HoloLens as a Tool for Computer Vision Research

ECCV 2018

Johannes Schönberger

Microsoft

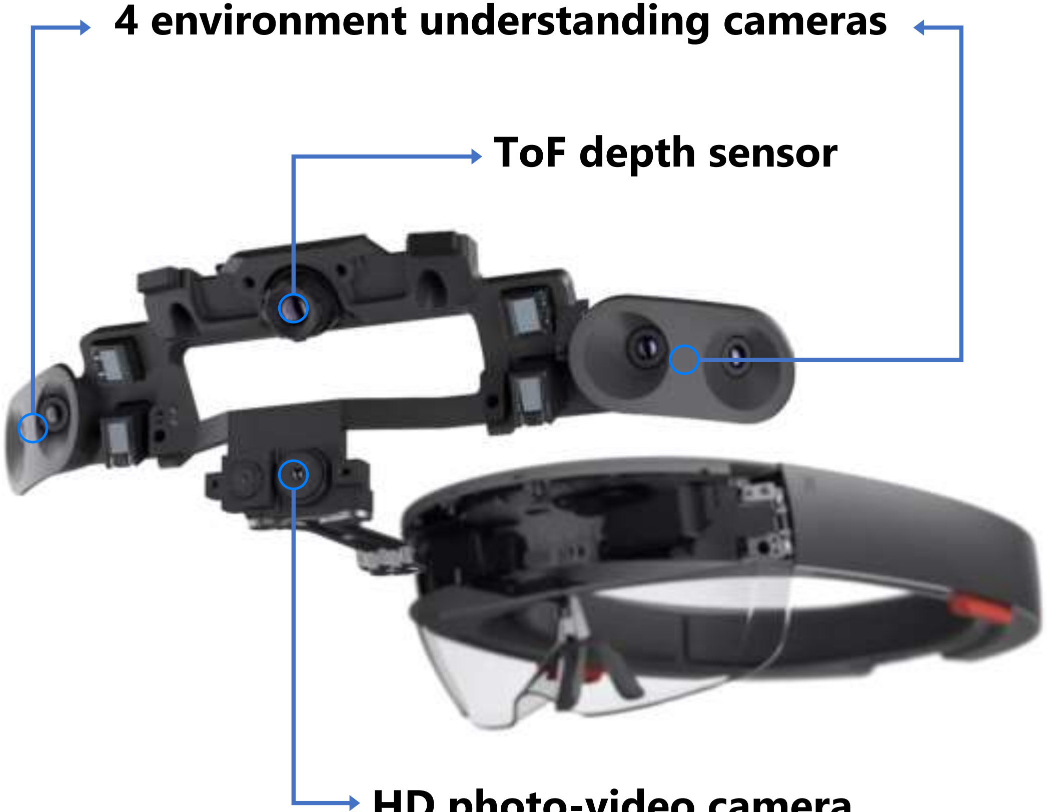
Agenda

1. Setup

2. Usage

3. Applications

Research Mode Sensors



4 environment understanding cameras

ToF depth sensor

HD photo-video camera

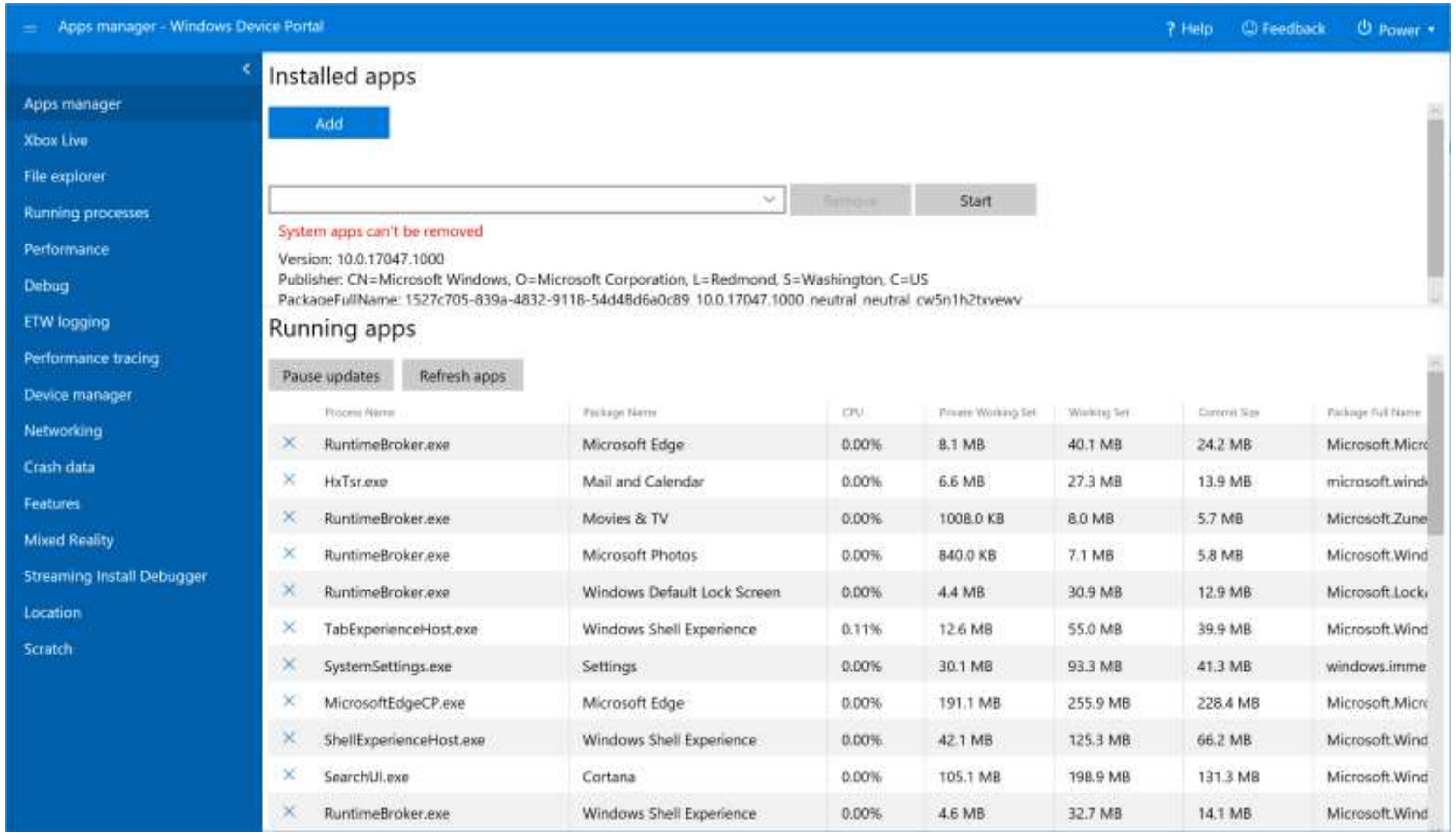
Research Mode Requirements

- **Windows 10 (Version 1803 or newer)**
  - Desktop
  - HoloLens

- **Visual Studio 2017 (Version 15.3 or newer)**
  - Download free copy of community edition
    (also works with professional and enterprise editions)
  - Install C++, UWP, Windows SDK components
    (in the Visual Studio installer)

Device Portal for HoloLens

- Manage HoloLens remotely over WIFI / USB (More Info)



Installed apps

System apps can't be removed

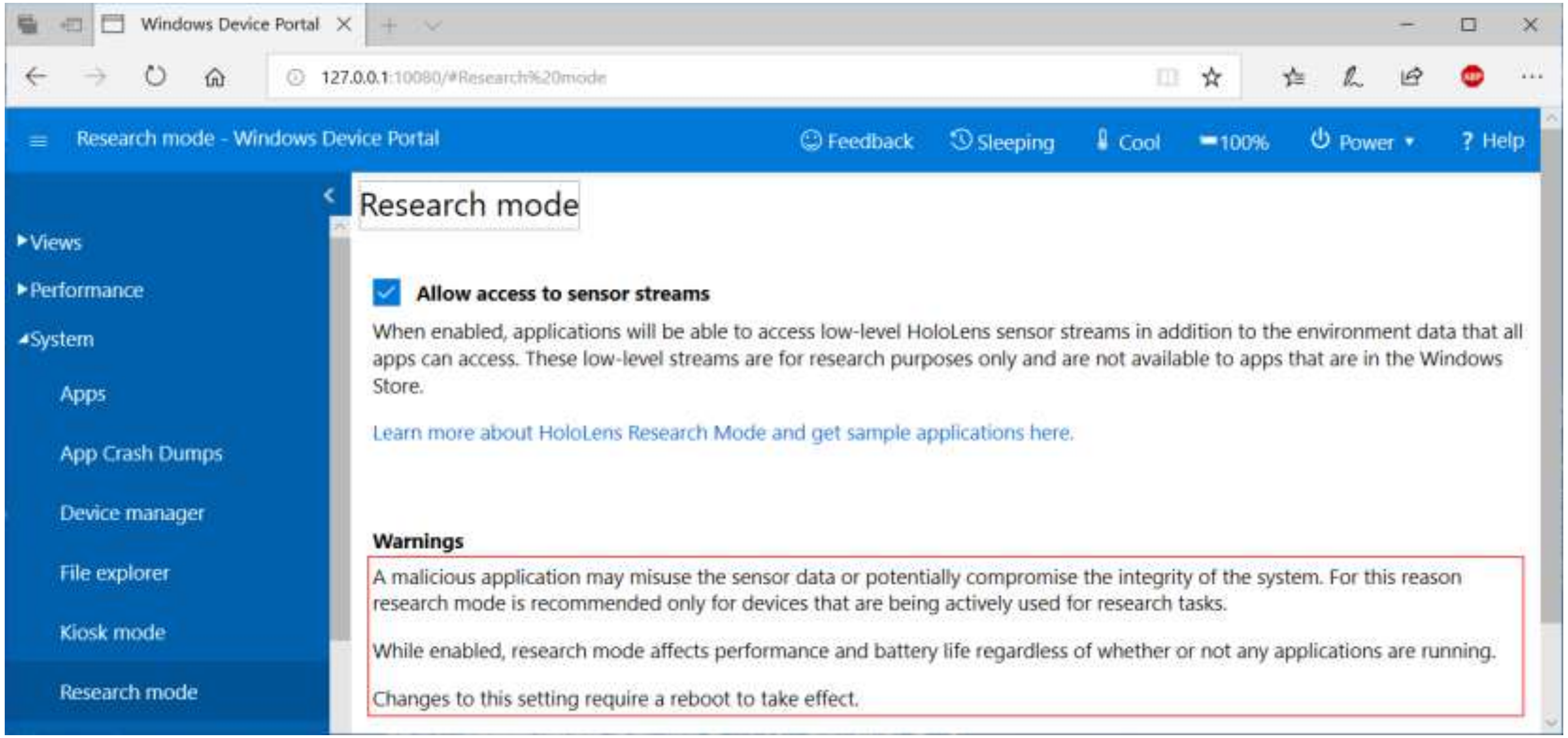
Version: 10.0.17047.1000
Publisher: CN=Microsoft Windows, O=Microsoft Corporation, L=Redmond, S=Washington, C=US
PackageFullName: 1527c705-839a-4832-9118-54d48d6a0c89_10.0.17047.1000_neutral_neutral_cw5n1h2txyewy

Running apps

	Process Name	Package Name	CPU	Private Working Set	Working Set	Commit Size	Package Full Name
×	RuntimeBroker.exe	Microsoft Edge	0.00%	8.1 MB	40.1 MB	24.2 MB	Microsoft.Micr
×	HxTsr.exe	Mail and Calendar	0.00%	6.6 MB	27.3 MB	13.9 MB	microsoft.wind
×	RuntimeBroker.exe	Movies & TV	0.00%	1008.0 KB	8.0 MB	5.7 MB	Microsoft.Zune
×	RuntimeBroker.exe	Microsoft Photos	0.00%	840.0 KB	7.1 MB	5.8 MB	Microsoft.Wind
×	RuntimeBroker.exe	Windows Default Lock Screen	0.00%	4.4 MB	30.9 MB	12.9 MB	Microsoft.Lock
×	TabExperienceHost.exe	Windows Shell Experience	0.11%	12.6 MB	55.0 MB	39.9 MB	Microsoft.Wind
×	SystemSettings.exe	Settings	0.00%	30.1 MB	93.3 MB	41.3 MB	windows.imme
×	MicrosoftEdgeCP.exe	Microsoft Edge	0.00%	191.1 MB	255.9 MB	228.4 MB	Microsoft.Micr
×	ShellExperienceHost.exe	Windows Shell Experience	0.00%	42.1 MB	125.3 MB	66.2 MB	Microsoft.Wind
×	SearchUI.exe	Cortana	0.00%	105.1 MB	198.9 MB	131.3 MB	Microsoft.Wind
×	RuntimeBroker.exe	Windows Shell Experience	0.00%	4.6 MB	32.7 MB	14.1 MB	Microsoft.Wind

Enabling Research Mode

- Device portal makes it easy to enable Research Mode
- Just don't forget to reboot your device afterwards!



Research mode

☑ **Allow access to sensor streams**

When enabled, applications will be able to access low-level HoloLens sensor streams in addition to the environment data that all apps can access. These low-level streams are for research purposes only and are not available to apps that are in the Windows Store.

Learn more about HoloLens Research Mode and get sample applications here.

Warnings

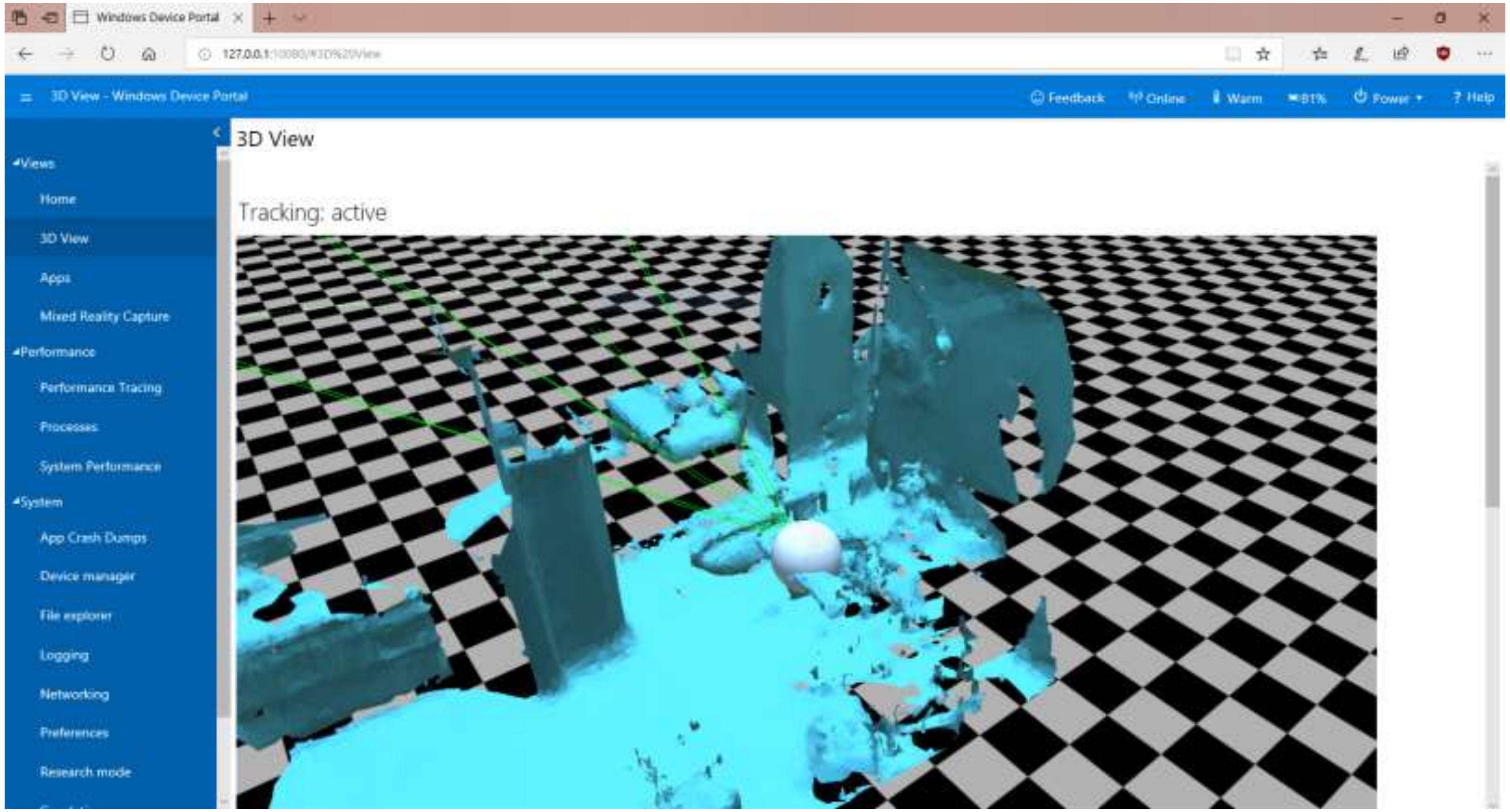
A malicious application may misuse the sensor data or potentially compromise the integrity of the system. For this reason research mode is recommended only for devices that are being actively used for research tasks.

While enabled, research mode affects performance and battery life regardless of whether or not any applications are running.

Changes to this setting require a reboot to take effect.

Device Portal for HoloLens

- Live 3D viewer of head tracking, mesh, etc.



3D View

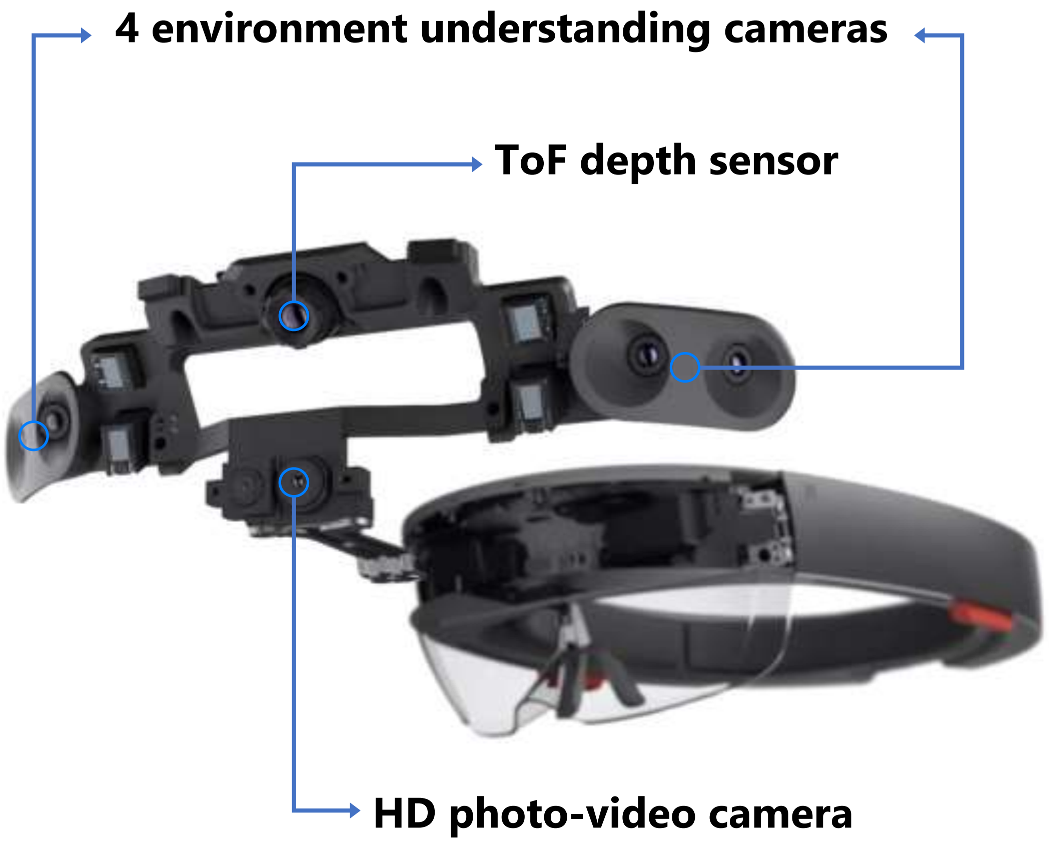
Tracking: active

Device Portal for HoloLens

- Provides REST API for management (More Info)
  - Access files, download meshes, etc.
  - For example, check battery status: http://127.0.0.1:10080/api/power/battery

Research Mode Sensor Streams

- Grayscale HeT cameras (4 streams)
- Color photo-video camera (1 streams)
- ToF cameras (4 streams)
  - Short-throw / long-throw
  - Depth / reflectivity



Research Mode API

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

Research Mode API

- No low-level access needed (but possible)

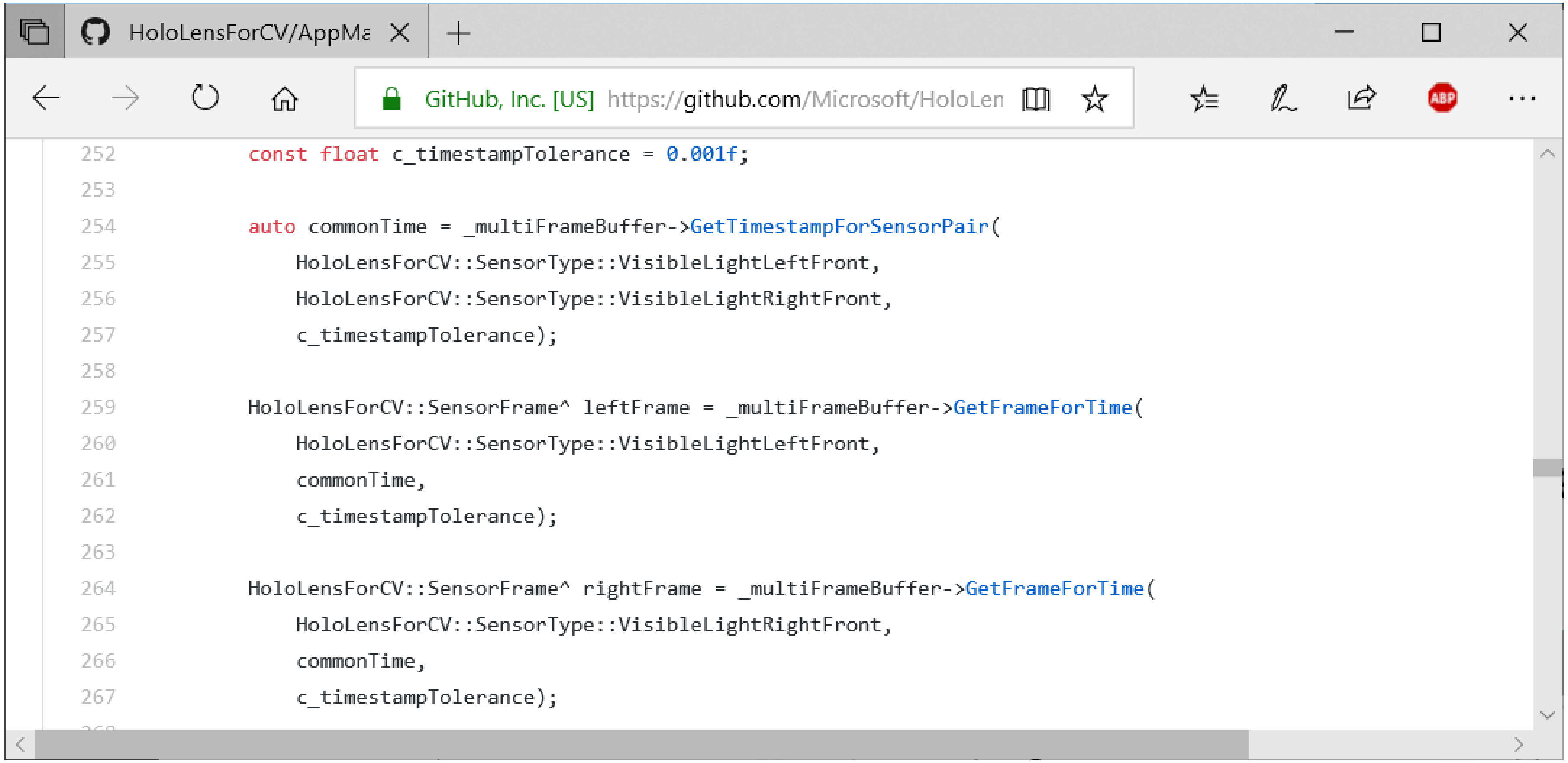
- We made it easy for you: ***HoloLensForCV***
  https://github.com/Microsoft/HoloLensForCV

  - Higher-level abstraction of Research Mode API
    (see Shared/HoloLensForCV/ folder)

  - Sample Desktop/HoloLens applications and tools
    (see Samples/ and Tools/ folders)

HoloLensForCV Features

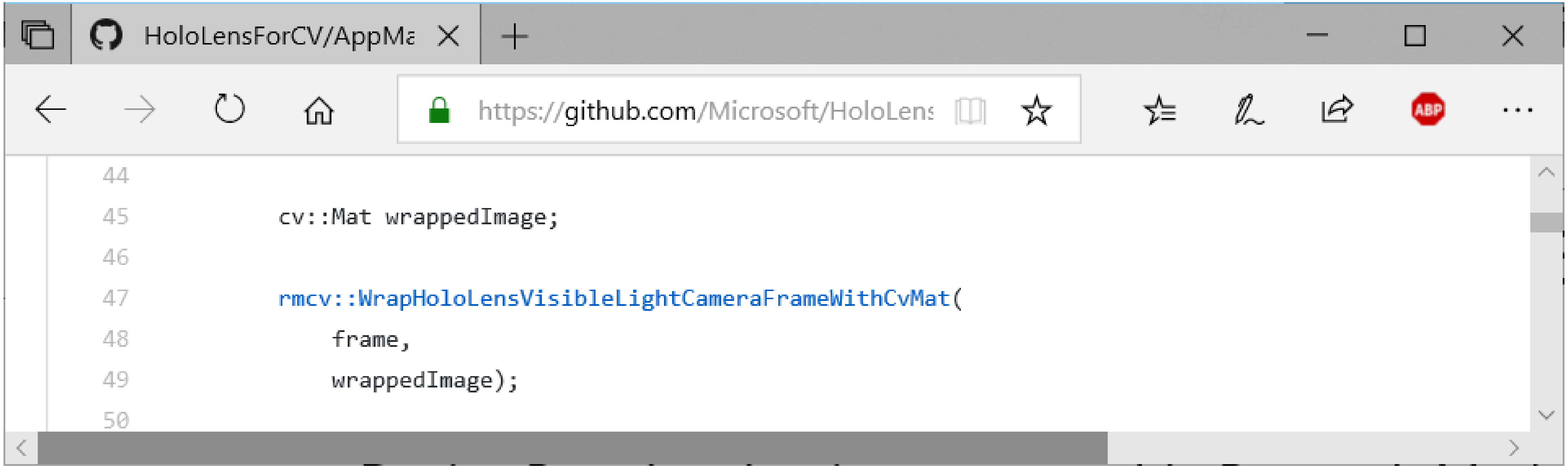
- Frame synchronization



```
const float c_timestampTolerance = 0.001f;

auto commonTime = _multiFrameBuffer->GetTimestampForSensorPair(
    HoloLensForCV::SensorType::VisibleLightLeftFront,
    HoloLensForCV::SensorType::VisibleLightRightFront,
    c_timestampTolerance);

HoloLensForCV::SensorFrame^ leftFrame = _multiFrameBuffer->GetFrameForTime(
    HoloLensForCV::SensorType::VisibleLightLeftFront,
    commonTime,
    c_timestampTolerance);

HoloLensForCV::SensorFrame^ rightFrame = _multiFrameBuffer->GetFrameForTime(
    HoloLensForCV::SensorType::VisibleLightRightFront,
    commonTime,
    c_timestampTolerance);
```

HoloLensForCV Features

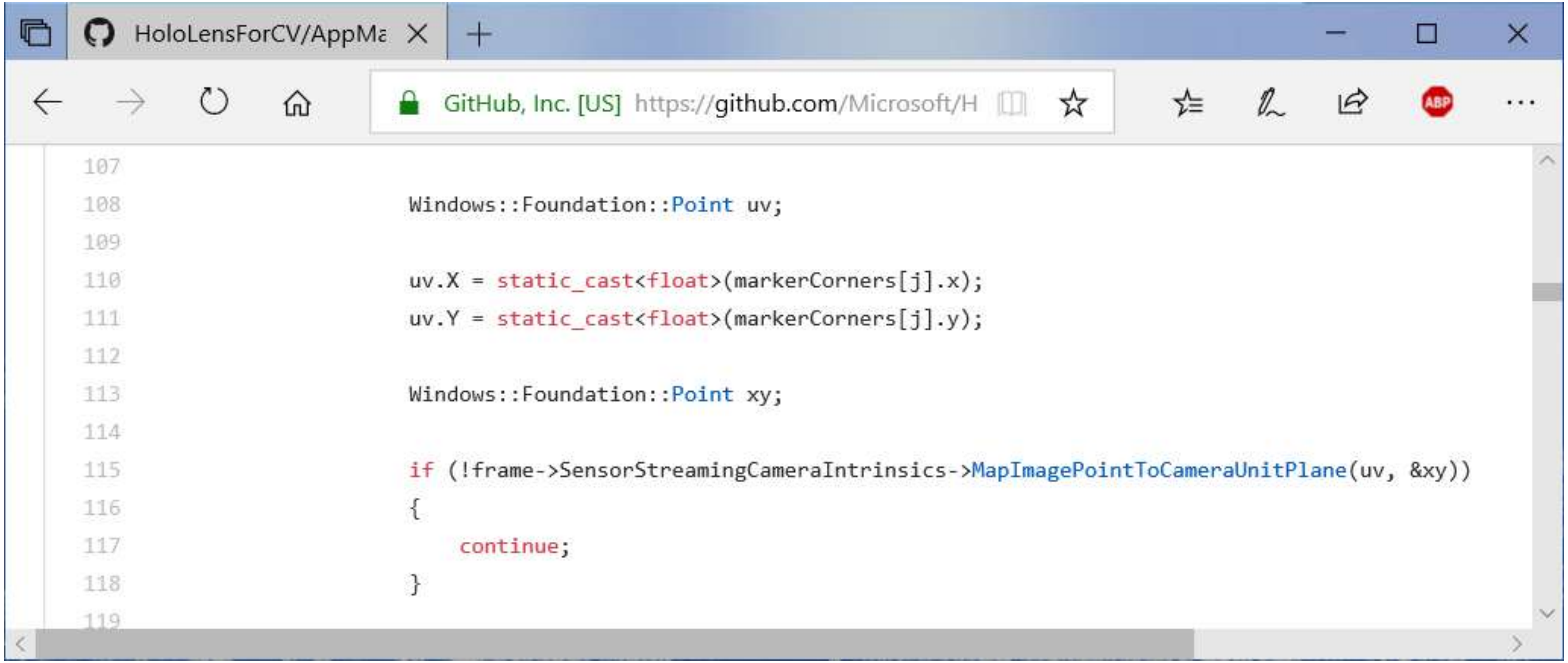
- Easily interface with OpenCV



```

        cv::Mat wrappedImage;

        rmcv::WrapHoloLensVisibleLightCameraFrameWithCvMat(
            frame,
            wrappedImage);

```

HoloLensForCV Features

- Undistortion using camera intrinsics



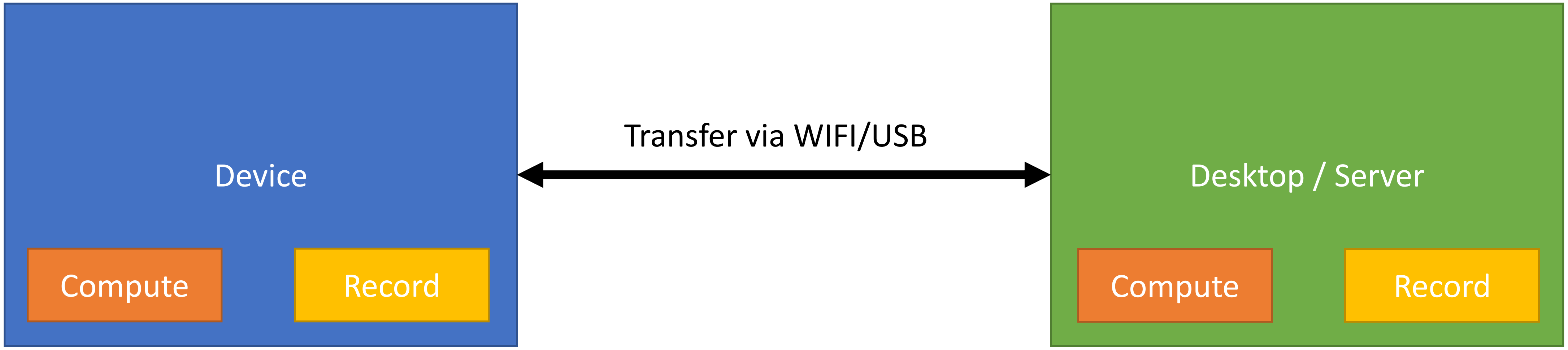
```
Windows::Foundation::Point uv;

uv.X = static_cast<float>(markerCorners[j].x);
uv.Y = static_cast<float>(markerCorners[j].y);

Windows::Foundation::Point xy;

if (!frame->SensorStreamingCameraIntrinsics->MapImagePointToCameraUnitPlane(uv, &xy))
{
    continue;
}
```

HoloLensForCV Demo

- How to get started
  - Open HoloLensForCV solution in Visual Studio
  - Select «Release/x86» configuration
  - Connect and pair HoloLens via WIFI/USB (More Info)
  - Deploy device applications 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)

- Tips for debugging on device
  - Select «Release» mode («Debug» mode most likely be too slow on the HoloLens)
    - Use logging/tracing with `dbg::trace("your message %d", your_int)`
    - Wrap code with breakpoints in `#pragma optimize("", off/on)`

Research Mode Scenarios



Device

Compute

Record

Transfer via WIFI/USB

Desktop / Server

Compute

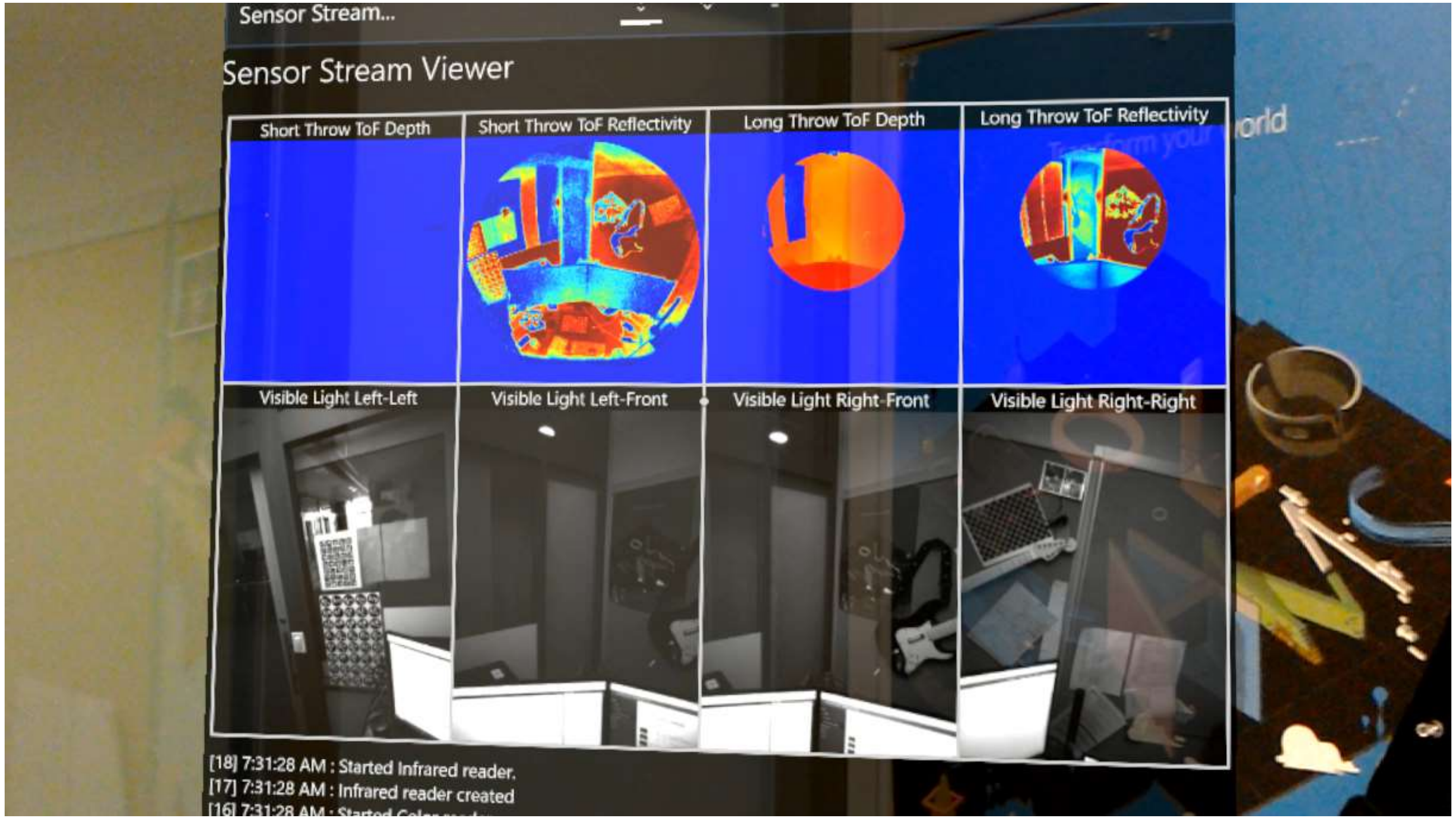
Record

Samples/ArucoMarkerTracker
Samples/ComputeOnDevice
Samples/SensorStreamViewer
Tools/Recorder
Tools/StreamerVLC
Tools/StreamerPV

Samples/ComputeOnDesktop
Samples/py/sensor_receiver
Samples/py/recorder_console
Tools/ReceiverVLC
Tools/ReceiverPV

Research Mode Samples

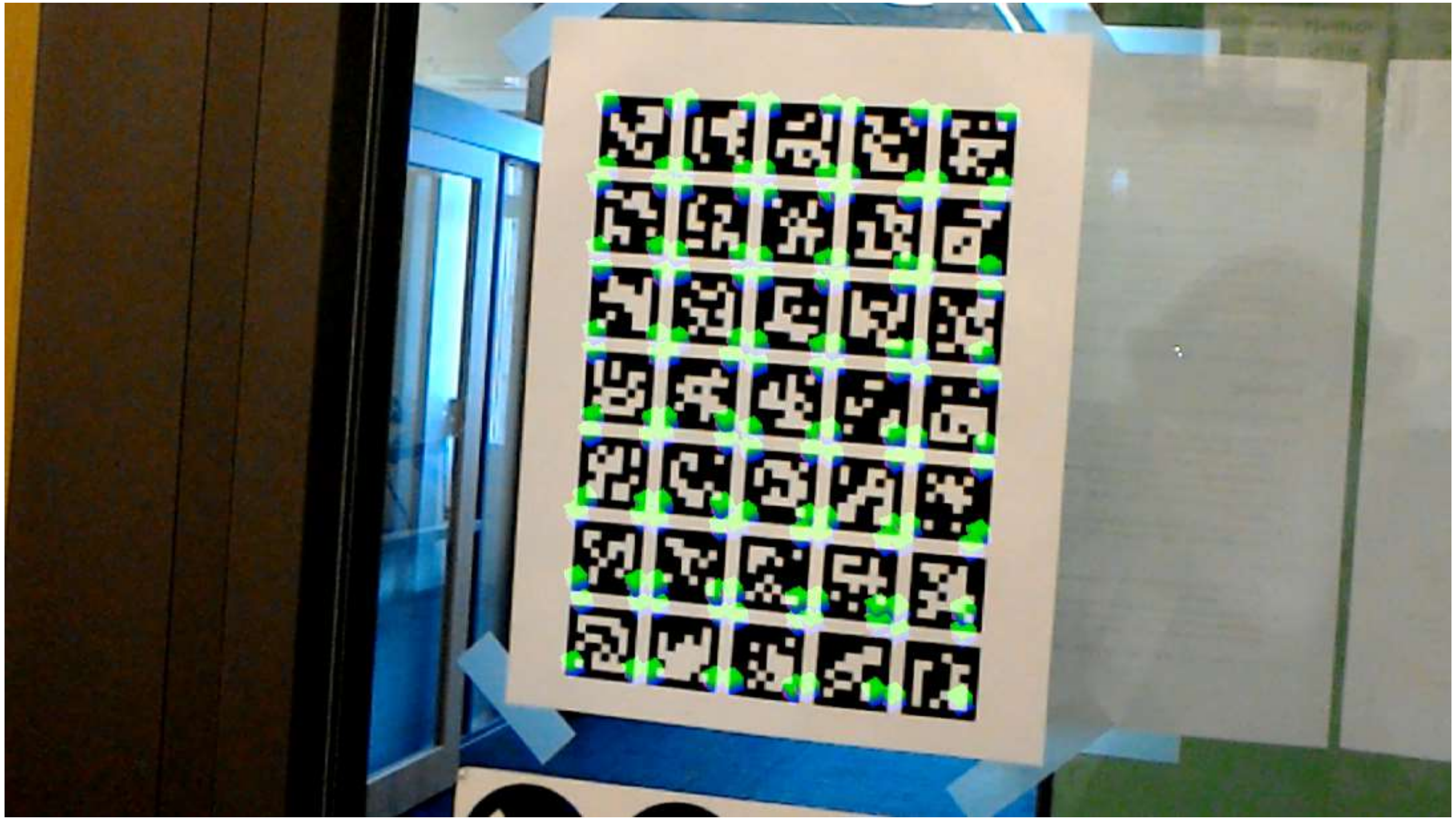
- Sensor stream viewer



```
[18] 7:31:28 AM : Started Infrared reader.
[17] 7:31:28 AM : Infrared reader created
[16] 7:31:28 AM : Started Color reader
```

Research Mode Samples

- Marker tracking, triangulation, and rendering

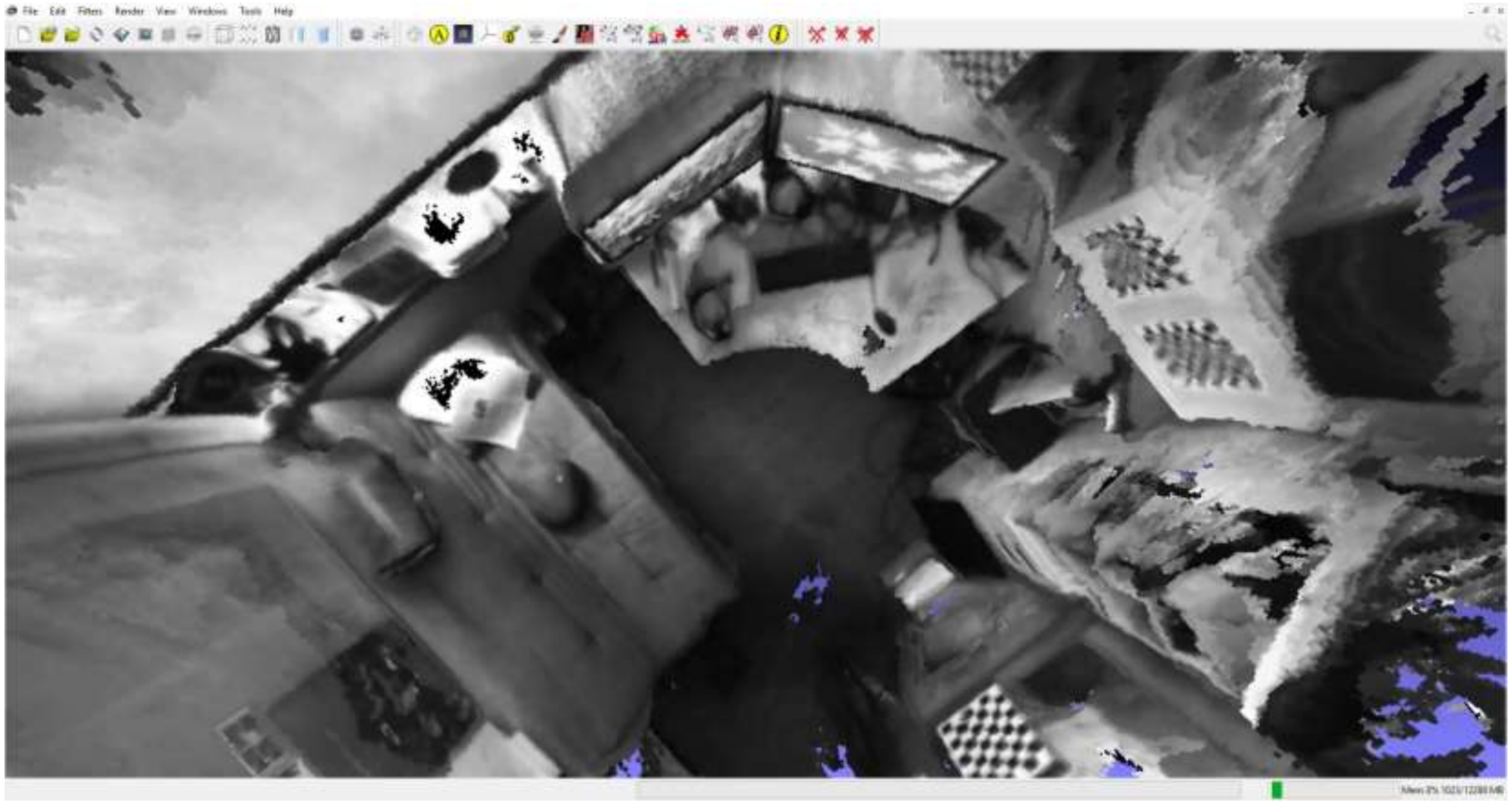


Research Mode Samples

- Wide field of view person tracking (MSR Cambridge)

Research Mode Samples

- Dense multi-view stereo from tracking cameras (MSR Redmond)



Research Mode Samples

- Streamer/Receiver applications
  - Stream sensor data from HoloLens to Desktop over WIFI/USB
    (Make sure to be on the same local network, find the IP address of the HoloLens in the device portal or on the HoloLens under Settings/Network/WIFI/Advanced Settings)
  - Process/visualize sensor streams on computer
    (Specify the IP address of the HoloLens as described above, specify the right ports for the stream as defined in the streamer application code)

Research Mode Samples

- Recorder application

  - Store sensor streams on HoloLens for later offline processing

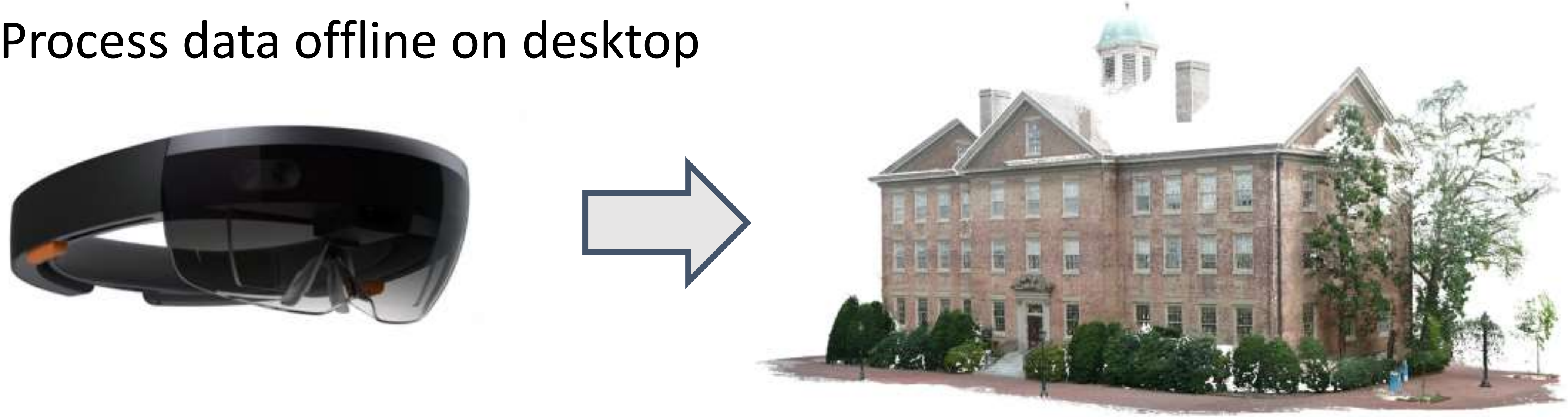
  - Download the data using the device portal file browser

    or

    Download/Process the data using the recorder_console.py script
    (see following slides for details)

Using HeT Cameras for 3D Reconstruction

- Record sensor streams using HoloLensForCV tool
- Process data offline on desktop



- COLMAP: End-to-end image-based 3D reconstruction pipeline

Open Source (pre-built binaries available) - https://github.com/colmap/colmap

Instructions

Record dataset on HoloLens

1. Open HoloLensForCV: Recorder application
2. Tap/click to start recording sensor streams
3. Walk around and capture scene
4. Tap/click to end recording

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-C432-4875-8C57-...
Searching for recordings...
=> Found a total of 6 recordings

Available commands:
  help:                       Print this help message
  exit:                       Exit the 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

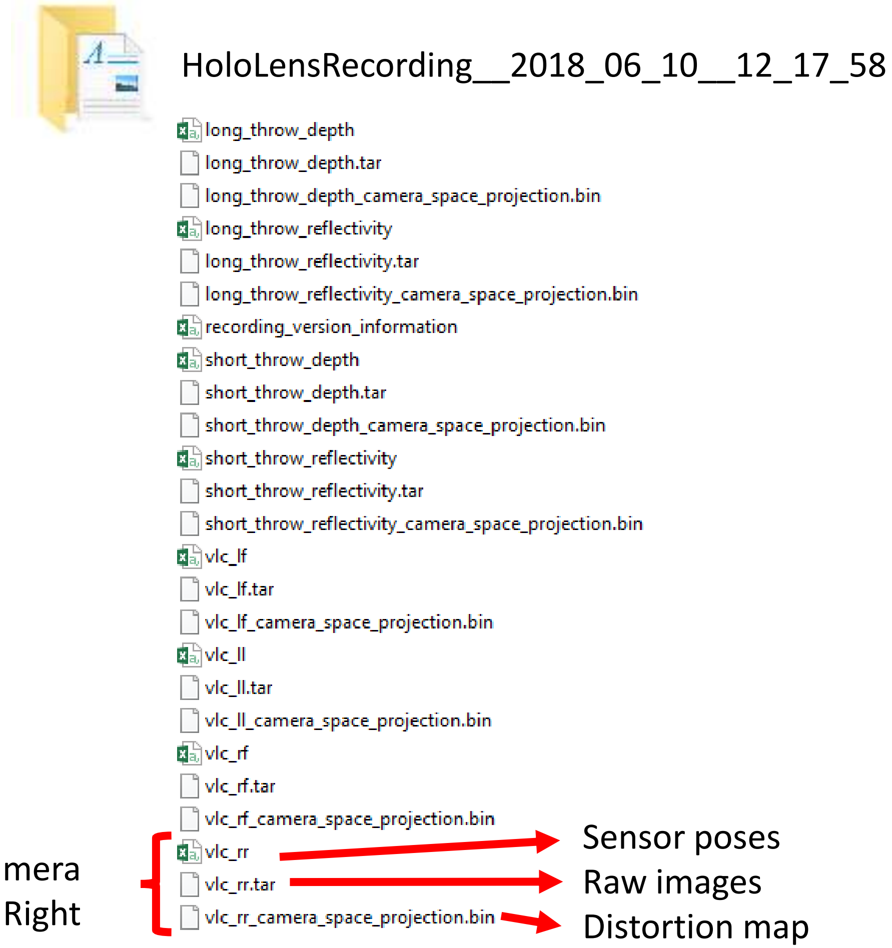
Download and reconstruct on desktop

2. List recordings

```
>>> list

Device recordings:
[      0]   HoloLensRecording__2018_06_10__12_17_58
[      1]   HoloLensRecording__2018_06_10__12_36_17
[      2]   HoloLensRecording__2018_06_10__12_39_00
[      3]   HoloLensRecording__2018_06_10__12_41_25
[      4]   HoloLensRecording__2018_06_10__12_44_11
[      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
```



HoloLensRecording__2018_06_10__12_17_58

long_throw_depth
long_throw_depth.tar
long_throw_depth_camera_space_projection.bin
long_throw_reflectivity
long_throw_reflectivity.tar
long_throw_reflectivity_camera_space_projection.bin
recording_version_information
short_throw_depth
short_throw_depth.tar
short_throw_depth_camera_space_projection.bin
short_throw_reflectivity
short_throw_reflectivity.tar
short_throw_reflectivity_camera_space_projection.bin
vlc_lf
vlc_lf.tar
vlc_lf_camera_space_projection.bin
vlc_ll
vlc_ll.tar
vlc_ll_camera_space_projection.bin
vlc_rf
vlc_rf.tar
vlc_rf_camera_space_projection.bin
vlc_rr
vlc_rr.tar
vlc_rr_camera_space_projection.bin

Visible Light Camera
Right Right

Sensor poses
Raw images
Distortion map

Instructions

Download and reconstruct on desktop

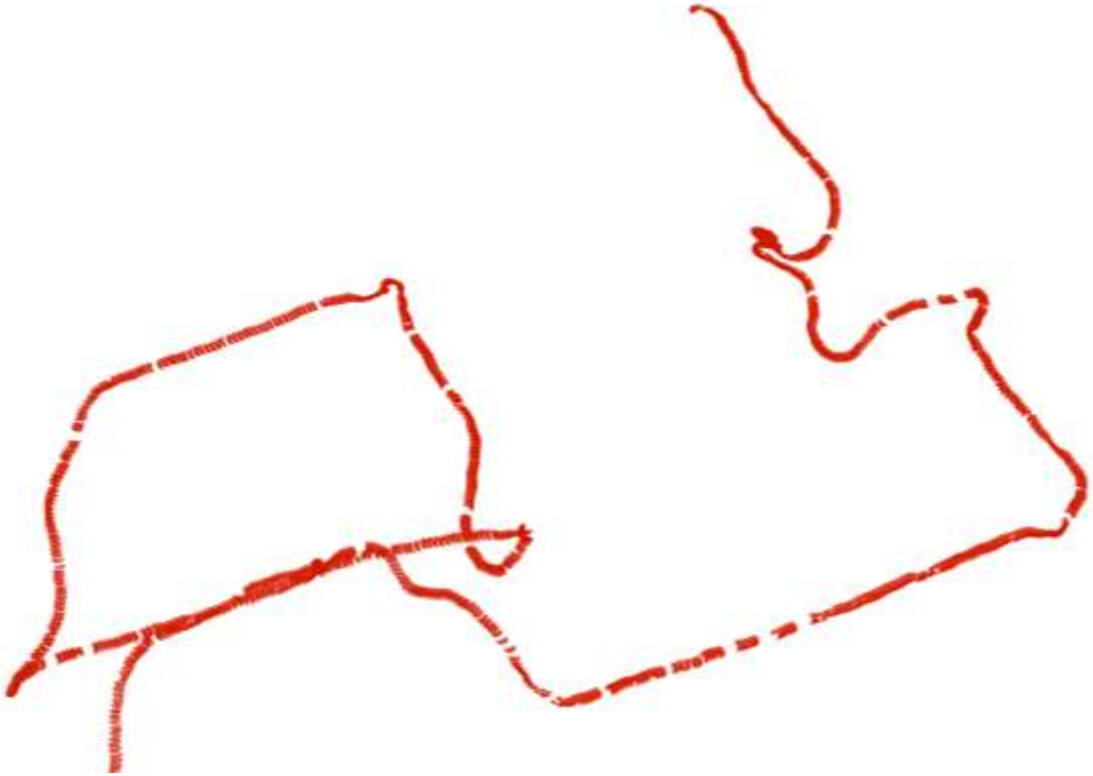
3. (Optional) Reconstruct recording

```
>>> reconstruct 0

=> Sparse Reconstruction ...
...
=> Dense Reconstruction ...
...
```

- Reconstruction code as an example for:
  - Sensor frame synchronization / resampling
  - Absolute camera pose extraction
  - Automatic camera calibration and undistortion

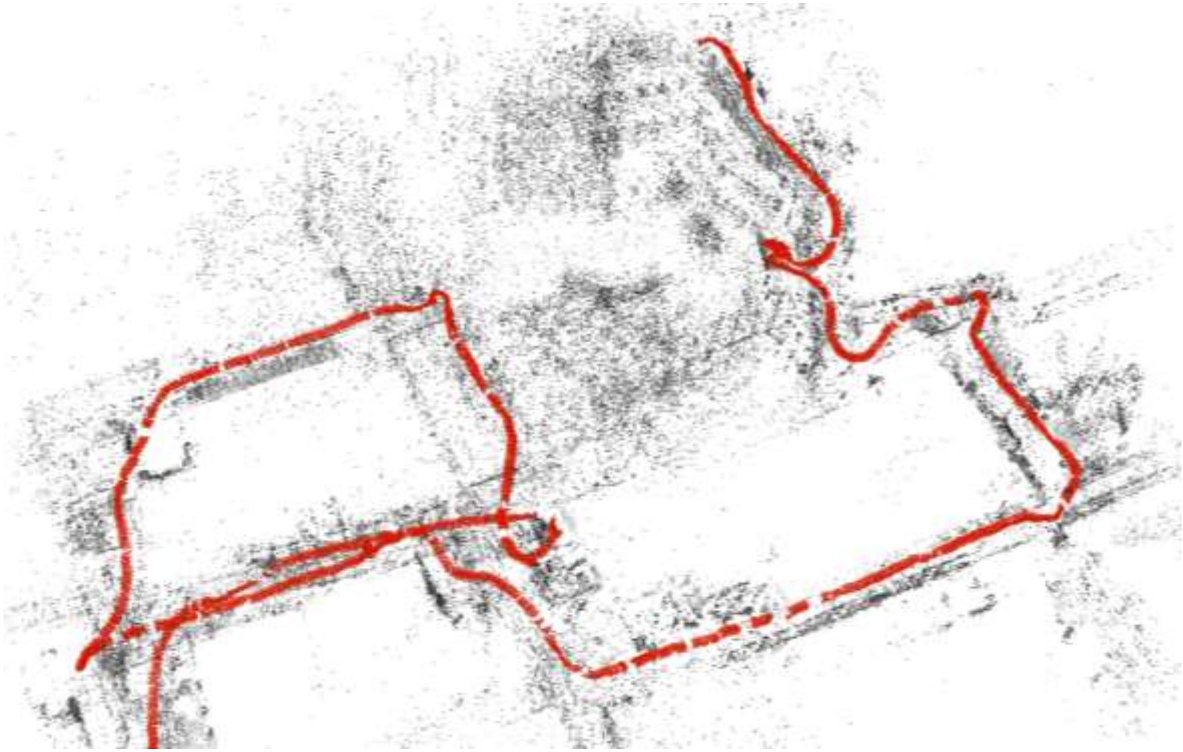
COLMAP Reconstruction – MSR Building 99

- Initial HoloLens camera poses

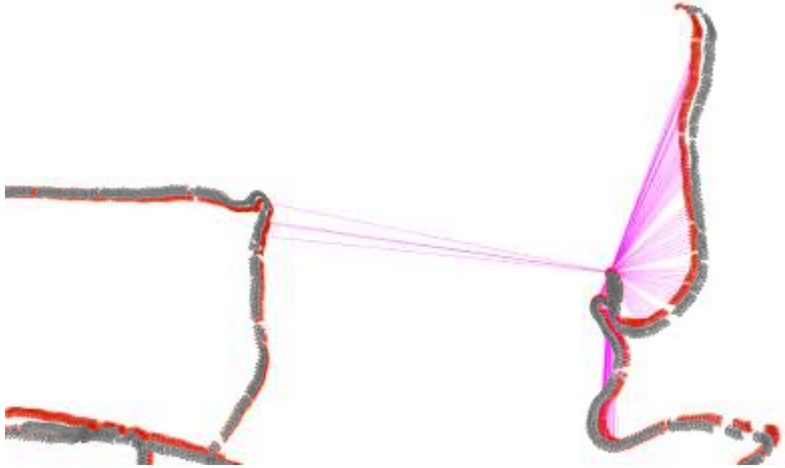


COLMAP Reconstruction – MSR Building 99

- Sparse reconstruction



Long-term loop closure constraints



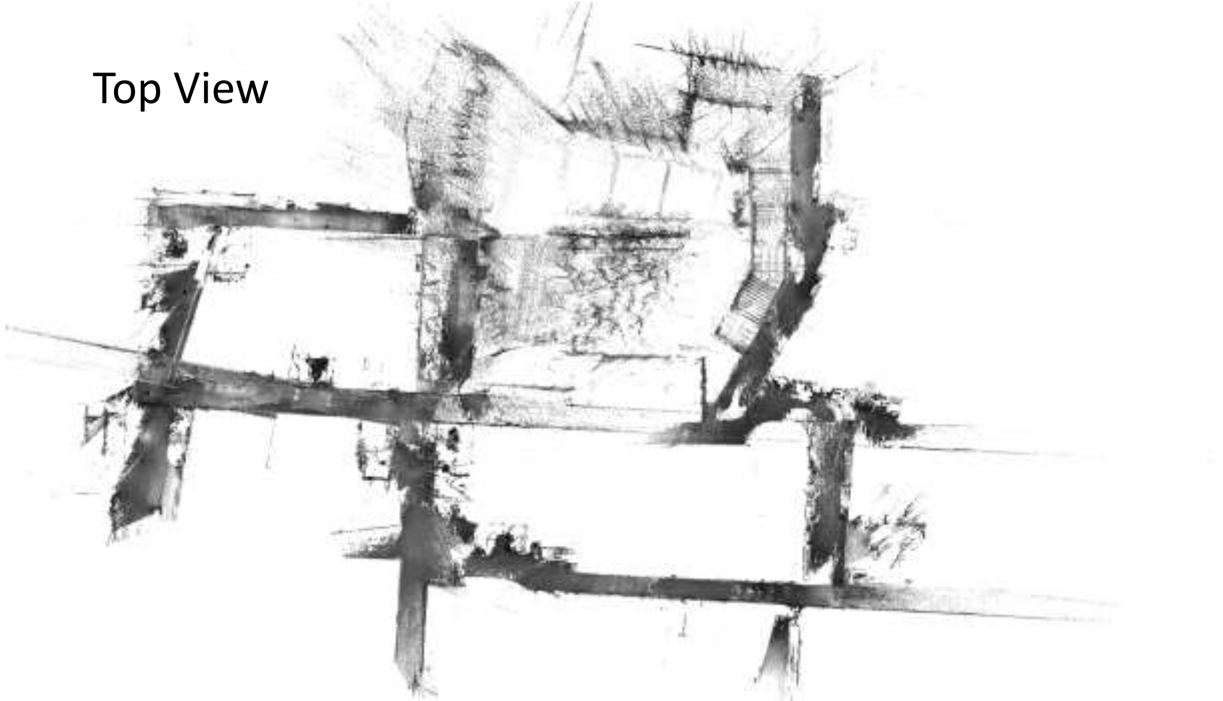
COLMAP Reconstruction – MSR Building 99

- Dense reconstruction
  (from HeT cameras only)

COLMAP Reconstruction – MSR Building 99

- Dense reconstruction (from HeT cameras only)

Top View



Side View