Normal Transformations for Overhang Reduction

Carlos de Castro and Leonardo Sacht

Abstract

We present a partial solution for the problem of reducing overhanging parts of a surface to 3D print it with minimal number of supports.

We first present a summary about how a 3D printer works and why the overhanging problem happens. This review focuses on a specific type of 3D printer that uses polymer melted to print solids on cross-sectional layers. We then do a fast review of three-dimensional surface representation in a computer and its discrete representation. Then we present our minimization problem and show some test results, using libigl library and gptoolbox functions, to observe the solution of problem.

Introduction

3D printing is a revolutionary method for bringing ideals to the world more easily. The 3D printing process is gaining space in many areas around the world. Its versatility can be used to print artistic objects, architectural mock ups, civil constructions, aerospace models, parts of physics experiments, educational instruments, as well as delicate objects as prosthesis and real representations of human organs. When we need to print some 3D solid in a 3D printer, some parts of this solid may be suspended in the air and need a support for a better print. However, these supports are detached from the surface and will not be reused, leading to a waste of material, time and money.

To avoid overhangs the 3D printers print columns to support the part of this solid that have no material underneath them. This extra material must be removed from the solid, leading to a waste of material, time and money.

However, some overhangs are tolerable. Each printer comes standard with a limiting angle to tolerate these overhangs. As we can see in Fig. 1, given a limit angle, the printer only prints an overhang support if the part of the solid that will be printed maintains an angle with the horizontal less than its limit.

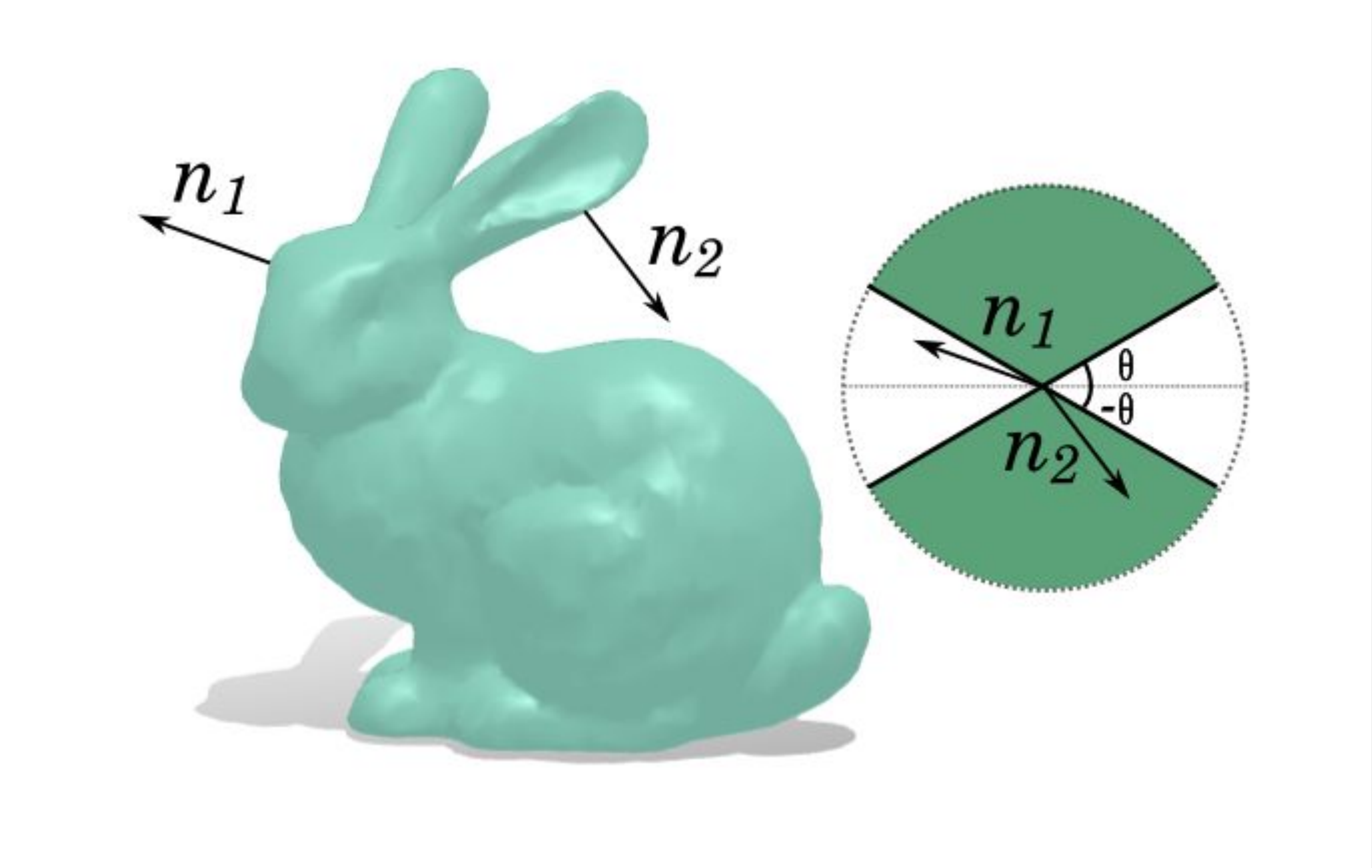

Fig. 1 - This two normal n_1 and n_2 of the bunny surface has different angle. The normal vector n_1 is in the angle parameters of the printer. The normal vector n_2 is in a overhang part and will need a support to print correctly.

We propose a formulation for the overhang problem based on the normal field of a surface and an optimization to find a global rotation of a surface that minimizes overhanging parts that cannot be printed without supports. This global rotation does not change the surface since the printed object can be derotated in the real world after printing.

E-mail: lealdecastro@gmail.com // leonardo.sacht@ufsc.br

DEPARTAMENTO DE MATEMÁTICA

SIBGRAPI'19

Normal Transformations for Overhang Reduction

Carlos de Castro and Leonardo Sacht

Objective Function

A solid can be represented in the computer by its vertex matrix $\mathcal{V} \in \mathbb{R}^{n\times 3}, n = |\mathcal{V}|$ and triangle mesh matrix $\mathcal{F} \in \mathbb{R}^{m\times 3}, n = |\mathcal{F}|$.
Each triangle in the mesh has a normal vector associated and the set of all normal vector in a triangle mesh is called by **Normal Field**. We can organize them on the normal matrix $\mathcal{N} \in \mathbb{R}^{m\times 3}$
An example of a normal vector associated to a triangle can be seen on Fig. 2.

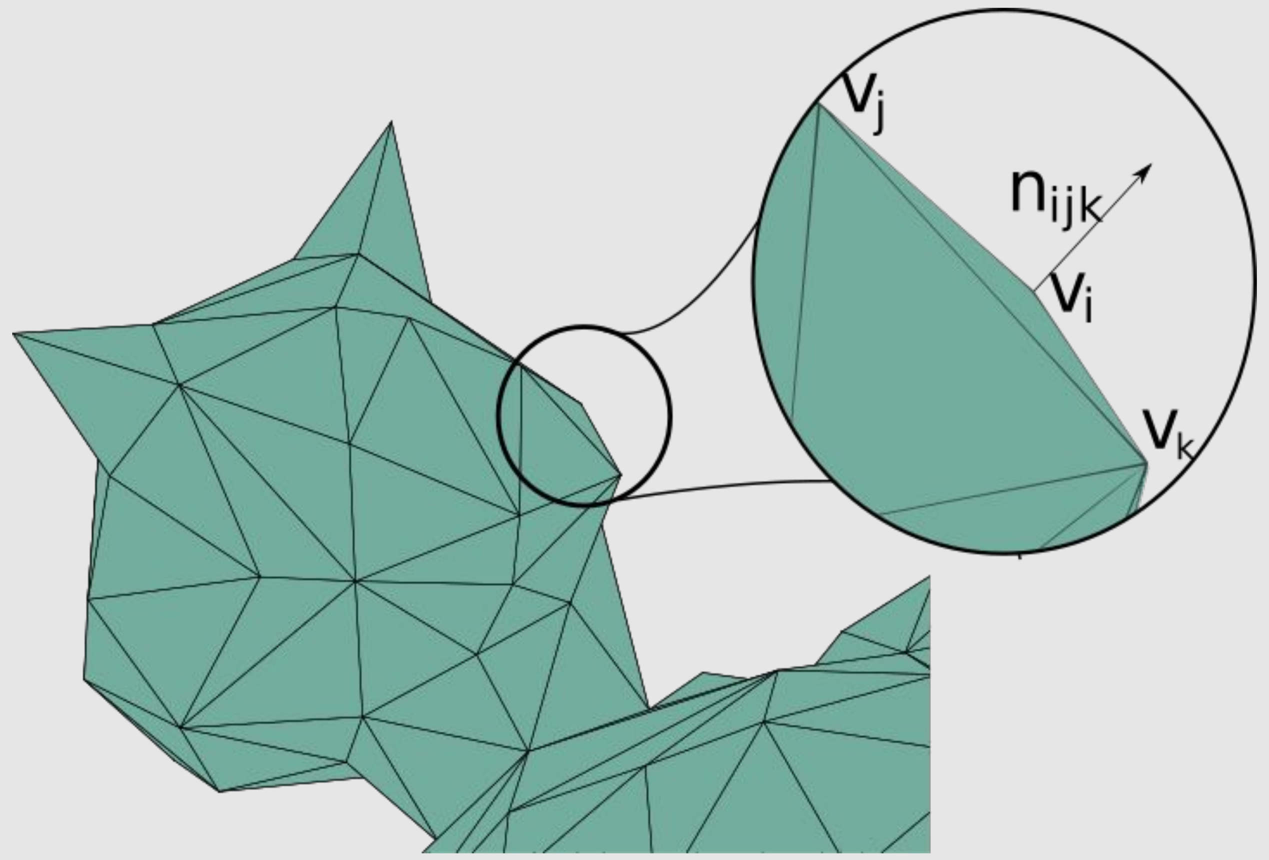

Fig. 2- In this surface, we have the vertex matrix and its triangle mesh matrix. Each triangle has three possible normal associated, but is easy to see that is all the same vector. So, we need only one of them to represent the triangle face normal vector.

Given a triangle mesh matrix, we can find the angle of the normal of each triangle with the vertical direction, defining the function

$$\alpha : \mathcal{F} \longrightarrow \mathbb{R}$$
$$\alpha(f) = \arccos\left(n(f)^T e_3\right)$$

where e_3 is the unit vector $e_3 = (0,0,1)^T$. So, with the normal field, the angle of each normal and the printer limit angle for overhang, we can seek for the minimal residual error of each normal angle and the threshold angle and sum over all triangle of mesh by

$$\min_{\mathcal{N}} \sum_{f\in\mathcal{F}} \max\left\{\left|\alpha(f) - \frac{\pi}{2}\right| - \theta, 0\right\}.$$

Implementation and results

The libraries we are working with are LIBIGL (C++) and GPTOOLBOX (Matlab). Initially, we are seeking for a global rotation of the surface that, given the normal field of the surface, find a new normal field that minimizes the overhang parts of the surface. We pre-processed all surfaces in our tests to be centered at the origin, with the goal of having the global rotations more effectively reducing overhangs.

Using some surfaces available on LIBIGL, we were able to perform some tests to evaluate if it is possible to find rotated surfaces with normal fields that satisfy our objective equation. Initial results show, as we can see in Tab. 1, that in every libgl surface tested we find an x-axis and y-axis angle global rotation that find a minimum value to our equation. Notice that varying rotations about the z-axis would not change overhangs, thus these rotations were not considered.

Surface(.obj)	**x-axis angle**	**y-axis angle**	**Initial**	**Minim**
camel_b	136^o	0^o	235.29	149.90
horse_quad	300^o	332^o	565.53	214.91
bunny	304^o	136^o	812.80	346.82
decimated-knight	268^o	208^o	120.44	43.27
face	24^o	192^o	461.54	432.22
cow	292^o	140^o	883.05	167.16
armadillo	48^o	0^o	6979.2	4646.4

Tab. 1 - All these surfaces present a smaller sum in our equation with an x-axis and y-axis angle rotation. These surfaces were chosen from the libigl library.

E-mail: lealdecastro@gmail.com // leonardo.sacht@ufsc.br

DEPARTAMENTO DE MATEMÁTICA

SIBGRAPI'19

Normal Transformations for Overhang Reduction

Carlos de Castro and Leonardo Sacht

We also show in next figure 2D graphs that show the variation of the value in our equation on those combinations. As we can see in the Fig. 3, the rotation of x-axis and y-axis, and its combination, generate regions of minimal value of our equation and has a periodicity, that indicates more than one minimizer.

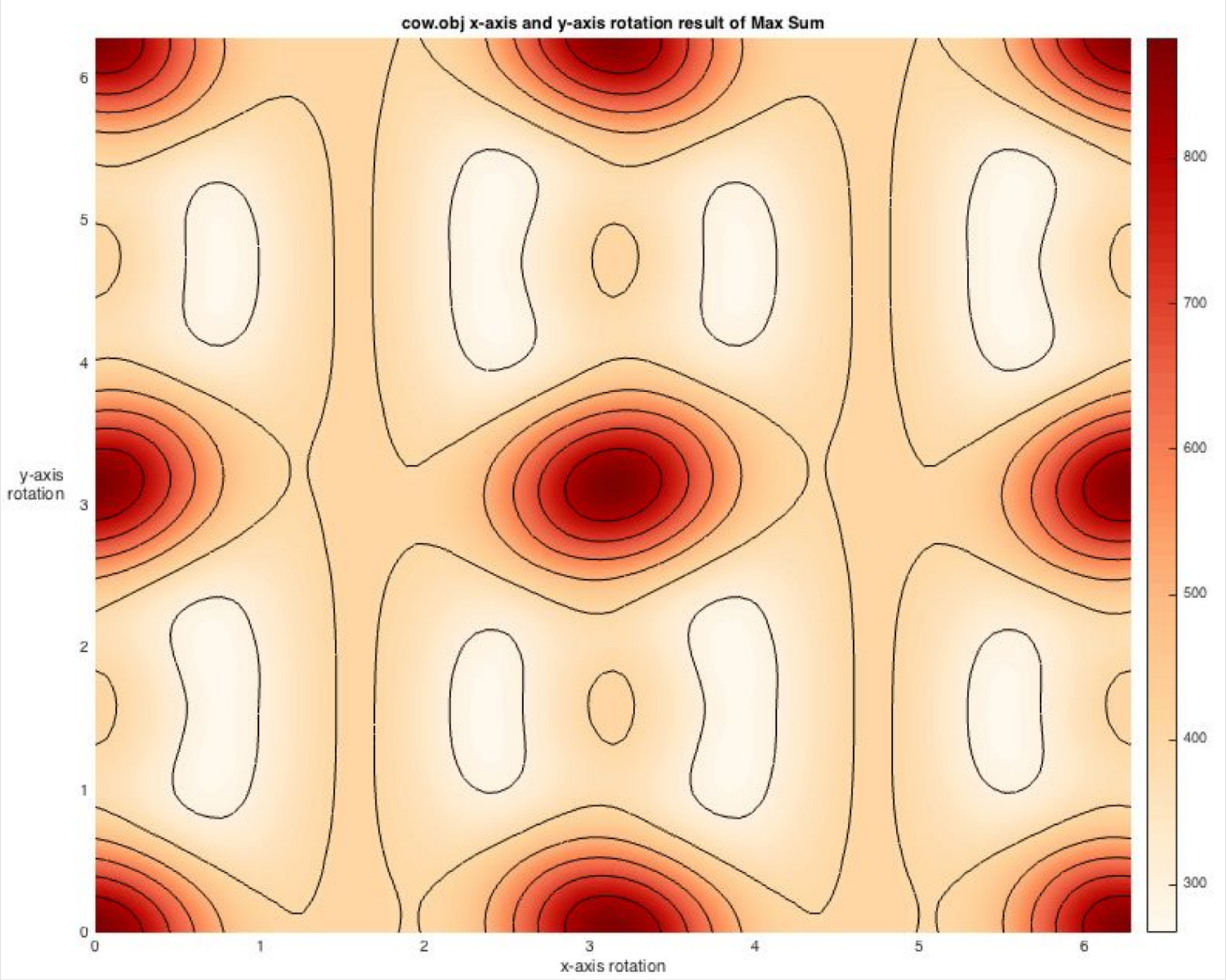

Fig. 3 - This graphic represent the value (in a gradient color) of sum of max on our equation, in a x-axis and y-axis rotation of *Cow.obj* libigl surface.

3D printers software, as Cura Maker, SLic3r, Kisslicer, and others, allow us to simulate solids and their overhang supports. So, we can visualize our tests result before printing them. We can see in Fig. 4, Fig. 5 and Fig. 6, that the vertex positions that result in the minimal value of our equation generate less supports for overhanging parts.

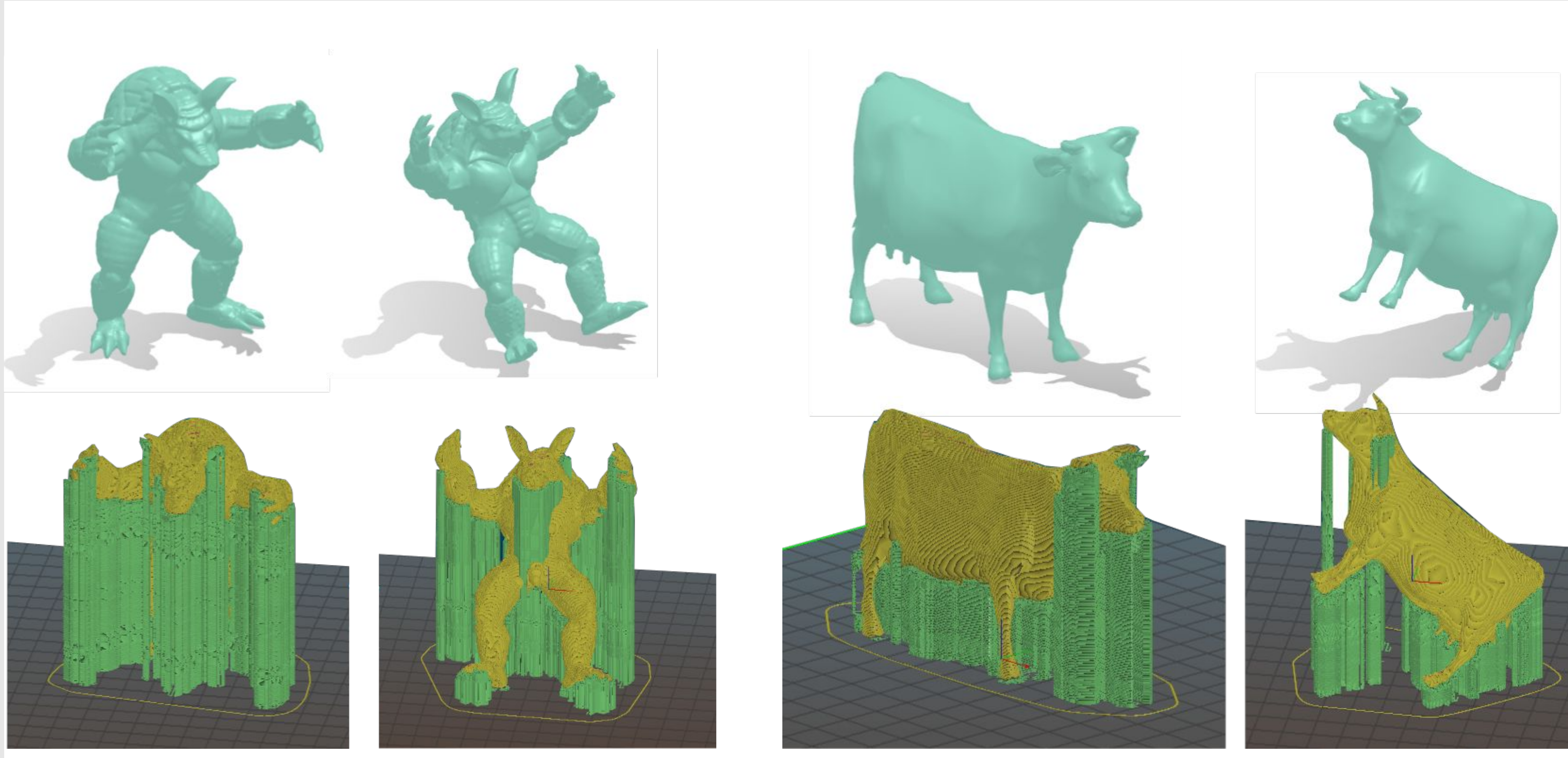

Fig. 4 - Armadillo.obj on initial vertex position (top-left) and after rotation (top-center-left). Cow.obj on the initial vertex position (top-center-right) and after rotation (top-right). Underneath them, their surfaces on a 3D print software.

E-mail: lealdecastro@gmail.com // leonardo.sacht@ufsc.br

DEPARTAMENTO DE MATEMÁTICA

SIBGRAPI'19

Normal Transformations for Overhang Reduction

Carlos de Castro and Leonardo Sacht

Fig. 5 - Face.obj on the initial vertex position (top-left) and after rotation (top-center-left). The same surface on the initial position on the print simulator (top-center-right) and after rotation (top-right). In yellow, the software indicates the solid that we want to print and in green the overhang supports. The bottom figure is a photo of same surface after printed. On the bottom-left, the surface is on the initial vertex position and on the bottom-right, the surface is positioned after rotation procedure.

Future Work

Our next step is to use a more principled way to determine the best rotation. The smoothness of the level curves in some of results suggests that a derivative-based numerical method or even the exact solution are possible.

After having the optimal global rotation determined, we plan to allow small local deformations in the surface to reduce overhangs even further. Then a local-global minimization (inspired in (SORKINE, 2007)) will be adopted to obtain an optimal state for the objective function.

Also, we will find an optimal horizontal partition of the surface to reduce overhang parts in a large height area of surface.

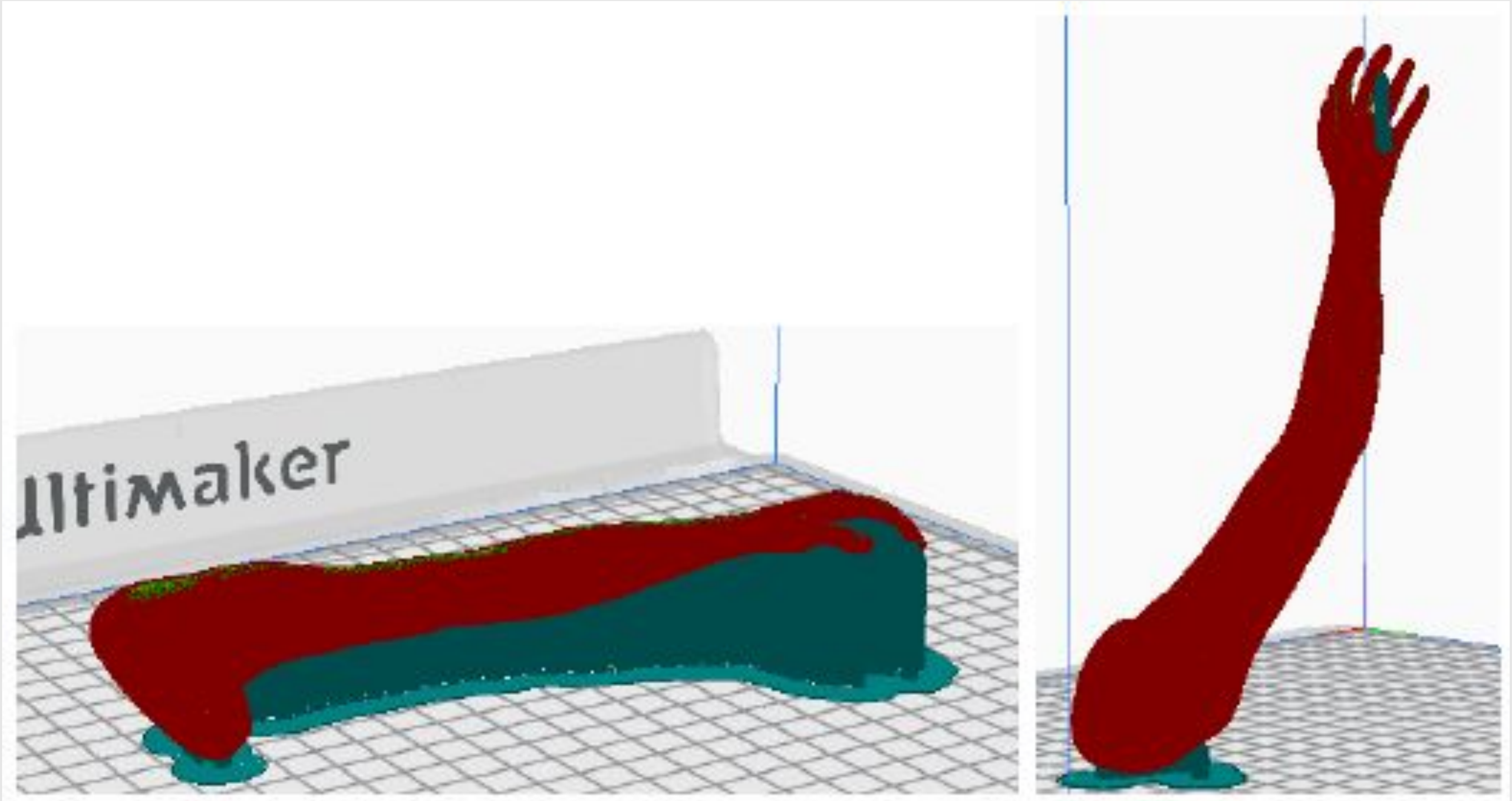

Fig. 6 - Arm.obj on the initial vertex position (left) and after rotation (right).

E-mail: lealdecastro@gmail.com // leonardo.sacht@ufsc.br