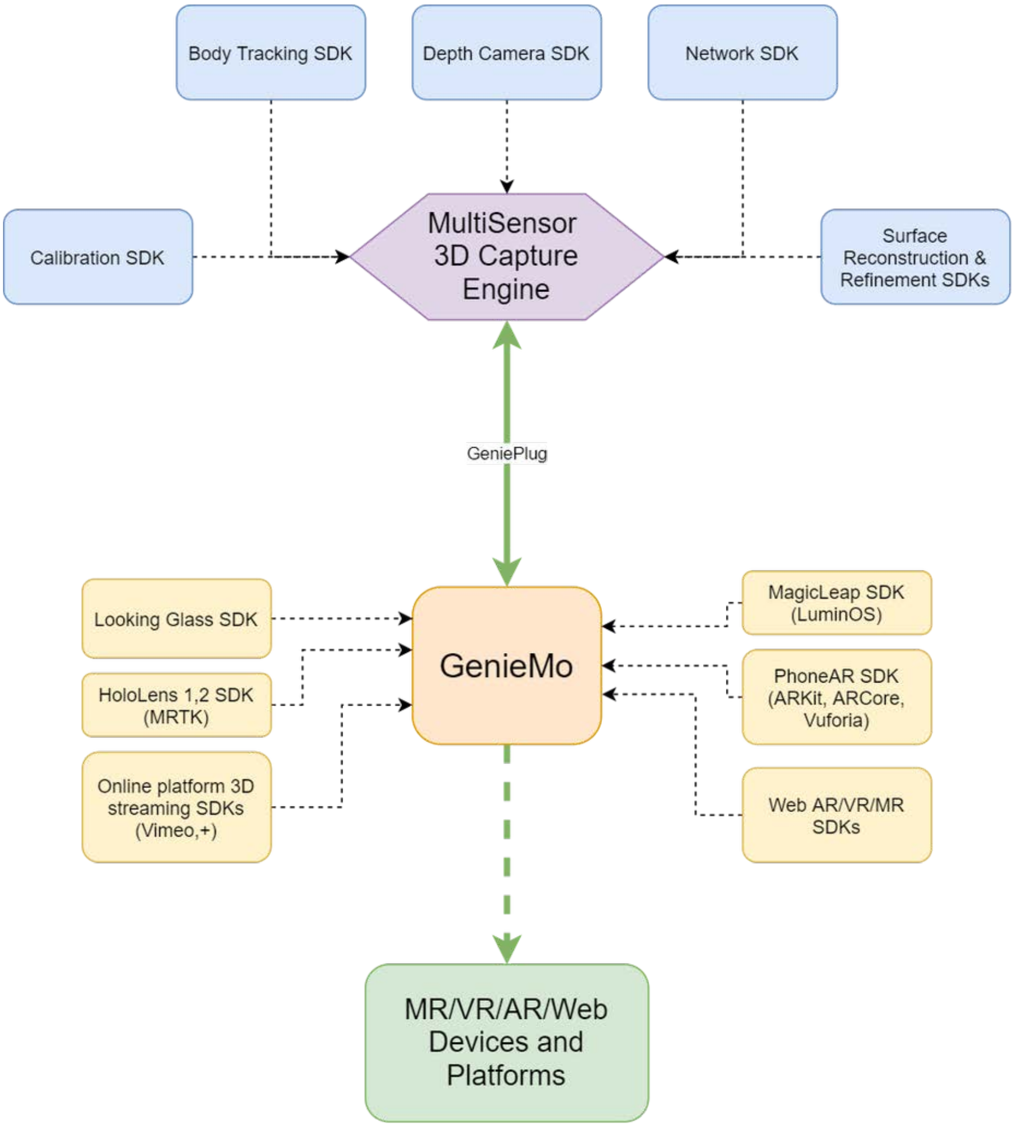
PlayLabZ

GenieMo

USER GUIDE
(0.1 ALPHA)

What is GenieMo?

GenieMo is a project that proposes to democratise volumetric processes by connecting the necessary open source elements to setup and run a feature rich 3D capture and communication system using affordable off the shelf depth cameras, computers and other display devices.


Body Tracking SDK
Depth Camera SDK
Network SDK
Calibration SDK
MultiSensor 3D Capture Engine
Surface Reconstruction & Refinement SDKs
GeniePlug
Looking Glass SDK
HoloLens 1,2 SDK (MRTK)
Online platform 3D streaming SDKs (Vimeo,+)
GenieMo
MagicLeap SDK (LuminOS)
PhoneAR SDK (ARKit, ARCore, Vuforia)
Web AR/VR/MR SDKs
MR/VR/AR/Web Devices and Platforms

Why GenieMo?

Aware of the current global marketing tendencies, the surreal up rising of surveillance capitalism as well as the geopolitical context of nowadays, PlayLa.bZ adopts new business strategies that have the community at it's core and presents a potentially democratising set of tools that would demystify technological processes that seem out of reach when presented by big corporations or other entities with business plans around it.

The impact virtual experiences have on the human mind are beyond a doubt as intense as reality can be, let alone the power of the narratives transmitted via such media, no alternative to the systems proposed by companies like Facebook can result in a monopolised world of virtual experiences literally mind washing all it's users into the reality wished by such companies. That is limiting the very essence of virtual reality, the freedom of playing around in a boundary free world.

Our aim is to inspire and awaken the artist inside everyone who comes in touch with GenieMo inlcuding all the people that are contributing to it's developments and social presence, bringing the free thinking creative human back to surface in a most organic and natural way.

Multi-sensor Overview

One of the advantages of resolving the use of multiple sensors on the same capture setup is that the same network solution used to link the streams from individual sensors is used to stream the stitched model further to other devices, even if the devices are on different continents. (Hatfield (UK) - San Francisco (US) - Holoportation - 2018)

Using a peer-to-peer model anyone can setup servers and communication pathways without the need of a third party or cloud service.

With Kinect Azure the restraint of single sensor per computer is lifted giving way for more compact and portable solutions with applications in volumetric filmmaking and journalism, interactive installations, communication and in many other art and social branches.

Calibration Overview

When more sensors are used one of the key factors influencing the outcome fidelity is the calibration. Using trackable calibration markers, a 3D reference point in the real space between the sensors can be created and then described back via the relative distance between the markers, seen by GenieMo as planes positioned in 3D space.

Using the 3D reference, the positions of the Kinects are matched to their position in reality, simply rotating and moving their individual captured part of the model in order to match the calibration computations.

The quality of the printed markers and overall support system dictates the quality of the calibration and final outcome.

Hardware Setup

Minimum specs:

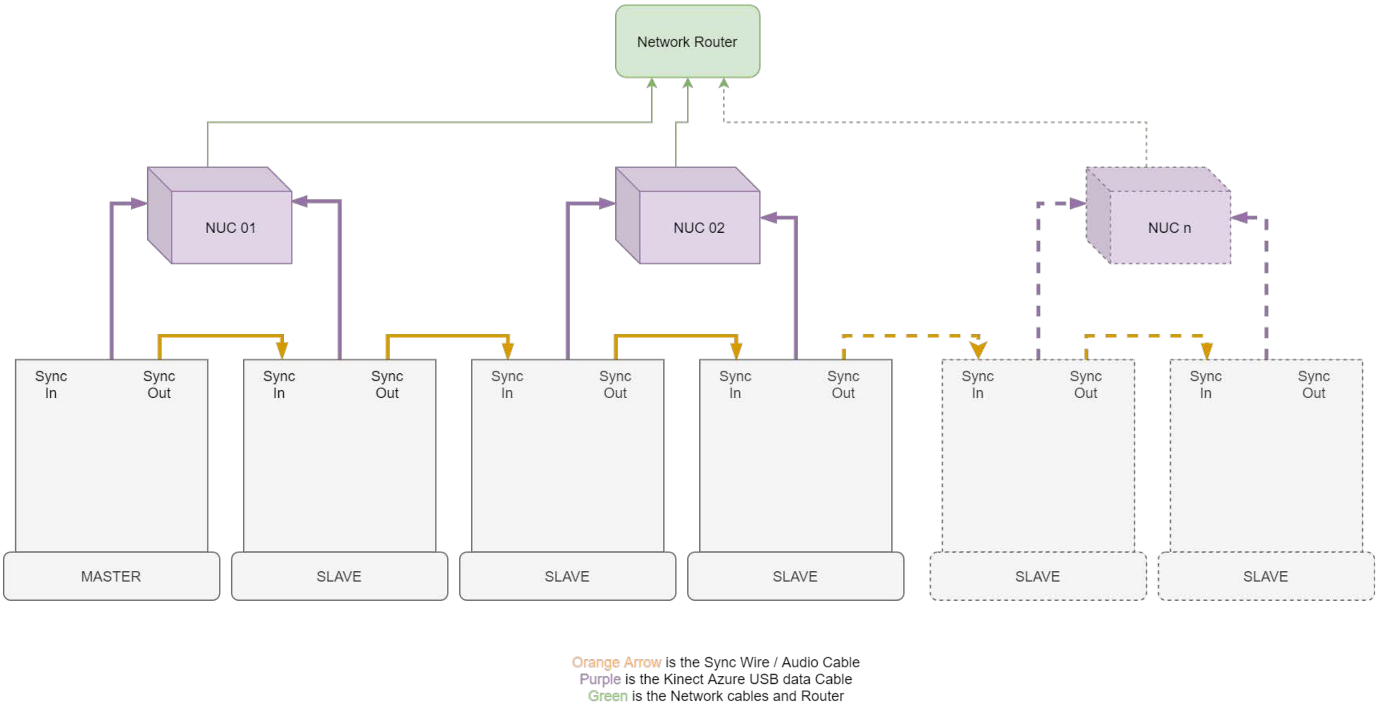
1x Kinect Azure
1x i5 with Nvidia GPU (in some cases a dedicated GPU has proven to work too)
Light setup (getting the scan area as evenly lit as possible)

Portable 2 Sensor setup minimum specs:

2x Kinect Azure
1x i5 NUC with at least 2 USB 3.0 ports
Light setup (getting the scan area as evenly lit as possible)

Max number of sensors Setup

n x Kinect Azure
one computer per 2 sensors
Light setup (getting the scan area as evenly lit as possible)
Network Router (the limit of sensors is given by the network capacity and speed)


Network Router
NUC 01
NUC 02
NUC n
Sync In
Sync Out
MASTER
SLAVE
SLAVE
SLAVE
SLAVE
SLAVE
Orange Arrow is the Sync Wire / Audio Cable
Purple is the Kinect Azure USB data Cable
Green is the Network cables and Router

N Sensor Capture setup overview

Software Setup

GenieMo’s 3D multisensor engine is based on LiveScan3D with all of it’s functionality kept and improved over streaming captured data for multiple devices.

At this stage the software install consists simply of copying the GenieMo/LiveScan3D folder to each computer that will have a sensor attached to it as well as the server one that could be any of the client computer if we need to reduce the number of used computers.

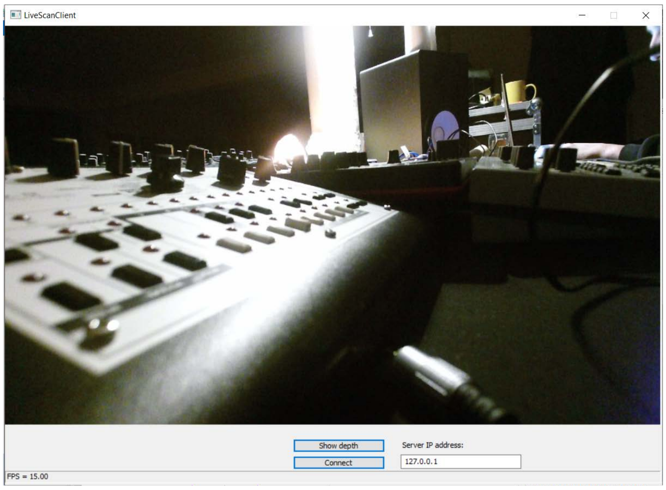
Step by step configuration

Next we describe step by step how to configure and run LiveScan3D with Kinect Azure sensors in a scenario where all Kinects are able to see at least one face of a double sided calibration marker.

In order to start you will need to:

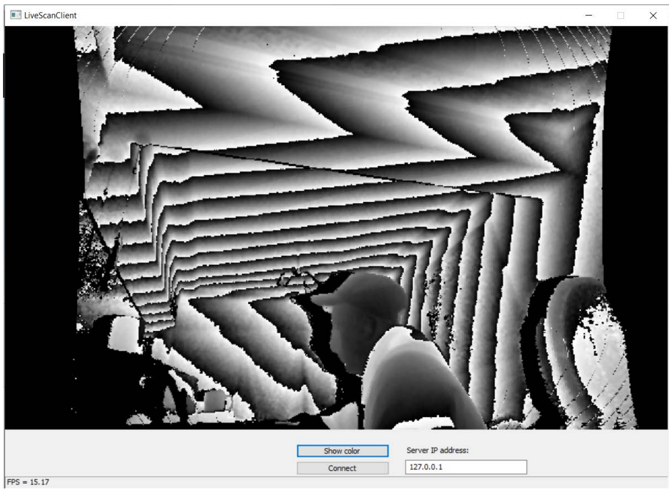
- have at least one Kinect Azure sensor,
- have a copy of GenieMo LiveScan3D or get one from https://github.com/PlayLabz/GenieMo
- download and install the Kinect Azure SDK 1.2.0 on each machine you intend to use as a client,
- have all of the computers you will use in the same network (port 48001 open and free of firewall),
- print the calibration pattern “calibration0.jpg” and “calibration1.jpg” on a piece of paper (A4 size should be enough, we have been using A3).

Once you have all of the preliminary steps completed, run the LiveScanClient application on each of your client computers. If everything is working fine, you should see the RGB camera stream of your Kinect inside the application window.



Server IP address: 127.0.0.1

FPS = 15.00

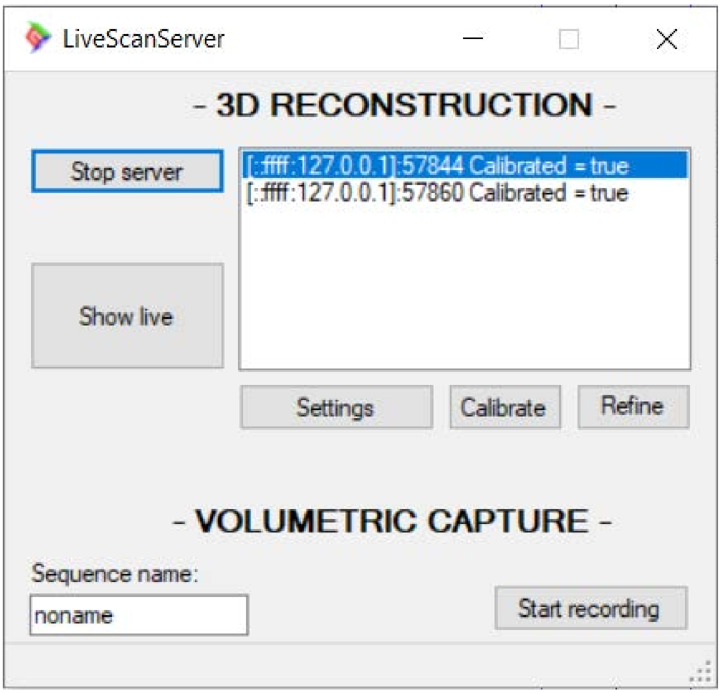


Server IP address: 127.0.0.1

FPS = 15.17

Next, choose a computer that will act as a server (this computer may also be a client at the same time), we recommend that it is the most powerful of the available machines. Run the LiveScanServer on the chosen machine and click "Start server" to begin listening for client connections. Connect each of the clients to the server by pressing the "Connect" button on each LiveScanClient app. If the client is running on the same machine as the server, there is no need for inputting an IP address (it defaults to 127.0.0.1).

At this point your server window should show some clients connected. You can press "show live" to see the output from the sensors.

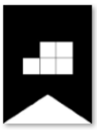


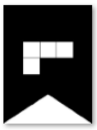
```
[::ffff:127.0.0.1]:57844 Calibrated = true
[::ffff:127.0.0.1]:57860 Calibrated = true
```

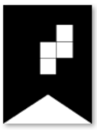
Sequence name: noname

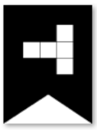
Calibration

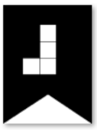
In order to calibrate the clients we will need two printed calibration patterns. The first two or any combination will work as long as the “Marker Id” values (0,1,2,3,4,5) in LiveScanServer Settings window is set accordingly.

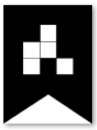

0b.gif


1b.gif

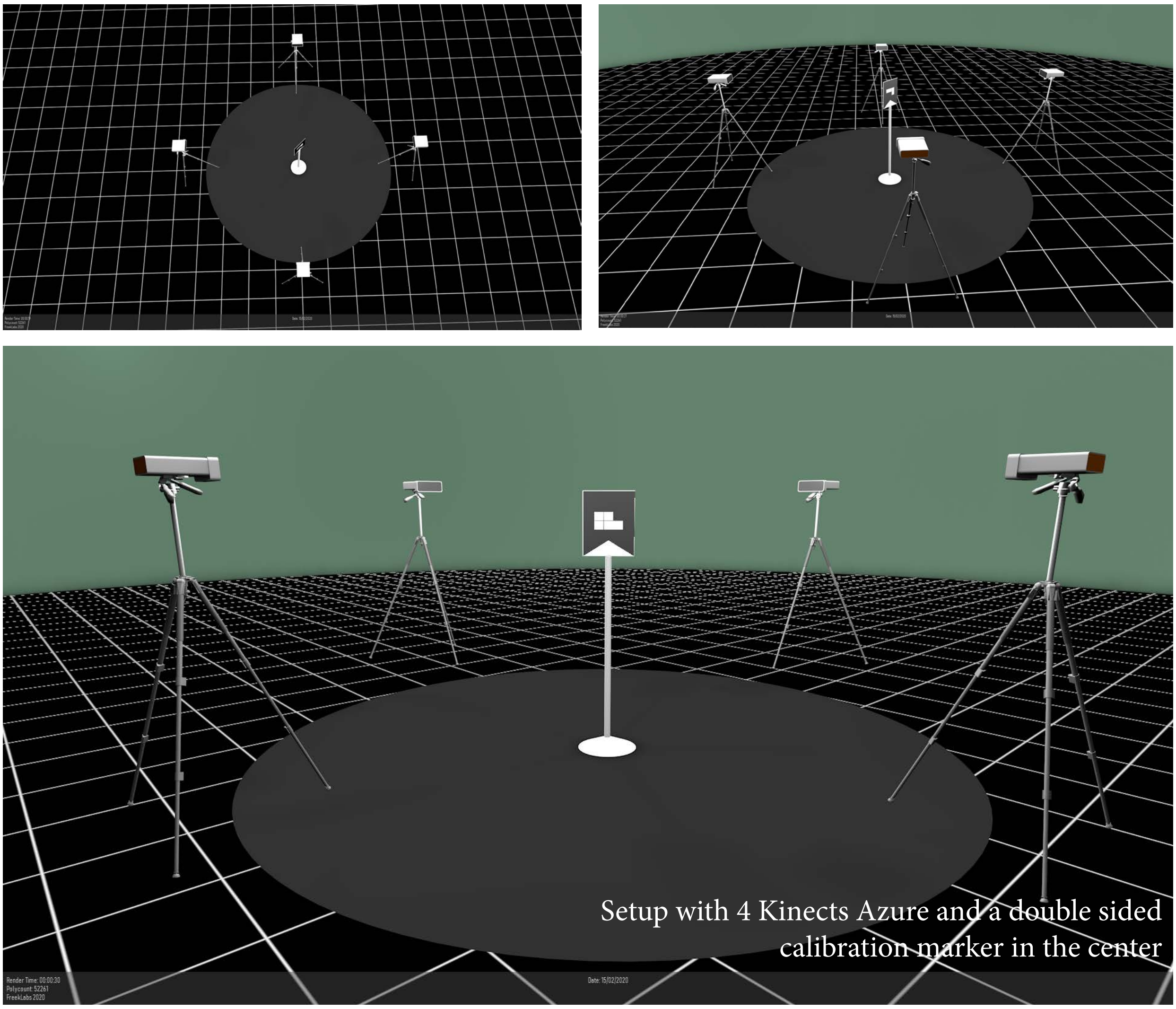

2b.gif


3b.gif


4b.gif

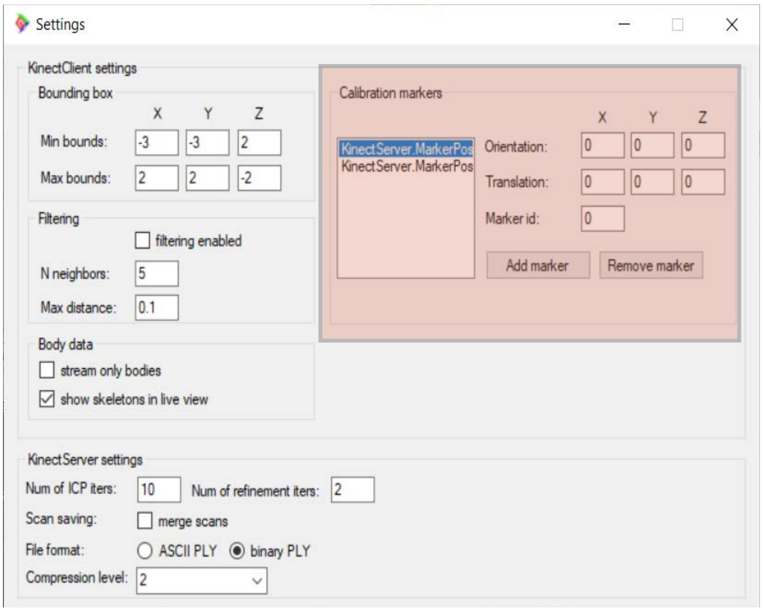

5b.gif

Once printed, no need to cut any white border on the marker, that border is helping the detection process so has to be left there. Attach the markers to a rigid surface on each side and place it in a position where at least one face of the double sided marker is visible to all sensors. Note that the calibration pattern (marker) must be visible to the sensor’s depth and color stream, you can check if it is by pressing “show depth” in the client window.



Setup with 4 Kinects Azure and a double sided calibration marker in the center

Calibration Settings



Settings

KinectClient settings

Bounding box

	X	Y	Z
Min bounds:	-3	-3	2
Max bounds:	2	2	-2

Filtering

□ filtering enabled

N neighbors: 5

Max distance: 0.1

Body data

□ stream only bodies

☑ show skeletons in live view

Calibration markers

KinectServer.MarkerPos
KinectServer.MarkerPos

	X	Y	Z
Orientation:	0	0	0
Translation:	0	0	0

Marker id: 0

KinectServer settings

Num of ICP iters: 10 Num of refinement iters: 2

Scan saving: □ merge scans

File format: ○ ASCII PLY ● binary PLY

Compression level: 2

To make sure that the server knows which marker you want to use,
go to settings and make sure that under “calibration markers” the following are set:

a maker with

“Orientation” set to 0,0,0 (x,y,z)
“Translation” set to 0,0,0(x,y,z)
“Marker id” 0 (or the corresponding value if the marker used is not the first one)

and one with

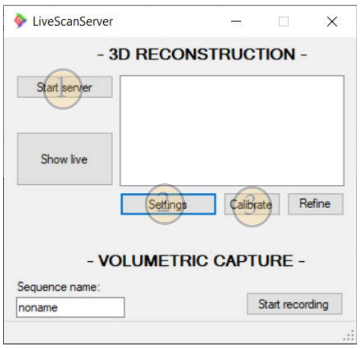
“Orientation” set to 0,180,0 (x,y,z) (we rotated the second marker relative to the first)
“Translation” set to 0,0,0(x,y,z)
“Marker id” 1 (or the corresponding value if the marker used is not the second one)

Now all you have to do is press calibrate and the data from you Kinects should align. You can check if that is the case in the live view window.

LiveScanServer

The LiveScanServer application governs the clients and allows for calibration, filtering, sequence recording, live preview of the reconstruction etc. Below you will find a separate section for each window of LiveScanServer with explanations of each of the program's functionalities.
Main window

The main window of the server application. It shows two clients connected, one of which is calibrated.



Sequence name:
noname

1. Start Server Button

Starts and stops the server – the server needs to be started before any work can be done. If there are any errors when trying to start, make sure that this is the only instance of LiveScanServer running and that there are no apps using port 48001.

2. Settings Button

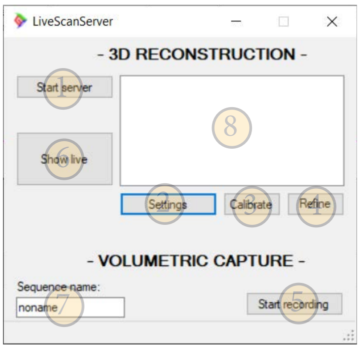
Opens the settings window – more details below.

3. Calibrate button
Performs calibration using markers
– each client performs the calibration separately and once it succeeds that client show "Calibrated = true". In order to perform calibration the clients need to know the locations of the markers in the scene, those locations are set in the settings menu. If a client sees many markers, the largest one is always used.

In some scenes where not all Kinects are able to see the same marker, you will need to use more than one. There are six different marker patterns included in the project files. Each of them corresponds to a unique id of 0 to 5. In order to use a marker with a given id it needs to be added to the set of known markers in the settings window.

If one of the clients does not calibrate, there can be several reasons:

- The marker is not getting detected – this is fairly common if there are bad light conditions. Try and move it around, or switch the light on in the room to see if it helps.
- The marker's pose is not known to the client – this usually manifests by a green border around the marker (which indicates that it is correctly detected), but no calibration. To fix this, simply add the marker with the proper id in the settings menu.



Sequence name: noname

4. Refine Button

Refines the current calibration – if all of the sensors are already calibrated and there still some misalignment between the point clouds, this functionality may help. It uses Iterative Closest Points (ICP) to refine the alignment between the point clouds. This function only works well if there is a fair deal of common surfaces between the point clouds from different sensors. You can use it multiple times for further improvement.

5.Start Recording Button

Performs recording – once this button is pressed the clients begin to capture and locally store frames in a synchronous manner. Once you press this button again, the recording will stop and the clients will begin uploading the frames to the server. The server will save them as PLY files in a directory specified by "sequence name". Pressing the button again before all frames are saved will stop saving and allow the recording of a new sequence.

6. Show Live Button

Opens the live view window – more details below.

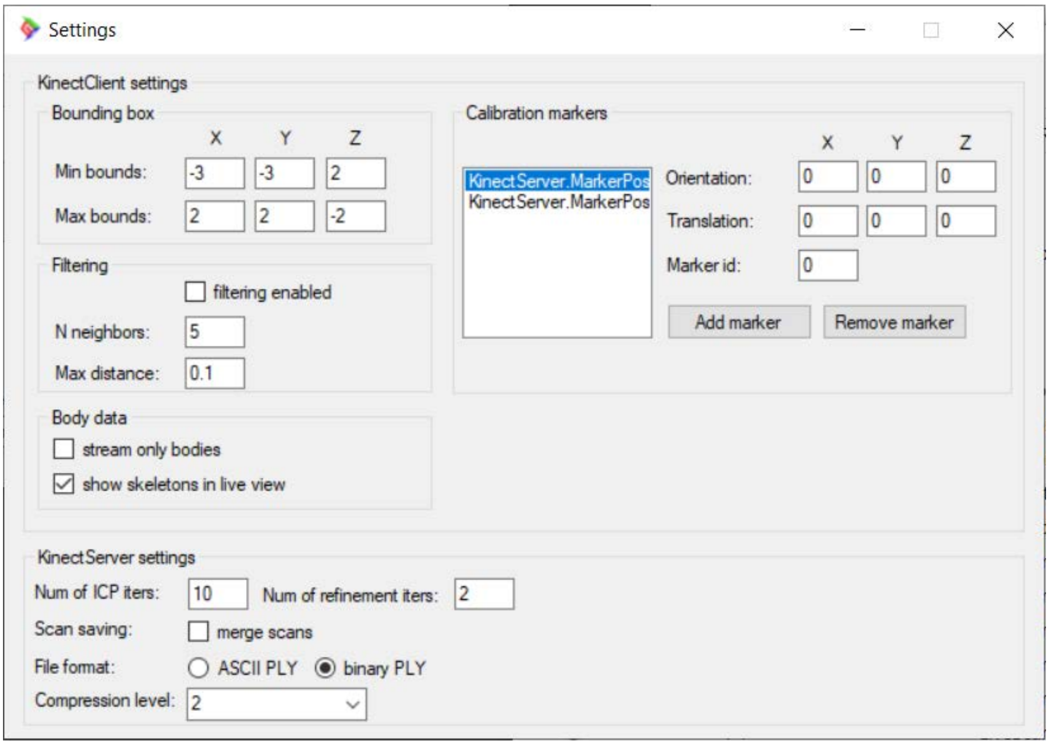
7.Sequence name

this specifies the directory to which the PLY files will be saved if you perform recording. The whole directory path will be "./out/<sequence name>".

8. List of all of the connected clients

it shows the client IP addresses along with the information on whether they are properly calibrated or not.

The Settings Window



KinectClient settings

Bounding box

	X	Y	Z
Min bounds:	-3	-3	2
Max bounds:	2	2	-2

Filtering

□ filtering enabled

N neighbors: 5

Max distance: 0.1

Body data

□ stream only bodies

☑ show skeletons in live view

Calibration markers

KinectServer.MarkerPos
KinectServer.MarkerPos

	X	Y	Z
Orientation:	0	0	0
Translation:	0	0	0

Marker id: 0

KinectServer settings

Num of ICP iters: 10 Num of refinement iters: 2

Scan saving: □ merge scans

File format: ○ ASCII PLY ● binary PLY

Compression level: 2

The most important part of the settings window is the calibration markers panel. Here you can define the positions of the markers in your scene, which will allow you to define the world coordinate system. The marker orientation and translation describe the marker's pose in the world coordinate system (orientation is in degrees, translation in meters). The marker id allows the program to identify individual markers, the marker's id is the number in its file name in /docs/calibration markers/. Using multiple markers is necessary in situations where not all off the sensors can see a single individual marker.

The bounding box panel allows for removing parts of the scene which are not necessary. The points outside of the bounding box are removed in the client application and never reach the server. This is important as it reduces the bandwidth between the clients and the server and thus increases the frame rate in the live view window. All the coordinates are in meters.

The filtering can be applied to remove noise and flying pixels. The filtering will remove any point which has its Nth neighbour further than max distance. In order to filter more points you can either decrease the max distance or increase the number of neighbours. It is important to remember that enabling filtering puts a lot of strain on the client computers and might decrease frame rate.

The KinectServer settings panel controls some of the settings of the server:

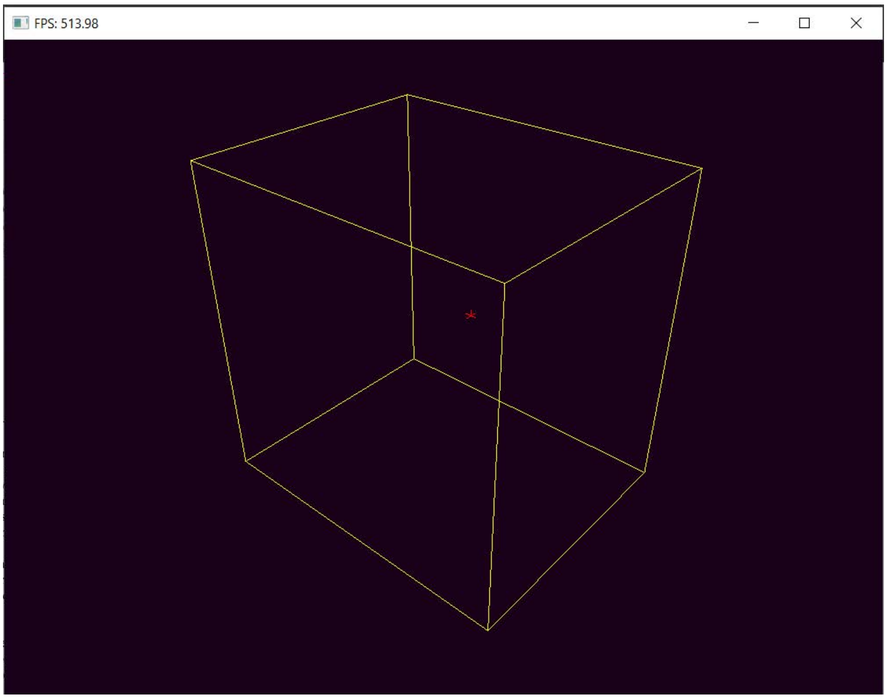
•Num if ICP iters, num of refinement iters both control calibration refinement, for details refer to [1].

•Scan merging controls whether the point clouds from separate clients are saved into a single, merged file or to separate files (this occurs after recording).

•File format controls the type of PLY that will be saved. The binary PLY format is much more compact and can be opened using the LiveScanPlayer.

•Compression level controls how much compression is applied to the data sent from the clients to the server. For details on the meaning of the levels refer to ZSTD's documentation.

The Live Window



This window shows the reconstructed point cloud from all of the sensors. Apart from the points themselves you will also find the marker positions in the scene as red lines and sensor positions as green lines. The frame rate here depends greatly on the speed of you network and on the number of devices you are using. In the future (hopefully near future), we plan to work on reducing the size of the point clouds, which should increase the FPS.
Remember than even when there are only a few frames per second in this window, the frame rate should be high if you record the frames (using the "Start recording" button in LiveScanServer).

There is a number of ways to move around the reconstructed point cloud in this window:

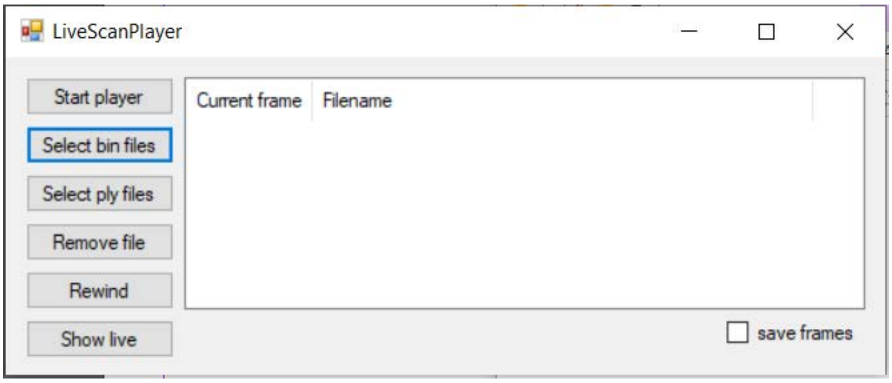
- Left mouse button – lets you rotate the point cloud,
- Right mouse button – lets you move the camera,
- Mouse wheel – zoom in/zoom out,
- W, S, A, D keys – let you move the camera around like in an FPS game,

There are also additional functionalities under the following keys:

- +, - keys – change the size of the points.
- F key – toggle full screen mode
- M key – toggle marker and bounding box visibility

LiveScanPlayer

This application allows for viewing and transmitting the pre-recorded sequences, which were saved by the clients (.bin format) or the server (.ply format). It also allows for saving data from the .bin files in the .ply format.



- Start player – starts or stops the player and the Transfer Server. The Transfer Server allows for streaming the played data to a remote device, for example the HoloLens using our GenieMo for HoloLens application. If save frames is selected during playback, the played frames will be saved as binary PLY in the ./outPlayer/ directory.
- Select bin files – allows for opening the bin files created by the clients during recording. Each bin file contains an entire sequence of frames.
- Select ply files – allows for open binary PLY files saved using the server. It is not guaranteed that it will allow opening PLY files saved by other software. Each PLY file contains a single frame so you need to select multiple files to have a sequence.
- Remove file – removes the selected sequence from the list. Any file can be selected by clicking on the current frame counter.
- Rewind – sets the current frame of all sequences to 0.
- Show live – shows the live view window, which works identically to the same window in the server.
- Changing the value of the current frame – makes the chosen sequence skip to that frame.

Bibliography

[1] Kowalski, M.; Naruniec, J.; Daniluk, M.: "LiveScan3D: A Fast and Inexpensive 3D Data Acquisition System for Multiple Kinect v2 Sensors". in 3D Vision (3DV), 2015 International Conference on, Lyon, France, 2015