

Choo, Ley Yen (lchoo)

- 1 [Week 21 Oct - 27 Oct 2018](#)
 - 1.1 [IMMERSIVE VIRTUAL ENVIRONMENTS: EXPERIMENTS ON IMPACTING DESIGN AND HUMAN BUILDING INTERACTION](#)
 - 1.2 [link: http://papers.cumincad.org/data/works/att/caadria2014_161.content.pdf](http://papers.cumincad.org/data/works/att/caadria2014_161.content.pdf)
 - 1.3 [Conclusion: no differences in either environment](#)
- 2 [Week 30 Oct - 6 Nov 2018](#)
 - 2.1 [Configure the Unity settings according to the documentation](#)
 - 2.1.1 [Tasks: Unity Object](#)
 - 2.1.2 [Adding cursor](#)
 - 2.1.3 [Goal : To let the users know where they are gazing at](#)
 - 2.1.4 [Adding Gestures](#)
 - 2.1.4.1 [Troubleshoot](#)
- 3 [Week 7 Nov 2018 - 14 Nov 2018](#)
 - 3.1 [Spatial Mapping](#)
 - 3.1.1 [Windows Device portal via browser](#)
 - 3.1.1.1 [Video Demo](#)
 - 3.2 [Spatial processing](#)
 - 3.2.1 [To process the spatial mapping data](#)
 - 3.2.1.1 [Preview inside Unity](#)
 - 3.2.1.2 [Video Demo](#)
- 4 [Week 15 Nov 2018 - 20 Nov 2018](#)
 - 4.1 [Object Placement](#)
 - 4.1.1 [Objectives](#)
 - 4.1.1.1 [Video Demo](#)
 - 4.2 [Occlusion](#)
 - 4.2.1 [Video Demo](#)
 - 4.2.2 [Difficulties and troubleshoots](#)
- 5 [Week 21 Nov 2018 - 27 Nov 2018](#)
 - 5.1 [HoloLensForCV Features](#)
 - 5.1.1 [Recorder application](#)
 - 5.2 [Important links](#)
 - 5.2.1 <https://news.microsoft.com/en-gb/2017/01/25/37491/#sm.00v6kb3h1agxcpj10jc1pl5rtvqm6#IXCyBVyJuHzvbMJz.97>
 - 5.3 [Notes](#)
- 6 [Prof Pingbo's Seminar](#)
 - 6.1 [Laser Scanning](#)
 - 6.2 [Could computer capture data collection and data processing behaviors so that computers mimic human](#)
 - 6.2.1 [How to collect data](#)
 - 6.3 [Conclusion](#)

- 7 [Prof Ioannis Brilakis' Seminar](#)
 - 7.1 [A big problem](#)
- 8 [Week 28 Nov 2018 - 4 Dec 2018](#)
 - 8.1 [Encountered errors](#)
 - 8.2 [Second attempt to deploy StreamerVLC separately](#)
 - 8.2.1 [To access the extracted data](#)
 - 8.2.2 [Extracted data](#)
 - 8.2.2.1 [PGM image opened using File Viewer Plus](#)
 - 8.2.2.2 [Long throw depth PGM](#)
 - 8.2.2.3 [Short Throw Depth](#)
 - 8.2.2.4 [Short Throw Reflectivity](#)
 - 8.2.2.5 [Links to possible solutions/similar discussions](#)
 - 8.2.3 [Joined HoloDevelopers' slack to get more information from other developers](#)
- 9 [Week 29th Jan 2019 - 5 Feb 2019](#)
 - 9.1 [Objective](#)
 - 9.1.1 [Computer Vision APIs](#)
 - 9.1.2 [Sending image to python server and back to HoloLens using TCP](#)
 - 9.1.3 [OpenCV](#)
 - 9.1.3.1 [Integrate OpenCV library into Unity and try to use its library to draw rectangle and show it through unity and then HoloLens](#)
 - 9.1.3.2 [HoloLens OpenCV Video](#)
 - 9.1.3.3 [GUI: Box](#)
 - 9.1.3.3.1 [Rect: 2D Rectangle defined by X and Y position, width and height](#)
 - 9.1.3.4 [LineRenderer: Render Lines in Unity](#)
 - 9.1.4 [World Anchor](#)
 - 9.1.4.1 [Videos](#)
- 10 [Week 6 Feb 2019 - 12 Feb 2019](#)
 - 10.1 [Using Microsoft Machine Learning Object Detection Algorithm](#)
 - 10.1.1 [Video Demo](#)
- 11 [2.mp4](#)
- 12 [Week 13 Feb 2019 - 19 Feb 2019](#)
 - 12.1.1 [The precision rate and recall is not high enough](#)
 - 12.1.2 [Video Demo](#)
- 13 [Week 20 Feb 2019 - 26 Feb 2019](#)
 - 13.1 [Plane finding](#)
 - 13.2 [Gaze Cursor with depth perception](#)
- 14 [Week 12 March 2019 - 19 March 2019](#)
 - 14.1 [Client server connection for Microsoft Hololens](#)
 - 14.1.1 [Implement using .NET framework](#)
 - 14.1.2 [Sample code to connect to an IP address, server endpoint and send buffer data](#)
 - 14.1.3 <https://www.youtube.com/watch?v=l2cwuyQVZMg>
- 15 [Week 19 March 2019 - 26 March 2019](#)

- 15.1 [Testing RemoteHostingSample from microsoft mixed reality companion toolkit git repo](#)
- 16 [Week 27 March 2019 - 2 April 2019](#)
 - 16.1.1 [Server](#)
 - 16.1.2 [Code in action](#)
- 17 [Week 3 April 2019 - 9 April 2019](#)
 - 17.1 [Go back to MixedRealityCompanionKit Remote Host Sample](#)
- 18 [Week 10 April 2019 - 16 April 2019](#)
 - 18.1 [Moved on to the HoloLens Camera Stream repo](#)

Week 21 Oct - 27 Oct 2018

Literature review

Augmented and Virtual Reality for Inspection and Maintenance Processes in the Aviation Industry

link: <https://www.sciencedirect.com/science/article/pii/S2351978918300222>

- Different use cases of AR and VR in inspection and maintenance.
- Implementation of Microsoft Hololens in real production environment

IMMERSIVE VIRTUAL ENVIRONMENTS: EXPERIMENTS ON IMPACTING DESIGN AND HUMAN BUILDING INTERACTION

link: http://papers.cumincad.org/data/works/att/caadria2014_161.content.pdf

- To evaluate if an Immersive Virtual Environment is an adequate representation of a e.g. Lighting of physical room and virtual room physical environment based on daily office tasks (e.g. reading, writing, communication, etc)
- Conclusion: no differences in either environment

Week 30 Oct - 6 Nov 2018

Starting the tutorials on Microsoft Hololens documentation website.

- Should have Windows 10, Unity 3D, Visual Studio 2017 installed and a Hololens/emulator.

Configure the Unity settings according to the documentation

1. Every new project created must follow the configuration written on the development page.
2. Link: <https://docs.microsoft.com/en-us/windows/mixed-reality/unity-development-overview>
3. Change user's transform position to (x:0, y:0, z:0) and background to solid black color.
4. Change near clip plane to 0.85 meters for Hololens (recommended distance for Augmented Reality especially when using Hololens)

Tasks: Unity Object

Link: <https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-100>

1. Created a very simple cube object with the correct transform positions
2. Rendered the cube object on Hololens remotely through wifi

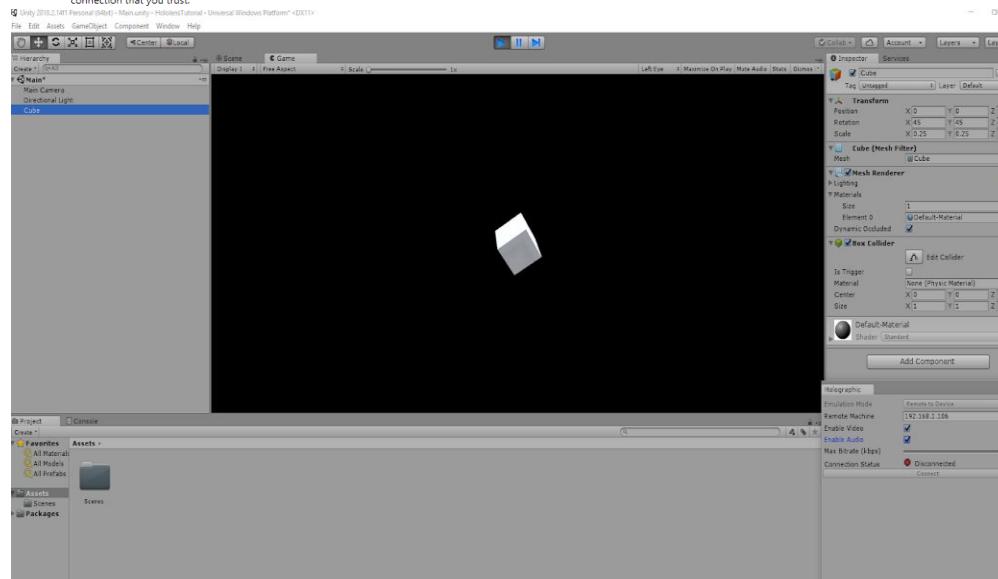
Connecting to the Holographic Remoting Player

Follow your app's instructions to connect to the Holographic Remoting Player. You will need to enter the IP address of your Hololens device, which you can see on the Remoting Player's main screen as follows:



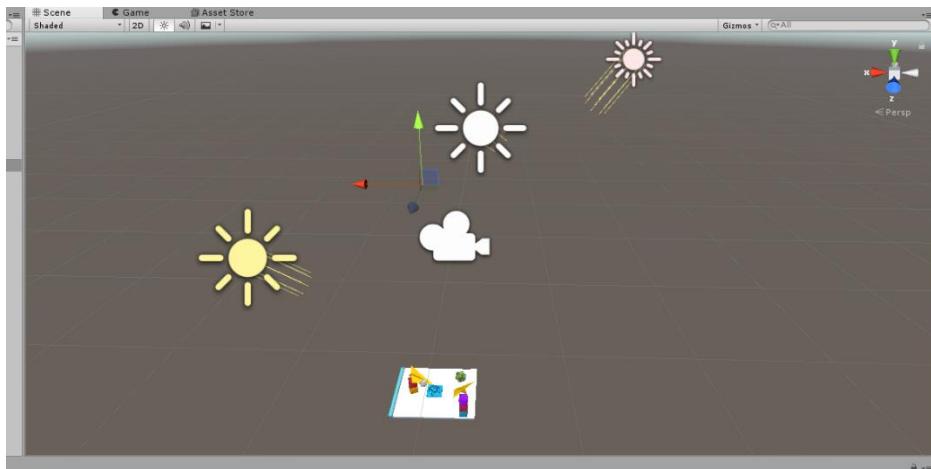
Whenever you see the main screen, you will know that you do not have an app connected.

Note that the holographic remoting connection is **not encrypted**. You should always use Holographic Remoting over a secure Wi-Fi connection that you trust.

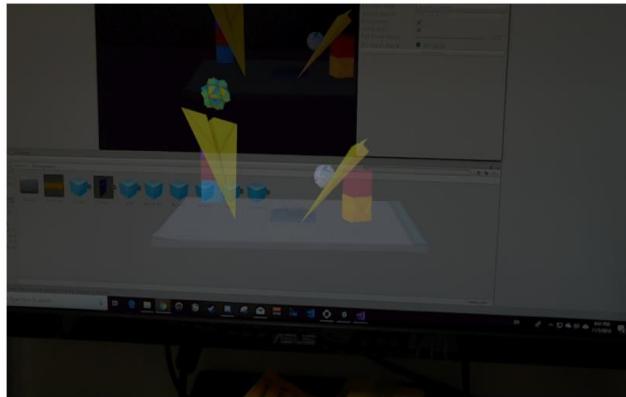


Light Source inside Unity

Link: <https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-101>



Hologram with light sources:



Holograms without any light source:

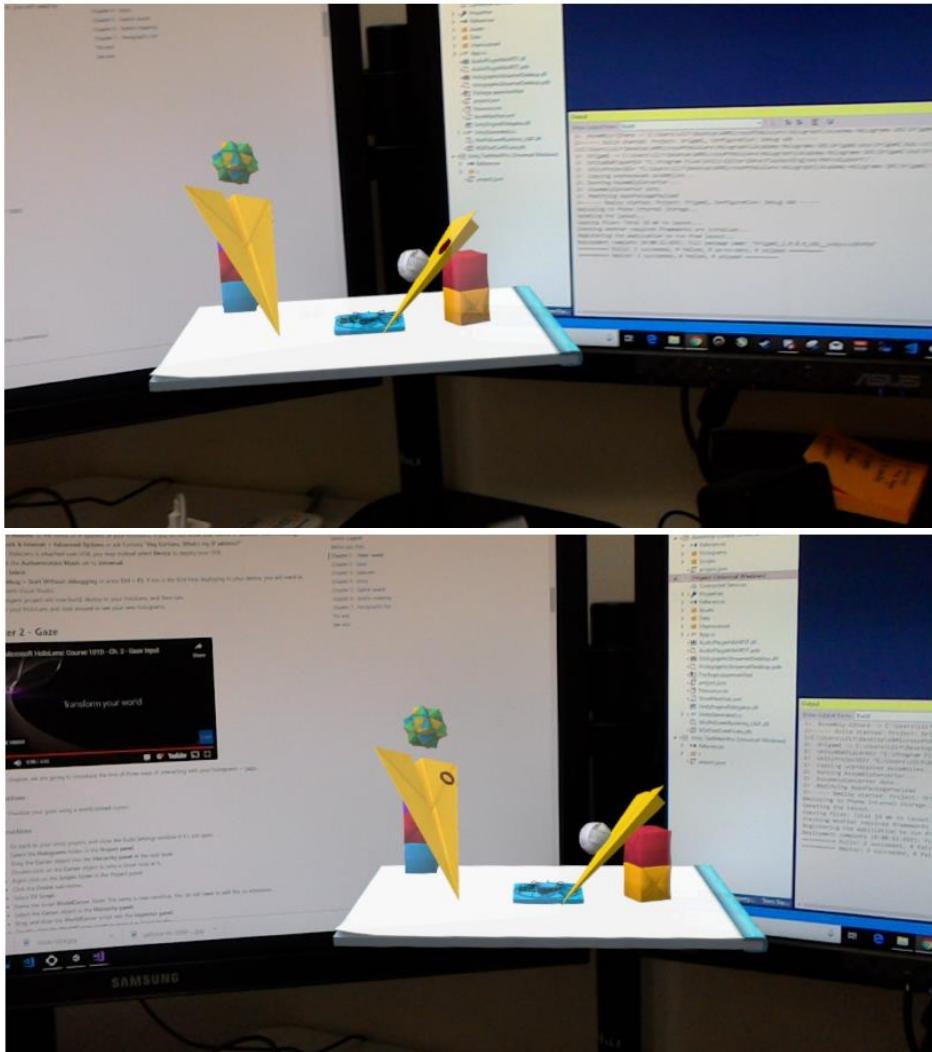


Adding cursor

Goal : To let the users know where they are gazing at

1. Add a script in the project to create a cursor
2. Each frame, do a Raycast into the scene to figure out where the user's gaze is.
3. When we figure out where that gaze point hits one of the holograms, we want to take the cursor and place it there.
4. Gaze cursor appear and the gaze cursor will "huge" the surface of the hologram

Note: RayCast is an operation from the camera point sending out a "ray" find the surface it collides with and then render the material on that surface for that pixel in the screen.



Adding Gestures

Goal: To let user to interact with hologram object using gestures

Use gaze to target a specific hologram and route gestures to that hologram

Video Demo



Added tap gesture, when tap onto a sphere object, it will apply physics to the object, causing it to drop due to gravity.

Troubleshoot

Errors encountered when building the project and importing it to Visual Studio 2017 for debugging purposes

1. Remove TextMeshPro from package manager inside Unity under "Window" tab, because the tutorial is not using it which is causing the error
2. Switch platform to UWP inside build settings under the "File" tab.
3. Change scripting backend to .NET under Inspector panel and check Unity C# projects before you can build the project
and convert it into a SLN (a structure file used for organizing projects in Microsoft Visual Studio) file.

Week 7 Nov 2018 - 14 Nov 2018

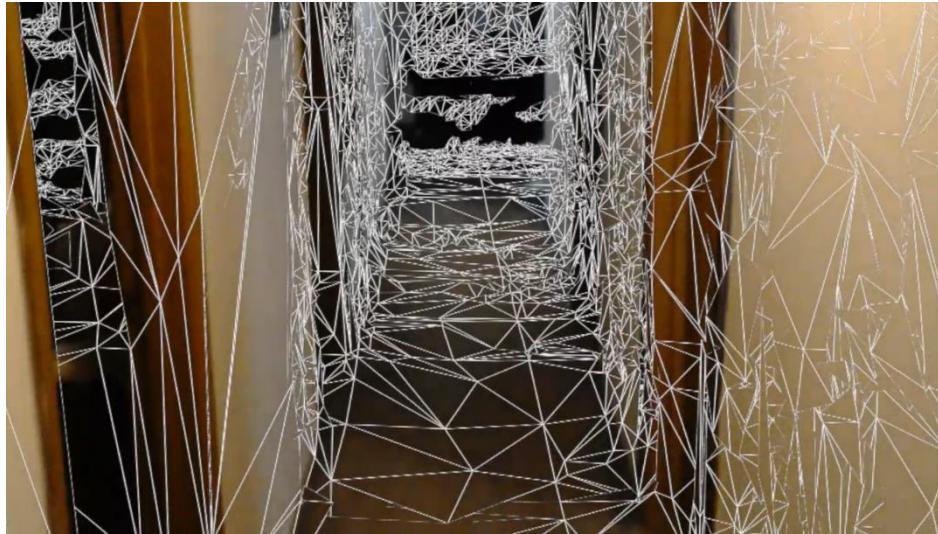
Spatial Mapping

- ability to visualize our world
- to place hologram on real world surfaces
- to interact with the world using physics
- navigate the world

link: <https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-230>

Spatial Mapping of a room

Running the program in Visual Studio

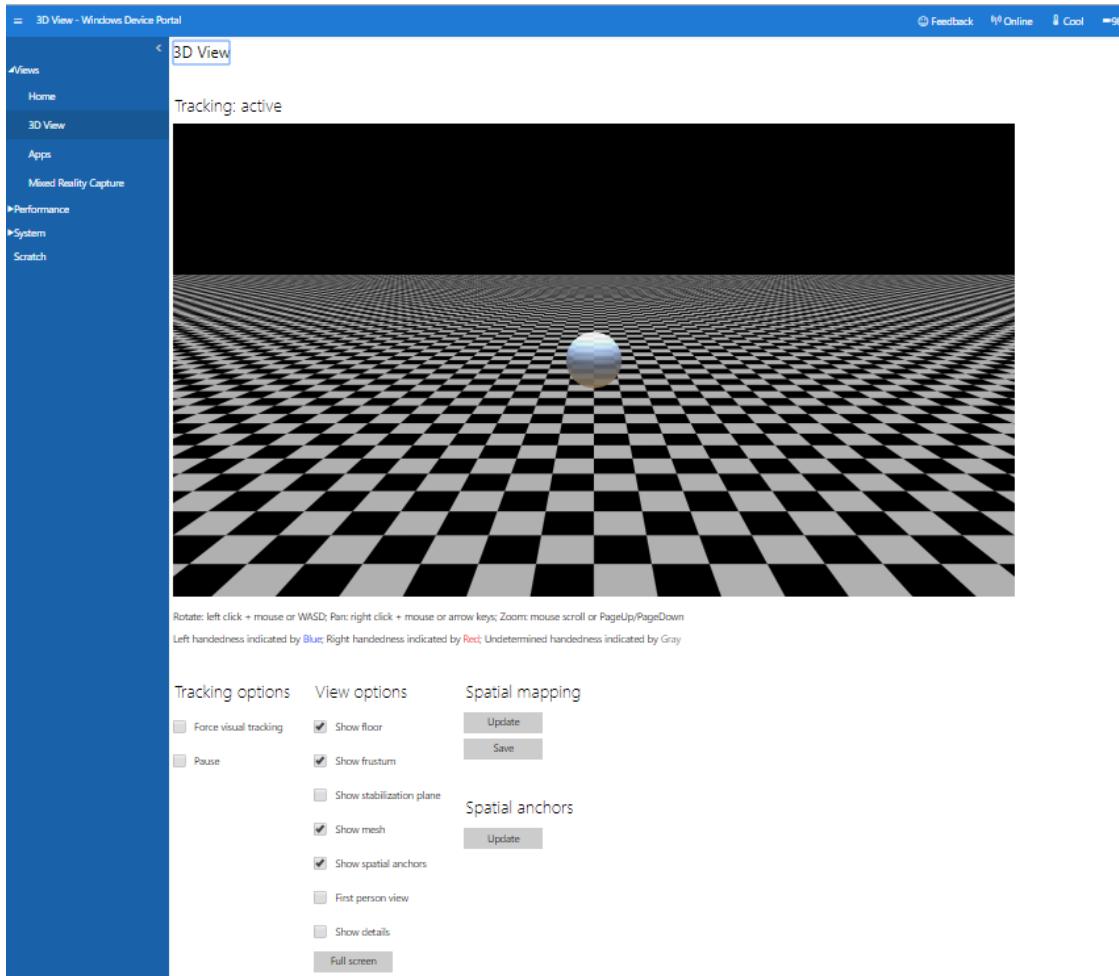


Video Demo:

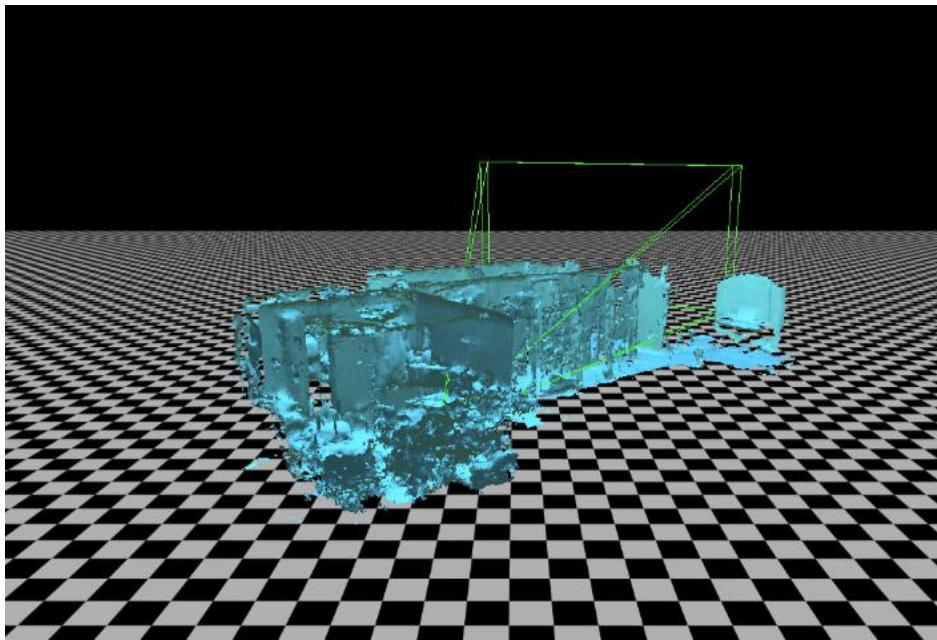


Windows Device portal via browser

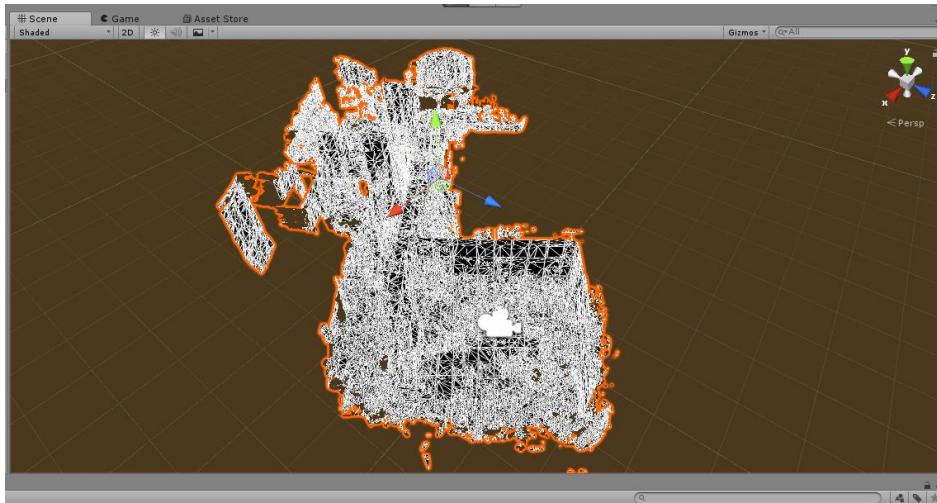
While running the program in Visual Studio, update the spatial mapping inside the Windows Device Portal



Save the 3D object into the Asset folder of the project



Viewing the scanned room (3D object) in Unity

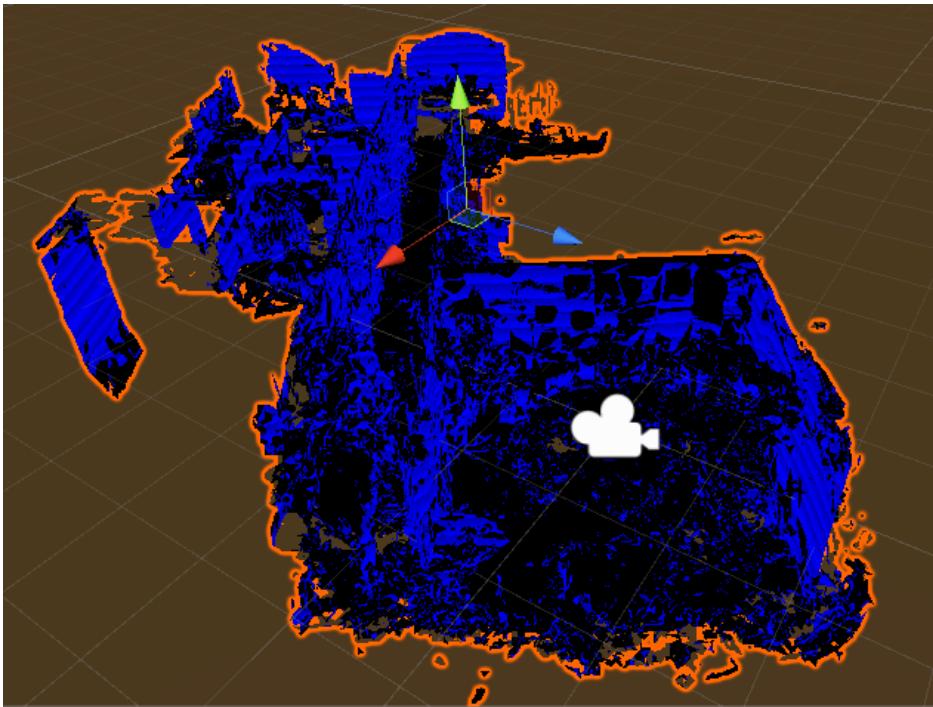


Video Demo



spatialmappinginunity.mp4

Apply shader to visualize the spatial mapping data



In Unity,

Adjust LineScale to make lines thicker/thinner

Adjust LinesPerMeter to change how many lines appear on each wall

Video Demo:



Spatial processing

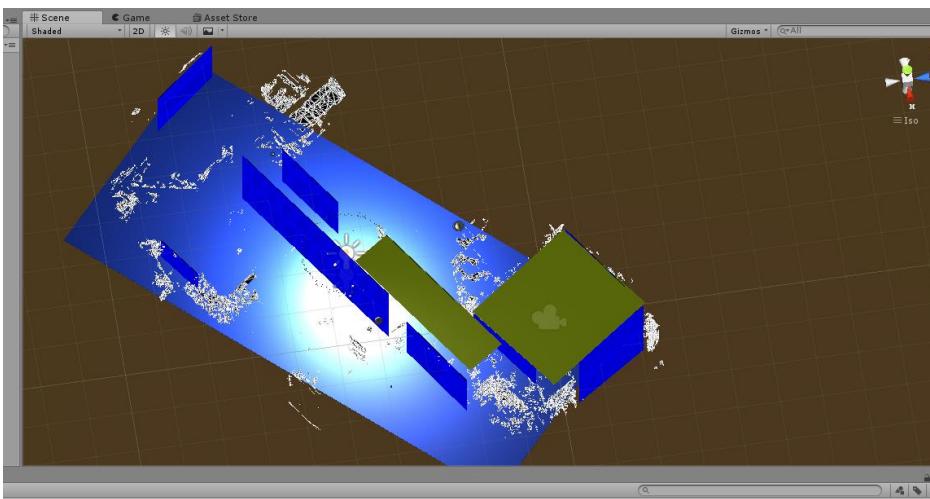
To process the spatial mapping data

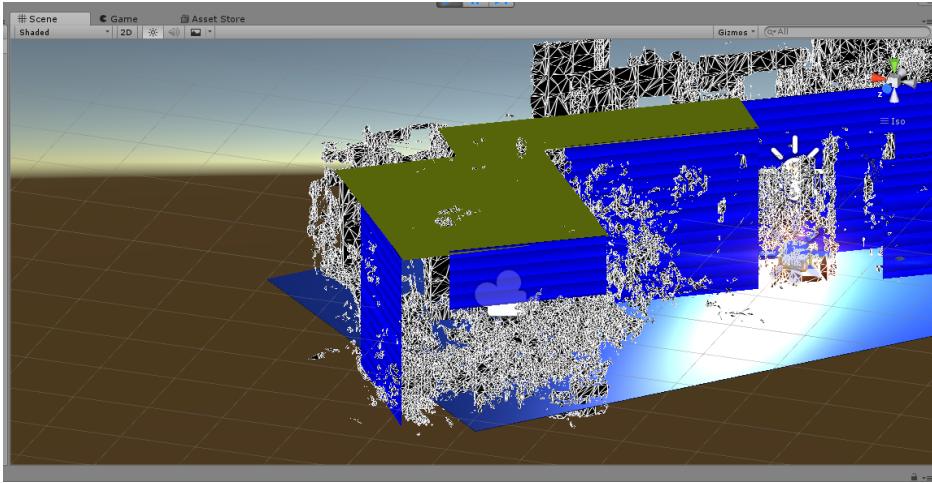
1. Generate planes - to represents walls, floors, ceilings
2. Remove surface vertices - remove excess triangles (smoothen it)
3. Generate hologram in the world and place on wall and floor tharts near the user

Plane finding, after found planes, it will render the secondary material

Using the plane finding library, it will analyze floor, wall, ceiling and generate 3D model planes

Preview inside Unity





Video Demo



Week 15 Nov 2018 - 20 Nov 2018

Object Placement

Objectives

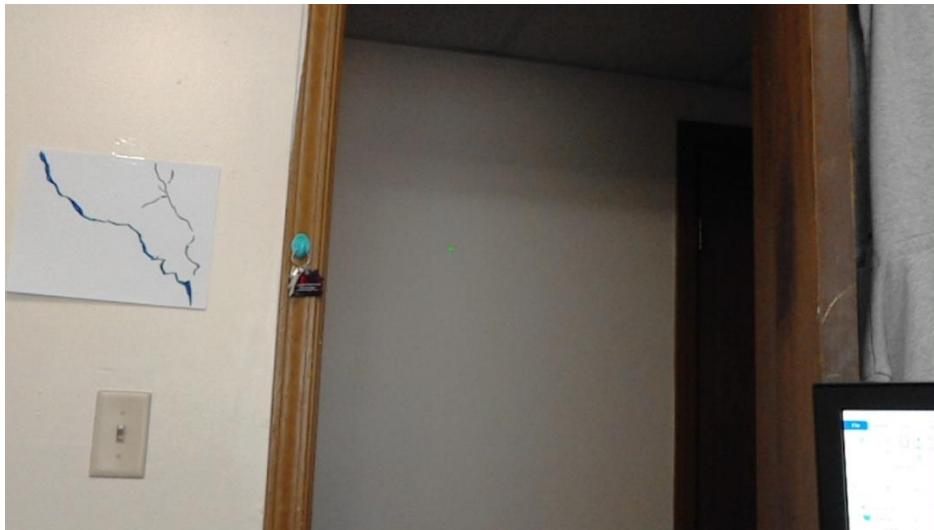
- Determine if hologram will fit on a surface
- Provide feedback to the user when a hologram can or cannot fit on a surface

Added a new script to the project, it will:

1. Determine if a hologram will fit on a surface
2. Check if the surface is smooth enough to place the hologram

3. Cast a shadow under/behind the hologram to show where it will be placed on the floor/wall
4. The shadow will be shown as red if cannot be placed on the surface, otherwise it's green
5. Hologram will be re-oriented to align with the surface

Hologram is not within the area of surface, causing it unable to be placed and the color of shadow to be red



Hologram is within valid area, color of the shadow turns green



Video Demo

After applying tap gesture, poster hologram will move a small distance away from the surface and follow the gaze of user and place onto a valid surface on second tap



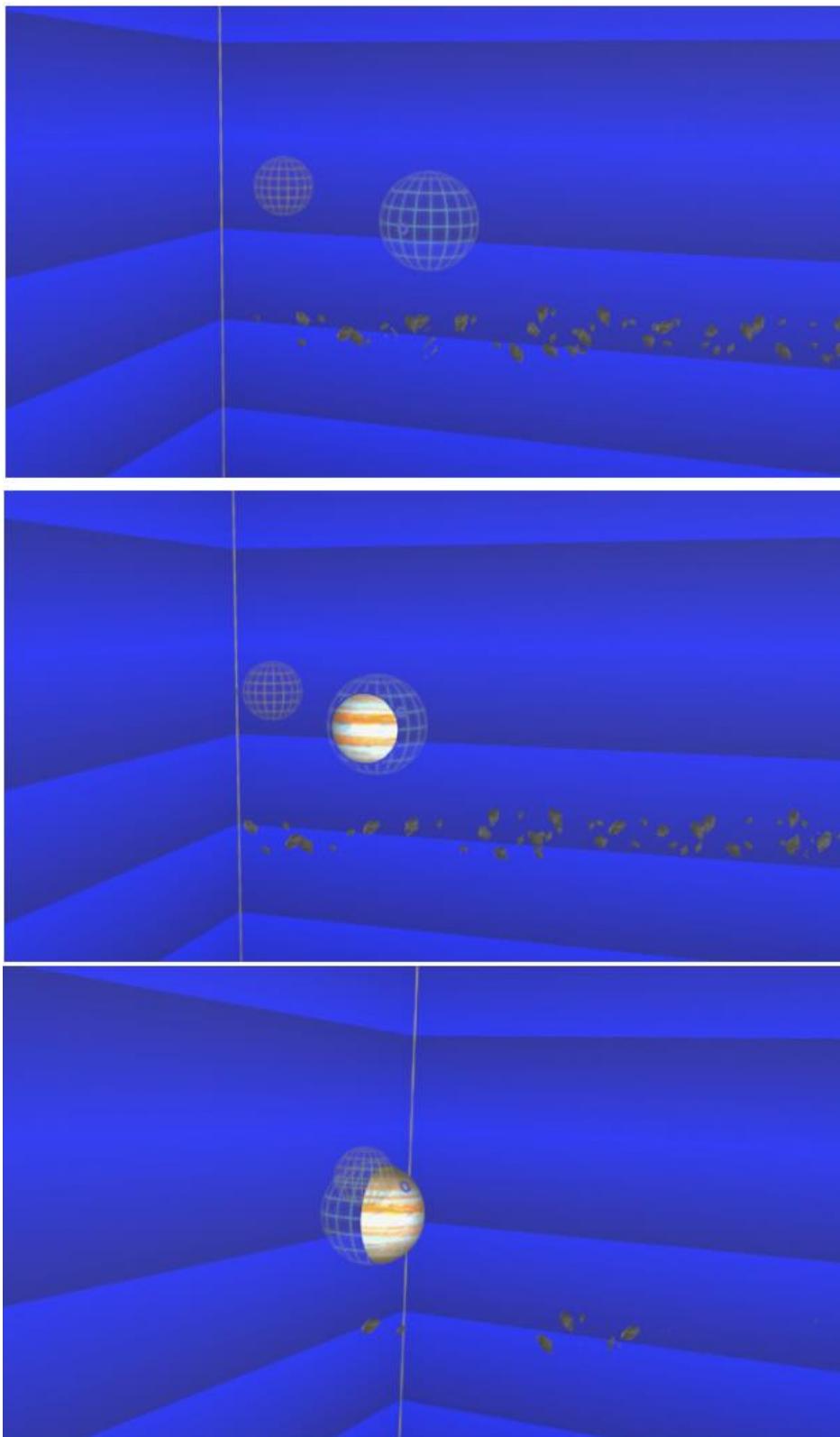
Occlusion

Hide holograms behind real world surfaces and highlight holograms even if it goes behind a real world wall



Added script that can

1. Determine if the object is obstructed by the SpatialMapping layer (room meshes and planes).
2. Show wire frame representation of a object whenever it is obstructed by the SpatialMapping layer.
3. Hide the wire frame representation when it is not blocked by the SpatialMapping layer.



Usage: Can determine whether the 3D object is behind a surface or not, and being able to see the object in a "x-ray vision" style

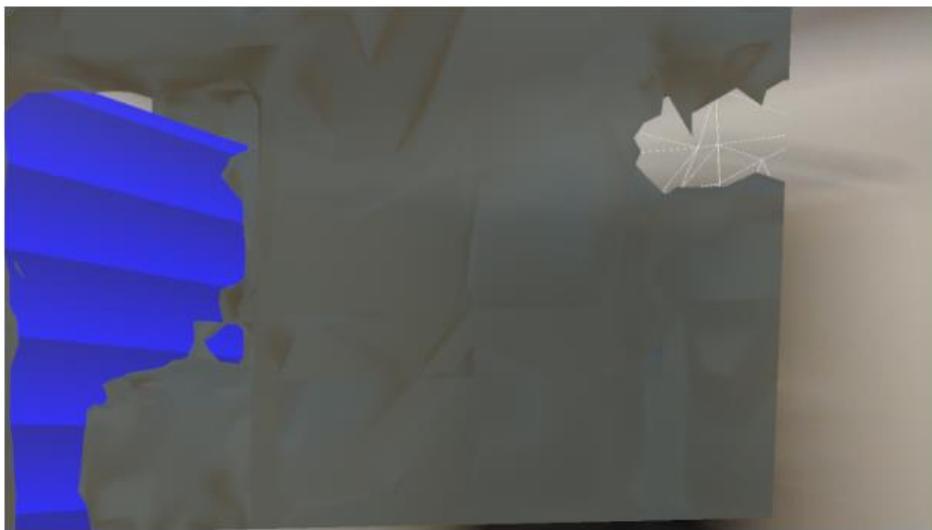
Video Demo



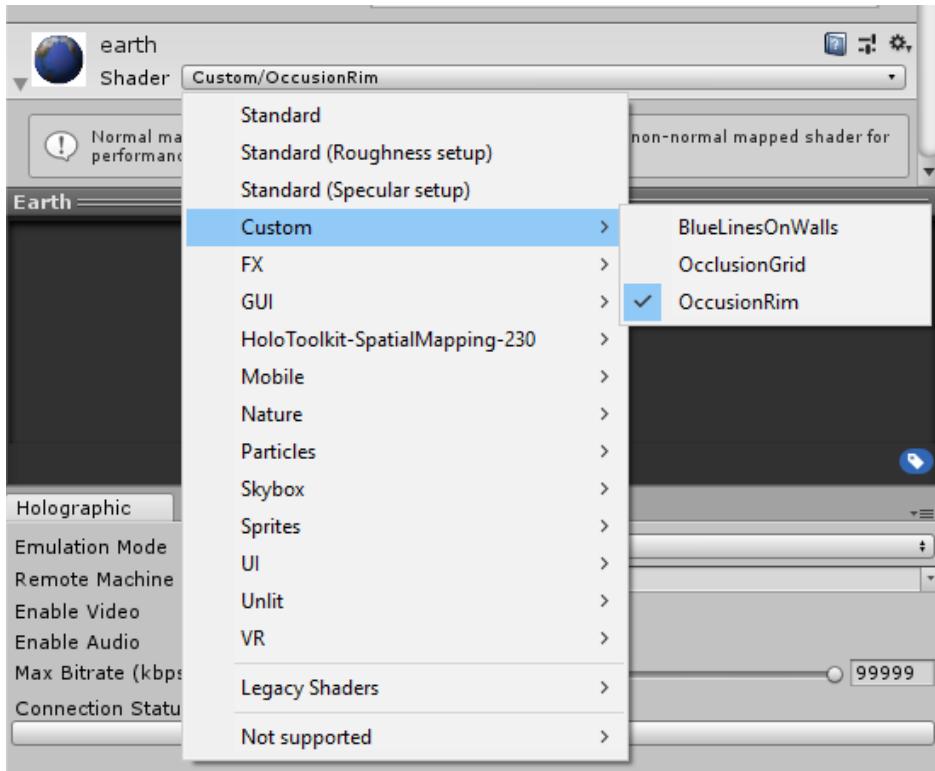
Difficulties and troubleshoots

Attempt to remove wire mesh from asset (secondary material in spatial processing), ended up with weird spatial mapping visual

Solution: reset commit using github



The position of the custom settings for OcclusionRim has changed in the newer version of unity



Link to explore:

<https://medium.com/@dongyoopark/open-source-building-blocks-for-windows-mixed-reality-experiences-hololens-mixedrealitytoolkit-28a0a16ebb61>

Notes:

Create a region of interest by plotting points

Week 21 Nov 2018 - 27 Nov 2018

Working on retrieving data

Github link: <https://github.com/Microsoft/HoloLensForCV>

<https://docs.microsoft.com/en-us/windows/mixed-reality/research-mode>

Get Started

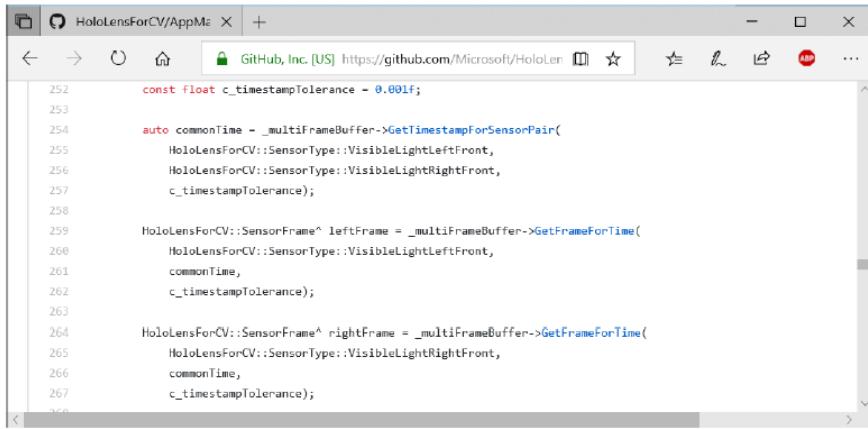
API available for research mode

Research Mode API

- Low-level access
 - Timestamps
 - Sensor images
 - Sensor poses
 - 6 DOF extrinsic calibration
 - Sensor calibration
 - Undistortion map from pixels to rays

HoloLensForCV Features

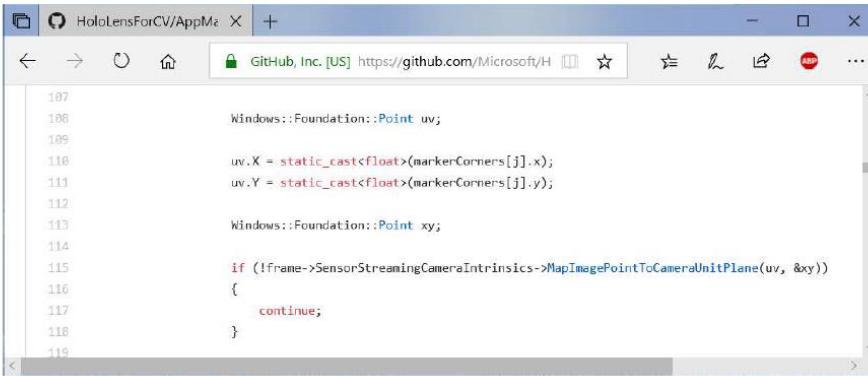
- Frame synchronization



A screenshot of a Microsoft Edge browser window displaying a GitHub page. The code shown is part of a C++ file, likely `AppMain.cpp`, demonstrating how to synchronize frames from two cameras. It uses the `HoloLensForCV` library to get frames for specific sensors at a common timestamp.

```
252     const float c_timestampTolerance = 0.001f;
253
254     auto commonTime = _multiFrameBuffer->GetTimestampForSensorPair(
255         HoloLensForCV::SensorType::VisibleLightLeftFront,
256         HoloLensForCV::SensorType::VisibleLightRightFront,
257         c_timestampTolerance);
258
259     HoloLensForCV::SensorFrame^ leftFrame = _multiFrameBuffer->GetFrameForTime(
260         HoloLensForCV::SensorType::VisibleLightLeftFront,
261         commonTime,
262         c_timestampTolerance);
263
264     HoloLensForCV::SensorFrame^ rightFrame = _multiFrameBuffer->GetFrameForTime(
265         HoloLensForCV::SensorType::VisibleLightRightFront,
266         commonTime,
267         c_timestampTolerance);
```

- Undistortion using camera intrinsics



A screenshot of a Microsoft Edge browser window displaying a GitHub page. The code shown is part of a C++ file, likely `AppMain.cpp`, demonstrating how to map image points to a camera's unit plane using sensor streaming camera intrinsics. This is used for undistortion.

```
107
108     Windows::Foundation::Point uv;
109
110     uv.X = static_cast<float>(markerCorners[j].x);
111     uv.Y = static_cast<float>(markerCorners[j].y);
112
113     Windows::Foundation::Point xy;
114
115     if (!frame->SensorStreamingCameraIntrinsics->MapImagePointToCameraUnitPlane(uv, &xy))
116     {
117         continue;
118     }
```

Follow the steps:

1. Install Python 2.7
2. Open HoloLensForCV sln in Visual Studio
3. Select <Release/x86> configuration
4. Connect and pair HoloLens
5. Deploy the application to HoloLens (Select the project next to <Release/x86> lists, select <Device> as the target, start new instance of project or deploy with right click on project in the solution explorer)

From the github folder:

Summary

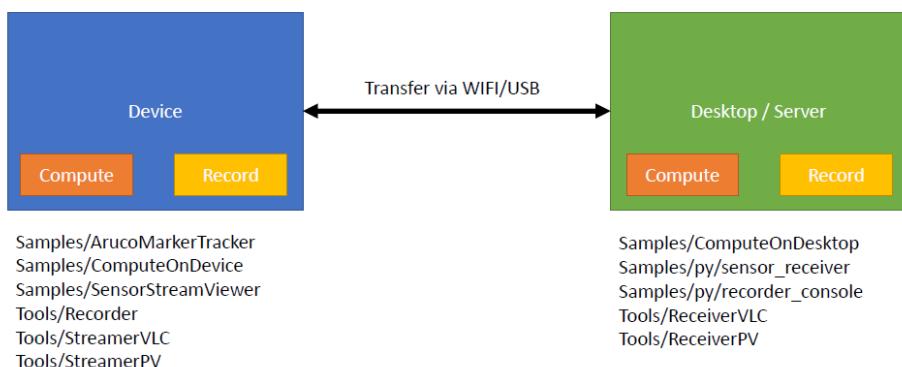
The 'Shared\HoloLensForCV' Universal Windows Platform (or, UWP) component provides an easy interface to enumerate HoloLens sensors and to allow apps easy access the sensor streams.

The component also includes both client and server code to enable streaming sensor data to a companion PC, as well as a recorder functionality that produces a tarball with the camera images and sensor metadata that can be used for offline/batch processing.

Note that support for additional HoloLens sensors (ToF Depth, Visible Light, ...) is not currently available publicly. Stay tuned for updates!

Note: Instead of sending it to azure, we can just reroute it to our own device to further process the data

Research Mode Scenarios



Work flow:

1. Connecting the device to a machine, whether a desktop or a server through wifi using IP address or USB.
2. Run/deploy the HoloLensForCV.sln using visual studio
3. On desktop, run the python script that connects, receives and parses the HoloLens Research Mode Streamer data
4. Note: The python script recorder_console.py is to download the data, another way is to download the data from device portal file browser

Recorder application

The 'Tools\Recorder' project is an app that makes use of HoloLensForCV component's API to record all sensor streams and their corresponding poses into tar archives on the internal HoloLens storage.

Once the application is deployed and launched on the HoloLens, follow the voice UI prompts to start and stop recording.

The tarballs containing the recordings can be downloaded from the HoloLens using the Samples/py/recorder_console.py script or the Device Portal's file explorer.

- It stores sensor streams on HoloLens for later offline processing
- We can download the data via device portal file browser or using the python script

Instructions

Download and reconstruct on desktop

1. Start the console script

```
$ python .\HoloLensForCV\Samples\py\recorder_console.py \
--dev_portal_user "user" \
--dev_portal_password "password" \
--workspace_path C:\Path\To\Workspace

Connecting to HoloLens Device Portal...
=> Connected to Hololens at address: http://127.0.0.1:10080
> Searching for CV recorder application...
> Found CV recorder application with name: 7A37D94C-C432-4875-0C57-...
Searching for recordings...
=> Found a total of 6 recordings

Available commands:
help: Print this help message
exit: EXIT to console loop
list: List all recordings
list device: List all recordings on the Hololens
list workspace: List all recordings in the workspace
download X: Download recording X from the Hololens
delete X: Delete recording X from the Hololens
delete all: Delete all recordings from the Hololens
extract X: Extract recording X in the workspace
reconstruct X: Perform sparse and dense reconstruction of recording X in workspace
reconstruct sparse X: Perform sparse reconstruction of recording X in workspace
```

Instructions

Download and reconstruct on desktop

2. List recordings

```
>>> list

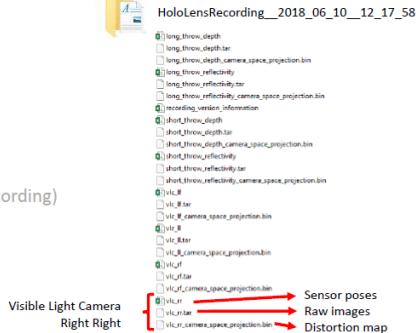
Device recordings:
[0] HololensRecording_2018_06_10_12_37_59
[1] HololensRecording_2018_06_10_12_36_17
[2] HololensRecording_2018_06_10_12_39_00
[3] HololensRecording_2018_06_10_12_34_25
[4] HololensRecording_2018_06_10_12_46_38
[5] HololensRecording_2018_06_10_12_46_38

Workspace recordings:
[ ] No Recordings found in workspace
```

3. Download recording (can also extract/delete recording)

```
>>> download 0
=> Downloading ...
...
>>> list workspace
```

```
Workspace recordings:
[0] HololensRecording_2018_06_10_12_17_58
```



Important links

1. <https://cit.eng.cam.ac.uk/news/getting-virtual-infrastructure-models-out-computer-and-workspace>

1. Engineers will soon be able to visualize Building Information Models (BIMs) in full scale at their offices
2. Bridge Inspector HoloLens app - engineers are able to inspect the condition of infrastructure in office as if they were standing on site in front of the real asset
3. The app allows construction inspectors to both visualize progress and automatically detect building elements that should have been built at the time of their visit but are missing
2. https://www.youtube.com/watch?v=qmqBE4OA_xM&feature=share
3. <https://blogs.windows.com/devices/2017/01/25/trimble-university-cambridge-build-future-construction-hololens/#1pDkZvMx4vWIsVDS.97>
4. <https://news.microsoft.com/en-gb/2017/01/25/37491/#sm.00v6kb3h1agxcpj10jc1pl5rtvqm6#lXCyBVyJuHzvbMJz.97>

Notes

- The python script can be found in HoloLensForCV/Samples/py/
- Will be trying to execute all the above in the coming week
- interpreting the data: pixel values, RGB /jpeg, png
- find out what kind of data i can get

Prof Pingbo's Seminar

Professor Pingbo focused his research in the CEM (Construction Engineering and Management) field

Mainly research about human systems

Laser Scanning

The tool that he used to produce 3D structure is laser scanning

Require better data to understand, so the goal is to use better technology to produce better data and help civil engineers

e.g. 3D data stitching

Another field is use computer to observe and help us understand human behaviors

e.g. Analyze how human interact with workspace and what kind of layout influence the human behavior

- digital world
- human behavior

- schedule
- workspace

How to track and analyze human behaviors in the field

- Before worker get in to reactor, a room called radiation protection island, need to check radiation, and get into room to work
- Put camera there bcoz everyone go through there

Use simulation model, find bottle neck of the construction workflow

- Use CV to track how many ppl are waiting
- Show which room is the bottle neck
- How to rearrange the schedule to improve productivity
- Need simulation model to simulate what's not in the situation

Case: can't put camera everywhere

- So derive information using just one camera
- project 2d to a top view
- project the video to a top view
- use occlusion to detect body parts of candidates
- create very accurate human 3D model (a lot of problems)

Use CV algorithm to analyze video and see what time is peak hour what time is not

- check and see which part is forming difficulties to the algorithm
- precision and recall

Challenge: person A becomes person B, B becomes A

- problem - ghost: mirror, shiny surfaces, reflection
- computer create historical data and smart label which part of the building is creating problem
- why ppl are moving, which one is troublemaker, purpose of doing what they are doing

Could computer capture data collection and data processing behaviors so that computers mimic human

- Geometric and visual details needed
- Hard to reach some locations to measure
- Time and resource constraints for data collection and processing

How to collect data

- Remotely collect detailed visual and geometric data through 3D imaging (laser scanning)
- Collect less but important data
- Details of building is not enough using by just 3D laser scanning
- 3D imaging planning
 - only look at certain group of data
 - scanning position
 - scanning parameters
- Computer learn from human and perform better than professional inspectors, scans better than human
 - e.g. want to get cross section of bridge columns
 - get all the cross section and area
 - too many repetitive tasks
 - use computer to repeat the measurements
- Under clearance
 - use computer to calculate thousands
- remember what engineers do
 - automatically combine what they did to solve new problems
 - suggest engineers how to better process data based on what it remembers

Impacts of human behaviors on productivity and safety

- airport delay because of miscommunication of plane and tower, almost crash into each other (runway incursion)
- limited incidents and accidents data
- things can go wrong in many ways

Get accident reports from organizations DOE

- lab experiments to capture and analyze detailed human behavioral data
- conduct simulation
- use bayesian network model for runway incursion prediction

Lab experiment to understand how human behavior influence delays and schedule reliability

- supervisor vs computer
- sometimes worker not looking at camera so computer cannot process

Conclusion

Learn from human, use computer to learn the behavior and use the data to perform better

Prof Ioannis Brilakis' Seminar

Digital Twin

- replicate the physical asset (geometry etc)
- using drone to scan ridge, get pixels, add texture
- Extract raw geometry
- mid range mobile videogrammetry - lines and planes

A big problem

- point clouds are meant for computers not human
- not directly exploitable
- need to generate raw data into visualizable 3d model

Challenge: took very long to laser scan an object

- bottom up method in computer vision, fail miserably
- From video to BIM
- use machine learning to generate the occluded object
- Use textures to detect the defect

DFMA - Design for Manufacture and Assembly.

- robotic teleoperation: remotely control robot
- use hololens to observe construction site
- use hololens to identify floor, wall, ceiling

Gygax - allow u to process point clouds, image, videos (Github)

PCL - point cloud library

To get measurement of the damage

Thermal, computer vision sensor, laser scanner

- RGB, thermal
- just walk around (google tangle) can be used for outdoor
- radio frequency at construction site

Week 28 Nov 2018 - 4 Dec 2018

Link to the showcase video of research mode

Encountered errors

1. When trying to deploy the whole solution

```
Checking whether required frameworks are installed...
Registering the application to run from layout...
DEP0700: Registration of the app failed. [0x80073CFD] Windows cannot install package 7A37D9
    only be installed on the following device families: Windows.Holographic
===== Build: 12 succeeded, 0 failed, 6 up-to-date, 0 skipped =====
===== Deploy: 4 succeeded, 6 failed, 0 skipped =====
```

2. Try to deploy the tools in the solution separately

Deployment failed. Deployment of the package failed. [0x800704FD] Windows cannot install package 7A37D94C-C452-4B75-BC57-FA18E51C9B84_1.0.0.0_x86_Sweaty53d9b0e because this package is not compatible with the device. The package can only be installed on the following device families: Windows.Holographic

3. Rebuild the solution and more errors occurred

Suspected issue: certificate does not exist

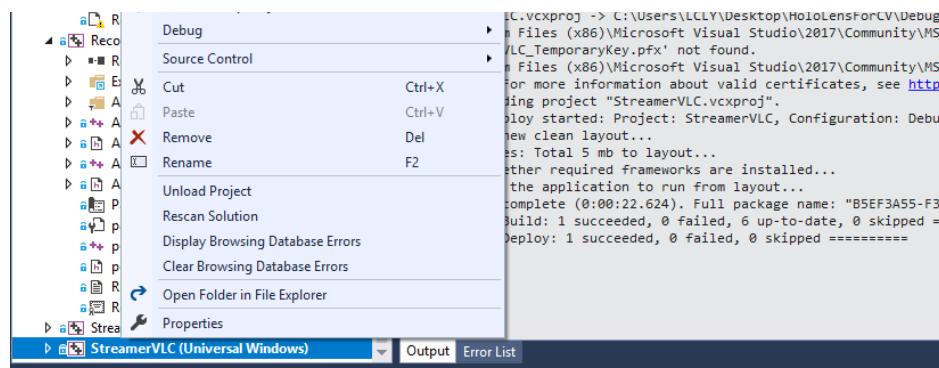
Might be due to the fact that the dev certificate is not checked in to the git repository.

Possible solution: create our own certificate

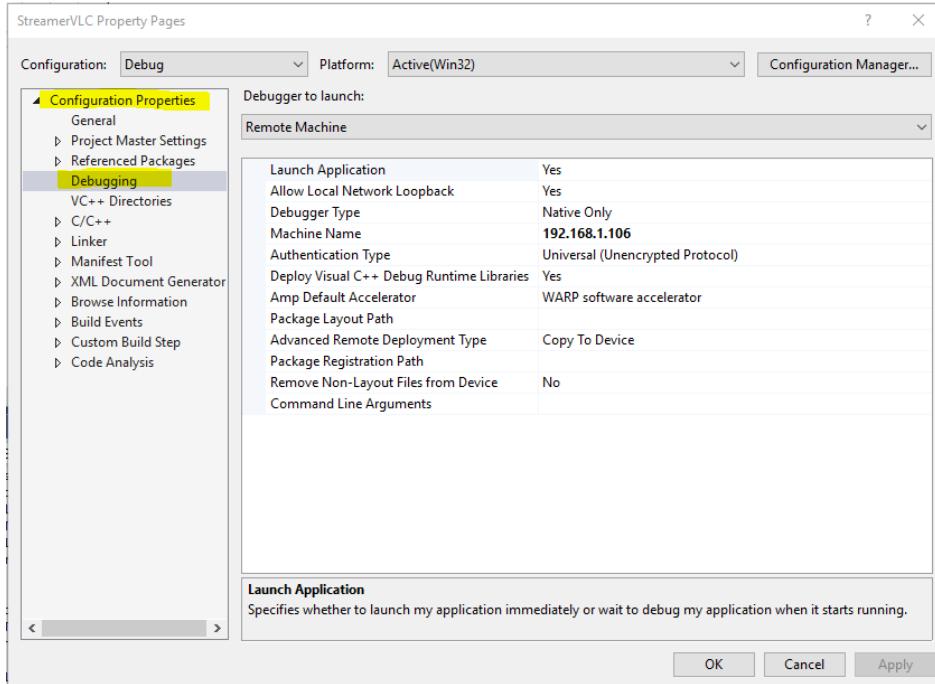
Second attempt to deploy StreamerVLC separately

Usage: To steam the visible light camera (VLC) images

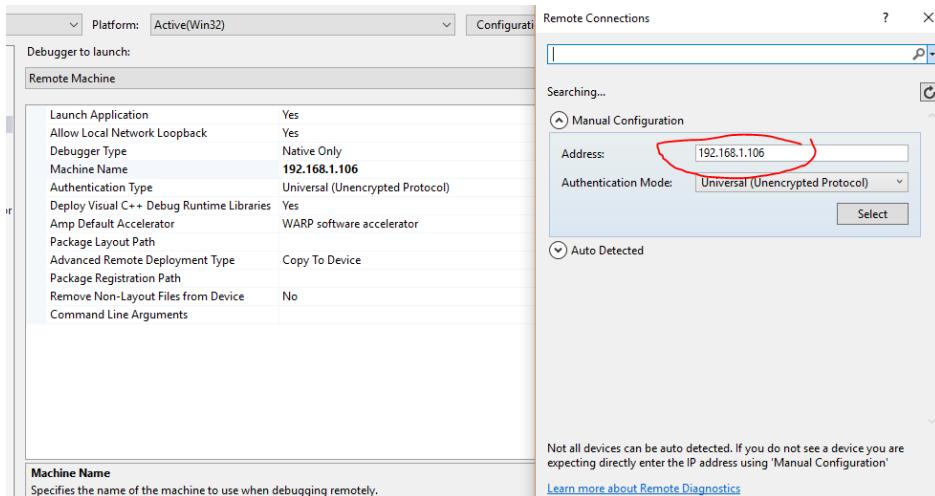
1. Right click the project and choose properties



2. Under Configuration Properties, go to Debugging



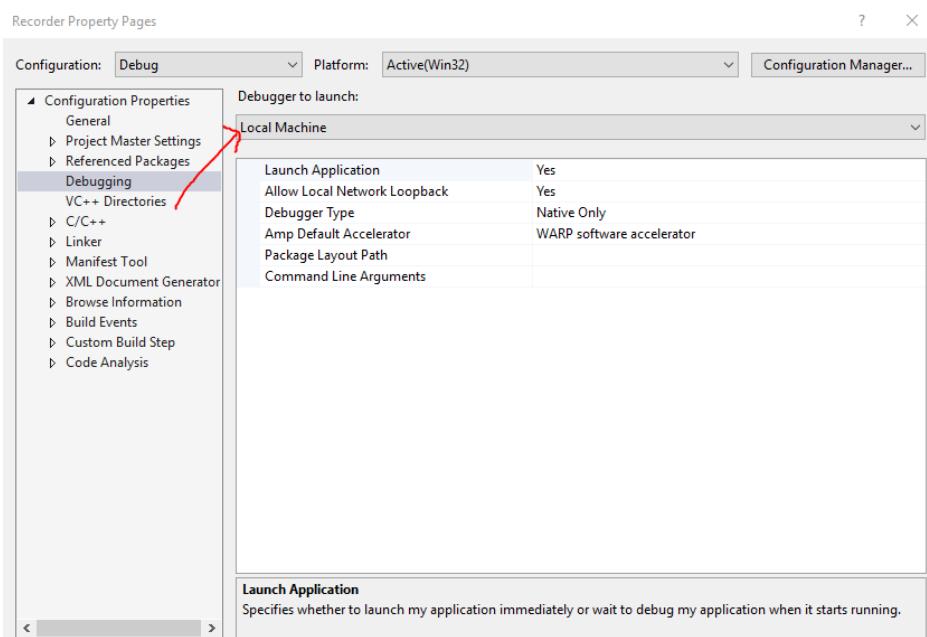
3. Insert the IP address of the machine

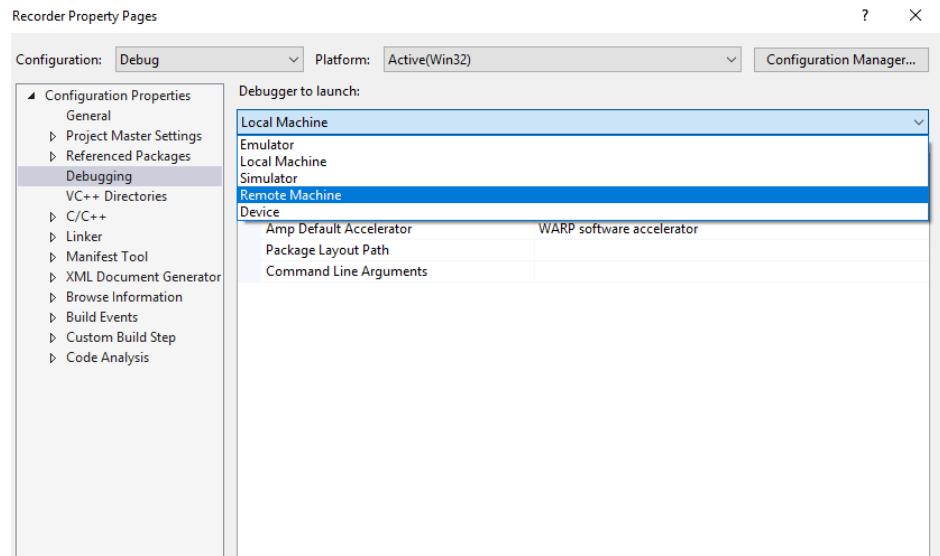


Demo of the stream

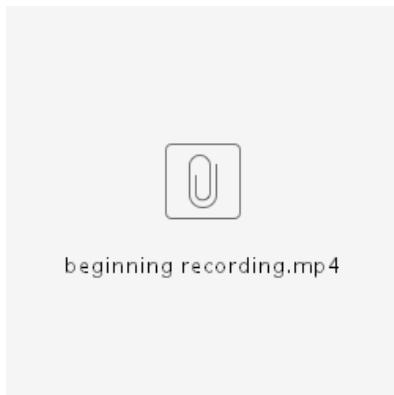


First time RecorderVLC is not able to deploy because it was set to launch on a local machine, it should be remote machine with the Hololens IP address

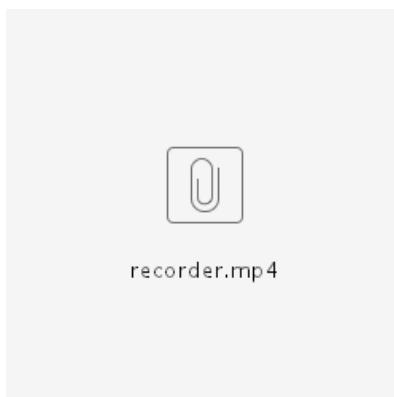




The voice will ask to air tap to begin recording the stream and air tap again to end recording



And then we will have to end the recorder



It will save the files in the Hololens

To access the extracted data

1. Go into the file explorer
2. Click on CV:Recorder
3. Click on TempState
4. Click into one of the recordings

Directory path
User Folders \ Local\AppData \ 7A37D94C-C432-4875-8C57-FA185E1C92B4 1.0.0.0 x64_8wekyb3d8bbwv \ TempState \

Directory contents

Type	Name	Date Created	File Size	Save	Delete	Rename
□	archiveSource	12/4/2018, 2:11:05 PM				✎
□	HoloLensRecording_2018_12_04_18_57_14	12/4/2018, 1:56:52 PM				✎
□	HoloLensRecording_2018_12_04_19_04_27	12/4/2018, 2:03:54 PM				✎
□	HoloLensRecording_2018_12_04_19_08_01	12/4/2018, 2:07:33 PM				✎
□	HoloLensRecording_2018_12_04_19_10_35	12/4/2018, 2:09:45 PM				✎

Upload a file to this directory

Extracted data

File explorer - Windows Device Portal

Views

- Home
- 3D View
- Apps
- Mixed Reality Capture
- Performance
- Performance Tracing
- Processes
- System Performance
- System
- App Crash Dumps
- Device manager
- File explorer
- Logging
- Networking
- Preferences
- Research mode
- Simulation
- Virtual Input
- Scratch

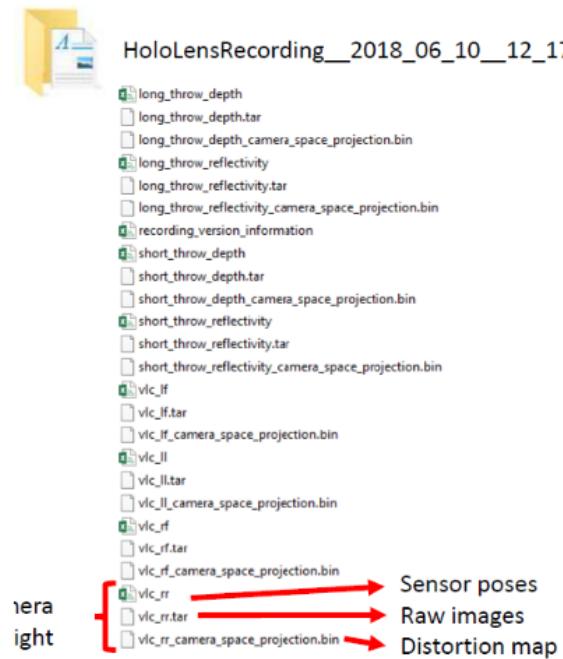
File explorer

Directory path
User Folders \ Local\AppData \ 7A37D94C-C432-4875-8C57-FA185E1C92B4 1.0.0.0 x64_8wekyb3d8bbwv \ TempState \ HoloLensRecording_2018_12_04_18_57_14 \

Directory contents

Type	Name	Date Created	File Size	Save	Delete	Rename
□	long_throw_depth.csv	12/4/2018, 1:56:52 PM	3.5 KB	✎	✉	✎
□	long_throw_depth.tar	12/4/2018, 1:56:52 PM	2.7 MB	✎	✉	✎
□	long_throw_depth_camera_space_projection.bin	12/4/2018, 1:57:02 PM	1.5 MB	✎	✉	✎
□	long_throw_reflectivity.csv	12/4/2018, 1:56:52 PM	3.6 KB	✎	✉	✎
□	long_throw_reflectivity.tar	12/4/2018, 1:56:52 PM	1.4 MB	✎	✉	✎
□	long_throw_reflectivity_camera_space_projection.bin	12/4/2018, 1:57:03 PM	1.5 MB	✎	✉	✎
□	pv.csv	12/4/2018, 1:56:52 PM	1.2 KB	✎	✉	✎
□	pv.tar	12/4/2018, 1:56:52 PM	1024.0 b...	✎	✉	✎
□	recording_version_information.csv	12/4/2018, 1:56:58 PM	34.0 bytes	✎	✉	✎
□	short_throw_depth.csv	12/4/2018, 1:56:52 PM	46.7 KB	✎	✉	✎
□	short_throw_depth.tar	12/4/2018, 1:56:52 PM	51.6 MB	✎	✉	✎
□	short_throw_depth_camera_space_projection.bin	12/4/2018, 1:57:00 PM	1.5 MB	✎	✉	✎
□	short_throw_reflectivity.csv	12/4/2018, 1:56:52 PM	46.9 KB	✎	✉	✎
□	short_throw_reflectivity.tar	12/4/2018, 1:56:52 PM	25.5 MB	✎	✉	✎
□	short_throw_reflectivity_camera_space_projection.bin	12/4/2018, 1:57:01 PM	1.5 MB	✎	✉	✎
□	vlc_if.csv	12/4/2018, 1:56:52 PM	52.7 KB	✎	✉	✎
□	vlc_if.tar	12/4/2018, 1:56:52 PM	46.4 MB	✎	✉	✎

According to the slides provided by microsoft, vlc_rr.tar contains the raw images



Downloaded the tar and extracted it

Name	Date modified	Type	Size
vlc_rr	12/4/2018 2:16 PM	File folder	
vlc_rr.tar	12/4/2018 2:15 PM	tar Archive	49,064 KB

The files are in PGM files, a grayscale image file saved in the portable gray map format and encoded with one or two bytes per pixel

Name	Date modified	Type	Size
00131884234126086760.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234127076764.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234127736770.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234128066778.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234128396776.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234128726785.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234129056783.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234129386791.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234129716789.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234130046797.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234130376795.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234130706803.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234131036801.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234131366810.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234131696808.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234132026816.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234132356814.pgm	12/4/2018 1:56 PM	PGM File	301 KB
00131884234132686822.pgm	12/4/2018 1:56 PM	PGM File	301 KB

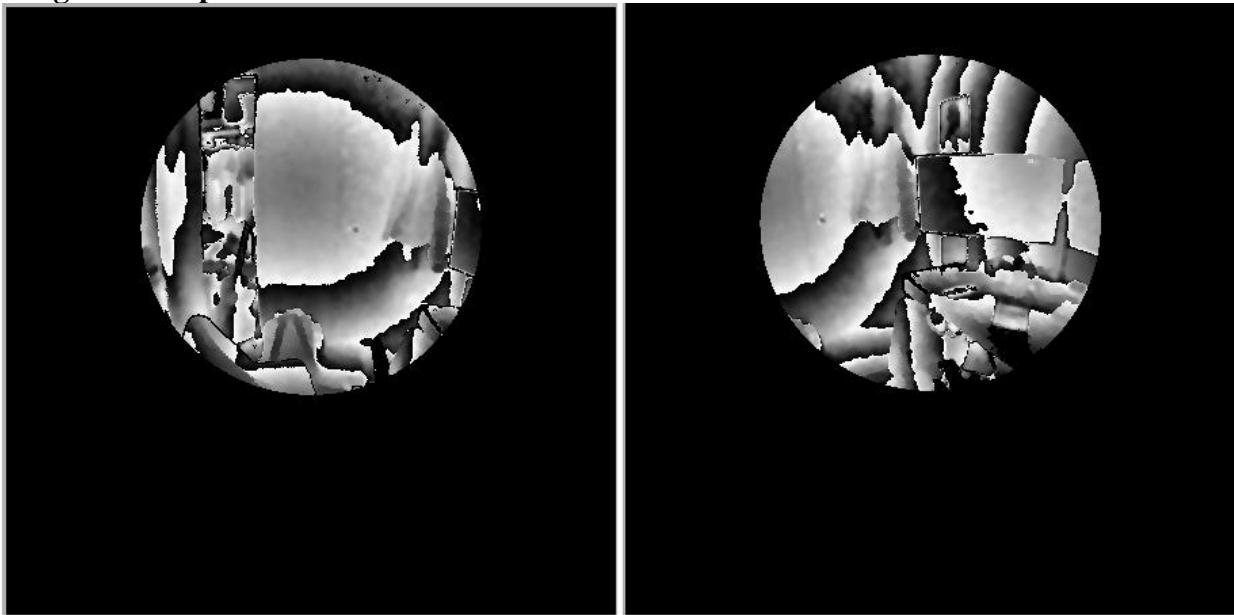
PGM image opened using File Viewer Plus

Link to the image viewer app:

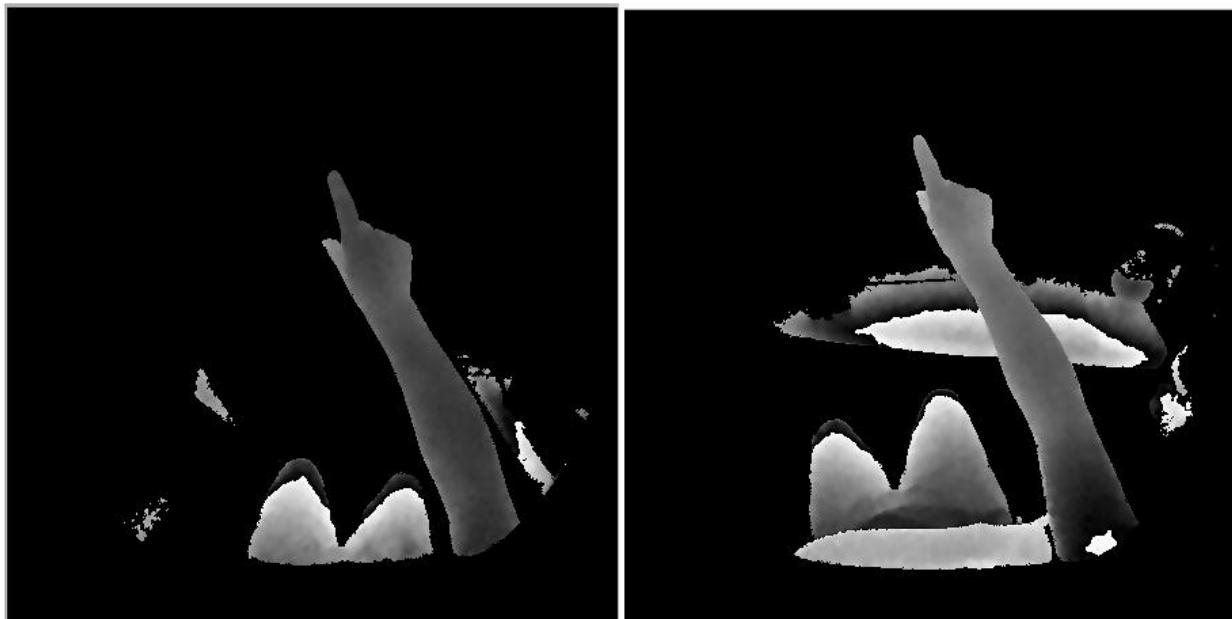
<https://fileviewerplus.com/>



Long throw depth PGM



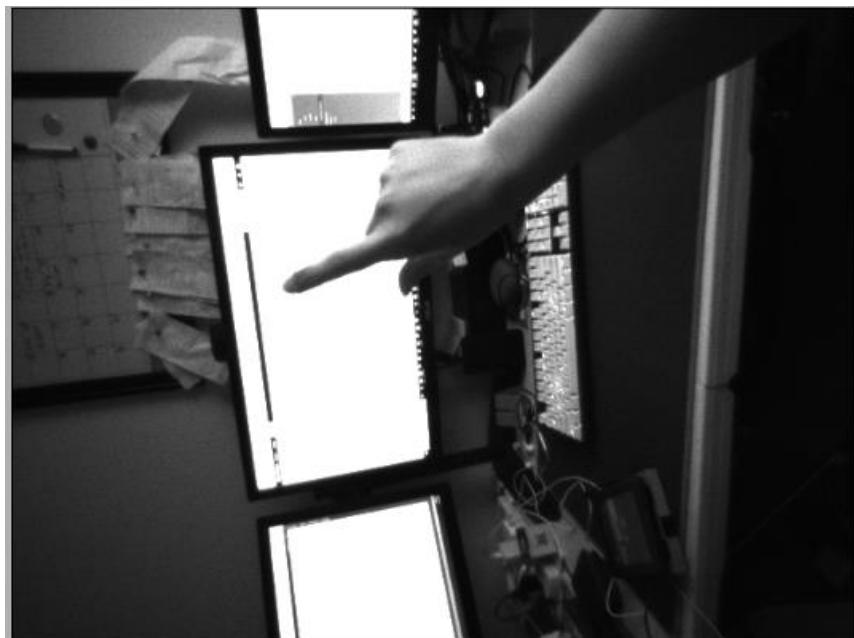
Short Throw Depth



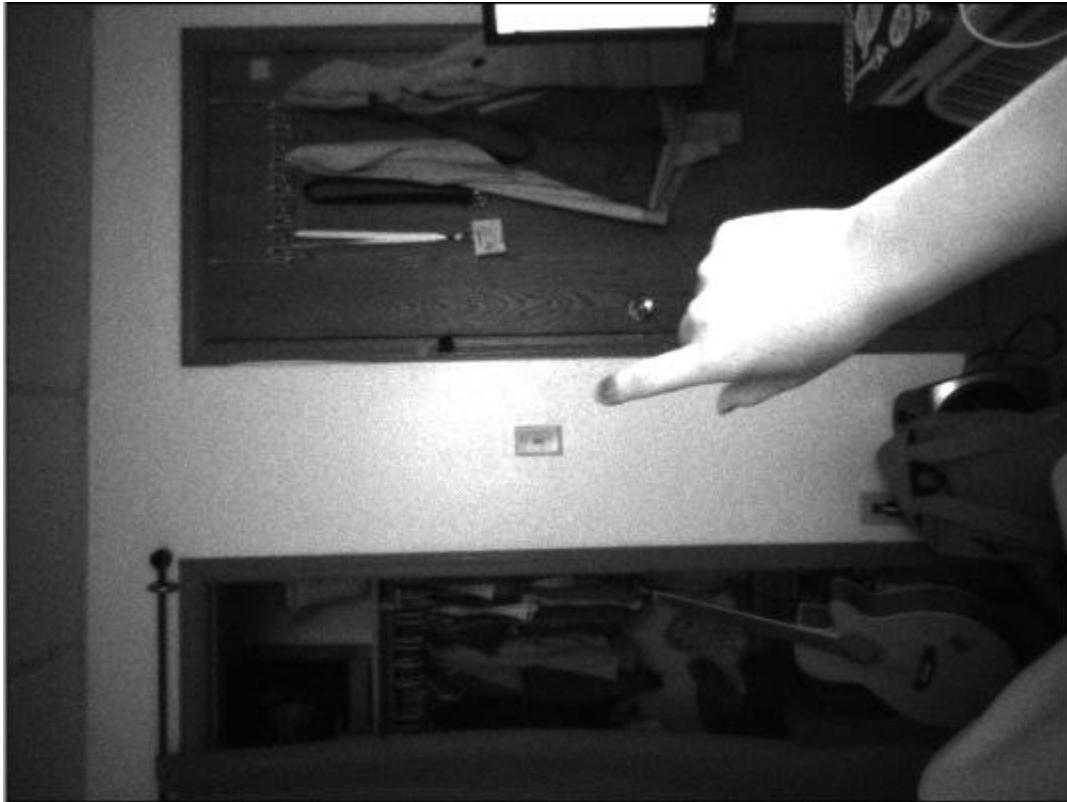
Short Throw Reflectivity



VLC_LF



VLC_RF



Links to possible solutions/similar discussions

Discussion about temporary key

missing: <https://forums.hololens.com/discussion/1405/certificate-file-missing-error-when-building-sessionmanageruniversal-ui>

Generate certificate: [https://docs.microsoft.com/en-us/previous-versions/br230260\(v=vs.140\)](https://docs.microsoft.com/en-us/previous-versions/br230260(v=vs.140))

Joined HoloDevelopers' slack to get more information from other developers

<https://holodevelopersslack.azurewebsites.net>

For future reference:

<https://arxiv.org/abs/1802.00278>

Annotate in the real world using hololens

<https://blogs.sap.com/2016/06/09/annotating-the-world-using-microsoft-hololens/>

<https://www.neowin.net/news/vuforias-new-sdk-lets-microsofts-hololens-recognize-real-world-objects/>

Week 29th Jan 2019 - 5 Feb 2019

Objective

Figure out how to use the data from the crack inspection algorithm and project it into the real world space using Hololens

Computer Vision APIs

Found example of **analyzing visual content** using HoloLens, Computer Vision APIs and Unity on Davide Zordan's blog

Link: <https://www.davidezordan.net/blog/?p=8234>

Inside the blog, there's a link to a page where Microsoft uses Cognitive Services APIs to recognize human's face.

Link: <https://blogs.windows.com/buildingapps/2017/02/13/cognitive-services-apis-vision/#kGJJ5dKM9yWTOhjD.97>

Sending image to python server and back to HoloLens using TCP

Found some discussions about people trying to do the same thing, sending captured image to a python server on a computer and the computer would do heavy processing and return the processed image to the HoloLens.

Keywords: HoloLens TCP, convert texture to png/jpg byte array and send to an Unity app with TCP.

Stack Overflow link: <https://stackoverflow.com/questions/51214972/image-processing-in-python-from-unity-hololens-development>

Byte array link: <https://stackoverflow.com/questions/42717713/unity-live-video-streaming/42727918#42727918>

OpenCV

Integrate OpenCV library into Unity and try to use its library to draw rectangle and show it through unity and then HoloLens

Related Links:

- Passing detection data to unity but there is no visual overlay
 - link: <http://thomasmountainborn.com/2017/03/05/unity-and-opencv-part-three-passing-detection-data-to-unity/>
- C# Wrapper of OpenCV for UWP applications
 - <https://github.com/Firifire/OpenCvSharp-UWP>
- HoloLens with OpenCV
 - <https://github.com/EnoxSoftware/HoloLensWithOpenCVForUnityExample/blob/master/README.md>

<https://stackoverflow.com/questions/43400447/unity-c-bounding-box-of-object-from-image>

HoloLens OpenCV Video

Video of OpenCV face detection in Unity (User might have to pay to use OpenCV in Unity)

<https://youtu.be/SdzsedkTpCI>

The *repository* of this project:

<https://github.com/EnoxSoftware/HoloLensWithOpenCVForUnityExample/blob/master/README.md>

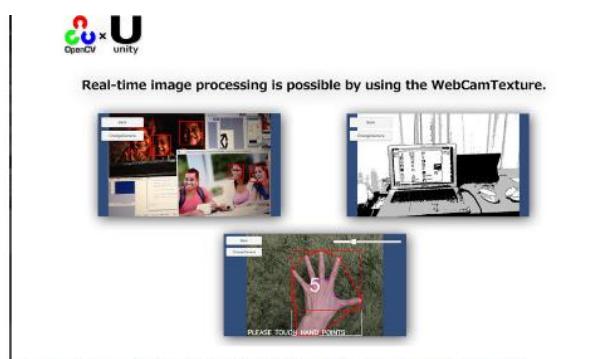
Trying to recreate the project but *OpenCVForUnity* is *required* to proceed

```
Setup the Mixed Reality Toolkit > Configure > Apply Mixed Reality Project Settings
1. Import the OpenCVForUnity.
   * Setup the OpenCVForUnity. (Tools > OpenCV for Unity > Set Plugin Import Settings)
   * Move the "OpenCVForUnity/StreamingAssets/haarcascade_frontalface_alt.xml" and "OpenCVForUnity/StreamingAssets/lbpcascade_frontalface.xml" to the
```

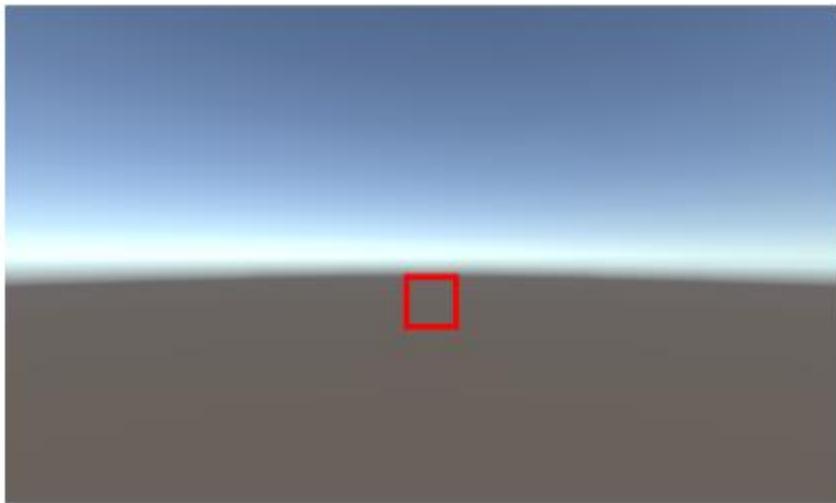
Other requirements:

```
## Environment
* Hololens 10.0.17763.134 (RS5)
* Windows 10 Pro 1803
* Windows 10 SDK 10.0.17134
* Visual Studio 2017 (v157.4.0)
* Unity 2018.3.0f2+
* [HoloToolkit-Unity](https://github.com/Microsoft/MixedRealityToolkit-Unity/releases) 2017.4.3.0
* [OpenCV for Unity](https://assetstore.unity.com/packages/tools/integration/opencv-for-unity-21088?aid=101114ehR) 2.3.3+
* [HoloLensCameraStream](https://github.com/VulcanTechnologies/HoloLensCameraStream)
```

OpenCV inside Unity's asset store



Discover a prefab object that is available inside the repository



It might take some time to understand what the code is doing because of the new syntax and unknown functionalities

Rough solutions (maybe) with Unity:

GUI: Box

It will be on GUI layer and Box will need to include a parameter of "Rect"

<https://docs.unity3d.com/ScriptReference/GUI.Box.html>

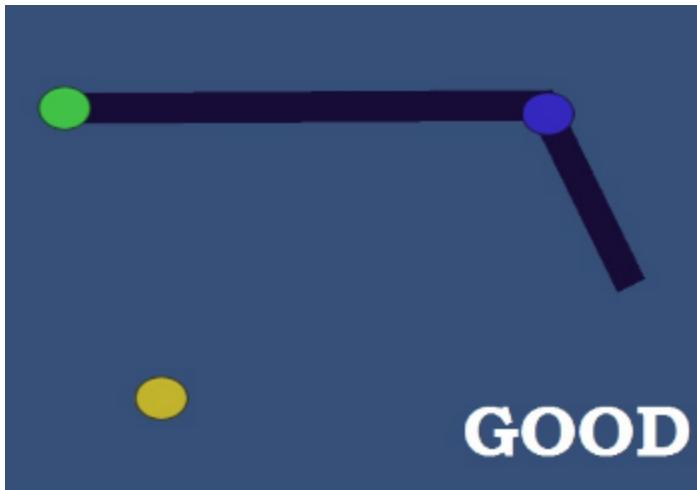
Rect: 2D Rectangle defined by X and Y position, width and height

It can contain text, image or combination of two

<https://docs.unity3d.com/ScriptReference/Rect.html>

LineRenderer: Render Lines in Unity

Retrieve the coordinates from the algorithm, create objects at the respective coordinates and generate lines in between them using LineRenderer in Unity



World Anchor

Physical locations in the real world, so the object can be placed at the same place after application is relaunch

HoloLens WorldAnchor persistence

Persistence is a system for saving [World Anchor](#) states across multiple runs of the same application. This can be thought of as a "save game" functionality for physical locations in the real world. An example of this is remembering where a game board is placed in the world when the application re-launches.

The `WorldAnchorStore` provides basic functionality for saving and loading World Anchors. Retrieve a `WorldAnchorStore` by calling `WorldAnchorStore.GetAsync` and providing it with a callback. Your callback saves the `WorldAnchorStore` returned so that it can be used for future operations.

To save an existing World Anchor, give it a name and call the `Save` function on the `WorldAnchorStore`. See an example below:

```
private void SaveAnchor()
{
    if (!this.savedAnchor) // only save this once
    {
        this.savedAnchor = this.MyWorldAnchorStore.Save("MyAnchor", MyWorldAnchor);
        if (!this.savedAnchor)
        {
            // Anchor failed to save to the store.
            // Handle errors here.
        }
    }
}
```

Week 6 Feb 2019 - 12 Feb 2019

Using Microsoft Machine Learning Object Detection Algorithm

<https://docs.microsoft.com/en-us/windows/mixed-reality/mr-azure-310#exercise-3>

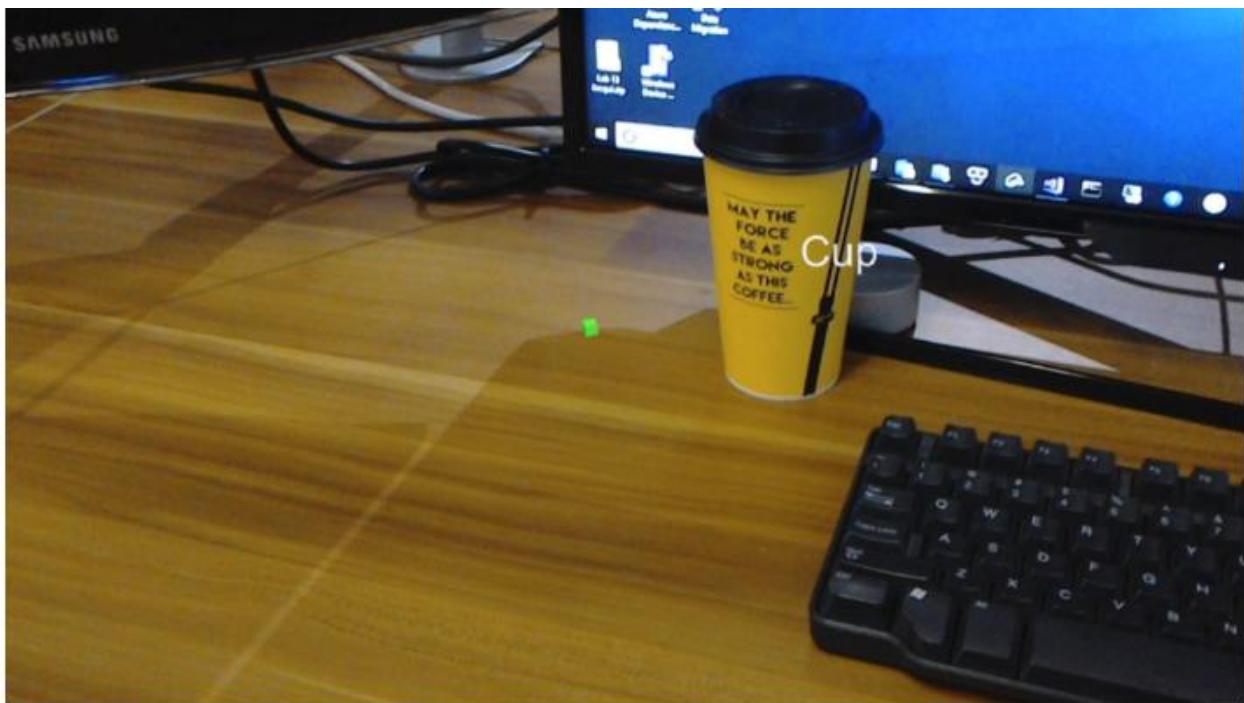
Link to the azure website: <https://www.customvision.ai/projects>

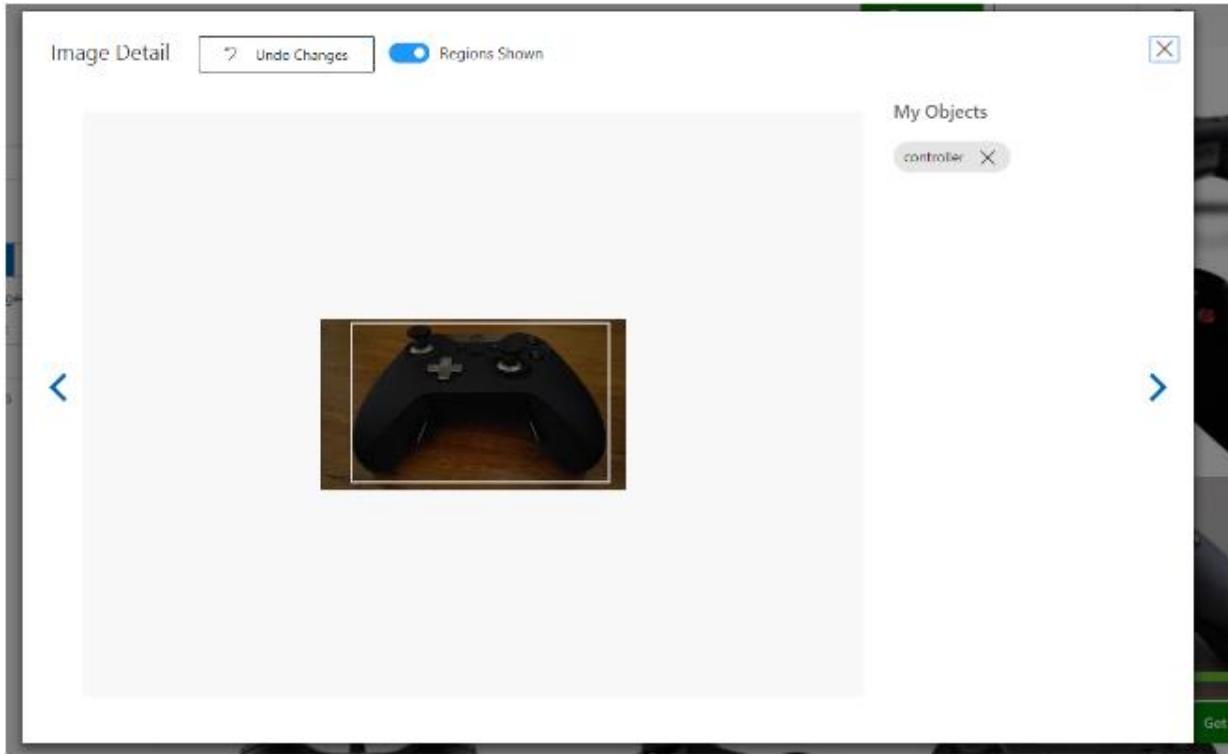
What it does

The Custom Vision service uses a machine learning algorithm to classify images. You, the developer, must submit groups of images that feature and lack the classification(s) in question. You specify the correct tags of the images at the time of submission. Then, the algorithm trains to this data and calculates its own accuracy by testing itself on that same data. Once the model is trained, you can test, retrain, and eventually use it to classify new images according to the needs of your app. You can also export the model itself for offline use.

Classification and object detection

Custom Vision functionality can be divided into two features. **Image classification** assigns a distribution of classifications to each image. Both multiclass (one tag per image) and multilabel (any number of tags per image) classification models are supported. **Object detection** is similar to multilabel classification, but it also returns the coordinates in the image where the applied labels can be found.





```

public Vector3 CalculateBoundingBoxPosition(Bounds b, BoundingBox boundingBox)
{
    Debug.Log($"BB: left {boundingBox.left}, top {boundingBox.top}, width {boundingBox.width}, height {boundingBox.height}");

    double centerFromLeft = boundingBox.left + (boundingBox.width / 2);
    double centerFromTop = boundingBox.top + (boundingBox.height / 2);
    Debug.Log($"BB CenterFromLeft {centerFromLeft}, CenterFromTop {centerFromTop}");

    double quadWidth = b.size.normalized.x;
    double quadHeight = b.size.normalized.y;
    Debug.Log($"Quad Width {b.size.normalized.x}, Quad Height {b.size.normalized.y}");

    double normalisedPos_X = (quadWidth * centerFromLeft) - (quadWidth / 2);
    double normalisedPos_Y = (quadHeight * centerFromTop) - (quadHeight / 2);

    return new Vector3((float)normalisedPos_X, (float)normalisedPos_Y, 0);
}

{"probability":0.0157823022,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.363956362,"top":0.19885768,"width":0.05111608,"height":0.05111608,"he
{"probability":0.0366205675,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.392357647,"top":0.206000984,"width":0.0601274371,
{"probability":0.0140684014,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.358094633,"top":0.325282276,"width":0.0442119241,
{"probability":0.0146229034,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.315674663,"top":0.720932543,"width":0.02786556483,
{"probability":0.0173829626,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.4306249,"top":0.02986288,"width":0.118659079,"height":0.02986288,"he
{"probability":0.0379529931,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.01262755048,"top":0.105515316,"width":0.1184770468,
{"probability":0.0113645159,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.22804755,"top":0.179073721,"width":0.122574687,"height":0.122574687,"he
{"probability":0.0193091463,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.5741858,"top":0.558101356,"width":0.1548089892,"height":0.1548089892,"he
{"probability":0.0209660446,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.704292953,"top":0.0, "width":0.291564167,"height":0.291564167,"he
{"probability":0.0146362325,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.264441043,"top":0.3703287,"width":0.428353518,"height":0.428353518,"he
{"probability":0.01838748874,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.121423781,"top":0.413218468,"width":0.527761161,"height":0.527761161,"he
{"probability":0.0115736146,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.442666918,"top":0.498317361,"width":0.524659634,"height":0.524659634,"he
{"probability":0.0371826775,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.0799863742,"top":0.5637804,"width":0.526638831,"height":0.526638831,"he
{"probability":0.820757,"tagId":"da8937d5-b1e8-49e5-bcc7-00f16fb9d387","tagName":"controller","boundingBox":{"left":0.296058416,"top":0.528003454,"width":0.441256881,"height":0.441256881,"he
}

```

```

//create rect
Instantiate(rect, cursor.transform.position, transform.rotation);
//rect.transform.localPosition = new Vector3(0.0f, 0.0f, 0.1f);
rect.transform.localPosition = new Vector3(1f, 1f, 0f);
rect.transform.localScale = new Vector3(0.0001f, 0.0001f, 0.0001f);

// Set the texture

```

It seems like localPosition is not affecting its position, only when the position when it is being instantiated will affect its position in real world space

In this case the cursor.transform.position is the position of the bounding box

Also, the size of the object should be set before instantiating it

Video Demo



After numerous of attempts



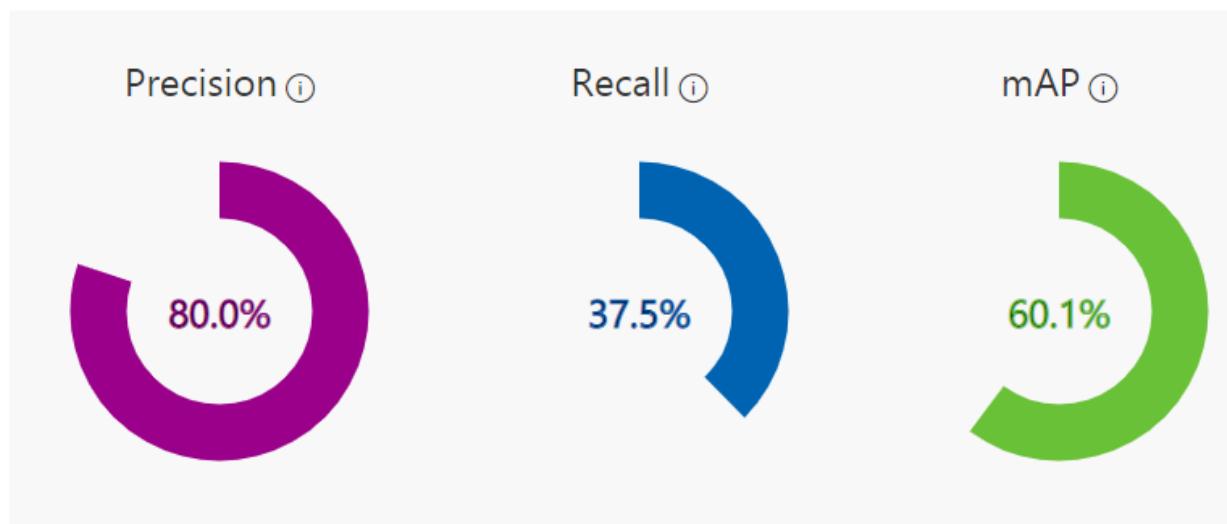
2.mp4

Week 13 Feb 2019 - 19 Feb 2019

Working on the crack inspection by training the crack data using Microsoft Azure Custom Vision AI service.

The precision rate and recall is not high enough

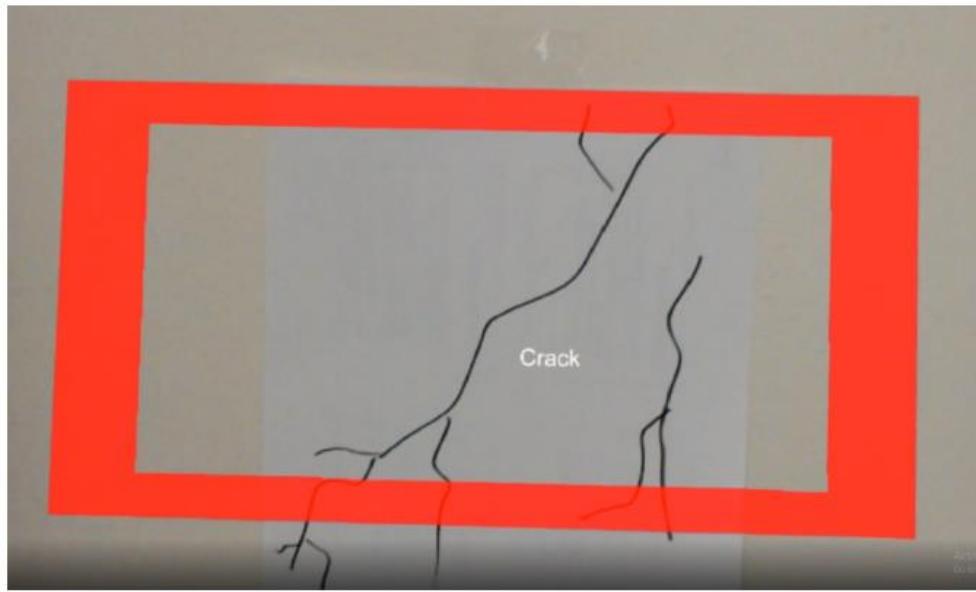
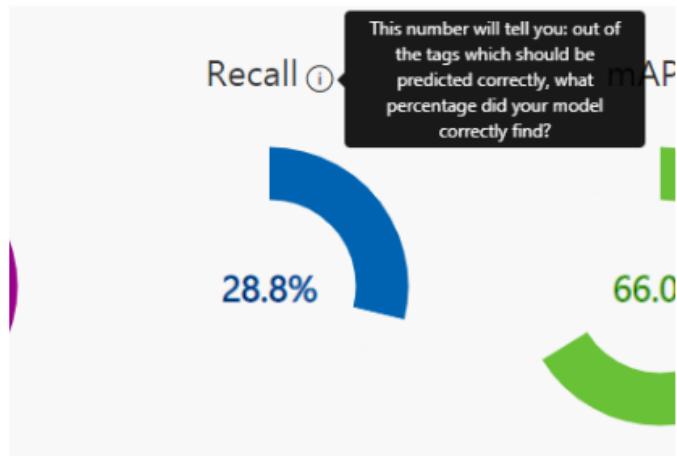
Finished training on **2/19/2019, 4:15:10 PM** using **General** domain



Performance Per Tag

Tag	Precision	Recall	A.P.	Image count
Crack	80.0%	37.5%	60.1%	206

41:24 PM using General domain



Video Demo



trim near.mp4



trim far.mp4

Maybe this kind of sample can improve the accuracy



Week 20 Feb 2019 - 26 Feb 2019

Plane finding

Idea: Trying to integrate the plane finding functionality so I could find the distance of the plane from the generated bounding box in the 3D virtual space

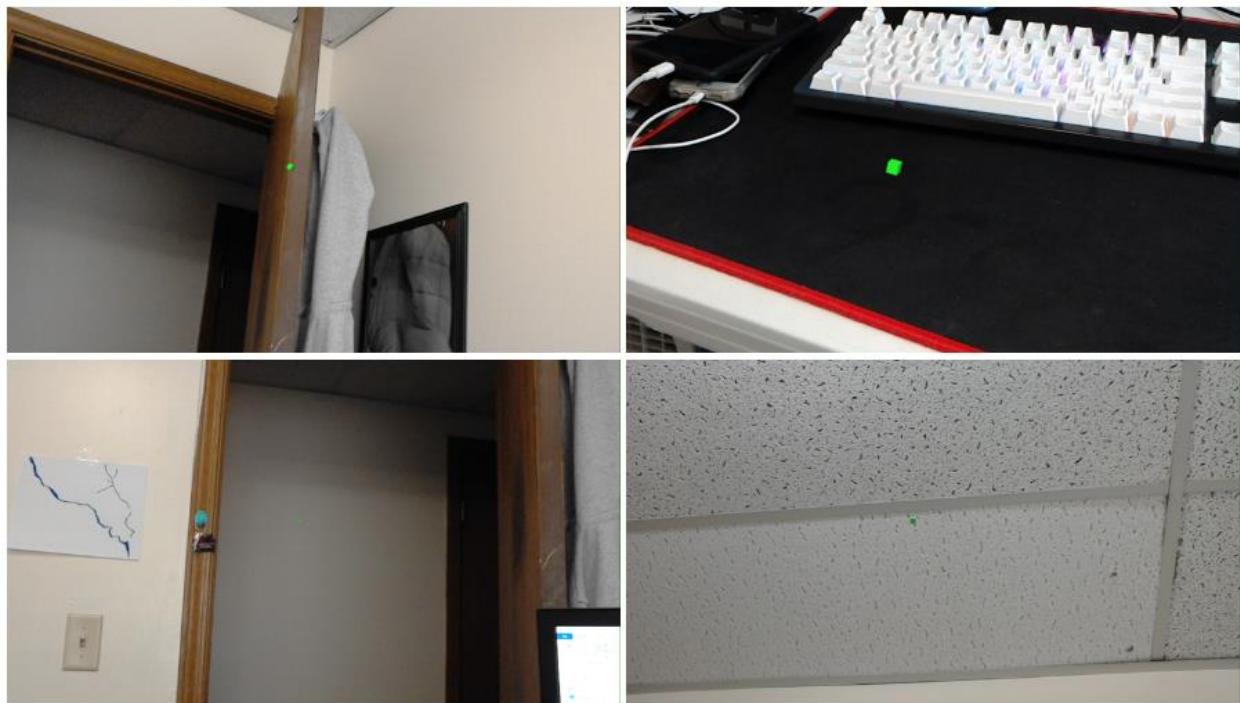
Encounter compilation errors:

```
Exception thrown: 'System.DllNotFoundException' in Assembly-CSharp.dll
Exception thrown: 'System.AggregateException' in System.Private.CoreLib.ni.dll
AggregateException: One or more errors occurred. (Unable to load DLL 'PlaneFinding': The specified module could not be found. (Exception from HRESULT: 0x8007007E))
  at System.Threading.Tasks.Task`1.GetResultCore(Boolean waitCompletionNotification)
  at System.Threading.Tasks.Task`1.get_Result()
  at Academy.HoloToolkit.Unity.SurfaceMeshesToPlanes.<MakePlanesRoutine>d__24.MoveNext()
  at UnityEngine.SetupCoroutine.InvokeMoveNext(IEnumerator enumerator, IntPtr returnValueAddress)
  at UnityEngine.SetupCoroutine.$InvokeL_invokeMoveNext(Int64 instance, Int64* args)
  at UnityEngine.Internal.$MethodUtility.InvokeMethod(Int64 instance, Int64* args, IntPtr method)
(Filename: <Unknown> Line: 0)
```

Gaze Cursor with depth perception

Realized that some scripts were actually not working in last session of testing

Integrated a cursor that indicates the depth of the spatial space when looking around



Demo with the updated version:



Look into channel 9, Azure for next week

Week 12 March 2019 - 19 March 2019

Client server connection for Microsoft Hololens

Implement using .NET framework

UnityWebRequest

class in UnityEngine.Networking / Implemented in:UnityEngine.UnityWebRequestModule

[Leave feedback](#) [Other Versions](#)

Description

The UnityWebRequest object is used to communicate with web servers.

UnityWebRequests handle the flow of HTTP communication with web servers. Other objects - specifically [DownloadHandler](#) and [UploadHandler](#) - govern downloading and uploading data, respectively.

A set of static functions are provided for convenience; these return UnityWebRequest objects properly configured for many common use cases. See: [Get](#), [Post](#), [Put](#), [GetTexture](#).

Note: Most properties on the UnityWebRequest object cannot be altered once the UnityWebRequest has begun communicating with a remote server by calling the [Send](#) method.

Sample code to connect to an IP address, server endpoint and send buffer data

```
1 using UnityEngine;
2 using System.Collections;
3 using System.Net.Sockets;
4 using System.Net;
5
6 public class Client : MonoBehaviour {
7
8     public string serverAddress = "127.0.0.1";
9     public const int serverPort = 32211;
10    public bool isConnected = false;
11
12    private static Client singleton;
13    private Socket sServer;
14
15    void Awake() {
16        sServer = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
17        IPAddress remoteIPAddress = IPAddress.Parse(serverAddress);
18        IPEndPoint remoteEndPoint = new IPEndPoint(remoteIPAddress, serverPort);
19        singleton = this;
20        sServer.Connect(remoteEndPoint);
21    }
22
23    void Update() {
24        if(isConnected != sServer.Connected)
25            isConnected = sServer.Connected;
26    }
27
28    void OnApplicationQuit() {
29        sServer.Close();
30        sServer = null;
31    }
32
33    static public void Send(MessageData msgData) {
34        if(singleton.sServer == null) {
35            return;
36        }
37        byte[] sendData = MessageData.ToByteArray(msgData);
38        byte[] buffer = new byte[1];
39        buffer[0] = (byte)sendData.Length;
40        singleton.sServer.Send(buffer);
41        singleton.sServer.Send(sendData);
42    }
43
44 }
```

Link

<https://www.youtube.com/watch?v=l2ewuyQVZMg>

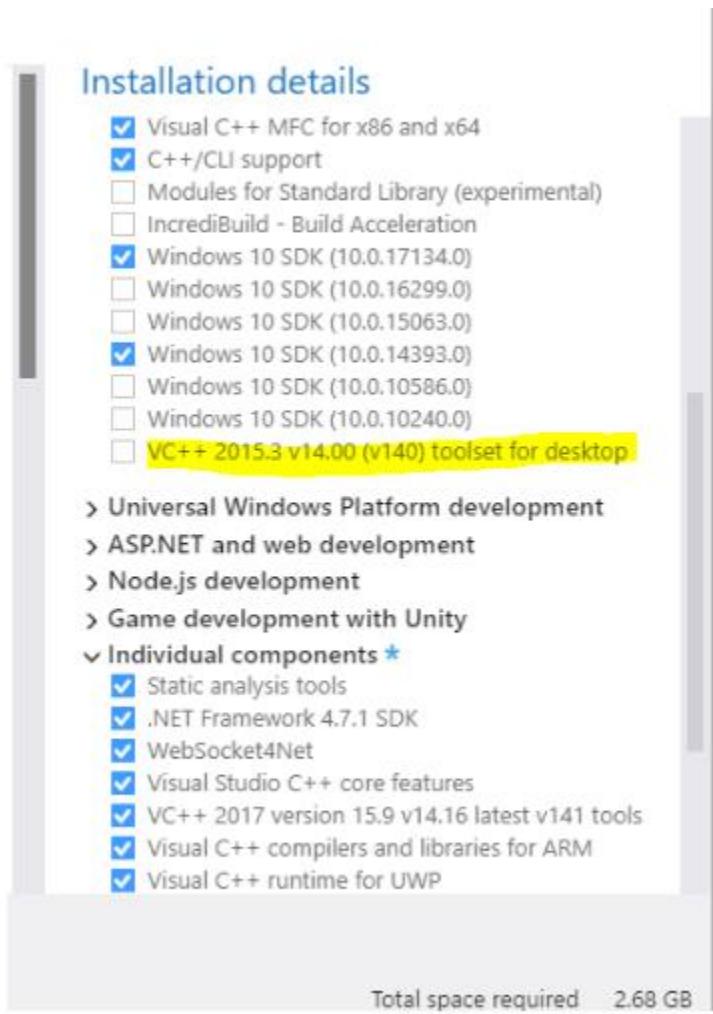
Week 19 March 2019 - 26 March 2019

Testing RemoteHostingSample from microsoft mixed reality companion toolkit git repo

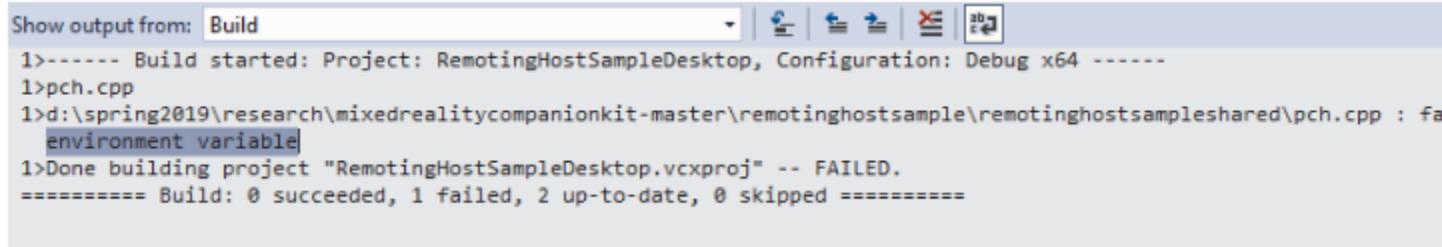
Issues: unable to build

Toolv141 is missing

```
1>----- Build started: Project: RemotingHostSampleDesktop, Configuration: Debug x64 -----  
1>C:\Program Files (x86)\MSBuild\Microsoft.Cpp\v4.0\V140\Microsoft.Cpp.Platform.targets(57,5): error MSB8020: The build tools for v141 (Platform Toolset = 'v141') cannot be found. To build using  
===== tools, please install v141 build tools. Alternatively, you may upgrade to the current Visual Studio tools by selecting the Project menu or right-click the solution, and then selecting "Retar  
===== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped ======
```



Another problem:



```
Show output from: Build
1>----- Build started: Project: RemotingHostSampleDesktop, Configuration: Debug x64 -----
1>pch.cpp
1>d:\spring2019\research\mixedrealitycompanionkit-master\remotinghostsamples\remotinghostsampleshared\pch.cpp : fa
  environment variable
1>Done building project "RemotingHostSampleDesktop.vcxproj" -- FAILED.
===== Build: 0 succeeded, 1 failed, 2 up-to-date, 0 skipped =====
```

Week 27 March 2019 - 2 April 2019

Web Socket in C#

Client

The code below acts as client that sends data

The screenshot shows a Windows application window titled "client_tutorial". Inside the window, there is a code editor displaying C# code for a Windows Form application. The code is as follows:

```
intTutorial
client_tutorial.Form1

9 using System.Windows.Forms;
8 using System.Net;
1 using System.Net.Sockets;
2
3
4 namespace client_tutorial
5 {
6     public partial class Form1 : Form
7     {
8         string serverIP = "localhost";
9         int port = 8080;
10
11         public Form1()
12         {
13             InitializeComponent();
14         }
15
16         private void button1_Click(object sender, EventArgs e)
17         {
18             TcpClient client = new TcpClient(serverIP, port);
19
20             int byteCount = Encoding.ASCII.GetByteCount(message.Text + 1); //sent how many byte we want
21
22             byte[] sendData = new byte[byteCount];
23
24             sendData = Encoding.ASCII.GetBytes(message.Text + ";"); //store the message
25
26             //send to the server
27             NetworkStream stream = client.GetStream();
28
29             stream.Write(sendData, 0, sendData.Length); //send entire message
30
31             //right now the remaining length is filled up with null so we need to stop the byte after its not detecting characters
32
33             stream.Close();
34             client.Close();
35         }
36     }
37 }
```

Server

The code below acts as the server that receive the data

```
class Program
{
    static void Main(string[] args)
    {
        IPAddress ip = Dns.GetHostEntry("localhost").AddressList[0]; //connecting to ip address, localhost
        TcpListener server = new TcpListener(ip, 8080);
        TcpClient client = default(TcpClient);

        try
        {
            server.Start();
            Console.WriteLine("Server started...");
        }catch(Exception ex)
        {
            Console.WriteLine(ex.ToString());
            Console.Read();
        }

        //infinite loop
        while (true)
        {
            client = server.AcceptTcpClient();

            byte[] receivedBuffer = new byte[100];
            NetworkStream stream = client.GetStream(); //get stream of the client, send it over through byte

            stream.Read(receivedBuffer, 0, receivedBuffer.Length);

            StringBuilder msg = new StringBuilder();

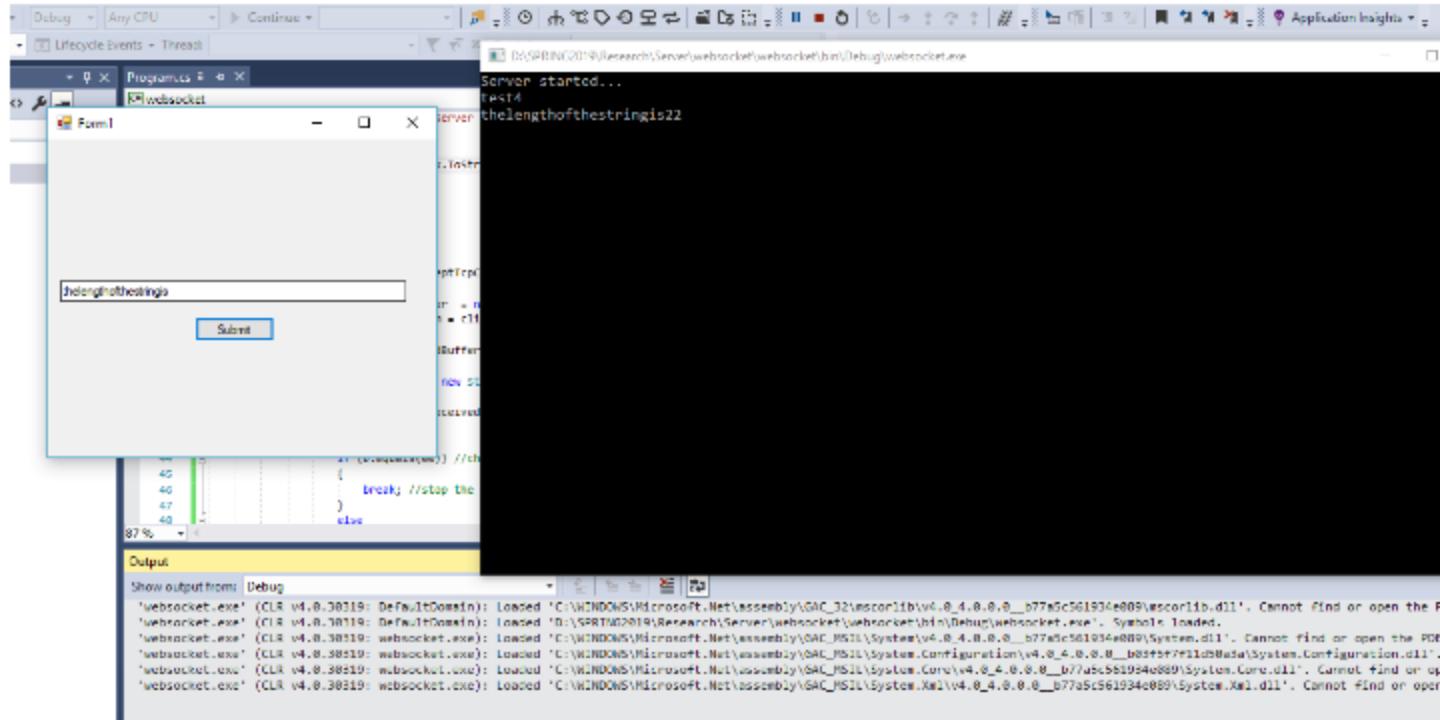
            foreach(byte b in receivedBuffer)
            {
                //ASCII for ";"
                if (b.Equals(00)) //check null
                {
                    break; //stop the looping when it find ";"
                }
                else
                {
                    msg.Append(Convert.ToChar(b).ToString());
                }
            }

            Console.WriteLine(msg.ToString() + msg.Length);
        }
    }
}
```

Code in action

Sending the string in the form of byte array from client after typing into the textbox and pressing the submit button. The server will receive the stream of data

and break the string once it reads a null value, displaying the text and also the length of the text.



Week 3 April 2019 - 9 April 2019

Go back to MixedRealityCompanionKit Remote Host Sample

Multiple build errors:

Had to delete and reclone repo

Uninstall visual studio 2017 and attempted to reinstall visual studio 2015 to test out the app

Unable to download Visual Studio 2015, requires the purchase of Visual Studio Subscription

Platform tool v140 missing

Output

Show output from: Build

```
1>----- Build started: Project: RemotingHostSampleUWPXAML, Configuration: Debug Win32 -----
1>C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\Common7\IDE\VC\VCTargets\Microsoft.Cpp.Platform
To build using the v140 build tools, please install v140 build tools. Alternatively, you may upgrade to the
selecting "Retarget solution".
1>Done building project "RemotingHostSampleUWPXaml.vcxproj" -- FAILED.
=========
Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped
=====
Deploy: 0 succeeded, 0 failed, 0 skipped =====
```

Updated VS2017 to latest and everything failed

Tried re-targeting the solution to Platform v140 but build is still not successful

Sample - Microsoft Visual Studio

Project Build Debug Team Tools Test Analyze Window Help

Debug x86 RemotingHostSampleUWPXAML (i Local Machine Native Only (Global Scope)

File (Ctrl+.) RemotingHostSample (4 projects) RemotingHostSampleDesktop (Global Scope)

ces Dependencies Files cs.config RemotingHostSampleDesktop.cpp RemotingHostSampleDesktop.h er.h RemotingHostShared RemotingHostUWP (Universal Windows) RemotingHostUWPXAML (Universal Windows)

16
17 #include <vector>
18 #include <Microsoft.Perception.Simulation.h>
19 #include <strsafe.h>
20
21 #define WINDOWCLASSNAME L"RemotingHostSampleDesktopWindowClass"
22 #define WINDOWNAME L"RemotingHostSampleDesktop"
23 #define TITLE_SEPARATOR L" | "
24 #define TITLE_CONNECT_TEXT L"PI"
25 #define TITLE_DISCONNECT_TEXT L"PI"
26 #define TITLE_ENABLE_PREVIEW_TEXT L"PI"
27 #define TITLE_DISABLE_PREVIEW_TEXT L"PI"
28
29 using namespace Microsoft::WRL;
30 using namespace Microsoft::WRL::Wrapper;
31 using namespace Microsoft::Holographic;
32 using namespace Windows::Graphics::Holo;
33 using namespace std;
34
35 LRESULT CALLBACK wndProc(HWND hwnd, UINT
36 {
37 static DesktopWindow* s_desktopWindow;
38
39 LRESULT result = 0;
40
41 switch (msg)
42 {
43 case WM_CREATE:
44 {
45 CREATESTRUCT* cs = reinterpret_cast<CREATESTRUCT*>(lParam);
46 s_desktopWindow = reinterpret_cast<DesktopWindow*>(cs->lpCreateParams);
47 }
48 }
49
50 return result;
51 }

86 %

Output

Show output from: Build

```
i>----- Build started: Project: RemotingHostSampleDesktop
i>C:\Program Files (x86)\Microsoft Visual Studio\14.0\BuildTools\MSBuild\14.0\Bin\Microsoft.Common.CurrentVersion.targets(103,5): To build using the v140 build tools, please select "Retarget solution".
1>Done building project "RemotingHostSampleDesktop" (target(s): RemotingHostSampleDesktop).
----- Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped
----- Deploy: 0 succeeded, 0 failed, 0 up-to-date
```

Review Solution Actions

Retarget Projects

Select Windows SDK version for the following projects

Windows SDK Version: 10.0.17763.0

..\RemotingHostSampleDesktop\RemotingHostSampleDesktop.vcxproj

Retarget Projects

Select Platform Toolset, Windows SDK Version, and Minimum version for the following projects. Upgrading your projects to a later toolset will prevent them from building in Visual Studio 2015. To build upgraded projects in Visual Studio 2015, upgrade the toolset to v140 in the project property page. This dialog can be re-launched by right-clicking the solution and selecting "Retarget Solution."

Windows SDK Version: 10.0.17763.0

Windows SDK Min. Version: 10.0.17763.0

Platform Toolset: Upgrade to v141

..\RemotingHostSampleUWP\RemotingHostSampleUWP.vcxproj

```
Show output from: Build * | = | + | - | X | Help | Help  
1>----- Build started: Project: RemotingHostSampleDesktop, Configuration: Debug Win32 -----  
2>----- Build started: Project: RemotingHostSampleUMP, Configuration: Debug Win32 -----  
3>----- Build started: Project: RemotingHostSampleUMPXAML, Configuration: Debug Win32 -----  
3>C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\Common7\IDE\VC\VCTargets\Microsoft.Cpp.Platform.targets(67,5): error MSB8020: The build tools for  
To build using the v140 build tools, please install v140 build tools. Alternatively, you may upgrade to the current Visual Studio tools by selecting the Project  
selecting "Retarget solution".  
3>Done building project "RemotingHostSampleUMPXAML.vcxproj" -- FAILED.  
2>C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\Common7\IDE\VC\VCTargets\Microsoft.Cpp.Platform.targets(67,5): error MSB8020: The build tools for  
To build using the v140 build tools, please install v140 build tools. Alternatively, you may upgrade to the current Visual Studio tools by selecting the Project  
selecting "Retarget solution".  
2>Done building project "RemotingHostSampleUMP.vcxproj" -- FAILED.  
1>pch.cpp  
1>d:\spring2019\research\mixedrealitycompanionkit\remotinghostsampleshared\pch.cpp : fatal error C1107: could not find assembly 'platform.winmd'  
AI or by setting the LIBPATH environment variable  
1>Done building project "RemotingHostSampleDesktop.vcxproj" -- FAILED.  
===== Build: 0 succeeded, 3 failed, 0 up-to-date, 0 skipped ======br/>===== Deploy: 0 succeeded, 0 failed, 0 skipped ======
```

Uninstall v141 and install v140 but now it says project is missing v141

fatal error C1107: could not find assembly 'platform.winmd'

Even after installing v140 using the installer, it still shows that it is uninstalled in the visual studio editor

Week 10 April 2019 - 16 April 2019

Moved on to the HoloLens Camera Stream repo

<https://github.com/VulcanTechnologies/HoloLensCameraStream>

When the app runs, you can see a stream of images showing up inside hololens in a box



Now is to figure out how to integrate this into our project

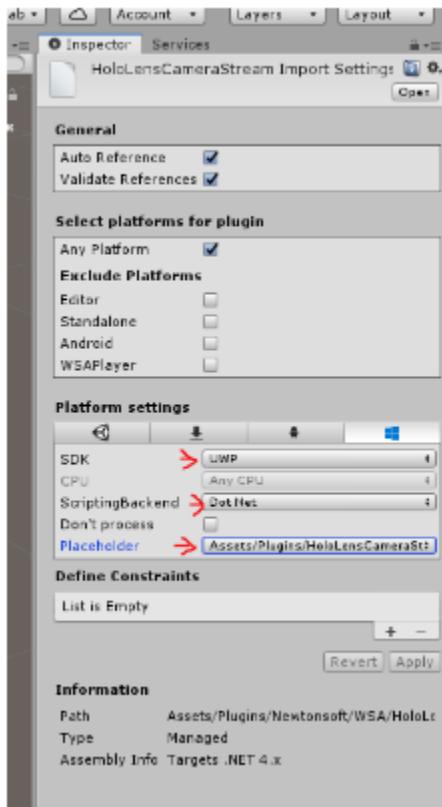
Integrating the plugin into a Unity project

If you made some changes to the plugin project, and you want to use your newly-built DLLs, follow these steps. If you haven't made changes to the plugin, but just want to learn how to *use the CameraStream plugin in Unity*, skip this part and read instead about running the example project.

1. **Find the DLLs you just build:** Look in the output window after you build the plugin solution. You will see two paths to the newly-build DLLs. Navigate to them.
2. **Paste the plugin DLL into Unity:** Copy the plugin DLL from the output directory and paste it into the your Unity app. It must be pasted into the `Assets/Plugins/MSA/` directory in your Unity project because it will only compile for WSA devices.
3. **Paste the placeholder DLL Into Unity:** Copy the placeholder (dummy) DLL from its output directory and paste it into your Unity app. It must be in the `Assets/Plugins/` directory in your Unity project. This is the DLL that the Unity editor compiles against uses while you're coding in Unity.
4. **Edit the plugin's settings:** In the Unity editor, select the `HoloLensCameraStream.dll` plugin file that you pasted in step 2. In the inspector, uncheck all platforms except `WSAP1ayer`. Set the SDK to `uWP`, and set the Scripting Backend to `Dot Net`. In the Placeholder dropdown, select the `HoloLensCameraStream.dll` option (it will likely be the only option).
5. **Click Apply.**
6. **Edit the placeholder plugin's settings:** Select the placeholder (dummy) plugin in `Assets/Plugins/HoloLensCameraStream.dll`. Uncheck all platforms except `Editor`.
7. **Click Apply.**

You should now be able to code against the `HoloLensCameraStream` plugin after importing said namespace in your Unity scripts. Note: Your Unity project needs to be *appropriately configured for HoloLens development*.

Build the HoloLensCameraStream.sln to generate two DLLs and copy them to the project



```
/// Creates environment colliders to work with physics
/// </summary>
private SpatialMappingCollider spatialMappingCollider;

/// <summary>
/// Initializes this class
/// </summary>
private void Awake()
{
    // Allows this instance to behave like a singleton
    Instance = this;
}

/// <summary>
/// Runs at initialization right after Awake method
/// </summary>>apping
void Start()
{
    // Initialize and configure the collider
    spatialMappingCollider = gameObject.GetComponent<SpatialMappingCollider>();
    spatialMappingCollider.surfaceParent = this.gameObject;
    spatialMappingCollider.freezeUpdates = false;
```

After integrating the plugin into an existing project, some of the variable types are not working anymore

! [16:32:26] PrecompiledAssemblyException: Multiple precompiled assemblies with the same name HoloLensCameraStream
UnityEditor.Scripting.ScriptCompilation.EditorBuildRules.CreateTargetAssemblies (System.Collections.Generic.IEnumerable`1[
! [16:32:27] Assets\Scripts\SpatialMapping.cs(24,17): error CS0246: The type or namespace name 'SpatialMappingCollide