Dual Balance

Delainey Ackerman, Srivamshi Pittala, Athina Panotopoulou

November 23, 2014

Abstract

Our project concerns with 3D-printing objects that can attain a balanced pose even when other objects/materials are added to/removed from them. We limit the scope of this project to two poses that a user can specify. So, our problem statement is to 3D-print objects which can balance in two distinct, stable configurations, one without any external material and the other with material. Users can specify the configurations and the amount/type of material that will be added.

1 Motivation and Related Work

Our primary motivation comes from the many products we use in our daily life, like cups, bowls, stands etc which can hold different materials and are stable in multiple configurations. We are also motivated by acts like egg balancing, cup balancing etc which have an illusory effect of impossible configurations. Such configurations are a result of clever design and physics. In this project, we set out to see if it possible to computationally fabricate such objects. Prior work concerning balance of 3D models was done in Make It Stand [1], where their program optimized a 3D model, ensuring that the printed object will stand in its intended configuration. Our work will be building on theirs to accommodate dual balanced configurations with the option to add other materials.

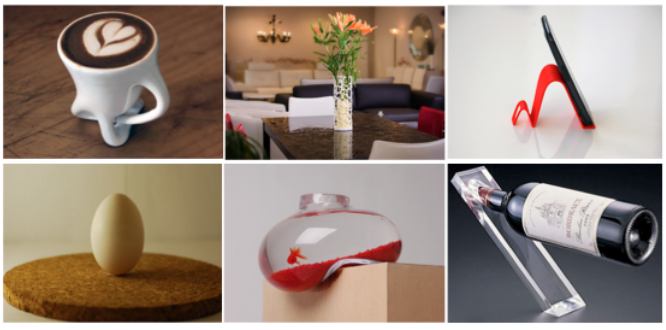


Figure 1: Illustrations of scenarios where objects are in stable configurations

2 Technical Approach

We follow the approach in Make It Stand[1] to optimize for the balanced configuration by carving the inner surface and deforming the shape. The algorithm's input constitutes the object's design M that is a triangle mesh representing the surface, the density of the constructing material ρ_c, the density of the filling material ρ_l, the desired points of contact for the unloaded c_u^*, and loaded object c_f^* respectively, the height h upto which the cup is going to be filled. The output of the program is going to be the M_O, M_I that is the outer and inner surfaces of the object we are going to print.

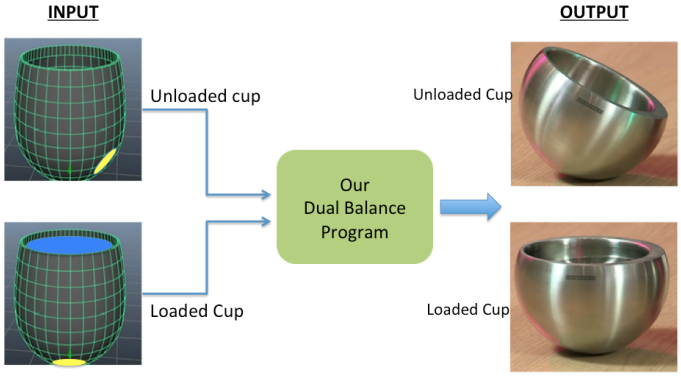


Figure 2: Proposed workflow of our program. User provides the mesh, material properties and two intended bases of support(shown in yellow).

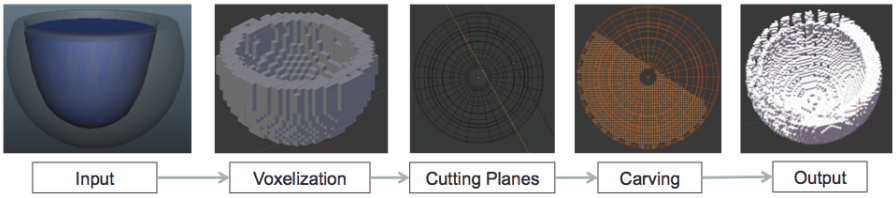


Figure 3: Outline showing the steps in our Dual Balance program

2.1 Voxelization

The input mesh is descretized into voxels to represent the volume inside. So, the mass carving method is effectively applied using this voxel representation. Each voxel is associated with a binary variable indicating if it is carved or not. The set of all binary values for the mesh is represented by α. In our experiments we used a resolution of $64 \times 64 \times 64$ for this step.

2.2 Computation of mass

Given the user input, we compute at each iteration of the algorithm, the real center of mass for the filled c_f and the unfilled cup c_u, as well as the mass m_f, and m_u for the filled and unfilled cup. Also, we define as $V = \{v_1, \ldots, v_{|V|}\}$ the set of voxels that we hollow from the interior of the cup, which we describe in detail how we compute in Section 2.3, and $c_u(i)$ and $c_f(i)$ represents the real center of mass for the unfilled and the filled cup if we hollow from the the cup the voxels v_1 through v_i.

The center of the mass of an object of density ρ is given by the formulas [1]:

$$c = \frac{\rho}{24m} \sum_{f \in F} ((v_j - v_i) \times (v_k - v_i)) \star g(v_i, v_j, v_k)$$

m is the mass defined as $m = \frac{\rho}{6} \sum_{f \in F} ((v_j - v_i) \times (v_k - v_i)(v_i + v_j + v_k)$, $f = (v_i, v_j, v_k)$ denotes a triangle face of the mesh F, defined by the vertices v_i, v_j, v_k and $\star$ denotes the component-wise product. The function g equals $g(v_i, v_j, v_k) = v_i \star v_i + v_i \star v_j + v_j \star v_j + v_j \star v_k + v_k \star v_k + v_k \star v_i$. We compute c_u and m_u using the equations above.

The mass of the voxel v (m_v) can be computed once at the beginning of the algorithm, since its length l is fixed. The center of the mass for the voxel v, c_v is the center of the voxel, that is computed during the voxelization. The center of mass of the carved mesh can be computed efficiently by subtracting the carved voxels' mass since all voxels are the same size and mass.

To compute the c_f, we need first to compute the mass of water m_w and its center of the mass c_w. We create the mesh representation of the water as following: we use the interior of the cup and a plane parallel to the surface at height h and we take their intersection. The center of the mass for the filled cup then is computed as [3]:

$$c_f = \frac{c_u m_u + c_w m_w}{m_u + m_w}$$

2.3 Carving

Similar to the carving technique in the paper "Make it stand" [1], we optmize to decrease the gap between the actual center of mass of the mesh and the desired contact point on the surface. Thus we formulate it as a minimization problem as follows

$$E_i = \frac{1}{2} \| (c_u(i) - c_u^*)^{\perp g} + (c_f(i) - c_f^*)^{\perp g} \|^2$$

where i indicates the set of voxels carved out and $x^{\perp g}$ is the projection of x in the direction of gravity.

We initialize this step with all voxels filled i.e $\alpha_0 = 1$. At the first iteration we draw two cutting planes P_u and P_f, one each through the desired centers of mass c_u^* and c_f^*, which are perpendicular to $(c_f - c_f^*)^{\perp g}$ and $(c_u - c_u^*)^{\perp g}$ respectively. We consider a voxel v_i as a candidate for carving if its signed

distance d_i from both planes is positive. The signed distance from the plane is given as $d_i = (r_i - c^*) \cdot (c_0 - c^*)^{\perp_g}$. We sort the voxels in decreasing order of their combined distance from both planes and begin carving from the highest distance. We do not carve voxels that are very close to the mesh surface as this would affect the appearance of the object. For every candidate voxel v_i we keep track of the energy of E_i.

We iterate until we have no more candidate voxels and select the i that gives the minimum energy value. We combine this set of voxels $v_1, \ldots, v_i$ by calculating the boolean union into one mesh and we subtract from the mesh of the original object M, that gives us the output mesh $M' = M_O \cup M_I$

3 Results and Limitations

We implemented our program on two cups. One cup was designed with an off-tilted design such that there is more mass on one side compared to the other. The second cup has a symmetrical design. The first, asymmetrical cup, was designed so that it would naturally achieve the empty, tilted position of the two desired balance positions with little to no carving from our program. This design was used as a proof of concept that dual balance with 3D printed objects with the addition and removal of a second material was possible. The second, symmetrical, cup was designed to rely fully on our program to achieve both the empty, tilted and full, upright balance positions. Figure 4 shows the results of running our program on the second, symmetrical cup design which achieves the desired dual balance positions.

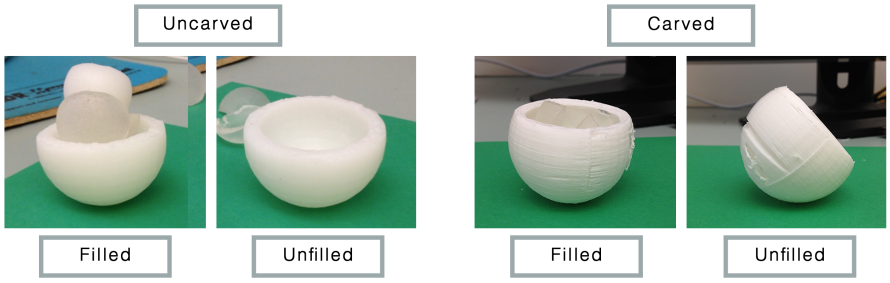


Figure 4: (Left) Shows the symmetrical cup design without any carving from our program. (Right) Shows the two distinct balance positions achieved after inner carving is applied to the symmetrical cup design.

Limitations to achieving a dual balanced position involve the limits of inner carving to change the center of mass of an object. Without any shape deformation, carving has an immutable limit to how much volume can be carved away from an object in order to shift its center of mass. This means that our program does not guarantee a solution, it will carve as many voxels as possible from within the object, but without deformation, the center of mass may not

change enough. An example of this is shown in Figure 5 where a symmetrical cup design is given as input with an empty desired balance on one side of the sphere while the filled desired balance is on the opposite side. Without deformation, no solution exists to get the symmetrical design to balance on the opposite side of the original center of mass compared to the desired carved position.

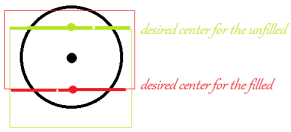


Figure 5: remains unsolved

4 Conclusions - Future Work

Here we need to notice that this technique does not always find a solution since it might be the case that the set V is too small and/or cannot reduce sufficiently the energy fig:6.

4.1 Extension to shape deformation

We can extend this method to the deformation part defined in the paper [1]. In addition to the derivatives described on the paper we compute also the derivatives that correspond to the filled cup:

$$\frac{\partial c_f}{\partial H} = \frac{c'_u m_u}{m_u + m_w} + \frac{c'_w m_w}{m_w + m_u} + \frac{c_u m'_u}{m_u + m_w} + \frac{c'_w m'_w}{m_w + m_u} - \frac{c_u m_u (m_u + m_w)'}{(m_u + m_w)^2} - \frac{c_w m_w (m_u + m_w)'}{(m_u + m_w)^2}$$

The terms c'_w, m'_w can be computed just like c' and m', substituting the cup mesh with the water mesh. In order the above to hold we need to think of the level of the water as a constant, in contrast to the mass of it that now changes.

4.2 Extension to stability

In contrast to the standing applications of the paper [1], we do not always have a convex hull that ensures stability for a specific toppling angle(γ). Since our contact surface is not bound, we use the assumption as the paper [1] does for the suspended objects that: "the user is responsible for selecting a position that once reached remains stable". One way to investigate quantitatively how stable is an object described here. We assume that the surfaces of standing support have smooth convex surfaces. Initially we find the standing support surfaces, by rotating the object in different positions and examining if the center of the mass projects into a small interval of the support point. We could investigate how stable these stable positions are using the following idea. At a specific stable position how much does it need the object to tilde to move to some other stable

position, what is the minimum angle to succeed that? If the projected center of mass is closer to some other projected stable position than to its existing, it seems like the object is going to transfer from one position to the other. An illustrative example of this is an egg-shape design that its center of mass is concentrated on the botton, compared to one that its mass is concentrated on the top, the later one is more unstable than the first fig:6.

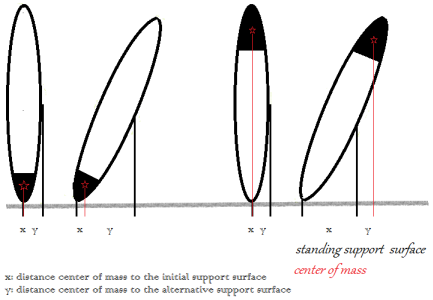


Figure 6: stability

Also, a cleverer way of carving that takes into account unsolvable problems could give us much better results.

More than two positions could also be used for identification tasks.

References

[1] R. Prvost, E. Whiting, S. Lefebvre, O. Sorkine-Hornung,*Make It Stand: Balancing Shapes for 3D Fabrication, ACM SIGGRAPH 2013*

[2] M. Baecher, E. Whiting, B. Bickel, O. Sorkine-Hornung, *Spin-It: Optimizing Moment of Inertia for Spinnable Objects, ACM SIGGRAPH 2014*

[3] http://hyperphysics.phy-astr.gsu.edu/hbase/cm.html