Augmented Reality for Bone Visualization in Orthopedic Surgery

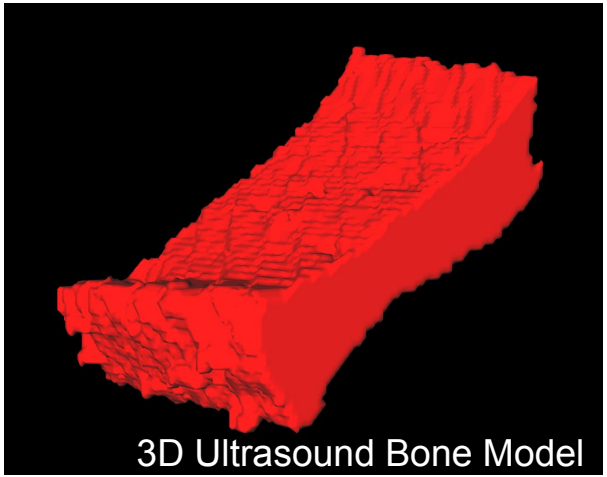
Davis Rempe (University of Nebraska) and
Josiah Smith (Georgia Tech)
Advisor: Dr. Brian Chen

Problem: Orthopedic Surgery is Blind

- No capable real-time imaging
  - High resolution MRI
  - Real-time ultrasound
- Cannot see these data during surgery
  - We try to solve this
- Image alignment

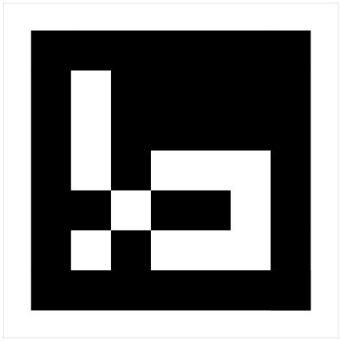
Project Goal

- Visualize 3D bone models in correct location through augmented reality (AR)



Outline

- ~~Background~~
- Tools
- Methods Implementation/Algorithms
- Demonstration
- Future Work



Mobile AR

- Microsoft HoloLens
  - Expensive
  - Just released
- Google Cardboard
  - Created for virtual reality (VR)
  - Inexpensive
  - Good for proof of concept
    - More powerful AR devices can handle it if phone can

Google Cardboard

- Android
- **Many apps use AR, few render in stereo for head-mounted display**
  - CMoar RPG
  - Vuforia Library
- Free Google API for Cardboard
  - No one has used it for AR

vuforia™

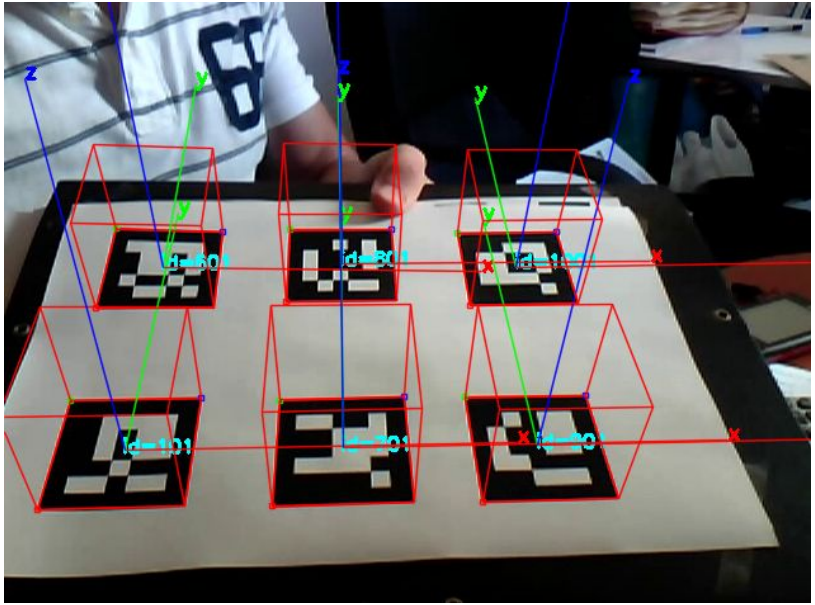
Cardboard

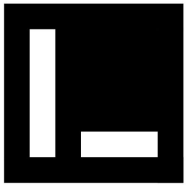
Possible with Software Libraries

- Marker Tracking
  - OpenCV
  - Aruco
- Graphics/Rendering
  - Google VR SDK (GVR)
  - OpenGL ES
- Bone model visualization easy with these capabilities
- **New goal: Cardboard AR library**

Tracking Markers

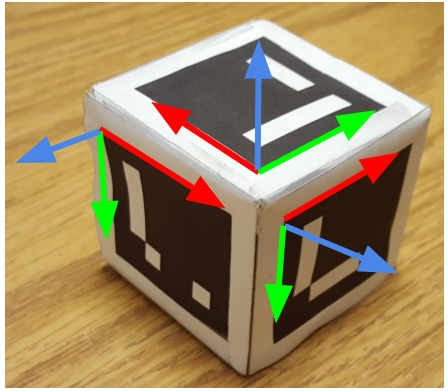
- What is tracking marker?
  - Unique pattern
  - High contrast
- Aruco library
  - Individual ID’s
  - Well-tested tracking
  - Variable marker size

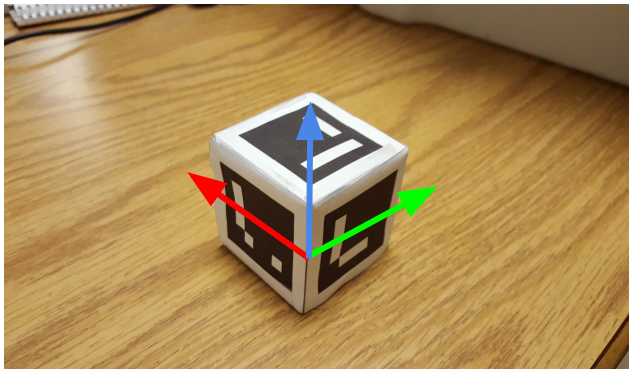


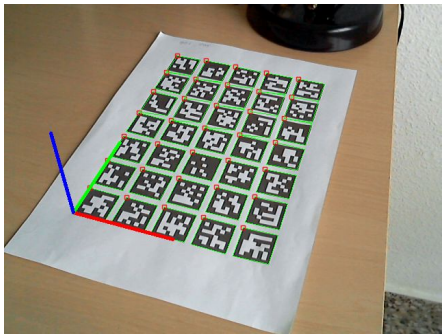


Robust Tracking

- Single marker limits tracking
- Multi-marker tracking
  - Board of markers
  - More accurate, but still limits angle
- Marker Cube
  - All angles, more accurate
  - Average detected cube faces

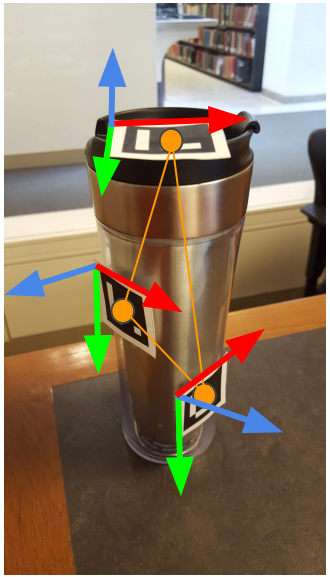


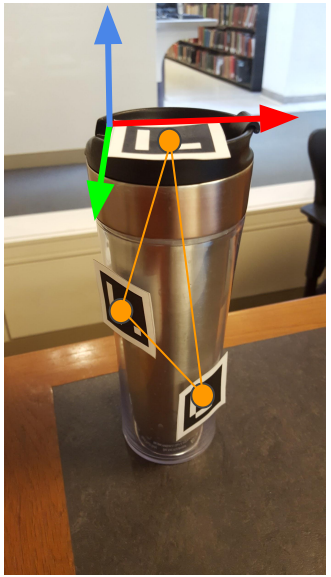


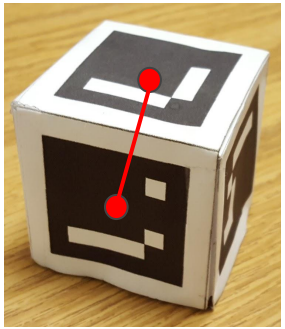


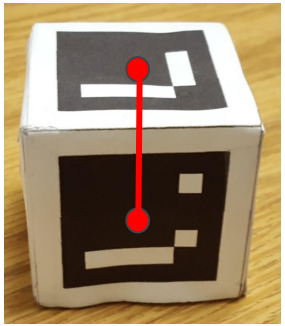
Rigid Body Detection

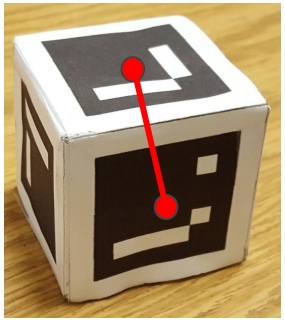
- Generalize cube detection to any rigid body
- Unknown markers and positions
- Two phases:
  - Formation
    - Strong correlation considered rigid
  - Tracking
    - **Any** visible marker provides position/pose

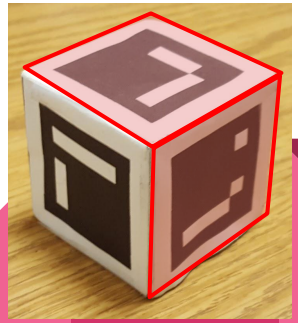






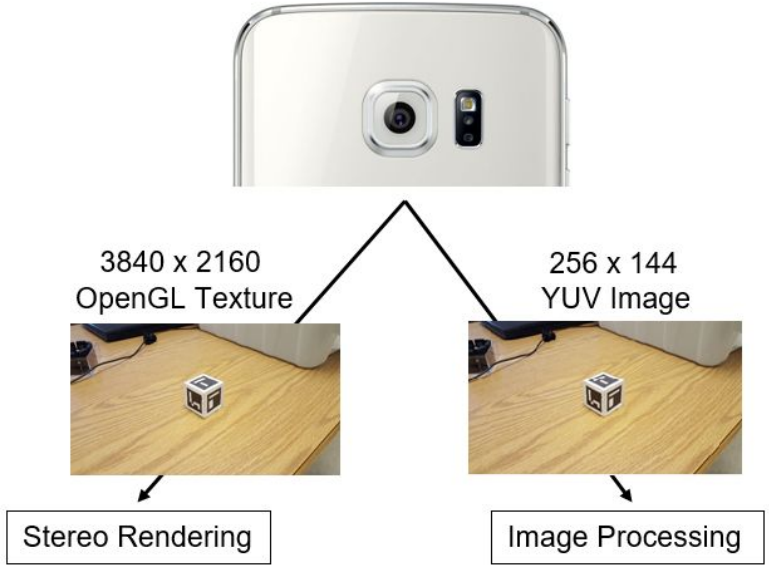






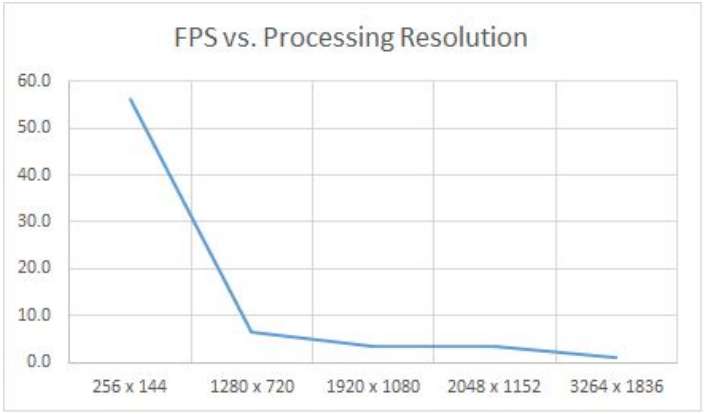
Getting the Camera Image

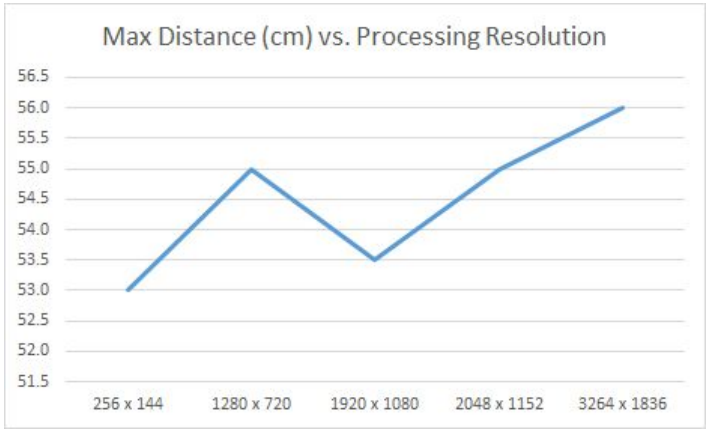
- Capture image with Android SDK
- Two “Surfaces”:
  - Texture for stereo rendering
  - Image to process
- Initially requested at same rate
  - Processing too slow
- Now request at different rates



Determining Processing Resolution

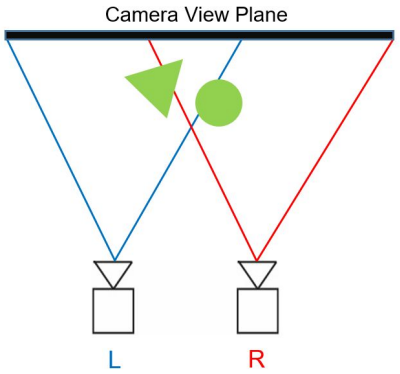
- Ran tests to determine performance
  - Samsung Galaxy S6 (Android 6.0.1), 35 mm markers with 5 mm padding

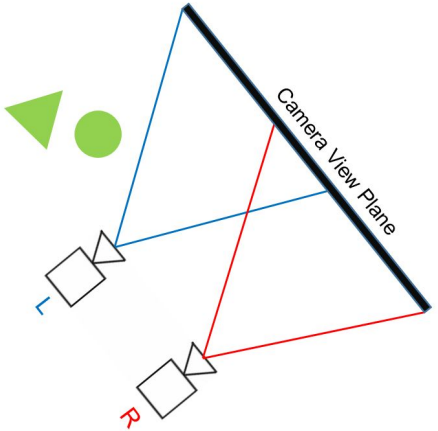


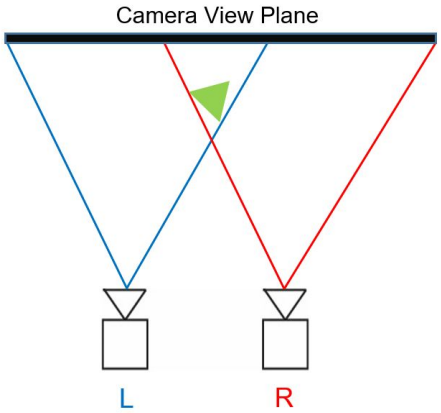


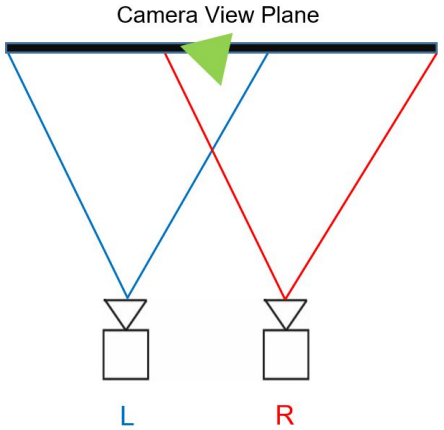
Stereo Rendering

- Single 3D scene
  - Rendered twice (left and right)
- Phone camera view plane
- 3D objects in front of plane
- Problems:
  - Parallax
  - Focus
- Move and scale causes collision



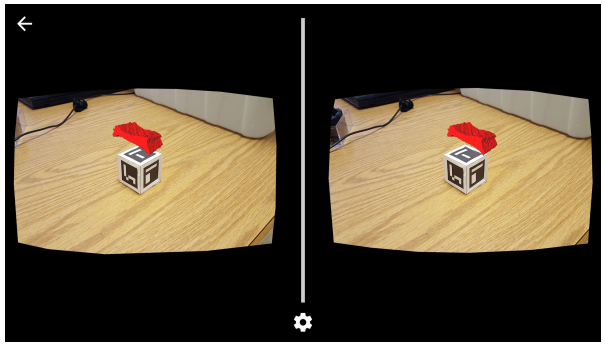




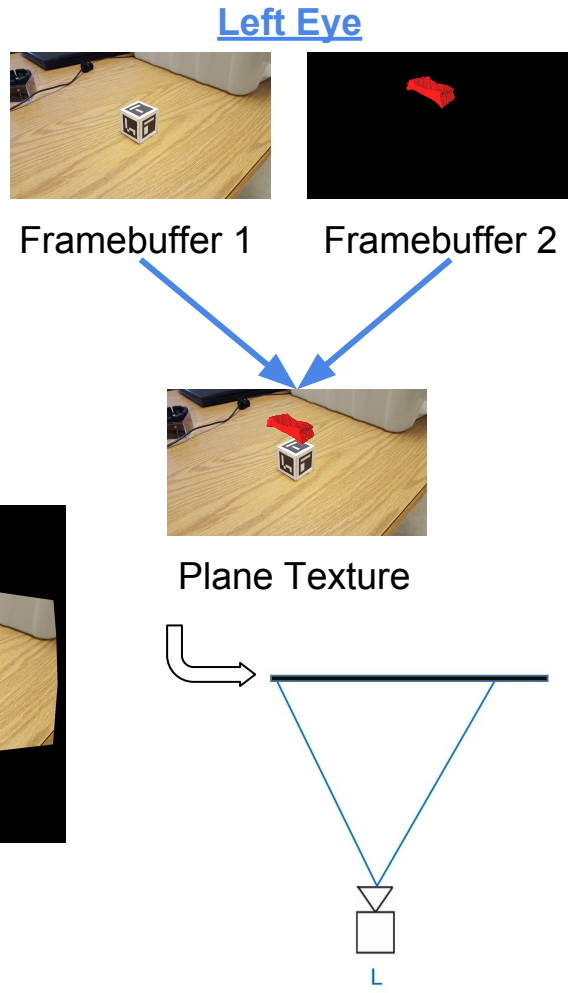


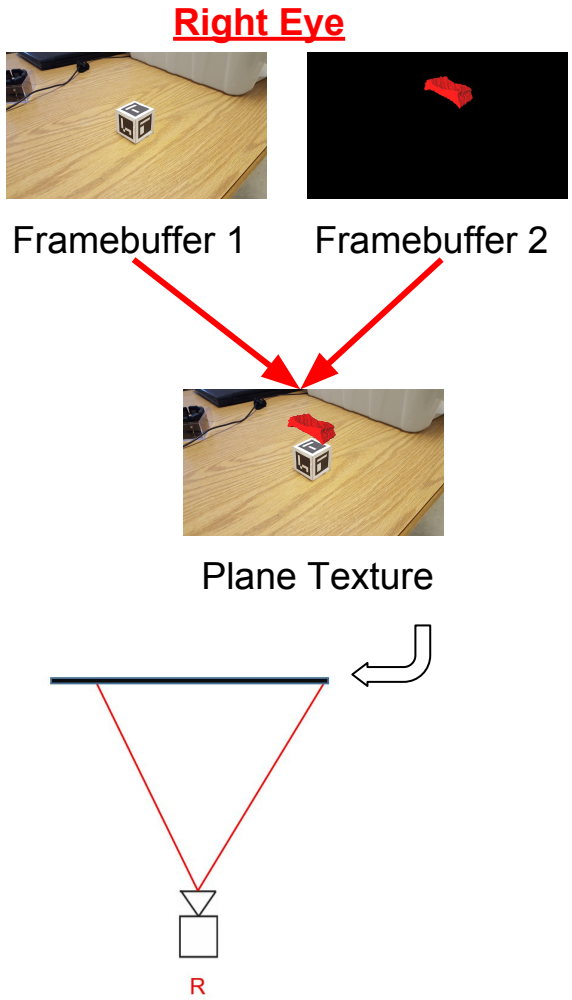
Framebuffers

- Render separately
- Combine to create texture



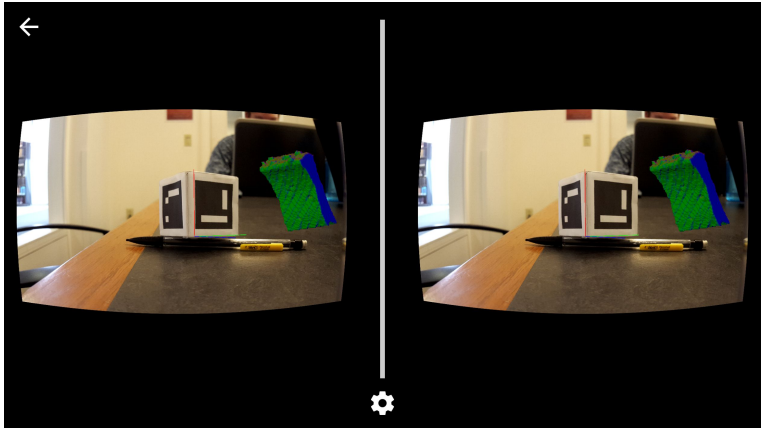
Stereo Cardboard Display





Software Product

- Open-source library
  - https://github.com/davrempe/cardboardAR-lib
- Many applications
  - Surgery, education, gaming, etc.
- Prototype of bone model visualization
  - Marker cube
  - Renders axes and ultrasound bone model
- Demo



Future Work

- Current implementation:
  - Jittery/inaccurate tracking
  - Performance testing (max angle of detection)
  - Overheating
- Improvements:
  - Rigid-body detection only minimum viable
  - Optimize marker detection
    - Tracking based on previous frame
  - Lacks computer vision capabilities
    - Occlusion detection, markerless object detection, etc.

Questions