A Process for Producing 3-D Printable 2-Manifold Trusses in STL Format from Wireframe Graphs

Luis Villafana

In this manuscript, a process for producing 3-D printable STL files of trusses with approximately cylindrical members obtained from 3-D line segment graphs is presented. In the presented process, a polyhedron is formed for each segment of the input 3-D wireframe independently. A non-manifold truss is created by overlapping the polyhedra of each thickened wire, and a 2-manifold truss is created from the Boolean union of the polyhedra from all thickened wires. The non-manifold truss is more practical for 3-D printing purposes because it requires less computing power and is visually identical to the 2-manifold version of the truss, while the triangular mesh of the 2-manifold truss has no self-intersections and can be utilized for finite element simulations. Furthermore, the presented process always produces a 3-D object with no errors due to the nature of the underlying mathematics.

Introduction

Transforming 3-D models of line segment wireframes into solid trusses is a popular practice in additive manufacturing for rapid prototyping due to reduced material costs. Trusses are also relevant in different areas of mechanical engineering and architecture, and their visualization and mechanical simulation are facilitated by computerized mesh representations. The presented method to add volume to 3-D wireframes that can cope with any wireframe geometry, including geometries with intersecting wires, and a polyhedral structure is produced even if the diameter of the beams exceeds a size that enables the resolution of individual beams.

Wireframe Creation

There are many available options to create a wireframe design, either by hand, from measurement specifications, or programmatically for graphs that have more segments than those that can be drawn by hand. For this paper, the open-source software Blender was used to create a free-form wireframe, which was exported as a WaveFront file. The WaveFront file was decomposed into the X, Y, and Z coordinates of the nodes of the wireframe and a vector of the connectivity of the nodes in the wireframe design. The wireframe point coordinate and connectivity data were input into Octave.

Individual Beam Creation

With functions programmed in Octave, each wire in the input wireframe graph was given thickness by defining points on spheres centered at each of its endpoints. The utilization of sphere geometries ensured joints without sharp edges in the resulting model. The points on the endpoint spheres were defined by a method that created approximately equidistant points on the surface of a sphere. The creation of equidistant points on the endpoint spheres maximized the smoothness of the joints of the resulting truss while promoting a minimal number of points to reduce computational complexity. A triangular mesh was created for the surface of each cylindrical beam from the convex hull of the Delaunay triangulation of the points of its endpoint spheres. To promote smoothness, sphere points for each node in the wireframe were repeated from the same point pattern and translated to each endpoint without rotation.

Rotation of the sphere points is avoided because individual cylindrical beams overlap at joints of the wireframe graph, and overlapping triangulated spheres will increase the number of facet intersections if sphere points do not overlap.

Figure 1 illustrates the process of the formation of individual cylindrical beams.

Truss creation

To create a solid truss from a wireframe design, the method of creating individual wireframes was applied to each wire in the wireframe. Octave was used to export a single STL file with facets from all truss members. This STL file was created to produce a 3-D object that could be readily 3-D printed. However, the 3-D object in this file is not a manifold. To create an STL file of a 2-manifold object, Octave was used to export separate STL files of all truss members. An OpenSCAD file was programmatically created with Octave to automatically import all separate truss member STL files into OpenSCAD and perform a Boolean union operation. The result of the Boolean operation was a 2-manifold polyhedron with triangular facets.

The trusses created with Boolean union and polyhedron superposition are compared in figure 2.

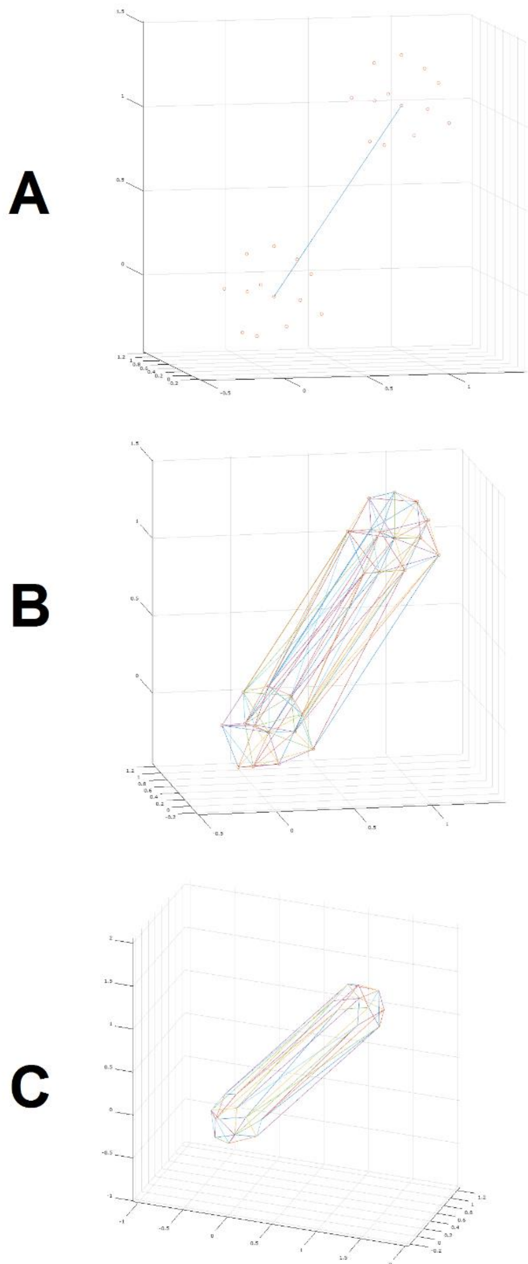

Figure 1.

A: An arbitrary segment is shown with 12 spherical points surrounding each endpoint.

B: Delaunay triangulation of all points. This triangulation creates facets inside of the resulting beam polyhedron.

C: A beam is formed from the convex hull of the triangulation in B. The resulting polyhedron is approximately cylindrical for large numbers of spherical points.

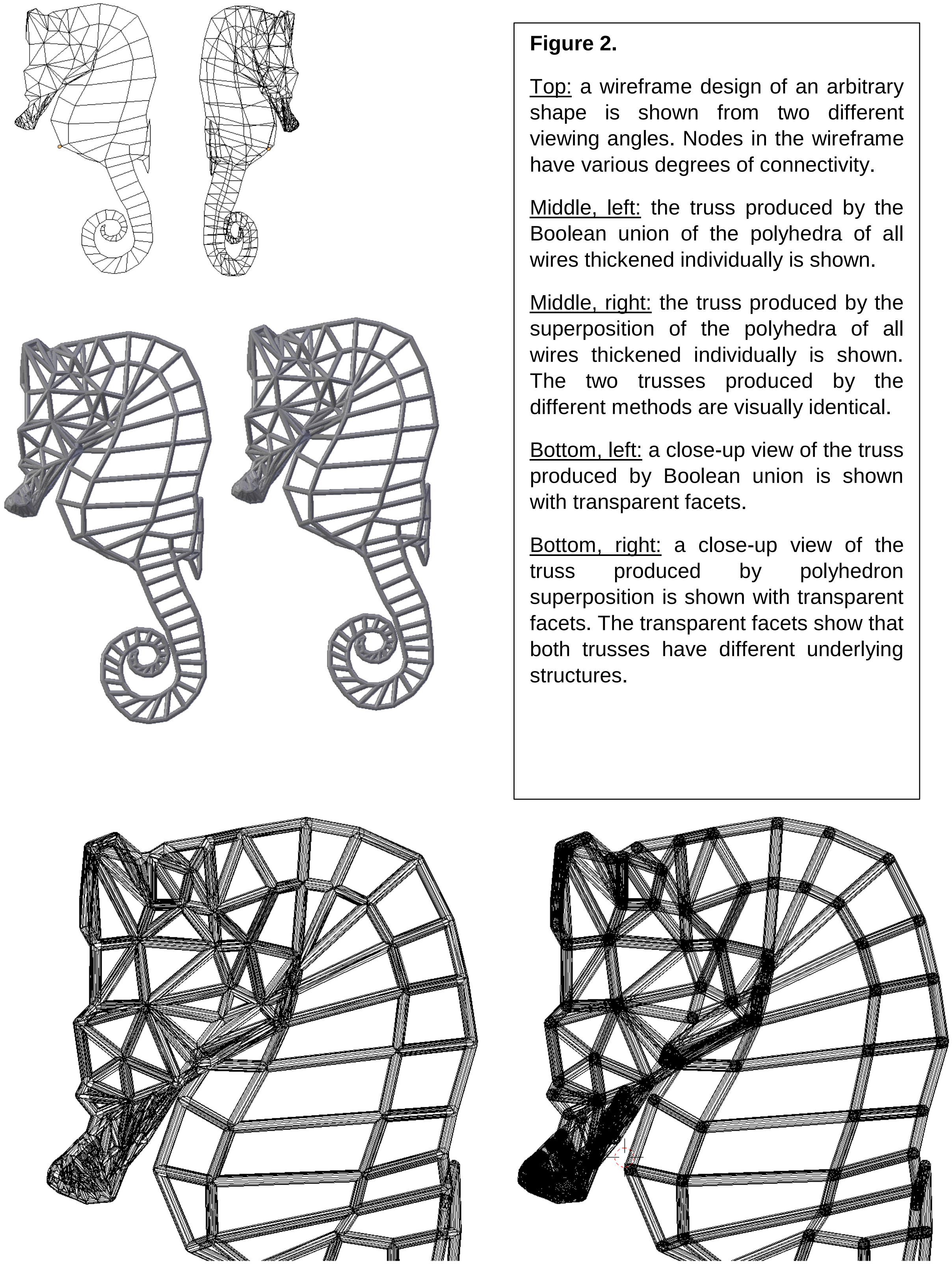

Figure 2.

Top: a wireframe design of an arbitrary shape is shown from two different viewing angles. Nodes in the wireframe have various degrees of connectivity.

Middle, left: the truss produced by the Boolean union of the polyhedra of all wires thickened individually is shown.

Middle, right: the truss produced by the superposition of the polyhedra of all wires thickened individually is shown. The two trusses produced by the different methods are visually identical.

Bottom, left: a close-up view of the truss produced by Boolean union is shown with transparent facets.

Bottom, right: a close-up view of the truss produced by polyhedron superposition is shown with transparent facets. The transparent facets show that both trusses have different underlying structures.