Browser Based Constructive Solid Geometry for Anatomical Models

Nicole Ortega and Vikram Chandrashekhar

Team Members

- Vikram Chandrashekhar, Biomedical Engineering 2016
- Nicole Ortega, Biomedical Engineering 2016

Mentors

- Alex Mathews, Fusiform Medical Devices
- Param Shah, Fusiform Medical Devices

Skills

Vikram

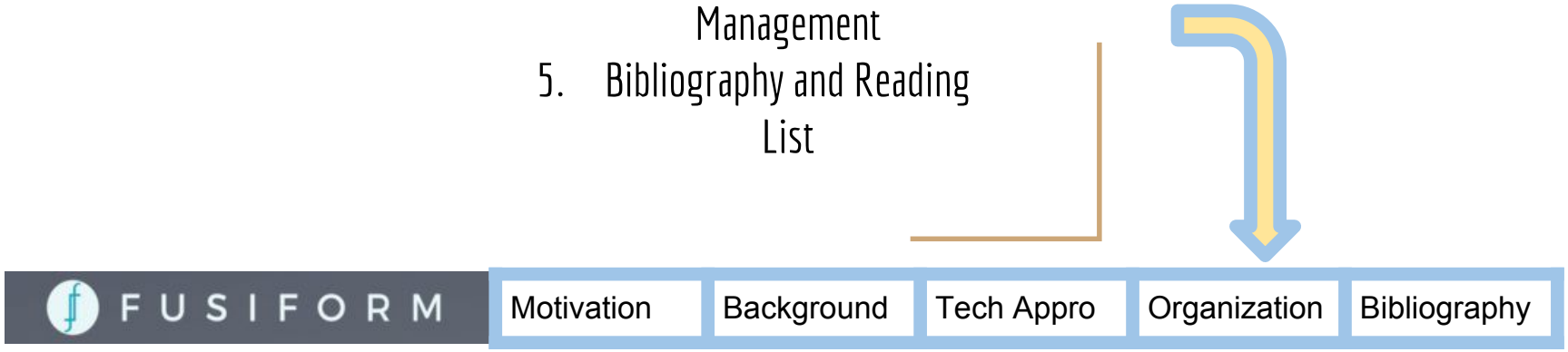
- Programming: Javascript (three.js), HTML, C, C++

Nicole

- Programming: C, C++, Java

Overview

1. Motivation
2. Background
3. Technical Approach
4. Organization and Management
5. Bibliography and Reading List



Motivation

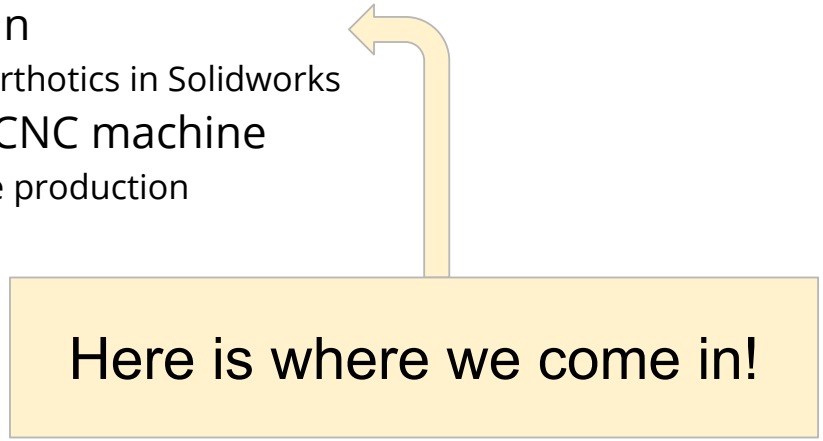
- 1 in 323 children are born with cerebral palsy in US
- 2 in 3 could walk if they had proper orthotics
- Ankle foot orthotics
  - Corrects gait and prevents deformities

Problems with Current Process

- Tedious and wasteful casting process
  - Mold→ fill mold → send to manufacturer → create orthotic → throw away remaining material and mold
- Replaced every 6-8 months on average
- ~ 3 weeks for custom device
- 6 month checkups - adjustments, new one
- Non-reusable (waste material)
- Cost
  - off the shelf: $10 -80
  - custom: $400 - 600

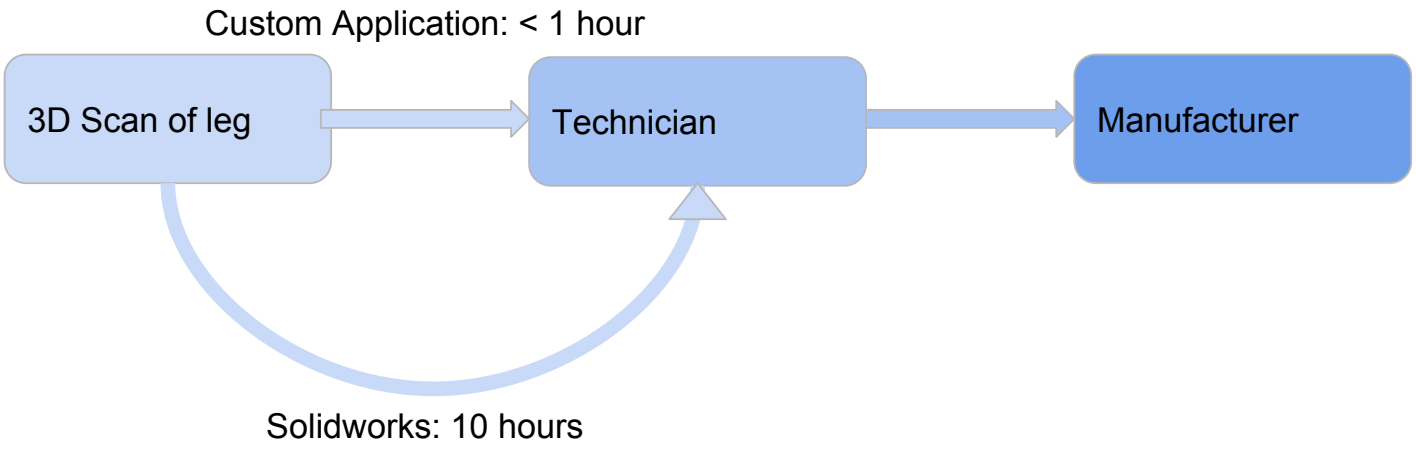
Fusiform Process

- Anatomical Scan of leg using Structure Sensor
  - iPad mounted 3D scanner, with our app to take anatomical scans to 1mm accuracy
- Create orthotic using scan
  - 10 hour process to layer orthotics in Solidworks
- Fabricate orthotic using CNC machine
  - CNC machine - subtractive production
  - Interchangeable parts
  - Less waste



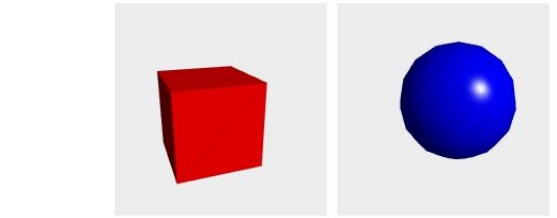
Our Role

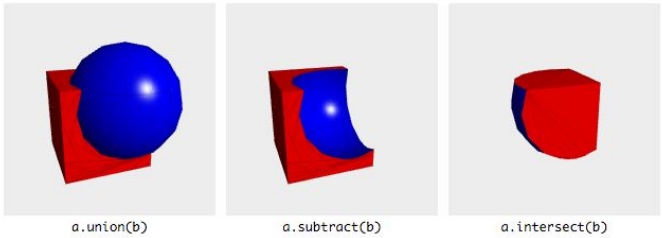
- Reduce the amount of time required to design and fit 3D cast



Technical Approach

- three.js playground (http://bit.ly/1QYe9HJ)
  - Develop Javascript base to add/delete simple shapes
- Constructive solid geometry
  - Combining objects/meshes using boolean operators
- Mesh modification
  - scaling
  - mesh cutting
  - defining multiple cuts
  - water-tight algorithms
- Anatomical model shell
  - 1 cm of material on top of anatomical model
  - mesh cutting → add components of cast → merge/modify cast
- Validation in third party application (like MeshLab or SolidWorks)





Deliverables

Minimum	-Implement constructive solid geometry algorithms for simple objects (sphere, cube, prism, etc)
Expected	-Expand above algorithms to anatomical objects, particularly 3D leg scans -Implement mesh modification algorithms to make mesh easier to work with (smoothing, cutting, reducing) -Implement an algorithm to create a planar cut in a 3D mesh
Maximum	-Using browser-based software, test cast fabrication using a 3D printer and test “fits” on patients

Dependencies

Minimum/Expected:

- Three.js - 3D Javascript library used to interface with WebGL
  - Open Source library
- Blender/CAD to verify algorithms' correctness
  - Mentors
- Anatomical scans of legs using iPad mounted scanner
  - Mentors

Maximum:

- 3D Printer to create the cast
  - Mentors

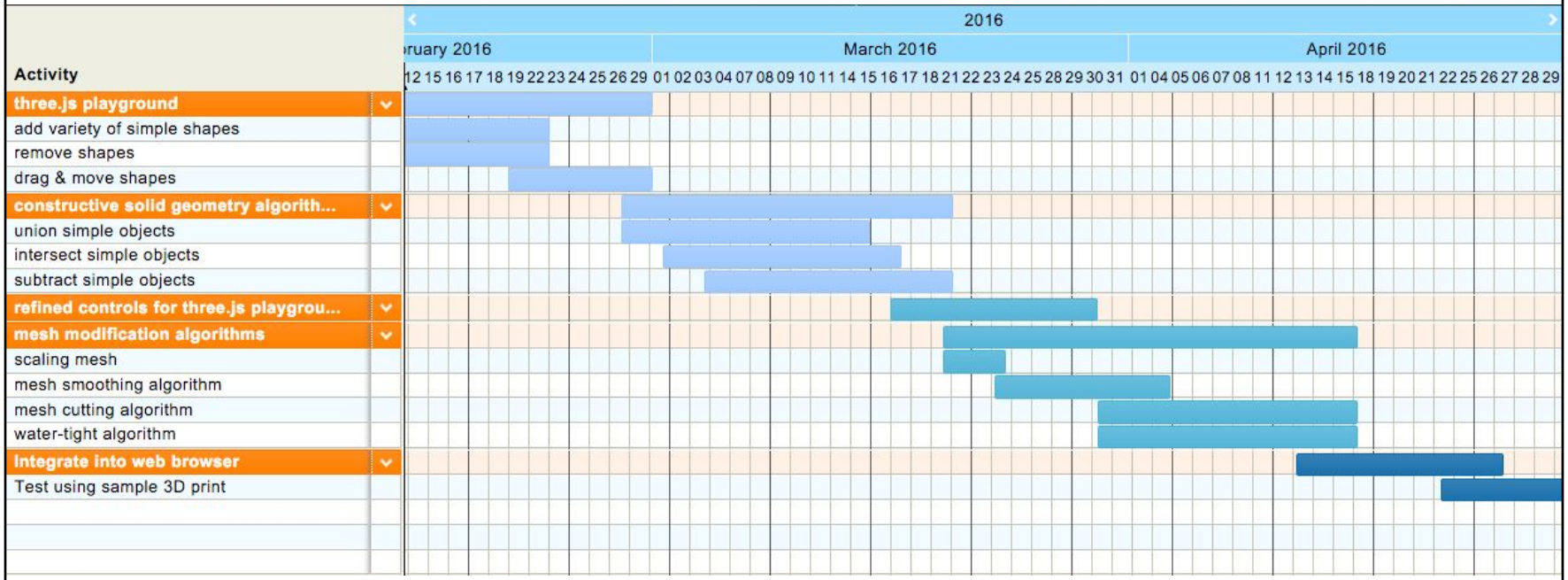
Management Plan

- BitBucket: Version Control (private repository)
- Slack: Keep track of milestones on timeline
- Weekly team meeting: Mondays @ 5 pm & Wednesdays @ 6 pm
- Weekly meeting with mentors: Wednesdays @ 5 pm

Management Plan

Vikram	Nicole
Lead creating three.js playground	Become familiar with three.js and mesh modification algorithms
work together on mesh modification implementation in Javascript	
work together on anatomical mesh modification	

Milestones


2016
ruary 2016
March 2016
April 2016
Activity
12 15 16 17 18 19 22 23 24 25 26 29 01 02 03 04 07 08 09 10 11 14 15 16 17 18 21 22 23 24 25 28 29 30 31 01 04 05 06 07 08 11 12 13 14 15 18 19 20 21 22 25 26 27 28 29
three.js playground
add variety of simple shapes
remove shapes
drag & move shapes
constructive solid geometry algorith...
union simple objects
intersect simple objects
subtract simple objects
refined controls for three.js playgrou...
mesh modification algorithms
scaling mesh
mesh smoothing algorithm
mesh cutting algorithm
water-tight algorithm
integrate into web browser
Test using sample 3D print

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.