



Title:

Creating More Flexible Loft Guides for Shapr3D

Authors:

Ákos Horváth, akos.horvath@shapr3d.com, Shapr3D
István Kovács, istvan.kovacs@shapr3d.com, Shapr3D

Keywords:

surface modeling, loft, interpolation, parametric modeling

DOI: 10.14733/cadconfP.2024.194-199

Introduction:

Shapr3D[1] is an innovative 3D modeling tool powered by the Parasolid geometry kernel[4], and it is used by industrial designers, engineers, product managers, manufacturers, and other design professionals worldwide. Shapr3D provides several different tools that allow the users to create and edit models, one of them is loft[2].

Loft[3] is a method to create smooth 3D shapes between cross-sectional profiles by interpolation. For more control over the shape in between the profiles, you can create guide curves and make use of connection points, see Fig. 1. There must be a minimum of two loft profiles, one located at the beginning of the shape and one at the end of the shape unless the loft is periodic, in which case the first and last profile is the same. Additional profiles can be defined between the start and end profiles as necessary.

Connection points are vertices of the loft profile or projected points between them, that help to identify how the vertices from a profile are mapped to vertices in the next profile. Connection points can be dragged to modify how the vertices are mapped. Making changes to connection points can be used to introduce twists into the loft feature, or to remove them. Guide curves are optional loft elements to control the shape of the resulting surface.

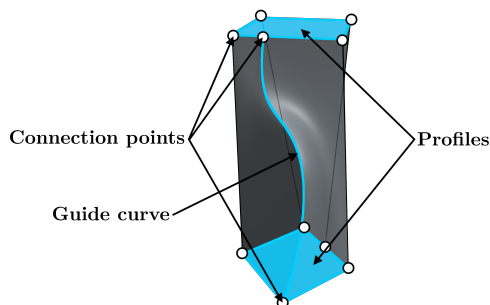


Fig. 1: Elements of loft operation.

More flexible loft guides

Previous work provides different approaches for creating loft surfaces: some methods try to find flexible representations of freeform surfaces[7], others introduce subdivision surfaces[8]. Lofting is widely used in all of the CAD systems, but to make them work you usually have to provide a complex curve system, that has connections in the right places. This needs a lot of user effort, which makes loft tools hard to use in the different software. There are other solutions, like in [9] that try to eliminate this complexity, by allowing the user to draw simple curves that can be converted into a lofted surface. These usually create inner cross-sections that make the created surface follow the drawn guide curves.

The existing solutions work in a similar way to Parasolid's loft operation, which mandates that guide curves are connected to the profiles, see Fig. 2. The goal of our project was to find a reliable method that allows choosing guide curves in a more flexible way. However, if a guide curve is connected to all of the profiles, we would like to retain the old behavior.

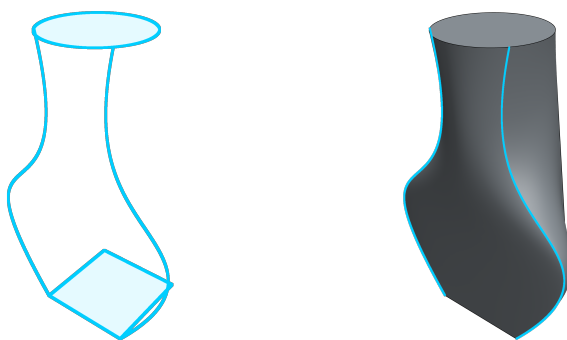


Fig. 2: Using loft tool with guide curves that are connected to the profiles.

Most of the current CAD systems (like Fusion360[5] and Onshape[6]) provide 2 different lofting methods: One where you provide guides that intersect every profile exactly once at corresponding vertices, and one where you provide a centerline type of guide that passes through each profile and will behave similarly to a sweep function. We would like to create a method where the selected guide curves are displaced in a way that they connect to the profiles so that one of the existing lofting methods can be applied to them. The main idea was to find points on the guide curves for each profile, and then find a method that would displace the guide curves so these points would be placed on the profiles at the connection points. Since we already had a mapping for the connection points between the different profiles, we simply had to find different displacements for the guide curves to pass through each chain of the connection points.

1. Find displacement vectors

First, we need to find points on the guide curves that can be matched to the different profiles, and to the connection points on the profiles. There were two alternative solutions we considered.

- Parameter matching, see Fig. 3
 1. Fit a cubic spline curve on the chain of connection points that are given on the profiles.
 2. Calculate the parameters of these vertices on the fitted spline.
 3. Calculate positions on the guide curve for the same parameter values.

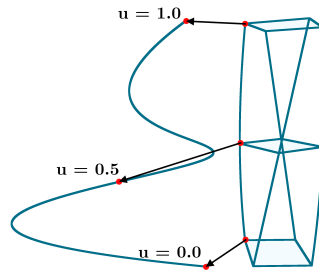


Fig. 3: Finding points on the guide curve by parameter matching.

Issues with this solution:

- If the original guide curve intersects all profiles, the calculated new connecting guide might be different from the original one, since the calculated points on the guide curve might not be in the intersections of the profiles and the guide. This would break the initial requirement that we needed a solution that would retain the old behavior.
- If we have a curve which overhangs the boundary profiles then we probably don't want to squeeze it between them, but rather trim the guide.

- Calculate planar intersections with the profiles, see Fig. 4
 1. Fit a plane on the connection point of the profile face.
 2. Find the intersection of the plane and the guide curve.

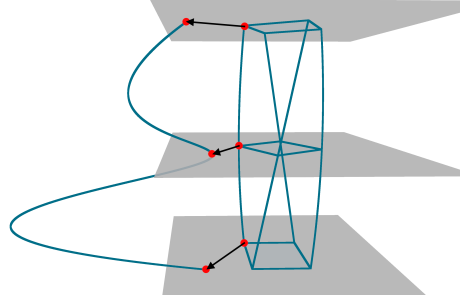


Fig. 4: Calculating displacement vectors with plane intersections.

Issues with this solution:

- The guide curve can have multiple intersections with one plane.
- If the guide curve is short it can happen that the planes won't even intersect the curve.

In the end we chose the intersection-based solution, since the multiple intersections problem doesn't happen regularly in real world use-cases. For the short guide curves, we use linear extension.

2. Calculate an interpolated spline curve from the displacement vectors

After acquiring the displacement vectors, we calculate an interpolated displacement spline, which we can later add to the original guide curve. We use the displacement vectors as points and we want to find a spline that interpolates these points. If the original guide spline is connected to the profiles, we will have 0 length displacement vectors, that can result in degenerate splines, which can cause Parasolid errors. Therefore, we used helper vectors, which would eliminate this problem for most of the cases, arriving at a more stable solution.

1. Calculate the normalized tangent vectors of the guide curve at each intersection point, then multiply it with the parameter of the guide curve at this point.
2. Add these helper vectors to the original displacement vectors. Later when calculating the final connecting guide curve we will subtract these.

$$\vec{v}_i^* = \vec{v}_i + \underbrace{\frac{\frac{\partial}{\partial u} C_{guide}(u_i)}{|\frac{\partial}{\partial u} C_{guide}(u_i)|} \cdot u_i}_{=\vec{h}_i} \quad (2.1)$$

where

$C_{guide}(u)$: original guide curve

\vec{v}_i : displacement vector for the i -th profile

u_i : parameter on the guide curve, for intersection point for \vec{v}_i

3. Calculate the connecting guide curves

We add the original guide curve and the interpolated displacement spline (that we calculated in the previous step) together. Then from the sum, we subtract the helper vectors by calculating an interpolated spline from these helper vectors and subtracting this spline from the sum.

$$C_{guide}^*(u) = C_{guide}(u) + C_{displacement}(u) - C_{helper}(u) \quad (2.2)$$

where

$C_{displacement}$: the interpolated spline from the previously calculated \vec{v}_i^* vectors,

where $C_{displacement}(u_i) = \vec{v}_i^*$

C_{helper} : the interpolated spline from the helper vectors, where $C_{helper}(u_i) = \vec{h}_i$

4. Use distance-based blending in case of multiple guide curves

We use a distance-based blending for the guide curves if multiple curves are given as inputs, see Fig. 5. This means that for each connection point chain, we use multiple guide curves, which are weighted based on the distance from the connection points. The requirement is that if a guide curve has 0 length displacement vectors (which means it is connected to each profile) then it should have a weight of 1, and all other curves should have 0 weights. The used blending functions was based on [10]. Let

$$D_{k0,k1,\dots,kn}^n = \prod_{k \neq k0,k1,\dots,kn} d_k \quad (2.3)$$

where

$$d_k = \sum_{i=0,1,\dots,m} |v_{ik}^{\vec{}}|$$

$v_{ik}^{\vec{}}$: i -th displacement vector for C_{guide}^k

Then the blend function for C_{guide}^k

$$B_k = \frac{D_k^n}{\sum_{j=0}^n D_j^n} \quad (2.4)$$

5. Use the displaced guide curves to loft with Parasolid

Now the curves we got from the methods in 3. and 4. satisfy Parasolid's requirements.

Conclusions:

We introduced a method for a more flexible lofting functionality. The method is based on finding displacement vectors for the guide curves to make them connected to the profiles. We tried several methods for the displacement vector calculation, and chose the plane intersection-based solution. With interpolations we can create displacement splines, and calculate the final, connected guide curves. We also introduced a solution where multiple guide curves can be blended together based on the distances.

The solution gives Shapr3D users a more intuitive loft tool functionality, where they can better capitalize on the strength of history-based modeling as well. The distance-based blending allows them to create even more complex surfaces with multiple guidelines.



Fig. 5: Distance-based blending for multiple guide curves. *First:* Both curves are close, *Second:* Left curve is close, right curve is twice as far, *Third:* Left curve is close, right curve is 3-times as far.

Ákos Horváth, <https://orcid.org/0009-0001-4594-9653>

István Kovács, <https://orcid.org/0000-0002-4902-3559>

References:

- [1] Shapr3D App <https://www.shapr3d.com>
- [2] Shapr3D Help Center - Loft Tool <https://support.shapr3d.com/hc/en-us/articles/7874427572508-Loft>
- [3] Loft(3D)-Wikipedia [https://en.wikipedia.org/wiki/Loft_\(3D\)](https://en.wikipedia.org/wiki/Loft_(3D))
- [4] Parasolid Kernel <https://www.plm.automation.siemens.com/global/en/products/plm-components/parasolid.html>
- [5] Autodesk - Get Smart With Fusion 360 Part 3: The Secrets of the Loft Command <https://www.autodesk.com/products/fusion-360/blog/get-smart-with-fusion-360-loft-command/>
- [6] Onshape help - Loft tool <https://cad.onshape.com/help/Content/loft.htm>
- [7] Chiyokura, H. & Kimura, F. Design of solids with free-form surfaces. *SIGGRAPH Comput. Graph.* **17**, 289-298 (1983,7), <https://doi.org/10.1145/964967.801160>
- [8] Schaefer, S., Warren, J. & Zorin, D. Lofting curve networks using subdivision surfaces. *Proceedings Of The 2004 Eurographics/ACM SIGGRAPH Symposium On Geometry Processing*. pp. 103-114 (2004), <https://doi.org/10.1145/1057432.1057447>
- [9] Deng, C., Huang, J. & Yang, Y. Interactive modeling of lofted shapes from a single image. *Computational Visual Media*. **6** (2019,12)
- [10] Tamas Varady, Alyn Rockwood, Peter Salvi: Transfinite surface interpolation over irregular n-sided domains, *Computer-Aided Design*, Volume 43, Issue 11, November 2011, Pages 1330-1340 <https://doi.org/10.1016/j.cad.2011.08.028>