Vision, Modeling, and Visualization (2016)
M. Hullin, M. Stamminger, and T. Weinkauf (Eds.)

Balancing 3D Models with Movable Masses

Romain Prévost[1,2] Moritz Bächer[2] Wojciech Jarosz[3] Olga Sorkine-Hornung[1]

[1]Department of Computer Science, ETH Zurich [2]Disney Research [3]Dartmouth College

Figure 1: *(From left to right) The **Sheared Cylinder**, the **Muscle Gnome**, the **Dolphin**, the **Breakdancing Teddy**, and the **T Shape** are all models whose complex balance is made possible by using movable masses and optimization with our method.*

Abstract

We present an algorithm to balance 3D printed models using movable embedded masses. As input, the user provides a 3D model together with the desired suspension, standing, and immersion objectives. Our technique then determines the placement and suitable sizing of a set of hollow capsules with embedded metallic spheres, leveraging the resulting multiple centers of mass to simultaneously satisfy the combination of these objectives. To navigate the non-convex design space in a scalable manner, we propose a heuristic that leads to near-optimal solutions when compared to an exhaustive search. Our method enables the design of models with complex and surprising balancing behavior, as we demonstrate with several manufactured examples.

1. Introduction

In recent years, 3D printing has become mainstream, allowing people to create customized objects while requiring little knowledge of the underlying technology or professional 3D modeling packages. This has been made possible by the proliferation of low-cost 3D printers, but also by the research on computational methods that provide easy and intuitive control over the physical behavior of the printed object. One recurrent research problem in this context is the design of balance and optimization of various inertial properties. Most of the previous attempts rely on placing the center of mass at the right location to obtain the desired static behavior [PWLSH13, BWBSH14, MAB*15]. With a single realizable center of mass, however, it is often physically impossible to optimize multiple objectives, making these solutions inherently limited. The simple sheared cylinder in Figure 2 exemplifies this limitation: it is impossible to enforce balance on both bases of the cylinder with a single center of mass since no point within the convex hull projects within both support polygons (convex hulls of the respective contact points).

In this paper, we overcome this limitation by introducing embedded movable masses inside the 3D printed model. This leads to the final object having different centers of mass depending on its pose, which we leverage to simultaneously optimize a combination of balancing objectives. We parameterize our design space using capsule-shaped voids carved out inside the object. Inside each capsule, we place a metallic ball that switches locations from one end to the other depending on the object's orientation relative to the direction of gravity. While this idea is conceptually simple, turning it into a practical system requires carefully accounting for various 3D fabrication constraints, e.g., the capsules may not overlap and must be enclosed inside the object. Since placing these capsules and determining their size is a challenging design task, we devise a non-

DOI: 10.2312/vmv.20161337

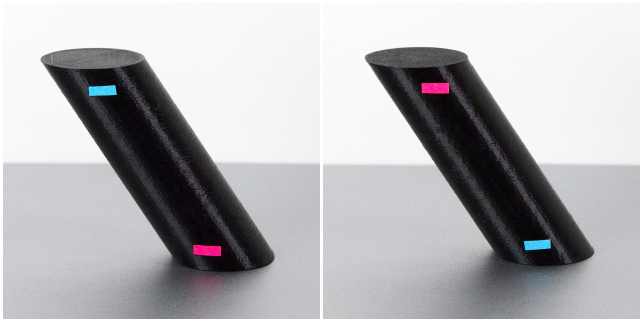


Figure 2: *The **Sheared Cylinder** can only balance on both its ends with an embedded movable mass, because a single center of mass cannot project inside both support polygons.*

linear constrained optimization that assists the user in achieving a combination of balancing objectives simultaneously. Moreover, the number of capsules adds a discrete component to this optimization. To parry this problem, we propose an effective heuristic leading to quasi-optimal solutions while avoiding an exhaustive search of the design space. Our system additionally supports articulated models, which we can balance in several poses. We demonstrate how our framework can be used to produce 3D objects with various combinations of balancing objectives, such as standing under gravity, suspension, as well as immersion in liquid.

2. Related work

Fabrication-oriented design. Modern advanced manufacturing technologies have triggered the development of computational design tools that aid engineers and non-expert users alike with the digital design of physical models. We complement these techniques with a method to optimize rigid and articulated models to stably stand, suspend, or float in several, user-specified poses.

Balancing. The existence of homogeneous, convex models with a particular number of stable and unstable equilibria has interested theorists for decades (see, e.g., [VD06]). Most prominent examples include Schoenhut's classic Rolly-Dolly toy [Sch09], the more recent Weeble® toy that wobbles back and forth, and the Gömböc, all of which exhibit only a single stable equilibrium. Recent efforts focus on optimizing the mass distribution of non-convex, custom shapes to stably stand [PWLSH13], spin [BWBSH14], right themselves [MAB*15, ZHL*16], or float in full [MHR*16] or partial [WW16] immersion. We propose an alternative formulation for static balancing objectives which leads to stability improvements, and further address the challenge of combining several competing objectives simultaneously.

To optimally distribute mass, Prévost et al. [PWLSH13] propose voxel carving and handle-based deformation, and Bächer et al. [BWBSH14] use an adaptive voxel discretization and cage-based deformation. Musialski and colleagues introduce a reduced-order approach on offset surfaces [MAB*15] and a projection on local subspaces [MHR*16] for improved efficiency. Like Christiansen and colleagues [CSB15], we use sparse patterns and infill to aid balancing. Unlike all previous efforts, however, we support pose-dependent, static balancing with several centers of mass instead of a single one, enabled by embedded, movable masses.

Modeling objects with movable masses. We are only aware of a few works on the design of objects with embedded movable masses. Akulenko et al. [ABKN06] describe the motion control of a cylinder, and Bolotnik and Figurina [BF08] discuss the optimal control of the rectilinear motion of a rigid body using movable masses. While their techniques focus on the more challenging problem of motion control, their treatments are tailored to models of particular shape or restricted number of masses and do not directly generalize to arbitrary shapes or articulated input.

Articulated models. Bächer et al. [BBJP12] describe techniques to convert a skinned character to an articulated toy model, while Cali and colleagues [CCA*12] design such jointed models from a static input using a rigging interface. The output of these techniques serves as input to our balancing approach.

3. Overview

Our goal is to allow novice users to create 3D printable objects that meet a complex set of balancing objectives. To this end, we introduce the idea of placing movable masses inside a 3D shape to allow for multiple centers of mass depending on the object's pose. We devise a practical parameterization of the design space and provide an optimization method to navigate this space.

Our method takes as input a (potentially articulated) 3D model and a set of balancing objectives. Our optimization automatically determines the number, size, and placement of capsules inside the shape to minimize the corresponding balancing energy. Each capsule encloses a metallic ball whose location depends on the pose of the model. Our system outputs an object that is ready for fabrication using 3D printing and off-the-shelf steel balls to be inserted during the printing process (see inset, right).

4. Balancing with movable masses

With embedded movable masses, we have the unique capability to satisfy multiple balancing objectives simultaneously. Depending on the pose of the model, the masses will be at different locations, resulting in different centers of mass. With a single center of mass, most objectives would be incompatible. Therefore, we introduce an optimization able to handle a multitude of objectives.

4.1. Balancing objectives

Standing under gravity. While our tool interfaces with previously introduced static balancing objectives [PWLSH13, BWBSH14], we propose a variation that leads to improved stability of the resulting equilibria.

Given the current center of mass **c**, Prévost et al. [PWLSH13] propose to first perpendicularly project it onto the plane containing

the support polygon, then minimize the distance between the projected center and the boundary of the support polygon, shrunk by a safety margin. In contrast, Bächer and colleagues [BWBSH14] force the center of mass to project to the "center" of the support polygon, penalizing its height for increased stability.

While sufficient for many cases, these objectives may lead to suboptimal solutions, especially if we aim to find the best trade-off between several competing balancing objectives. Given the support polygon of a given pose, we propose to minimize the maximal "toppling" angle θ_i (see Figure 3 left for an illustration) where i denotes the edge index of the convex polygon. The "toppling" angle corresponds to the signed angle between the negative gravity direction $-\mathbf{g}$, and the plane containing $\mathbf{c}$ and the edge i, with negative values indicating fulfillment of the objective. By favoring smaller angles, we minimize both the abscissa and ordinate of the projection of $\mathbf{c}$ onto the $\mathbf{a}$ and $-\mathbf{g}$ axes, indicated in red and green in Figure 3, respectively ($\mathbf{a}$ is the normalized cross product between $\mathbf{g}$ and the edge segment). Hence, we combine the advantages of the two previously proposed alternatives in a single objective.

Note, however, that this objective is continuous but only C^0. Hence, it is ill-suited for descent-based optimization. To overcome this limitation, we use a common smooth maximum function instead:

$$f_{\text{bal}}(\mathbf{c}) = \frac{\sum_i \theta_i e^{\beta\theta_i}}{\sum_i e^{\beta\theta_i}} \tag{1}$$

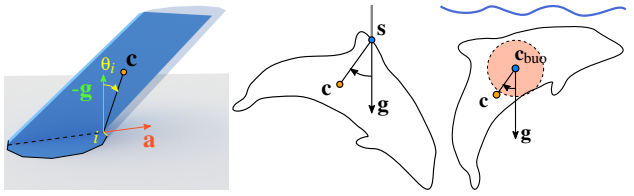
with a large positive β (in practice we choose $\beta = 100$).

Interestingly, we can turn a balance into an imbalance objective by simply flipping the sign of f_{bal}, allowing users to imbalance previously balanced configurations and vice versa, thereby providing them with a previously unseen level of control over static balancing.

Suspension by a string. For a model suspended by a string from a ceiling (see Figure 3 middle), the model remains in the desired pose only if the center of mass $\mathbf{c}$ lies on the line through the attachment point $\mathbf{s}$ in the user-specified direction of gravity $\mathbf{g}$. Hence, it is natural to minimize the angle between $\mathbf{c}-\mathbf{s}$ and $\mathbf{g}$:

$$f_{\text{sus}}(\mathbf{c}) = \arccos((\mathbf{c}-\mathbf{s})\cdot\mathbf{g}). \tag{2}$$

The suspension point $\mathbf{s}$ is given as input by the user.



Standing Suspension Immersion

Figure 3: *Balancing objectives (from left to right).* ***Standing under gravity****: the toppling angle θ_i (in yellow) for an edge i is the signed angle between the negative gravity $-\mathbf{g}$ and the plane containing the edge and the center of mass $\mathbf{c}$.* ***Suspension by a string****: the model is optimized when the vector between the center of mass and the suspension point aligns with the gravity. The corresponding angle serves as our objective.* ***Immersion in a fluid****: the metric is again the angle between the center of mass and the gravity. Additionally, the mass constraint (Equation 3) and the stability constraint (Equation 5) represented as a light red disk must also be satisfied.*

Immersion in a fluid. A fully immersed object remains stable, if the net force and torque acting on the model evaluate to zero. However, the pressure difference in the liquid leads to an upward pointing force acting on the center of mass $\mathbf{c}_{\text{buo}}$ of the displaced fluid—also known as the center of buoyancy—with magnitude equal to its weight. This force counteracts the gravitational force $\mathbf{f}_G$, acting on the center of mass $\mathbf{c}$ of the model with magnitude equal to its weight. Setting the net force to zero leads to a *mass constraint*

$$m - \rho_F V_{\mathcal{M}} = 0, \tag{3}$$

where ρ_F denotes the fluid's density and $V_{\mathcal{M}}$ the model's volume.

To form the rotational equilibrium condition, it is convenient to choose $\mathbf{c}_{\text{buo}}$ as the reference point, leading to a net torque $(\mathbf{c}-\mathbf{c}_{\text{buo}})\times\mathbf{f}_G$. This net torque evaluates to zero whenever the vectors $\mathbf{c}-\mathbf{c}_{\text{buo}}$ and $\mathbf{f}_G$ point in the same or opposite direction, motivating our balancing objective

$$f_{\text{buo}}(\mathbf{c}) = \arccos((\mathbf{c}-\mathbf{c}_{\text{buo}})\cdot\mathbf{g}). \tag{4}$$

Note that the translational equilibrium is unstable, and—because it is impossible to manufacture a model at the precision necessary to fulfill the mass constraint exactly—the fabricated model rises or sinks independently of how carefully we place it in the surrounding fluid. The rotational equilibrium is, however, stable as long as $f_{\text{buo}}(\mathbf{c})$ is smaller than $\frac{\pi}{2}$ and the distance between the two centers $\mathbf{c}_{\text{buo}}$ and $\mathbf{c}$ is non-zero. Our objective $f_{\text{buo}}(\mathbf{c})$ already penalizes violations of the first rotational stability criterion, and we introduce an additional inequality constraint

$$\|\mathbf{c}-\mathbf{c}_{\text{buo}}\|^2 \geq d_{\min}^2 \tag{5}$$

to enforce a minimal distance $d_{\min}$ between $\mathbf{c}_{\text{buo}}$ and $\mathbf{c}$, referred to as the *buoyancy stability constraint*.

If the objective f_{buo} evaluates to zero, the desired pose is rotationally stable. Therefore, when thrown into a container with the targeted fluid, the fabricated model rectifies its orientation to match this pose, but as explained previously may rise or sink [MAB*15]. An interesting extension is floating [WW16] with stable translational equilibria. Our objective and constraints could easily be extended to enable such floating/partial immersion optimization.

Combining objectives. In contrast to previous balancing approaches, our method aids with the design of models with a *multitude* of centers of mass, each calibrated to meet a single or several balance or imbalance objectives. With our technique, we can further optimize the mass distribution to, e.g., support stable standing in one and suspension in another pose of *a single model*.

Given an input model, a user specifies a combination of standing, suspension, and immersion objectives f_i by selecting for each a corresponding pose T. This transformation $\mathrm{T}(\mathbf{x}) = \mathbf{R}\mathbf{x}+\mathbf{t}$ defines the desired orientation of the model w.r.t. the world coordinate system. To jointly optimize objectives f_i, we could minimize their sum $\sum_i f_i$. By doing so, however, we would favor an on-average,

overall smaller objective over further reducing the objective f_i with the highest value, i.e., the most visible deviation from the user's intent. We therefore minimize the maximum f_i with the same smooth maximum function we used for f_{bal}:

$$f = \frac{\sum_i f_i e^{\alpha f_i}}{\sum_i e^{\alpha f_i}} \tag{6}$$

with a large positive α (in practice we use $\alpha = 100$). If several immersion objectives are present, only one mass constraint is necessary, while we keep a buoyancy stability constraint for each.

4.2. Adding movable masses

To support several centers of mass, we propose a two-step manufacturing process where we 3D print models and add metallic balls to internal channels, which we call capsules. Because capsules cannot be seen from the outside, their convexity is pivotal so that the balls are correctly routed towards the capsules' ends when the model is brought into target poses. Hence, we propose to use linear capsules for the routing of movable masses.

Mass-related quantities. To discuss the effect of capsules on a model's mass properties, we first give formulas for these properties.

The center of mass $\mathbf{c}_k$ and mass m_k of an entity k with constant density ρ_k can be expressed with volume integrals over the enclosed domain Ω_k:

$$\mathbf{c}_k = \frac{\mathbf{b}_k}{V_k} \quad \text{and} \quad m_k = \rho_k V_k \tag{7}$$

where

$$\mathbf{b}_k = \int_{\Omega_k} [x,y,z]^\top dV \quad \text{and} \quad V_k = \int_{\Omega_k} 1\, dV \tag{8}$$

with V_k referring to the volume of k.

Note that we can transform these volume integrals to surface integrals using the divergence theorem, resulting in closed form expressions for simple volumes such as spheres and capsules and analytical expressions for triangle meshes. For the reader's convenience, we include relevant mass-related quantities in Appendix A.

Given the masses m_k and centers of mass $\mathbf{c}_k$ of a set of individual components k, the center of mass $\mathbf{c}$ of their combination is

$$m\mathbf{c} = \sum_k \mathrm{s}(k)\, m_k\, \mathbf{c}_k \quad \text{with} \quad m = \sum_k \mathrm{s}(k)\, m_k, \tag{9}$$

where the sign function $\mathrm{s}(k)$ returns 1 if we add component k and -1 if we subtract it.

Application to our problem. We use two materials with different densities for fabrication: a printer material with density ρ_P and metallic balls with dominant mass per unit volume $\rho_B \gg \rho_P$.

To balance our input model $\mathcal{M}$, we introduce capsules $\mathcal{C}_j$ of radii $r_{\mathcal{C}_j}$ and with cylinder start and end positions $\mathbf{p}_{\mathcal{C}_j}$ and $\mathbf{q}_{\mathcal{C}_j}$, each containing a ball $\mathcal{B}_j$. Using the definitions above, the mass properties of this system are:

$$m\mathbf{c} = \rho_P \left(V_{\mathcal{M}}\, \mathrm{T}(\mathbf{c}_{\mathcal{M}}) - \sum_j V_{\mathcal{C}_j}\, \mathrm{T}(\mathbf{c}_{\mathcal{C}_j}) \right) + \rho_B \sum_j V_{\mathcal{B}_j}\, \mathrm{T}(\mathbf{c}_{\mathcal{B}_j}) \tag{10}$$

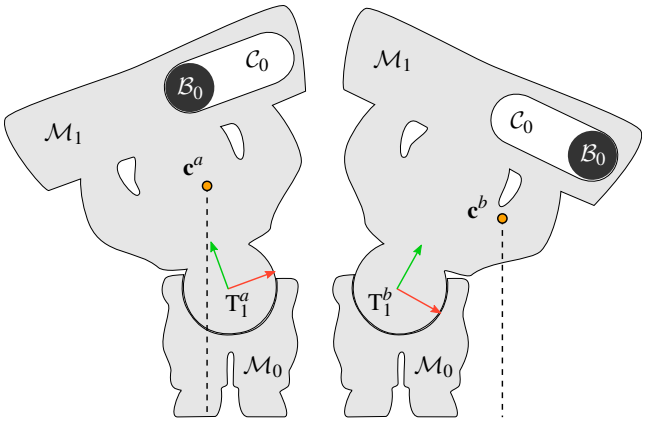


Figure 4: ***Optimizing articulated models.*** *We show two poses T_i^a (left) and T_i^b (right) of our* ***Muscle Gnome****, consisting of two components $\mathcal{M}_0$ and $\mathcal{M}_1$. A single capsule $\mathcal{C}_0$ is embedded in $\mathcal{M}_1$, i.e., $\sigma(0) = 1$. Dependent on the pose, the ball $\mathcal{B}_0$ is at either end of the capsule. The model has two different centers of mass, $\mathbf{c}^a$ and $\mathbf{c}^b$, optimized to balance (left) or imbalance (right) in respective poses.*

with mass

$$m = \rho_P \left(V_{\mathcal{M}} - \sum_j V_{\mathcal{C}_j} \right) + \rho_B \sum_j V_{\mathcal{B}_j}. \tag{11}$$

Depending on the pose, the ball centers sit at either end of the capsule (compare left and right configurations in Figure 4). At the transition between these configurations, there is a discontinuity that leads to non-smooth objectives, violating a pivotal requirement for descent-based optimization. To overcome this limitation, we smooth this discontinuous step, as described in more detail in Appendix A.

If the targeted printer has an infill option, we include it as a parameter in our optimization, as it significantly enlarges our design space. To this end, we multiply the mass properties belonging to the model $\mathcal{M}$ with a scale factor. Alternatively, we could pair our capsule optimization with either voxel-carving [PWLSH13, BWBSH14] or surface offsetting [MAB*15].

4.3. Articulated models

For articulated models, the model itself is split into several components $\mathcal{M}_i$, each positioned with an objective-dependent pose T_i, again given as a rigid transformation. Note, however, that in this case the translational parts of the transformations matter when positioning components with respect to one another. Equations 10 and 11 therefore become

$$\begin{aligned} m\mathbf{c} = \rho_P &\left(\sum_i V_{\mathcal{M}_i}\, \mathrm{T}_i(\mathbf{c}_{\mathcal{M}_i}) - \sum_j V_{\mathcal{C}_j}\, \mathrm{T}_{\sigma(j)}(\mathbf{c}_{\mathcal{C}_j}) \right) \\ &+ \rho_B \sum_j V_{\mathcal{B}_j}\, \mathrm{T}_{\sigma(j)}(\mathbf{c}_{\mathcal{B}_j}) \end{aligned} \tag{12}$$

with mass

$$m = \rho_P \left(\sum_i V_{\mathcal{M}_i} - \sum_j V_{\mathcal{C}_j} \right) + \rho_B \sum_j V_{\mathcal{B}_j}, \tag{13}$$

where $\sigma(j)$ maps each capsule j to its assigned component. We keep this assignment fixed throughout the optimization. See Figure 4 for an explanatory illustration.

4.4. Ensuring fabricability

To ensure fabricability, we prevent overlaps between pairs of capsules, as well as capsules and model components using constraints. At first glance, it seems most reasonable to control the distance between closest points. This metric, however, has a discontinuous first derivative. To overcome this limitation, we quantify the distance between any two point pairs $\mathbf{x}_{\mathcal{C}_i}$ and $\mathbf{x}_{\mathcal{C}_j}$ on the capsules' cylinder axes and constrain them to be greater than the sum of radii:

$$\|\mathbf{x}_{\mathcal{C}_i} - \mathbf{x}_{\mathcal{C}_j}\| \geq r_{\mathcal{C}_i} + r_{\mathcal{C}_j}. \tag{14}$$

To prevent intersections with component boundaries, we follow a similar approach, adjusting our distance metric:

$$d_{\mathcal{M}_{\sigma(j)}}(\mathbf{x}_{\mathcal{C}_j}) \geq r_{\mathcal{C}_j}, \tag{15}$$

where $d_{\mathcal{M}_i}$ is the closest distance to the component $\mathcal{M}_i$. For fast evaluations of distance queries, we pre-compute discrete, per-component distance fields, sampled over uniform grids, then tricubically interpolate the distances of the 64 grid points around a given query point. The gradient of this distance field can be computed analytically and has the advantage of being smoother than a trilinear interpolation.

In practice, we use 32 samples per capsule to evaluate these constraints, with two samples placed at their end points. Because the number of constraints does not scale well with the number of capsules, we reduce the number of samples if we optimize with a larger number of capsules. Alternatively, we could add one-sided penalties to the objective with a cutoff, enabling early culling of distant sample pairs.

For fabrication, we rely on fused deposition modeling (FDM). Many FDM technologies support the fabrication of curvature-bound overhangs without the need for a supporting structure. This enables fabrication of our models with embedded metallic balls without the need for introducing cuts nor relying on a manual assembly after fabrication. We can simply pause the printer, place a ball, and then resume printing. To enable this placement during printing, we need to guarantee that the two end points are at least a radius length $r_{\mathcal{C}_j}$ apart, leading to additional constraints

$$\|\mathbf{q}_{\mathcal{C}_j} - \mathbf{p}_{\mathcal{C}_j}\|^2 \geq r_{\mathcal{C}_j}^2. \tag{16}$$

For technologies other than FDM, manual cutting and assembly is unavoidable, rendering the above constraints unnecessary.

4.5. Placing capsules

Placing capsules is challenging for several reasons: the design space delimited by the boundary of the model $\mathcal{M}$ is non-convex for all but the most simple shapes. Hence, solutions depend on the initial capsule placement. Secondly, the optimal number of capsules is unclear. To avoid an exhaustive search on capsule placements and a treatment of the number of capsules as a discrete parameter, we propose the heuristic detailed below. It is based on two main observations: if we initially place capsules onto an approximate medial axis, solutions get reasonably close to an exhaustive search in a convex neighborhood around the starting point. And to identify the optimal number of capsules, it is reasonable to start with a capsule per member of a convex partitioning of $\mathcal{M}$, then remove capsules that collapse to a small radius after an optimization step, indicating that they are not needed.

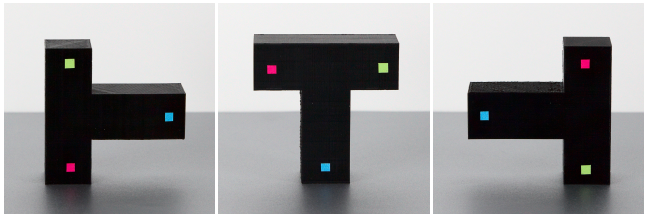


Figure 5: *Without optimization, our **T Shape** only stands stably in the natural orientation (middle). With two embedded capsules, the model also stands stably on its two sides (left, right).*

Loosely following these observations, we summarize our heuristic that works well in practice:

(a) We compute a skeleton of our input model $\mathcal{M}$ using mean curvature skeletons [TAOZ12].
(b) We then break the skeleton into segments at all branch points and identify the segments' midpoints. These midpoints serve as initial positions of capsules. If a more conservative approach is desired, one could further split the skeleton at high curvature points.
(c) We minimize our balancing objectives f with these initial set of capsules, then mark for removal the capsules that did not grow from an initially small, user-specified radius. We remove the marked capsules, but only at most 50 percent of the remaining capsules. And even if no capsule was marked for removal, we remove at least one. We then rerun the optimization with reset capsule parameters and continue this process until no capsules remain.
(d) We finally present all solutions that fulfill the objectives. While a solution with minimal objective value is preferable for stability, a smaller number of capsules is easier to manufacture. This trade-off depends on external criteria, and we therefore let the user choose the preferred solution.

In Section 5, we compare our heuristic to results of an exhaustive random sampling of the entire design space.

In summary, we minimize our objective f with a fixed number of capsules in each step. Besides an optional infill percentage, our unknowns include the capsules' positions $\mathbf{p}_{\mathcal{C}_j}$ and $\mathbf{q}_{\mathcal{C}_j}$, and their radii $r_{\mathcal{C}_j}$. The resulting non-linear programming problem is constrained by buoyancy constraints (Equations 3 and 5) and fabricability constraints (Equations 14, 15, 16, and 22).

5. Results

We have used our technique to optimize a multitude of objects with a variety of balancing objectives, all shown in Figure 1 and detailed in this section. For fabrication, we rely on an Ultimaker 2.

In Figure 2 we show a toy example to emphasize the importance of movable masses. The two extruded support polygons of our

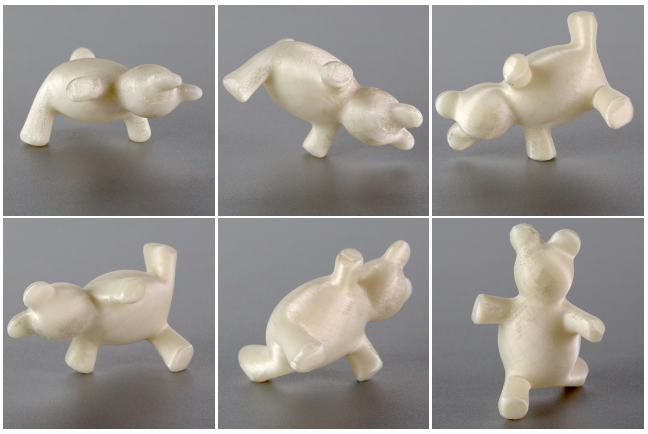


Figure 6: *Using only two capsules we can optimize the **Breakdancing Teddy** to stand steadily in six surprising orientations.*

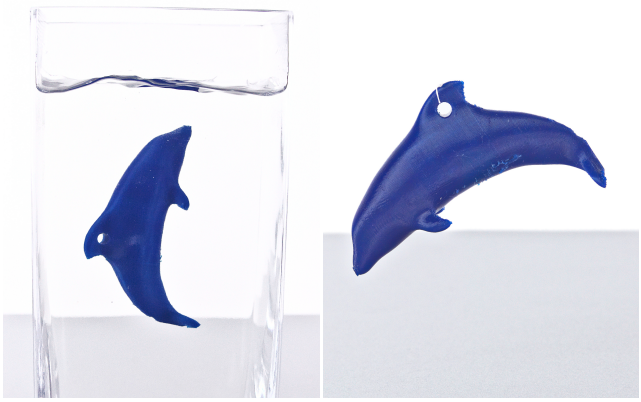


Figure 7: *The **Dolphin** combines a full immersion objective (left) with a suspension (right).*

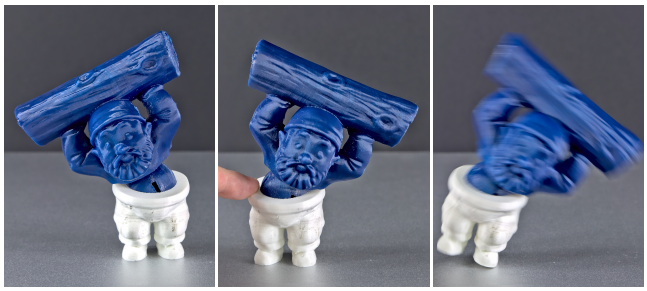


Figure 8: *The **Muscle Gnome** is an example of an articulated model. Though the two poses look symmetric, the model stands stable when brought into the left-tiled pose but tipples over in the right-tilted pose.*

Sheared Cylinder do not intersect. Hence, it is physically impossible to balance this model on its two ends with a single center of mass.

Figure 5 shows our ***T Shape***, which by default stands in the natural orientation ⊤ and its opposite ⊥ but not on the sides ⊢ and ⊣. Note that the orientation ⊥ is trivially balanced because the orthographic projection of any point in $\mathcal{M}$ onto the respective support polygon plane is contained in this convex polygon. Hence, any attempt of imbalancing this configuration is destined to fail. By adding a single capsule at the junction, we can balance the model in all four orientations. And by adding a second capsule, we can further increase stability: we can tilt the initially horizontal ground plane by at least 13 degrees (compared to 8 degrees for a single capsule) before the model topples over in any of the three non-trivial orientations.

Our method is able to umpire the competition between numerous objectives, as we illustrate in Figure 6 with our ***Breakdancing Teddy***. This character stands stably in a total of six poses. In Table 1 we summarize statistics and compare our heuristic to the results of an exhaustive search. From these numbers we can deduce three main points: firstly, already a single capsule leads to significant improvements in stability compared to only optimizing the infill. The latter leads to a tolerance of less than a degree for a tilted ground plane – a value clearly insufficient when taking manufacturing imprecisions into account. Secondly, our heuristic provides near-optimal solutions within seconds. For the exhaustive search we use 360 random sets for each considered number of capsules. Our solutions get close to the optimal toppling angles found by the exhaustive search, implying that our heuristic provides a scalable alternative with a high success rate.

Table 1: *The skeleton of the **Breakdancing Teddy** has 9 branches. Our heuristic optimizes and removes capsules until none are left. The optimized results for 9 and 4 capsules are discarded because there are more stable solutions (i.e., smaller f) with fewer capsules. For the exhaustive search, we report energies and toppling angle for the best set out of 360 random samples. The last column lists the ranking of the solution found by our heuristic with respect to these 360 sets (1 being the best and 360 the worst solution).*

	Our heuristic			Exhaustive search		
# capsules	Opt. time (sec)	Energy f	Max top. angle	Energy f	Max top. angle	Ranking
~~9~~	10	−8.33e−3	−4.22	−9.21e−3	−4.62	324/360
7	3.8	−8.96e−3	−4.50	−9.22e−3	−4.61	111/360
5	24	−8.85e−3	−4.50	−9.17e−3	−4.57	100/360
~~4~~	7.7	−8.55e−3	−4.34	−9.07e−3	−4.55	136/360
3	3.9	−8.70e−3	−4.46	−9.07e−3	−4.50	57/360
2	0.7	−8.28e−3	−4.21	−8.98e−3	−4.44	61/360
1	0.05	−8.00e−3	−4.03	−8.26e−3	−4.11	86/360
0	0.02	−2.14e−3	−0.79			

With our ***Dolphin*** (see Figure 7), we showcase how different types of objectives can be jointly minimized by our technique. This model suspends and immerses in desired orientations.

In contrast to previous techniques, our method can balance articulated models in various poses, as we illustrate in Figure 8 with a set of two poses for our ***Muscle Gnome***. Due to symmetry, one expects the same behavior when bringing the character into sideways tilting poses. We balance one pose and imbalance the other (see also Figure 3), leading to a surprising behavior.

Implementation. To precompute grid point to mesh distances, we use Jacobson et al.'s winding numbers [JKSH13] available in LIBIGL [JP*16]. For minimizing our non-linear objectives, we rely on

the KNITRO optimization library [BNW06] with analytical derivatives. Overall, precomputations take less than a minute for 3D models with up to 180k triangles. This includes the skeleton extraction [TAOZ12, The16].

Limitations. We have demonstrated the viability of our approach with numerous examples. However, our method is not without limitations. Navigating our design space is an utterly challenging task because it exhibits a mix of discrete and continuous components, paired with non-convexity. While our heuristic approach scales and results in solutions close to the ones found by exhaustive search, a more thorough and principled evaluation remains. Finding globally optimal solutions within a reasonable amount of time is, however, close to impossible due to the non-linearity of our objectives and the non-convexity of the underlying design space. Moreover, if the objectives are too difficult – either individually or collectively – with respect to the geometric constraints, a feasible solution might not exist.

6. Conclusion and future directions

We presented a computational technique for the design of models capable of balancing in several user-defined poses. In contrast to previous techniques, we proposed to embed calibrated capsules with metallic balls to support a multitude of centers of mass. Our method enables the optimization of a mixture of standing, suspension, and immersion objectives for one and the same model. Furthermore, we support balancing of articulated models.

While we use capsules and metallic balls for balancing, we would like to investigate the use of granular materials such as sand or fluids to meet our balancing objectives. This would, however, shift the focus towards simulation-based design. As another interesting direction, we would like to extend our method to design dynamically balanced models. Lastly, a design system for the full spacetime control [WK88] of physical models through optimal conversion between potential and kinetic energy is an interesting but ambitious extension of our work.

Acknowledgements. The authors would like to thank Jan Wezel for his help with the 3D printers and his valuable advice on manufacturing matters, Maurizio Nitti for model design, and our colleagues from Disney Research, IGL, and CGL for insightful discussions and support. This work was supported in part by the ERC grant iModel (StG-2012-306877).

References

[ABKN06] AKULENKO L., BOLOTNIK N., KUMAKSHEV S., NESTEROV S.: Control of motion of an inhomogeneous cylinder with internal movable masses along a horizontal plane. *Journal of Applied Mathematics and Mechanics 70*, 6 (2006), 843 – 858. 2

[BBJP12] BÄCHER M., BICKEL B., JAMES D. L., PFISTER H.: Fabricating articulated characters from skinned meshes. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4 (July 2012), 47:1–47:9. 2

[BF08] BOLOTNIK N., FIGURINA T.: Optimal control of the rectilinear motion of a rigid body on a rough plane by means of the motion of two internal masses. *Journal of Applied Mathematics and Mechanics 72*, 2 (2008), 126 – 135. 2

[BNW06] BYRD R. H., NOCEDAL J., WALTZ R. A.: KNITRO: An integrated package for nonlinear optimization. In *Large Scale Nonlinear Optimization* (2006), Springer Verlag, pp. 35–59. 7

[BWBSH14] BÄCHER M., WHITING E., BICKEL B., SORKINE-HORNUNG O.: Spin-it: Optimizing moment of inertia for spinnable objects. *ACM Trans. Graph. (Proc. SIGGRAPH) 33*, 4 (July 2014), 96:1–96:10. 1, 2, 3, 4, 7

[CCA*12] CALÌ J., CALIAN D. A., AMATI C., KLEINBERGER R., STEED A., KAUTZ J., WEYRICH T.: 3D-printing of non-assembly, articulated models. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 31*, 6 (Nov. 2012), 130:1–130:8. 2

[CSB15] CHRISTIANSEN A. N., SCHMIDT R., BÆRENTZEN J. A.: Automatic balancing of 3D models. *Computer-Aided Design 58* (2015), 236 – 241. Solid and Physical Modeling 2014. 2

[JKSH13] JACOBSON A., KAVAN L., SORKINE-HORNUNG O.: Robust inside-outside segmentation using generalized winding numbers. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4 (2013), 33:1–33:12. 6

[JP*16] JACOBSON A., PANOZZO D., ET AL.: libigl: A simple C++ geometry processing library, 2016. http://libigl.github.io/libigl/. 6

[MAB*15] MUSIALSKI P., AUZINGER T., BIRSAK M., WIMMER M., KOBBELT L.: Reduced-order shape optimization using offset surfaces. *ACM Trans. Graph. (Proc. SIGGRAPH) 34*, 4 (Aug. 2015). 1, 2, 3, 4

[MHR*16] MUSIALSKI P., HAFNER C., RIST F., BIRSAK M., WIMMER M., KOBBELT L.: Non-linear shape optimization using local subspace projections. *ACM Trans. Graph. (Proc. SIGGRAPH)* (2016). 2

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

[Sch09] SCHOENHUT A.: Design for a toy figure, Jan. 1909. US Patent D39,775. 2

[TAOZ12] TAGLIASACCHI A., ALHASHIM I., OLSON M., ZHANG H.: Mean curvature skeletons. *Comp. Graph. Forum 31*, 5 (2012), 1735–1744. 5, 7

[The16] THE CGAL PROJECT: *CGAL User and Reference Manual*. CGAL Editorial Board, 2016. http://www.cgal.org/. 7

[VD06] VARKONYI P., DOMOKOS G.: Static equilibria of rigid bodies: Dice, pebbles, and the Poincare-Hopf theorem. *Journal of Nonlinear Science 16*, 3 (2006), 255–281. 2

[WK88] WITKIN A., KASS M.: Spacetime constraints. *Computer Graphics (Proc. SIGGRAPH) 22*, 4 (June 1988), 159–168. 7

[WW16] WANG L., WHITING E.: Buoyancy optimization for computational fabrication. *Comp. Graph. Forum (Proc. Eurographics) 35*, 2 (2016), 49–58. 2, 3

[ZHL*16] ZHAO H., HONG C., LIN J., JIN X., XU W.: Make it swing: Fabricating personalized roly-poly toys. *Computer Aided Geometric Design 43* (2016). Geometric Modeling and Processing 2016. 2

Appendix A: Mass-related quantities

For computations of mass properties for triangulated solids we refer the reader to the supplemental material by Bächer and colleagues [BWBSH14].

The volume and center of mass of homogeneous capsules $\mathcal{C}_j$ and balls $\mathcal{B}_j$ can be readily computed from their parameters $r_{\mathcal{C}_j}$, $\mathbf{p}_{\mathcal{C}_j}$, and $\mathbf{q}_{\mathcal{C}_j}$:

$$V_{\mathcal{C}_j} = \frac{4}{3}\pi r_{\mathcal{C}_j}^2 + \pi r_{\mathcal{C}_j}^2 \|\mathbf{q}_{\mathcal{C}_j} - \mathbf{p}_{\mathcal{C}_j}\| \quad \text{and} \quad V_{\mathcal{B}_j} = \frac{4}{3}\pi r_{\mathcal{C}_j}^2, \tag{17}$$

$$\mathbf{c}_{\mathcal{C}_j} = \frac{1}{2}(\mathbf{p}_{\mathcal{C}_j} + \mathbf{q}_{\mathcal{C}_j}) \quad \text{and} \quad \mathbf{c}_{\mathcal{B}_j} = (1-\kappa)\,\mathbf{p}_{\mathcal{C}_j} + \kappa\,\mathbf{q}_{\mathcal{C}_j}, \tag{18}$$

where the latter depends on the orientation of the overall model as explained below.

For a given pose, the position of the ball $\mathcal{B}_j$ is a function of the angle between the gravity direction and the capsule axis

$$\mathbf{c}_{\mathcal{B}_j} = (1-\kappa)\,\mathbf{p}_{\mathcal{C}_j} + \kappa \mathbf{q}_{\mathcal{C}_j}, \quad \text{where} \tag{19}$$

$$\kappa = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z > 0 \end{cases}, \quad \text{with} \quad z = \frac{\mathbf{g} \cdot \mathbf{R}_{\sigma(j)}(\mathbf{q}_{\mathcal{C}_j} - \mathbf{p}_{\mathcal{C}_j})}{\|\mathbf{q}_{\mathcal{C}_j} - \mathbf{p}_{\mathcal{C}_j}\|}. \tag{20}$$

To remove the discontinuity we instead use a sigmoid function

$$\kappa = \frac{1}{1+e^{-\gamma z}}, \tag{21}$$

where γ is a sufficiently large number.

To avoid convergence to a solution where a ball is not clearly located at either end for the desired poses, we add constraints

$$(\arccos(z) - \frac{\pi}{2})^2 \geq \phi_{\min}^2, \tag{22}$$

where $\phi_{\min}$ is a tolerance angle (in practice 3 degrees are sufficient).