DOI: 10.1111/cgf.12810
EUROGRAPHICS 2016 / J. Jorge and M. Lin
(Guest Editors)

Volume 35 (2016), Number 2

Buoyancy Optimization for Computational Fabrication

L. Wang and E. Whiting

Dartmouth College, USA

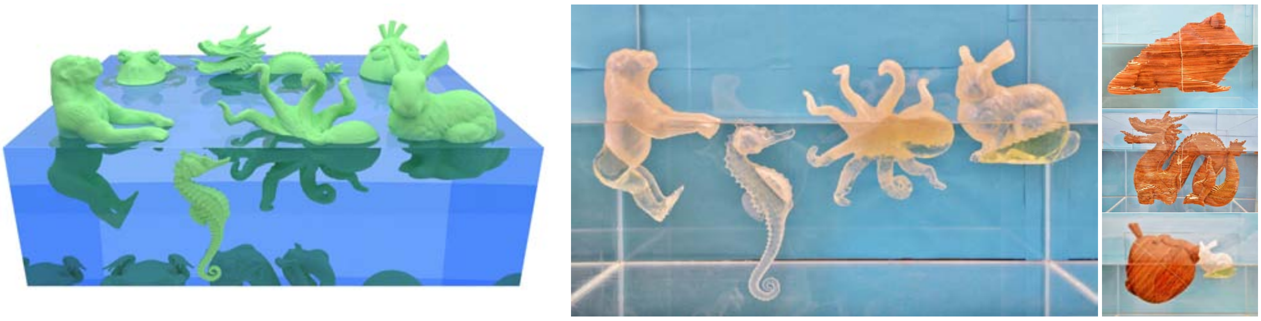


Figure 1: We introduce a novel design objective for 3D objects that float in user-specified ways. The user chooses the desired orientation and waterline (left). An optimization determines the internal configurations necessary for the shapes to float in the desired positions. The shapes can then be 3D printed and dropped in a water tank (right). For shapes too large to be efficiently 3D printed, we introduce a construction method using laser-cut, plywood slices (far right).

Abstract

This paper introduces a design and fabrication pipeline for creating floating forms. Our method optimizes for buoyant equilibrium and stability of complex 3D shapes, applying a voxel-carving technique to control the mass distribution. The resulting objects achieve a desired floating pose defined by a user-specified waterline height and orientation. In order to enlarge the feasible design space, we explore novel ways to load the interior of a design using prefabricated components and casting techniques. 3D printing is employed for high-precision fabrication. For larger scale designs we introduce a method for stacking lasercut planar pieces to create 3D objects in a quick and economic manner. We demonstrate fabricated designs of complex shape in a variety of floating poses.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

1. Introduction

The science of buoyancy dates back to Archimedes. Since he discovered his principle, the physics of flotation has been a well-studied topic [Lau11], leading to innovations such as the design of offshore platforms with complex underwater structures, and objects for entertainment such as floating furniture, toys, and sculpture. The pose of a floating object is defined by the object's shape, mass distribution and fluid displacement. These factors must be simultaneously optimized to float an object stably with a desired orientation and waterline position. Because of the number of constraints, designing buoyant objects by hand is a difficult, trial-and-error process.

This paper proposes a method to automatically determine the internal structure necessary to make a given shape float with a desired orientation and waterline. We show how to design shapes for both 3D printing, where the density of the material is greater than water, and laser-cut plywood, where the density is less. Since plywood cannot simply be hollowed to achieve a desired floating orientation, we develop techniques to design loading masses for our plywood models and to ensure they are watertight. The plywood laser-cutting approach enables us to fabricate large objects faster and more economically than typical 3D printing, and could be used to create full-sized floating furniture and sculpture.

We adopt the voxel carving method of [PWLSH13, BWBSH14] to manipulate internal mass distribution of an object. Leveraging this established optimization framework, we introduce a novel objective for stable floating objects, and incorporate constraints for manufacturable watertight designs. The user provides the mesh of a 3D object, desired waterline position and floating orientation, and the densities of the surrounding fluid and the fabrication material. The algorithm calculates the critical mechanical parameters of the model and determines a mass distribution such that the object floats stably in the desired way. With the mass distribution known, the interior of the 3D object is hollowed or loaded accordingly. Finally, the result is sent to a 3D printer or laser cutter for fabrication.

2. Related Work

Fabrication-oriented design. Designing objects for particular fabrication-oriented goals is a popular topic in the graphics literature. Recent work exploring special physical properties includes inflatable structure design [STK*14], metallophone design [UMI10], image from shadowing [BBAM12] and free-form model airplane design [UKSI14]. Work in guided sculpting [RAD12] and position-correcting cutting tools [RMD12] aims to allow novice craftsmen to construct computationally-designed shapes. Our work extends this theme into the realm of floating objects.

Shape optimization for balance. Techniques for designing balanced static objects [PWLSH13, CSB14] and spinnable objects [BWBSH14] optimize the internal mass distribution of the shape. The interior of the objects is discretized into voxels, and some voxels are hollowed to redistribute the object's mass. We adopt a similar design approach but focus on buoyancy analysis and the constraints for planar slice construction. Casting high density material to give further control over the mass distribution has also been discussed [BWBSH14], and we show how to apply this idea to both prefabricated high density elements and molded filling material.

Recently, [MAB*15] proposed a method to control the mass distribution of an object by offsetting surfaces to thicken or thin the object's wall. Their approach applies to several applications for optimizing physical properties, including balanced static objects, spinnable objects, and a special case of flotation where the object is totally submerged in fluid. Our methods investigate the more involved scenario of floating objects that are partially submerged. Further, we explore dual-material optimization and techniques which extend to multiple voids which are common necessities for the design of buoyant objects.

Fluid simulation. Fluid simulation presents another possible approach to buoyancy-oriented design [BMF07, DWLF12, CTG10]. Computational fluid dynamics simulates the motion of fluid flows described by Navier-Stokes equations, and is applied in the design of efficient automobiles, aircraft and ship hulls [AW95]. In computer graphics, fluid simulation is widely used for visualization and animation as special effects. In contrast to dynamic effects, we study static buoyancy and apply an analytic model.

Ship design. A long legacy of ship and submarine design exists [BS98,BP13], where the focus is on optimizing hull shape for economy, efficiency, speed and safety. A review of key developments in Computer-Aided Ship Design is given by Nowacki [Now10]. Unlike hull design, we are presented with a particular shape and asked to make it float without changing its external appearance.

Planar-piece design. Recent work has also examined fabrication using planar pieces [MUS14, SP13, SP12, CPMS14]. This line of work is concerned with the structural soundness and visual effect of the interlocking planar pieces. In our work, we assume that stacking and gluing together the pieces is sufficient, focus on designing those pieces to be properly balanced and watertight. We also demonstrated effective ways to position loads inside the planar-piece designs.

3. Buoyancy Background

Our floating objects are designed to achieve a stable equilibrium when placed in fluid. Equilibrium ensures the object remains still in its designed orientation, and stability ensures that the object restores its orientation when disturbed.

3.1. Equilibrium

An unrestrained object is in mechanical equilibrium when the total force and total moment acting on the object are zero. The total force vanishes if the gravitational force $\mathbf{F_G}$ is equal and opposite to the buoyancy force $\mathbf{F_B}$ produced by the weight of the displaced fluid (Archimedes' principle). The total moment vanishes if the moment of gravity $\mathbf{C_G} \times \mathbf{F_G}$ is equal and opposite to the moment of buoyancy $\mathbf{C_B} \times \mathbf{F_B}$, where $\mathbf{C_G} = (x_{\mathbf{C_G}}, y_{\mathbf{C_G}}, z_{\mathbf{C_G}})$ is the center of gravity of the object and $\mathbf{C_B} = (x_{\mathbf{C_B}}, y_{\mathbf{C_B}}, z_{\mathbf{C_B}})$ is the center of buoyancy or the center of gravity of the displaced fluid. Note that when the total force vanishes, i.e. $\mathbf{F_G} = -\mathbf{F_B}$, the total moment is independent of the origin of the coordinate system. Formally, mechanical equilibrium leads to the constraints:

$$\mathbf{F_G} + \mathbf{F_B} = \mathbf{0} \tag{1}$$

$$\mathbf{C_G} \times \mathbf{F_G} + \mathbf{C_B} \times \mathbf{F_B} = \mathbf{0} \tag{2}$$

We assume the direction of gravity to be $\mathbf{g} = (0, 0, -\| \mathbf{g} \|)$.

3.2. Stability

In addition to mechanical equilibrium, a stable floating object must restore itself to equilibrium after a small disturbance is applied. When an object is tilted there is a change in fluid displacement, and thus in $\mathbf{C_B}$, which results in a *restoring moment*. To achieve stability, the restoring moment must counteract the tilt. For a small tilt angle, α, the moment is given by $-\alpha(z_M - z_{\mathbf{C_G}})mg$, where mg is the weight of object and z_M is the height of the *metacenter* [Lau11]:

$$z_M = z_{\mathbf{C_B}} + (I/V) \tag{3}$$

The metacenter lies above the center of buoyancy and remains fixed as an object tilts. V is the volume of displaced fluid, $z_{\mathbf{C_B}}$ is the

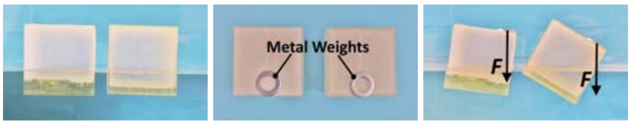


Figure 2: (Left) Two cubes with same weight, shape and waterline position. The left cube has larger metacentric height. (Middle) Identical external force applied by metal weight. (Right) Left cube exhibits smaller tilt angle.

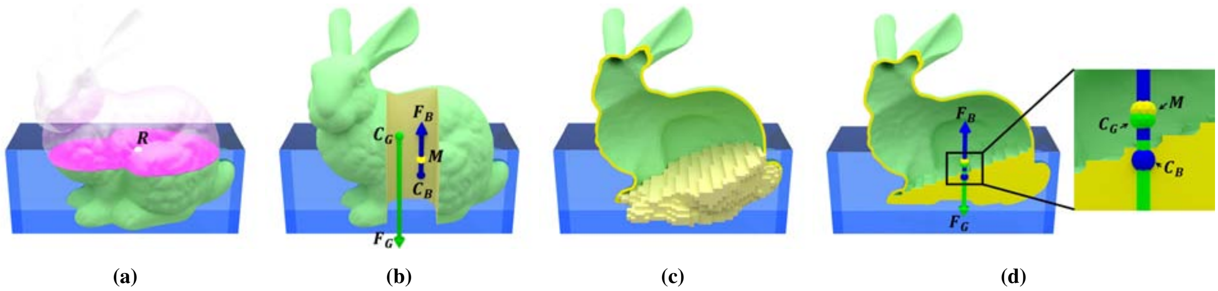


Figure 3: Buoyancy Optimization Overview: (a) User input: solid model placed in fluid. The fluid displacement (green), waterline area (purple) and center of roll (R, centroid of waterline area) are calculated for optimization. (b) Initially, center of gravity ($\mathbf{C_G}$) and center of buoyancy ($\mathbf{C_B}$) are not vertically aligned. Additionally, the gravity force (green arrow, $\mathbf{F_G}$) and buoyancy force (blue arrow, $\mathbf{F_B}$) are typically not equal magnitude. (c) A minimal thickness boundary shell is fixed. The interior space is voxelized and then hollowed. (d) Result: center of gravity and center of buoyancy align vertically, and center of gravity lies below the metacenter (M). Gravitational force negates the buoyancy force.

height of center of buoyancy and I is the smallest second moment of the waterline area (see Appendix).

We define our metric for stability as the *metacentric height*: the distance between the metacenter and the center of gravity ($z_M - z_{\mathbf{C_G}}$). fluid The restoring moment can only counteract a tilt if the center of gravity $z_{\mathbf{C_G}}$ is below z_M, so an object must satisfy $z_M > z_{\mathbf{C_G}}$. The larger the metacentric height, the larger the restoring moment and the harder the object is to overturn. Intuitively, the term I/V in (3) implies that a wide and shallow object has a higher metacenter for a given displaced volume. Our work aims to control z_M without changing the external shape. For example, Fig. 2 shows two cubes of different stability. The more stable cube (left) tilts by a smaller angle under the same external force.

Note that if the object is totally submerged (i.e., $I = 0$), the metacenter coincides with center of buoyancy. For totally submerged bodies (e.g., submarines, balloons, fish), constraint $z_M > z_{\mathbf{C_G}}$ requires that the center of gravity lies directly below the center of buoyancy. For bodies floating stably on the fluid surface (e.g., ships, icebergs, ducks), the center of gravity can be above the center of buoyancy but must be below the metacenter.

4. Optimization Overview

Our optimization maximizes the stability of an object for the desired floating pose:

$$\max_{v_i} \quad z_M - z_{\mathbf{C_G}} \tag{4}$$

$$s.t.: \quad \mathbf{F_G} + \mathbf{F_B} = \mathbf{0} \tag{5}$$

$$\mathbf{C_G} \times \mathbf{F_G} + \mathbf{C_B} \times \mathbf{F_B} = \mathbf{0} \tag{6}$$

$$z_M > z_{\mathbf{C_G}} \tag{7}$$

$$0 \leq v_i \leq 1 \quad \forall\, v_i \tag{8}$$

Stability is measured by metacentric height $z_M - z_{\mathbf{C_G}}$ as defined in Section 3.2. $\mathbf{F_G}$ is the gravitational force on the object. $\mathbf{F_B}$ is the upward buoyancy force, and $\mathbf{C_B}$ is the center of buoyancy, as defined in Section 3.1. Both values are obtained by integration on the mesh of displaced fluid. Section 5 describes our method for computing the fluid displacement and waterline area.

The buoyancy force $\mathbf{F_B}$, center of buoyancy $\mathbf{C_B}$ and height of metacenter z_M are defined by the input geometry and floating pose, and are thus held constant in the optimization. We assume the direction of gravity to be $\mathbf{g} = (0, 0, -\| \mathbf{g} \|)$, in which case (6) is implied by $x_{\mathbf{G}} = x_{\mathbf{B}}, y_{\mathbf{G}} = y_{\mathbf{B}}$.

An overview is illustrated in Fig. 3. The optimization is a linear programming problem where the variables are the fill variables, v_i, for the interior of the solid object. The fill variables control the mass distribution of the design, and are further explained in Sections 6 and 8.1-8.2. Refer to the Appendix (12)-(13) for definitions of $\mathbf{F_G}$ and $\mathbf{C_G}$ with respect to the fill variables.

5. Fluid Displacement

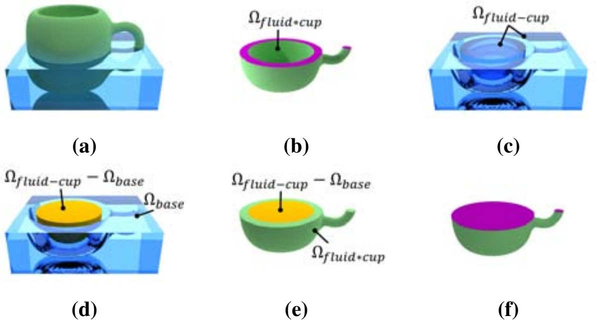
Central to computing buoyancy is the problem of computing the displaced fluid. The shape of the displaced volume is necessary to determine properties of buoyancy force ($\mathbf{F_B}$), center of buoyancy ($\mathbf{C_B}$) and waterline area, necessary to compute the second moment I. The displaced fluid can be found through constructive solid geometry techniques.

Displaced Volume. For a convex shape and other simple objects, the geometry of the displaced fluid is the intersection of the object and the fluid: if the fluid is Ω_{fluid} and the object is Ω_{object}, the fluid displacement is $\Omega_{object} \cap \Omega_{fluid}$ (hereafter denoted by $\Omega_{object*fluid}$).

However, buoyant designs commonly contain non-convex regions as voids to reduce the ratio of mass to displaced fluid and increase buoyancy (e.g. the cup in Fig. 4(a)). In this case, $\Omega_{object*fluid}$ will not give the correct displacement volume since it does not account for concavities. The correct displacement D_{object} of object Ω_{object} is given by:

$$D_{object} = \Omega_{object*fluid} \cup (\Omega_{fluid-object} - \Omega_{base}) \tag{9}$$

Fig. 4 illustrates computing the intersection and difference operations for a cup. The difference between the fluid and the object $\Omega_{fluid-object}$ can consist of multiple isolated components (Fig. 4 (c,d)). The component with the deepest vertex is named the base-component. Intuitively, the base-component Ω_{base} is the main body of fluid outside of Ω_{object} and is the remaining fluid after displacement. The base-component always exists and consists of only one



(a) (b) (c)

(d) (e) (f)

Figure 4: Fluid displacement: (a) Cup in fluid. (b) Intersecting the cup and fluid, $\Omega_{cup*fluid}$, underestimates total displaced fluid and waterline area (purple). (c) The difference of fluid and cup, $\Omega_{fluid-cup}$, consists of multiple components. (d) Subtracting the base component Ω_{base} from $\Omega_{fluid-cup}$ gives the interior volume of the cup (orange). (e) The intersection together with interior volume gives total fluid displacement. (f) Total waterline area.

continuous volume. The other components of $\Omega_{fluid-object} - \Omega_{base}$ are the components isolated by Ω_{object} from the exterior fluid due to concavities and considered as displaced fluid.

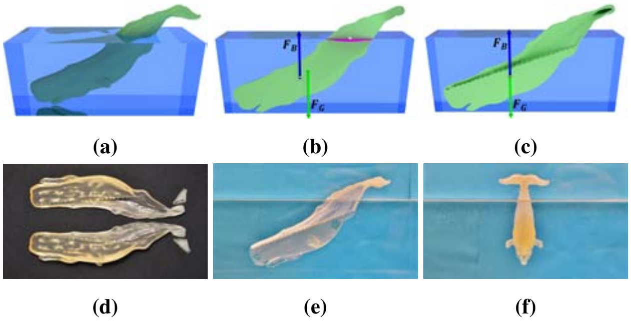
Waterline Area. Given the fluid displacement D_{object}, the waterline area corresponds to the 2D component of D_{object} that lies along the upper surface of the fluid (see Fig. 4(f)). We find the waterline area by simply selecting all mesh faces on the top surface of D_{object}. Given the shape of the waterline area, the height of the metacenter z_M is computed by Eq. (3).

6. Design for 3D Printing

We use 3D printing to prototype high-resolution floating models. The user provides the surface mesh of a solid 3D object along with the floating height and orientation. The optimization then proceeds by voxelizing the interior and hollowing individual voxels such that buoyancy and stability constraints are achieved. Since the density of polymerized 3D printing material (1.18-1.19 g/cm^3) is greater than the density of pure water, hollowing the shape is sufficient to achieve a wide range of floating configurations. The full process is illustrated in Fig. 5.

Voxelization. The interior of the input object is hollowed such that the mass distribution satisfies the constraints (Eq. 5-7) and maximizes floating stability. We discretize the interior space of the object by voxelization, following the method of [BWBSH14, PWLSH13]. Each voxel is a mass element associated with a fill variable v_i, where $0 \leq v_i \leq 1$. The voxel is hollowed when $v_i = 0$ and solid when $v_i = 1$. For fabrication purposes a partially filled voxel is not possible, so we round v_i to a binary value. Error introduced after thresholding is reduced within fabrication tolerances by using sufficiently fine voxels. We use the multi-resolution voxelization method from [BWBSH14] to accelerate the optimization.

To guarantee that the resulting model can be fabricated, we enforce a minimal wall thickness. The surface of the input mesh is offset to create a shell, which remains solid during the hollowing process. The offset surface is created by marching cubes [LC87] on a signed distance field [SPG03] and incorporated as a constraint in the shape optimization. See Section 9.1 for a selection of 3D printed results.



(a) (b) (c)

(d) (e) (f)

Figure 5: "Whale": (a) Desired floating orientation. (b) Before hollowing, buoyancy force ($\mathbf{F_B}$) and gravity force ($\mathbf{F_G}$) do not align vertically and therefore create a moment of rotation. The purple area is waterline area. (c) After optimization, buoyancy and gravitational forces are equal and opposite. The center of gravity is below metacenter. (d) 3D printed pieces of the whale model. (e,f) Actual floating pose, matching the desired orientation.

7. Planar Pieces Design

Large scale designs such as furniture or sculpture are prohibitively costly to fabricate with 3D printing, both in time and material expense. To realize large-scale designs for buoyant objects we explore fabrication using stacked layers of planar pieces. We used laser-cut plywood as a proof of concept in our experiments. Since the density of plywood is less than water, the space of feasible floating configurations cannot be fully explored by simply hollowing the shape. To expand the feasible space, we also investigate loading the interior with a dense, secondary material (see Section 8).

7.1. Overview

Similar to the process for 3D printing, users provide the mesh of an object together with the desired floating pose. Rather than outputting a hollowed object for 3D printing, however, our design tool generates patterns for laser cutting. The generated patterns are mapped onto a large sheet of plywood, which is then lasercut into many planar pieces. These planar pieces are manually stacked and glued together in a water tight manner to obtain the fabricated result. The process is shown in Fig. 6.

7.2. Slice Optimization

Given the 3D mesh, the input object is sliced into layers of equal thickness along a direction $\mathbf{r}$ (see Fig. 6). Direction $\mathbf{r}$ is chosen manually by the user since it influences both the appearance of the object and the ease of assembly. The width of each slice is determined by the thickness of the raw material, which in our case is a 1/8" plywood sheet. For accurate analysis, the sliced object rather than the input mesh is used to determine the center of buoyancy, volume of fluid displacement and position of metacenter. Note that a slice can contain holes and multiple disconnected parts, which

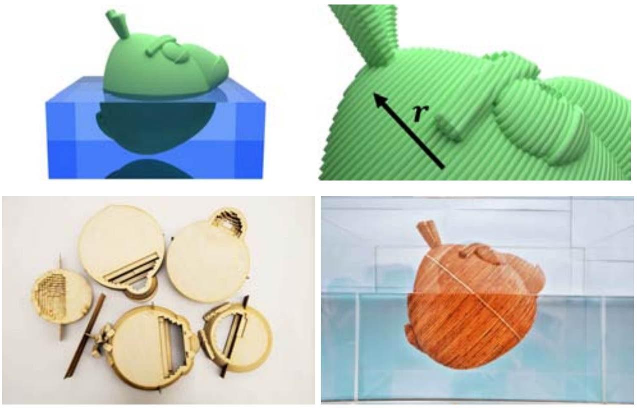
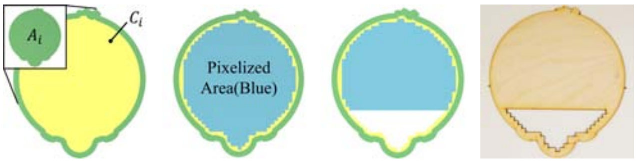


Figure 6: Planar pieces design: The input “Angry Bird” model (top-left) is sliced along the user-specified direction **r** (top-right). Slices are hollowed by our algorithm and cut from a sheet of plywood by a laser cutter (bottom-left). The model is manually assembled and scaffolding is removed, producing the final floating model (bottom-right).



(a) **(b)** **(c)** **(d)**

Figure 7: Sample slice from “Angry Bird”: The interior region is hollowed while preserving watertightness. (a) Yellow region C_i is the maximal area that is safe to cut while satisfying minimal side wall thickness, minimal overlapping thickness and solid overhanging area. This area is pixelized (b), and the optimization removes pixels inside the region to meet design constraints (c). The optimized slice is fabricated using a laser cutter (d).

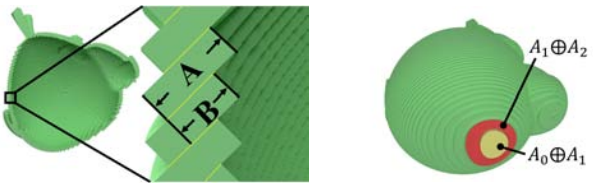


Figure 8: Watertight constraints: (Left) The thickness of the side wall A and the overlapping zone between two adjacent slices B must satisfy $A > T_{wall}$ and $B > T_{overlap}$. (Right) The constraint must be satisfied for the entire overlapping area on either side of the slice A_1. The area between A_0 and A_1 is shown in yellow, and the area between slices A_1 and A_2 is shown in red.

must be accounted for when determining the area of the slice that may be hollowed.

To perform the optimization, each slice is treated as a 2D contour and discretized into pixels. Analogously to the voxelization method of Section 6, each pixel is associated with a fill variable v_i, which determines whether or not the pixel remains in the slice after optimization. Plugging in the mass m_i and center of gravity $\mathbf{c}_i$ of each pixel, we find that the linear programming problem of (Equations 4 - 8) still captures the objective and constraints for planar pieces design. The shell mass M_s and center of mass $\mathbf{C_s}$ are now the totals over all slices after hollowing all interior pixels.

7.3. Watertight Constraints

As with our 3D printed designs, we constrain the planar piece designs to be watertight. For the object to to be watertight, the following must hold for all slices (see Fig. 8):

- The side wall thickness of each slice cannot be less than a minimal value $T_{wall} > 0$.
- The overlapping thickness between any two adjacent slices cannot be less than a minimal value $T_{overlap} > 0$. Notice that the overlapping thickness cannot be thicker than the side wall thickness, i.e. $T_{wall} \geq T_{overlap}$.
- The overhanging area of each slice must be solid. The highlighted red and yellow areas in Fig. 8 (right) are two overhanging areas of the Angry bird model.

Intuitively, we identify the largest interior region C_i on a given slice A_i that can be hollowed without introducing leaks. Only pixels in region C_i are associated with a fill variable v_j.

Formally, Let A_i be the i^{th} slice without hollowing. A_i possibly contains holes, islands and disconnected parts. SDF_{A_i} is the signed distance field of A_i [SPG03], where $SDF_{A_i}(\mathbf{x}) > 0$ if $\mathbf{x}$ is outside of A_i and $SDF_{A_i}(\mathbf{x}) < 0$, if inside. The slice itself is given by all points on and inside the boundary, $A_i = \{\mathbf{x}|SDF_{A_i}(\mathbf{x}) \leq 0\}$. We then define interiors of slice A_i as:

$$W(A_i) = \{\mathbf{x}|SDF_{A_i}(\mathbf{x}) \leq -T_{wall}\}$$
$$L(A_i) = \{\mathbf{x}|SDF_{A_i}(\mathbf{x}) \leq -T_{overlap}\}$$

Points in $W(A_i)$ have a distance which is at least T_{wall} to the boundary of A_i, while points in $L(A_i)$ have a distance at least $T_{overlap}$.

We then define the restricted area C_i inside of A_i where hollowing can occur, while guaranteeing a watertight model:

$$C_i = L(A_{i-1}) \cap W(A_i) \cap L(A_{i+1}) \qquad (10)$$

where A_{i-1} and A_{i+1} denote the adjacent slices of A_i. The object is cut into n slices, where $i \in \{1 \ldots n\}$ and slices A_0 and A_{n+1} are defined as empty, i.e. $\{\mathbf{x}|SDF_{A_0}(\mathbf{x}) \leq 0\} = \emptyset$ and $\{x|SDF_{A_{n+1}}(\mathbf{x}) \leq 0\} = \emptyset$.

While the conditions are properties of individual or adjacent slices, the global property of watertightness holds when assembled into the final structure. Refer to the Appendix for assertions on the watertight constraints.

In our implementation we find $W(A_i)$ and $L(A_i)$ for each slice using marching squares on the signed distance field. C_i is then the intersection of the relevant 2D polygonal shapes.

7.4. Slice Alignment

Each slice must be aligned globally with all other slices during construction. We add a narrow strip to every slice to help in this registration process. The red line in Fig. 9 (right) shows the strip used for positioning, which is positioned identically for all slices relative

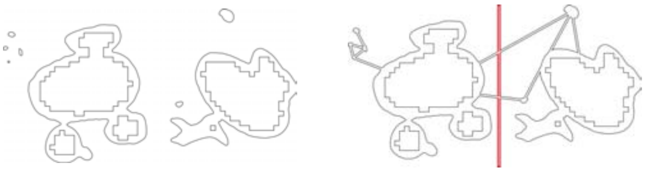


Figure 9: Slice connection and alignment: A sample slice from the dragon model (Fig. 11) shows several disconnected components (left). To aid positioning, isolated parts are connected by struts created using Delaunay triangulation (right). The red strip is used to align adjacent slices.

to a global coordinate system. The user can then align the strips to ensure that each slice is correctly positioned. This process can also be seen in the construction of the toad model in Fig. 10.

Since there can be multiple disconnected parts in a single slice (Fig. 9 (left)), a framework of struts is used to connect all parts of a slice together and keep them in their positions. To generate a strut framework, each disconnected part is abstracted as a vertex, and the vertices are connected with a Delaunay triangulation. Usually there are more edges than necessary in the triangulated graph, so the user may manually select and remove some of the edges. The edges are then thickened in the slice plane to produce the final struts. Since the struts are removed in the final model, they need not be accounted for in the optimization.

8. Dual Material Design

The feasible design space for buoyant models is limited by the density of the material. In this section we propose two ways to load the interior of an object which increases the average density of plywood designs. To make use of standard parts we incorporate prefabricated metal elements, and as an alternative we present a prototyped method for high density material casting.

8.1. Embedded Bolts

In order to control the mass distribution of our models, we investigated methods for embedding high density discrete masses. In contrast to previous work on dual density design with custom parts, we make use of standard elements that are readily available from hardware vendors. We selected steel bolts as our mass units to simplify the fabrication pipeline – they are accessible and easy to assemble.

Fig. 10 shows a model mid-construction. A grid is integrated on each slice for mounting the bolts. The mass of the grid is considered while calculating the model mass and center of gravity.

Bolt Placement. Buoyancy optimization with embedded elements follows the linear programming formulation (4-8). The calculations of $\mathbf{F_G}$ and $\mathbf{C_G}$ incorporate the masses of the bolt elements. Fill variables v_i determine whether or not to place a bolt at a pixel position, and each unit mass m_i corresponds to a bolt mass. The shell's mass and center of mass are determined from the planar slice areas, including contributions from the mounting grid which remains fixed through the optimization.

A difficulty that arises is that the length of a bolt is larger than a single slice of plywood. It is therefore necessary to consider positioning constraints across multiple layers. Let the thickness of each slice be T_{slice}. For a bolt of length d_z, the element occupies $k = \lceil d_z/T_{slice} \rceil$ slices. For a slice A_i, let $E_i = \bigcap_{j=i}^{i+k-1} C_j$. Region E_i gives the area available for loading through k consecutive slices. In our optimization, for each slice A_i, region E_i is discretized into pixels and assigned to fill variables v_i. The pixel resolution is chosen accordingly with the bolt dimensions.

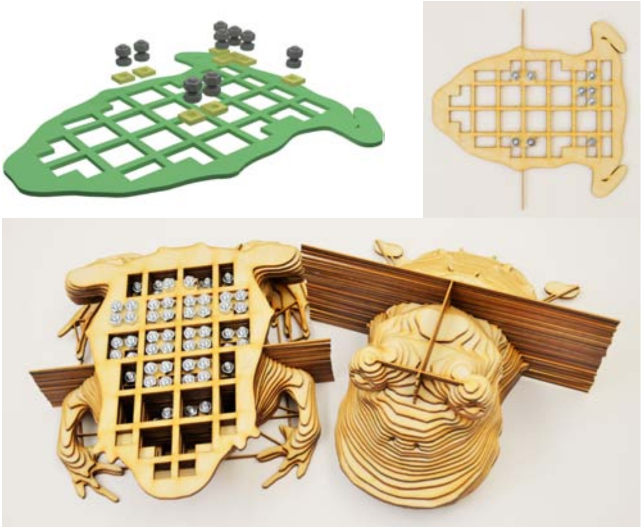


Figure 10: Loading "toad" with embedded bolts: Sample slice and partially assembled toad. The bolts mounted on the grid are visible. The fin-like structure surrounding the toad is strips used to align slices (later removed).

Bolt Collision. It is advantageous to embed elements as densely as possible to increase the precision of interior loading. However, elements placed on nearby slices can overlap in 3D space. To avoid space conflict between elements, a set of linear constraints is added to our optimization formulation (4)-(7):

$$\forall i \; \forall v_{id(x,y,i)} \sum_{j=i}^{i+k-1} v_{id(x,y,j)} \leq 1 \tag{11}$$

Fill variable $v_{id(x,y,j)}$ corresponds to the pixel at position (x,y) on slice j. The constraint applies to $n-k+1 \geq i \geq 1$, and all (x,y) in region E_i. The value of $v_{id(x,y,j)}$ is binary, meaning a bolt element is embedded only when $v_{id(x,y,j)} = 1$. The constraint (11) ensures that at most one bolt element can be placed at position (x,y) from slice i to slice $i+k-1$, and no collisions will occur.

As constraint (11) requires the fill variables to be binary, it renders the optimization a mixed integer linear programming problem. To keep the optimization tractable we use a similar approach to the 3D printed voxels: the fill variables are relaxed to take a fractional value. The optimized value is then rounded to its nearest binary (i.e. fill variables $v \leq 0.5$ are rounded to 0). Rounding does not break constraint (11). Since the optimization result satisfies (11) in the fractional stage, at most one fill variable at a given position (x,y) in consecutive k slices can have a value strictly greater than

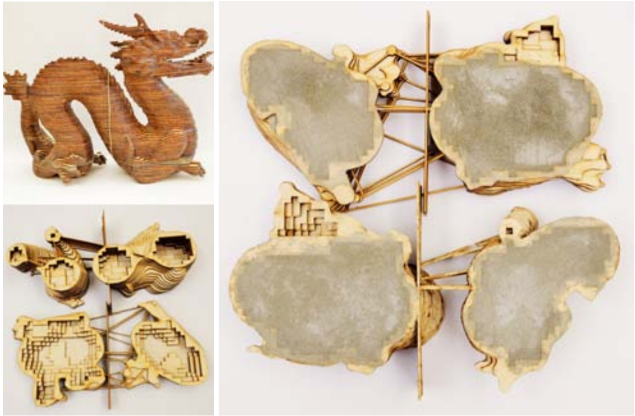


Figure 11: Cement casting of "dragon": (Left-top) Assembled dragon model. (Left-bottom) The cement pocket of the dragon model and is opened in two halves. (Right) Cement filled pocket in two halves.

0.5. As such, bolts are embedded for optimal load distribution considering both their in-plane and multi-slice positioning.

8.2. Load by Casting

An alternative method for loading an object is to cast dense material in a mold which matches the shape of the object shell. Metal casting using a 3D printed mold is seen in the work of [BWBSH14]. However, if the shape of the casting is non-convex, complexities arise in removing the casting from its mold, and assembling the casting inside the object's shell. Though a non-convex shape can be partitioned into multiple convex parts [Cha84], casting these parts separately and assembling them is prone to imprecision and added expense. We propose *integrating* the mold with the geometry of the final design. To fabricate, the casting material is simply poured into the object's shell. No cast-mold separation is required and assembly is greatly simplified. An example is shown for a dragon model in Fig. 11.

Integrated Mold. The casting mold is integrated inside of the object and can be thought of as an extension of the object shell. The mold is designed to be watertight, constructed with the same material as the shell and, similarly to the shell, consists of a boundary with minimal thickness set by fabrication constraints. A challenge is that the mold geometry varies with the casting shape, and has non-negligible mass for buoyancy optimization. The shapes of the casting and mold must therefore be simultaneously optimized. We propose an iterative algorithm.

Mold Creation. Similarly to the process for embedded bolts (Sec. 8.1), the mass properties for each pixel correspond to the solid casting material. A pixel is hollowed if the fill variable $v_i = 0$, otherwise the pixel is solid casting. The computations for gravitational force and center of mass (see Appendix) apply the high density material parameters.

The integrated mold is created in an iterative manner (see Fig. 12). At initialization the optimization solves only for an initial shape of casting (without mold). Each iteration then consists of two steps: 1) A mold is created according to the shape of the casting. The mass of the newly obtained mold modifies $\mathbf{C_G}$ from its desired position. 2) The shape of casting is updated by solving the optimization again, taking into account the mass and positioning of the mold. Note the mold geometry is held fixed. At each iteration these steps are repeated, progressively updating the mold and shape of casting. The error at each iteration is measured by shape mismatch between mold and casting. Equivalently this can be determined by the error in $\mathbf{F_G}$ and $\mathbf{C_G}$ after the mold is updated. The design is considered to be converged when the error reaches a set threshold. We demonstrate our method in the Dragon model (Fig. 11) which can be seen floating as designed in Fig. 18.

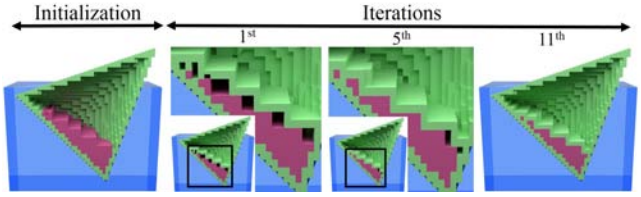


Figure 12: Casting optimization: Initialization: only casting shape (purple) is optimized. Iteration 1: mold is added, fit to the cast shape. Casting is then re-optimized considering mass of the mold. Error measured as gap between mold and casting. Iteration 5: Gap between casting and mold decreases. Iterations terminate when shape mismatch is below a threshold.

9. Results

We constructed a variety of floating objects and verified their behavior by placing them in a tank filled with pure water.

9.1. 3D Printing Results

All models were printed using an Objet Eden250 with a transparent photopolymer. The resolutions of the printer are 600dpi, 300dpi and 1600dpi on x-axis, y-axis and z-axis, respectively. The density of polymerized printing material is 1.18-1.19 g/cm^3. The Objet Eden250 builds models layer-by-layer from bottom-up, so supporting material is necessary for overhanging parts. To allow removal of supporting material from the hollowed space, we printed the models in multiple segments and glued them afterward. The segmentation of the model is done manually using standard software. Existing research guides this cut [LBRM12].

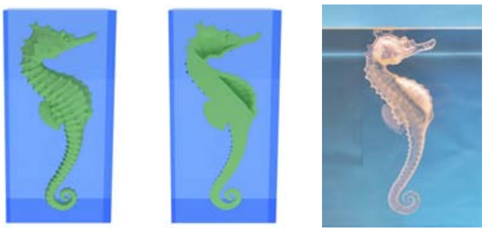


Figure 13: "Seahorse": The fully submerged seahorse model (left) is hollowed to meet design constraints (middle). The fabricated 3D model floats upright, just beneath the fluid surface (right).

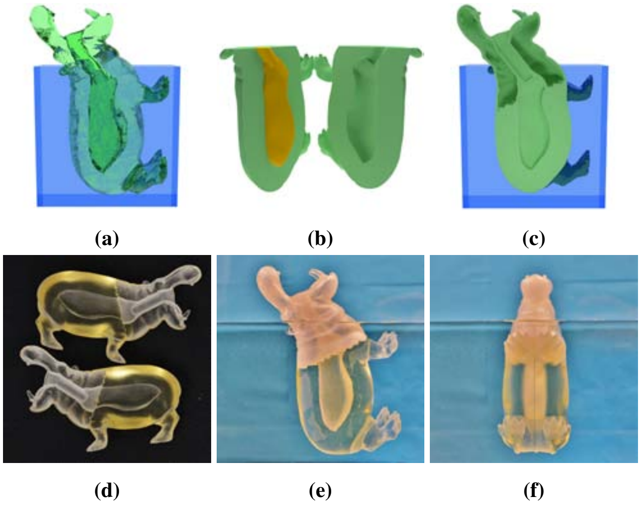


Figure 14: "Hippo cup": The hippo model has a pre-existing void in its body (a). The fluid displacement of the hippo model includes the internal void (b, orange volume). The optimized design for the hippo model (c) is 3D printed in halves (d), demonstrating the pre-existing void and hollowing. The resulting fabricated object floats upright in the fluid (e, f).

Design Verification. Fig. 5 demonstrates a whale design where the waterline area is not directly above the center of buoyancy. In this case, there is a large horizontal distance between the center of roll and the center of buoyancy. This distance produces a long moment arm, leaving the model sensitive to any potential design or fabrication error. However, we attained a result that floats in the correct orientation (Fig. 5e, 5f), confirming the effectiveness or our design workflow.

The Seahorse model (Fig. 13) demonstrates a totally submerged design. Since the waterline area is zero for totally submerged objects, the second moment of area also goes to zero (i.e., $I = 0$). As a result, the metacenter coincides with the center of buoyancy. A stable totally submerged design must have its center of mass directly below its center of buoyancy.

Our algorithm also handles models with concavities such as the Hippo (Fig. 14a). The void is open to air through the mouth of the hippo, similar to the cup result (Fig. 4).

Fig. 15 demonstrates three different designs of the same bunny model, where the floating pose is modified to rest on its tail, back, and upright. The optimization produces an effective hollowed design for each desired orientation.

Limitations. We observed that certain model geometries can be very sensitive to fabrication error. For example, for the whale model (Fig. 5), a difference of 1.73% in assumed material density led to an orientation error of 18 degrees. A future tool could explore the design space around the optimal solution to examine the sensitivity of the result, and ideally provide suggestions on how to make the design more error tolerant.

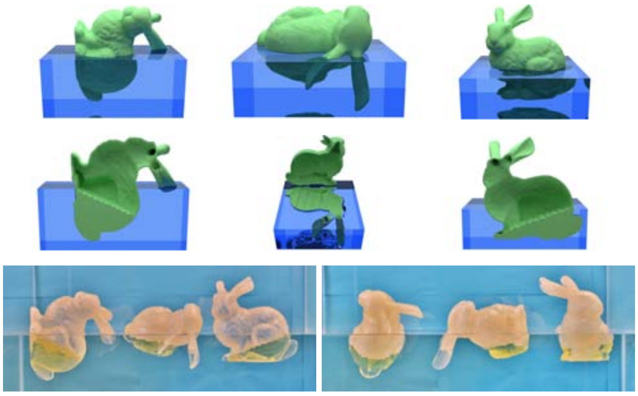


Figure 15: "Bunnies": (Top row) Desired orientations. Each column demonstrates a different design. (Middle row) Interior hollowing. (Bottom row) Verification of the bunny models.

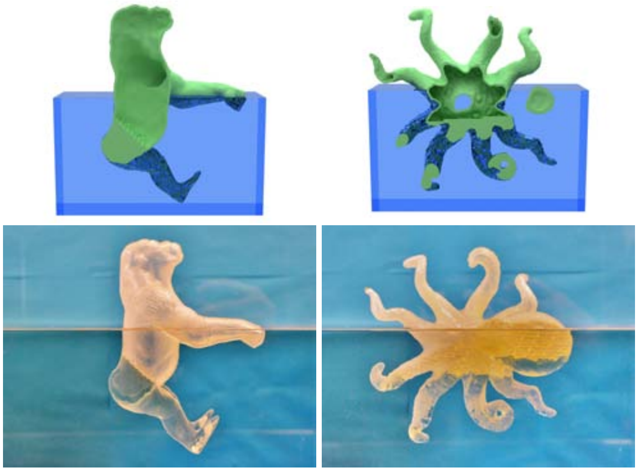


Figure 16: "Gorilla" and "Octopus": (Top) Design result cross sectional view. (Bottom) Verification with 3D printing.

9.2. Planar Slices

All planar slice designs are fabricated with 1/8 inch plywood. The density of the plywood is 0.63 g/cm^3 and the mean square error of density is 0.0089 g/cm^3. We randomized the order of the slices' layout on the plywood to average out the deviation in density. The density of the cement used for dual material casting is 2.37 g/cm^3.

Design Verification. We fabricated several models using plywood pieces to verify our algorithm. The Angry Bird model (Fig. 6) demonstrates a design that is feasible with a single material of low density plywood and no additional loading. The Toad model (Fig. 17) demonstrates a design with embedded bolt elements. The steel bolts add necessary weight so that the toad can sit low in the water. The Dragon model (Fig. 18) demonstrates a design with pockets for casting high density material. We use cement as the casting material. The integrated mold creation algorithm converges after 4 iterations. In each case, the floating poses of our prototypes closely match the input design.

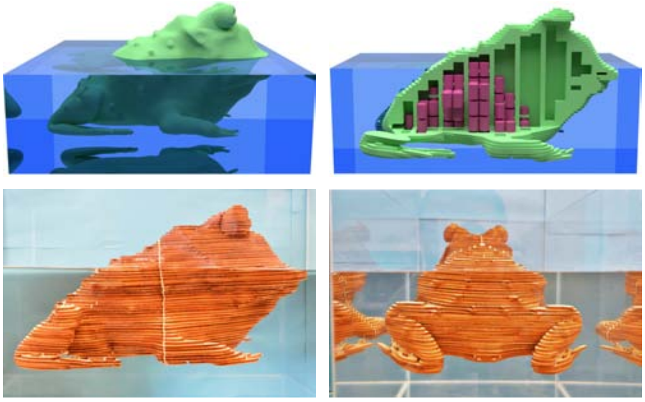


Figure 17: "Toad": (Top-left) Desired floating orientation. (Top-right) Cross sectional view of the optimized toad. Purple cylinders represent embedded bolts. (Bottom) The floating fabricated toad in different views.

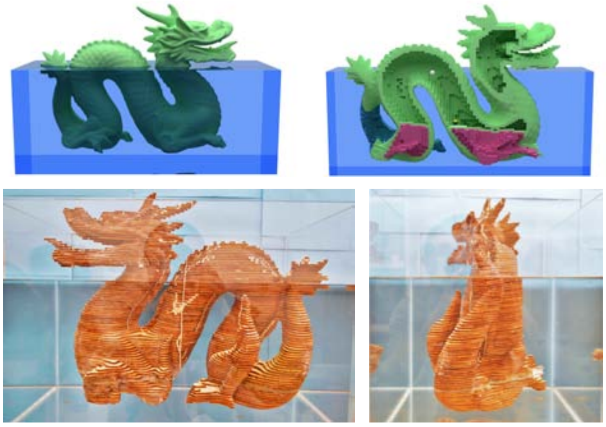


Figure 18: "Dragon": (Top-left) Desired floating orientation. (Top-right) The purple solid volume represents cement fill. (Bottom) The cement is in a pocket. The dragon model matches the input specification.

9.3. Optimization Timing

The runtimes for our optimization are given in the following table. For the Toad and Dragon models the reported times include optimization of the secondary material with embedded bolts (Toad) or integrated mold (Dragon). The volume of each model is given for reference. The optimization software was written using the Python API for Blender and run on a PC with 8GB of main memory and an Intel i5-4300U CPU.

	3D printing			Planar pieces		
Model	gorilla	hippo	octopus	bird	toad	dragon
Time (s)	55	52	97	60	28	219
Vol. (cm^3)	140	157	102	3698	3714	3835

10. Conclusion

We have shown how to create buoyant versions of complex 3D objects that float in a desired position and orientation. Our 3D printed objects have the same external appearance as their original models and are robust and stable when floating. For objects too large for practical 3D printing, we introduce a fabrication method using planar slices. Our planar fabrication method includes an optimization for practical dual density construction using either readily available parts or a custom mold. Because of the dual density construction, we can design floating objects from materials less dense than water, and also conserve these materials by trading off weight between the two densities. We have created several large, plywood models and shown that they achieve their target buoyant positions.

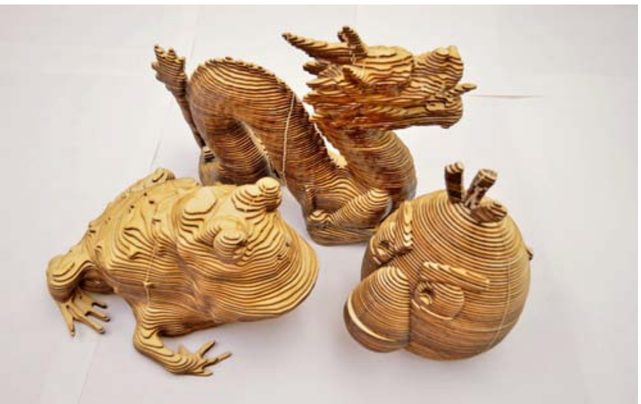


Figure 19: Planar pieces designs

Our work applies to stability under small tilt angles. In future work, we hope to investigate challenges in optimizing for global floating stability. The formulation of floating objects having only one stable equilibrium – where an object restores to a prescribed stable orientation from any initial orientation – is a compelling problem for water safety applications. The related problem of designing an object with exactly two (or more) stable points also deserves attention, and could facilitate the design of unusual watercraft such as the U.S. Office of Naval Research's FLIP [FLI], which converts between a ship and a floating platform.

Acknowledgments

We thank Devin Balkcom, Lorie Loeb, and the anonymous reviewers for helpful comments. Kevin Baron and Pete Fontaine of Thayer School Machine Shop provided fabrication assistance. Models provided courtesy of Stanford Computer Graphics Laboratory: bunny, dragon; Thingiverse: whale, seahorse, gorilla, hippo, octopus, toad; Myminifactory: angry bird. This project is partially supported by the National Science Foundation under Grant No. 1464267.

References

[AW95] ANDERSON J. D., WENDT J.: *Computational fluid dynamics*, vol. 206. Springer, 1995. 2

[BBAM12] BERMANO A., BARAN I., ALEXA M., MATUSK W.: Shadowpix: multiple images from self shadowing. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 593–602. 2

[BMF07] BRIDSON R., MÜLLER-FISCHER M.: Fluid simulation: SIGGRAPH 2007 course notes. In *ACM SIGGRAPH 2007 courses* (2007), ACM, pp. 1–81. 2

[BP13] BIRAN A., PULIDO R. L.: *Ship hydrostatics and stability*. Butterworth-Heinemann, 2013. 2

[BS98] BERTRAM V., SCHNEEKLUTH H.: *Ship design for efficiency and economy*. Butterworth-Heinemann, 1998. 2

[BWBSH14] BÄCHER M., WHITING E., BICKEL B., SORKINE-HORNUNG O.: Spin-it: optimizing moment of inertia for spinnable objects. *ACM Trans. Graph. 33*, 4 (2014), 96. 1, 2, 4, 7

[Cha84] CHAZELLE B.: Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM Journal on Computing 13*, 3 (1984), 488–507. 7

[CPMS14] CIGNONI P., PIETRONI N., MALOMO L., SCOPIGNO R.: Field-aligned mesh joinery. *ACM Trans. Graph. 33*, 1 (2014), 11:1–11:12. 2

[CSB14] CHRISTIANSEN A., SCHMIDT R., BÃĘRENTZEN J.: Automatic balancing of 3D models. *Computer-Aided Design 58*, January 2015 (2014), 236–241. 2

[CTG10] COHEN J. M., TARIQ S., GREEN S.: Interactive fluid-particle simulation using translating eulerian grids. In *Proc. ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (2010), ACM, pp. 15–22. 2

[DWLF12] DE WITT T., LESSIG C., FIUME E.: Fluid simulation using Laplacian eigenfunctions. *ACM Trans. Graph. 31*, 1 (2012), 10. 2

[FLI] FLIP: Floating instrument platform, scripps institution of oceanography. *http://www.mpl.ucsd.edu/resources/flip.intro.html*. 9

[Lau11] LAUTRUP B.: *Physics of continuous matter: exotic and everyday phenomena in the macroscopic world*. CRC Press, 2011. 1, 2

[LBRM12] LUO L., BARAN I., RUSINKIEWICZ S., MATUSIK W.: Chopper: partitioning models into 3D-printable parts. *ACM Trans. Graph. 31*, 6 (2012), 129. 7

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In *ACM SIGGRAPH Computer Graphics* (1987), vol. 21, ACM, pp. 163–169. 4

[MAB*15] MUSIALSKI P., AUZINGER T., BIRSAK M., WIMMER M., KOBBELT L.: Reduced-order shape optimization using offset surfaces. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 102. 2

[MUS14] MCCRAE J., UMETANI N., SINGH K.: Flatfitfab: interactive modeling with planar sections. In *Proc. ACM symposium on User interface software and technology* (2014), ACM, pp. 13–22. 2

[Now10] NOWACKI H.: Five decades of computer-aided ship design. *Computer-Aided Design 42*, 11 (2010), 956–969. 2

[PWLSH13] PRÉVOST R., WHITING E., LEFEBVRE S., SORKINE-HORNUNG O.: Make it stand: balancing shapes for 3D fabrication. *ACM Trans. Graph. 32*, 4 (2013), 81. 1, 2, 4

[RAD12] RIVERS A., ADAMS A., DURAND F.: Sculpting by numbers. 157. 2

[RMD12] RIVERS A., MOYER I. E., DURAND F.: Position-correcting tools for 2D digital fabrication. *ACM Trans. Graph. 31*, 4 (2012), 88:1–88:7. 2

[SP12] SCHWARTZBURG Y., PAULY M.: Design and optimization of orthogonally intersecting planar surfaces. In *Computational Design Modelling*. Springer, 2012, pp. 191–199. 2

[SP13] SCHWARTZBURG Y., PAULY M.: Fabrication-aware design with intersecting planar pieces. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 317–326. 2

[SPG03] SIGG C., PEIKERT R., GROSS M.: Signed distance transform using graphics hardware. In *Proc. IEEE Visualization 2003* (2003), IEEE Computer Society, pp. 83–90. 4, 5

[STK*14] SKOURAS M., THOMASZEWSKI B., KAUFMANN P., GARG A., BICKEL B., GRINSPUN E., GROSS M.: Designing inflatable structures. *ACM Trans. Graph. 33*, 4 (2014), 63. 2

[UKSI14] UMETANI N., KOYAMA Y., SCHMIDT R., IGARASHI T.: Pteromys: interactive design and optimization of free-formed free-flight model airplanes. *ACM Trans. Graph. 33*, 4 (2014), 65. 2

[UMI10] UMETANI N., MITANI J., IGARASHI T.: Designing custom-made metallophone with concurrent eigenanalysis. In *Proc. Conference on New Interfaces for Musical Expression (NIME)* (2010). 2

Appendix

Physical Properties. The gravitational force of an object:

$$\mathbf{F_G} = \left(\sum_i v_i m_i + M_s\right)\mathbf{g} \tag{12}$$

is expressed as the sum of shell mass M_s and mass of the voxels/pixels $\sum_i v_i m_i$. The center of gravity of an object, $\mathbf{C_G}$, is computed by:

$$\mathbf{C_G} = \frac{M_s\mathbf{C_s} + \sum_i v_i m_i \mathbf{c_i}}{M_B} \tag{13}$$

where $\mathbf{C_s}$ is the center of mass of the object shell, and m_i and $\mathbf{c}_i$ are the mass and center of gravity of the voxel v_i.

I of (3) is the smallest second moment of the waterline area and is given by $I = \min_\theta\left\{I'(\theta)\right\}$. Property $I'(\theta)$ is the second moment of the waterline area with tilt axis $\mathbf{n} = (cos\theta, sin\theta)$:

$$I'(\theta) = \cos^2\theta \int_A x^2 d\sigma - 2\sin\theta\cos\theta \int_A xy d\sigma + \sin^2\theta \int_A y^2 d\sigma \tag{14}$$

where A is the waterline area. The origin of the coordinate system is placed at the center of roll.

Watertight conditions. We illustrate how the conditions in Section 7.3 are satisfied when hollowing region C_i on a slice A_i.

Assertion 1: $A_i \setminus C_i$ guarantees the side wall thickness T_{wall}.

$A_i \setminus C_i$ gives the remaining area after cutting C_i from A_i. The side wall of slice A_i is given by $S = A_i \setminus W(A_i)$. It can be seen that $S \subseteq A_i \setminus C_i$, and thus the sidewall is retained when C_i is hollowed.

Assertion 2: $(A_i \setminus C_i) \cap (A_{i+1} \setminus C_{i+1})$ guarantees the overlapping thickness $T_{overlap}$.

$(A_i \setminus C_i) \cap (A_{i+1} \setminus C_{i+1})$ is the overlapping area of slice A_i and slice A_{i+1} after hollowing C_i and C_{i+1}, respectively. Let $U = (A_i \setminus L(A_i)) \cap (A_{i+1} \setminus L(A_{i+1}))$. U contains the overlapping area of two adjacent slices A_i and A_{i+1} that guarantees the overlapping thickness $T_{overlap}$. It can be shown that $U \subseteq (A_i \setminus C_i) \cap (A_{i+1} \setminus C_{i+1})$ and thus the minimal overlapping area will remain after hollowing.

Assertion 3: Absence of holes on overhanging areas.

The overhanging area can be described by symmetric difference between adjacent slices. The symmetric difference between two slices A_i and A_{i+1} is $A_i \oplus A_{i+1}$. This assertion is equivalent to $A_i \oplus A_{i+1} \subseteq (A_i \setminus C_i) \oplus (A_{i+1} \setminus C_{i+1})$. Intuitively, this expression says that the overhanging area between two hollowed slices $(A_i \setminus C_i)$ and $(A_{i+1} \setminus C_{i+1})$ is larger than the overhanging area between two solid slices A_i and A_{i+1} which are not hollowed.

Assertion 4: C_i gives the maximal area which can be hollowed from A_i with respect to Assertions 1, 2 and 3.

It can be show that we cannot find a larger area H_i ($C_i \subset H_i$) for slice A_i such that $A_i \setminus H_i$ still satisfies Assertions 1, 2 and 3.