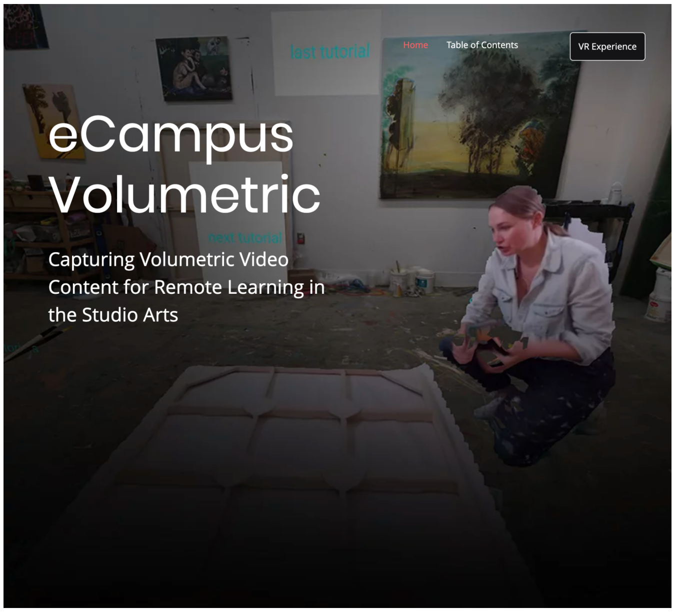
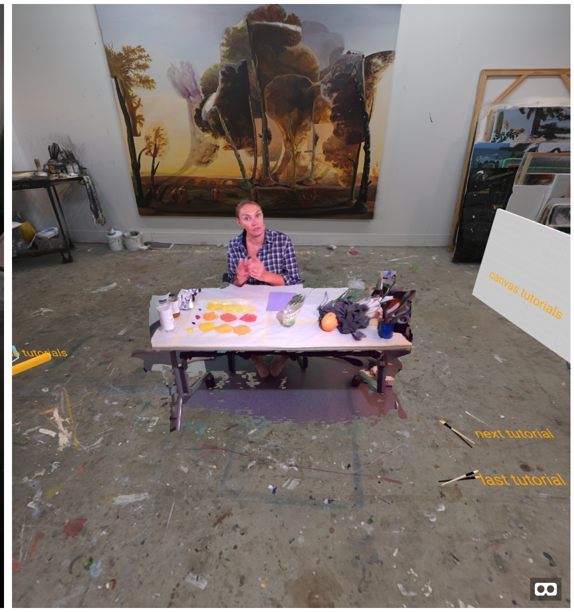
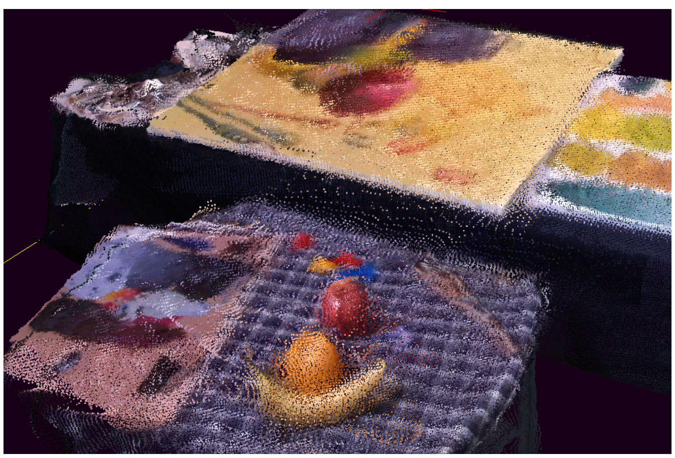
Capturing Volumetric Video Content for Remote Learning in the Studio Arts







Ontario Tech University

Dr. Andrew Hogue (andrew.hogue@ontariotechu.ca)
Dr. Alvaro Quevedo
Jakob Anderson, MSc Candidate
Colin Orian, MSc Candidate

OCADU

Dr. Cindy Poremba
Veronika Szkudlarek, M.FA
Yutong (Jessie) Zheng, Masters

York University

Nick Fox-Gieg, PhD Candidate

Project Objectives

This project aims to develop volumetric content to be used by Ontario institutions in their fine arts programs to enhance remote learning through an embodied practice paradigm. Specifically, we are focusing on recording elements of studio arts practice in a number of fine arts programs (with a primary focus on material arts such as painting and sculpture) to create teaching content that will enable learners to better study particular artistic creation techniques. This content serves to address a commonly articulated gap that has emerged for studio-based instructors who have had to shift their instruction into online and/or hybrid contexts, which is communicating spatial, material and performative aspects of their instruction practice, to primarily visual and/or haptic learners.

This document serves as a documentation of the approach providing instructors and learners interested in volumetric capture techniques best practices, software, and suggestions for capturing volumetric video. We will also provide links to external websites that contain volumetric recordings and software to create and process their own recordings.

Goals:

Our goals were:

- develop an open and documented workflow to capture both high- and low- fidelity volumetric content allowing others to learn the process and tools necessary to create such content,
- record volumetric content for demonstration of material and performative arts techniques in painting and other material (and hybrid digital/material) studio-arts
- capture photogrammetry of the resulting artefacts and incorporate these into the volumetric scenes where appropriate
- create a simple framework to present the recordings on web & mobile devices with a particular focus on accessible augmented reality deployment.

Embodied Practice

The role of embodied practice is increasingly recognized in fine arts and design pedagogy (O'Connor 2005, Budge 2015), much of it building on foundational work by Polyani (tacit knowledge), Schön (knowing-in-action), O'Connor (embodied knowledge), among others. Fine arts students tend to be visual-spatial (Gardner 1983) and/or tactile/kinaesthetic (Fleming 2011) learners, requiring sensory modalities to effectively learn (Brown 2002). Many of our material arts practices are traditionally taught "hands on," through both instructor modeling, and student practice. By modeling practice, artist-instructors teach both "how to do" and "how to know" (Budge 2016)-- in other words, not just the steps to perform, but what to be attuned to in the act of making (often "tacit" knowledge experienced through being co-present in the studio, such as the texture of materials, the feel of a gesture, how the artist moves in space, etc.). Volumetric video affords capturing tacit, embodied knowledge within its spatial context, something artist-instructors have identified as being insufficiently communicated through video alone. Perceptually immersive technologies (like VR and AR) establish a different relationship between observer and environment than traditional media, going from the “representation to the presentation of perceptual spaces” (Woodcock 2013, emphasis in original). Mel Slater (2003) has observed such technologies take on their immersive quality through a faithfulness to equivalent real-world sensory modalities-- creating the signature feeling of “being there.” Maintaining more of these perceptual cues can be integral to teaching particular fine arts concepts that engage embodied material and/or performative practices. Volumetric video resources will allow artist-instructors to better address the gap between online delivery and the rich, perceptual information present in live studio teaching.

Volumetric Video

Volumetric video is a computational fusion of digital video recording and depth sensor data, resulting in a spatialized, and potentially navigable, 3D captured moving image, similar to a hologram. Volumetric images are becoming increasingly prevalent and sophisticated, driven by interest in augmented reality (AR) and virtual reality (VR) production. Volumetric video content, appearing snatched from the world and evoking feelings of co-presence, embodiment and spatiality, has affordances beyond traditional video content in representing material and performative artistic practices, allowing for better representation of things like gesture, texture, scale, etc. There are two main technologies involved in recording volumetric video content: (1) Low-fidelity solutions such as RGBD cameras (i.e. Microsoft Kinect) which provide colour AND depth from the camera per pixel allowing software to reconstruct a surface model (or point cloud) that contains geometric information of the environment as well as colour information (2) High-fidelity solutions such as Photogrammetry which computes 3D models from hundreds of images of the same scene resulting in a dense point cloud and triangular mesh that is very representative of the original artifact. Photogrammetry is utilized by land surveyors to analyze land structures, eHeritage applications to create digital replicas of historical locations, and video games/simulations to incorporate a high degree of realism into their scenes. Volumetric video encompasses the use of both of these technologies to varying degrees. Low-fidelity solutions such as RGBD cameras provide lower resolution depth information at video framerates, thus each frame of video has both colour and depth information. By fusing multiple synchronized RGBD cameras together through a calibration process, entire volumes of space can be captured simultaneously resulting in every frame of “video” having a representative 3D model of the subject/space being captured. This technique allows for entire performances/demonstrations to be captured and subsequently viewed from any perspective. Coupling this Low-fidelity solution with Higher fidelity photogrammetry for the resulting artefacts or of static objects in the scene, we can create 3D instructional videos that can be viewed interactively in virtual and augmented reality or through a modern web browser.

Traditionally, Volumetric Video is captured using a dedicated studio setup such as those from 4DViews, 8i or Microsoft Reality Studios which require a significant investment of a large dedicated space and hundreds of cameras placed in the space to capture the performance accurately. While these studios are effective at capturing high quality volumetric assets, they are prohibitive in their costs and it is very challenging for indie filmmakers or game developers or artists to experiment with this form of media. As such, we aim to provide discussion on tools that are more accessible to enable artists of all forms to enter into this space.

Figure 1: A traditional volumetric video setup. Image: [Bringing realism to VR is complex, but these developers found a way in holograms]

Figure 2: Image: [4DViews | Volumetric video: Basketball]

Speaking Volumes Symposium

SPEAKING VOLUMES

As part of a grander agenda surrounding Volumetric Video and the Arts, Dr. Cindy Poremba at OCADU organized a 1-day volumetric video symposium event on (Feb 11, 2022) bringing together volumetric artists and creators in a full-day of presentations discussing the current state and future of volumetric video. Dr. Andrew Hogue presented the VolNodes open source toolset made possible by this eCampus VLS project.

- Speaking Volumes: A 1-day symposium on volumetric images | OCAD University
- Speaking Volumes - EventBrite Link

As part of this project, we make it available online at the link below for those interested in volumetric video, this is a great way to start your journey and be inspired:
- Speaking Volumes: Youtube - Full Recording

Volumetric Discord Community

This project has allowed us to develop and strengthen an online community of filmmakers, software developers, game developers, artists, students, and academics that is open to all interested in joining.

Invite Link to Volumetric Discord Community (Open to all):
- https://discord.gg/jMptJNZVmG

The Open Source Volumetric Video Pipeline

Capturing and processing volumetric video (VV) is a challenging task in general. As VV is a relatively new paradigm, there are a number of things that need to be considered when embarking on the process of creating VV. In this section, I will explain the techniques we explored and the advantages/disadvantages of each path. Finally, I will provide links to our final solution and some examples of the workflow. The descriptions of each workflow is important to those wishing to embark into the volumetric video space as you can learn about tools and techniques that you may not have thought of before.

These workflows are intended to provide you a starting point for your journey into the volumetric space. Also, we focused specifically on the use of single and multiple Azure Kinect cameras (RGB-D) as this is the most prevalent camera used in the consumer volumetric space currently.

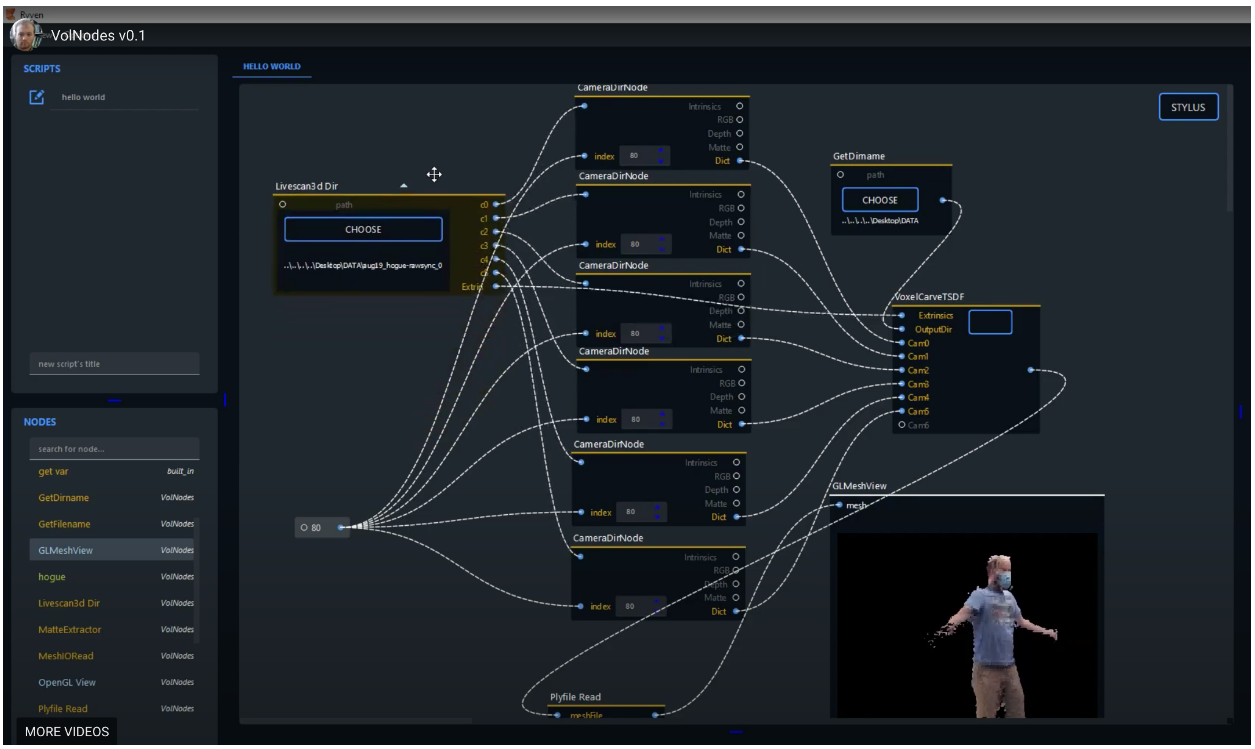
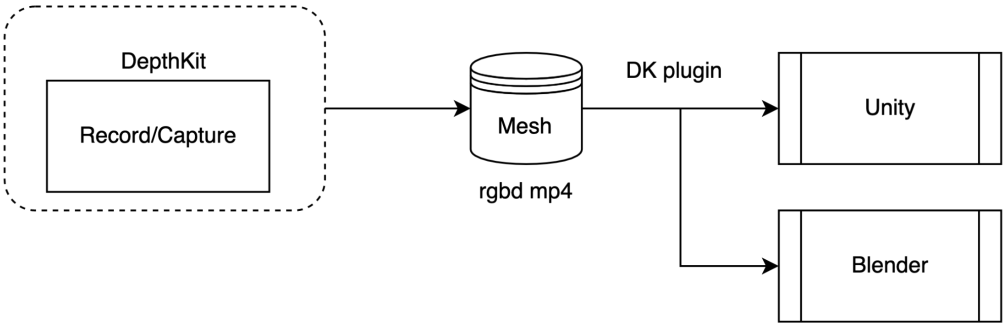


Figure 3: Image of VolNodes: an interface we developed to aid experimentation in the volumetric video space

OpenSource Pipeline v0.0:



Goal:

- What is the simplest and fastest way to get started in volumetric video capture?

Hardware Used:

- Single RGB-D Azure Kinect
  - Azure Kinect developer kit – Microsoft

Software Used:

- Depthkit (single camera)
  - Depthkit
- Unity 3D
  - Unity
- Blender
  - Blender

Description

- This pipeline is the simplest and easiest to use solution to volumetric video as it requires only a single camera and the software has a free trial
- Depthkit is a standalone piece of software that enables you to capture RGB-D video from a single Azure Kinect. The output from DepthKit is a RGB-D video in MP4 format. This contains both the colour, depth, and audio information.
- Depthkit comes with a Unity plugin that allows you load the sequence into Unity and utilize the result in your game engine scene as another asset.

Advantages:

- Easy to get up and running
- Audio is supported out of the box

Disadvantages

- Monthly subscription costs after trial
- Single Camera solution only
- Data format is proprietary and mesh is created at runtime in Unity, thus it is difficult to manipulate/process the resulting mesh sequence in standard animation packages such as Blender

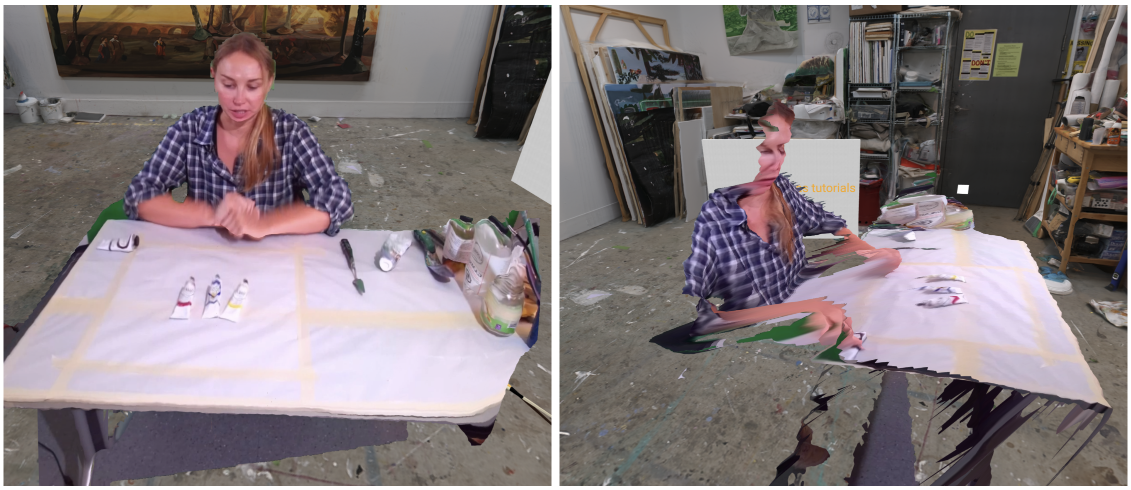
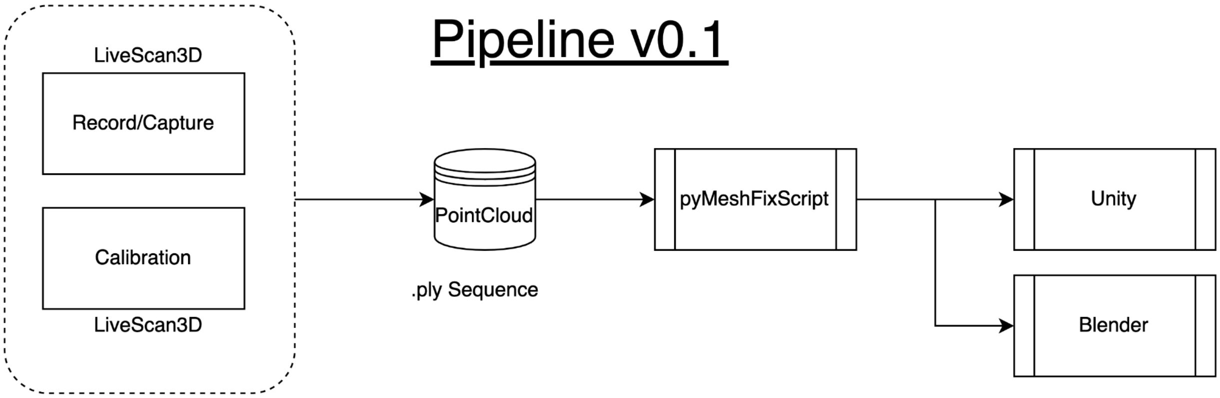


Figure 4: Images of a Single RGB-D Depthkit Solution. The image on the right demonstrates the limitations of a single camera as it cannot infer the information from a different viewpoint effectively.

OpenSource Pipeline v0.1



Goal:

- What is the simplest **Open Source** and **$Free** pipeline to get started with volumetric video

Hardware Used:

- Single or Multiple Azure Kinect cameras

Software Used:

- LiveScan3D - Azure Kinect Branch
  - https://github.com/MarekKowalski/LiveScan3D/tree/AzureKinect
- Unity
- Blender
- Stop-Motion-OBJ Addon for Blender:
  - https://github.com/neverhood311/Stop-motion-OBJ

Description:

- LiveScan3D is an open source project developed by researchers Marek Kowalski and Jacek Naruniec for the original Microsoft Kinect v2 camera platform. Since then, researcher Chris Remde (@D4E on Discord) has extended LiveScan3D to work with Azure Kinect Cameras.
- LiveScan3D works with multiple Azure Kinect cameras and is a distributed solution, i.e. can run on multiple PCs networked together on a local network.
- There are two parts to capturing VV with LS3D, Calibration and Capture
  - Calibration.
    - Calibration requires that all cameras view a calibration target. LS3D looks for special target images that you can print off on a laser printer and

attach to a surface. The cameras then find the target and determine their rotation and translational offsets thus producing a calibration transformation that allows the data from each camera to be transformed into a common world coordinate system.

- Capture.
    - Once the cameras are calibrated the raw volumetric point cloud data can be previewed on screen.
    - LS3D then allows you to record these point clouds which in turn saves the recordings to each client and then sends the results back to the main computer in a simple format.
- Player.
    - LS3D also comes with a player to view the recordings

- Output.
    - The output is a sequence of .ply files. A PLY file is the simplest mesh/point cloud as described here:
        - PLY (file format) - Wikipedia
- PLY Sequence to Blender
    - Using the Stop-Motion-OBJ Addon for Blender, the PLY file sequence can be loaded into Blender. This loads the entire sequence into Blender such that each frame is a separate entity. This allows you to place the resulting volumetric point cloud in any animation scene and render it normally.

Advantages:

- Free and Open Source
- Stop-motion-obj add-on works to load .ply/.obj sequences into Blender.

Disadvantages:

- Requires knowledge of visual studio and thus is intended primarily for those who have developed software (or has at least one developer on their team).
- No simple way to convert the .ply sequence into something that Unity requires
- Audio must be re-sync'd carefully afterwards and recorded separately.

Links to Video examples of our LiveScan3D tests:

- Volumetric Test - 2021 04 08 11 23 52
- Volumetric Test - 2021 04 08 11 41 25
- Volumetric Test - 2021 04 14 12 38 52
- Volumetric Test - 2021 04 14 12 39 39
- Volumetric Test - 2021 04 14 12 42 54
- Volumetric Test - 2021 05 12 11 35 34
- Calibration process Video: LiveScan3D Calibration process

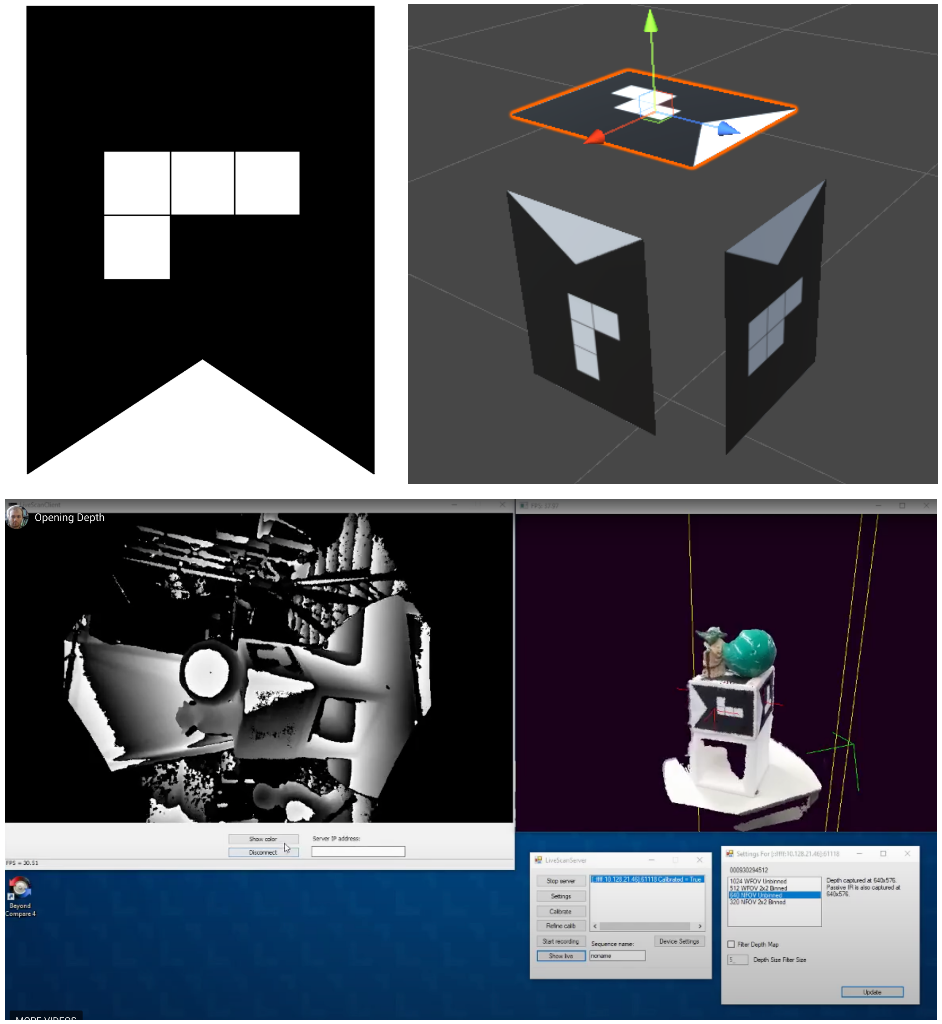
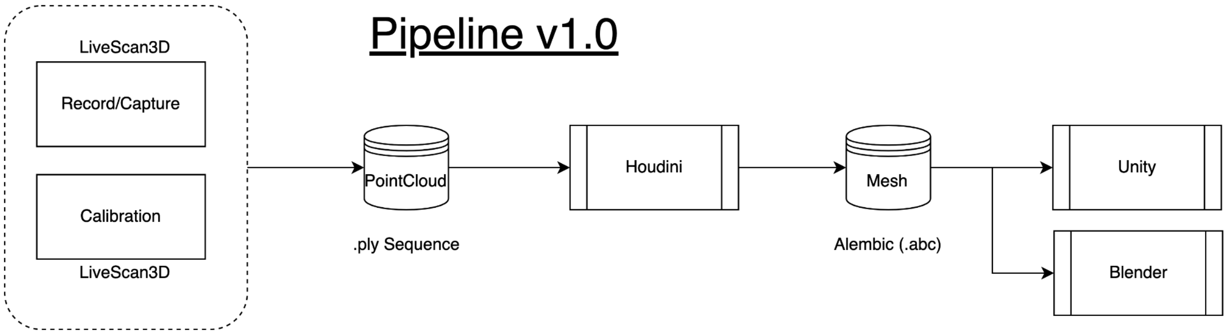


Figure 5: Top - calibration targets for LiveScan3D. Bottom - Depth image (left), point cloud (right) and client server (this is for a single camera connected).

[*Video: LiveScan3D - Single Camera - Toys*]

OpenSource Pipeline v1.0



Goal:

- Building on the Open Source LS3D pipeline, how can we create a pipeline that extracts a mesh of the resulting volumetric point cloud using standard visual effects packages.

Hardware Used:

- Azure Kinect (multiple)

Software Used:

- Same as Pipeline v0.1
- Houdini

Description:

- Once a .ply sequence is recorded from LS3D, the sequence can be loaded into Houdini and a mesh can be created using a number of fluid operations seen below.
- Output.
    - Houdini can output an Alembic (.abc) file which can then be loaded into Blender for rendering

Advantages:

- Houdini is well known and there is a free version. There is also an educational version that can be used as well.

Disadvantages:

- Learning curve for Houdini is quite steep
- The meshing pipeline to convert a sequence of .ply into a .abc is extremely time consuming, 30 seconds took about 10 hours of processing time to extract the mesh.

- Audio must be re-sync'd carefully afterwards and recorded separately

Figure 6: Point Cloud .ply sequence from LiveScan3D to Houdini. Houdini is used to mesh the point cloud sequence and you can see the resulting mesh both with colour and without.
Video: june22 hogue2 0 mp40001 0234

Links to More Video Examples:

- Closeup: close mp40001 0662
- Houdini Test 0
- Houdini Test - Fun in the lab
- Houdini - May14-nocolors
- Houdini - May14-vertexcolorstest
- Houdini - Test
- Houdini Test: 0001 0402
- Houdini- High Quality

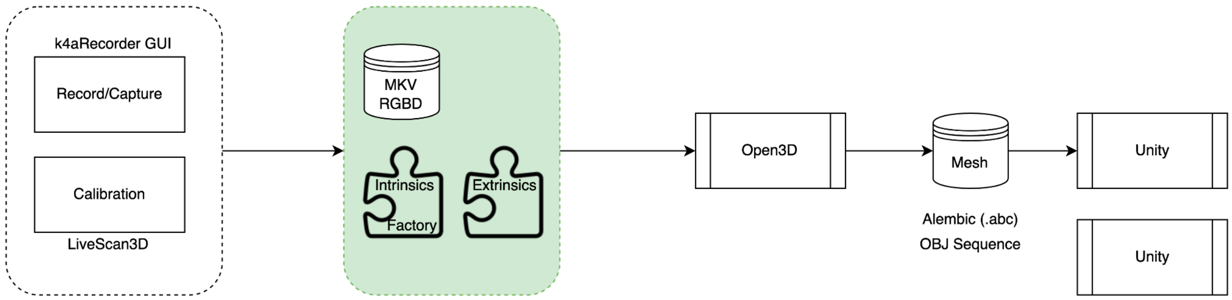
Painting tutorial tests:

- Houdini - Painting 1
- Houdini Painting 2
- Houdini Painting 3
- Houdini - Canvas stapling 1
- Houdini - Canvas Stapling 2

Closeup:

- Houdini - super close - mannequin
- Houdini - closeup person
- Houdini - punch
- Houdini - move in and out

OpenSource Pipeline v2.0



Goal:

- Rather than using Livescan3D to output the .ply sequence, can we use LS3D to provide the calibration transformations and record the raw video data from each Azure Kinect and develop software to transform this raw video into a usable mesh?

Hardware Used:

- Azure Kinect (multiple)

Software Used:

- LiveScan3D (for calibration only)
- k4aRecorder https://docs.microsoft.com/en-us/azure/kinect-dk/azure-kinect-recorder
- Open3D SDK + software we developed
  - https://github.com/Elite-Volumetric-Capture-Sqad/Open3D-Rendering

Description:

- To record the raw video data, the Microsoft Azure Kinect SDK contains a command line tool to record synchronized raw video data from the Azure Kinect cameras. This is called the ‘k4arecorder.exe’.
  - Output:
    - The k4a recorder saves a Matroska file (.mkv)
- We developed a software solution that loads in a set of .mkv files plus the calibration information provided by LS3D. This uses the Open3D sdk to project the RGBD information into a volume and extract an untextured mesh.
- As part of this solution, we wrote an exporter that exports a .obj sequence (so we can load the sequence into Blender) and an Alembic (.abc) exporter that saves the entire sequence into a single file we can load into Unity using our own .abc importer.

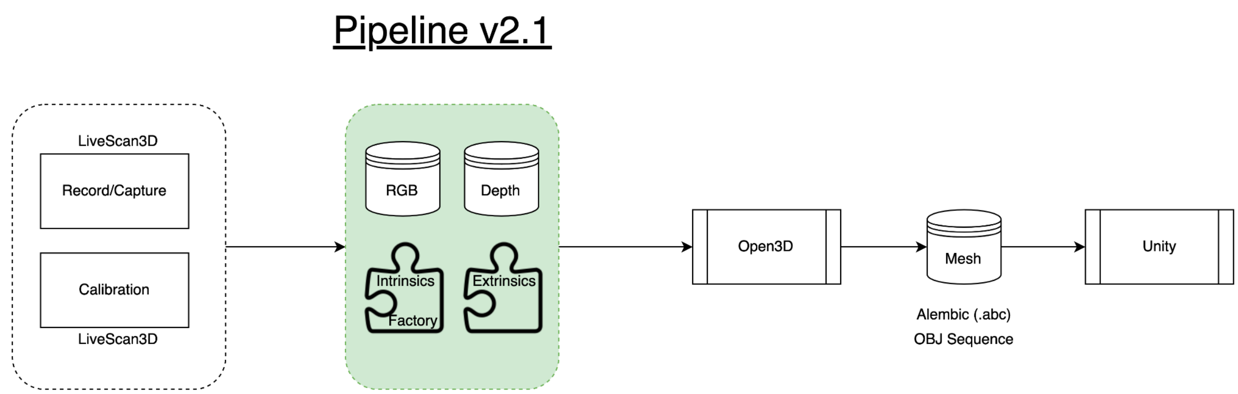
Advantages:

- Completely open source

Disadvantages:

- Difficult to setup correctly
- Open3D is extremely difficult to get working and compile
- It is difficult to get high quality meshes using the Open3D SDK functions
- There are no easy texturing solutions so the output contains vertex colours only.
- K4arecorder.exe is tricky to use on multiple machines and produces raw output that is not always synchronized.
- The raw data is in the range of 0.5-1GB/second of data for the .mkv files

Pipeline v2.1



Goal:

- The goal here was to modify LiveScan3D to output the raw image files and process with our Open3D pipeline.

Github:

LiveScan3D development branch with Raw data export:
https://github.com/Elite-Volumetric-Capture-Sqad/LiveScan3D/tree/development

Description:

- Building on the v2.0 pipeline, we removed k4arecorder.exe from the mix and with the help of researcher Chris Remde in Germany, we developed a special branch of LiveScan3D that is capable of saving synchronized raw images from each camera
- We tested the system on a single PC with 6 cameras and were capable of recording raw RGB-D synchronized images at 30fps.

Advantages:

- Removing k4arecorder.exe allowed us to ensure that cameras were synchronized and triggered at the same time reducing extraneous data and also allowed the LS3D solution to mature into a system that contains the ability to record raw data for further processing as well as output .ply and perform calibration.

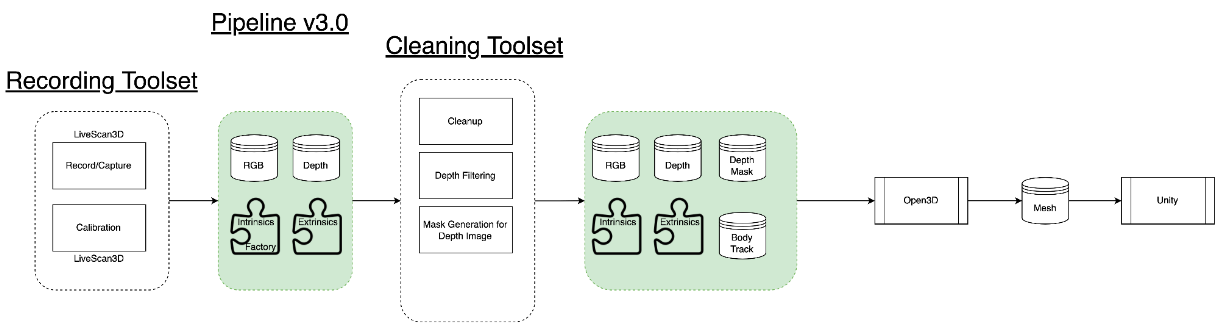
Disadvantages:

- Same as v2.0 for the Open3D issues

Links to videos:

- [LiveScan3D - Calibrating multiple cameras]()

Pipeline v3.0



Goal:

- The goal of this was to utilize the v2.1 LS3D recording toolset for capturing raw data but then develop a set of tools to post process and experiment with different algorithms.

Software Used:

- V2.0 software plus:
- DDRNet (Depth Denoising and Refinement for Consumer Depth Cameras using Cascaded CNNs)
  - https://github.com/neycyanshi/DDRNet
- BackgroundMattingV2
  - https://github.com/PeterL1n/BackgroundMattingV2

Description:

- Experimented with DDRNet to improve the depth map output and the BackgroundMattingV2 machine learning approach to extract the background from our camera images
- DDRNet.
  - We found that while we were able to get the DDRNet scripts to work, the approach did not aid in producing a viable improvement to our depth map results. The difficulty for this was that the model DDRNet uses is good for a single camera but when using multiple cameras the data is transformed in such a way that they are inconsistent and inaccurate and cannot be combined.
- BackgroundMattingV2
  - We found that this approach works extremely well in segmenting the background from the foreground. The approach requires you to take an image without the actor in place first and then we can run the scripts on our data sets to extract a Black and White matte image where black means background pixels and white means foreground pixels.
- Open3D tool.

    - We modified our Open3D-based tool to accept these matte images as well and we found that this reduced the noise tremendously and as such our reconstruction quality improved.

Advantages:

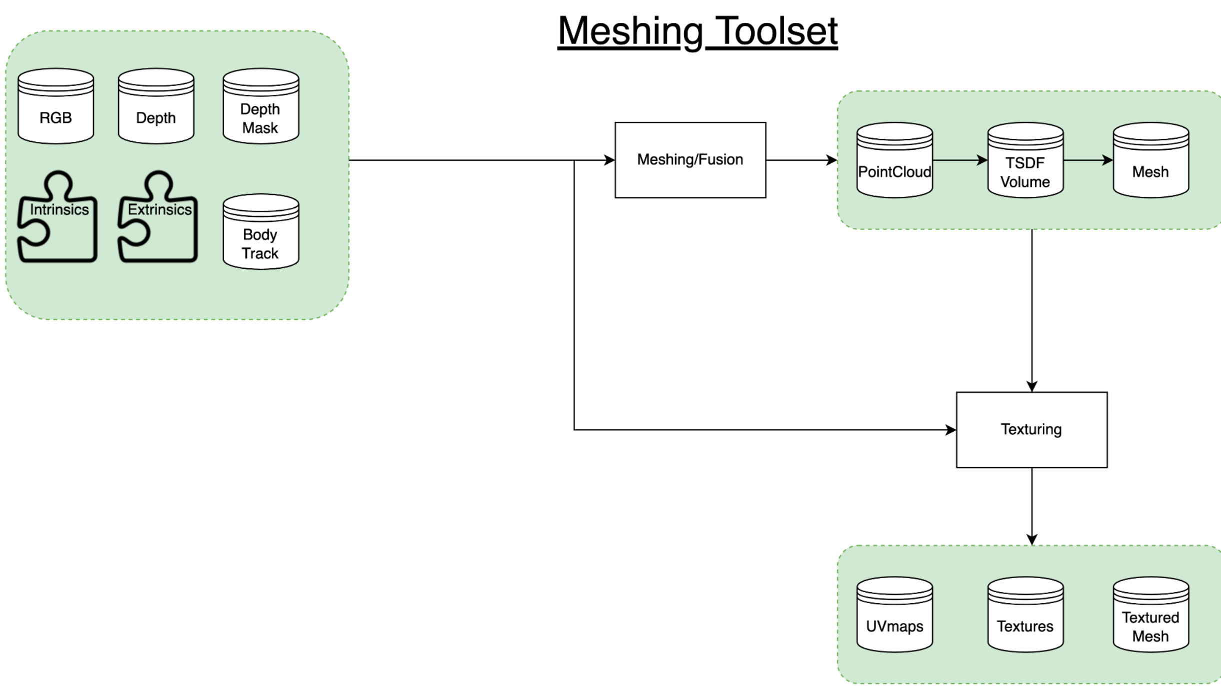
- Background removal is extremely important to isolate regions of the image that matter. This reduces noise in the final result and increases performance as the background pixels need not be processed.

Disadvantages:

- Open3D SDK still proves to be quite cumbersome and produce bad results due to their meshing approach.

- Colour:
    - Video: Colour Sequence
- Matte Extracted:
    - Video: BG Matte of Colour Sequence using Machine Learning

Pipeline v3.1 - SimpleTSDF



Goal:

- Our goal was to improve the mesh quality by writing our own solution for 3D reconstruction we called SimpleTSDF.

Github:

- https://github.com/Elite-Volumetric-Capture-Sqad/volumetricpipeline

Description:

- SimpleTSDF is a standalone .exe that we developed that takes a LiveScan3D Raw recording directory that includes the Background Matte already extracted and converts this to a mesh using a Truncated Signed Distance Field approach.
- The algorithm:
    - Initialize a Voxel Grid of specified resolution (128x128x128)
    - For each frame (independently), load in all of the (RGB, Depth, Matte) + calibration
        - Project each voxel onto each camera and keep only the ones that fall onto the foreground using the Matte images as a guide
        - Process the remaining voxels by storing the distance to the surface determined by the minimum consensus distance from all of the cameras that can see the voxel. The signed distance is truncated at a maximum distance specified by the user.

        - Once the TSDF volume is generated, using Marching Cubes extract a mesh.

Advantages:

- Improved mesh quality over Open3D implementation
- Can create a watertight mesh

Disadvantages:

- Limited by resolution of voxel grid which is memory intensive. This can be alleviated in the future using better data structures.
- Fairly slow but still faster than Houdini approach.

Links to video:

- SimpleTSDF - low resolution
- SimpleTSDF - higher resolution

VolNodes

Goal:

To create an open source system for processing volumetric data with the primary goal being 'ability to experiment' with algorithms rather than focusing on performance issues.

Github:

- https://github.com/Elite-Volumetric-Capture-Sqad/volumetricpipeline
- In VolNodes/ sub directory

Description

- VolNodes developed as part of this project to create a visual programming language focused on the pipeline developed above and with the ability to add/remove algorithms into the pipeline with ease.
- We have implemented (to date) the following nodes:
  - LiveScan3D Directory Node
    - A Node that when passed a directory of a LiveScan3D Recording, parses the directory and creates outputs for the extrinsics, intrinsics, and individual camera directories
  - Camera Dir Node
    - A Node that parses a given camera recording directory and creates outputs for a Color Image, a Depth Image, a Matte image and calibration information.
    - Can specify which frame to output and drive that with a separate node
  - VoxelCarveTSDF
    - Takes a list of camera information (images[rgb,depth,matte], and calibration information) and extracts a mesh and outputs it on an output node
  - MatteExtractor
    - Wrapper for the Background Matte Extractor
    - Takes a reference image and image to process, outputs the matte image png
  - ExtrinsicsLogParse
    - Parses the camera calibration file from LS3D and outputs transformation matrices that can be used by other nodes

  - GLMeshView
    - A node that has an OpenGL Display to view a mesh as input to it
  - MeshIORead
    - Read a mesh from file
  - Get Filename
    - Utility node to ask the user for a filename (specify with file dialog)
  - Get Directory Name
    - Utility node to ask the user for a directory name (specify with file dialog)
  - Print Node
    - Debug node to print the input pin as a string
  - Read Image
    - Read image from filename, output is the actual image data
  - Display Image
    - Display a passed in image, good for checking the camera directories
  - OpenGL Viewer
    - Base class node for all 3d viewing capabilities

Advantages:

- Can try out new algorithms and tweak their parameters with a visual interface
- Made for the non-programmer
- Easy to add new Nodes for processing data

Disadvantages:

- Focus was not on performance so it can take time for final processing of long sequences
- Development in flux but active

Gaps

- No simple texturing solution that takes in a set of cameras (images), their calibration parameters, and a mesh and exports a single texture that correctly maps the colour images onto the model.

Other tools we experimented with:

- InstantMeshes
  - https://github.com/wjakob/instant-meshes
  - Great tool for Retopologizing 3D meshes and reducing complexity while maintaining shape
  - Note: you MUST ensure that close vertices are ‘welded’ together (i.e. using MeshLab) PRIOR to using InstantMeshes to ensure a good result.
  - Verdict: Need to integrate functionality into VolNodes
- Meshlab:
  - https://www.meshlab.net/
  - Gui-based toolset for various mesh refinement algorithms

    - This is a great tool for working with single frames
    - Does have texture projection but it is buggy currently
    - Scriptable with pyMeshLab
        - https://pymeshlab.readthedocs.io/en/latest/
    - Verdict: Need to integrate functionality into VolNodes
- MVS-texturing:
    - https://github.com/nmoehrle/mvs-texturing
    - A promising texturing codebase for creating textures. We experimented with this but could not get it to work properly with our dataset and calibration information.
    - Verdict: promising but requires work to incorporate it properly and verify its correctness
- SSMVTex
    - https://github.com/rafapages/SSMVtex
    - Verdict: older codebase that would not compile or run properly with current toolset

Future Nodes for VolNodes:

- Add nodes that wrap functionality for:
    - InstantMeshes to retopologize
    - pyMeshLab functionality
        - Welding close vertices
        - Potential for using texture projection (in the future)
        - Poisson Reconstruction for Point Clouds
        - Mesh Editing operations
    - Wrap Blender scripting interface in a node to trigger the creation of a compressed GLTF (.gltf) or Alembic (.abc) animation file from a sequence of .ply or .obj files.
- Add nodes for Machine Learning algorithms to find faces and apply various mesh operations to enhance textures in these areas
- Add nodes for tracking Volumes of Interest (i.e. track faces/heads/hands/feet) and provide that as output each frame for other algorithms that require this information.

Link to Presentation on VolNodes:

- Presentation Slides: VolNodes - Feb 11 Symposium

Links to video:

- VolNodes v0.1

Solutions for Capturing Volumetric video

Hardware

For our purposes we are using RGB+D Azure Kinect cameras as our basic hardware camera setup. However, to develop a usable setup for multiple cameras, I recommend the following additional hardware based on our experience throughout this project.

PC Configuration

- CPU: 11th Gen Intel® Core™ i9-11900F (16 MB cache, 8 cores, 16 threads, 2.50 GHz to 5.20 GHz Turbo)
  - LiveScan3D is very CPU intensive, so more cores means better performance and more cameras. This is the minimum and might have issues with 7+ cameras.
- Graphics Card: NVIDIA GeForce RTX 3070+
  - A good graphics card is needed for all volumetric processing
- RAM: 64 GB, 4 x 16 GB, DDR4, 3466 MHz, XMP
  - It is important to not only have a lot of RAM but that the RAM is FAST and distributed as much as possible increasing parallelism and decreasing bottlenecks. Thus, higher throughput (faster) plus more channels (dual at minimum, quad or more preferred)
- 2 TB SSD
  - Raw Volumetric video data is huge so you will require as much HD space as possible. To ensure recording is done at proper framerates you need fast SSD drives.
- PCI USB card::
  - StarTech PCIe USB 3.0 Card with 4 Dedicated 5Gbps Channels (1 for every 4 cameras).
  - We use 2 of these cards for 8 cameras connected to one PC. This is FUNDAMENTALLY necessary due to data bandwidth from the cameras.
- Optional:
  - Green or Black backdrop to block out any extraneous light (i.e. sunlight)
  - LED Lights if you wish to control the lighting conditions (standard photography/video lighting setup)

Miscellaneous Hardware

- Tripods
  - We recommend speaker stands from Amazon Basics or similar:
    - Amazon Basics Adjustable Speaker Stand - 4.1 to 6.6-Foot, Steel
- Clamps
  - LimoStudio Super Clamp with Standard Stud for Photo Photography Studio, AGG1108 : Amazon.ca

- Sync Cables
  - 3.5mm male to male sync cables (standard audio), 25 foot+
  - TRS (Tip, Ring, Sleeve) or TRRS (Tip, Ring, Ring, Sleeve).
- USB 3 Extenders
  - CableCreation USB 3.0 Extender or Unitek USB 3.0 Extender.
  - 25’ feet

Software

Livescan3D (modified)

URL:

- https://github.com/Elite-Volumetric-Capture-Sqad/LiveScan3D/tree/development

Description:

LiveScan3D is a free and open source solution to record point clouds and (now raw image data) from multiple synchronized Azure Kinect cameras. The version above is a forked and modified version of the original that has many features that make volumetric video capture more accessible. Throughout this project, we relied heavily on LiveScan3D for capturing synchronized data for our testing purposes.

Advantages:

- Network infrastructure (client/server). Cameras may reside on one or multiple PCs on a local network.
- Calibration works in seconds making it possible to reconfigure and recalibrate quickly.
- Realtime preview of point clouds
- Single or Multi-Camera solution
- Multiple Export Options:
    - Point Cloud sequence (.ply)
    - Raw Data sequence (RGB, Depth)
- $Free and Open Source

Disadvantages:

- Development in flux but stable
- Tricky to set up initially
- No audio support
- When raw data exports are used, preview is disabled to ensure maximum throughput. Thus actors must trust that they are within the volume as they won’t see the results until after processing the data.

DepthKit

URL:

- https://www.depthkit.tv/

Description:

Depthkit was the first RGB-D product that was commercially available in the market. They have single camera (DepthKit Core) and Multi-camera (DepthKit Studio) solutions. We used DepthKit Core for our single-camera recordings for this project and DepthKit Studio for our StudioArts recordings at Cinespace studio in Toronto.

Advantages:

- Free trial for single-camera solution
- Records Audio
- Integrates with Unity 3D

Disadvantages:

- Monthly fees - single camera is reasonably priced, multi-camera is expensive
- Free trial only includes single-camera solution
- Calibration process is extensive and can take hours each time so ensure you fix your cameras in a way that they cannot be moved, touched, or vibrated easily.

Soar Capture Suite:

URL:

- https://www.streamsoar.com/

Description:

Throughout this project we connected with Soar, a company in the U.S. developing a multi-camera volumetric solution called the "Soar Capture Suite". Their software provides real time recording of volumetric video with multiple Azure Kinect's and simplifies the capture process. We used an early prototype version of Soar's Capture suite to record Studio Arts performances for this project and they came out great, however as it was a prototype version it did not include the audio recordings. Since they have included Audio in the Capture Suite and we are looking to re-record these in the future and make them available through our Volumetric Discord Community.

Advantages:

- Calibration process is easy and fast.
    - Within seconds the cameras find and calibrate themselves by viewing a common target which makes experimentation with multiple cameras very easy to do.
- Output Quality is good
- Export options:
    - proprietary compressed format
    - textured OBJ sequences
    - Raw data stream, which can be reprocessed in various formats afterwards
- Records Audio
- Integrates with compressed output in Unity3D
- Mac, iOS, Windows, Android compatible
- Live Streaming capable to mobile devices
- Currently uses the same Calibration markers as LiveScan3D so you can use the same target for both (ensure you have them in the order that Soar requires though)

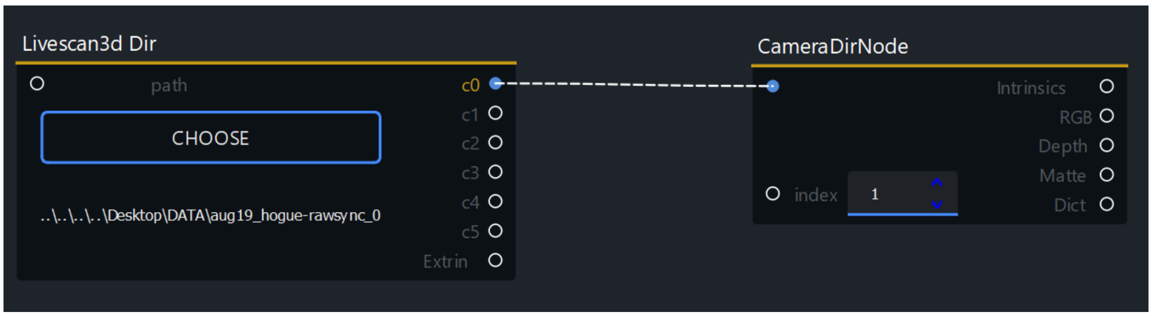
Disadvantages:

- Interface is not as polished (still in development)
- Expensive but flexible pricing
- Free trial available for testing

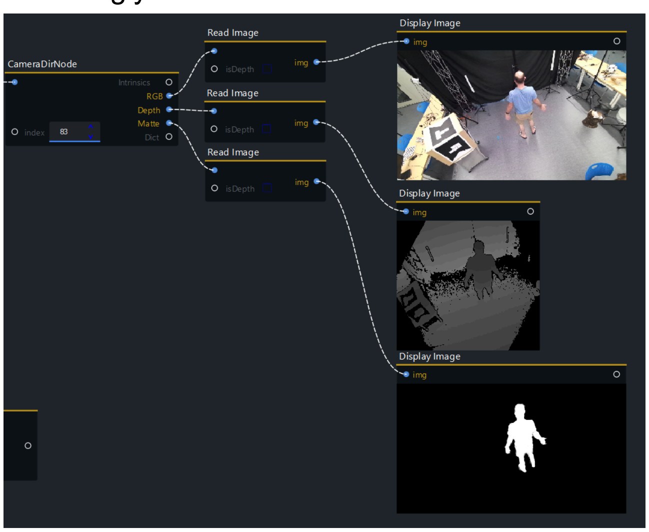
Processing Volumetric Video

Volnodes

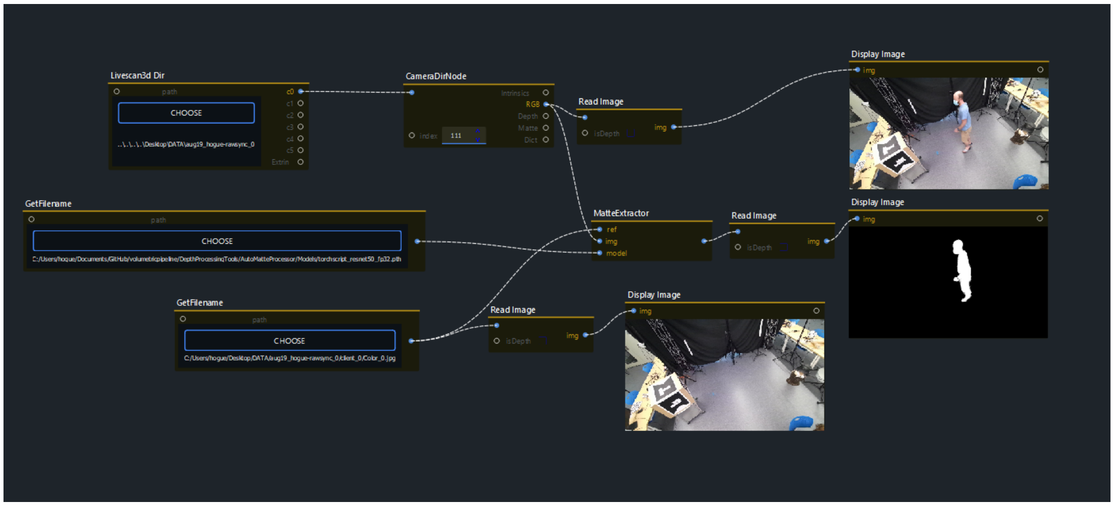
- In VolNodes the first step (after capturing a recording from a LiveScan3D session), you can drag in the “Livescan3d Dir” node and choose the directory where your recording lives on your hard disk. This automatically creates output pins for each camera and extrinsics (camera transformations from the calibration).
- The CameraDirNode can then be connected to each of these output pins which will automatically parse the directory and fill the output pins with the filenames of the images for the frame selected at the current frame index. Note we can connect another node to the frame index input to drive this for automatically processing the entire sequence once we are happy with our single frame pipeline we create.



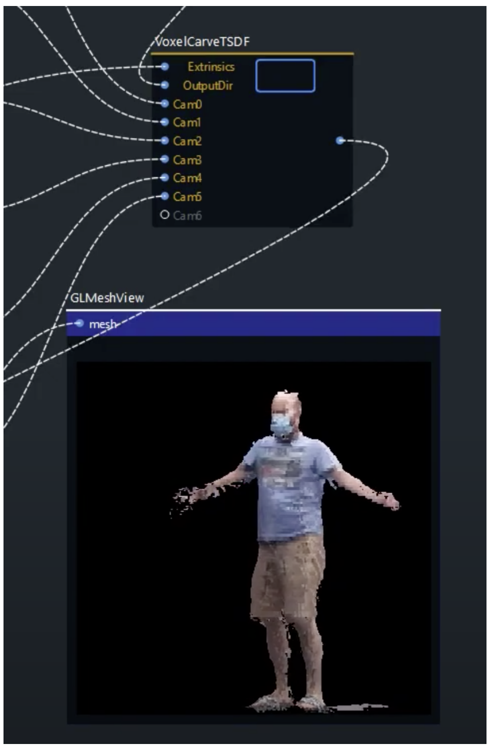
- The “Read Image” Node allows you to connect the outputs from the CameraDirNode to read the image from disk and using the Display Image node you can display them to ensure they are the correct frame of data you wish to process. These automatically update and determine if the image is a colour/depth/matte image and displays them accordingly.



- BG Matte Extraction can be done through VolNodes also using the ‘MatteExtractor’ Node. This requires you to specify the filename of the pre-trained model which can be selected in the “GetFilename” node.



- The ‘VoxelCarveTSDF’ Node is used to convert the input images from the cameras and their calibration transformations and output a 3D Mesh



- Simply by changing the index of the CameraDirNode the pipeline will automatically update all of the connected nodes and save the mesh to disk as an .obj sequence which can then be loaded into Blender and output as a .gltf or .abc file for use in other applications.

LATK

Description:

Artistic Experimentation with LiveScan3D output for artistic volumetric video.
Another option is to use the LATK (Lightning Artist Took Kit) which has been developed and modified by our team (Nick Fox-Gieg) to work with LiveScan3D output .ply point cloud sequences.

The Lightning Artist Toolkit (Latk) is a complete pipeline for hand-drawn volumetric animation, as far as I know the only open-source example of its kind. The goal for this sub-project was to integrate live-action volumetric video with hand-drawn volumetric animation.

- An applied ML solution to convert volumetric video into brushstrokes requires a collection of data, and prior approaches primarily relied on 2D drawings. Fortunately, Google's Tilt Brush, and its open-source successor Open Brush, have become popular enough with the general public to provide meaningful quantities of suitably licensed 3D drawing data in public archives.
- A model was trained on pairs of depth maps and normalized XYZ coordinates reconstructed from camera intrinsics. This approach proved highly temporally unstable, impractical for animation.

Vox2Vox

Using selections from a dataset of over 4,000 artworks, a point cloud (.ply) is read from file, voxelized, and trained a brushstroke segmentation pattern using Vox2Vox.

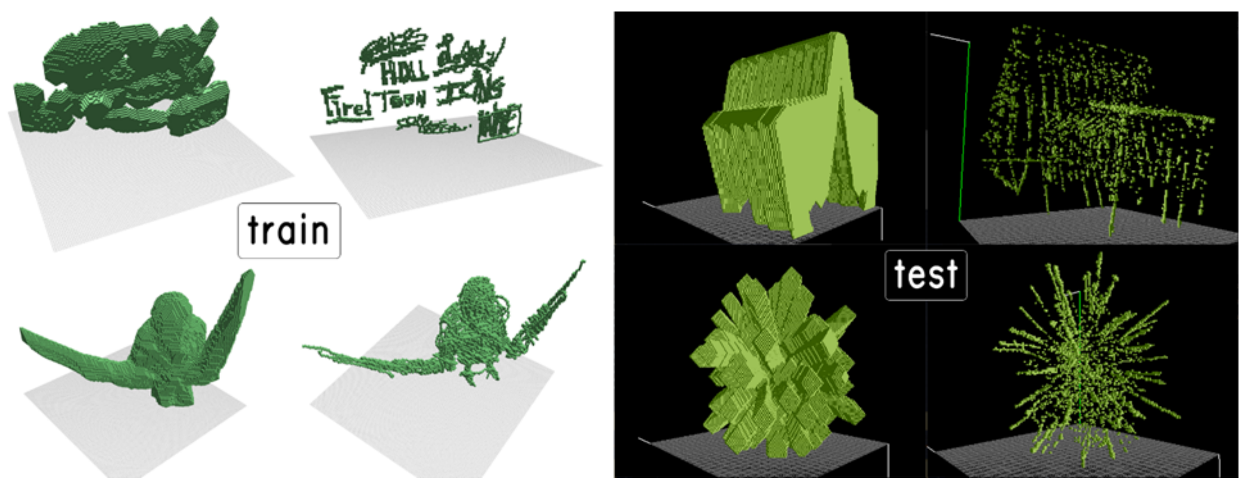


Figure 7: Training and testing examples using Vox2Vox.

The voxel output is translated into a network of sparse brushstrokes using the libigl-based SynDraw tool. Once this system generated the final brushstroke information, it was saved in a compressed JSON format readable by industry-standard 3D creation software such as Blender, Maya, and Houdini. See below for some examples of varying the parameters and experimenting with different styles.

Links:

- Vox2Vox:
  - https://github.com/mdciri/Vox2Vox
- GitHub repository:
  - https://github.com/n1ckfg/latk-ml-003
- Google Colab notebook for training: n1ckfg-vox2vox-pytorch-TRAIN.ipynb
  - https://colab.research.google.com/drive/1HoWkeXPRfKCBR3OPJhuNnOoNI3MazgNA
- Google Colab notebook for testing: n1ckfg-vox2vox-pytorch-TEST.ipynb
  - https://colab.research.google.com/drive/1ZEddv6dbWpNfWnhXC9_M5d2WMlkaRyOh
- Latk file format library repository:
  - https://github.com/LightningArtist

Demo:

- https://lightningartist.org

Notes on software and hardware support:

- At present, you can read, write, and edit Latk files using libraries written in Python (Blender, Houdini, Maya), C# (Unity), JavaScript (WebXR, three.js, p5.js), Java (Processing), and C++ (openFrameworks, Unreal).
- Experimental read- or write-only support is available for a wide range of other formats, including SculptrVR CSV, ASC point cloud, Graffiti Markup Language (GML), Corel Painter script, NormanVR, VRDoodler, Tilt Brush / Open Brush, Quill, Sketchfab FBX sequence, SVG SMIL animation, After Effects JSX, and Unreal XYZ point cloud.
- XR hardware support includes SteamVR (Vive, Index, WMR), Oculus desktop, Android headsets (Quest, Mirage, Magic Leap, HoloLens 2), UWP headsets (HoloLens 1), and iOS / Android Phones (ARKit, ARCore, ARFoundation).

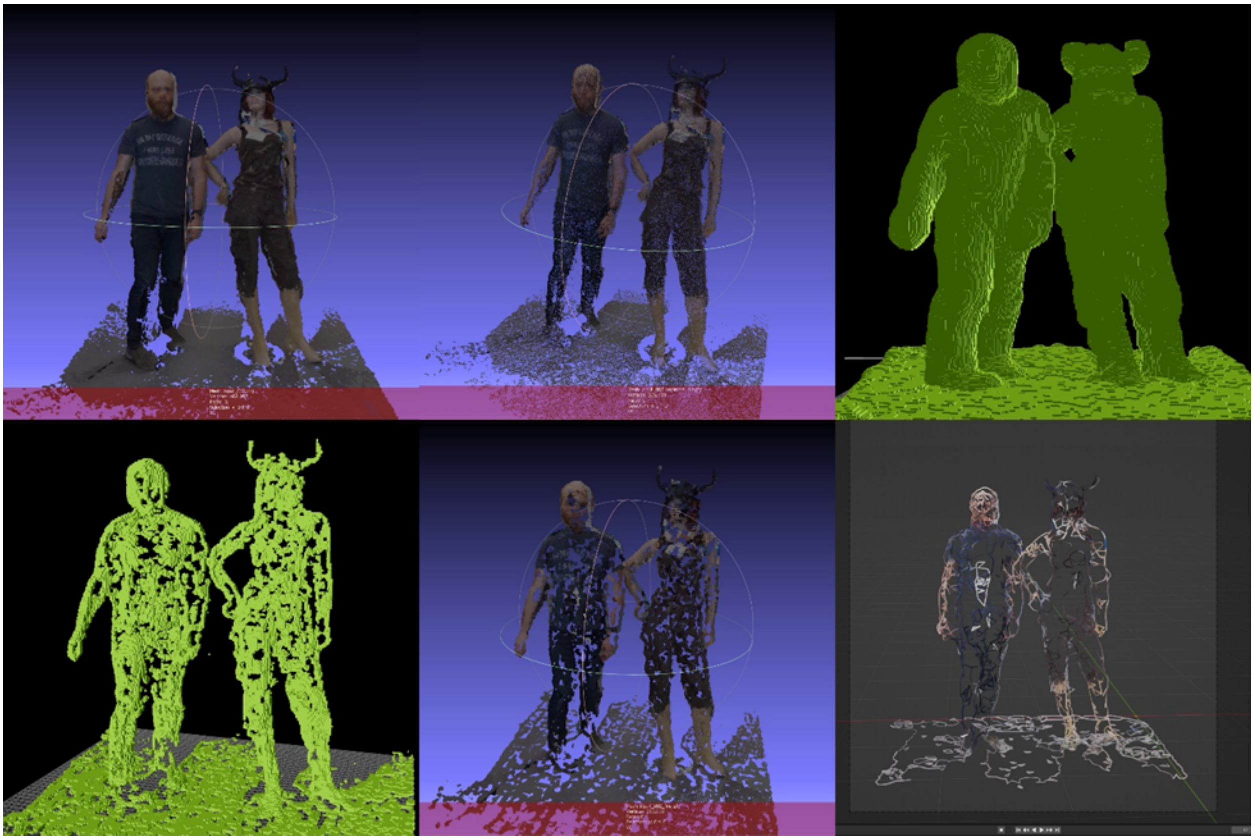


Figure 8. Output of each step in the Vox2Vox pipeline.

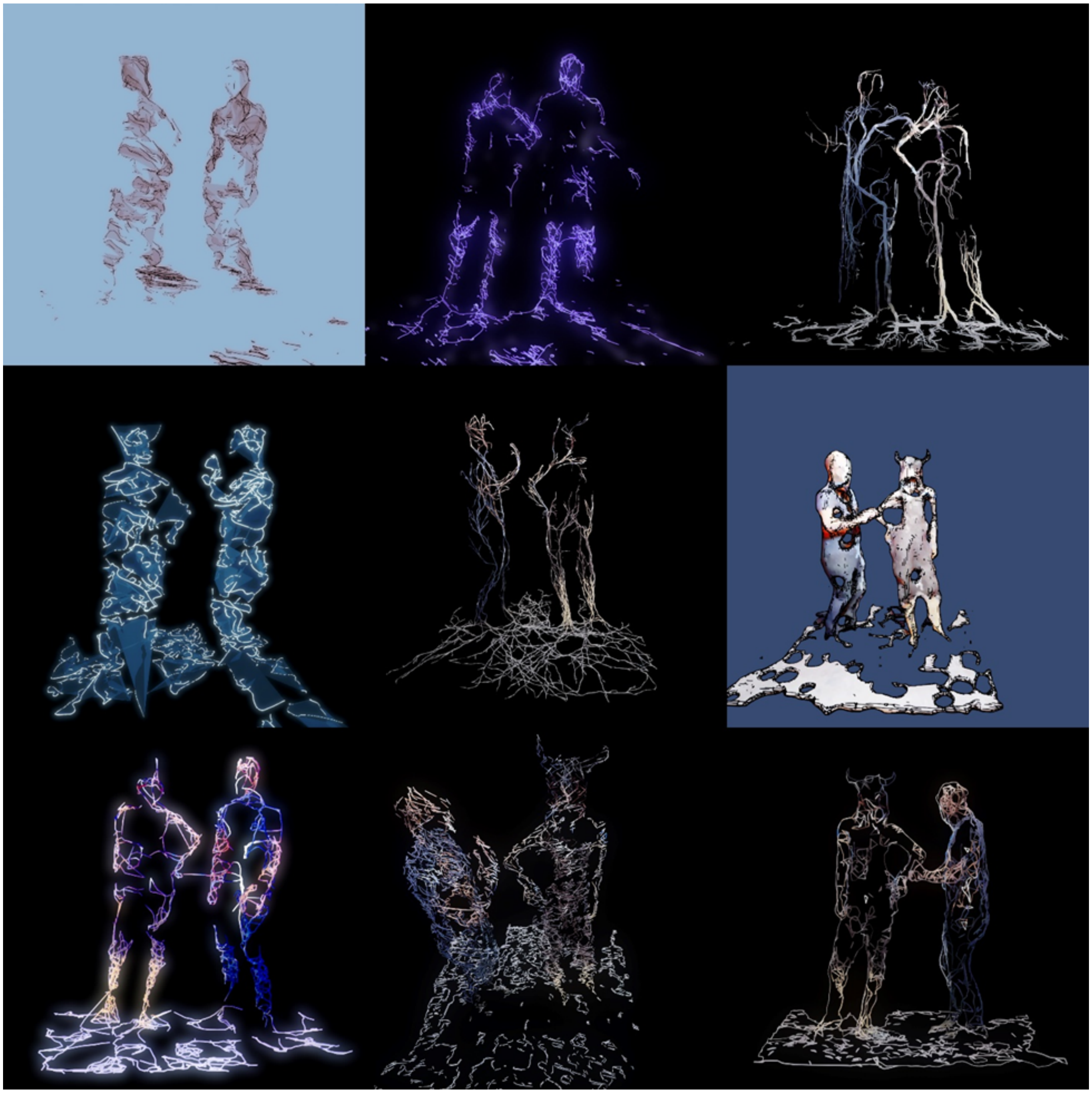


Figure 9. Nine different tests of Vox2Vox using the same volumetric capture sequence

Best Practices

Hardware setup

Lighting conditions

- Due to the technology used in Azure Kinect, you cannot use it effectively in direct sunlight. We found that exterior windows should be blacked out effectively to reduce noise.
- Standard photography or video lighting is helpful for increasing quality of the performances, standard halogen or fluorescent lighting is not recommended

Multiple cameras

- Single camera solutions are fine for situations where you are viewing the scene from one direction mainly however that defeats much of the purpose of volumetric video. Thus we recommend at least 4 cameras and a multi-camera software solution for recording a small volume.

Camera Positioning

- Position the cameras specifically for each shoot and calibrate accordingly, this might require you to re-print smaller or larger calibration targets and re-measure them for LS3D
- For a full-body area, use 8 cameras pointing inwards
  - 1x Front - eye level
  - 1x Side-left - mid torso height
  - 1x Side-right - mid torso height
  - 1x Back-Super High - pointed downwards (to see top of head)
  - 1x Front - low pointing upwards
  - 1x Front-Left (or right) corner low pointing upwards
  - 1x Back-Left - low pointing upwards
  - 1x Back-Right - low pointing upwards
- Minimal 4 camera setup pointing inwards
  - 1x Front - eye level
  - 1x Side-left mid torso height
  - 1x Side-right mid torso height
  - 1x Back - mid torso height
  - Note: you may have issues with some areas due to lack of visibility and occlusion but this is a decent setup for single actor full-body
- The Goal for positioning is to balance visibility with resolution. The more cameras you have the better you can accommodate visibility issues because you can position them to directly view those areas. But be wary of resolution. The Azure Kinect has a sweet spot for data around 0.75m - 2m from the camera.
- Camera Orientation:

    - Azure Kinect Cameras should be placed vertically to maximize resolution and field of view of the cameras when possible. This will allow you to move them closer together and still achieve a full-height view of a person.

Calibration for LS3D:

- We found that a CUBE with markers on each side of the cube (and the top) works the best
- Ensure you measure the cube and marker locations accurately, the more accurate your measurements are the more accurate your calibration will be.
- Ensure your calibration cube is manufactured to a high standard of accuracy and use clamps where possible to ensure corners are 90 degrees

Materials

- Do NOT use Transparent materials such as glass or plastic bottles
- Brighter colours work better than dark materials.
- Matte Materials work better than shiny materials
- Shiny Black Hair should be avoided, spray with a matte hairspray
- Shiny foundation makeup should be avoided, use a matte powder makeup for your actors
- Small and thin objects are hard to reconstruct, i.e. brush handles are hard unless you bring cameras closer to achieve higher resolution on these objects

Recording Process:

- Plan your recordings in advance
- Practice each take for time and content
- Organize each take by name, time, date
- Each Take should be short (< 2 min)
- Allow for a few seconds before and after a take of silence
- If you are using external audio recorders then ensure you clap loudly (or have an audibly loud BEEP) at the start AND end of each take to ensure you can synchronize them. It is best to also have a visual component to the ‘clap’ (actors can clap with their hands, or assistants can use a standard film clapboard).
- Upload raw data to a shared drive immediately for backup
- For each Recording, ensure you capture the space without actors or props for a few seconds to ensure you have data for future background removal.
    - This should be performed at each re-calibration
- Calibration MUST be done prior to the recording as close as possible to the actual take
- Re-calibration MUST be done any time a camera is touched or moved or vibrated
- Ensure you make production notes for each take noting any mishaps or need to recalibrate

Capture Sessions for the Studio Arts

We performed 4 capture sessions to develop the volumetric tutorials and provide insights, lessons learned, and links to the resulting data for use by other educators.

Capture Session 1: OCAD University

Capture Setup:

- Single-camera Azure Kinect, laptop, LED lighting, green screen backdrop
- Depthkit Core

Description:

We conducted a series of test volumetric captures at OCAD University using a single sensor Depthkit setup.

The purpose of these tests were a) to trial-run the production process in advance of more refined capture demos; b) to generate some preliminary content that could be used to mock-up an online interface/portal for student access. Through these tests we were able to identify issues with various capture materials and processes (for example, challenges in capturing shining and delicate tools), and decide on tutorial types that might work best given these constraints.

Graduate Research Assistant Yutong Zheng used these test captures to create an example web portal for students to access these volumetric tutorials: https://yutongzheng0810.wixsite.com/my-site-9.

The demo web portal offers students the ability to access the volumetric content as stand-alone tutorials, or as part of an integrated WebVR experience. For increased accessibility, the site uses WebVR (A-Frame).

Insights:

We found that this was the simplest and inexpensive way to have new artists jump into the volumetric video creation process. The single-camera solution is easy to setup and requires no calibration making the initial capture setup much simpler than a multi-camera setup. The resulting file format contains all audio, video, depth information and can easily be transferred to a WebVR platform with the provided plugin.

Challenges

- We found that although simple to set up, the single camera solution has its limitations when viewing from alternative angles due to the fact that the system just does not have the information from other angles. For situations where you need users to be able to move around the recording to view from other angles this solution will visually break down after about 45 degrees due to mesh inconsistencies and texture stretching. For these scenarios, multi-camera solutions are required.

- Due to the optics in the Azure Kinect, it is better to use natural light when viewing paint rather than harsh LED studio lighting. The paint colours are much more natural and vibrant when viewed with more natural light settings however this comes with production and procedural considerations (time of day, location, windows).

Links:

- Volumetric Tutorials for the Studio Arts:
  - eCampus Volumetric in the Studio Arts
- Data:
  - Studio Arts - Paint - SingleCamera - Depthkit - Recordings

Capture Session 2: Cinespace

Capture Setup:

- Multi camera (x5) Azure Kinect
- Depthkit Studio
- Professional capture space and lighting

Description:

We set up a multi-sensor Depthkit volumetric capture of a series of painting tutorial demos by Veronika Szkudlarek. This series was captured at York University's Cinespace stage.

Insights:

- Once the capture space was properly calibrated, the capture process itself was unobtrusive and did not interfere with the artist's practice.
- The volumetric capture clips, once processed and integrated into a 3D environment (in this case Unity), allow a student to move in and around the painting demo, allowing them to improve their viewing position, and/or move closer to the artist giving the tutorial.

Challenges:

- Post production processing time required expensive software licenses that were inaccessible offsite.
- The time required for processing, even on a powerful computer, was fairly lengthy (in some cases up to 6 hours of post production time was needed for roughly 20 minutes of volumetric capture), which hampered our efforts to produce a more extensive library of clips.
- It was nearly impossible to evaluate whether a capture was successful until this processing was complete, which created further challenges for our pipeline.
- The size of the capture meant much time was spent transferring and processing files.
- Finally, while the quality of the clips we produced could potentially be improved (with better lighting and more time spent refining the clips), the sensors still struggled to capture small objects, including knives, brushes, and containers, creating glitchy artifacts surrounding the image.

Links:

- Images and Videos from Session:
  - Cinespace Session: Images and Videos
- Data and Build:
  - Cinespace Session: Studio Arts Recordings

Images from session

Figure 10: Left: The eCampus team, Right: 5x Camera Azure Kinect Setup at Cinespace Studios

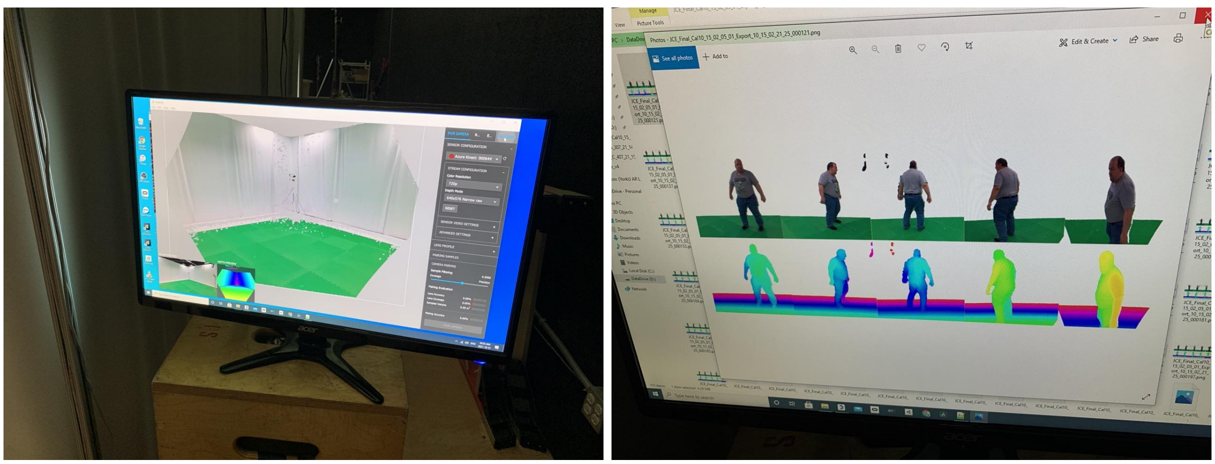


Figure 11: Left: DepthKit Studio view of the space Right: segmented color and depth output per camera

Figure 12: Left: Calibration target (calibration is done in pairs) Right: Live preview of point clouds

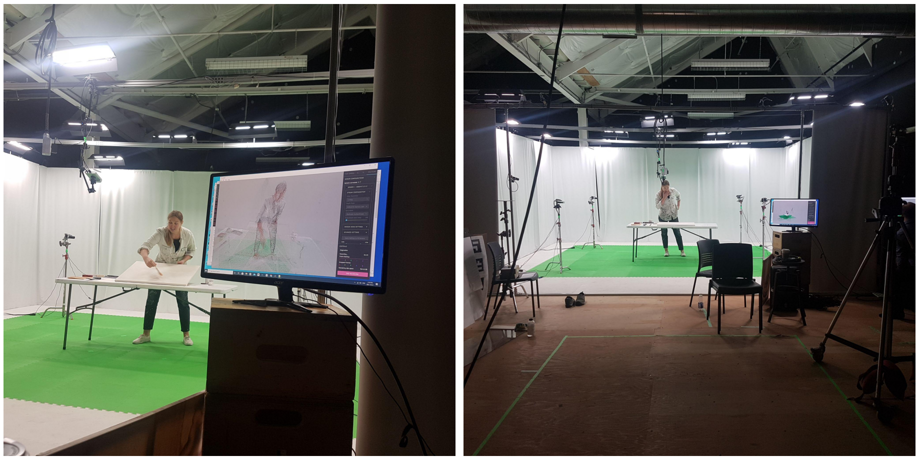


Figure 13: Live Capture Session in progress

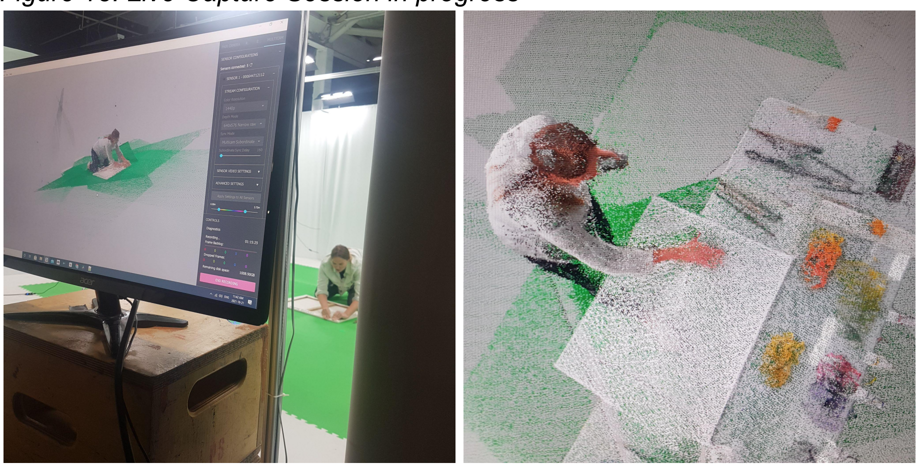


Figure 14: More images from live capture session and preview point clouds

Figure 15: DepthKit Studio Output in Unity 3D Game Engine

Capture Session 3: Ontario Tech

Capture Setup:

- Multi camera (x6) Azure Kinect
- Soar Capture Suite

Description:

We set up a multi-sensor Soar volumetric capture system and captured a series of painting tutorials by Veronika Szkudlarek. This series was captured at Ontario Tech's Game Development and Interactive Media Lab.

Insights

- The calibration process for Soar was very easy to perform (< 1 minute from start to finish).
- While the intent of the capture suite is mainly for single performers, our setup required a single performer plus a number of props (paint, brushes, tables, canvas). The setup worked well but required significant experimentation with the camera locations. The ease of calibration of the system made this experimentation possible.
- The capture process was easy to perform and recording was done by naming your recording output files and pressing record.

Challenges

- At the time of our recording, the Soar Capture Suite did not have Audio recording capability (this has since been added to the new versions) as such our recordings do not have audio embedded in them. We recorded audio separately, yet due to the lack of tools for editing these recordings the audio cannot easily be synchronized. To rectify this, we aim to re-record with the newest version of the capture suite in the near future with audio embedded in the files.
- The Lack of editing tools available for this format of volumetric video made it difficult to post-process. We do however provide the data in our data folder for others to use and explore.

Links:

- Images and Videos from Session:
    - Ontario Tech Session: Soar: Images and Videos
- Recordings and Data from Session:
    - Ontario Tech Session : Soar: Recordings

Images from Session

Figure 16: Left: Setting up the lights and capture area. Right: the calibration target

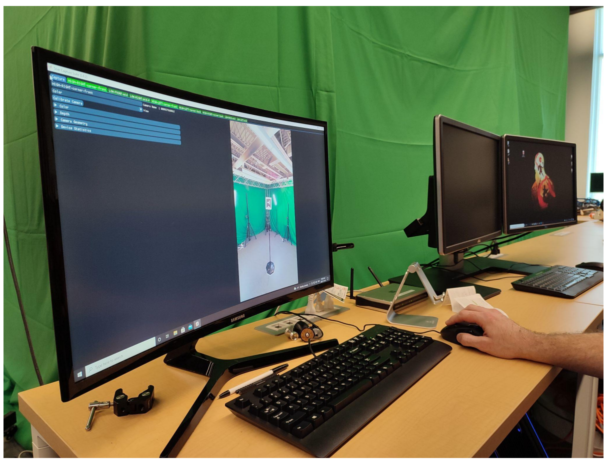


Figure 17: Left: Calibrating with Soar Capture Suite. Right: Calibrating with props and tables

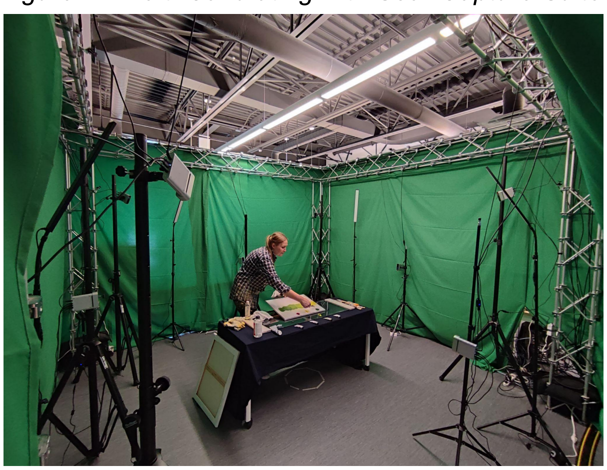


Figure 18: Starting the capture session

Figure 19: Live capture session

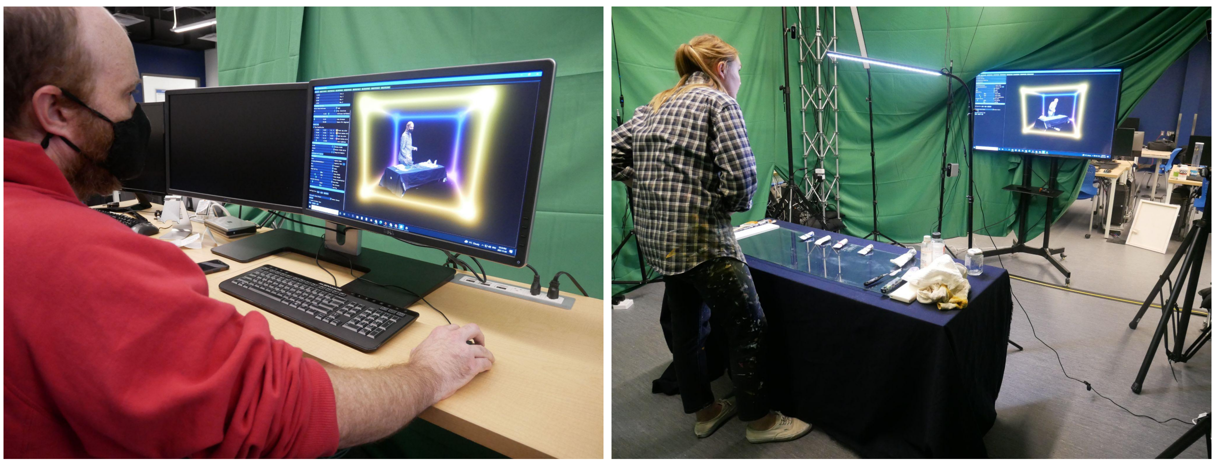


Figure 20: Left: Live view of operator in Soar Capture suite. Right: Live view for performer.

Capture Session 4: Ontario Tech

Capture Setup:

- Multi camera (x6) Azure Kinect
- LiveScan3D (modified)

Description:

We set up a multi-sensor LiveScan3D volumetric capture system and captured a series of painting tutorials by Veronika Szkudlarek. This series was captured at Ontario Tech's Game Development and Interactive Media Lab.

Insights

- The calibration process was relatively simple, however due to the fact that we had a very large calibration target it was a challenge to move it in/out of the recording area without disturbing the cameras in the optimal position. Thus, we had to move the cameras to a sub-optimal location to allow for this capability.
- The capture process was easy as our version of LiveScan3D allows for one-button click recording of raw data or point cloud sequences.

Challenges

- Audio recording is not supported by LiveScan3D at this time. Thus it must be recorded separately and synchronized with other tools afterwards.
- Post processing generally is difficult as the tools do not yet exist in a mature state for this type of data. We developed VolNodes to aid this and are able to process this data yet this is an ongoing task due to development timelines.
- The lack of texture support makes the result of lower quality yet this can be rectified and the data re-processed as improvements to VolNodes are made in the future.

Links

- Images and Videos from session:
    - Ontario Tech Session: LiveScan3D: Images & Videos
- Recordings and Data:
    - Ontario Tech Session: LiveScan3D: Recordings

Images from Session:

Figure 21: Calibrating LiveScan3D

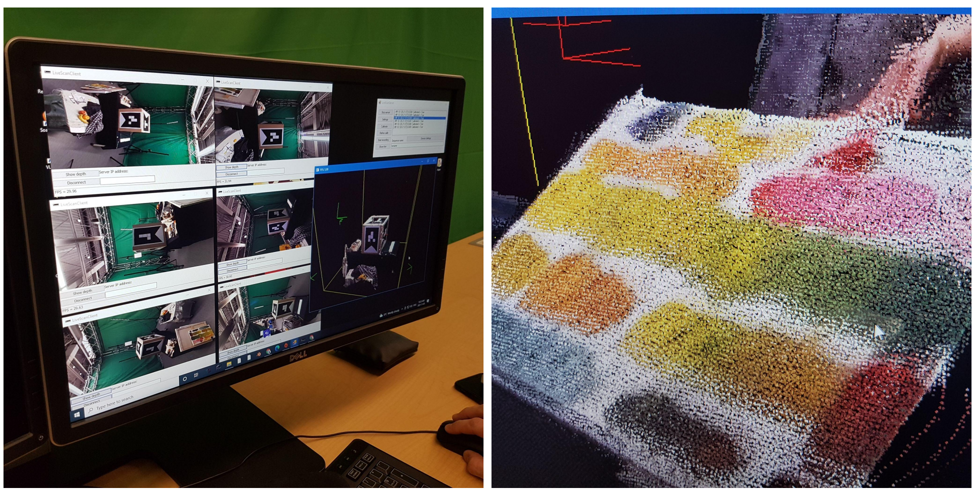


Figure 22: LiveScan3D view of calibration process and verification

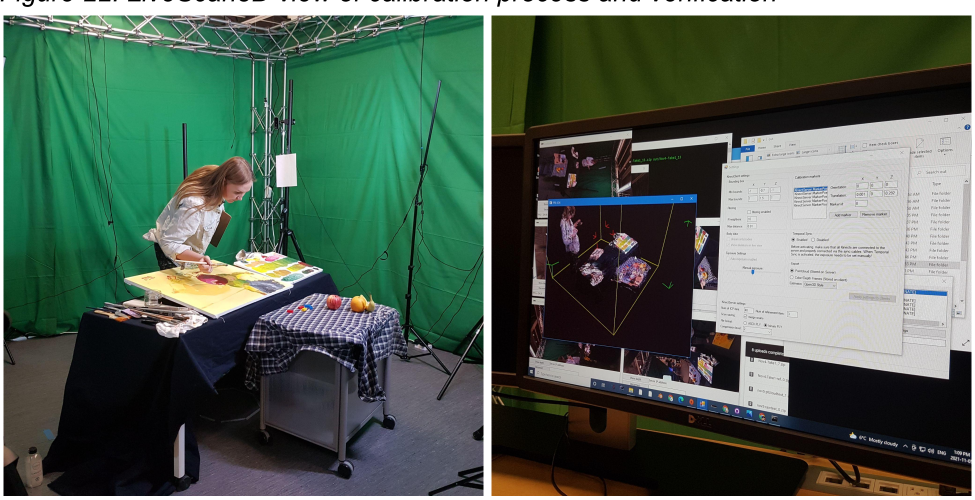


Figure 23: Left: recording session in progress. Right: live preview of point cloud

Figure 24: Live preview of point cloud from LiveScan3D

Conclusions

We hope this document serves as a guide to start on your volumetric video journey. Please connect with us on the Volumetric Discord Community and share your experiences and thoughts. Our community is strong and open to helping others with volumetric video production. We hope the volumetric video recordings and data we captured are helpful for other educators.

Please join the Volumetric discord community and join in the conversation on how to utilize Volumetric video in the Studio Arts!