Mixed Reality Independent Study

Nikita Shokhov, School of Visual Arts

D. Pillis, ICAT

Aisling Kelliher, ICAT, Computer Science

Spring 2020

Accomplishments of Independent Study:

- Setting up a studio for volumetric video capture: depth camera, background, lighting, computer with software

- Production and post-production in Depthkit with Intel Realsense

- Developing for Oculus Quest in Unity

- Have been introduced to Hololens development in Unity

- Have been introduced to photogrammetry workflow with DSLR camera and Photoscan Pro, and tablet-based Structure depth-camera

- Created a VR experience based on volumetric video assets captured by Daniel:
    - Installing necessary packages into Unity project: Depthkit, Oculus
    - Connecting Oculus Rift with Unity project
    - Managing depthkit assets
    - Filtering characters with *Zero Days Depthkit* plugin
    - Filtering camera with *Post-Processing* package
    - 360 video capture in Unity project with *Unity Recorder* plugin

Hardware & Software:

Scatter’s Depthkit

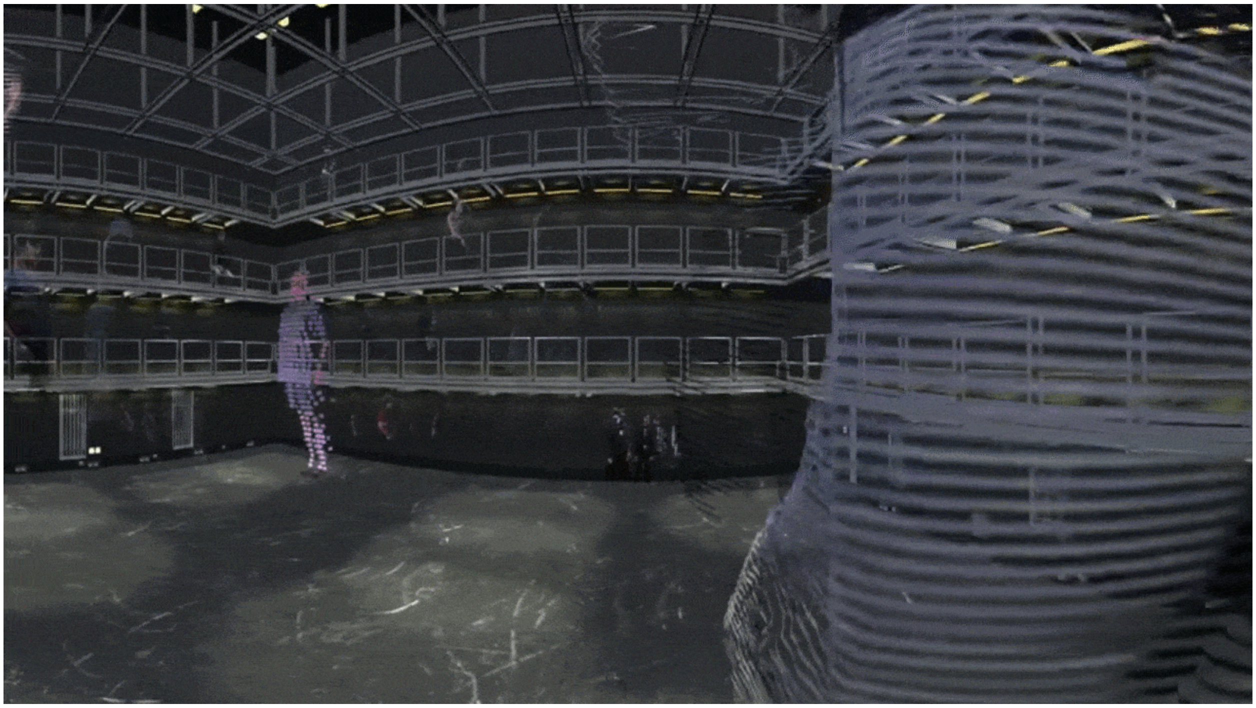
Unity

Hololens Development

Oculus Quest

Project:

Over 45 volumetric capture sequences have been assembled documenting two years of collaboration at Virginia Tech. Each volumetric clip is looped and modified with shaders to integrate the viewer into a holographic experience of the past two years.



This is 360 video documentation of the project:

Bibliography:

Light3DPose: Real-time Multi-Person 3D PoseEstimation from Multiple Views

https://arxiv.org/pdf/2004.02688.pdf

An approach to perform 3D pose estimation of multiple people from a few calibrated camera views. Our architecture, leveraging the recently proposed unprojection layer, aggregates feature-maps from a 2D pose estimator backbone into a comprehensive representation of the 3D scene. Such intermediate representation is then elaborated by a fully-convolutional volumetric network and a decoding stage to extract 3D skeletons with sub-voxel accuracy. Our method achieves state of the art MPJPE on the CMU Panoptic dataset using a few unseen views and obtains competitive results even with a single input view. We also assess the transfer learning capabilities of the model by testing it against the publicly available Shelf dataset obtaining good performance metrics.

Cloud Rendering-based Volumetric Video Streaming System for Mixed Reality Services

https://arxiv.org/pdf/2003.02526.pdf

Volumetric video is an emerging technology for immersive representation of 3D spaces that captures objects from all directions using multiple cameras and creates a dynamic 3D model of the scene. However, rendering volumetric content requires high amounts of processing power and is still a very demanding tasks for today's mobile devices. To mitigate this, we propose a volumetric video streaming system that offloads the rendering to a powerful cloud/edge server and only sends the rendered 2D view to the client instead of the full volumetric content. We use 6DoF head movement prediction techniques, WebRTC protocol and hardware video encoding to ensure low-latency in different parts of the processing chain. We demonstrate our system using both a browser-based client and a Microsoft Hololens client. Our application contains generic interfaces that allow easy deployment of different augmented/mixed reality clients using the same server implementation.

State of the Art on Neural Rendering

https://arxiv.org/pdf/2004.03805.pdf

Progress in computer vision and machine learning have given rise to a new approach to image synthesis and editing, namely deep generative models. Neural rendering is a new and rapidly emerging field that combines generative machine learning techniques with physical knowledge from computer graphics, e.g., by the integration of differentiable rendering into network training. With a plethora of applications in computer graphics and vision, neural rendering is poised to become a new area in the graphics community. This state-of-the-art report summarizes the recent trends and applications of neural rendering. Starting with an overview of the underlying computer graphics and machine learning concepts, we discuss critical aspects of neural rendering approaches. Specifically, our emphasis is on the type of control, i.e., how the control is provided, which parts of the pipeline are learned, explicit vs. implicit control, generalization, and stochastic vs. deterministic synthesis. The second half of this state-of-the-art report is focused on the many important use cases for the described algorithms such as novel view synthesis, semantic photo manipulation, facial and body reenactment, relighting, free-viewpoint video, and the creation of photo-realistic avatars for virtual and augmented reality telepresence.

Through the Looking Glass: Neural 3D Reconstruction of Transparent Shapes

https://arxiv.org/pdf/2004.10904.pdf

Recovering the 3D shape of transparent objects using a small number of unconstrained natural images is an ill-posed problem. Complex light paths induced by refraction and reflection have prevented both traditional and deep multiview stereo from solving this challenge. We propose a physically-based network to recover 3D shape of transparent objects using a few images acquired with a mobile phone camera, under a known but arbitrary environment map. Our novel contributions include a normal representation that enables the network to model complex light transport through local computation, a rendering layer that models refractions and reflections, a cost volume specifically designed for normal refinement of transparent shapes and a feature mapping based on predicted normals for 3D point cloud reconstruction. We render a synthetic dataset to encourage the model to learn refractive light transport across different views. Our experiments show successful recovery of high-quality 3D geometry for complex transparent shapes using as few as 5-12 natural images.

BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration

https://arxiv.org/pdf/1604.01093.pdf

Real-time, high-quality, 3D scanning of large-scale scenes is key to mixed reality and robotic applications. Recent online methods demonstrate compelling results, but suffer from: (1) needing minutes to perform online correction, preventing true real-time use; (2) bring frame-to-frame (or frame-to-model) pose estimation resulting in many tracking failures; or (3) supporting only unstructured point-based representations, which limit scan quality and applicability. We systematically address these issues with a novel, real-time, end-to-end reconstruction framework. At its core is a robust pose estimation strategy, optimizing per frame for a global set of camera poses by considering the complete history of RGB-D input with an efficient hierarchical approach. We remove the heavy reliance on temporal tracking, and continually localize to the globally optimized frames instead. We contribute a parallelizable optimization framework, which employs correspondences based on sparse features and dense geometric and photometric matching.

A low-cost, flexible and portable volumetric capturing system

https://arxiv.org/pdf/1909.01207.pdf

Multi-view capture systems are complex systems to engineer. They require technical knowledge to install and complex processes to setup. The capturing systems developed with these methods are intertwined with them, relying on custom solutions and seldom - if not at all - publicly available. We design, develop, and publicly offer a multi-view capture system based on the latest RGB-D sensor technology.

References:

Queer Skins
Ilya Szilak

Resources:

LinkedIn Learning Tutorials:

Intro to Hololens Development

Intro to WebGL and Interactive 3D Graphics

Questions & Answer:

Week 1
Android Debug Bridge Debugging

- +Loading applications onto Oculus Quest involves connecting to the device via a high powered USB C cord and a protocol called "Android Debug Bridge".
- +Try to build and load a sample empty project onto the Quest. - I'll try.
- More info about Android Debug Bridge can be found here:

https://developer.oculus.com/documentation/native/android/mobile-adb/?locale=en_US

- Two softwares that assist with ADB transfer:

https://drive.google.com/file/d/1RhXzITwjZ_WEL3IEWDfVCEXyFg_PXfgW/view?usp=drivesdk https://drive.google.com/file/d/1WrFRIxD3D8wOmDH_SizSLOSP2d2crNa6/view?usp=drivesdk

- Android debug bridge requires a number of elements, including SDK tools.
- +Make sure you follow all of these steps:

https://developer.oculus.com/documentation/native/android/mobile-device-setup/

- Insert Screenshots of your Player Settings and Project Settings Below:

Week 2

Solution Log:

1. Built an empty scene - it behaves the same - doesn't work on Quest.
2. Created a completely new app in Oculus developing - now for Go (previous one was for Rift) - still doesn't work on Quest.
3. Tried using another adb - VRsideloader - it uploads very well and easy - but the app still doesn't launch
4. ADB link is of the latest version - it doesn't help
5. Ok, I will double check all of the steps. You are sure you correctly created a com.company.title and ID, right?
6. Double check these steps
   a. https://circuitstream.com/blog/oculus-quest-unity-setup/
7. The resolve in this case was a checkbox to allow ARM processor architecture in settings: Player/Other/Configuration/Target Architectures - ARMv7 .
   Link where I found this:
   https://androidlad.blogspot.com/2016/10/installfailednomatchingabis-when.html
8. Generate OSIG number for a particular device.

Now it plays the Volumetric Crowd scene. But the quest processor is so slow that it produces only little movements of the characters sequentially. So all the characters freeze beside one of them at a time which moves a few frames, and then another character does the same, and so on in sequential order.

Week 3

Questions:

- Is there a way to simplify these depth capture assets to make them affordable for Quest?
- We should definitely try to build this for Rift and check how fast it plays via a tethered headset.

Week 4

Questions 05/02/2020:

- Have you ever dealt with WebGL?

Check this out on LinkedIn Learning:
https://www.linkedin.com/learning/learning-3d-graphics-on-the-web-with-three-js?trk=share_android_course_learning

> For example, if I want to make a VR experience that would be available online to anyone either on a screen or HMD, what would be the best platform for this? This is just in case if you already know the best workflow for this. Otherwise, I will have to do research when I have time.

Look into Sandy Ressler WebVr

- Have you ever dealt with syncing multiple VR headsets for interactive experiences? I did work with syncing 12 Go headsets for 360 video, but never for .apk files for Quests.

This requires external content management software that tends to be expensive, in order to run and manage multiple Android based headsets remotely.

There are options internally to lock an HMD to certain functionality which is useful for gallery installations

It depends on the scale and need for the exhibit, and so each use case has specific solutions, none of which are easy