

Project: Make it Stand

Computational Fabrication

Alessandro Gobbetti

Contents

1	Introduction	1
2	Method	1
2.1	Volume Representation	1
2.2	Identifying the Convex Support Polygon	1
2.3	Center of Mass	1
2.4	Stabilizing the Object	2
3	Expanding the Base	2
4	Results	2
5	Running the Code	4



June 23, 2024
Università della Svizzera italiana
Faculty of Informatics
Switzerland

1 Introduction

This project aims at stabilizing shapes for 3D fabrication. The main idea is to carve the interior of the object to move the center of mass in a way that the object can stand on its own. The project is inspired by the work of [1].

2 Method

An object is considered stable if its center of mass lies on top of the convex support polygon.

2.1 Volume Representation

To simplify the problem, the mesh is first voxelized. This is done by creating a grid of voxels and checking if the center of the voxel is inside the mesh by shooting rays in multiple directions and counting the number of intersections with the mesh. If the number of intersections is odd, the voxel is considered inside the mesh. The ray directions are chosen following the Fibonacci sphere algorithm to ensure a uniform distribution of rays and then applying some random jittering to avoid artifacts.

Since every voxel computation is independent, the algorithm can be parallelized using OpenMP to speed up the voxelization process.

2.2 Identifying the Convex Support Polygon

The base of the object is the set of voxels in the lowest layer of the object. The object is balanced not only if the barycenter lies in the base but also if the barycenter lies on top of the convex hull of the base. The convex hull is computed using the Graham scan algorithm. A starting point is chosen as the point with the lowest y coordinate, and the points are sorted by the angle they form with the starting point. The first two points always belong to the convex hull, for the rest of the points, we keep track of the last three points in the convex hull and check if the new point forms a left turn with the last two points. If it does not, we remove the last point from the convex hull until the new point forms a left turn with the last two points. The algorithm is $O(n \log n)$, where n is the number of points.

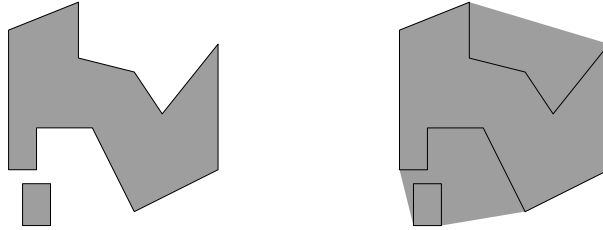


Figure 1: Convex hull of a complex shape: left the original shape, right the convex polygon.

The polygon is then triangulated to keep a voxelized representation of the convex hull for future easier computation of the barycenter.

2.3 Center of Mass

The center of mass is computed as the sum of the mass of each voxel times its position divided by the total mass. In this case, we consider 1 as the mass of each voxel that is inside the object and 0 otherwise. The center of mass is then projected on the convex hull to check if it lies inside. If it does not, the object is not stable.

2.4 Stabilizing the Object

To stabilize the object, we need to move the center of mass on top of the convex hull. This can be done by carving the object (removing voxels from the interior of the object). But which voxels should be removed? We first determine the direction from the barycenter to the closest point on the convex hull. We then define two semiplanes orthogonal to this direction that intersect at the barycenter (see [Figure 2](#)). If any voxel on top of the semiplane not containing the convex hull is removed, the center of mass will move towards the convex hull, thus improving the stability of the object.

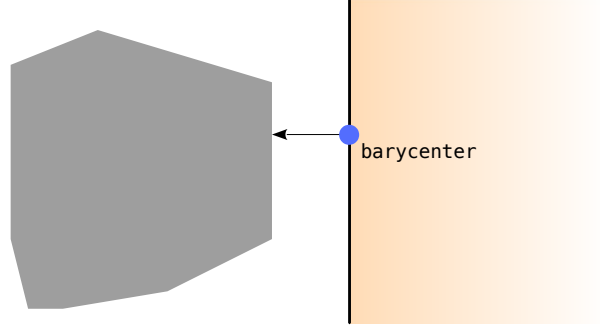


Figure 2: The barycenter (in blue) and the semiplane (in orange) defined by the direction from the barycenter to the closest point on the convex hull.

Due to voxelization, and in cases where the barycenter is already close to the convex hull, the direction might not be accurate. To improve the accuracy, we consider the closest 16 voxels in the base to the barycenter and we average them to get a more accurate direction.

The interior voxels of the object are then sorted by the distance from the semiplane. The voxels are removed starting from the farthest ones (that have a higher impact on the center of mass) until the center of mass lies on top of the convex hull.

The interior voxels are previously marked as such by checking their distance from the exterior voxels. For further stability, the convex hull can also be eroded to avoid the barycenter being on the edge of it making the object more stable.

During the carving process, the semiplanes are recalculated every n voxels removed to ensure the direction is still accurate.

3 Expanding the Base

Contrary to the paper [\[1\]](#), the shape of the object is not deformed, instead when stability is not achieved with carving, the base is dilated in the direction of the barycenter. The base is expanded by adding voxels in the direction of the barycenter until the center of mass lies on top of the convex hull or a maximum number of iterations is reached. For fabrication purposes, voxels are not added only in the layer of the base but also in the layers above it to ensure the object is stable.

4 Results

To test the performance of the algorithm, we created inclined rectangular prisms with different angles: 10° , 15° and 20° . The results are shown in [Figure 3](#). The prism (a) was already stable without any carving. The prism (b) was carved to achieve stability. The prisms (c) at 20° were not stabilized by carving, but their base was expanded to make them stand on their own as shown in (d). In this way, the object is not deformed, and the original shape is preserved and a small number of voxels are added to the base to stabilize the object.

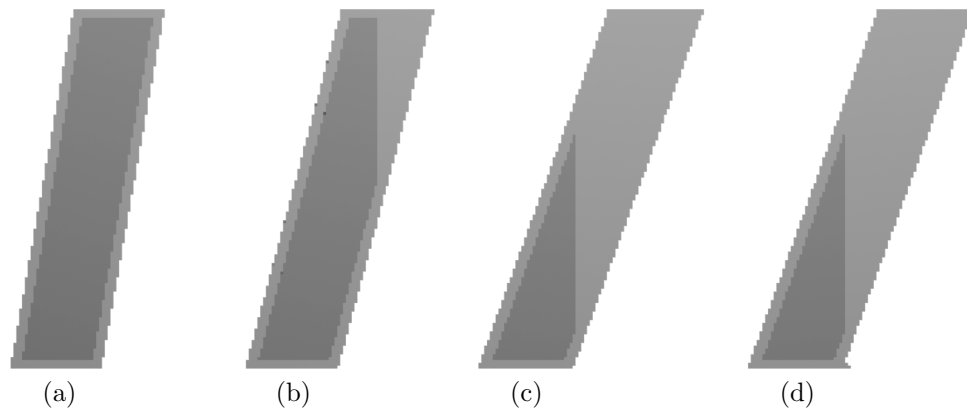


Figure 3: Inclined rectangular prisms with different angles: (a) 10° , (b) 15° , (c) and (d) 20° . The light gray voxels are the exterior voxels, the dark gray voxels are the interior voxels. The base of (d) has been expanded to stabilize the object.

A more complex shape is shown in [Figure 4](#). The banana shape was stabilized by carving the interior of the object. The base was also dilated in the direction of the barycenter to further stabilize the object.

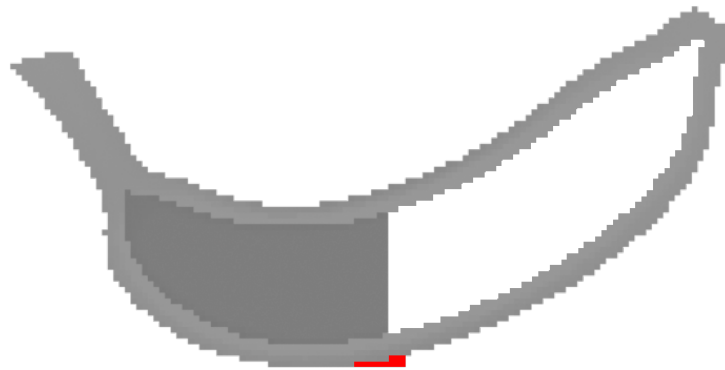


Figure 4: Complex shape of a banana. The light gray voxels are the exterior voxels, the dark gray voxels are the interior voxels. In white the removed voxels. The base has been expanded (in red) to further stabilize the object.

Another example is the model of the monster-like figure shown in [Figure 5](#). The object was inclined and balanced on its head and the base was slightly expanded to stabilize it.

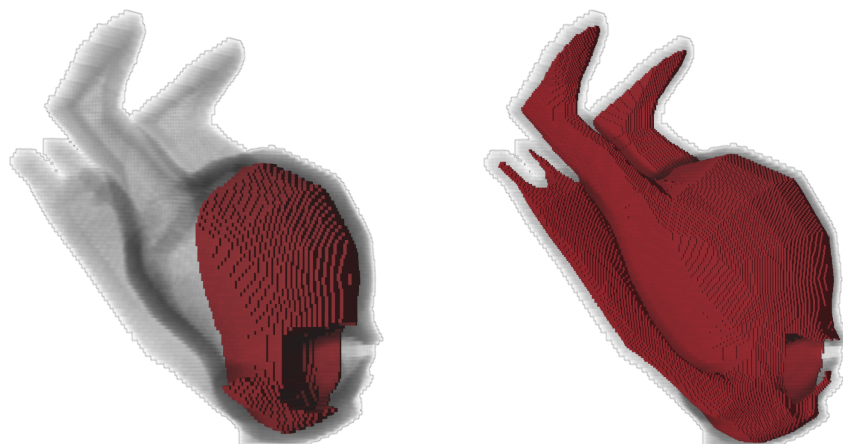


Figure 5: Monster-like figure. In red the interior voxels, in gray the rest of the voxels. The base has been expanded to stabilize the object. Left: stabilized object, right: original object.

5 Running the Code

The code is written in C++ and uses cmake to compile. It has been successfully tested on Linux and MacOS. To compile the code, run the following commands:

```
1 mkdir build
2 cd build
3 cmake ..
4 make
```

To run the code, navigate to the `build` directory and run the following command:

```
1 ./make_it_stand <input_file> <output_file> [OPTIONS]
```

where `<input_file>` and `<output_file>` are obj files and the options are:

- `--dim <dimension>` : Dimension of voxel grid (default: 32)
- `--num-rays <number>` : Number of rays per voxel (default: 2)
- `--jitter <jitter>` : Jittering factor for ray directions (default: 0.1)
- `--distance <distance>` : Distance to mesh surface when carving (default: 3)
- `--base-distance <distance>` : Distance to base border when computing stability (default: 1)
- `--max-base-dilation <number>` : Number of iterations for base expansion (default: 10)
- `--help` : Display help message

References

- [1] Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. Make it stand: balancing shapes for 3d fabrication. *ACM Trans. Graph.*, 32(4), jul 2013. [1](#), [2](#)