

Normal Transformations for Overhang Reduction

Carlos Eduardo Leal de Castro
Federal University of Santa Catarina (UFSC)
Florianópolis, SC, Brazil
Email: lealdecastro@gmail.com

Leonardo Sacht
Federal University of Santa Catarina (UFSC)
Florianópolis, SC, Brazil
Email: leonardo.sacht@ufsc.br

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.

I. INTRODUCTION

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.

Currently, the most affordable 3D printers use a printing technology known as *fused deposition modeling* (FDM). This printing process, as shown in figure 1, melts a type of polymer, that solidifies on the platform, or on the surface itself, printing the desired solid with cross-sectional layers.

Some parts of the solid need support because they are suspended in the air, which can damage the object or generate wrong prints. These parts are called *overhangs*.

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. As we can see represented in figure 2, the solid has overhanging parts that need the support to print correctly.

However, some overhangs are tolerable. Each printer comes standard with a limiting angle to tolerate these overhangs. Given a θ 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 θ .

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.

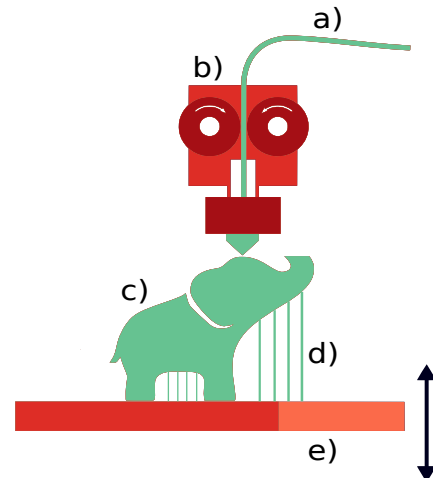


Fig. 1. A plastic filament (a) is horizontally heated by a controlled nozzle (b), allowing the template (c) to be printed over the horizontal platform (e), that moves on the vertical direction. Some sloping parts of the template need supports (d) to be printed.

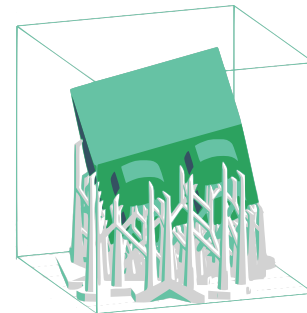


Fig. 2. This solid has parts that need support to print correctly. So the printer will build some structures to hold this parts while it is printed.

II. RELATED WORK

During the last years, the problem of reducing overhangs from a surface has been extensively studied and some partial solutions have been proposed.

Some of these solutions [4]–[6] partition the surface into smaller parts such that each subpart has little or no overhangs. These methods introduce seams between the different parts that are clearly visible after printing. Our method does not suffer from this limitation since we search for a global solution for the whole input surface.

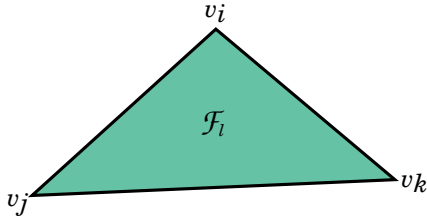


Fig. 3. In this example, the l -th row of triangle mesh matrix \mathcal{F} stores the row positions of the vertex v_i, v_j and v_k of the vertex matrix \mathcal{V} . So, this face, on the matrix, is represented by $F_l = (i, j, k)$.

Other methods [7]–[9] propose altering the surface to improve the final quality of the printed piece. These improvements can lead to overhang reduction. On the other hand, we directly formulate the overhanging problem and find a specific solution for it.

In [10] the authors propose to change the geometry of the supports to minimize the waste of material. We, instead, propose a general solution that is optimal for most supports generated by common 3D printing software. Other approach to the problem was proposed in [11], where the authors design new printing hardware to eliminate overhangs. Our current solution is purely based on software.

III. OBJECTIVE FUNCTION

To better analyze solids in the computer, we need to remember some properties of surfaces and their discretizations. This analysis will help us to understand a different way to compare the surface angles and the threshold angles for overhangs.

A solid can be represented in the computer by its vertex matrix $\mathcal{V} \in \mathbb{R}^{n \times 3}$, $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$, $n = |\mathcal{V}|$, in which each vertex $v_i \in \mathbb{R}^{1 \times 3}$ has the x-, y-, z- coordinates of the vertex $i = 1, \dots, n$. A way [1] to represent a surface mesh is storing the set of vertex indices on a matrix $\mathcal{F} \in \mathbb{R}^{m \times 3}$, $m = |\mathcal{F}|$, that represents the connection between the vertices and originates the triangular faces, i.e., the matrix \mathcal{F} has in each row the indices of rows of \mathcal{V} that, together, form a triangle mesh. In figure 3, we can see an example of an individual triangle mesh and its vertex.

Now we can define the important concept of normal vector:

Definition 1. Given a vertex matrix $\mathcal{V} \in \mathbb{R}^{n \times 3}$, $n = |\mathcal{V}|$ and a triangle $f = (i, j, k)$ of its triangle mesh $\mathcal{F} \in \mathbb{R}^{m \times 3}$, $m = |\mathcal{F}|$, the normal vector to the triangle f is

$$n(f) = \frac{(v_j - v_i) \wedge (v_k - v_i)}{\|(v_j - v_i) \wedge (v_k - v_i)\|}, \quad (1)$$

where $\wedge : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is the Euclidean cross product.

Each triangle in the mesh has a normal vector associated, as shown in figure 4. The set of all normal vector in a triangle mesh is called by *Normal Field*. We can store them in a normal matrix $\mathcal{N} \in \mathbb{R}^{m \times 3}$.

We can analyze overhanging parts of our printing solid using the normal field of its surface, as illustrated in figure 5: the faces that have normals belonging to the white region of the

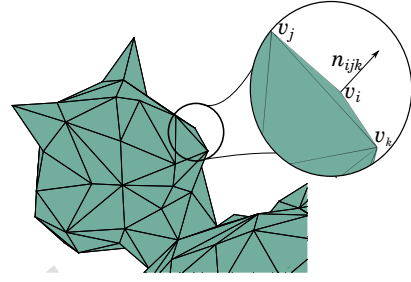


Fig. 4. In this surface, we have the vertex matrix \mathcal{V} and its triangle mesh matrix \mathcal{F} . Each triangle has three possible normal associated, but is easy to see that is all the same vector. So, we need only one of them to represent the triangle face normal vector.

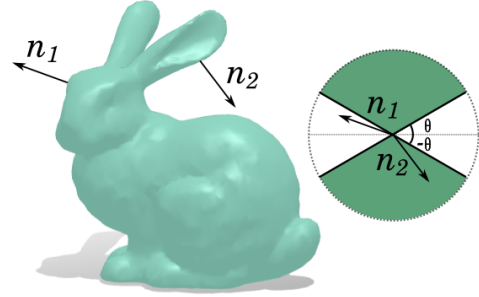


Fig. 5. This two normal n_1 and n_2 of the bunny surface has different angle. The normal vector n_1 is in the angle parameters of the printer. The normal vector n_2 is in a overhang part and will need a support to print correctly.

sphere on the right can be printed without supports and the ones that have normals in the green region need supports to be printed. At the time of printing, the user can choose the amount of surface infill, ranging from 0% to 100%. In our work we consider 0% infill, which corresponds to the worst possible case in terms of support needed for printing. So the parts of the surface that have normal vector pointing upwards will also be considered in the problem.

Given a triangle mesh matrix \mathcal{F} we can find the angle of the normal of each triangle with the vertical direction, defining the function $\alpha : \mathcal{F} \rightarrow \mathbb{R}$,

$$\alpha(f) = \arccos(n(f)^T e_3), \quad (2)$$

where e_3 is the unit vector $e_3 = (0, 0, 1)^T$. So, with the normal field \mathcal{N} , the angle of each normal and the printer limit angle for overhang, we can measure the residual error of each normal angle and the threshold angle θ and sum over all triangles of the mesh:

$$\sum_{f \in \mathcal{F}} \max \left\{ \left| \alpha(f) - \frac{\pi}{2} \right| - \theta, 0 \right\}. \quad (3)$$

Since we want to find a normal field that minimizes the sum in equation (3), we want to obtain the solution for

$$\min_{\mathcal{N}} \sum_{f \in \mathcal{F}} \max \left\{ \left| \alpha(f) - \frac{\pi}{2} \right| - \theta, 0 \right\}. \quad (4)$$

IV. IMPLEMENTATION AND RESULTS

The libraries we are working with are *libigl* [2] (C++) and *gptoolbox* [3] (Matlab).

Our main job will be to analyze the surface to be printed and return new vertex positions \mathcal{V} to be printed with minimal overhang support. 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 equation (4). Initial results show that in every *libigl* surface tested we find an x-axis and y-axis angle global rotation that find a minimum value to equation (3), as we can see in table I.

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

TABLE I

ALL THESE SURFACES PRESENT A SMALLER SUM IN EQUATION 3 WITH AN X-AXIS AND Y-AXIS ANGLE ROTATION. THESE SURFACES WERE CHOSEN FROM THE LIBIGL [2] LIBRARY.

Surface(.obj)	$ \mathcal{V} $	$ \mathcal{F} $	Time (s)
camel_b	2032	3576	648.05
horse_quad	2400	4796	781.32
bunny	3485	6966	1095.58
decimated-knight	502	1000	198.69
face	5272	10540	1813.17
cow	2762	5520	956.38
armadillo	43243	86482	15547.87

TABLE II

TIMINGS FOR RUNNING OUR ALGORITHM ON DIFFERENT INPUT SURFACES.

These tests were done on MATLAB, using some of *gptoolbox* [3] functions to read the vertex matrix \mathcal{V} and triangle mesh matrix \mathcal{F} of each surface file (.obj). Using these matrices, we can calculate the sum in (3) and test for all possible x-axis and y-axis angle combination with a $\Delta\theta = \frac{\pi}{45}$ for both axis. Notice that varying rotations about the z-axis would not change overhangs, thus these rotations were not considered. In the way the test algorithm was written, the search for the minimizer angles has resulted in long execution times. Table II shows the time of each test and the number of vertices and faces of each

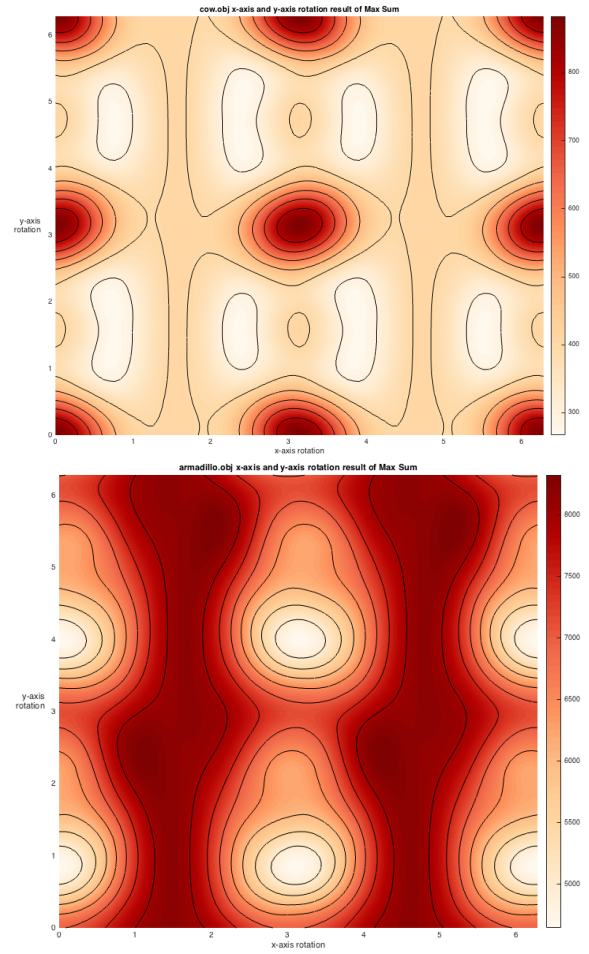


Fig. 6. These graphs represent the value (in a gradient color) of sum of max in equation 3 in a x-axis and y-axis rotation of *cow.obj* (top) and *armadillo.obj* (bottom) surface.

surface. We also show in figure 6 2D graphs that show the variation of the value in equation (3) on those combinations.

As we can see in figure 6, the rotation of x-axis and y-axis, and its combination, generate regions of minimal value of equation (3) and has a periodicity, that indicates more than one minimizer. The minimal value in equation (4), rotation result of *cow.obj* and *armadillo.obj* found on our initial tests can be seen in figure 7. Both results show that small rotations applied to the surfaces can lead to normal fields that result in less overhanging parts.

3D printing 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 figures 8 and 9 that the vertex positions that result in the minimal value of equation (3) generate less supports for overhanging parts. Figure 9 also shows physical prints of the input and output surfaces.

V. CONCLUSION AND FUTURE WORK

In this work we proposed a new objective function for the overhang problem and minimized it by sampling the space

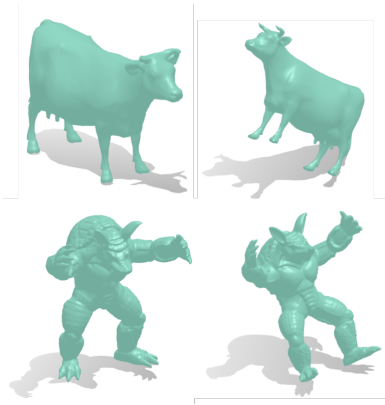


Fig. 7. Cow.obj on the initial vertex position (top-left) and after rotation (top-right). Armadillo.obj on initial vertex position (bottom-left) and after rotation (bottom-right).

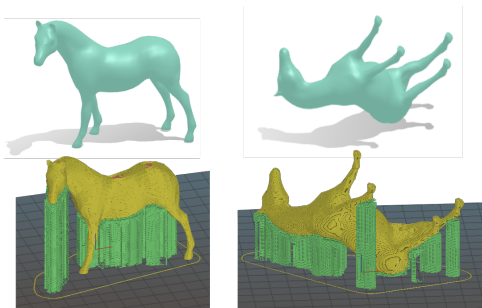


Fig. 8. Horse_quad.obj on the initial vertex position (top-left) and after rotation (top-right). The same surface on the initial position on the print simulator (bottom-left) and after rotation (bottom-right). In yellow, the software indicates the solid that we want to print and in green the overhang supports.

of rotations. Our next step is to use a more principled way to determine the best rotation. The smoothness of the level curves in figure 6 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 [12]) will be adopted to obtain an optimal state for the objective function.

Finally, we consider to generalize this methodology to other problems in geometry processing that involve transforming normal vectors and having them belonging to some region of the sphere. Some possibilities for this generalization include mesh smoothing [13] and bilateral filtering [14].

REFERENCES

- [1] BOTSCH, M. et al. *Polygon mesh processing*. AK Peters/CRC Press, 2010.
- [2] JACOBSON, A., PANOZZO, D., *libigl: A simple C++ geometry processing library*, ONLINE: <http://libigl.github.io/libigl/>, 2017.
- [3] JACOBSON, A. et al. *gptoolbox: Geometry Processing Toolbox* ONLINE: <http://github.com/alecjacobson/gptoolbox>, 2018.
- [4] HU, R., LI, H., ZHANG, H., COHEN-OR, D., *Approximate Pyramidal Shape Decomposition*, ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2014), volume 33, number 6, 2014.



Fig. 9. 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.

- [5] YAO, M., CHEN, Z., LUO, L., WANG, R., WANG, H., *Level-set-based Partitioning and Packing Optimization of a Printable Model*, ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2015), volume 34, number 6, 2015.
- [6] KARASIK, E., FATTAL, R., WERMAN, M., *Object Partitioning for Support-Free 3D-Printing*, Computer Graphics Forum, volume 38, number 2, 2019.
- [7] ZHANG, X., LE, X., PANOTOPOULOU, A., WHITING, E., WANG, C.C.L., *Perceptual Models of Preference in 3D Printing Direction*, ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2015), volume 34, number 6, 2015.
- [8] WANG, W. M., ZANNI, C., KOBBELT, L., *Improved Surface Quality in 3D Printing by Optimizing the Printing Direction*, Proceedings of the 37th Annual Conference of the European Association for Computer Graphics, 2016.
- [9] MARTÍNEZ, J., HORNUS, S., SONG, H., LEFEBVRE, S., *Polyhedral Voronoi Diagrams for Additive Manufacturing*, ACM Transactions on Graphics, volume 37, number 4, 2015.
- [10] VANEK, J., GALICIA, J.A.C., BENES, B., *Clever Support: Efficient Support Structure Generation for Digital Fabrication*, Computer Graphics Forum, volume 33, number 5, 2014.
- [11] DAI, C., WANG, C.L.C., WU, C., LEFEBVRE, S., FANG, G., LIU, Y., *Support-free Volume Printing by Multi-axis Motion*, ACM Transactions on Graphics, volume 33, number 5, 2014.
- [12] SORKINE, O., ALEXA, M., *As-Rigid-As-Possible Surface Modeling*, Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing, 2007.
- [13] YAGOU, H., OHTAKE, Y., BELYAEV, A., *Mesh Smoothing via Mean and Median Filtering Applied to Face Normals*, Proceedings of the Geometric Modeling and Processing — Theory and Applications (GMP'02), 2002.
- [14] ZHENG, Y., FU, Y., BELYAEV, A., *Bilateral Normal Filtering for Mesh Denoising*, IEEE Transactions on Visualization and Computer Graphics, volume 17, number 10, 2011.