Group 16: Browser-based Constructive Solid Geometry for Anatomical Models

Team Members: Vikram Chandrashekhar and Nicole Ortega
Mentors: Alex Mathews, Param Shah

Goal The goal of our project is to develop a *browser-based* constructive solid geometry application for the efficient creation of 3D anatomical models, specifically the creation of a modular orthosis.

Motivation and Background One in 323 children in the US are born with cerebral palsy. Two out of every three children born with cerebral palsy could walk if they had proper orthotics to alleviate their condition. Currently, ankle foot orthotics are used to correct gait and prevent muscle deformities. The current process for making custom ankle foot orthotics is tedious and wasteful. The process begins by creating a hard cast around the foot of interest. The cast in then filled to create a mold of the foot. Once the mold hardens, the cast is removed and the orthotic is created and cut around the mold. Ultimately, the cast, mold, and scraps of material from the orthotic are thrown away. The cost to make a custom orthotic ranges from \$400 - \$600 (compared to \$10 - \$80 for off the shelf orthotics) and can take up to 3 weeks. Checkups every 6 months are required for adjustments. On average, these orthotics are replaced every six to eight months, which involves repeating the tedious and wasteful casting process.

Fusiform, a medical device group, has developed a process to reduce waste, reduce time, and increase efficiency of attaining a custom orthotic. The Fusiform process begins by using a structure sensor mounted on an iPad to take a 3D anatomical scan of the leg to a 1mm accuracy. The orthotic is then created on Solidworks using the scan and fabricated by CNC machine. The Fusiform process reduces waste by no longer requiring the creation of a mold and taking advantage of the CNC machine's subtractive production. Additionally, their modular orthotic design allows patients to upgrade individual parts, and thus produces less waste.

However, the Fusiform process has room for improvement. On average, it takes 10 hours to layer an orthotic on Solidworks. The goal of our project is to develop a browser-based constructive solid geometry application for the efficient creation of 3D anatomical models, specifically the creation of a modular orthosis. This approach is unique in that no other browser-based interface like this exists nor has it been applied to anatomical models.

Technical Approach

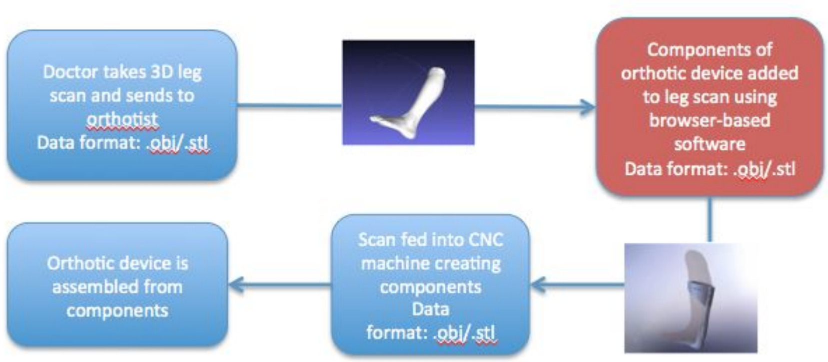


Figure 1. Workflow for clinical use. The red box is our contribution to the workflow.

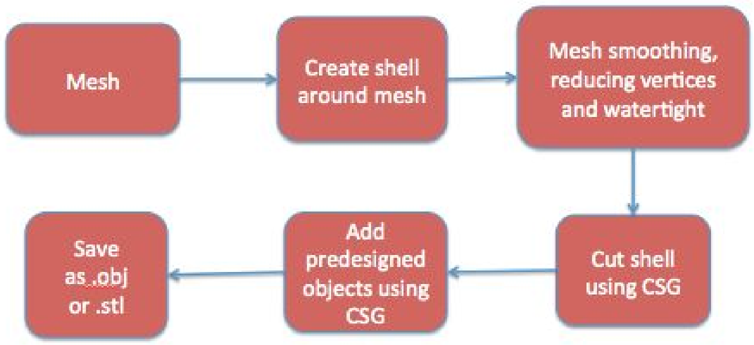


Figure 2. Workflow for our software.

Input	Function	Output
2 (or more) meshes	CSG.union	Single combined mesh
2 meshes	CSG.subtract	Single mesh with the volume of the second mesh eliminated
2 meshes	CSG.intersect	Single mesh made up of the volume of the overlap between the two meshes
1 mesh	smooth	Eliminates sharper edges on mesh
1 mesh, decimation factor	decimate	Reduces number of vertices in mesh by factor
1 mesh	watertight	Eliminates holes in mesh

Figure 3. Software architecture with general overview of functions

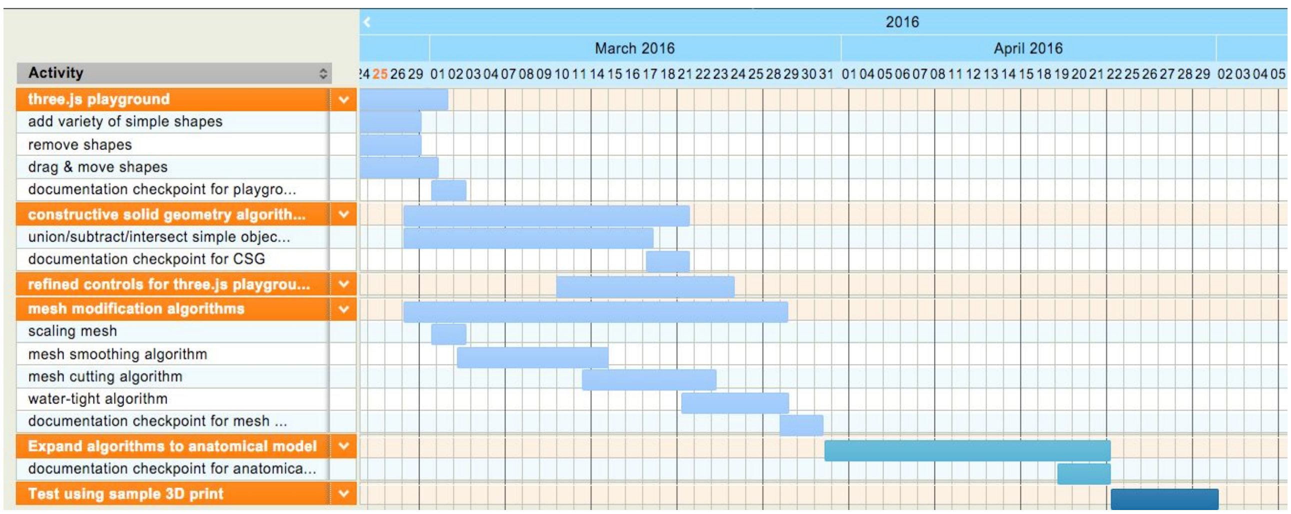
Deliverables All minimum and expected deliverables will be done in a browser-based environment.

Minimum	-Implement constructive solid geometry algorithms for simple objects (sphere, cube, prism, etc) using three.js in a browser -Implement mesh cutting, smoothing, reducing and watertight modification algorithms on simple objects
Expected	-Expand above algorithms to anatomical objects, particularly 3D leg scans -Implement mesh cutting, smoothing, reducing and watertight modification algorithms on anatomical models
Maximum	-Using browser-based software, test cast fabrication using a 3D printer and test "fits" on patients

Dependencies

Dependency	Resolution
Three.js - Javascript software package that interfaces with WebGL to perform 3D rendering	open-source software available - DONE
Blender/CAD - software to perform constructive solid geometry and mesh modification algorithms	Available through Johns Hopkins using education license - expected by 3/1
Object (.obj/.stl) files of anatomical leg scans	Will be provided by mentors - expected by 3/15
3D Printer to test print casts	Will be provided by mentors - if needed by 4/17

Timeline/Milestones



Management Plan To enable version control, BitButcket will be used to create a private repository. We will use Slack, a messaging application for teams, to communicate with our mentors and share files. We have scheduled weekly team meetings for Wednesdays at 5pm and Mondays at 6pm. Additionally, we scheduled weekly meetings with our mentors for Mondays at 5pm.

In addition, we will take charge of different aspects of the project. Vikram will lead the creation of a three.js playground and constructive solid geometry for simple objects in browser. Meanwhile, Nicole will lead the implementation of mesh modification algorithms, including the mesh smoothing, reducing and watertight algorithms in browser. Ultimately, we will work together on anatomical mesh modification using constructive solid geometry and mesh cutting, reducing, smoothing and watertight algorithms applied to the mesh.

Reading List

1. Amenta, Nina, Marshall Bern, and David Eppstein. "Optimal Point Placement for Mesh Smoothing." *Journal of Algorithms* 30.2 (1999): 302-22. Web.
2. Dey, T. K., & Goswami, S. (2003). Tight Cocone: A Water-tight Surface Reconstructor. *Journal of Computing and Information Science in Engineering J. Comput. Inf. Sci. Eng., 3*(4), 302.

3. Fan, Lubin, Ligang Lic, and Kun Liu. "Paint Mesh Cutting." *Computer Graphics Forum* 30.2 (2011): 603-12. Web.
4. Gregori, R. M., Volpato, N., Minetto, R., & Silva, M. V. (2014). Slicing Triangle Meshes: An Asymptotically Optimal Algorithm. *2014 14th International Conference on Computational Science and Its Applications*.
5. Ji, Zhongping, Ligang Liu, Zhonggui Chen, and Guojin Wang. "Easy Mesh Cutting." *Computer Graphics Forum* 25.3 (2006): 283-91. Web.
6. Plato, Jan Von. "The Axioms of Constructive Geometry." *Annals of Pure and Applied Logic* 76.2 (1995): 169-200. Web.
7. Ricci, A. "A Constructive Geometry for Computer Graphics." *Computer-Aided Design* 6.1 (1974): 53. Web.