Optimizing Design of Physical Objects for Fabrication

Ph.D. Dissertation Prospectus

Tomer Weiss
Department of Computer Science
University of California, Los Angeles
tweiss@cs.ucla.edu

April 24, 2016

Contents

1 Introduction

3D printers have become prevalent. They have been used to fabricate various objects, ranging from quick prototypes of commercial products [20], to customized sensors [7], to consumer products [5], to props used in the entertainment industry [12], to machine parts [4]. 3D printers have also become invaluable to the medical industry, such as for creating substitute bone parts [10], dental implants [2], medical models [6], artificial skin [18], and even tissue [15]. The possibilities of 3D printing are far-reaching.

A 3D printing application that has not previously been investigated is the fabrication of objects that float in a predetermined way. The possibilities opened up by fabricating floating objects range from specialized toys, decorative items for pools and aquariums, floating sensors as in [11], and novelty objects that can float in counter intuitive ways.

In making objects that float in fluid, we must deal with the buoyancy force, which is best summarized by Archimedes' principle: *Any object, wholly or partially immersed in a fluid, is buoyed up by a force equal to the weight of the fluid displaced by the object*.

The focus of this thesis proposal is a novel method that, given an input 3D geometric model of an object, produces an output object with the same exterior shape that floats on a prescribed *fluid plane*. To that end, my optimization-based method manipulates the object's material distribution by carving away material. By adjusting the liquid density parameters, my method is not limited to water, but works with any fluid. We also discuss alternative formulations and limitations of 3D printing.

This prospectus is organized as follows: In chapter 2 I review work in modeling objects for fabrication in 3D printers and discuss the latest results. In chapter 3 and 4 I present my optimization approach, starting from definitions, method and summary of current results. We discuss the complexities of 3D printing and future research directions in 5. Chapter 6 presents my conclusions.

2 Related work

There is a series of published works on the topic of manufacturing physical objects from virtual objects using 3D printers. This transformation is challenging because of the numerous physical constraints a given target object has to satisfy. In recent years, several papers have appeared on how to model and manipulate objects for 3D printer fabrication, where the methods for manipulating these objects vary in approach and final goals, yet focus on adjusting the shape of objects under physical and mechanical objectives.

"Make it Stand" [19], an interactive optimizer that produces a balanced standing model by carving and deforming an existing model, is closely related to our work. The input to the optimizer is an unbalanced, asymmetric model that is unstable on a surface. If interior carving does not suffice to achieve stability, the optimizer attempts to deform the model in areas initially designated by the user. The deformation is restricted to affine transformations and the optimizer numerically minimizes a deformation energy to find the best deformation that minimally modifies the exterior while making the object as stable as possible. The optimizer alternates between inner carving and shape deformation until a final stable object is achieved. While the final fabricated model fulfills the objectives, there are some limitations; e.g., a poor choice of handles by the user will result in an object that is visibly different from the original. Also, depending on the shape of the object, some deformations can result in collisions between different parts of the object, resulting in an unprintable outcome.

"Spin-it" [3] proposed a technique to generate spinnable toys from arbitrary 3D objects. The user provides a solid 3D model and a desired axis of rotation, and then the algorithm modifies the internal mass distribution by internal carving. If the internal carving is not sufficient, the method deforms the object and adds other types of material as necessary in order to optimize for rotational stability.

The work in [16] builds upon [3] and [19] and shows how to use manifold harmonics for the shape optimization of objects. The input is the object's surface and a desired global goal. The output is two surfaces—external and internal—determining the shell of the object. The global goal can range from static stability, such as standing, to non-static goals like rotational stability. The goal can also be adapted to adding floating capability to the object, but the work does not go as far as controlling the angle and position of a floating object, or in optimizing for flotation stability.

Previous papers in shape optimization use complex techniques for determining which parts of the object to deform; e.g., energy-based optimization [19], laplacian-based surface editing [3, 21], stress map optimization with stochastic sampling [13], and so on. We propose to use a more direct approach in solving the floating problem — we manipulate and transform voxels in a discrete approximation of the continuous object.

3 Optimization Framework

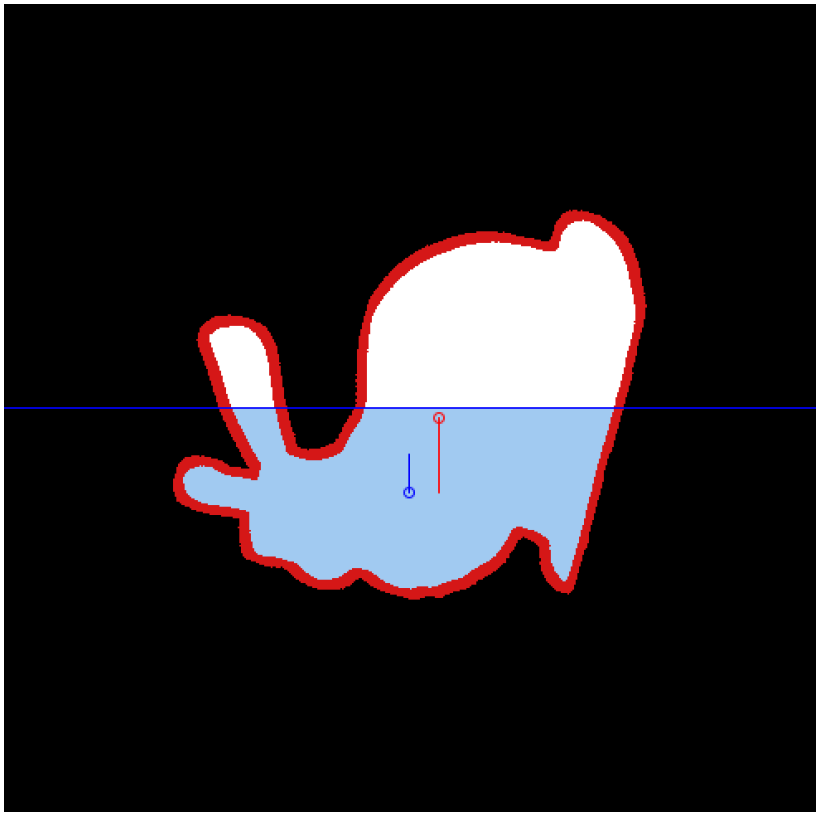

Figure 1: 2D view of the Stanford bunny. The blue line denotes the fluid plane, where the light blue area indicates the submerged part. The red curve denotes the outer shell of the bunny. The blue circle and line refers to the buoyant force f_b and the red circle and line refers to gravitational force f_g. The length of these lines is proportional to the magnitude of the forces.

3.1 Definitions

Our aim is to fabricate a floating object given an input solid 3D object model and a desired fluid plane, which is a plane that divides the submerged part of the object from the part above the fluid, as shown in Fig. 1.

3.2 Object Properties

We define the set of points inside the object as Ω and the set of points of the object that are submerged as Ω_w. Similarly, we define the density function for the material of the object as $\rho(\mathbf{x})$. We assume that the fluid has constant density ρ_w. Finally, we define the density of the material used for 3D printing as ρ_m. We can then compute the object's mass as

$$m = \int_{\Omega} \rho(\mathbf{x}) dV, \tag{1}$$

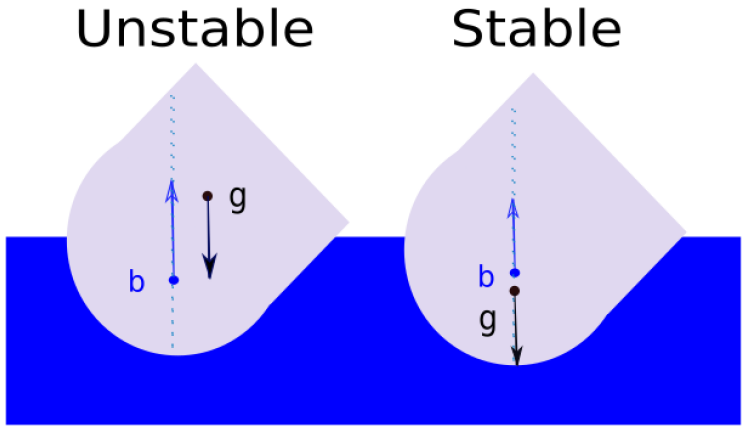

Figure 2: Gray objects in blue fluid: $\mathbf{b}$ and $\mathbf{g}$ depict the forces of buoyancy and gravity, and their point of origin, which is $\mathbf{c}_m$ and $\mathbf{c}_b$. The object on the left is unstable because these forces not aligned along the line of action of the force of buoyancy. The object on the right is stable, since $\mathbf{c}_m$ is lower then $\mathbf{c}_b$, and the forces are aligned and equal.

it's center of mass as

$$\mathbf{c}_m = \frac{1}{m} \int_{\Omega} \mathbf{x} \rho(\mathbf{x}) dV, \tag{2}$$

the gravitational force, which originates at the center of mass, as

$$\mathbf{f}_g = m\mathbf{g}, \tag{3}$$

where $\mathbf{g}$ is the acceleration due to gravity, the mass of displaced fluid as

$$m_w = \rho_w \int_{\Omega_w} dV, \tag{4}$$

the center of buoyancy as

$$\mathbf{c}_b = \frac{\rho_w}{m_w} \int_{\Omega_w} \mathbf{x} dV, \tag{5}$$

and the buoyant force, which originates from the center of buoyancy [23] and is equal to the weight of the fluid displaced by the object, as

$$\mathbf{f}_b = -m_w \mathbf{g}. \tag{6}$$

Both $\mathbf{f}_g$ and $\mathbf{f}_b$ act along $\mathbf{g}$, which is normal to the fluid plane, but in opposite directions.

3.3 Equilibrium

In order for the object to be stable and float in fluid (Fig. 2), we want equilibrium between $\mathbf{f}_g$ and $\mathbf{f}_b$:

$$\mathbf{f}_g + \mathbf{f}_b = \mathbf{0}. \tag{7}$$

The lines of action of these two forces must coincide in order to balance the object. Since the forces act along the same vector, but in opposite directions, we need only make sure that the origins align:

$$\mathbf{c}_m = \mathbf{c}_b + \lambda \mathbf{g}, \quad \lambda \geq 0. \tag{8}$$

For a stable equilibrium, λ must be non-negative. To make the object more stable in the fluid, we want the center of mass to be as low as possible, that is maximizing λ. We will explain how we achieve this numerically in chapter 4.

4 Method

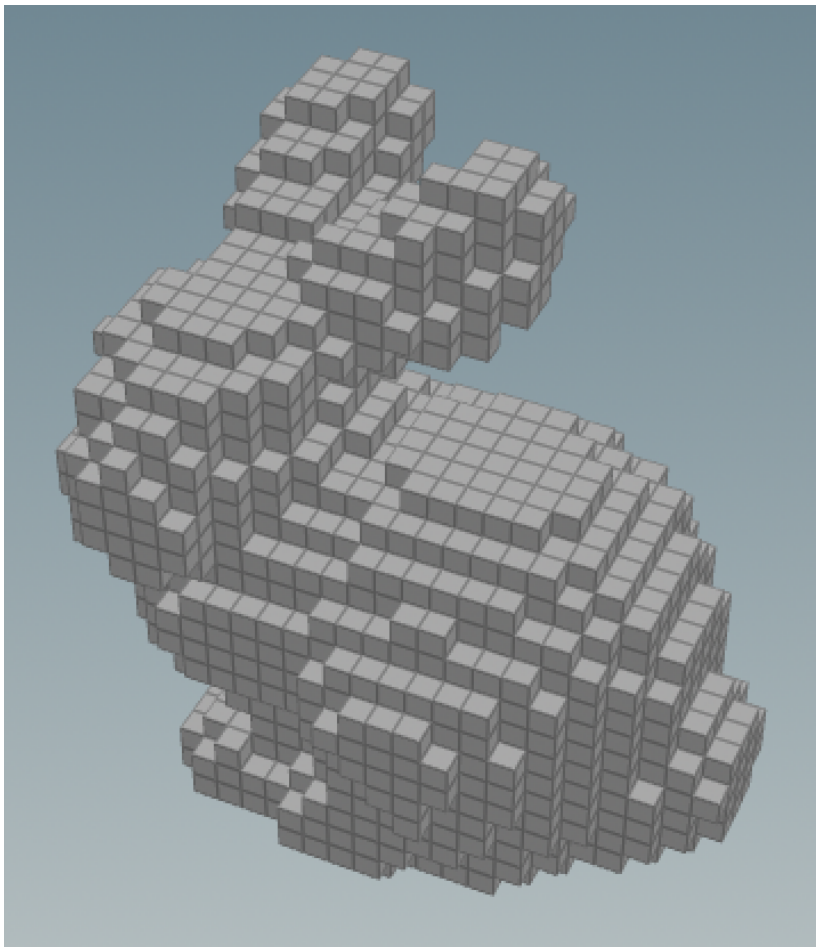

Figure 3: A discrete (voxelized) Stanford Bunny.

Given an input representing a discretized solid object and a fluid plane, our goal is to produce a physical instantiation of that object that floats on that fluid plane. Our method has two carving phases: Phase 1 sets up the mass of the object and physical forces, whereas Phase 2 aligns the center of mass and center of buoyancy, and positions the center of mass as low as possible within the body of the object. The following subsections describe this approach in detail.

4.1 Discretization

Our method works with discrete versions of the continuous equations described in chapter 3.2. We discretize the object by converting its volume to voxels (Fig. 3). Without the loss of generality, we align the voxels with the fluid surface. We denote the set of voxels of the object as Γ, and the set of voxels that are inside the object and submerged as Γ_w. The size of a voxel is h. Note that $\rho(\mathbf{x}_v) = \rho_m$ for a filled voxel and $\rho(\mathbf{x}_v) = 0$ if the voxel is empty. The discrete equations corresponding to (1)–(6) are

$$m = \sum_{v \in \Gamma} \rho(\mathbf{x}_v) h^3, \tag{9}$$

$$\mathbf{c}_m = \frac{1}{m} \sum_{v \in \Gamma} h^3 \mathbf{x}_v \rho(\mathbf{x}_v), \tag{10}$$

$$\mathbf{f}_g = m\mathbf{g}, \tag{11}$$

$$m_w = \rho_w \sum_{v \in \Gamma_w} h^3, \tag{12}$$

$$\mathbf{c}_b = \frac{\rho_w}{m_w} \sum_{v \in \Gamma_w} \mathbf{x}_v, \tag{13}$$

$$\mathbf{f}_b = -\, m_w \mathbf{g}. \tag{14}$$

4.2 Moving Material

Let S be the set of voxels of an object that are filled with material; i.e., $\forall v \in S : \rho(\mathbf{x}_v) = \rho_m$. Consider the scenario of carving out a voxel in location $\mathbf{x}_c$ (previously $c \in S$) and filling with material a voxel that was previously empty in location $\mathbf{x}_a$. If $\rho(\mathbf{x}_v)$ is constant, the original position of the center of mass is given by

$$\begin{aligned} \mathbf{c}_m &= \frac{\sum_{v \in S} h^3 \mathbf{x}_v \rho(\mathbf{x}_v)}{\sum_{v \in S} h^3 \rho(\mathbf{x}_v)} = \frac{\sum_{v \in S} \mathbf{x}_v}{||S||} \\ &= \frac{\sum_{v \in S \setminus x_c} \mathbf{x}_v}{||S||} + \frac{\mathbf{x}_c}{||S||}. \end{aligned}$$

After moving material from $\mathbf{x}_c$ to $\mathbf{x}_a$, the center of mass is

$$\begin{aligned} \mathbf{c}'_m &= \frac{\sum_{v \in S \setminus v_c} \mathbf{x}_v}{||S||} + \frac{\mathbf{x}_a}{||S||} \\ &= \frac{\sum_{v \in S \setminus v_c} \mathbf{x}_v}{||S||} + \frac{\mathbf{x}_c + \Delta}{||S||}, \end{aligned}$$

where $\Delta = \mathbf{x}_a - \mathbf{x}_c$. Hence, the relationship between the original center of mass and the modified center of mass is given by

$$\mathbf{c}'_m = \mathbf{c}_m + \frac{\Delta}{||S||}. \tag{15}$$

The y component of Δ gives the amount by which we raise or lower the voxel. If $\Delta_y > 0$, we are moving the $\mathbf{c}_m$ upwards. It is important to note that in Phase 2, when we are selecting $\mathbf{x}_a$ and $\mathbf{x}_c$, we can be sure that our algorithm will only move $\mathbf{c}_m$ upwards, which means that we cannot improve $\mathbf{c}_m$; i.e., position $\mathbf{c}_m$ lower than it is after Phase 1.

4.3 Phase 1: Balance Gravity and Buoyancy

The goal of Phase 1 of our method is to balance the forces of gravity and buoyancy, disregarding any moments. To that end, we want to satisfy (7), which simplifies to

$$m = m_w. \tag{16}$$

Consequently, we carve the inside of the object until its mass is equal to m_w, the mass of displaced fluid. The carving is performed such that $\mathbf{c}_m$ will be as low as possible, as that will increase the stability of the floating object. The carving is preformed top-down, beginning from the voxel farthest away from the fluid surface, until the above condition is satisfied.

We assume the object has a shell with a predefined thickness, as illustrated in Fig. 4, where the thick red boundary of the object represents its shell. This shell remains fixed throughout the carving process, because we want to preserve the object's exterior surface. The shell contributes to the mass and center of mass of the object, and we take that into account in our calculations by defining the object's density as a function of the coordinates of each voxel.

4.4 Phase 2: Align $\mathbf{c}_m$ and $\mathbf{c}_b$

The goal of Phase 2 of our method is to align $\mathbf{c}_m$ and $\mathbf{c}_b$ on the same line of action, as well as to position $\mathbf{c}_m$ as low as possible per (8). Since we want to keep the balance between $\mathbf{f}_b$ and $\mathbf{f}_g$, we achieve this goal by redistributing the remaining internal material of the object, moving material from one voxel to another, thereby preserving the mass of the object from the end of Phase 1. We divide the object's voxels into two groups, hollow voxels (the voxels that were hollowed during the Phase 1), and the remaining non-hollow voxels.

We search for two candidate voxels, one on each side of the buoyancy line of action. We carve out one of these candidate voxels and add its material to the other voxel. Let $\mathbf{x}_c$ be the location of the (non-hollow) voxel to be carved and $\mathbf{x}_a$ location of the (hollow) voxel where we add material. To evaluate which voxels are the best candidates for carving or adding material, we define a fitness function $\Phi(\mathbf{x})$, where $\mathbf{x}$ is a voxel location.

Let $\mathbf{c}_m$ denote the center of mass of the object before modifying the material at that voxel and $\mathbf{c}'_m$ be its center of mass after the material modification. We define

$$\mathbf{d}(\mathbf{x}) = \mathbf{c}_b - \mathbf{c}'_m(\mathbf{x}), \tag{17}$$

where

$$\mathbf{c}'_m(\mathbf{x}_c) = \frac{\mathbf{c}_m m - \mathbf{x}_c \rho_m h^3}{m - \rho_m h^3} \tag{18}$$

if we are carving material from the voxel and

$$\mathbf{c}'_m(\mathbf{x}_a) = \frac{\mathbf{c}_m m + \mathbf{x}_a \rho_m h^3}{m + \rho_m h^3} \tag{19}$$

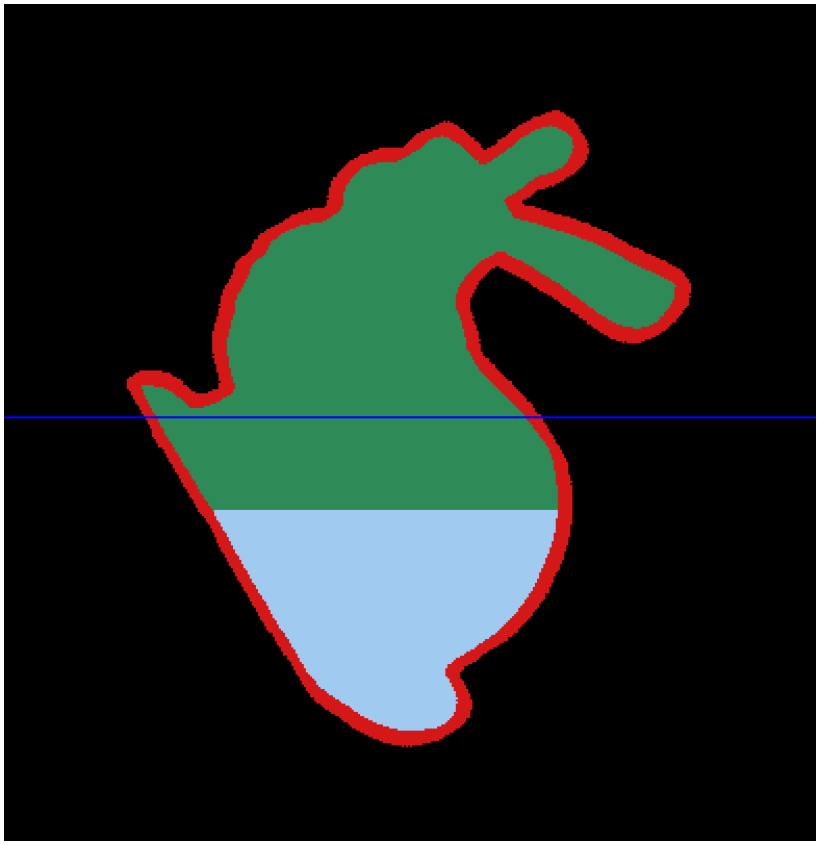

Figure 4: Outcome of Phase 1. The red boundary denotes the shell of the object. The green area denotes the carved material. The other colored areas have the same meaning as in Fig. 1

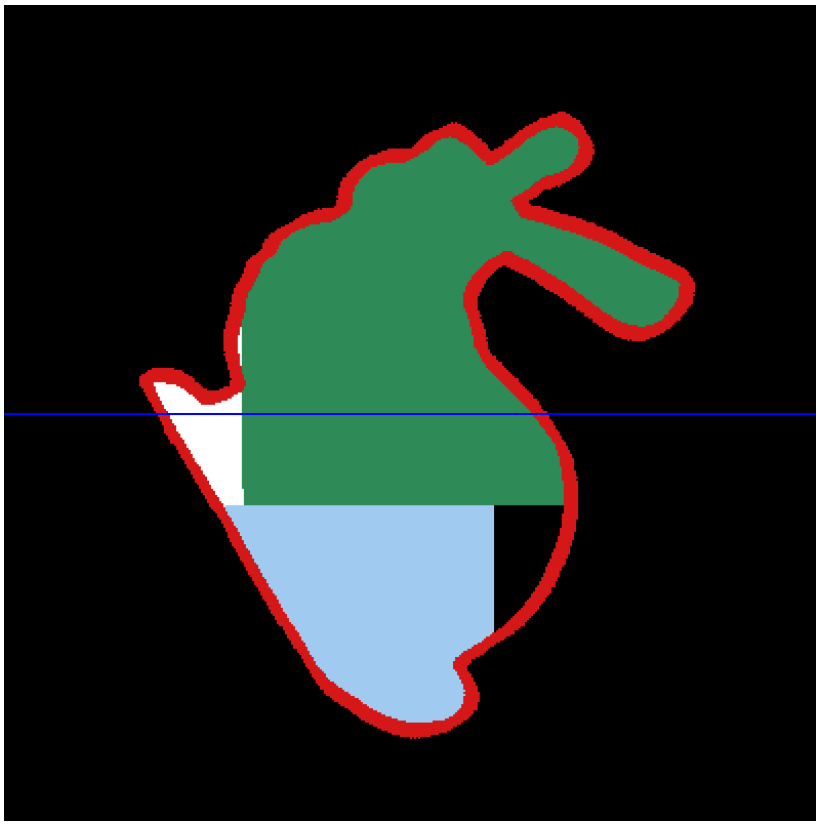

Figure 5: Outcome of Phase 2. The white color depicts the material added in Phase 2, while the black color inside the bunny depicts the material carved. The other colors have the same meaning as in Fig. 4.

if we are adding material. Finally, the objective function is defined as

$$\Phi(\mathbf{x}) = \lambda_h |d_h(\mathbf{x})| + \lambda_v |d_v(\mathbf{x}) - d_{v_{p1}}|, \tag{20}$$

where λ_h and are λ_v are positive parameters, d_h is the distance between $\mathbf{c}'_m$ and the buoyancy line of action, and d_v is the vertical distance between $\mathbf{c}'_m$ and $\mathbf{c}_b$. The vertical distance between $\mathbf{c}_m$ at the end of Phase 1 and $\mathbf{c}_b$ is $d_{v_{p1}}$.

We want to find $\mathbf{x}$ for which $\mathbf{c}'_m(\mathbf{x})$ has zero distance horizontally from $\mathbf{c}_b$ ($d_h = 0$), while the vertical distance between $\mathbf{c}'_m$ and $\mathbf{c}_b$ should remain as close as possible to the vertical distance at the end of Phase 1 between $\mathbf{c}_m$ and $\mathbf{c}_b$. The best candidate voxel is the one that minimizes Φ.

For example, in the 2D case, $d_h(\mathbf{x}) = d_x(\mathbf{x})$ and $d_v(\mathbf{x}) = d_y(\mathbf{x})$. If $\lambda_h > \lambda_v$, we give more weight to the x-axis distance than to the y-axis distance, and likewise for the converse. Generally, we want $\lambda_h \gg \lambda_v$, since we know that $\mathbf{c}_m$ is already as low as possible, and we must keep a balance between carving out voxels, which causes $\mathbf{c}_m$ to rise too much in height, and aligning $\mathbf{c}_m$ with the buoyancy line of action.

There is no difference between selecting $\mathbf{x}_a$ or $\mathbf{x}_c$ first; our method will yield the same results in either case. We begin by finding $\mathbf{x}_c$, by traversing all the filled voxels and picking one that minimizes Φ. We carve this voxel at $\mathbf{x}_c$, and update m and $\mathbf{c}_m$ of the object. Next, we find $\mathbf{x}_a$ by traversing all the carved voxels and picking one that minimizes Φ. We add material at this voxel at

Object	Voxels	Carved	P2
Bunny (Fig. 6)	87672	43777	342
Duck (Fig. 7)	78500	45080	368
Totoro (Figs. 9, 8)	132648	49631	919

Table 1: Summary of results. Column P2 gives the number of times we transferred material from one voxel to another in Phase 2.

$\mathbf{x}_a$, and similarly update m and $\mathbf{c}_m$ of the object. Since we are trying to minimize Φ, $\mathbf{x}_a$ and $\mathbf{x}_c$ are going to be on opposite sides of the buoyancy line. We repeat the above procedure until $\mathbf{c}_{b_h} = \mathbf{c}'_{m_h}$, such that the gravity and buoyancy forces are aligned along the same line of action.

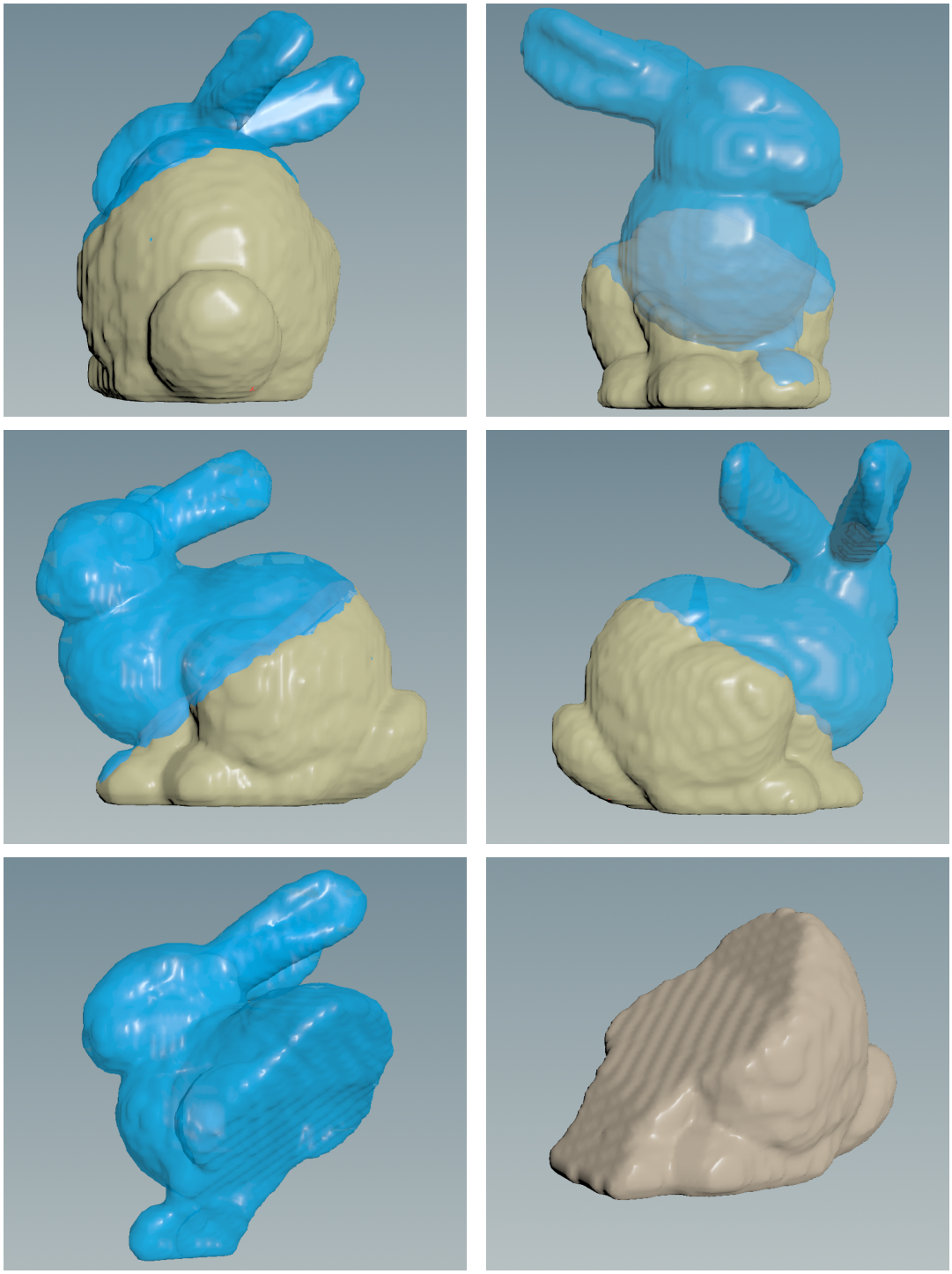

Figure 6: Stanford Bunny: blue denotes carved area, beige denotes voxels with material.

4.5 Current Results

We applied our method to the 3D objects listed in Table 1. These objects are represented via a uniformly sampled point cloud Obj file, where every point represents the center of a voxel.

We tested our method on a computer with 2.5 GHz Intel i7 Core processor and 16GB RAM, running OS X. Our implementation is in Python v2.7.10 and we used Houdini FX v15 to visualize the results.

Fig. 6 shows the result of our optimization on the Stanford Bunny. To position $\mathbf{c}_m$ on the same action line as the buoyant force, our method carves material from one side of the bunny and adds it to the other side. The same behavior can be seen in Fig. 7.

Fig. 8 shows how the material of a Totoro object is carved for floating. Notice that the material is not evenly carved around the fluid plane due to the need to balance the buoyant and gravitational forces. Fig. 9 shows the carved and filled parts of the object. Fig. 10 shows our floating experiment setup and fabrication results.

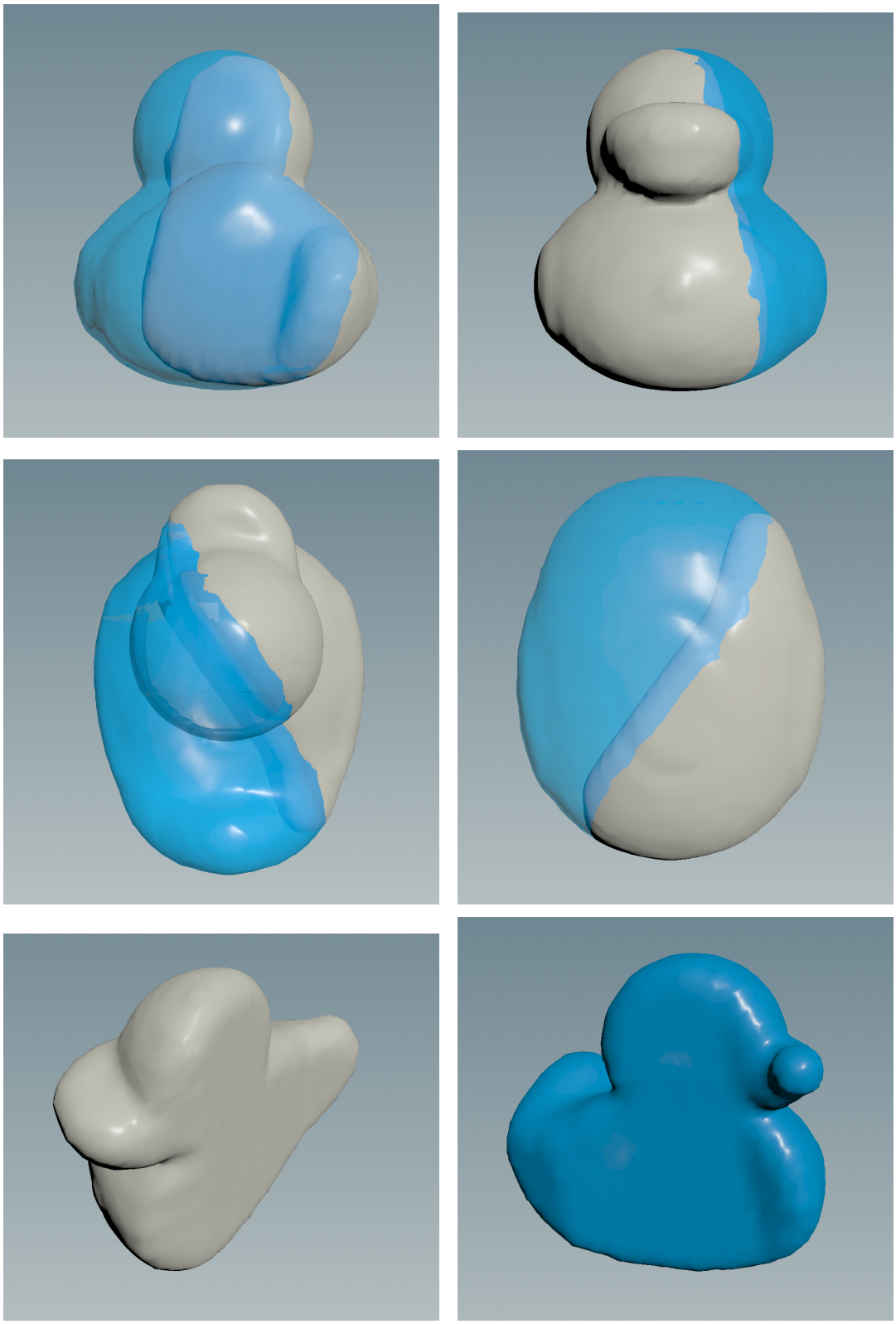

Figure 7: Rubber Duck: blue denotes carved area, gray denotes voxels with material.

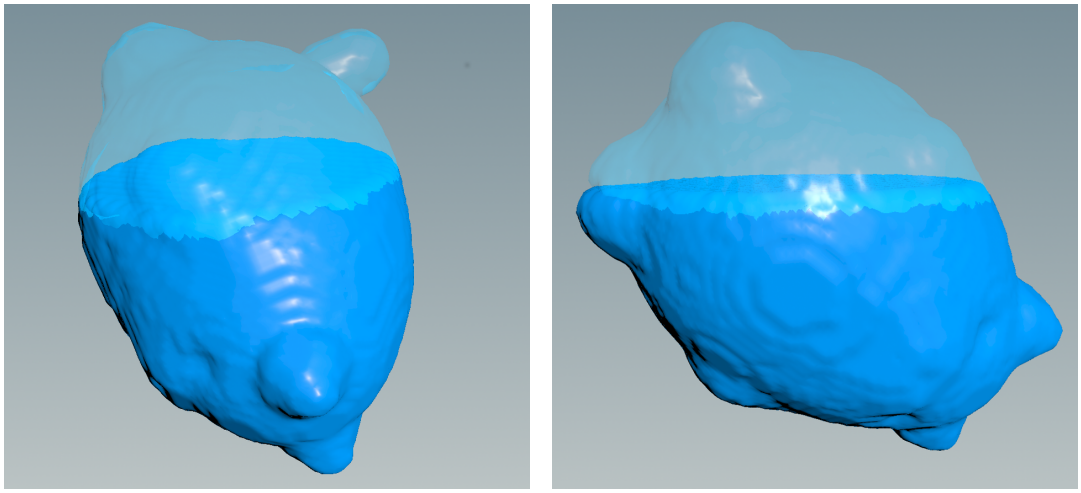

Figure 8: Submerged Totoro: dark blue denotes submerged area, transparent material denotes carved area.

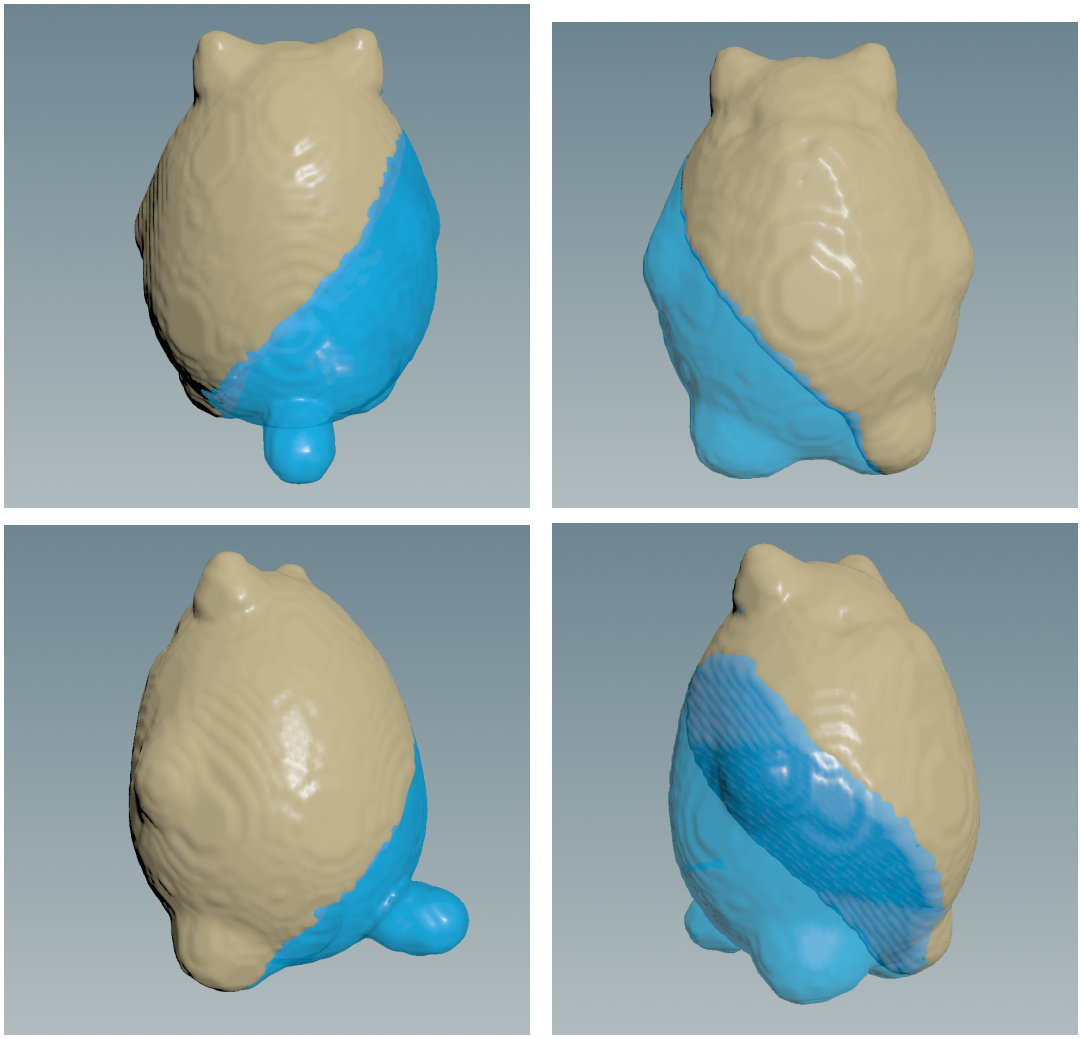

Figure 9: Totoro: blue denotes carved area, beige denotes area with material.

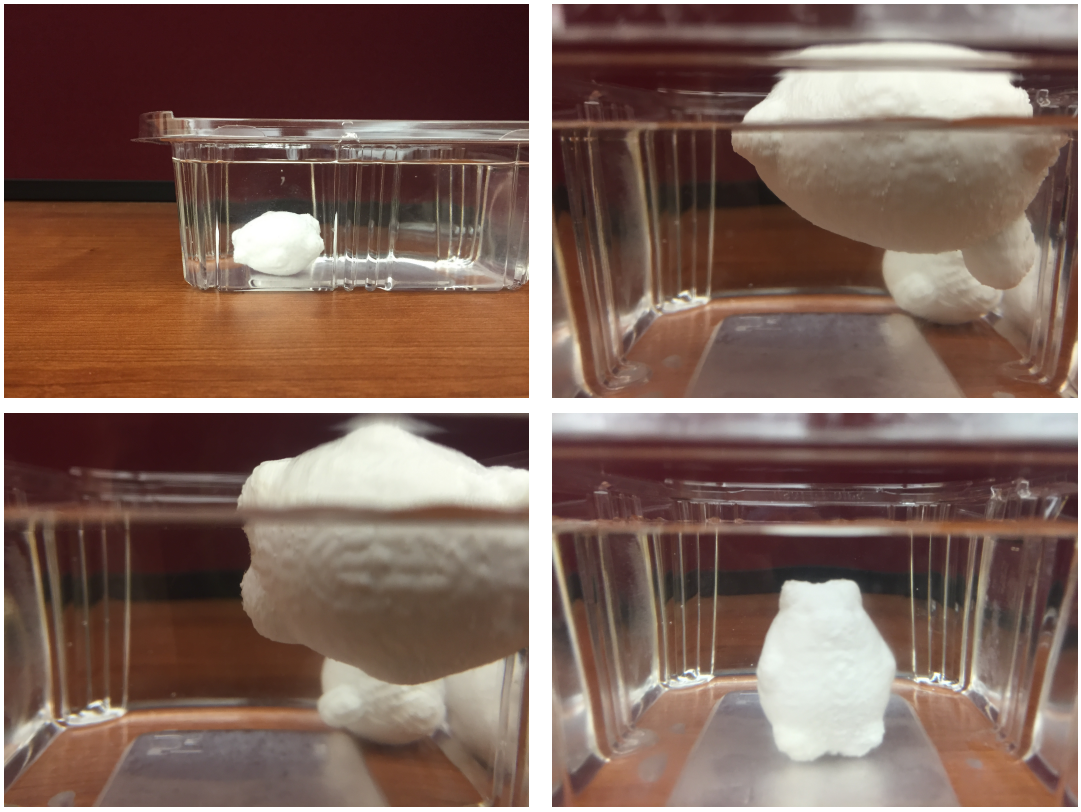

Figure 10: Totoro floating in different orientations. Starting from the upper-left corner and turning clock-size, we display our experiment setup, beginning with a non-floating totorto, totoro floating on the back, a totoro without applying the method, and a totoro floating on the side.

4.6 Summary

We proposed a method that, given an input 3D solid object model and a fluid plane, produces an object that floats in accordance with this fluid plane. The key idea was to optimally manipulate the object's inner material so as to balance the forces of buoyancy and gravity while lowering the center of mass as much as possible.

Our current approach requires traversing all the voxels several times in the different phases. It would be interesting to explore traversing the inner material of the object in a more efficient multi-resolution manner. In later chapters we discuss alternative, non-voxel based formulations to attack the problem.

Our method may not always succeed. For example, Phase 1 may fail in cases where the fluid plane is positioned such that there is insufficient inner material to carve in order to balance the forces of gravity and buoyancy. Additionally, Phase 2 may cause the center of mass to rise too much, such that it is no longer below the center of buoyancy. One avenue to counteracting such failures is altering the shape of the object in certain regions (including the shell), so that the altered object remains visually similar to the original shape; e.g., by adapting the technique from [19]. Another option is to use different types of inner material, similar to [3], which also offers more options to control the location of the center of mass.

The thickness of the shell depends on the robustness of the material used for 3D printing, as well as on the 3D printer, since 3D printers differ in their abilities. Thus, we plan to optimize the object shell thickness in a future version of our algorithm.

5 Alternative Formulations and Next Steps

5.1 3D Printing Limitations

There are various components involved in the pipeline when transforming a virtual design to a physical object. When designing a model for fabrication, we need to take each of these components into account:

- **3D printer resolution**
  Each 3D printer has a different printing resolution. For most consumer oriented 3D printers, the nozzle diameter is usually 0.40mm, which limits the size of a voxel. Also, the layer height (z-axis) resolution is different than the resolution on the x-y plane. This limits the precision of the fabricated model.

- **Fabrication filament**
  The plastic filament density is not consistent, and this is especially important for our problem, where we want to balance out the weight of the model and the weight of the displaced water.

- **External support material**
  It is not possible to extrude the plastic filament to a certain location without some base or support material under that location. This is why the 3D printer prints the model from the bottom up, adding support along the way. We manually remove support material from the final result. Changing the orientation on which the model is fabricated can also help in controlling the amount of support.

- **Inner support material**
  Similarly to external support material, internal support materials are needed for fabricating hollow models. However, internal support changes the mass and mass distribution of the output model, which undermines our optimization goal. One way to circumvent this problem is to fabricate the model in parts, remove the inner support material, and then glue the parts together [14].

Another important phase in the fabrication process is the transformation of the geometry into 3D printer instructions, i.e. the instruction set. The instruction set specifies where the printer should move the nozzle head, how much material should flow and so on. A *slicer* software (i.e. slices the geometry into layers) processes the 3D geometry, and produces the instruction set, which is usually a G-code file [9]. Each slicer has different options that users can configure, but these options do not allow full control over the transformation from geometry to the 3D printer instruction set. This transformation is a discrete approximation of the original model. For example, the resolution of voxels might be too small for the resolution of the nozzle. Also the slicer software might produce an instruction set that is a crude approximation, e.g. the staircase effect in Fig 11. Additionally, the instruction set might skip some features such as thin walls and so on. Though this approximation might be good enough in cases where the distribution of the mass of the object is not important, it does matter in our problem and in others where the design is directly influenced by physical properties of the object.

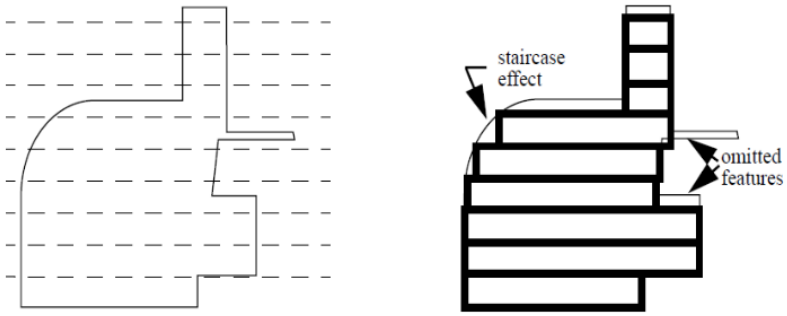

Figure 11: Input 3D geometry versus actual output

One possible course of action is to directly output the instruction set for the printer. That will give us control over the fabrication process. For example, we can fine tune what level of detail we want in each layer, thereby getting better a resolution. This approach is called Adaptive Slicing [22], and is not readily available in consumer slicing software.

To conclude, we presented some of the main difficulties and limitations in 3D printing. These limitations are especially relevant when fabricating a high-detail virtual 3D object with complex geometry and hollowing.

5.2 Euler-Lagrange and Lagrangian Multipliers

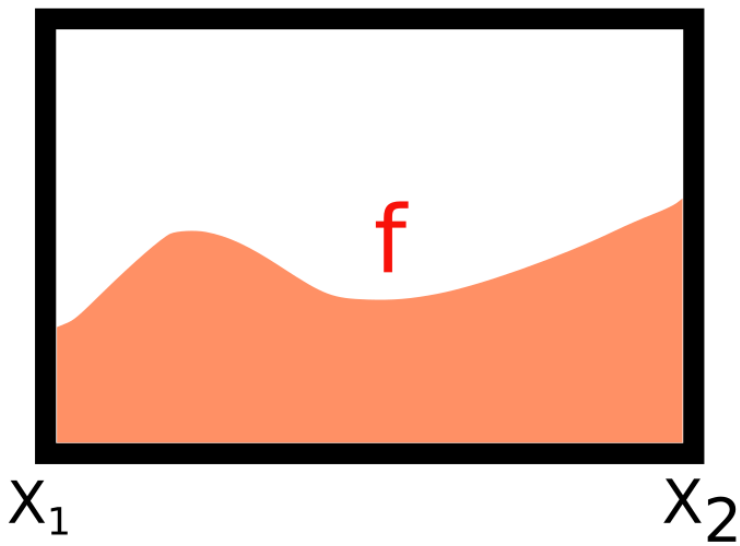

Figure 12: Function f describes the surface of the inner material of the model.

Calculus of Variations deals with functionals, i.e. the problem of finding a function for which the value of a specific integral is maximized or minimized. Since we can express c_m in terms of such a functional, we formulate our optimization problem as a variational optimization problem. Again, in order for the object to float, we are going to hollow specific parts of the inner volume. In order to simplify the discussion, let us tackle a simple box object in 2D, as in Fig. 12. Let $f(x)$ be a function that describes the height of the inner material of the object in position x, i.e. the space above f is hollowed. Our goal is to minimize the height of $\mathbf{c}_m$, so that it is as far away and below $\mathbf{c}_b$ as much as possible. Let c_{m_y} be the height of $\mathbf{c}_m$, and let c_{m_x} be the horizontal location of $\mathbf{c}_m$. We require that the horizontal distance between $\mathbf{c}_m$ and $\mathbf{c}_b$ be equal to zero, i.e. $c_{m_x} = c_{b_x}$. For floating stability, we require that $\mathbf{F}_g = \mathbf{F}_b$, so that the mass of the object is fixed to the mass of the displaced water. The contribution of the shell to the mass of the object is calculated separately.

To that end, we want to find a function f that minimizes the height of the center of mass. In 2D, we want that $\mathbf{c}_m$ to be as low as possible and under $\mathbf{c}_b$, i.e. we want to minimize c_{m_y}. In calculus of variations, Euler-Lagrange and Lagrange Multipliers allows us to solve a type of optimization problems that are subject to equality constraints.

Recall equation (2). Assuming $\rho(\mathbf{x})$ is constant throughout the material, and the bounds of the inner material of the object are $[x_1, x_2]$ and $[y_1, y_2]$ as in Fig. 12, then the 2D version will be:

$$c_{m_y} = \frac{\rho}{m} \int_{x_1}^{x_2} \int_{y_1}^{y_2} y dy dx \tag{21}$$

Recall that $f(x)$ describes the height of the inner material (the surface of for the 3D case), and let f be integrable on $[x_1, x_2]$. Without loss of generality, assuming the base of the object is at $y_1 = 0$:

$$c_{m_y} = \frac{\rho}{m} \int_{x_1}^{x_2} \int_0^f y dy dx = \frac{\rho}{2m} \int_{x_1}^{x_2} f^2 dx \tag{22}$$

Similarly:

$$c_{m_x} = \frac{\rho}{m} \int_{x_1}^{x_2} \int_0^f x dy dx = \frac{\rho}{m} \int_{x_1}^{x_2} x f dx \tag{23}$$

Finally, we constrain the target mass m to be equal to m_w:

$$m = \frac{\rho}{m} \int_{x_1}^{x_2} f dx = m_w \tag{24}$$

Given the equations (22), (23),(24), we want to minimize (22) under the constraints in (23) and (24).

$$\begin{aligned} \text{minimize } F &= \int_{x_1}^{x_2} f(x)^2 dx \\ \text{subject to } A &= \int_{x_1}^{x_2} x f(x) dx \\ B &= \int_{x_1}^{x_2} f(x) dx \end{aligned}$$

We minimize F in respect to:

$$G = \int_{x_1}^{x_2} \left(x f(x) - \frac{A}{x_2 - x_1} \right) dx = 0 \tag{25}$$

$$H = \int_{x_1}^{x_2} \left(f(x) - \frac{B}{x_2 - x_1} \right) dx = 0 \tag{26}$$

Let us denote λ_1 and λ_2 as the Lagrangian multipliers. Both are scalars and unknown. The Lagrangian associated with the constrained problem is:

$$\Psi(f, \lambda_1, \lambda_2) = F + \lambda_1 G + \lambda_2 H \tag{27}$$

The Euler-Lagrange differential equation is:

$$\frac{\partial F}{\partial f} - \frac{d}{dx} \frac{\partial F}{\partial f'} + \lambda_1 \left(\frac{\partial G}{\partial f} - \frac{d}{dx} \frac{\partial G}{\partial f'} \right) + \lambda_2 \left(\frac{\partial H}{\partial f} - \frac{d}{dx} \frac{\partial H}{\partial f'} \right) = 0 \tag{28}$$

Our objective is to solve the above in order to find λ_1 and λ_2. We get:

$$\frac{\partial F}{\partial f} = 2f \tag{29}$$
$$\frac{\partial G}{\partial f} = x \tag{30}$$
$$\frac{\partial H}{\partial f} = 1 \tag{31}$$
$$\frac{\partial F}{\partial f'} = 0 \tag{32}$$
$$\frac{\partial G}{\partial f'} = 0 \tag{33}$$
$$\frac{\partial H}{\partial f'} = 0 \tag{34}$$

Substituting the above into equation(28):

$$2f + \lambda_1 x + \lambda_2 = 0 \tag{35}$$

Therefore:

$$f = -\frac{\lambda_1 x + \lambda_2}{2} \tag{36}$$

Substituting the above into (25) we get:

$$\int_{x_1}^{x_2} -\frac{\lambda_1 x + \lambda_2}{2} x - \frac{A}{x_2 - x_1} dx = \int_{x_1}^{x_2} -\frac{\lambda_1 x^2 + \lambda_2 x}{2} - \frac{A}{x_2 - x_1} dx = 0$$
$$\int_{x_1}^{x_2} \lambda_1 x^2 + \lambda_2 x + \frac{2A}{x_2 - x_1} dx = 0$$
$$\left(\frac{x_2^3 - x_1^3}{3}\right) \lambda_1 + \left(\frac{x_2^2 - x_1^2}{2}\right) \lambda_2 + 2A = 0$$

And from (26):

$$\int_{x_1}^{x_2} -\frac{\lambda_1 x + \lambda_2}{2} - \frac{B}{x_2 - x_1} dx = 0$$
$$\int_{x_1}^{x_2} \lambda_1 x + \lambda_2 + \frac{2B}{x_2 - x_1} dx = 0$$
$$\frac{x_2^2 - x_1^2}{2} \lambda_1 + (x_2 - x_1) \lambda_2 + 2B = 0$$

To obtain λ_1, λ_2 we solve the system:

$$\left(\frac{x_2^3 - x_1^3}{3}\right) \lambda_1 + \left(\frac{x_2^2 - x_1^2}{2}\right) \lambda_2 + 2A = 0 \tag{37}$$
$$\left(\frac{x_2^2 - x_1^2}{2}\right) \lambda_1 + (x_2 - x_1) \lambda_2 + 2B = 0 \tag{38}$$

Where A, B, x_1, and x_2 are known.

Solving the linear system above for the 2D box case, will result in f to a straight line. Furthermore, these results can be easily expanded to 3D.

In conclusion, we showed how to formulate the problem in terms of Calculus of Variations for the goal of designing an object that floats. However, in more complex models with irregularities in the geometry, an analytic solution that determines the shape of f is harder to obtain. We can also discretize the above minimization above to a non-linear optimization problem. We discuss this discretization in the following section 5.3

5.3 Column Height Optimization

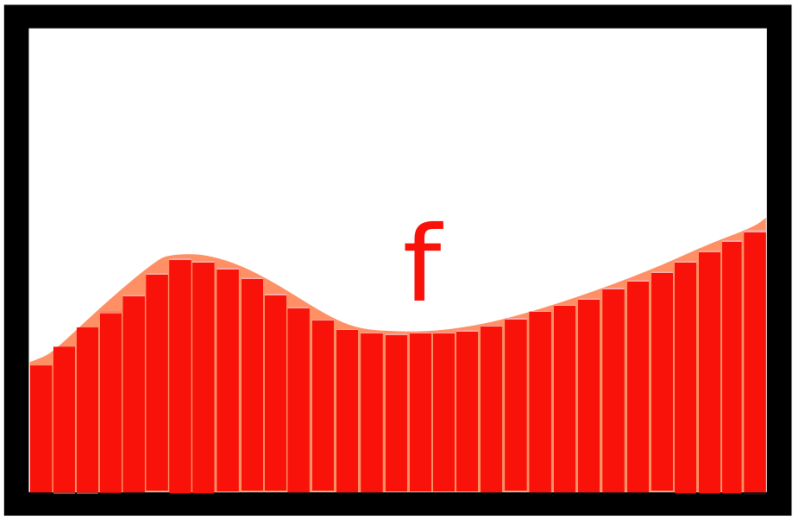

Figure 13: Columns approximation to the area under f.

Our main approach consisted of optimizing the design of the object via the discrete voxel domain. However, there are other mathematical models. Instead of discretizing the object into voxels, we discretize it to columns. Specifically, we divide the inner material to columns, and optimizing the height of these columns. The base of each column is a rectangle of the same size, and so is the top of the column. The height of each column varies between 0 and the maximum possible height before the column overlaps the shell of the object. Again, to simplify the discussion, let us tackle the problem in 2D. We assume that for each column i, the width is constant and equal to Δx. As in section 5.2 ,we want c_m to be as low as possible for the object to be more stable in water, under the constraints that c_m is on the buoyancy line of action, i.e. $c_{mx} = c_{bx}$, and $F_g = F_b$, which means the mass of the object is fixed and determined by mass of the displaced water m_w. y_i is the initial height of h_i, as determined by the object's shell.

$$c_{m_y} = \frac{\rho}{m} \sum_i \int_{x_i}^{x_i+\Delta x} \int_{y_i}^{y_i+h_i} y dy dx \approx \frac{\rho \Delta x}{2m} \sum_i 2y_i h_i + {h_i}^2 \tag{39}$$

And:

$$c_{m_x} = \frac{\rho}{m} \sum_i \int_{x_i}^{x_i+\Delta x} \int_{y_i}^{y_i+h} x dy dx \approx \frac{\rho}{2m} \sum_i 2h_i(x_i \Delta x + \Delta x^2) \tag{40}$$

Then the optimization problem is:

$$
\begin{aligned}
\text{minimize } c_{my} &= \frac{\rho \Delta x}{2m} \sum_i 2y_i h_i + {h_i}^2 \\
\text{subject to } c_{mx} &= \frac{\rho}{2m} \sum_i 2h_i(x_i \Delta x + \Delta x^2) = c_{bx} \\
m &= \Delta x \rho \sum_i h_i = m_w \\
y_{imin} &\leq h_i < y_{imax}
\end{aligned}
$$

The unknowns are the h_is. Since the function we are optimizing is convex, the local minimum will also be the global minimum [17]. Similar to the previous formulation in 5.2, we can easily expand our results to 3D.

In summary, we showed how to formulate the floating goal as an minimization problem constraints in terms of column heights. We partition the internal volume of the object into columns with the same base size, and then describe the mass and mass distribution of the object as functions of the column heights. We can use a numerical solver such as [8] or [1] to solve the optimization problem. Additionally, it would be interesting to investigate a multi-width column approach, where each columns has a different width, depending on the geometry of the object.

6 Conclusion

The focus of this thesis proposal is a novel method for generating floating objects. The input is a 3D object and a waterline, which determines the floating orientation. The output is an object with the same external appearance that floats in water. My method is not limited to water, but works in any fluid. This method can be seen as part of an optimization-based framework that can be adapted for goals other than designing objects that float. For example, the method can be adapted to solve other design goals described in previous work, e.g. transforming objects that are initially unstable, to stable objects that stand as in [19], or spin as in [3]. I also show how to mathematically formulate the optimization problem as a Calculus of Variations problem, and also as a non-linear minimization problem with constraints. These alternative formulations shows us an even more general way to approach designing objects that fit one or more real-world design goals.

References

[1] APS, M. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015.

[2] AZARI, A., AND NIKZAD, S. The evolution of rapid prototyping in dentistry: a review. *Rapid Prototyping Journal 15*, 3 (2009), 216–225.

[3] BÄCHER, M., WHITING, E., BICKEL, B., AND SORKINE-HORNUNG, O. Spin-It: Optimizing moment of inertia for spinnable objects. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH) 33*, 4 (2014), 96:1–96:10.

[4] BERMAN, B. 3-d printing: The new industrial revolution. *Business horizons 55*, 2 (2012), 155–162.

[5] ISHIGURO, Y., AND POUPYREV, I. 3d printed interactive speakers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2014), ACM, pp. 1733–1742.

[6] JACOBS, S., GRUNERT, R., MOHR, F. W., AND FALK, V. 3d-imaging of cardiac structures using 3d heart models for planning in heart surgery: a preliminary study. *Interactive cardiovascular and thoracic surgery 7*, 1 (2008), 6–9.

[7] KESNER, S. B., AND HOWE, R. D. Design principles for rapid prototyping forces sensors using 3-d printing. *Mechatronics, IEEE/ASME Transactions On 16*, 5 (2011), 866–870.

[8] KRAFT, D., ET AL. *A software package for sequential quadratic programming*. DFVLR Obersfaffeuhofen, Germany, 1988.

[9] KRAMER, T. R., PROCTOR, F. M., AND MESSINA, E. The nist rs274. *National Institute of Standards and Technology (NIST), NISTIR 6556* (2000).

[10] LEUKERS, B., GÜLKAN, H., IRSEN, S. H., MILZ, S., TILLE, C., SCHIEKER, M., AND SEITZ, H. Hydroxyapatite scaffolds for bone tissue engineering made by 3d printing. *Journal of Materials Science: Materials in Medicine 16*, 12 (2005), 1121–1124.

[11] LI, M., YANG, Z., AND LIU, Y. Sea depth measurement with restricted floating sensors. *ACM Transactions on Embedded Computing Systems (TECS) 13*, 1 (2013), 1.

[12] LIPSON, H., AND KURMAN, M. *Fabricated: The new world of 3D printing*. John Wiley & Sons, 2013.

[13] LU, L., SHARF, A., ZHAO, H., WEI, Y., FAN, Q., CHEN, X., SAVOYE, Y., TU, C., COHEN-OR, D., AND CHEN, B. Build-to-last: Strength to weight 3d printed objects. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 97.

[14] LUO, L., BARAN, I., RUSINKIEWICZ, S., AND MATUSIK, W. Chopper: Partitioning models into 3D-printable parts. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (Dec. 2012).

[15] MIRONOV, V., BOLAND, T., TRUSK, T., FORGACS, G., AND MARKWALD, R. R. Organ printing: computer-aided jet-based 3d tissue engineering. *TRENDS in Biotechnology 21*, 4 (2003), 157–161.

[16] MUSIALSKI, P., AUZINGER, T., BIRSAK, M., WIMMER, M., AND KOBBELT, L. Reduced-order shape optimization using offset surfaces. *ACM Transactions on Graphics (ACM SIGGRAPH 2015) 34*, 4 (Aug. 2015), 102:1–102:9.

[17] NOCEDAL, J., AND WRIGHT, S. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

[18] PARK, Y.-L., CHEN, B.-R., AND WOOD, R. J. Design and fabrication of soft artificial skin using embedded microchannels and liquid conductors. *Sensors Journal, IEEE 12*, 8 (2012), 2711–2718.

[19] PRÉVOST, R., WHITING, E., LEFEBVRE, S., AND SORKINE-HORNUNG, O. Make It Stand: Balancing shapes for 3D fabrication. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH) 32*, 4 (2013), 81:1–81:10.

[20] SACHS, E., CIMA, M., WILLIAMS, P., BRANCAZIO, D., AND CORNIE, J. Three dimensional printing: rapid tooling and prototypes directly from a cad model. *Journal of Manufacturing Science and Engineering 114*, 4 (1992), 481–488.

[21] SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2004), ACM, pp. 175–184.

[22] TYBERG, J., AND HELGE BØHN, J. Local adaptive slicing. *Rapid Prototyping Journal 4*, 3 (1998), 118–127.

[23] YOUNG, H., FREEDMAN, R., AND FORD, A. *Sears and Zemansky's University Physics*. No. v. 1 in Addison-Wesley series in physics. Addison-Wesley, 2008.