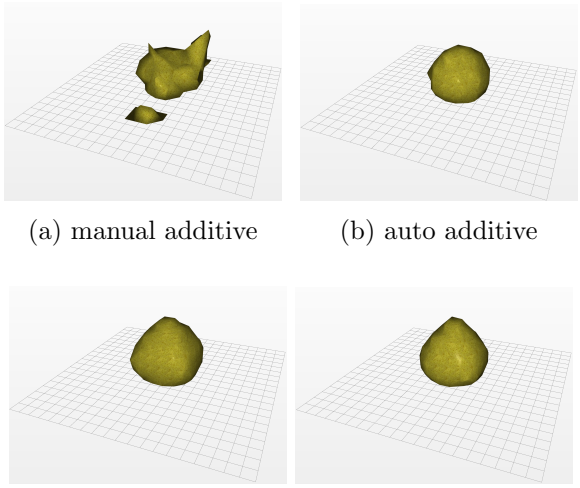
Chapter 6: Preliminary Experiments

The approach for conducting experiments is simple: first gather models via crowdsourcing, secondly determine if the aggregate method generates a consensus model in a way that makes sense, thirdly ensure both methods are scalable, and finally conclude that the consensus model is considerably better than the majority of individual contributions again using crowdsourcing. In gathering the models from a random sample, the method needs to be easy enough for novices to accomplish the task as well as expert sculptures. This means that the first set of experiments must ensure an optimal way to produce usable data in which to aggregate. Secondly, the aggregate model needs to make sense. The methods we'll look at include participant weights, generated weights, isosurface functions to provide better granularity, and aligning the shapes. Thirdly the aggregate method needs to be able to combine data in a way that scales with complexity. We define a complex sculpting task as one that requires more definition, in this case a more refined grid size, and thus more actions to accomplish the task. Finally, we establish a method in which to rate the individual contributions along side the consensus. We use methods recommended by various findings in psychology. This ultimately provides clarity to the question of this thesis and direction for future work.

6.1 Comparing Additive and Subtractive Sculpting

Two sculpting paradigms could be applied and we wanted to determine the one that provided the best Crowdsourcing results. The first concept is that of building or constructing a piece of art. This paradigm would emulate the idea of building an object from basic components to create a model based on the source image. This concept relies on take existing components to build a house, car, and other objects by placing them in the correct order

on a blank canvas. The second paradigm is that of chiseling or the true sense of sculpting. This concept bases it's foundation on an existing block of material where the sculptor would remove the negative space. In several experiments, the goal was to determine which paradigm was best at supporting the aggregation of multiple parallel work of the same source image to a more beautiful object.



(a) manual additive (b) auto additive

(c) manual subtractive (d) auto subtractive

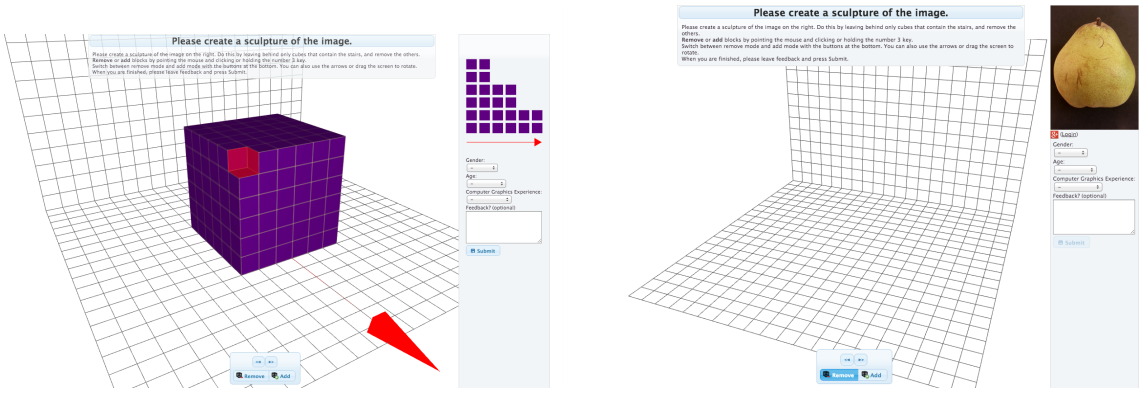
Figure 6.1: Pear Aggregate Marching Cubes

To confirm this, we would need to run a simple experiment based on a simple to sculpt image. Once we received the results, the aggregation would be compared along with how like they were to the original model. Additionally, we would compare the number of models that were tossed because of sculptors who did not follow the guidelines of the task.

Method

General Setup. Following Gingold et al. (2012) [41] we limited the data entry by size, by time, and chose a simple source, an image of a pear. To ensure simplicity only one color could be used in this experiment. Dimensions of the canvas were 20x20 for a sculptor to add or remove points. For the additive experiment, the sculptor had a blank canvas with the default action allowing voxels to be added to the canvas at key ticks, 20 on the x-axis and 20 on the z-axis. The subtractive experiment had a similar setup concerning dimensions but had a default action to remove voxels and included a preset set of voxels in the center of the canvas of 6 wide, 6 tall, and 6 deep. Sculptors were then given ten minutes to remove voxels.

For all experiments, the results were collected and imported for analysis. Each model was ranked by participants on how well the model was like the source image on a scale from



(a) stairs subtractive (b) pear additive

Figure 6.2: Interfaces for Additive and Subtractive

1 to 5 where 1 indicated that there was no resemblance and 5 was an amazing resemblance. Additionally, an automatic weight algorithm calculated the "badness" of how different the sculpture was to the average. If the participant rating was 1, the model was not used as it was determined that the participant did not follow directions. The resulting models were then averaged using a binary median approach, an opacity power based approach, and utilizing marching cubes [23] based on the weighted value of the binary, participant ranked, and system generated weights. Statistics were collected for the participants, the model, the browser type used, and other general statistics.

Participants and Procedure. Participants (manual rotation for n = 50, 25 additive, 25 subtractive; automatic rotation for n = 160, 60 additive 100 subtractive) were asked "Please create a sculpture of the image [a pear] on the right." Additional instructions read "Click/Spacebar to add a box. Shift + Click/Spacebar to remove a box. Drag to rotate." Each submitted a model using Amazon Mechanical Turk and 69 added feedback. It is important to note that in order to prove a later hypothesis, 75 addition participants conducted the automatic rotation subtractive experiment to make the total of 100 participants for this hit.

Results.

To determine whether subtractive or additive was more successful, we looked at the percentages the rated objects as well as the resulting generated aggregate object. For the

additive results using a manual rotation (**Figure 6.1a**), 19 models of 25 (76%) were ranked an average of 1 where ranks 5 had 0, 4 had 2 (8%), 3 had 1 (4%), and 2 had 3 (12%). For the additive automatic rotation results (**Figure 6.1b**), ranks 5 had 5 models (8%), 4 had 3 (5%), 3 had 6(10%), 2 had 13 (22%), and 1 had 33 (55%). For the subtractive manual rotation (**Figure 6.1c**), ranks 5 had 18 models (18%), 4 had 20 (20%), 3 had 22 (22%), 2 had 15 (15%), and 1 had 25 (25%). For the subtractive automatic rotation (**Figure 6.1d**), ranks 5 had 42 (42%), 4 had 15 (15%), 3 had 14 (14%), 2 had 22 (22%), and 1 had 7 (7%). Based on the rankings, it seems that the additive submissions had a significantly larger participant failure rate than the subtractive submissions. Additionally, the subtractive submission had a much larger percentage of rank 5 models. We believe this is the case because the preset block gives a set of expected dimensions, location (translation), and a general sense of direction where as the free form addition does not intuitively guide the participant.

The aggregated models reinforce these findings as seen in Figure 6.1. The aggregate model for additive manual rotate does not fill the necessary locations of a model to properly represent a pear. The automatic additive does slightly resemble a pear. Both subtractive submissions for automatic and manual rotation resemble a pear remarkably well. Analyzing using different visual techniques, the similarities to a pear also reinforce subtractive as the best method. In conclusion, it seems that manual could possibly work, but the best results come from subtractive as the default action for crowdsourced sculpting.

Discussion.

Though some may argue that not enough additive models were tried in comparison to the subtractive models, it is important to know that with the 25 additive vs 25 subtractive run, that these small results supported a subtractive default over the additive. Also, others may argue that providing a additional guides for the additive mode may improve the results much like Gingold's pear experiment [41]. The difference is that a 2-dimensional image was used as the source of a 2-dimensional result. In these experiments, if we were to use a 3-dimensional source, it would defeat the purpose as a voxel model could be easily generated

through a simple algorithm. For this reason, we are using a 2-dimensional source with a 3-dimensional resulting model rather than a 3-dimensional source.

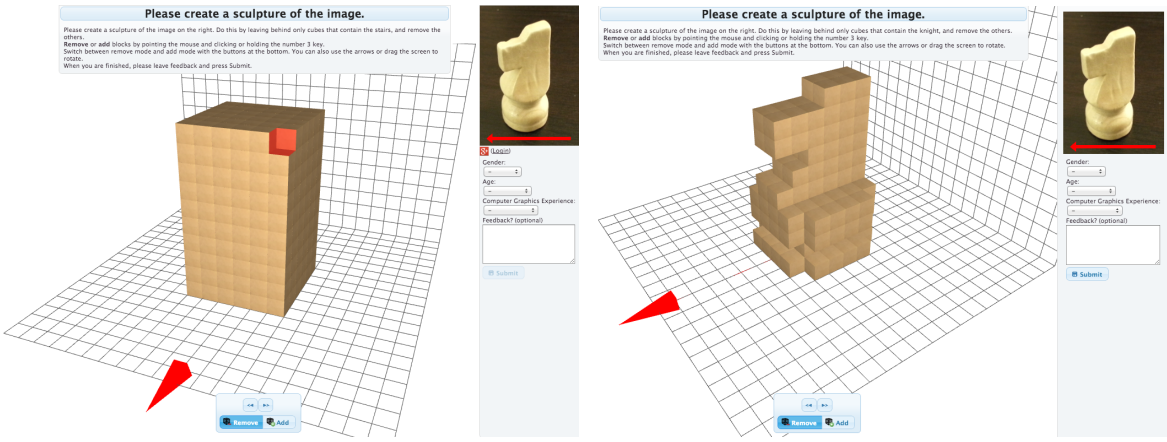
6.2 Comparing Automatic and Manual Rotation

Section 6.1 established that how the editing tool operates can directly influence the sculptors' ability to contribute good models to the aggregate model. We wanted to determine the best way to produce an optimal set of contribution based on the way the model editor navigates about the object. One of the primary concerns was that the participant was using the 3-dimensional tool to produce 2-dimensional drawings of the source image. To improve the contribution to the aggregate model, the new goal was to communicate to the participant that the tool was in three dimension. The best way we thought to handle this was to tie it to the rotational modes of the tool.

There were two modes used to convey the dimensionality of the tool as well as improve intuitive design to produce a better set of contributions to the aggregate model. The first way that the rotation can occur is through a manual process. In this mode, the participant can rotate the tool manually, giving them complete access to the model at any given time. Additionally, it would be important to have an intro spin to let the participant know that the object was indeed 3-dimensional. In the second mode, the rotation is automatic and can only be paused temporarily, modify the rotation direction, and the rotation speed. This second mode would optimally operate like a pottery table and also help the participant recognize that the object is indeed 3-dimensional.

General Setup. The same baseline setup was used for Experiments 2 as was in Experiments 1. In addition to the pear image, multiple other images were used based on symmetry to compare if rotation success was based on the source image symmetry. These images include pawn, knight, car, stairs, flower, airplane, and bunny. For objects that have direction, an arrow was added to the source image and to the 3D canvas.

For automatic rotation, the rotation speed r was set to $r = \frac{\theta(1+x)}{s}$ where $\theta = 45\,^{\circ}$ and



(a) Knight before sculpture begins (b) Knight sculpture completed

Figure 6.3: An arrow was added to the image on the upper right and to the canvas to indicate direction of the sculpture to avoid confusion and provide a more accurate consensus.

$x = 0$ initially and increases or decreases by increments of 0.5 depending on participant interaction. $1 + x$ is never 0 unless the participant pauses the rotation for 10 seconds. The participant is not able to rotate above or below the object in automatic rotation mode. The participant could also not zoom in this mode.

For manual rotation, the participant has complete control over the angle and zoom that the viewport has of the object. When the model first loads, the canvas and preset block rotates 360° using a quintic tween for rotation speed to provide an ease in/out smooth animation. In addition, for participants who cannot drag the screen was provided buttons to allow for manual control of the rotation using quadratic ease in/out animation. A limit was added for the max zoom as well as rotation around the object so that the object should never appear upside down or inverse the mouse controls.

Participants and Procedure. Participants ($n = 808$, 454 automatic and 354 manual) were asked, "Please create a sculpture of the image on the right." If it was an automatic hit, the tool would read, "Please create a sculpture of the image on the right. Do this by leaving behind only cubes that contain the [image name], and remove the others. Remove or add blocks by pointing the mouse and clicking or holding the number 3 key. You can switch between remove mode and add mode with the buttons at the bottom. You can also

pause the rotation (temporarily). When you are finished, please leave feedback and press Submit."

If it was a manual hit, the tool would read, "Please create a sculpture of the image on the right. Do this by leaving behind only cubes that contain the [image name], and remove the others. Remove or add blocks by pointing the mouse and clicking or holding the number 3 key. Switch between remove mode and add mode with the buttons at the bottom. You can also use the arrows or drag the screen to rotate. When you are finished, please leave feedback and press Submit." Each submitted a model using Amazon Mechanical Turk. Feedback was submitted by 443 participants using the tool. For the 231 that reported gender, 157 were male, 73 were female and 1 was other (11 did not report gender and 201 were from hits that did not track gender).

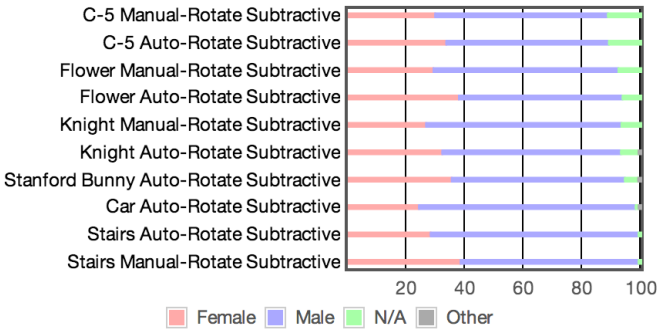
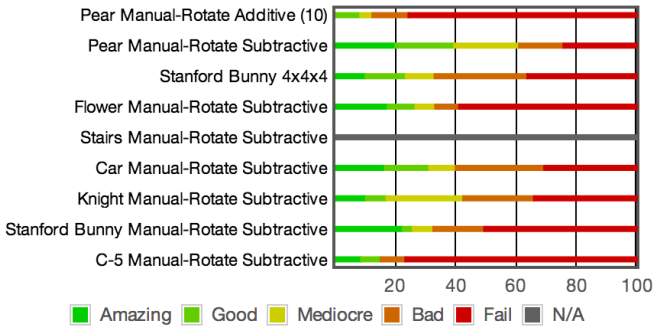


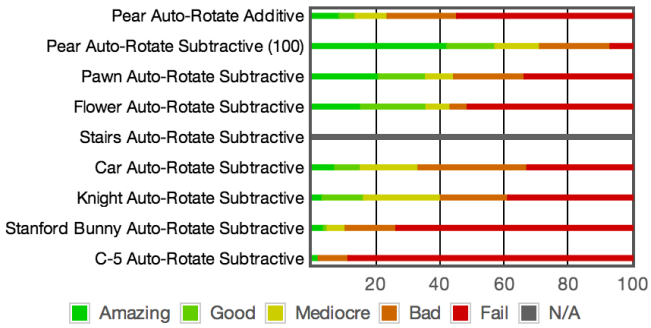
Figure 6.4: Hits by Gender

Results. Considering subtractive only, 34% of manual was ranked at 1 (unusable) where 37% of automatic was ranked unusable. In addition, 15% of manual was ranked at 5 (best quality) while 15% of automatic was also ranked best quality. From a totalistic perspective, this seems to mean that there was little difference between automatic and manual rotate. From a model based perspective, a different set of results may occur. This was particularly found to be true for stairs, a very simple model with a direction as well as bunny, a very complex model with a direction. Interestingly, car, pawn, and knight seemed to have little difference from automatic and manual rotate. For stairs manual rotate subtractive, 48 models (48%) had a consistent incorrect direction. This is problematic for the automatic weight as it is the largest representation for direction. 35% had the correct direction is given reductions in weight as it is different from the average more than the 48%. For stairs automatic rotate subtractive, 73% had the correct direction. This can

be attributed to the idea that participants weren't aware of the arrow on the model or on the image during manual rotation. For automatic, the arrow was more obvious as it was rotating and the participant could identify the direction from the source image. For the bunny, the automatic rotate had 60 models (68%) that were ranked 1 (unusable) and had only 30 models marked unusable (53%) for manual rotate. The surprising thing to note is that 12 models (21%) were ranked 5 (amazing) for manual rotate but for automatic rotate only 3 were ranked as amazing (3%). This leads the conclusion to indicate that for complex models, better results come from a manual rotation mode, giving more control to the participant. For simpler models where participants may rush in, an automatic rotation seems to render better results. For the other models, little difference showed between both automatic and manual rotation modes.



(a) Manual



(b) Automatic

Figure 6.5: Rotate Hits by Rating Sorted by Ascending Complexity

Discussion. We did receive feedback that the time limit on the bunny may have been too little time, which may have skewed the results. However, for stairs, there was an ample amount of time. The complexity of the bunny and the significant amount of time required may also have discouraged beginners from finishing the task. This would be a motivation

or an ease of use problem more than rotation, but also may have skewed results.

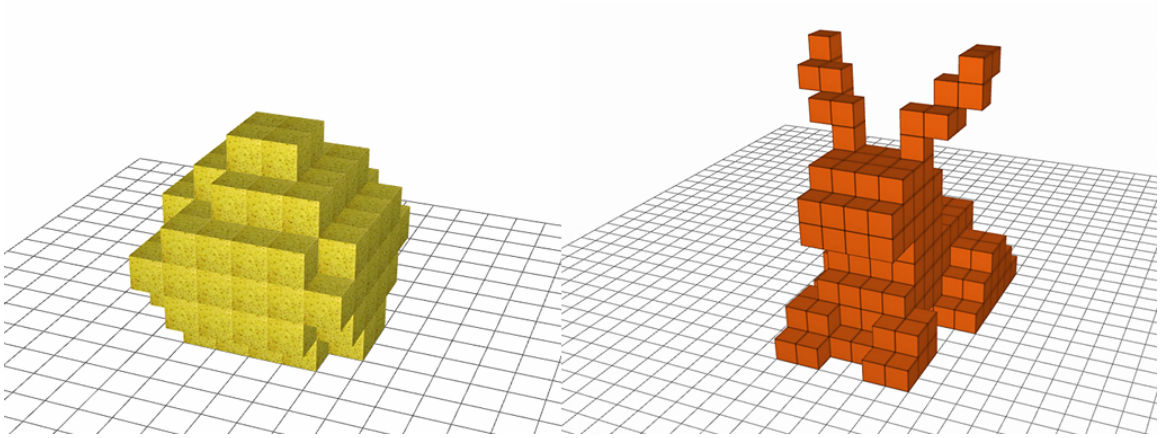


Figure 6.6: The models of complexity ranged from a simple pear to a more complex Standford Bunny.

6.3 Scaling with Complexity

How well crowdsourcing can be applied to a scaling complexity relies heavily on the tool and the aggregation techniques used. This experiment was to determine if our method scales well with complex models. The hypothesis was that it does not scale well and that different aggregate methods can be applied to help mitigate this. The various techniques we used were ranking weights, automatic weights/scrubbing bad data, iterative refinement, and PCA realignment.

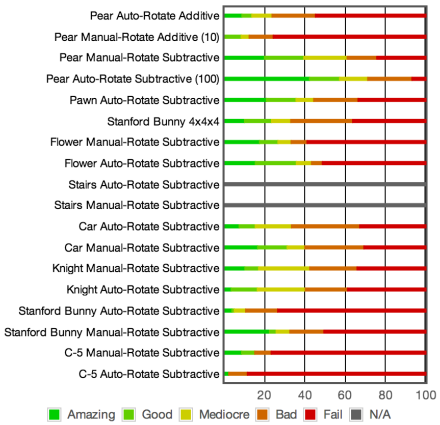


Figure 6.7: Hits by Rating Sorted by Ascending Complexity

Ranking Weights. We looked at various ways of ranking weights, whether it should be based on how like the abstract object it is or whether it is like the picture. For example, when we received some of the data back for the car, we received some great automobiles that didn't really look like source image. It was also important to note that how the question is defined changes the results of our output and the context of the analysis. If it's based on an abstract notion, the aggregate methods would be incredibly

difficult. If we showed a picture of a C-5 and we received a 747, it would change the way the aggregate object would look. There were also several people who thought it was funny to draw random objects as well. For example, a participant provided a very well proportioned chair instead of a flower. It was then decided that the question to ask was how well did the object resemble the item in the image. Based on this, two of us went through and manually ranked the items to further our hypothesis to make sure our method made sense and to test aggregate methods prior to approaching a much larger audience.

Automatic Weights and Data Scrubbing. The next logical step was to develop a machine that could determine if a model was like the aggregate. Initially, we used a simple average technique without weights. This worked well for simple models like the pear and pawn. This didn't work at all for the car, knight, bunny, and other mid to comlex models. The reason behind this is that some models were just terrible and shouldn't be counted in the aggregate. The presupposition must be that the participant tries to sculpt the object. With this in mind, it was clear that if the participant submitted the default preset, then they didn't attempt the experiment and thus provided trash data. It's important to note that we did have validation tools and that some participants would add/remove objects randomly so that they could submit. The validation tools we used were time, action, number of voxels, and data types expected. This did help improve quality of data, however we still were receiving data that was hindering the aggregate. The second machine that was then built was one to determine if the model was a good model. This tool looked at each of the corners to see if all of them still existed, then it determined if a certain percent of the voxels still existed. Finally, a simple algorithm was appended that checked to see if the center of the object was missing the majority of the expected voxels. This was much more accurate to what was expected. The only concern is that of losing valid data. The algorithm typically errors on the side of good data rather than junk, it is likely that one or two valid attempts may be automatically removed.

The automatic weight machine could then be improved with valid data. Here, we developed a process that checked how close a model was like the aggregate, added badness,

inversed to gather the weight, applied the weight back to the aggregate and then restarted the process for a good depth. The depth we used for the Likert was 15. We included the change for each depth in our analysis tools with that of charts so that we could tell how much each model was effecting the outcome. This method greatly improved the quality of the knight, pawn, bunny, and mid to complex pieces.

Iso Weighting Visualization. Another weighing technique we used was once we started utilizing isosurfaces to smooth the model. We wanted to give a more aggregate feel to the model and logically wanted to attach weights to each of the steps in the marching process. To determine whether the value was in or outside of the object during a step, the point in question would utilize the average "rank" and compare it with the neighbor ranks. In doing this, a more refined model is constructed utilizing the ranking data and the way in which isosurfaces operate.

PCA Realignment. As mentioned in other sections, we used PCA to realign models and provide a more accurate aggregate between models. This greatly affected the way that weights are calculated and applied. On each step of isosurface construction (marching), the machine looks at the point inversely aligned to see if the point was in the original shape, then we'd apply the weight, and then we'd average it to give the point the correct value. This effectively gave us a much more accurate aggregate model and a clean way of integrating it to the existing weight and ranking system.

Chapter 7: Perceptual Evaluation

7.1 Experiment

Our final approach to determine whether or not the aggregate is indeed as beautiful according to the general public would require a census or survey of sorts that would pitch the individual contributions with the aggregate models. In order to accomplish this, an experiment would need to be setup that would have a crowd determine whether or not the model closely resembled the source picture.

Method

General Setup. Again, following Gingold et al. (2012) [41] we limited the data entry by size, by time, and chose a simple source, one model at a time. To ensure simplicity, the subject could only view a rotating sculpture and choose between 1 (not at all close), 2, 3 (somewhat close), 4, and 5 (very close). Before the subject could choose an option, they must wait one half a rotation, to ensure that all sides of the model have been observed prior to rating. Once the subject clicks or presses the 1-5 key, the next model in the series appear. Once the last model in the series has been rated, the hit is submitted. Choosing the correct wording for the Likert so that each contribution accurately reflects each subject's decision is important and difficult. We used previous work to determine the proper wording [42].

The series only contains models created in a single set from experiments 1 and 2. This means that automatic rotation subtractive would render x objects and so these would rated together. Additionally, a shuffle algorithm was used to ensure that the models were in random order as to not create a bias for the first or last model in the set. In this series, it is incredibly important to know that the aggregate models were also shuffled into the series. This way, the subject would not know that some were created by people while other models were generated aggregates.

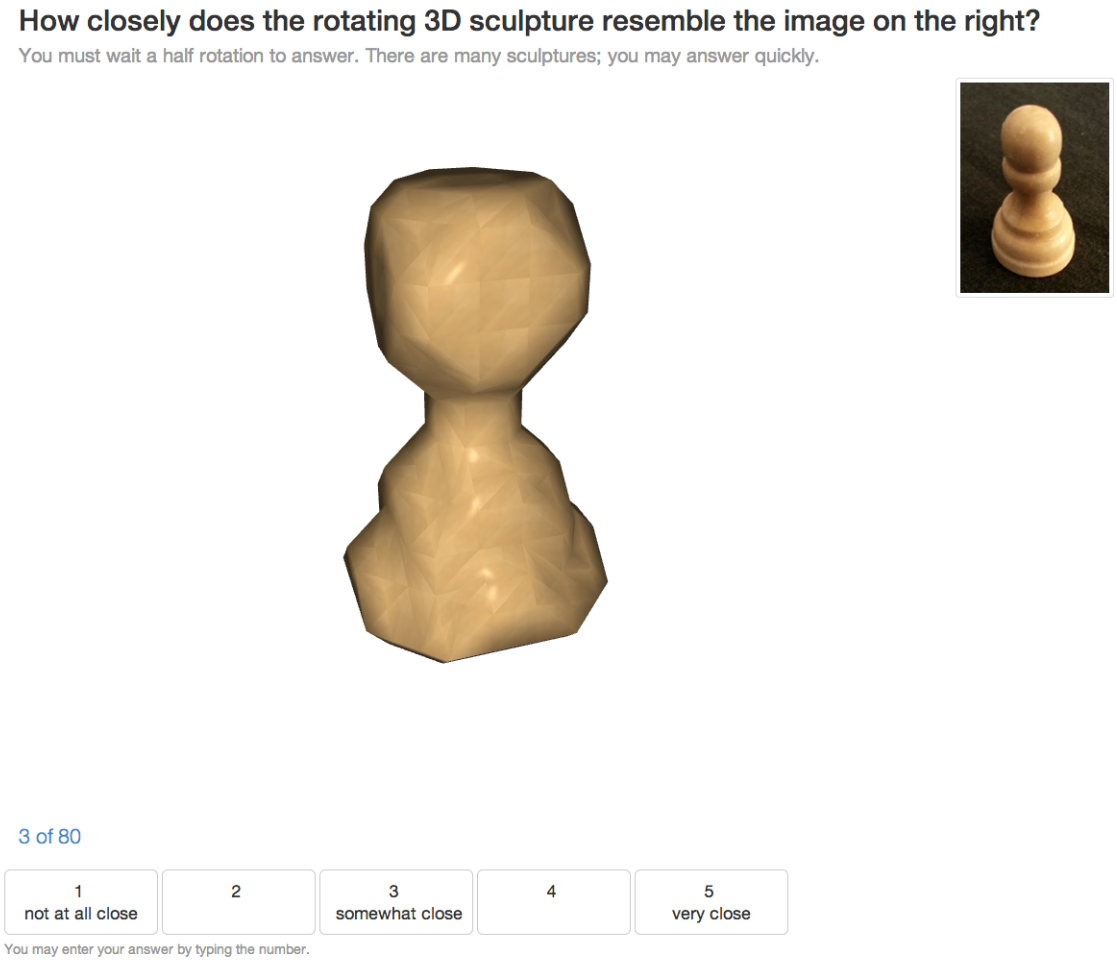


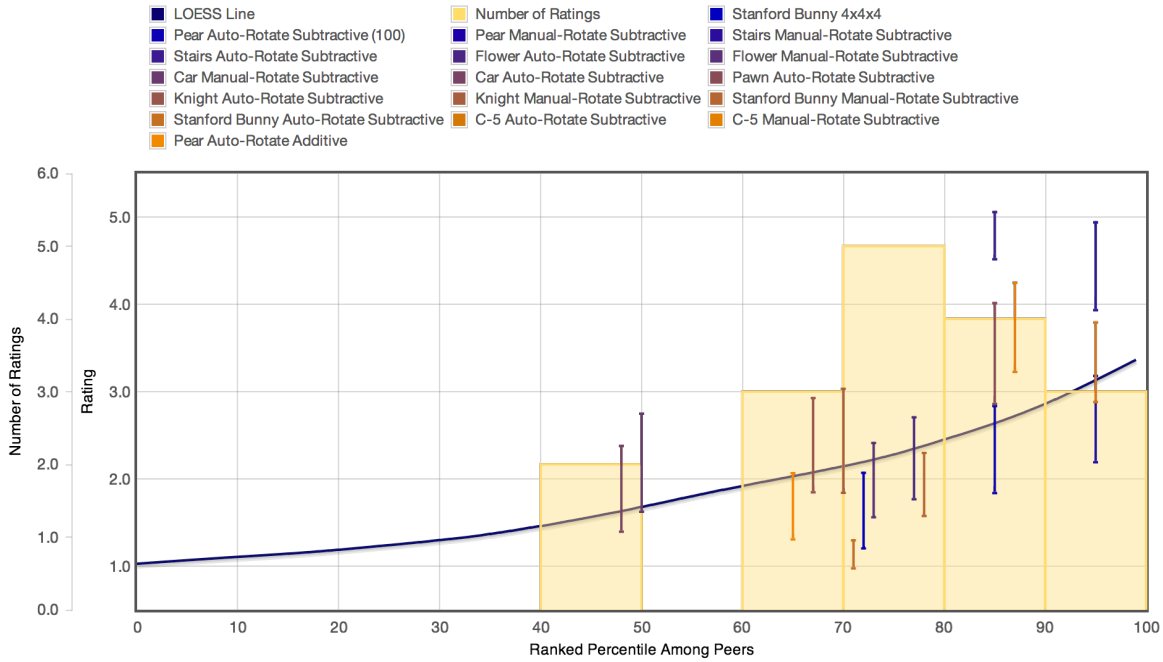
Figure 7.1: Example Likert Survey

Two sets of each series were also provided, one that were Voxels and the other that were isosurfaces (marching cubes). Q1, Q2, and Q3 binary and generated weighted aggregates were added to the Voxel Series (+6). For marching cubes, we added Q1, Q2, Q3 for both binary/weighted and pca/non-pca resulting in an additional 12 variations.

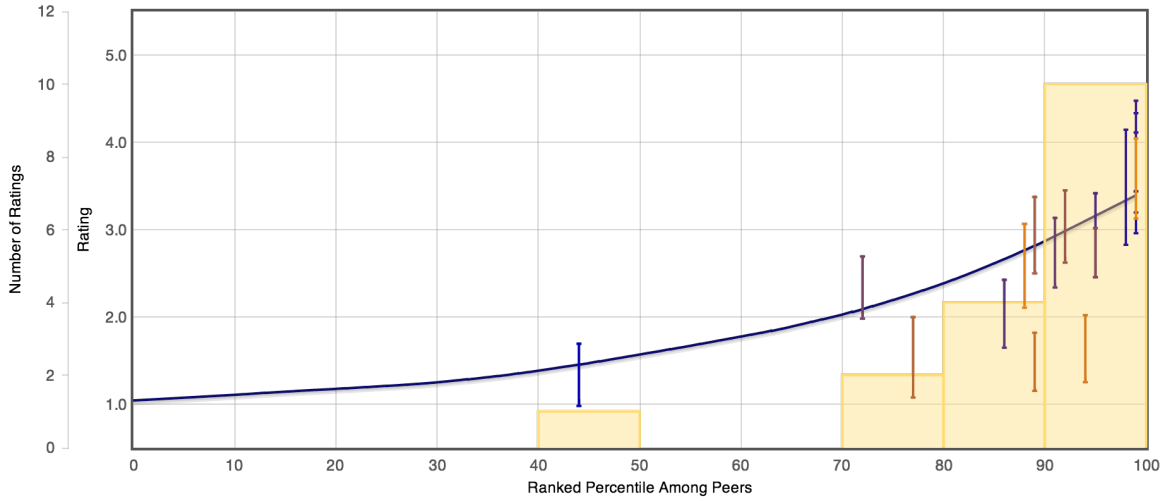
Participants and Procedure. Participants (sources s = 17, assignments a were roughly 70 - 100 depending on the source, participants p = 20 per hit) were asked "How closely does the rotating 3D sculpture resemble the image on the right?" Additional instructions read "You must wait a half rotation to answer. There are many sculptures; you may answer quickly." Below this was the rotating model, to the right is the image and below

this are buttons with numbers 1-5 with the text "You may enter your answer by typing the number.". Each submitted a model using Amazon Mechanical Turk.

Results.



(a) Voxel



(b) Isosurface

Figure 7.2: The results for all experiments run suggests that the Isosurface aggregates, including pca, for the weighted and unweighted were chosen by crowdsourcing to show better resemblance to the source.

To determine success, we utilized various stacked charts and candlestick diagrams using R and JavaScript tools. We take the top consensus model for each to determine the percentile and compare to the rest of the contributions. For the voxel Likert survey results, we see that 17.6 percent are in the 90-100 percentile, 23.5 percent are in the 80-90 percentile, 23.5 percent are in the 70-80 percentile, 23.5 percent are in the 60-70 percentile, and 11.8 percent are in the 40-50 percentile. For the isosurface Likert survey results, we see that 58.8 percent are in the 90-100 percentile, 23.5 percent are in the 80-90 percentile, 11.8 are in the 70-80 percentile, and 5.9 percent are in the 40-50 percentile. It is important to note that the hit in the lowest percentile was part of a very low resolution model of the Stanford Bunny, which had only 4x4x4 possible blocks. The low resolution Stanford Bunny was deliberate to determine how to establish iterative refinement and a poor aggregate was expected.

7.2 Complexity

We also grouped the experiments into separate categories: no complexity, low complexity, mid complexity, and high complexity. High complexity pertains to a higher resolution model requiring more advanced modifications or actions to the default voxels in order to produce a sculpture resembling the image. Mid complexity required less actions but was less symmetric than the Low complexity models. Low complexity models were radially symmetric on one or two axes and reflective symmetry was observable on all 3 axes. No complexity involved 2 experiments where the subject was given exact blueprints on how to build the model. Because exact instructions were given, there should have been no work in the figuring out of how translate the source to sculpture.

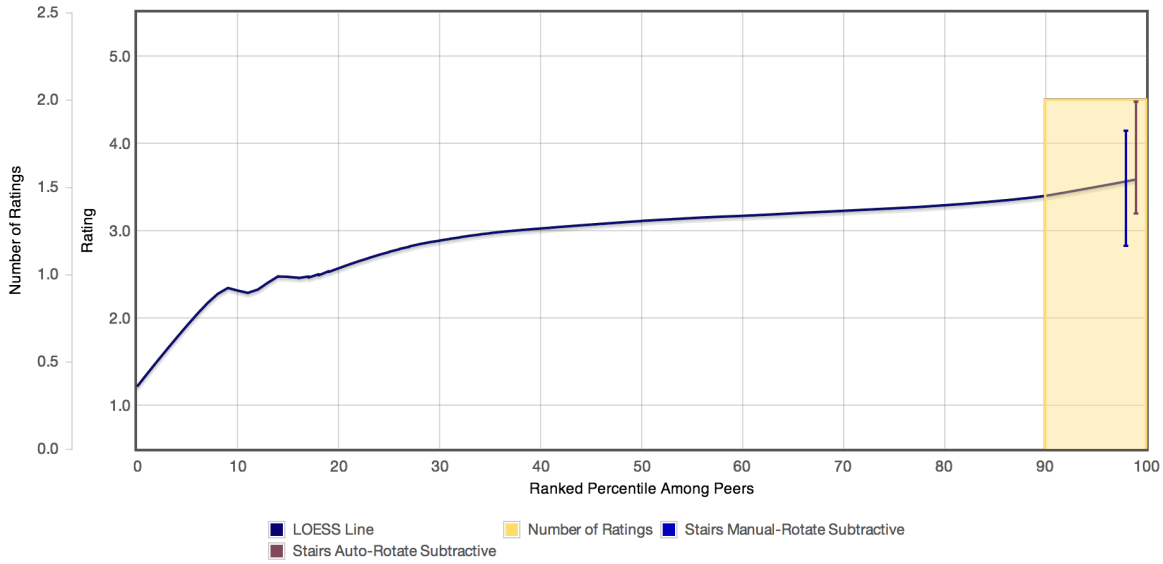


Figure 7.3: No Complexity

No Complexity Likert results included "Stairs Manual Rotate Subtractive" and "Stairs Automatic Rotate Subtractive". For the manual rotation, the top aggregate model was in the 98th percentile for the isosurface and the 95th percentile for the voxel aggregate model. The best manually rotated voxel model was the "Q3 Weighted" model and the best isosurface was the "Q1 Weighted PCA" one ahead of the "Q3 Weighted PCA" model. The best automatic rotated voxel model was the "Q2 Weighted" model at the 81 percentile while the isosurface best was "Q2 Weighted PCA" at the 100 percentile. For no complexity models, the aggregates were without a doubt among the top representatives according to the crowdsource audience.

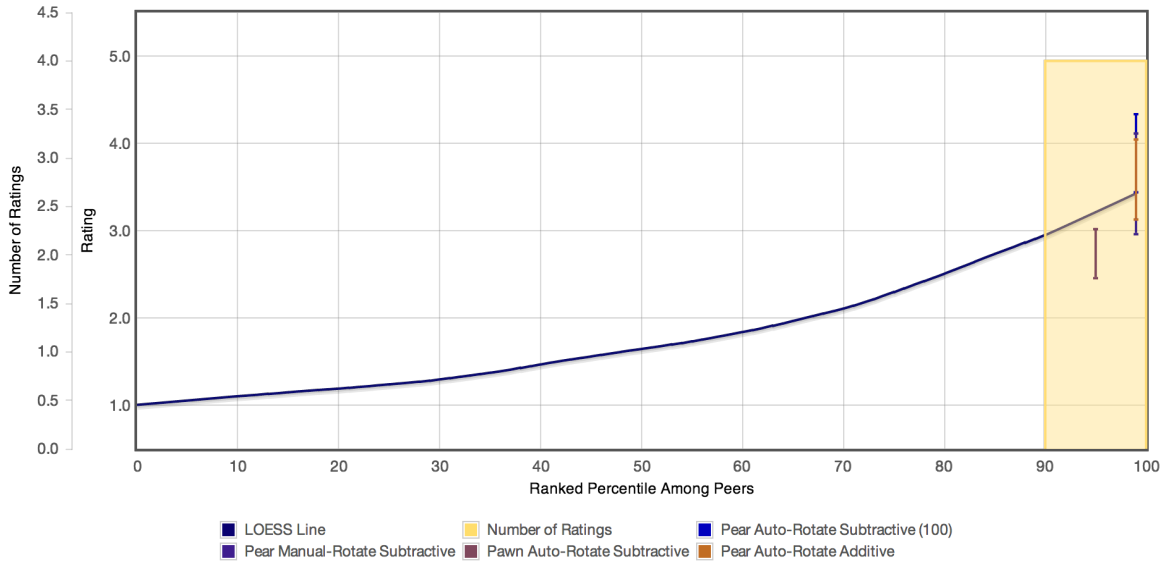


Figure 7.4: Low Complexity

Low Complexity Likert results included "Pear Manual Rotate Subtractive", "Pear Automatic Rotate Additive", "Pear Automatic Rotate Subtractive" and "Pawn Auto-Rotate Subtractive". All results were the 99 percentile except the "Pawn Auto-Rotate Subtractive" which was the 95 percentile. This means that all mid complexity experiments resulted in consensus models that were in the 90-100 percentile.

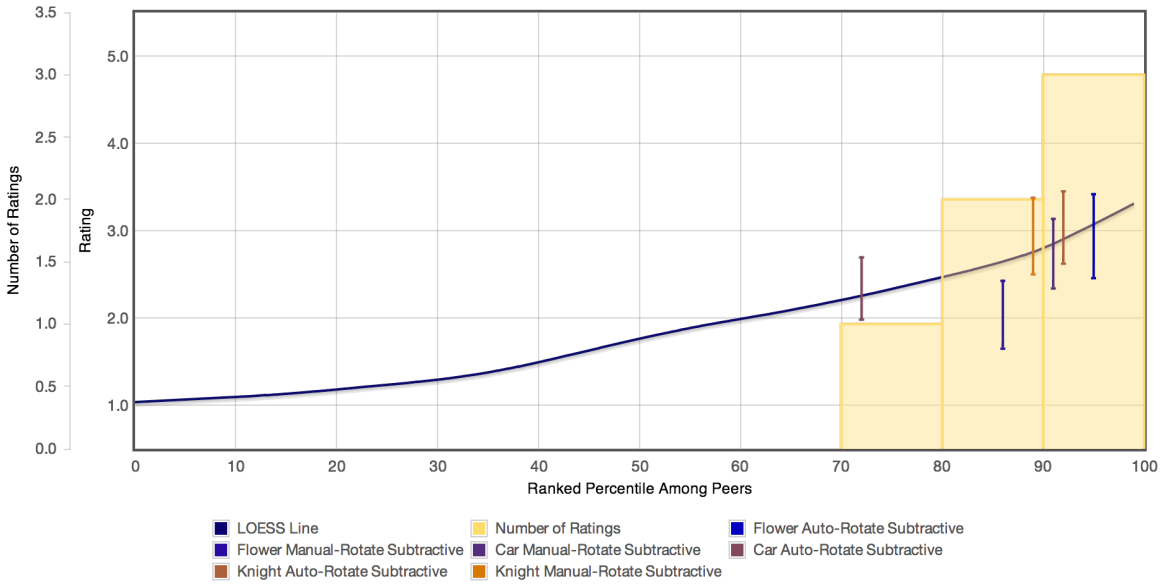


Figure 7.5: Mid Complexity

Mid Complexity Likert results included "Flower Manual Rotate Subtractive", "Flower Auto-Rotate Subtractive", "Car Manual Rotate Subtractive", "Car Auto-Rotate Subtractive", "Knight Manual Rotate Subtractive", and "Knight Auto-Rotate Subtractive". The "Car Auto-Rotate Subtractive" was the lowest at the 72 percentile while the "Flower Auto-Rotate Subtractive" was the highest. The wheels in the Car models tended to not show up appropriately in the aggregates as roughly half of the good models from subjects placed the wheels under the car while others placed them on the outside of the car. This is where future work may find a better solution to this, some of which we've initiated. The knight models both seem to be quite good and did exceptionally well according to the Likert.

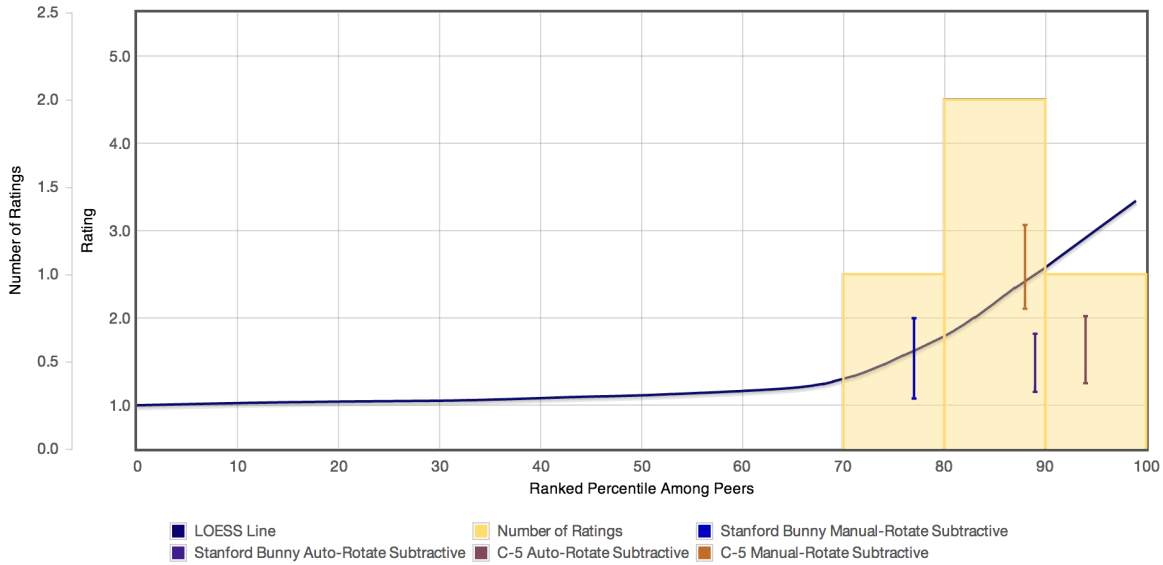
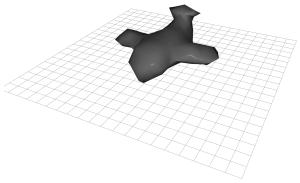
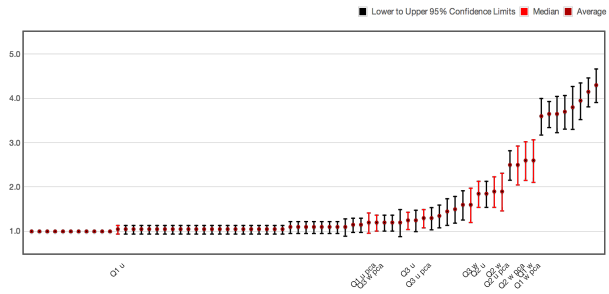


Figure 7.6: High Complexity

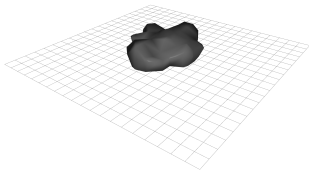
High Complexity Likert results included "Stanford Bunny Manual Rotate Subtractive", "Stanford Bunny Auto-Rotate Subtractive", "C-5 Manual Rotate Subtractive", and "C-5 Auto-Rotate Subtractive". According to Figure 7.6, the results show that the experiments did not produce consensus models that were necessarily chosen as the best representative model for the set. It's hard to say that the aggregate methods were good or not, however, since the quality of the data returned was mostly unusable at best. I also find the chart to be misleading since many of the aggregates were simply better than the individual contributions which were terrible attempts at sculpting the source image.



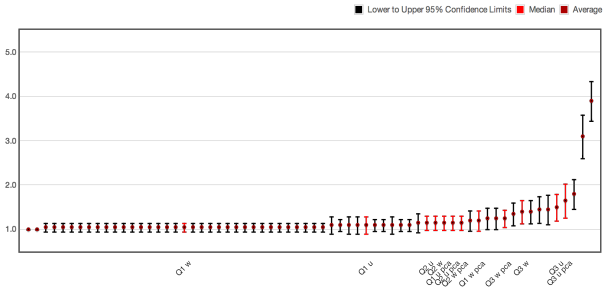
(a) Iso Q1 Weighted PCA



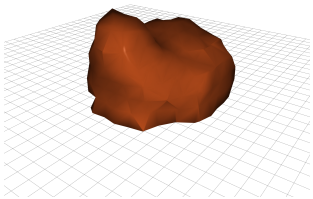
(b) C-5 Manual Rotate Subtractive Isosurface



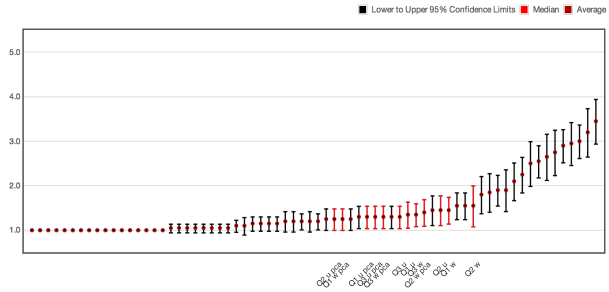
(c) Iso Q3 Unweighted PCA



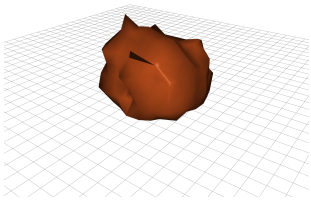
(d) C-5 Auto-Rotate Subtractive Isosurface



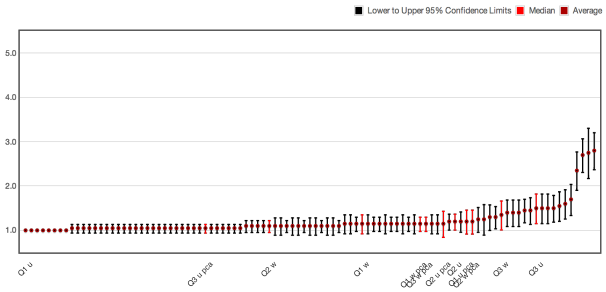
(e) Iso Q2 Weighted



(f) Stanford Bunny Manual Rotate Subtractive Isosurface



(g) Iso Q3 Unweighted



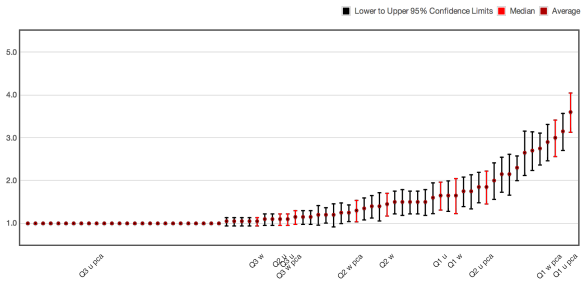
(h) Stanford Bunny Auto-Rotate Subtractive Isosurface

Figure 7.7: High Complexity Best Aggregate Isosurface

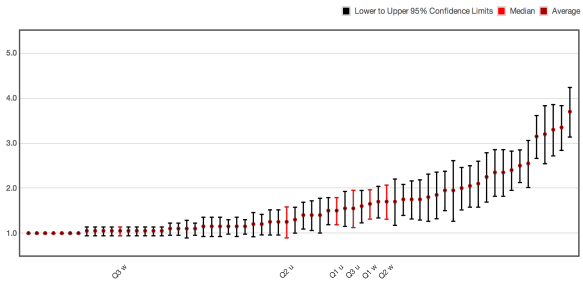
7.3 Additive vs Subtractive

As suggested in the Chapter 6, additive had some pretty obvious issues dealing with alignment and size differences. We applied various techniques to address this with excellent results, specifically regarding the principal component analysis (PCA) algorithm. We found great improvements utilizing this model but in the end, the subtractive method still had better success in our preliminary analysis and according to the Likert survey results.

When we ran the Likert experiment, PCA was applied to the isosurface models introduced a much improved result. Initially the voxel experiment that had been run as seen in Figure 7.8b resulted in showing that the aggregates were ranked around the 50th percentile. This would mean that the consensus model was no better than the other provided models in the experiment. Once the isosurface with PCA Likert experiment was conducted, the results in Figure 7.8a indicated that the PCA was accepted at a much greater rate with a large difference for the 100th percentile model (Q1 w pca), greatly reducing the margin of error.



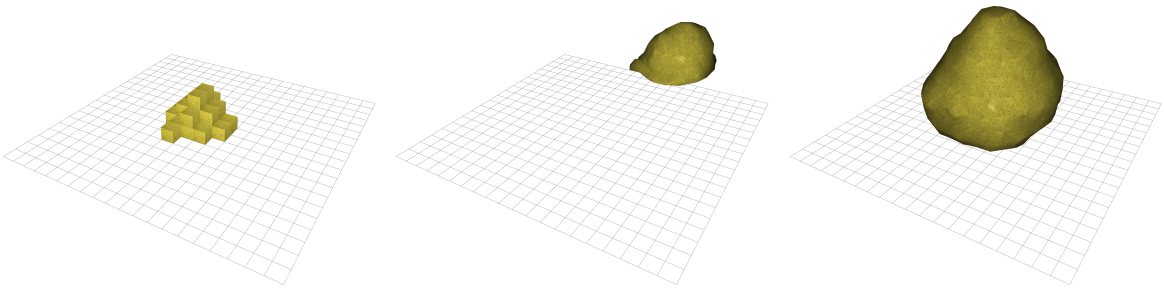
(a) Pear Auto-Rotate Additive Isosurface



(b) Pear Auto-Rotate Additive Voxel

Figure 7.8: The results for the Pear auto additive suggests that the Isosurface aggregates, including PCA, for the weighted and unweighted were chosen by crowdsourcing to have better resemblance to the pear.

Aligning the results using helped a bit with the additive vs subtractive. What PCA accomplished is it translated the shapes so that they were centered, it rotated the shapes so that they were aligned in the same direction, and it scaled the shapes so that small or large shapes would be compared at a constant size. Though we had hoped for better results with PCA introduced, we were still quite surprised with the results.

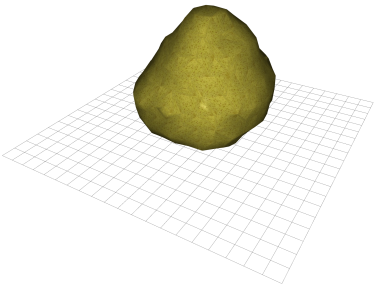


(a) Voxel Q2 Weighted (b) Iso Q1 Unweighted (c) Iso Q1 Unweighted PCA

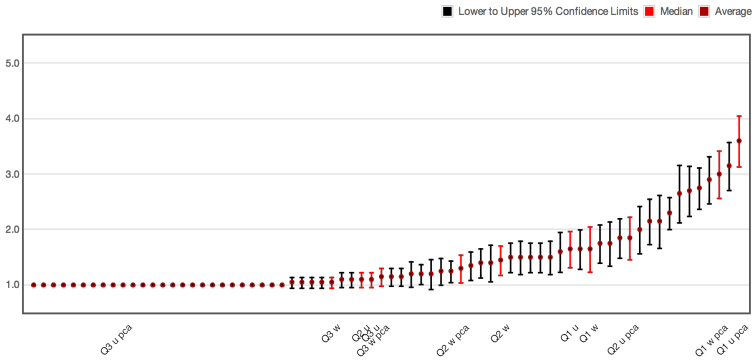
Figure 7.9: Auto-Rotate Additive Voxel vs Isosurface vs Isosurface PCA

To further show the results, consider Figure 7.9 to understand visually what the voxel and PCA models looked like. Figure 7.9a was the best aggregate model from the Pear Auto-Rotate Additive Voxel Likert results. Whereas Figure 7.9c is the best model from the Pear Auto-Rotate Additive Isosurface Likert results. It is clear to us and the crowdsourcing results that the PCA did an excellent job with generating a consensus model through the the alignment technique. Additionally, the Isosurface Q1 Unweighted has been added for reference to see what the aggregate model looked like prior to applying PCA. This model also is closer to the 50th percentile, further illustrating the usefulness of PCA in the Additive approach.

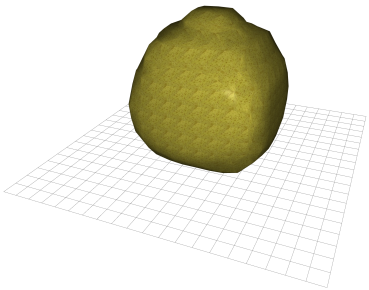
In spite of the great advanced provided for by PCA, the subtractive still had a better set of results than the additive results. PCA provided even better results for subtractive. All aggregates for the Auto Rotate Subtractive were in the top quarter percentiles where there were more aggregates closer to the 50th percentile for the auto rotate additive. Furthermore there were a significant amount of additive models that were ranked 1 by 20 people. Given the odds that 26 models were ranked a 1 suggests that subjects had a difficult time using the tool to produce quality models give an additive approach. Whatever the reasons may be, the subtractive approach yielded better results.



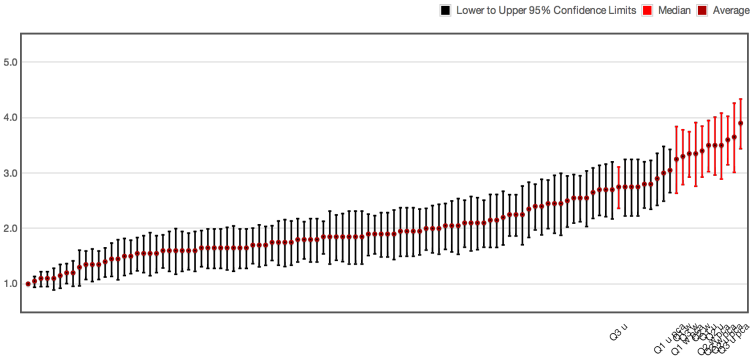
(a) Iso Q1 Unweighted PCA



(b) Pear Auto-Rotate Additive Isosurface



(c) Iso Q3 Unweighted PCA



(d) Pear Auto-Rotate Subtractive Isosurface

Figure 7.10: Pear Aggregate Marching Cubes

7.4 Automatic vs Manual Rotation

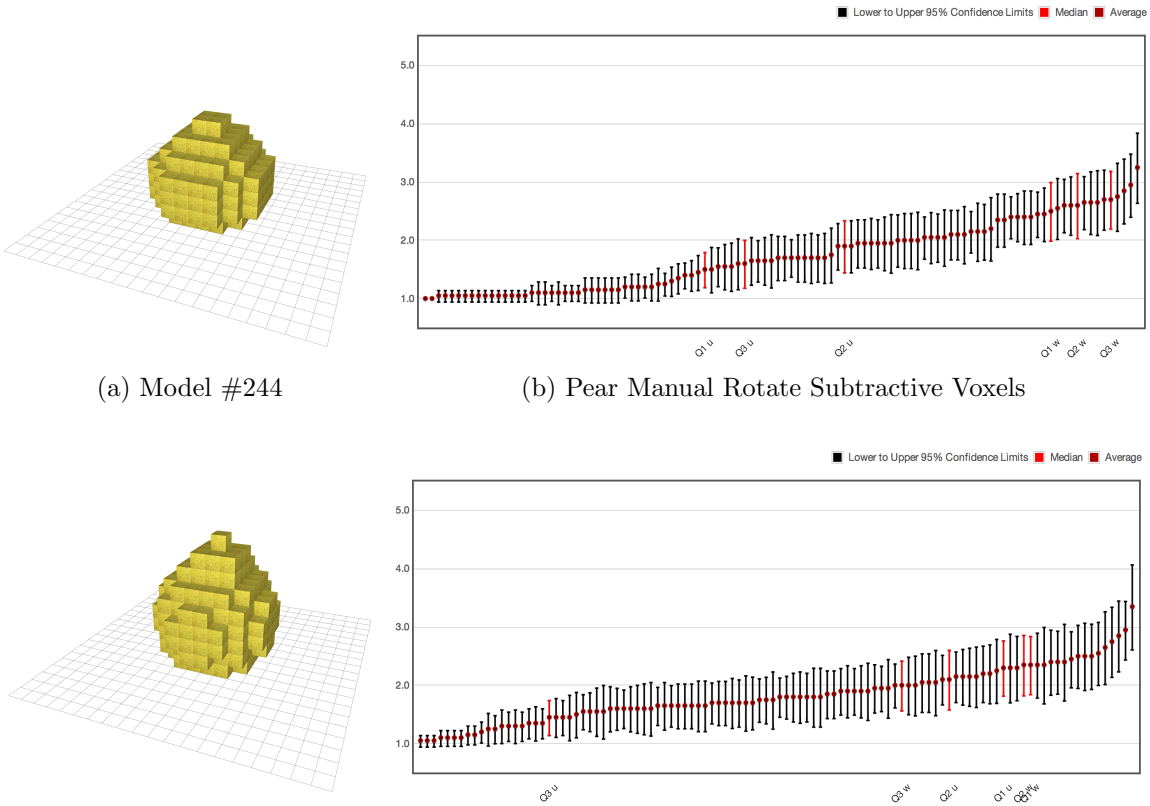
Looking at the perceptual evaluation results, one of the questions to answer was whether or not rotation occurring automatically or letting the participant control the rotation of the model had better results. We look to the Likert results to look for trends to see if the consensus models were rated higher than the individual contributions and if certain aggregate methods were favored. Since two perceptual visualizations were generated, voxel and isosurface, we will look at them separately as different methods were applied.

It is incredibly important to note that since the subjects in the Likert study rated a random set of rotation and additive/subtractive for each visualization separately. This means that the ratings are likely to not translate well from automatic to rotation. Model A may have an average rating of 3.4 because all of its peers in the same set are terrible and have an average rating of 1. Model B may be a very good model but rated at 2.4 because its peers are rated at an average of 3.8. It wouldn't be a stretch to say that Model A and Model B may be nearly identical but given the peers, the Likert survey would favor one over the other. This is why comparing across is not as simple as comparing numbers but rather trends and extreme differences, such as 20 of the models having been rated an average of 1.

One important trend that needs to be addressed is how well a consensus model does among its peers. This is to say that it was chosen to be among the best representative of the stimulus image by being rated in a Likert survey. We can compare this without issue because it is in the same set as the other contributions. If the same algorithm is used on another set with success, we then know that the algorithm produces good results. If the same algorithm is used on many other sets and rated above the median, then we can say with confidence that the algorithm is a successful method for aggregating models.

For the voxel perceptual study, we look at a the ratings from crowdsource of the individual contributed models included with the consensus models. In Figure 7.11 the models on the left refer to the furthest right black box plot. This indicates that these are the best rated individual contributions. Because of the error margin, it is difficult to say that automatic rotation is better than manual rotation for this model. Again, we can't compare

the ratings directly as mentioned previously, but we can identify that the similar trends occur not showing much difference.



(a) Model #244

(b) Pear Manual Rotate Subtractive Voxels

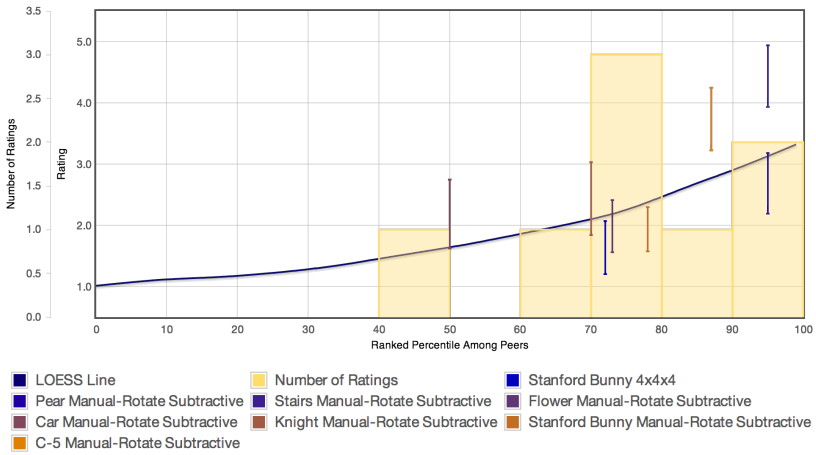
(c) Model #73

(d) Pear Auto-Rotate Subtractive Voxels

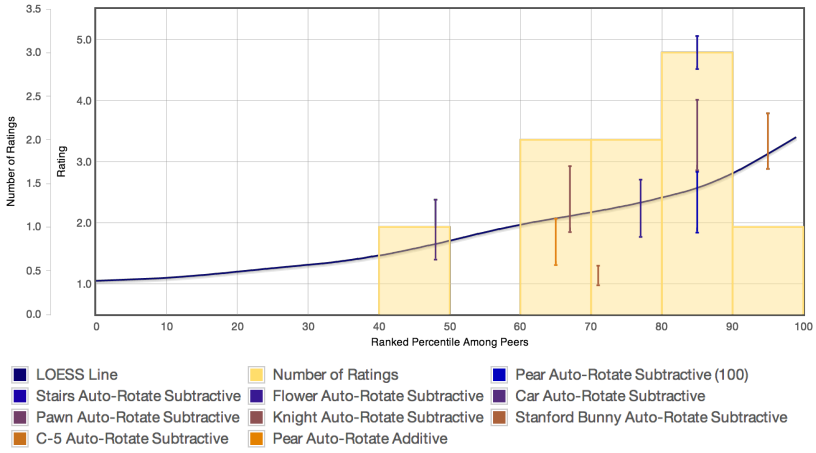
Figure 7.11: Pear Individual Contribution Voxels

We then look at the overall difference with automatic and manual rotation to the best consensus models to see if perhaps the aggregate produces an overall better model for one mode of rotation vs automatic. In the Figure 7.11, it is important to note that the furthest red to the right belongs to the manual rotate, where the average set of red box plots are higher in the automatic rotation results. When we compare all models with the best aggregate weight, we see that for some models the manual rotation is clearly better, for some the automatic rotation is clearly better, and for some there is little change (7.12). We see that the pear is rated at the 95% for manual rotation while automatic rotation is

at the 85%. However the C-5 is rated at the 87% for manual while the automatic is at 95%. Overall this is not a significant change. Many of the other models had less than a 5% change, which does not give any confidence of any reasonable difference when looking at all models using a voxel visualization.



(a) Manual Rotation

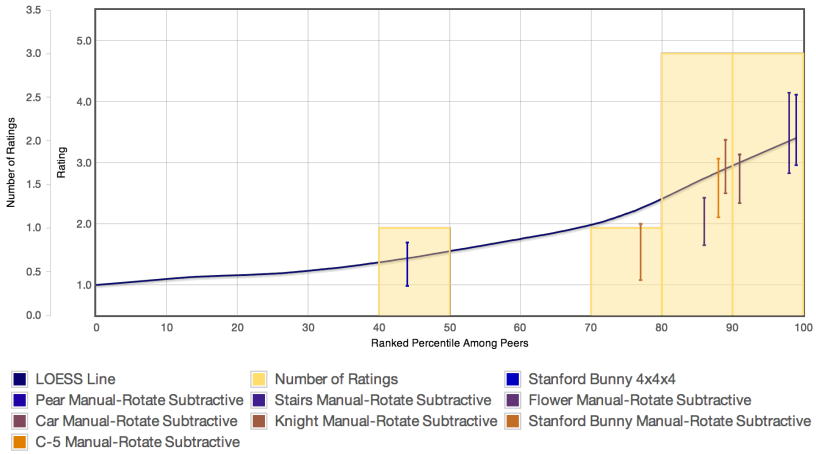


(b) Auto Rotation

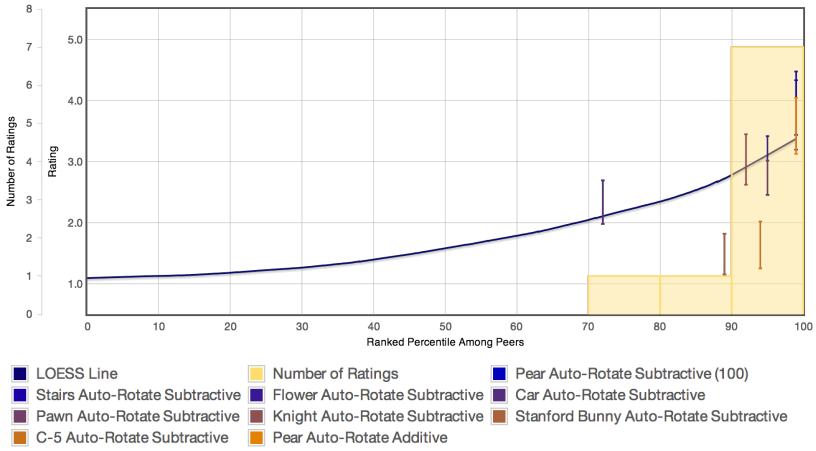
Figure 7.12: Voxel Highest Rated Consensus Models

For the isosurface perceptual study, we also look at a the ratings from crowdsource of the individual contributed models included with the consensus models. Here we see that it appears that a variety of models did much better with automatic rotation vs manual rotation. In Figure 7.13 we see more aggregates in the 90 percentile block in the automatic

than in the manual rotation. However, the 4x4x4 Stanford Bunny was a test to see how users would handle a low res version of the harder bunny in hopes to follow up with iterative refinement (See Chapter 8). Additionally, the pawn was never tested in a manual rotation environment. For this reason, we look at Tukey's Honestly Significantly Different Statistics (HSD) to see if rotation mode actually provides meaningful differences.



(a) Isosurface with Manual Rotation



(b) Isosurface with Auto Rotation

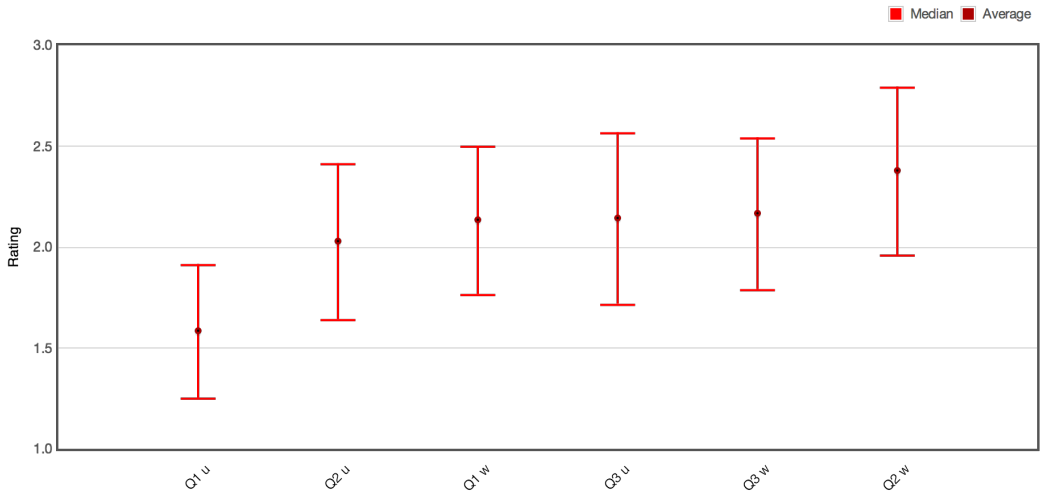
Figure 7.13: Isosurface Highest Rated Consensus Models

When we look at HSD to analyze the Likert results on response to the difference of rotation method used in the editor tool, results appear to favor some models over others. For high complexity models, there a significant difference favoring manual rotation over

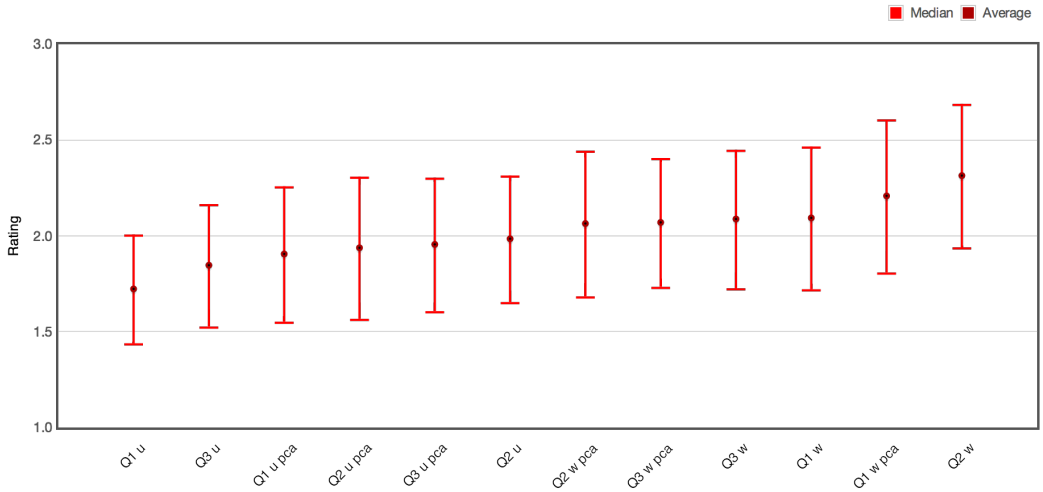
automatic rotation. For mid complexity models, there was no significant difference. For low complexity stimulus images, it seems that there was no significance difference for voxels but for isosurface the favored method was manual rotation. For no complexity, auto rotation was favored.

7.5 A Final Look at the Aggregate Models

When looking at 7.14 we compare the different consensus models and see various compelling findings. The higher rated results are weighted, Q2 weighted (50 percentile - the median consensus) seems to be the favorite. For Isosurface results, Q1 weighted with PCA is second. To corroborate we utilized Tukey's HSD and found that weighted is significantly rated higher in the Likert results than unweighted. We find this to be true with both voxel and isosurface experiments.



(a) Voxel Consensus Models Boxplot



(b) Isosurface Consensus Models Boxplot

Figure 7.14: Consensus Models Boxplots

Chapter 8: Future Work

Iterative Refinement. A method in which a participant or a group could rerun the same experiments with greater granularity had been attempted and postponed to focus more on weighting techniques and realignment. Since this has been tried and seemed quite successful, it made sense to discuss as a viable future work. This would effectively add a great level of detail and provide much higher quality detail to the generally sculpted objects in previous experiments. There are several ways to approach this that were discussed.

The first was to simply increase the resolution on the voxel models and run it through Amazon Turk. Ideally, if this process works, it would eventually create a very high quality model. The problems with this is that the web browser can only support a max number of voxels prior to crashing. We specifically ran into this after going to a second depth of iteration. The next methods were designed to avoid this. Additionally, the amount of time to improve resolution over an entire object would mean that a participant would need to

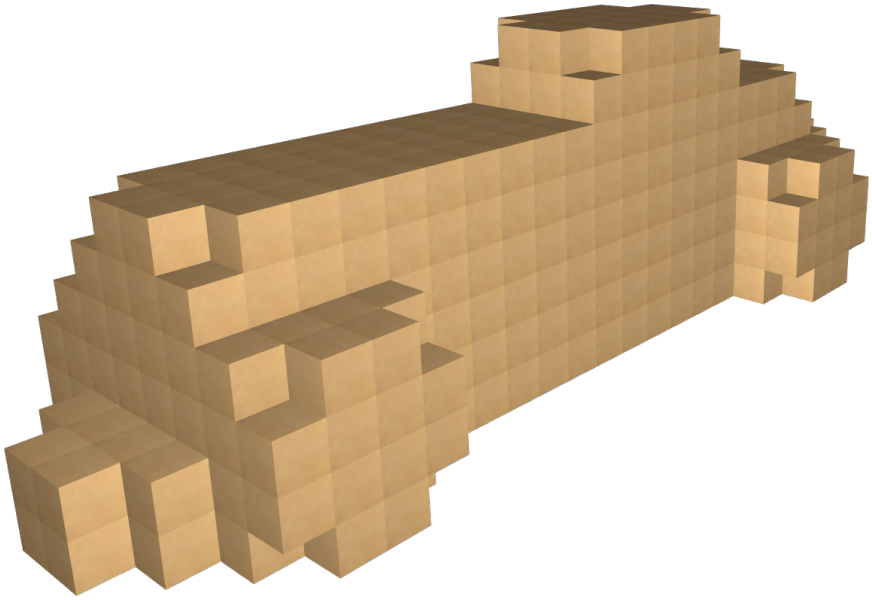


Figure 8.1: Iterative Refinement applied to a car

spend a great deal of time to create this quality. This also seemed unreasonable considering crowdsourcing works well with small observable tasks.

The second method was to allow participants to edit quadrants (or some subset) of the original model, providing higher resolution. This would also reduce the amount of time required by a participant to finish the task, thus likely resulting in better work. This also would allow the software to be constructed such that the rest of the object would be merged, thus completely one large uneditable object where the quadrant would contain voxels.

The third method was to make all non visible voxels a solid merged object. This would mean that participants couldn't effect the structure of the existing object but could only add/remove a shallow set of voxels on the model. This could become a time issue as mentioned in th first method, but would unlikely be as much time as the participant would be restricted to a depth of voxels.

Utilizing a different tool for refinement. We looked specifically at SculptGL since it was easy to provide data. We were able to export our files from the analysis tool, we fixed the stitching prior, and then SculptGL was able to modify the object very well. The only problem with SculptGL is that it requires a good understanding of modeling. What we proposed was that we'd remove most actions in the UI and only give very few UI panels to simplify for the crowdsourcing audience. SculptGL is open source and would be a prime candidate. The aggregate methods would require to use PCA with isosurface as a binary aggregate would not be feasible without some set of directions/arrows which would complicate the participant experience

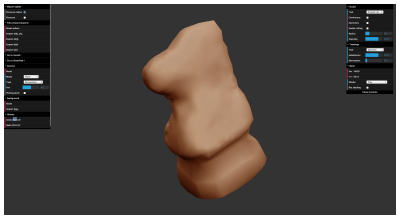


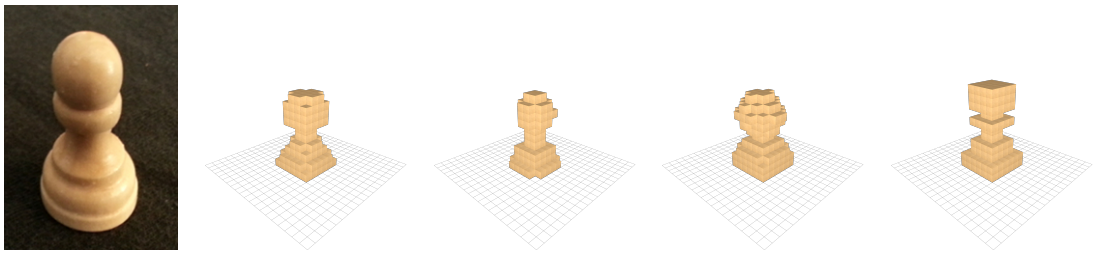
Figure 8.2: An aggregate knight in SculptGL [4]

Utilizing a different tool for crowdsourcing. We wanted to use some kind of 2D to 3D tool to see if this would somehow be easier to the audience than voxel editing. If given more time, this was something we had planed to build out, to see if the results would be better with different modeling behavior. The aggregate methods again would most certainly reuse

the PCA and isosurface techniques, but the result could be quite different if not better.

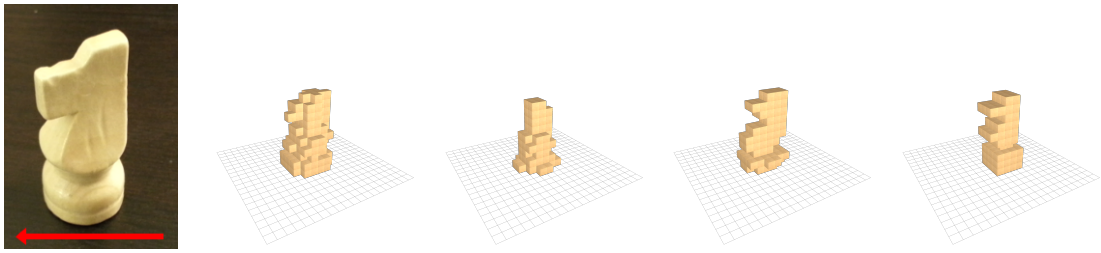
Appendix A: An Appendix

This is a list of the best consensus models for several of the stimulus images as well as some random user entries.



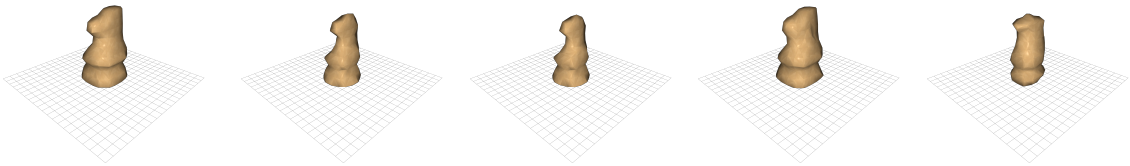
(a) Stimulus (b) Q2 W (c) Q3 W (d) Participant (e) Participant

Figure A.1: Pawn Auto Rotate Voxels



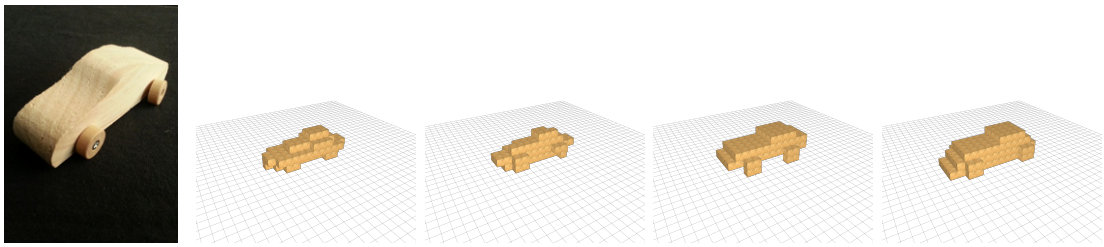
(a) Stimulus (b) Q2 W (c) Q3 U (d) Participant (e) Participant

Figure A.2: Knight Auto Rotate Voxels



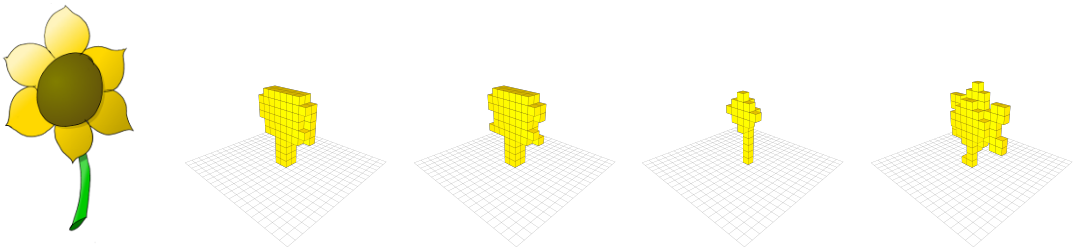
(a) Q2 W (b) Q3 W (c) Q3 U (d) Q2 U (e) Q3 W PCA

Figure A.3: Knight Manual Consensus Iso



(a) Stimulus (b) Q3 W (c) Q3 U (d) Participant (e) Participant

Figure A.4: Car Auto Rotate Voxels



(a) Flower (b) Q2 U (c) Q2 W (d) Participant (e) Participant

Figure A.5: Flower Manual Rotate Voxels

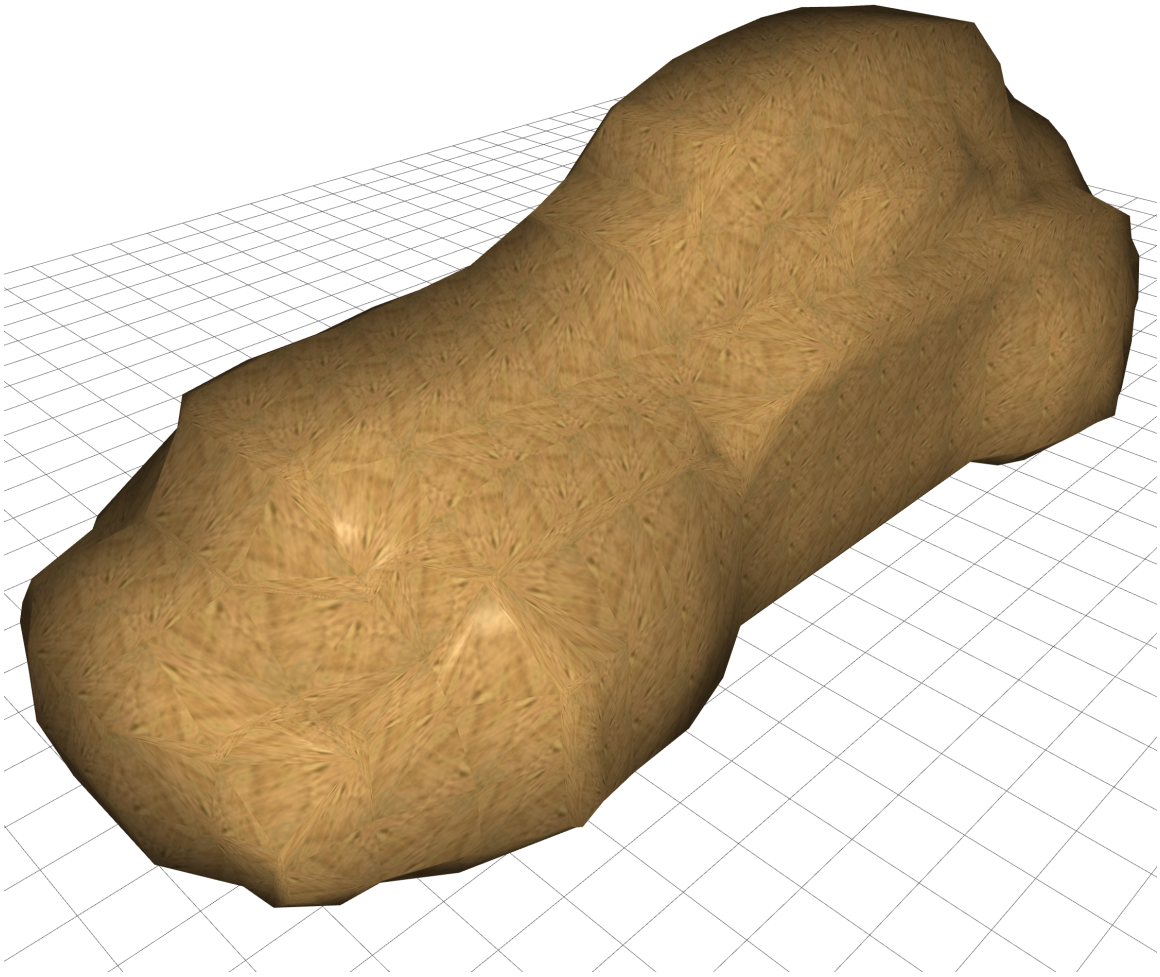


Figure A.6: Aggregate Car with PCA Iso with rating of 3 or higher and weight better than 1

Bibliography

Bibliography

[1] R. Cabello, "Voxels," November 2013. [Online]. Available: http://www.mrdoob.com/projects/voxels/

[2] Y. Gingold, T. Igarashi, and D. Zorin, "Structured annotations for 2D-to-3D modeling," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, p. 148, 2009.

[3] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of educational psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[4] S. Ginier, "Sculptgl," 2013. [Online]. Available: http://stephaneginier.com/

[5] J. Surowiecki, *The wisdom of crowds.* New York, NY: Random House, 2004.

[6] D. J. Cohen and S. Bennett, "Why can't most people draw what they see?" *Journal of Experimental Psychology: Human Perception and Performance*, vol. 23, no. 3, p. 609, 1997.

[7] T.-S. Hwang and D. G. Ullman, "The design capture system: Capturing back-of-the-envelope sketches," *Journal of Engineering Design*, vol. 1, no. 4, pp. 339–353, 1990. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/09544829008901662

[8] P. Rodgers, G. Green, and A. McGown, "Using concept sketches to track design progress," *Design Studies*, vol. 21, no. 5, pp. 451–464, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142694X00000181

[9] Y. I. Gingold, P. L. Davidson, J. Y. Han, and D. Zorin, "A direct texture placement and editing interface," in *Proceedings of ACM UIST*, 2006, pp. 23–32. [Online]. Available: http://doi.acm.org/10.1145/1166253.1166259

[10] S.-H. Bae, R. Balakrishnan, and K. Singh, "ILoveSketch: As-natural-as-possible sketching system for creating 3D curve models," in *Proceedings of ACM UIST*, 2008, pp. 151–160.

[11] R. Schmidt, A. Khan, G. Kurtenbach, and K. Singh, "On expert performance in 3D curve-drawing tasks," in *Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBIM)*, 2009, pp. 133–140. [Online]. Available: http://doi.acm.org/10.1145/1572741.1572765

[12] C. Grimm, "Results of an observational study on sketching," Washington University in St. Louis, Tech. Rep. 2011-57, Jun 2011, presented as a poster at Sketch-based interfaces and modeling, 2011. Full images and videos here.

[13] T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: a sketching interface for 3d freeform design," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 409–416. [Online]. Available: http://dx.doi.org/10.1145/311535.311602

[14] A. Shtof, A. Agathos, Y. Gingold, A. Shamir, and D. Cohen-Or, "Geosemantic snapping for sketch-based modeling," *Computer Graphics Forum*, vol. 32, no. 2, pp. 245–253, 2013, proceedings of Eurographics 2013. [Online]. Available: http://dx.doi.org/10.1111/cgf.12044

[15] L. Stanculescu, R. Chaine, and M.-P. Cani, "Freestyle: Sculpting meshes with self-adaptive topology," *Comput. Graph.-UK*, vol. 35, no. 3, pp. 614–622, June 2011, special Issue: Shape Modeling International (SMI) Conference 2011.

[16] A. Inc., "Mudbox," 2013. [Online]. Available: http://www.autodesk.com/products/mudbox/overview

[17] B. Foundation, "Blender," 2013. [Online]. Available: http://www.blender.org

[18] A. Inc., "Maya," 2013. [Online]. Available: http://www.autodesk.com/products/autodesk-maya/overview

[19] E. A. Inc., "Spore," 2009. [Online]. Available: http://www.spore.com/what/scc

[20] E. Weiszfeld, "Sur le point pour lequel la somme des distances de n points donnés est minimum," *Tohoku Mathematics Journal*, vol. 43, pp. 355–386, 1937.

[21] E. Weiszfeld and F. Plastria, "On the point for which the sum of the distances to n given points is minimum," *Annals of Operations Research*, vol. 167, no. 1, pp. 7–41, 2009. [Online]. Available: http://dx.doi.org/10.1007/s10479-008-0352-z

[22] D. Prelec, "A bayesian truth serum for subjective data," *Science*, vol. 306, no. 5695, pp. 462–466, 2004. [Online]. Available: http://www.sciencemag.org/content/306/5695/462.abstract

[23] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, Aug. 1987. [Online]. Available: http://doi.acm.org/10.1145/37402.37422

[24] H. Muller and M. Wehle, "Visualization of implicit surfaces using adaptive tetrahedrizations," *Scientific Visualization Conference*, vol. 0, p. 243, 1997.

[25] S. F. F. Gibson, "Constrained elastic surface nets: Generating smooth surfaces from binary segmented data," in *Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, ser. MICCAI '98. London, UK, UK: Springer-Verlag, 1998, pp. 888–898. [Online]. Available: http://dl.acm.org/citation.cfm?id=646921.709482

[26] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, ser. SGP '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 61–70. [Online]. Available: http://dl.acm.org/citation.cfm?id=1281957.1281965

[27] J. Kopf and D. Lischinski, "Depixelizing pixel art," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)*, vol. 30, no. 4, pp. 99:1 – 99:8, 2011.

[28] I. L. Dryden and K. V. Mardia, *Statistical shape analysis.* John Wiley & Sons, Chichester, 1998.

[29] K. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[30] V. G. Kim, Y. Lipman, and T. Funkhouser, "Blended intrinsic maps," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 79:1–79:12, Jul. 2011. [Online]. Available: http://doi.acm.org/10.1145/2010324.1964974

[31] V. G. Kim, W. Li, N. J. Mitra, S. DiVerdi, and T. Funkhouser, "Exploring collections of 3d models using fuzzy correspondences," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 54:1–54:11, Jul. 2012. [Online]. Available: http://doi.acm.org/10.1145/2185520.2185550

[32] B. Wirth, "Variational methods in shape space," Dissertation, University Bonn, 2009. [Online]. Available: http://hss.ulb.uni-bonn.de/2010/2195/2195.htm

[33] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel, "Coupled quasi-harmonic bases," *Computer Graphics Forum*, vol. 32, no. 2pt4, pp. 439–448, 2013. [Online]. Available: http://dx.doi.org/10.1111/cgf.12064

[34] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 175:1–175:10, Dec. 2009. [Online]. Available: http://doi.acm.org/10.1145/1618452.1618521

[35] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev, "3d self-portraits," *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2013)*, vol. 32, no. 6, November 2013.

[36] A. W. Fitzgibbon, "Robust registration of 2d and 3d point sets," *Image and Vision Computing*, vol. 21, no. 13, pp. 1145–1153, 2003.

[37] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2–3, pp. 114 – 141, 2003, ¡ce:title¿Nonrigid Image Registration¡/ce:title¿. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314203000092

[38] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2010.46

[39] M. J. C. Crump, J. V. McDonnell, and T. M. Gureckis, "Evaluating amazon's mechanical turk as a tool for experimental behavioral research." *PLoS One*, vol. 8, no. 3, p. e57410, 2013. [Online]. Available: http://www.biomedsearch.com/nih/Evaluating-Amazons-Mechanical-Turk-as/23516406.html

[40] J. C. Eyal Peer, Gabriele Paolacci and P. Mueller.

[41] Y. Gingold, E. Vouga, E. Grinspun, and H. Hirsh, "Diamonds from the rough: Improving drawing, painting, and singing via crowdsourcing," in *Proceedings of the AAAI Workshop on Human Computation (HCOMP)*, 2012.

[42] B. Pelham and H. Blanton, *Conducting Research in Psychology: Measuring the Weight of Smoke*. Cengage Learning, 2012. [Online]. Available: http://books.google.com/books?id=bbIJAAAAQBAJ

Curriculum Vitae

Matthew Sabol is currently a technical lead in the Strategic Innovation Group (SIG) at Booz Allen Hamilton supporting the Digital Ecology team. He leads various Research and Development investments for the firm among several developers across multiple teams. His focus is typically mobile solution, user interface/experience, computer graphics, visualizations and big data analysis.