3D printing support reduction via skinning deformation

Extended Abstract

Julia Gilenko
University of Toronto
julia@mail.utoronto.com

Peiqi Wang
University of Toronto
wangpeiq@mail.utoronto.com

Eris Zhang
University of Toronto
eris@mail.utoronto.com

ABSTRACT

This paper provides a sample of a LaTeX document which conforms, somewhat loosely, to the formatting guidelines for ACM SIG Proceedings. [1]

CCS CONCEPTS

• **Computing methodologies** → *Mesh geometry models.*

KEYWORDS

3D printing, deformation, geometry processing

ACM Reference Format:
Julia Gilenko, Peiqi Wang, and Eris Zhang. 2018. 3D printing support reduction via skinning deformation: Extended Abstract. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Introduction

When a model needs supports in order to be 3D printed, the printer must use extra material, which makes printing more expensive. Existing work on the problem of how to reduce supports focusses predominantly on how to print support the same areas using less material. However, if the user is not particular about how the model is positioned when it is printed, there may also be a deformation that would require fewer supports. Given a skeleton for the model, we suggest a method to find a deformation using linear blend skinning to minimize the number of supports the model would need to be printed.

Problem formulation. Let $\mathcal{M} = (\mathbf{V}, \mathbf{F})$ be a mesh living in dimension $d \in \{2, 3\}$. Let $\mathbf{V} = \{\mathbf{v}_1^T, \cdots, \mathbf{v}_n^T\}^T \in \mathbb{R}^{n \times d}$ be the rest-pose vertex positions. Given a set of control handles $\mathcal{H} = \{\mathbf{h}_1, \cdots, \mathbf{h}_m\}$, we can apply an affine transformation $\mathbf{T}_j \in \mathbb{R}^{d \times (d+1)}$ on each handle $\mathbf{h}_j$. Let $\mathbf{T} = \{\mathbf{T}_1^T, \cdots, \mathbf{T}_m^T\}^T \in \mathbb{R}^{(d+1)m \times d}$. Linear blend skinning is a deformation method whereby the positions of the vertices $\mathbf{V'}$ on the deformed shape are calculated with a weighted linear combination of the handles' transformations,

$$\mathbf{v}'_i = \sum_{j=1}^{m} w_j(\mathbf{v}_i)\mathbf{T}_j \begin{pmatrix} \mathbf{v}_i \\ 1 \end{pmatrix}$$

where $w_j : \mathcal{M} \to \mathbb{R}$ is computed using bounded biharmonic weights. Equivalently, $\mathbf{V'} = \mathbf{MT}$, where $\mathbf{M} \in \mathbb{R}^{n \times (d+1)m}$ is a matrix combining $\mathbf{V}$ and $\mathbf{W}$. Note that the energies below are a function of $\mathbf{V'}$, and hence a function of $\mathbf{T}$ if we substitute $\mathbf{MT}$ in its place, i.e.

$$E_{arap}(\mathbf{T}) = E_{arap}(\mathbf{T}, \mathbf{M}) = E_{arap}(\mathbf{V'})$$

where $\mathbf{M}$ is fixed.

Energy terms

ARAP energy. We can define the *as-rigid-as-possible* deformation energy, which measures local distortion, to be

$$E_{arap}(\mathbf{V'}, \mathbf{R}) = \frac{1}{2} \sum_{f \in \mathbf{F}} \sum_{(i,j) \in f} c_{ij} ||(\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}(\mathbf{v}_i - \mathbf{v}_j)||^2$$

Reformulating the ARAP energy to be a function of $\mathbf{V'}$ only, we get

$$E_{arap}(\mathbf{V'}) \qquad \mathbf{R} = \arg\min_{\mathbf{R}} tr(\mathbf{R}\tilde{\mathbf{K}}\mathbf{T})$$

In matrix form, this becomes

$$\begin{aligned} E_{arap}(\mathbf{V'}) &= tr(\frac{1}{2}\mathbf{V'}^T\mathbf{L}\mathbf{V'} + \mathbf{V'}^T\mathbf{K}\mathbf{R}) \\ &= tr(\frac{1}{2}\mathbf{T}^T\tilde{\mathbf{L}}\mathbf{T} + \mathbf{T}^T\tilde{\mathbf{K}}\mathbf{R}) \end{aligned}$$

where $\tilde{\mathbf{L}} = \mathbf{M}^T\mathbf{L}\mathbf{M} \in \mathbb{R}^{(d+1)m \times (d+1)m}$, $\tilde{\mathbf{K}} = \mathbf{M}^T\mathbf{K} \in \mathbb{R}^{(d+1)m \times dn}$ and $\mathbf{K}$ as defined in the deformation assignment

Overhang energy. An overhanging region that can be 3D printed without support is called *self-supported.* We call the angle between the region's tangent plane and printing direction the *self-supported angle* α. Let α_{max} be the maximum supporting angle. Let $\tau = cos(\alpha_{max})$ be the *maximal supporting coefficient.* Let $\partial\mathcal{M} \subset \mathbf{F}$ be boundary of mesh $\mathcal{M}$, i.e. surface faces. A surface face f is *risky* and thus requires support if $\mathbf{n}_f \cdot \mathbf{d}_p < -\tau$, where $\mathbf{n}_f$ is unit normal of f and $\mathbf{d}_p$ is the printing direction. Let $A(\cdot)$ the area function and $c(\cdot)$ be the centroid function for a face f. We can approximate the volume of support required for any face f by computing the volume of a rectangular prism,

$$\lambda(f) = \begin{cases} A_{base}h = A_f|\cos(\theta_f)|(\mathbf{c}_f \cdot \mathbf{d}_p) = A_f|\mathbf{n}_f \cdot \mathbf{d}_p|(\mathbf{c}_f \cdot \mathbf{d}_p) & \text{if } \mathbf{n}(f) \cdot \mathbf{d}_p < - \\ 0 & otherwise \end{cases}$$

where A_f is the area of the face, $\mathbf{c}_f$ is the centroid of the face, and θ_f is the angle between $\mathbf{d}_p$ and $\mathbf{n}_f$. We define an overhang energy that measures the volume of support required,

$$E_{overhang}(\mathbf{V'}) = \sum_{f \in \partial\mathcal{M}} \lambda(f)$$

[1] Code is available in a github repo

Self-intersection energy. To prevent ARAP from deforming the mesh in such a way as to cause self-intersections, we add a third energy to measure them. We define a *self-intersecting region* as the region inside the mesh that is bordered by intersecting tetrahedra. Given j such regions we define the self-intersection energy to be

$$E_{intersect}(\mathbf{V}') = \sum_{i=1}^{j} V_i$$

where V_i is the volume of the i^{th} region. To find these volumes, we define a grid below the mesh and trace a ray up from every grid point. If it enters the mesh twice in a row, we know the ray must be in a self-intersecting region. "Entering" simply means the direction of the ray is opposite to the vertical component of the normal direction of the surface at that point. When it exits this region, we record the distance the ray traversed inside it. The sum of all these lengths produces an approximation of the combined volume. The finer the grid, the more accurate the approximation.

Optimization

We define an objective function $E(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ as the sum of the three energy terms $E = E_{arap} + E_{overhang} + E_{intersect}$ and minimize E using the particle swarm optimization. Let $\mathbf{x} \in \mathbb{R}^{2dm}$ consisting a representation of centroid and rotation (as Euler's angle) about the centroid of for each edge handle. For shape in

Finally, we can convert each component to $\mathbf{T}_j$, affine transformations for edge handles, from which can compute deformed shaped $\mathbf{V}' = \mathbf{MT}$ with LBS.

Results?

Future work

In future work, we would like to speed up the calculations with a GPU. This would reduce wait time for the user and introduce the possibility of user interactivity, so that a user could have some control over the model's final position. We would also like to have the optimizer consider how the deformation shifts the model's center of mass, so that the model would still be able to stand on its own once it has been printed.

REFERENCES