Lab 4: AR Concepts

CS410/510: VR/AR Development

Ehsan Aryafar earyafar@pdx.edu

Sam Shippey sshippey@pdx.edu

Refresher: What is AR?

- “Augmented Reality”
- Distinct from “virtual” reality (VR) and part of “extended” reality (XR)
- Very general definition
- Features “rich” interaction with the user’s environment
- Still somewhat an emerging tech
- Pulls from many different fields of knowledge, from AI to game development.

The “reality” part

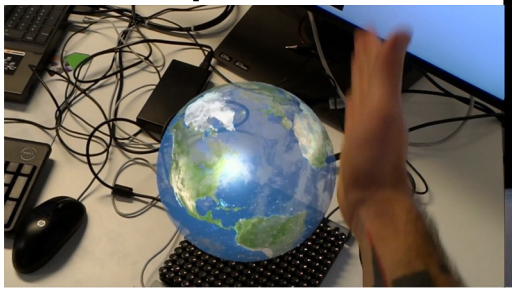
- Distance estimation
- Figuring out orientation of a surface
- QR Code interaction
- Face detection and recognition
- GIS (Geographic Information System) data integration
- Locating pedestrians in real time video

The “augmented” part

- Modifying or super-imposing the environment
- Examples
  - Superimposing a 3d model of a giant dancing hot dog onto real time video
  - Adding cat ears to someone’s face
  - Showing players which pokemon they should cross train tracks to catch
  - Pointing a robot where to go in a physical environment

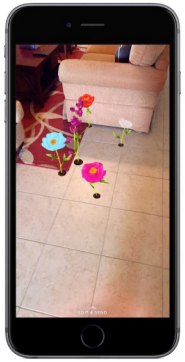
Some AR focused devices

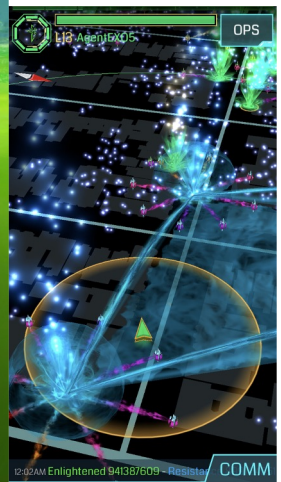
- Headsets are really expensive
- Microsoft Hololens, Hololens 2
  - Limited availability, $3500
- Varjo XR-1 Augmented
  - $9995
- Other headsets may try to catch up
- Most users will just use a phone.



Some AR Apps

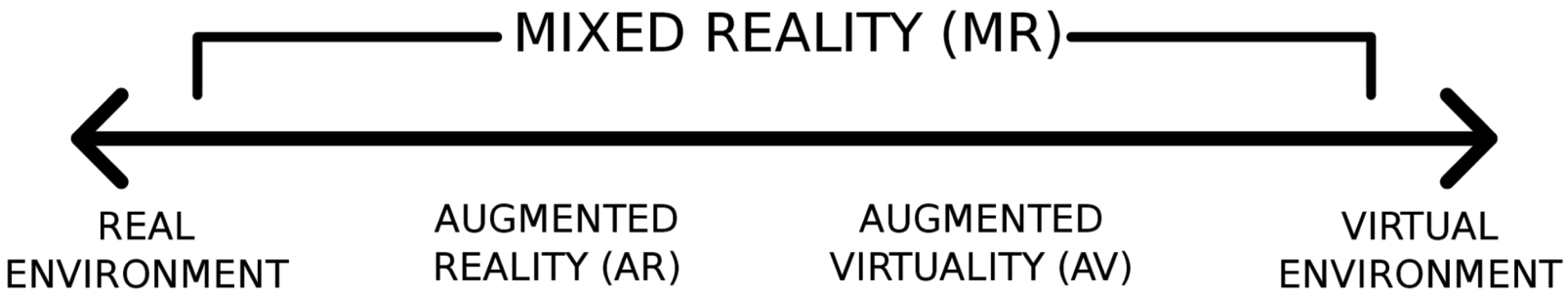
- Snapchat
- Ingress
  - (first release: 2013)
- Pokemon Go
  - 2016
- All AR apps, but make use of several kinds of augmented reality.





The mixed reality spectrum

- What's "virtual reality" and what's "augmented reality"?
- Less of a hard division, more of a continuum.
- This continuum is called the "mixed reality spectrum" or the "reality-virtuality continuum".



VR Intersection

- Oculus quest hand tracking
  - Computer vision techniques with direct AR application
- Inside-out tracking
  - This is part of “Environmental understanding”
- Controller and HMD tracking
  - Use special objects to achieve the same thing that hand-tracking does, but more reliably.
- Passthrough
  - Video of the real world while in VR space
  - Often grainy and low-quality, activated automatically to prevent players from falling down stairs and punching monitors.

Data we can use

- Camera input
  - Very common, flagship product for AR backends
  - Reliable: Almost every phone has 2 cameras
  - Heterogeneous: Almost every phone has its own kind of camera

The most basic AR feature: Face and eye detection.

Data we can use

- Audio input
  - Transcription, voice commands, etc
- Location information
  - Easy(-ish) to implement, engages users
  - Pokemon
- Normal peripheral devices
  - Keyboard, mouse, screen taps, etc

Limits and problems

- Computer vision is expensive
  - Even simple classifiers can run into performance issues if not tuned properly
  - More complicated classifiers run into performance issues even when properly tuned and run on GPUs!
- Rendering in real time is also expensive
- All of these inputs eat battery life
- Location information not always available

Marker-based AR

- Specific to visual AR
- Track an object within scene
- Doesn’t have to be an image, but that’s the easiest and least likely to turn the phone into a space heater!
- Examples
  - QR codes
  - Look at a sign, pull up information about that sign
  - Render a 3d model on top of a card depicting that model

Good markers

- A *specific* image
  - Images which are too general have false positives (if other things would look like them).
- A decent quality image (not blurred)
- Something with easy to recognize lines
- We care because in Lab 5 we'll make use of marker-based AR
- Marker-based AR with very general markers
  - Apps that leverage facial recognition
  - Hand-tracking

Markerless AR

- Very broad set of features
- Does not rely on a marker: Figures out surroundings independently in 6dof (e.g., may use GPS data).
- “Environmental understanding”
  - Object detection and classification
  - What does the room look like?
  - What’s in the room?
  - Not necessarily all of these at once, but some
- Probably what you think of when you think about AR

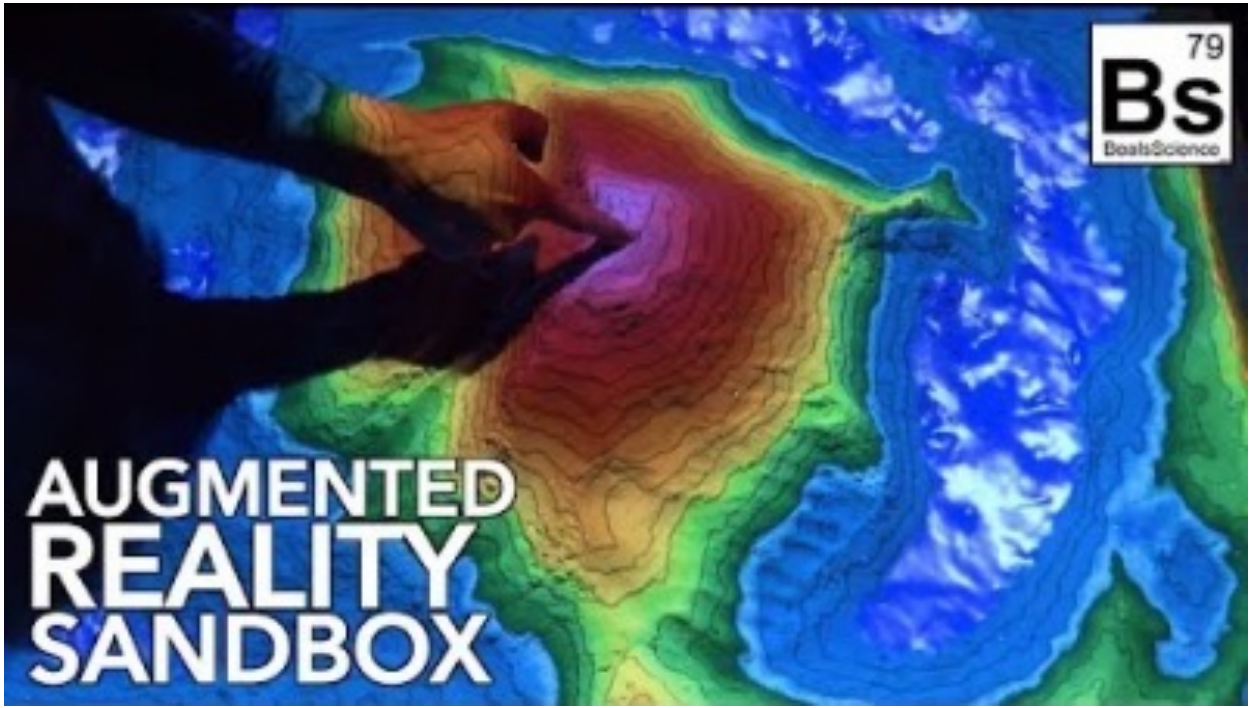
Markerless AR: Location-Based AR

- Location-based AR ties augmentation to a specific place
- The reliability of marker-less AR on positional information
- Examples:
  - Pokemon Go

Markerless AR: Projection-Based AR

- Projection AR, sometimes also referred to as spatial AR, is a method of delivering digital information within a stationary context.
- Light is projected onto a surface

Example: AR Sandbox



Markerless AR: Superimposition-Based AR

- Partial or full replacement of an original view of an object with an augmented view of the same object.
- Object recognition plays a vital role.

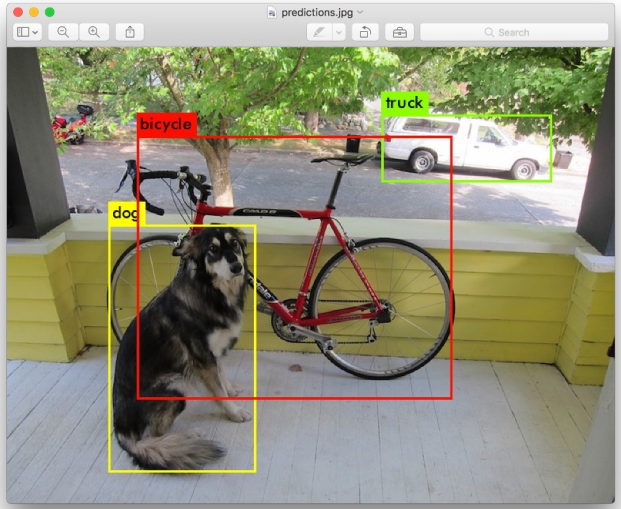
Example: IKEA Place (Released 2017)

How markers work

- Many, many ways to make markers work
- Simplest: Locate an object and draw something over it
- Requirements:
  - As low power as possible
  - Work across many resolutions (“scale invariant”)
  - Marker shouldn’t be that complicated
  - Remember, phones aren’t very powerful

Image classification

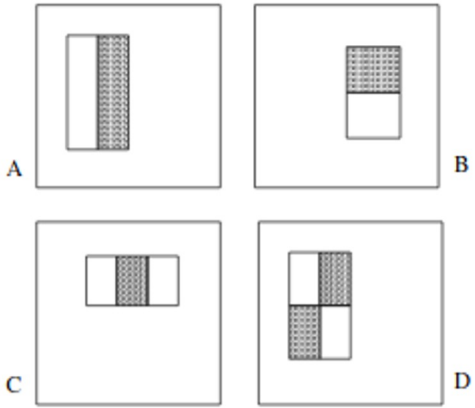
- Take an image in, spit out a classification and possibly a bounding box.
- Example solutions
  - Cascading classifiers
  - Convolutional neural networks
- Limitations
  - Cascading classifiers have trouble with rotations.
  - Inference time for neural networks can take too long.



Example of a bounding box, taken from darknet's YOLO classifier. This takes ~15s to generate on an integrated graphics card of a laptop.

Cascading classifiers

- First brought to attention in 2001
- Designed specifically for low-power devices
- Look for certain vague “features” in sections of an image
- Haar features
- LBP features



Example Haar features, Viola and Jones 2001

Image classification

- Not impossible to do in real time, just hard
- Advances in computer vision and hyperparameter tuning have gotten inference time into acceptable ranges in the last decade or so
  - Many techniques don't rely on improving recognition, but just making recognition easier
  - Ex. Filters that remove uninteresting information and simplify images before passing them to a classifier
- Still limited in some regards especially on older phones.

SLAM

- Simultaneous Location And Mapping
  - Can help with orientation detection as well
- Studied previously for robotics and self-driving car applications
- Depth mapping
- Environmental understanding
- Spatial anchors: Positions in space that many clients are tracking and remain consistent throughout a session.

Back to markers

- Find objects that we care about in scene (Classification)
- Determine their orientation (SLAM)
- Do something with them
  - Render a model/video
  - Interact with a server
  - Play a sound
  - Open a webpage
  - Etc

ARCore

- Google's AR library
- Best support tends to be in Java
- Features listed to include:
  - Environmental understanding via feature tracking
  - Motion tracking
  - Depth mapping
  - Light estimation
  - Marker detection
  - Anchors

ARKit

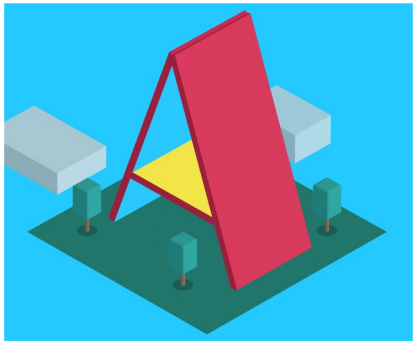
- Apple's ARCore
- Basically the same set of features
- Support and examples tend to be in Swift
- Emerging tech: LiDAR for depth estimation

AR APIs and SDKs

- You have choices and a lot of them cost money
- Vuforia: Can be used for free, also includes paid solutions.
- Unity: Both through Vuforia and several other plugins, including one for markerless AR (AR Foundation)
- Directly using ARCore/ARKit is free, but not that great an experience unless you want cutting edge features.
- WebAR
- Many, many more backends.

AR for the Web

- Again, in case you like web development
- WebAR
  - Three.js and AR.js
  - AWS Sumerian
  - A-Frame
- AR for the web can help provide users with a seamless experience.



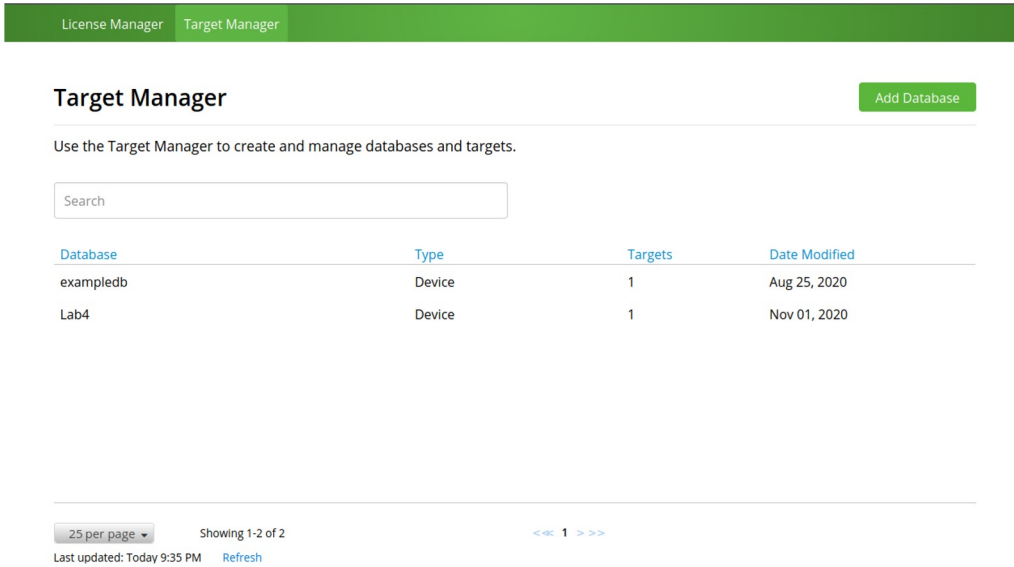
three.js

Vuforia

- What
  - AR platform
  - Includes a “target platform” that we can use to make AR markers
  - Provides most of the heavy lifting for computer vision
- Why
  - User-friendly
  - Simple to set up
  - Good Unity integration

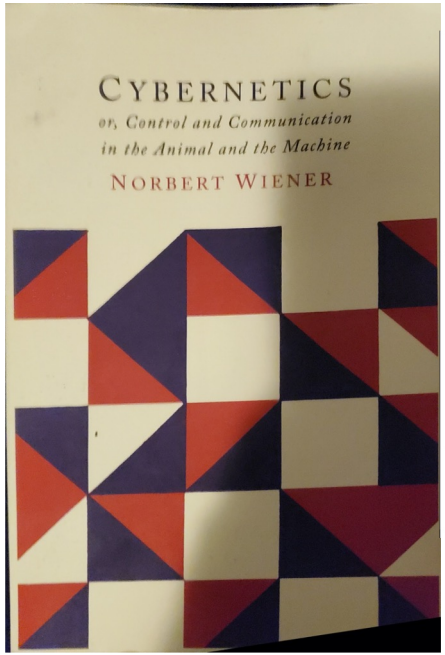
Vuforia setup (For Lab 5)

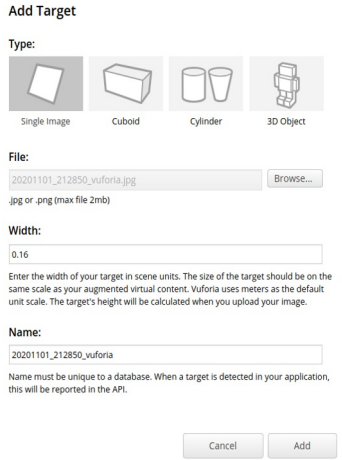
- Go to https://developer.vuforia.com/
- Make an account
- Go to “Develop”
- Select “Target manager”



Vuforia set (continued)

- Find a suitable image with lots of edges
  - If you don't have access to a printer, look for a book
- Add a database
- Add target to database
- Set size to roughly match up with the real object (0.16m in this example)





Lab 5 overview

- Vuforia + Unity
- Render a few cubes onto a book
- Should be simple