Free3D - Free-viewpoint 3D video creation

Felix Brunn, Jan Christmeier, Nick Philipp Häcker, Laura Christin Stempfle, Lukas Willmann, Simon Zakowski, Uwe Hahne
Hochschule Furtwangen University

Summary

- Capturing dynamic 3D scenes using highly reduced equipment (only three Azure Kinect cameras) and processing time
- Keeping high visual quality by using the depth data from the cameras
- Combining the captured dynamic 3D scenes with static Gaussian Splatting scenes to create impressive results
- Highlighting the potential of this method by being more accessible and cost-effective

Main Idea

- Investigating in the ability to generate high-quality dynamic 3D scenes using much less equipment than in other dynamic 3D capturing projects
- Using the depth data of three Azure Kinect cameras to compensate for the much lower input image number
- Further utilizing Neural Radiance Fields (NeRF) or Gaussian Splatting for processing and as visual output for the captured data
- Finally, creating an easy-to-use workflow to build the setup and capture dynamic 3d scenes with much less equipment and lower costs
- The workflow should then be available and more accessible for e.g. online learning

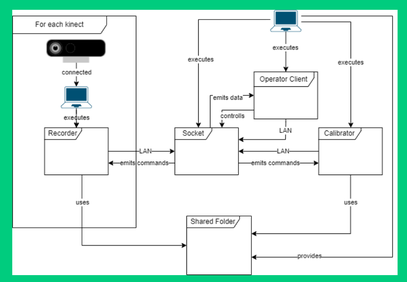
Dataset

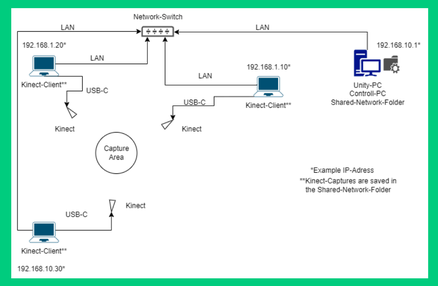
- Self-produced videos in 1920 x 1080 pixel format using three Kinect cameras from three different perspectives
- Videos contain RGBD-data from the camera and the depth-sensor of the kinect

METHOD

Capturing

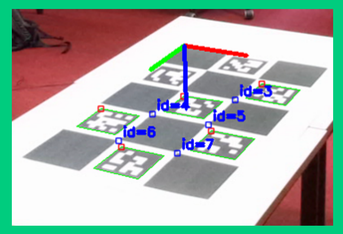
- For processing, one video from each of the three Kinects has to be recorded simultaneously
- For that, the Kinect cameras had to be placed around the desired scene or object
- Each Kinect was connected to a laptop which executes the recording in accordance to the message of the Master with which all Sub-Kinects are connected to via a Web-Socket
- [In our case] Four IP addresses were assigned manually to our computers and one of them shared a folder via Windows Network Share, where every laptop in charge of a Kinect camera saved their recording





Calibrating

- The camera calibration was achieved via marker tracking using a wooden cube with ArUco markers printed on and a ChArUco poster
- The relative positions of the cameras were determined by an algorithm comparing pairs of the detected marker positions which were calculated with OpenCV
- The point clouds could be assembled using the calculated camera positions

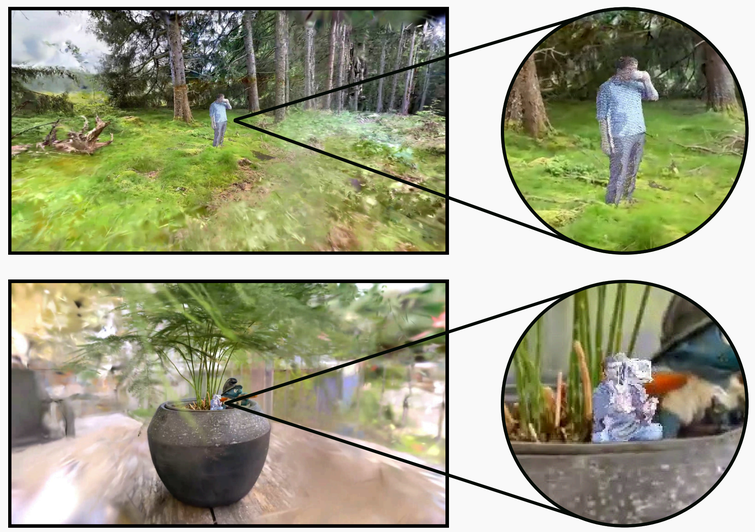


Post-Processing

- RGB-D data from recordings and the calibration data are processed to form a point cloud. Utilizing Open3D, we can isolate the desired object from the background
- With “DB-Scan” and “Plane-Selection” algorithms, the objects can be cropped from the scene

Results

The output was a “point cloud video” combined with a static Gaussian Splatting scene in Unity



- Reduced time consumption:
  - Building the setup: 7 minutes
  - Complete setup ready to capture: 90 minutes
  - Video processing: 6 minutes
- The ArUco cube provides more accurate calibration than the ChArUco poster
- The post-processing method proved wholly unreliable, as it led to errors where parts of the central object disappeared or the center was removed entirely. To circumvent this problem, the “Depth_trunc” of Open3D for generating the point clouds from depth images was often changed.