BREKEL

POINTCLOUD

V3

WHAT IS BREKEL POINTCLOUD V3

Brekel PointCloud v3 is a Windows application for recording volumetric video using one or multiple depth sensors from various brands and types.

Multiple sensors on the same machine (depending on the sensors) and/or across multiple networked machines are supported.

It is written by Jasper Brekelmans, so in case you're wondering that's what "Brekel" refers to.

"Brekel" is pronounced as "Break–uhl".

SUPPORTED SENSORS

- Kinect for Xbox 360 / Kinect for Windows v1 (with USB2/power adapter)
    - Using the official Microsoft Kinect v1.8 drivers
    - Multiple sensors per machine are supported
    - If your machine has dedicated USB2 ports (black) it's best to prefer these

- Kinect for Xbox One / Kinect for Windows v2 (with USB3/power adapter)
    - Using the official Microsoft Kinect v2 drivers
    - Due to driver/SDK restrictions only a single sensor per machine is supported
    - Multiple networked machines each with 1 sensor are supported
    - Only USB3 ports (blue) with Intel or Renesas chipsets are supported by the drivers

- Azure Kinect
    - Dedicated USB3 port required

- Orbbec Astra
    - All Astra variants (Pro, non-Pro, S, Mini) are supported
    - Astra Persee is not supported
    - Astra Femto / Astra+ not supported (yet)
    - If your machine has dedicated USB2 ports (black) it's best to prefer these

- Intel RealSense
    - 3xx, 4xx and 5xx series sensors should work

SENSOR DRIVER INSTALLATION

Some sensors need drivers in order to work.

The Brekel app will try to automatically detect if drivers are installed on your system already, if not there should be a “download & install” button next to the sensor type in the “Connect” panel on the bottom left.

The app will always download the latest and official drivers from the manufacturer’s website, if that link is down it will fall back to a mirror.

You can also use the options under “Settings” on the top menu to download & install the latest sensor drivers.

Note that you may need to restart your computer for some sensors to be detected after installing drivers.

Note that if you already have 3rd party or non-official drivers installed you may need to uninstall those yourself first to avoid driver conflicts.

USB CONNECTIVITY & BANDWIDTH

Depth sensors are high bandwidth devices that can easily use all the available bandwidth on your USB chipset.

A few tips for best results:

- Many newer sensors (Azure Kinect, RealSense) need a USB3 port to function.
- If your machine has dedicated USB2 (black) ports it's best to prefer those for USB2 sensors like (Kinect v1 & Orbbec Astra), they should however work on USB3 (blue) ports in most cases as well.
- Most desktop/workstation machines have up to 2 internal USB controller, when having problems switching a sensor to a different port on the front/side/back of your machine may help balancing the load to a different controller
- In many cases laptop or mini computers (like Intel NUC) may only have a single USB controller limiting the amount of bandwidth they have.
- You may need to purchase additional PCI USB controllers to expand your (desktop/workstation) machine, the StarTech PEXUSB3S44V is an excellent card with 4 dedicated high bandwidth ports that I can recommend.
- Typical symptoms of reaching your machine's bandwidth include things like connection issues, dropped frames, sensors freezing datastreams.

MINIMUM SYSTEM REQUIREMENTS

It is hard to state minimum required system specifications as this is highly dependent on sensor types and the number of sensors used.

- 64-bit Windows 10 (for older sensors Windows 8/8.1 may work but this is untested)
- 8 GB or more RAM
- Modern Intel i5 or faster CPU (or AMD equivalent)
    - dual core 2.4GHz or i3 may be enough for older sensor types with single sensor setups
    - more cores help for multi sensors setups as the app is multi-threaded
    - 3 Ghz or faster will help for newer sensors and multi sensor setups
- Modern GPU with OpenGL support
    - Microsoft's Azure Kinect body tracking runs on Nvidia CUDA, requiring a GeForce GTX 1070 or better GPU (and currently uses around 150 GFLOPS & 1.9 GB of VRAM per instance)
- 1280×1024 screen (recommended: 1920×1080 or higher)

NETWORK SENSORS

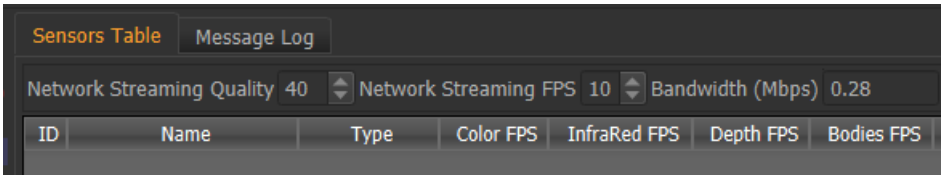
Besides utilizing the sensors directly connected to your machine it's possible to use data from sensors connected to other machines in your network.

Make sure the Brekel app is installed on all machines and the sensors work when running the software locally in GUI mode. (You can use the same license on each machine)

- Pick one machine (the fastest) to run the Brekel GUI application.
- On the other machines start the Brekel app in headless mode (there should be a dedicated shortcut in your Windows Start Menu and on your desktop).
- In the GUI select "Network" sensors and hit "Connect Sensors".
- A broadcast signal will be sent out on your network (UDP port 9875).
- The apps running in headless mode will pick this up and reply back by reporting with details on what sensors are connected to them.
- The GUI and headless apps will then communicate further over TCP port 9876

Note that this should work over a wireless connection but in most practical use cases you'll want a wired connection for maximum reliability and speed.

In the GUI you can use the Network Streaming Quality and FPS settings that will appear above the sensors table to adjust bandwidth vs quality, note that this only affects quality of the points/video in the 3D/2D viewports, body tracking data (as well as BPC pointcloud recording) is always at the highest quality.



NETWORK TROUBLESHOOTING

There can be several reasons why network communication does not work, the most common ones:

- Make sure all your machines are connected to the same network
- Security software and/or firewalls blocking communication
    - make sure UDP port 9875 and TCP port 9876 are not blocked
    - Make sure the apps are not blocked
- Multiple network cards (see below)

The app should automatically try to find the most suitable network adapter (in case there are multiple in your system) and it should prefer ethernet over Wi-Fi.

If auto detection fails, you can manually specify which network adapter to use using Settings > Select Preferred Network Adapter from the top menu.

Please consult with your system/network admin to resolve any of these things.

HEADLESS/CONSOLE MODE

The headless mode exists to run on remote computers sending their sensor data over the network to the main computer running the GUI version. It will consume less CPU/GPU resources as it won't need to draw visualizations on the screen.

You'll find a dedicated shortcut in the Windows start menu and your desktop to start it.

To run the software in headless mode the "-headless" command line option is all that is needed.

By default, it will try to find all supported sensors, by adding the following optional command line options you can control which connected sensors will be used.

-kinect1	find and use Kinect v1 / Kinect for Xbox 360 sensors
-!kinect1	don't use Kinect v1 / Kinect for Xbox 360 sensors
-kinect2	find and use Kinect v2 / Kinect for Xbox One sensors
-!kinect2	don't use Kinect v2 / Kinect for Xbox One sensors
-orbbec	find and use Orbbec Astra sensors
-!orbbec	don't use Orbbec Astra sensors
-azurekinect	find any Azure Kinect sensors
-!azurekinect	don't use Azure Kinect sensors
- realsense	find any Intel RealSense sensors
-! realsense	don't use Intel RealSense sensors

QUICK START GUIDE

To get started:

- Select which sensor(s) you have on your system in the “Connect” tab on the bottom left
- Hit the “Connect Sensors” button and wait for your sensors to show up in the “Sensors Table”
- Sensors will need to be aligned first, use the features on the “Align Sensors” panel at the top right
    - See the mouse-over tooltips and chapters in this manual regarding sensor alignment
- Set the output folder/filename in the Recording panel at the bottom right
- Hit “Start Recording” and “Stop Recording”
- Switch to the “Timeline (File)” tab at the bottom and load the recorded .BPC files
- Set your preferred “Output” file types in the Geometry/Particles/Textures panels on the right
- Hit “Export” on the bottom right tab
- Load the files in your favorite 3D app

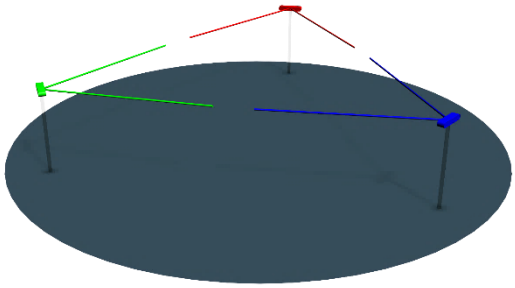
SENSOR PLACEMENT

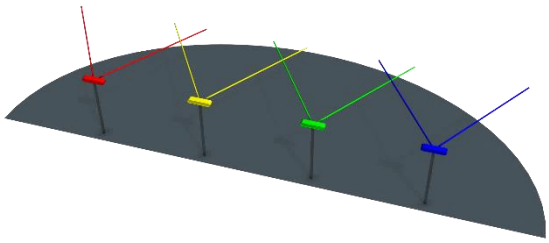
The only requirement the software has on sensor placement are:

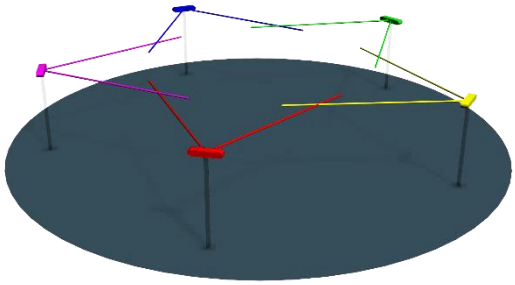
- Multiple sensors need to have some overlap for calibration and data fusion

Some guidelines for sensor placement:

- Between hip and head is usually a good height for a sensor
- It is recommended that sensors can see part of the floor
- Placing your sensors in a circle, facing inwards, allows for 360-degree tracking
- Placing sensors on a line, facing the same direction and with some overlap allows for expanding the tracking area
- When adding more sensors try to space them evenly
- With enough sensors it may be beneficial to place some of them lower and others higher







KINECT V1 (XBOX 360) SPECIFIC

Kinect v1 sensors use USB2 speeds, they can be connected to USB2 or USB3 ports.

Since they use relatively low bandwidth you should be able to use multiple sensors on pretty much all machines.

There may be some interference, in the form of added pointcloud noise, in areas where multiple sensors overlap.

KINECT V2 (XBOX ONE) SPECIFIC

Kinect v2 sensors need to be connected to a USB3 port on your machine.

Due to driver/SDK restrictions only a single sensor per machine is supported, you can of course use multiple networked machines to use setups with multiple Kinect v2 sensors.

There may be some interference at times, in the form of fluctuating distance readings, in areas where multiple sensors overlap. This is caused by these sensors using Time Of Flight and not having hardware sync options.

AZURE KINECT SPECIFICS

Note that you will need an USB3 port for your Azure Kinect.

When running multiple sensors make sure your machine can handle the needed USB bandwidth, laptops may have limited bandwidth, desktop machines can be extended with a card like StarTech PEXUSB3S44V for example.

HARDWARE SYNC

Azure Kinect sensors have sync in/out ports on the back (remove the cover first) that allow multiple sensors to be synchronized. This has the benefit or reducing interference to a minimum so is generally a good idea to use.

All you need to do is run sync cables between devices as indicated here:

https://docs.microsoft.com/en-us/azure/Kinect-dk/multi-camera-sync

(note that the ports are located underneath the cover, remove the screws in the back to take it off)

The Brekel app will automatically identify which sensors are connected as master and which as subordinate (based on if there are connections to the sync in/out ports) and set things up accordingly when accessing the sensors. It will report what it detected in the message log after connecting the sensors.

Note that the body solver was designed to work both with synchronized and non-synchronized data.

When running a single sensor or multiple sensors without sync cables the app will use the Infrared video stream for fastest computation speed.

When running multiple sensors with sync cables the app will use the Color video stream as this is currently required for drivers to use synchronization, this may be a little slower as a bit more computation is needed internally.

ORBBEC ASTRA SPECIFIC

The Orbbec Astra Pro models are currently supported, these are USB2 sensors with comparable specs to Kinect v1 (Xbox 360) sensors.

They can be connected to USB2 or USB3 ports.

There may be some interference, in the form of added pointcloud noise, in areas where multiple sensors overlap.

Orbbec Embedded S sensors should work but given their form factor are not recommended.

Orbbec Stereo S sensors are not recommended, since they rely on stereo sensing and are rather and their form factor is designed for embedded use.

INTEL REALSENSE SPECIFIC

Currently the following Intel RealSense sensors are supported:

SR305 – structured light, short range

D415 – stereo, long range

D435 – stereo, long range

D455 – stereo, long range

L515 – LIDAR, medium range

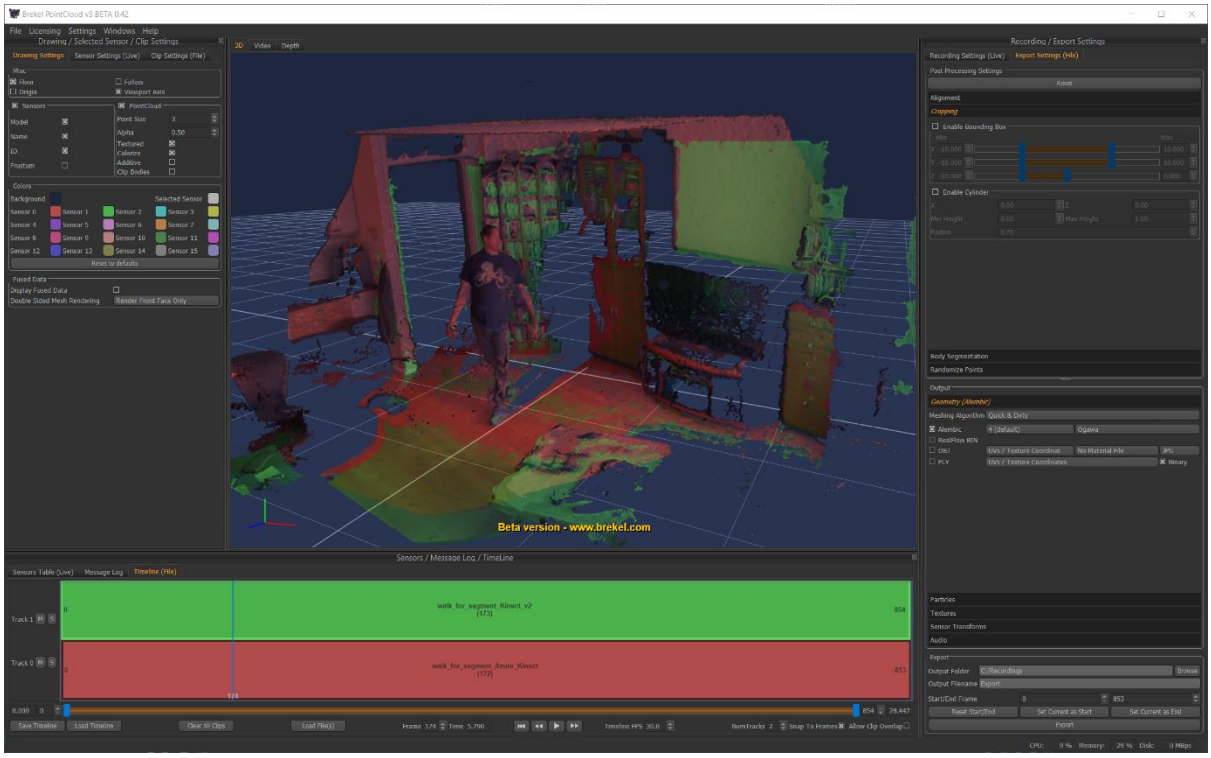
All sensors work best with USB3 speeds/ports but can work (with restrictions) on USB2 ports.

STEREOLABS ZED SPECIFIC

StereoLabs ZED 1, ZED Mini and ZED sensors are supported.

A USB3 port is needed on your machine.

INTERFACE



Left: Drawing Settings
Middle: Main 3D Viewport / Video / Depth
Right: Settings
Bottom: Sensors Table, Message Log & Timeline

- The docks on the left, right and bottom can be undocked and turned into a floating window by dragging from their title bar
- Double clicking on the title bar of a floating window will dock it back into the main window
- You can toggle windows on/off under “Windows” in the top menu
- Right Clicking on the title bar of the main window (next to Help) will pop up a list that allows showing hidden windows again
- The overall GUI state will automatically be saved on exit and reloaded on start
- Most buttons and widgets have tooltips if you hover your mouse over them for a few seconds

Most of the data is visualized in the main 3D viewport, the Video & Depth windows show the raw sensor streams and can be of great help to help understand which part of your volume is seen by each sensor especially when setting things up.

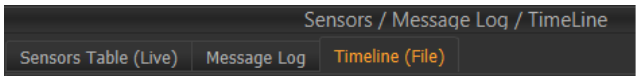
LIVE VS FILE MODE

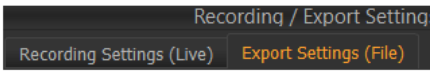
The software operates in two different modes:

Live mode, where it connects to sensors, visualizes their data and records them to disk in the BPC (Brekel Point Cloud) format, designed for efficiency in both recording speed and file size.

File mode, where it loads recorded BPC files on to a timeline applying filters, operations like cropping and green screen for example to enhance the data and fusing/exporting the data to a multitude of file formats for use in your favorite 3D application.

To switch between these mode simply select the Sensors Table (Live) / Timeline (File) on the bottom panel or the Recording Settings (Live) / Export Settings (File) on the right panel.





3D VIEWPORT NAVIGATION

TO ORBIT

- Left mouse button, click & drag

TO DOLLY IN/OUT

- Middle mouse button, click & drag
- Shift + Left mouse button, click & drag
- Mouse wheel

TO PAN

- Right mouse button, click & drag
- CTRL + Left mouse button, click & drag

TOP MENU

FILE MENU

Check for new version on startup

If enabled the application will automatically checks if a newer version is available for download from the website at startup.

Exit

Closes the application

LICENSING MENU

Show License Info

Shows a window with your licensing information

Install License (Automatic)

Will ask you to browse for your license file (the zip attached to the license email you received within 24 hours after purchase). The app will try to automatically install the license by extracting and copying the .key file to your “ProgramData” folder.

Install License (Manually)

In case automatic license installation (see above) fails you can use this to open the folder where the license file needs to be put. Extract the zip file (attached to the license email you received within 24 hours after purchase) to this folder so the .key files is present.

Remove License File

Remove license file from this machine so you can install it on another machine

SETTINGS MENU

Save Settings Now

Saves all the current GUI settings in the registry immediately (also done automatically on application exit and periodically when changing settings)

Show Advanced Options

Shows/hides additional advanced options in the GUI, for example to tune the skeleton solver. (not needed for most users)

Enable High DPI Scaling

Adjust the GUI for high DPI screens, you need to restart the application for this to take effect.

Use Native File Dialogs

Switches which type of file/folder dialogs to use, Windows native or Qt.

Warn for File Overwriting

When enabled shows a popup message warning if files will be overwritten when starting a new recording.

Select Preferred Network Adapter

Allows you to overrule which network adapter the software should use (just in case the automatic selection that is used by default doesn't work for you)

Print Network Interface Information

Prints details of your network interfaces to the message log, please provide these details if you are contacting support regarding network issues.

Install Drivers <sensor type>

Manually downloads & installs drivers for the supported sensor types.

WINDOWS MENU

Toggles visibility of the various windows like settings and such

HELP MENU

Downloads Page

Opens up the downloads web page in your default browser.

Brekel Forum

Opens the forum web page (hosted on Google groups) in your default browser.

PDF Documentation

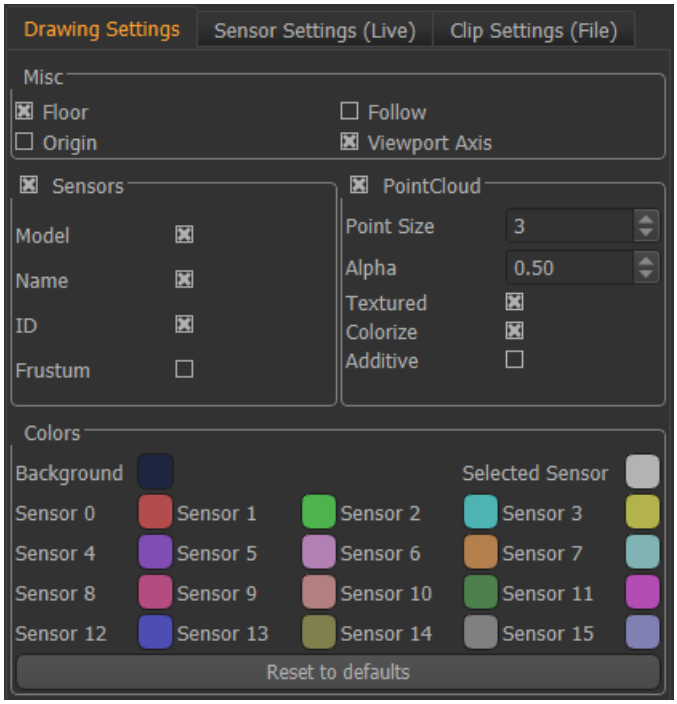
Opens this manual in your default PDF browser.

KEYBOARD SHORTCUTS

The following keyboard shortcuts are available:

CTRL + Space:	start/stop recording
CTRL + 1:	toggle display sensors
CTRL + 2:	toggle display pointcloud
CTRL + 3:	toggle display raw bodies
CTRL + 4:	toggle display solved bodies
CTRL + 5:	switch main viewport to 3D view
CTRL + 6:	switch main viewport to video view
CTRL + 7:	switch main viewport to depth view
F1	switch to drawing settings tab
F2	switch to selected sensor settings tab
F5	switch to sensors table tab
F6	switch to message log

DRAWING SETTINGS



MISC

Floor	draw the floor plane/grid in the 3D viewport
Origin	draw an axis at the world's origin (0,0,0) in the 3D viewport
Viewport axis	draw an axis in the bottom left corner of the 3D viewport
Follow	toggles if the camera should keep all tracked bodies in the view by following them

SENSORS

Model	draw sensor 3D models in the 3D viewport
Name	draw sensor names in the 3D viewport
ID	draw sensor IDs in the 3D viewport
Frustum	draw approximations to the sensor lenses in the 3D viewport

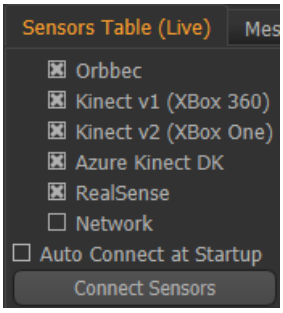
POINTCLOUD

Point Size	size of the points (in pixels) of the pointcloud in the 3D viewport
Alpha	transparency of the pointcloud in the 3D viewport
Textured	apply texture or display solid color
Colorize	colorize pointclouds by their sensor color in the 3D viewport
Additive	use additive shading for pointclouds in the 3D viewport

COLORS

Background	color of the background in the 3D viewport
Selected Sensor	color of the currently selected sensor from the “Sensors Table” in the 3D viewport
Sensor 0 -15	color for all the individual sensors when they’re not selected in the 3D viewport

CONNECT



On the “Connect” tab on the bottom left of the GUI you can select which sensors to use.

If a sensor needs a driver and it’s not found on your system a little “download & install” icon will appear next to it if needed.

The various sensors types depict sensors connected to your current machine.

The “Network” sensor depicts sensors connected to other machines on your network, to use these you will need to run the application in “Headless Mode” on that machine, see the chapter about that.

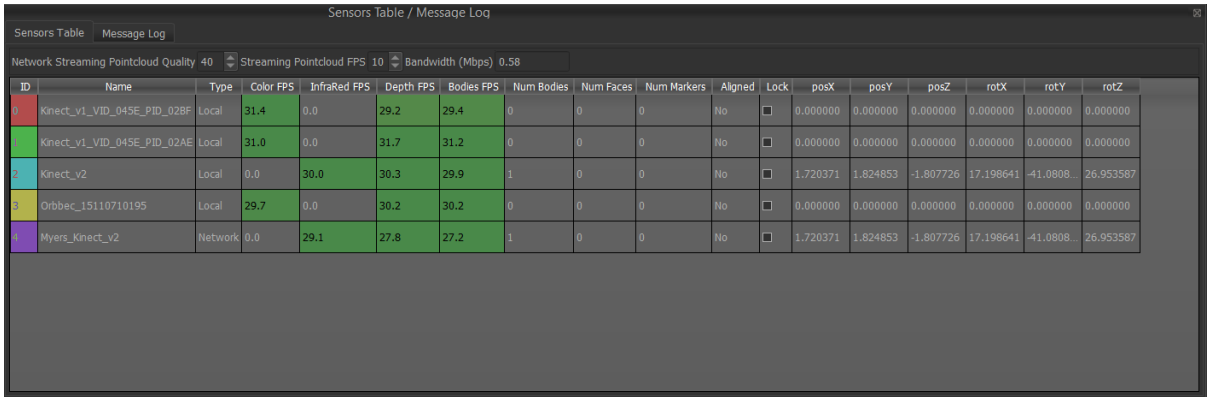
Auto Connect at Startup will automatically hit the “Connect Sensors” when the program has started, which can be handy if you always use the same setup.

“Connect Sensors” will find all the sensors (of the types you selected) it can connect to and start streaming data from them into the application.

After connecting sensors, the button will chance into “Disconnect Sensors” which will allow you to stop streaming data and disconnect.

The software will internally split of each sensor to a different system thread to utilize today’s multicore machines. On the same token some processes like the skeleton solver will also run in different threads.

SENSOR TABLE



On the bottom of the GUI you'll find the "Sensor Table" which shows all connected sensors.

You can select a sensor by clicking somewhere on it in the table with the left mouse button, simply click again to unselect.

You can change some settings of the selected sensor in the tab "Selected Sensor Settings" located in the top left tab (next to Drawing Settings). The specific settings vary depending on the type of sensor.

For each sensor a number of things will be displayed in the table.

ID	The number of the sensor in the list
Name	The name of the sensor (can be changed in the “Selected Sensor Settings” tab
Type	Local (connected to this machine) Network (connected to a network machine)
Color/Infrared/Depth/Bodies FPS	The actual framerate of the data that is streamed.

Note that if your machine can not cope with the amount of data streaming in framerate may drop. This can either mean your USB bandwidth is not sufficient (see chapter on that) or your CPU/GPU is not fast enough to handle the amount of data being generated by the sensors. You can try to connect some sensors to a networked machines to balance the load.

Num Bodies	Number of bodies detected and tracked by a sensor (not used for PointCloud v3)
Num Faces	Number of faces detected on the tracked bodies by a sensor (not used for PointCloud v3)
Num Markers	Number of markers detected (during marker-based alignment) by a sensor
Aligned	Depicts if a sensor has been aligned or not (note that only aligned sensors will contribute to the skeleton solver)
Lock	Locked sensors will not be moved during alignment operations
Pos XYZ	position/translation of the sensor in the shared coordinate system
Rot XYZ	rotation of the sensor in the shared coordinate system Note that you can manually adjust these positions/rotations on the “Selected Sensor Settings”, they will automatically be adjusted by alignment operations

When using networked sensors, you’ll have a few options to control the amount of bandwidth vs quality consumed:

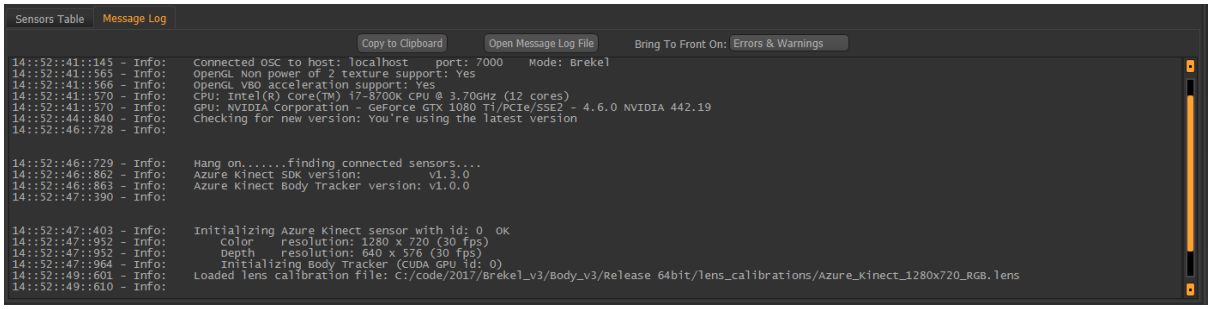
Quality	lower numbers will require less bandwidth but will look less detailed
FPS	lower numbers will stream fewer frames over the network to reduce bandwidth requirements

Note that this will only affect visual quality in the 3D viewport.

Skeleton data is always streamed and solved at full quality and framerate.

Recording BPC pointclouds will always happen at full quality and framerate as well.

MESSAGE LOG



The “Message Log” tab will display a history of all info, warning and error messages the application has reported since startup.

This will provide useful information on many operations as well as indicate if something went wrong.

The log will also be saved to the “Brekel” folder in your Windows Documents folder.

For example in: C:\Users\YourUsername\Documents\Brekel\ Brekel_Tester_v3.log

Copy to Clipboard

Copies the data to the Windows clipboard buffer so you can paste it into other applications.

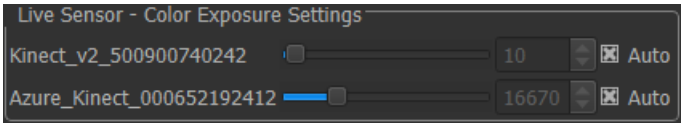
Open Message Log File

Opens the message log file in the system’s default text editor.

Bring To Front On

Sets if the message log window will automatically be brought to the front on errors, warnings or never.

EXPOSURE SETTINGS



After connecting to one or more sensors the Color Exposure Settings will appear on the tab on the left (for those sensor types that support it)

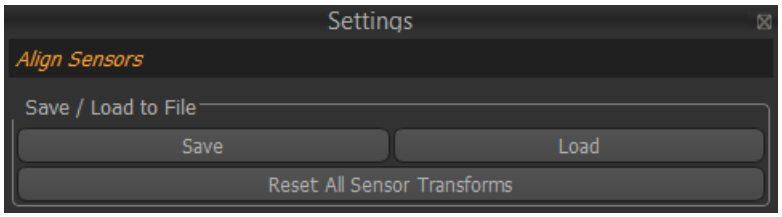
Generally you'll want to set the exposures so the colors in the video match well across sensors.

Auto exposure can be convenient but may not always work depending on lighting, in those cases it can be a good idea to disable "Auto" and manually set the exposure while watching the results in the Video window.

SENSOR ALIGNMENT

Sensor alignment calibration can be done in various ways and the result is that the sensor transforms are set and are no longer at the origin (0,0,0) of the world.

- Once calibration is done do not move your sensors, or you will have to do another calibration
- Your calibration will automatically save/load upon exit/start of the app
- You can tweak a selected sensor's transform on the "Selected Sensor" tab



You can find all alignment options on the "Align Sensors" tab in the "Settings" Window.

Let's start with the basic alignment calibration options:

Save

Saves sensor alignment data for all the currently connected sensors to a text file.

Load

Loads a sensor alignment text file and for all applies the alignment data to all currently connected sensors it finds a match for. Note that when your sensors have physically moved, you'll have to perform a new alignment calibration.

Reset All Sensor Transforms

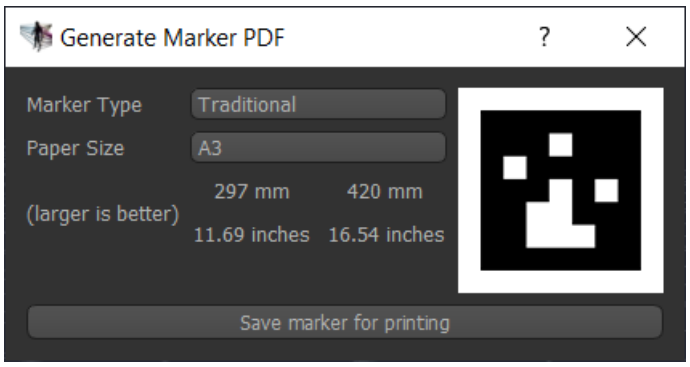
Resets all sensors and removes all alignment calibration data.

Note that you can also load sensor alignment files from Brekel Body v3, which in addition also provides functionality to align sensors based on body tracking data.

2D PRINTED MARKERS

The most flexible way to calibrate your sensors is by using a 2D printed marker.

“Settings > Save Marker For Printing” from the top menu opens this to generate a PDF file you can print.



- You’ll want your print to be as be big as possible/practical (the more pixels seen by the sensor the more accurately it can calibrate itself)
- In most cases A3 is a good default size.
- It is important that you leave white borders around the marker when it’s printed, the PDF file already has these in place. Note that the exact size of these margins isn’t important.
- After printing you will need to attach the marker to a rigid surface so it cannot bend, cardboard or foamboard are great for this purpose.

It is important that during calibration the paper size (and marker length) in the GUI matches the physical size as it’s internally used for the initial guess of the marker size.

For best results you can measure the large black square on your printed marker and use “Custom Size” instead of the default paper sizes in the tracker settings, or use the “Auto Optimize Marker Length”.

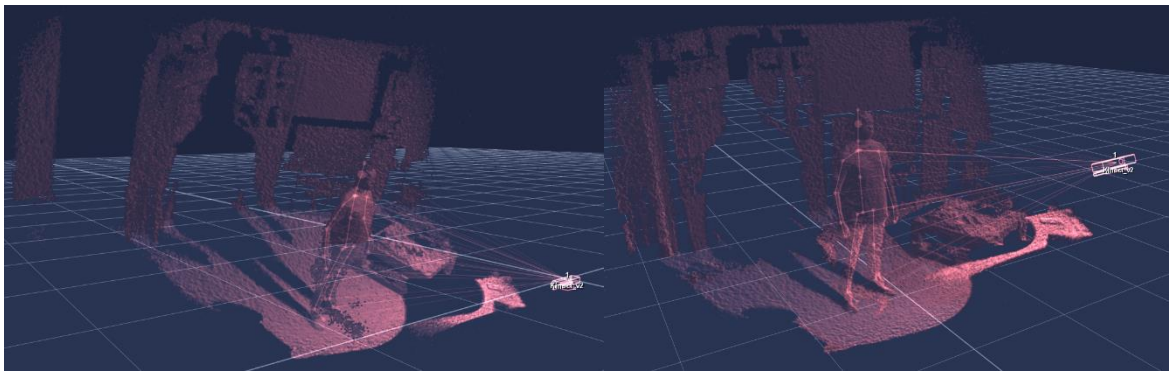
The “traditional” marker type is the default pattern but there is a “new” type which is a bit faster to detect, should in theory be a bit more robust during movement but is still experimental.

During calibration it is important to both **move & rotate** the marker around your capture volume **slowly & smoothly**, while making sure multiple sensors will be able to see it from different positions and orientations to give the calibration algorithm enough variation.

Note that (based on the original transform of the primary sensor) your result may not be in a correct final orientation and position based to the 3D grid/world in the viewport, until the floor level refinement functionality is used.

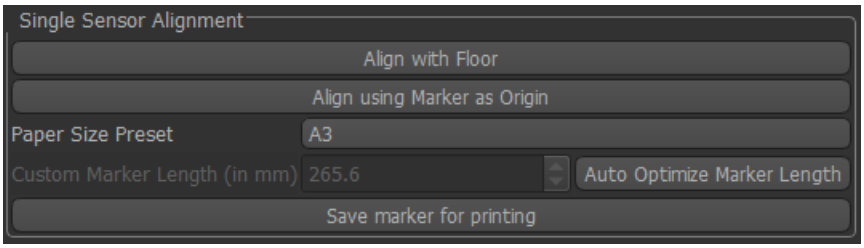
SINGLE SENSOR ALIGNMENT

When dealing with a single sensor setup all you need is the physical floor to match the floor in the 3D viewport. Adjusting for the physical sensor height from the floor and its rotation.



Not Aligned Aligned

When running a single sensor, you can align it in different ways:



Align with Floor

This will try to detect the floor and adjust the sensor's translation and rotation such that the floor is level and at y=0 in the 3D viewport. Note that when the floor cannot be seen this may give incorrect results.

Align using Marker as Origin

This will align the sensor so a visible 2D marker is at the origin.

Please read the chapter "sensor alignment using 2d printed markers" below on more information on markers.

Paper Size

Size of the physically printed marker, the tracker needs this to accurately estimate where it is in 3D space.

Custom Marker Length

Measured length (in millimeters) of the large main black square of your printed marker, the tracker needs this to accurately estimate where it is in 3D space.

Auto Optimize Marker Length

Automatically optimize the marker length (based on the hint you give it) using video and pointcloud data.

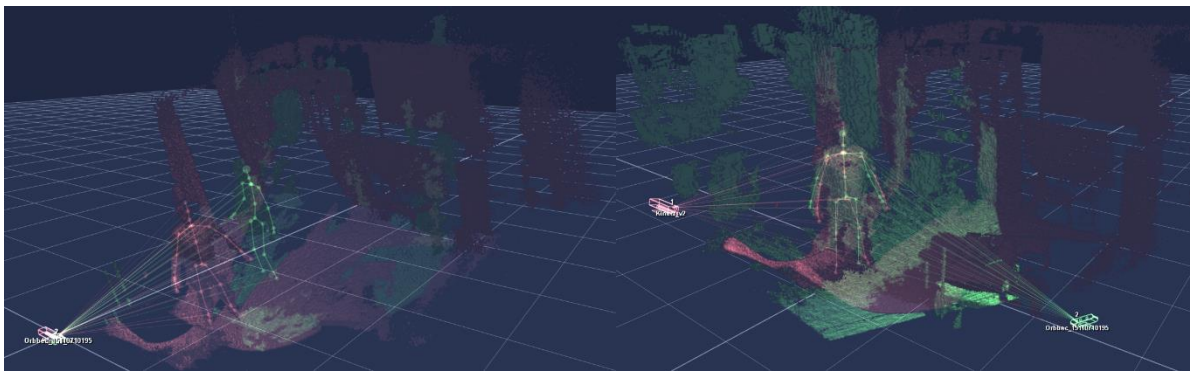
This can help getting the most accurate results

Save marker for printing

This opens this to generate a PDF file you can print. (same as “Settings > Save Marker For Printing” from the top menu)

MULTI SENSOR ALIGNMENT USING MARKERS

When dealing with multiple sensors (two in this example for clarity but you can of course use more) the sensors need to be aligned with the 3D world coordinate system but also be aligned to each other so their data overlaps correctly.



Not Aligned Aligned

Multi sensor alignment calibration can be performed in 2 ways:

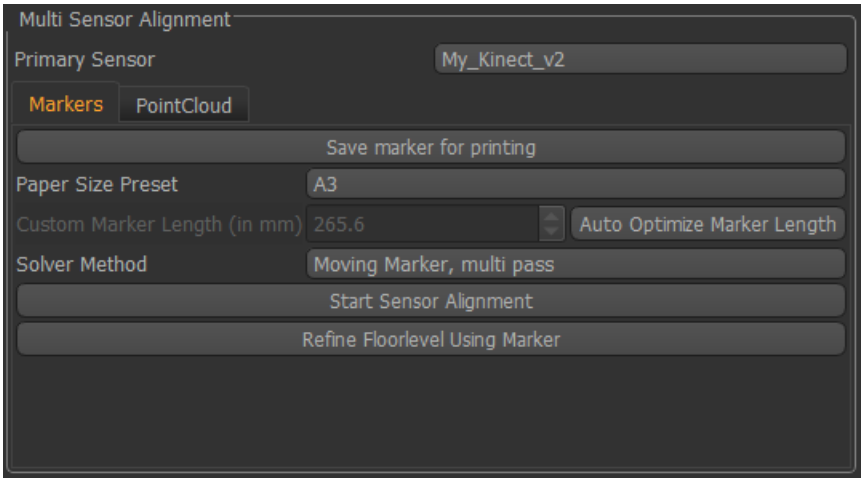
- Using 2D printed markers (most flexible)
- Using pointcloud data (heavily dependent on sensor overlap)

Note that you can do marker alignment both with live sensors as well as using recorded clips on the timeline.

Doing it offline with recorded clips can be more accurate as the solver can take advantage of time samples from both the past and future.

Note that you can lock specific sensors, using the toggle in the Sensor Table, to prevent them from being adjusted, this also allows you to re-calibrate a single sensor if it has moved for example.

Note that you can also load sensor alignment files from Brekel Body v3, which in addition also provides functionality to align sensors based on body tracking data.



Save Marker for printing

This opens this to generate a PDF file you can print. (same as “Settings > Save Marker For Printing” from the top menu)

Paper Size

Size of the physically printed marker, the tracker needs this to accurately estimate where it is in 3D space.

Custom Marker Length

Measured length (in millimeters) of the large main black square of your printed marker, the tracker needs this to accurately estimate where it is in 3D space.

Auto Optimize Marker Length

Automatically optimize the marker length (based on the hint you give it) using video and pointcloud data.

This can help getting the most accurate results

Solver Method

"Static Marker" assumes a marker on the floor at the center of your volume where all sensors can see it.

"Moving Marker, single pass" assumes you slowly move and rotate the marker across the capture volume.

The solver will align sensors based on all frames where the marker was seen by the primary sensor and other sensors.

"Moving Marker, multi pass" assumes you slowly move and rotate the marker across the capture volume.

The solver will do an initial alignment based on all frames where the marker was seen by the primary sensor and other sensors (like the single pass method). And then refine it using all data from all frames of all sensors top get an optimal result with the least amount of numerical error.

Start Sensor Alignment

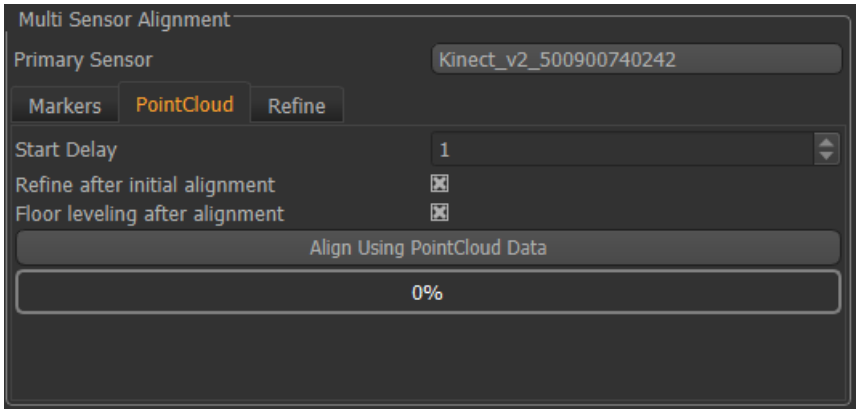
Starts/Stops the alignment calibration.
Note that the solver doesn't know yet where the floor is so while aligning sensors to each other they may still be rotated in regards to the floor

Refine Floorlevel Using Marker

Assumes a marker placed on the floor in the center of your volume where one or more sensors can see it.

All sensors will be adjusted (translation & rotation) so the marker is at the origin (0,0,0) of the virtual world and level with the floor.

SENSOR ALIGNMENT USING POINTCLOUD DATA



Besides using skeleton and marker data it is also possible to align sensors based on pointcloud data. This can be highly accurate but is also very dependent on the angle between sensors as it's highly dependent on overlapping points of neighboring sensors seeing the same parts of a person.

After starting the alignment the app will start a countdown so you have time to walk into the volume.

During alignment you should **stand still**

It helps to stand in an asymmetric pose for example by raising one arm like this:

Initial alignment will be guessed based on your primary sensor.

Note that this is highly dependent on pointcloud overlap and there is a small element of randomness so you may need to try again if alignment fails.

Start Delay

Start delay (in seconds), gives you some time to walk into the volume after starting alignment.

Refine after initial alignment

After estimating the initial (rough) alignment this will run additional refinement passes (same as manually using “Refine Using PointCloud” on the “Refine” tab.

Floor leveling after alignment

After estimating the initial (rough) alignment this will try to detect the common floor plane between pointcloud data from all sensors and both level and center things based on that.

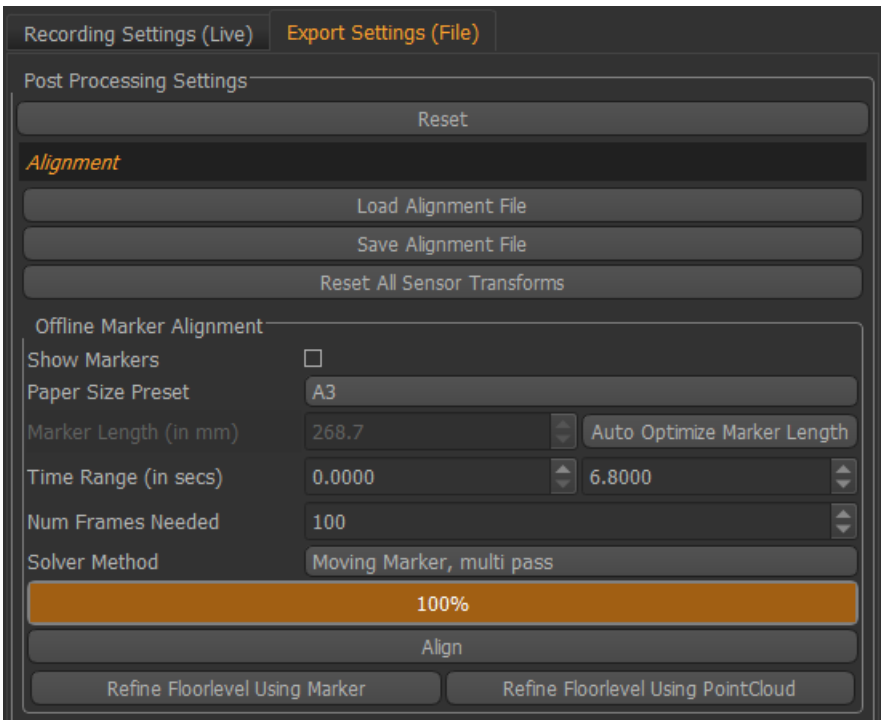
Align Using PointCloud data

Start/Stop alignment process.

OFFLINE SENSOR ALIGNMENT REFINEMENT USING MARKERS

Besides doing sensor alignment from live sensors you can also do it using clips on the timeline.

You will need to record clips where you move and rotate your marker slowly through the volume, for floor leveling it's a good idea to either start or end the clip with the marker on the floor in the middle of the volume where one or more sensors can see it.



Load Alignment File

Loads a previously saved alignment and applies it to the clips on the timeline. A popup window may appear to relate the sensors in the file to the sensors of the clips on your timeline.

Save Alignment File

Saves the current alignment to a file that can be re-used later on.

Reset All Sensor Transforms

Resets all sensor transforms to their defaults.

Show Markers

Detect and show markers while scrubbing through the timeline.

Paper Size

Size of the physically printed marker, the tracker needs this to accurately estimate where it is in 3D space.

Custom Marker Length

Measured length (in millimeters) of the large main black square of your printed marker, the tracker needs this to accurately estimate where it is in 3D space.

Auto Optimize Marker Length

Automatically optimize the marker length (based on the hint you give it) using video and pointcloud data.

This can help getting the most accurate results.

Time Range

Start and end time (in seconds) of the timeline between which to sample frames with markers.

Num Frames Needed

Amount of frames to sample (between start/end time) on your timeline to collect markers.

Solver Method

"Static Marker" assumes a marker on the floor at the center of your volume where all sensors can see it.

"Moving Marker, single pass" assumes you slowly move and rotate the marker across the capture volume.

The solver will align sensors based on all frames where the marker was seen by the primary sensor and other sensors.

"Moving Marker, multi pass" assumes you slowly move and rotate the marker across the capture volume.

The solver will do an initial alignment based on all frames where the marker was seen by the primary sensor and other sensors (like the single pass method). And then refine it using all data from all frames of all sensors top get an optimal result with the least amount of numerical error.

Align

Loads markers from the clips (and caches them) and performs alignment.
Note that the solver doesn't know yet where the floor is so while aligning sensors to each other they may still be rotated in regards to the floor

Refine Floorlevel Using Marker

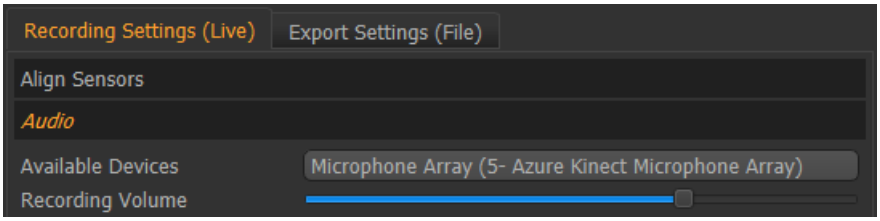
Assumes marker placed on the floor in the center of your volume on one or more clips of the current frame.

All clips will be adjusted (translation & rotation) so the marker is at the origin (0,0,0) of the virtual world and level with the floor.

Refine Floorlevel Using PointCloud

If the floor was visible on the current frame of one or more clips this will adjust the clips translation & rotation to match that floor.

AUDIO SETTINGS



On this panel you can set audio related settings, note that the recorded audio is embedded in the BPC (Brekel PointCloud) files along with the pointcloud and video data.

Available Devices

Lists all detected audio devices (from sensors and other sources) from your system and selects which one to use for audio recording. Select “None” (first option in the list) to not record audio.

Number of Channels

Select to record mono or stereo audio.

Sampling Rate

Select which sampling rate to use, higher is better but produces larger file sizes.

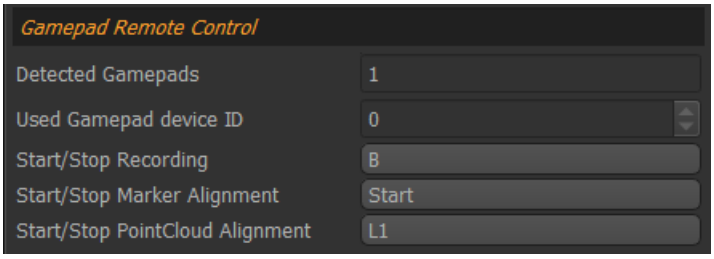
Recording Volume

Sets the volume used for audio recording.

Speakers Passthrough

When enabled passes the sound from the input device through to your speakers.

GAMEPAD REMOTE CONTROL



If you own a gamepad, for instance a wireless or wired Xbox controller, it can be used as a remote control for certain features in the app.

Any device that shows up as a gamepad device in Windows should work.

For example it can be handy to start/stop recording or alignment functions using a wireless gamepad while standing in the capture volume.

Detected Gamepads

Gamepads are automatically recognized when they connect/disconnect, this lists how many devices were detected by Windows.

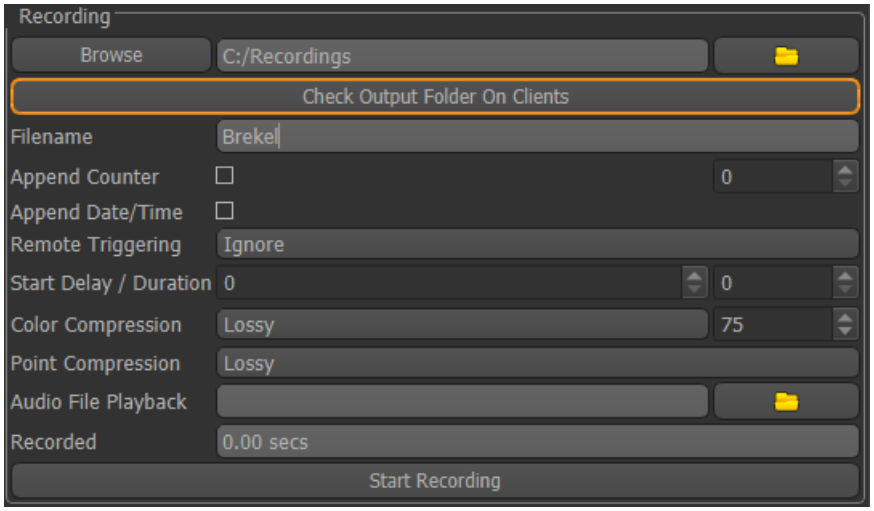
Used Gamepad Device ID

In case you have multiple gamepads connected to your system this settings selects which one is used.

Start/Stop

Available standard gamepad buttons can be connected to actions to start/stop various options in the software here. A particular button can only be connected to a single action at a time.

RECORDING SETTINGS



Folder

You can set the output folder for the BPC (Brekel PointCloud) files here, either by using the Browse button to bring up a file browser or by manually setting the path.

Folder Icon

Open the output folder in Windows Explorer

Check Output Folder On Clients

If you're using network sensors this will ask those machines to check if the set folder exists locally.

Networked computers will always record data locally as recording over a network would more than likely result in dropped frames.

So if you set the recording folder to "C:/Recordings" each machine will write to it's own C: drive.

Filename

Output filename to use for recording.

Note that counters and date/time can be appended to this (see below) to make sure filenames are unique.

Sensor names will also be added to the file name for this same purpose.

Append Counter

When enabled adds a counter to the end of the filename to make then unique and prevent overwriting older files. You can adjust the current counter manually using the setting.

Append Date / Time

When enabled adds the current date and time to the end of the filename to make then unique and prevent overwriting older files.

Remote Triggering

Allows synchronized recording across multiple Brekel applications.

One application can be in Primary mode, all others in Secondary or Ignore mode.

The Primary application will send a signal when recording is started and stopped so all applications start/stop at the same time and are using matching filenames.

Note that this works across multiple apps on the same machine and even across multiple machines on the same network.

Make sure your firewall is not blocking port 8880-8890.

Start Delay / Duration

Start Delay will postpone the start of the recording for the set number of seconds after pressing the “Start Recording” button. A countdown will appear in the viewport and audio beeps will play for reference. A setting of 0 will disable this and start recording immediately after pushing the “Start Recording” button.

Duration sets a fixed length (in seconds) for the recording, after reaching it the recording will automatically stop. A setting of 0 will disable this.

Color Compression

Lossless will result in the best quality but also produces much larger files sizes.

Lossy compression reduces the file size but will also introduce some (minor) quality loss.

In general, lossy compression is the best option as the quality reduction is minimal.

1 will produce the best compression but lowest quality

100 will produce least compression but highest quality

Point Compression

Lossless will result in the best quality but also produces much larger files sizes.

Lossy compression reduces the file size but will also introduce some (minor) quality loss.

In general, lossy compression is the best option as the quality reduction is minimal.

Depending on the contents reductions of 30% in file size are quite typical.

Audio File Playback

Plays back an audio file to during recording so your actor can use it for timing reference (in WAV file format)

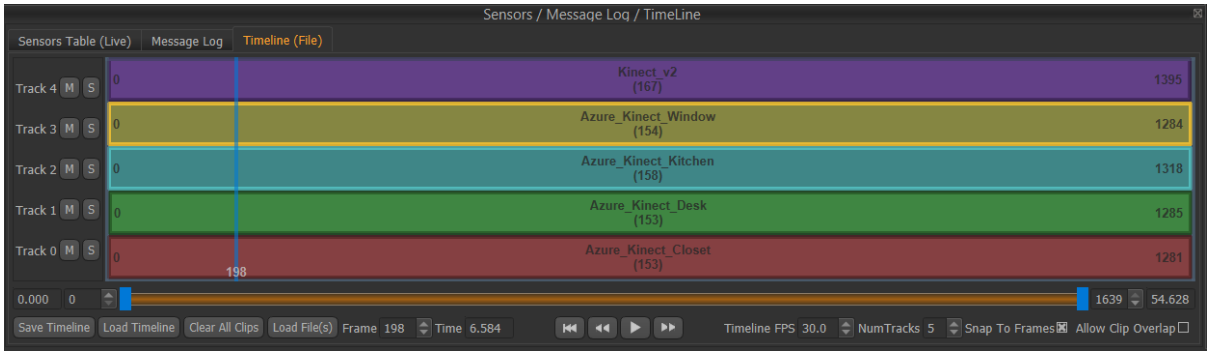
Recorded

Shows the number of seconds of the current (or last) recording.

Start Recording

Starts the recording, hit the button again to stop recording.

TIMELINE



The timeline is where you load and manipulate recorded BPC (Brekel PointCloud) clips so their data can be fused and exported to the various file formats.

If you are familiar with video editing applications the concept is similar here.

The timeline consists of several tracks stacked vertically.

Each track contains clips (BPC files) displayed as colored bars based on their duration.

Tracks can be “Muted” using the M button, clips on a muted track will not be displayed in the viewport and will not used during data fusion/export.

Tracks can be set to a “Solo” state using the S button, setting a track to Solo is the same as muting all other tracks and has the effect of seeing the clips on the track in isolation.

TIMELINE NAVIGATION

Left Mouse Button Click/Drag

Sets the current frame (displayed as a vertical blue line) and reloads the frames from the clips so they can be displayed in the viewports.

Shift + Left Mouse Button Click

Selects a clip.

Middle Mouse Button Click

Deselect clips.

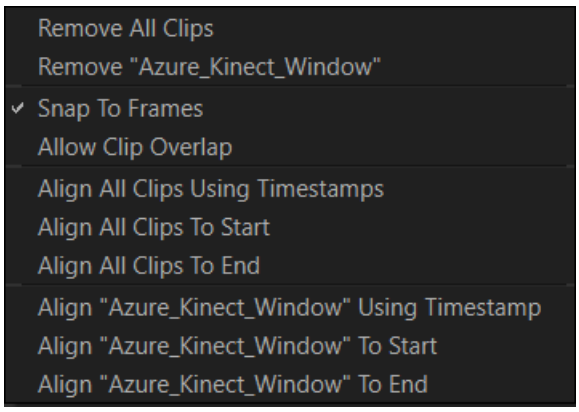
Shift + Left Mouse Button Drag

Click/drag on the middle of a clip allows you to drag it on the timeline and to different tracks.

Click/drag on the start/end of a clip will drag it’s in- or out-point .

RIGHT MOUSE CLIP CONTEXT MENU

Right clicking on a clip brings up a context menu based on where you click:



Remove All Clips

Clears all the clips from the timeline and empties the tracks.

Remove “Clip Name”

Removes the clip under the mouse pointer from the timeline.

Snap To Frames

When enabled snaps clips, when dragging them, to whole frames.

When disabled allows clips offset to be fractions of whole frames.

Allow Clip Overlap

When enabled allows clips to overlap when dragged onto the same track.

When disabled tries to prevent collisions from overlapping clips.

Align All Clips Using Timestamp

Moves all clips on the timeline so their timestamps match to how they were recorded.

Align All Clips To Start

Moves all clips so their start points align

Align All Clips To End

Moves all clips so their end points align

Align “Clip Name” Using Timestamp/Start/End

Same as the “All Clips” variants above but just for the clip under the mouse pointer.

The orange line underneath the timeline and above the buttons allows you to zoom into the timeline by dragging the ends towards the middle or setting the first/last frame manually.

Save Timeline

Save timeline settings to a file, note that this only stores references to the file locations of the clips and not the actual clip data itself. This includes all things like clip, data fusion and export settings.

Load Timeline

Load timeline settings from a saved file. This includes all things like clip, data fusion and export settings.

Clear All Clips

Clears all clips from the timeline and empties the tracks.

Load File(s)

Opens a file browser to load BPC (Brekel PointCloud) files.

Note that you can also drag & drop files from Windows Explorer onto the timeline to load them.

Frame / Time

The current frame/time to be displayed in the viewport.

Rewind

Sets the current frame to the start of the timeline.

Previous Frame

Moves current frame to the previous frame.

Play/Pause

Starts/stop playback, note that this currently tries to display all frames and does not skip frames if realtime playback is not possible due to file loading times.

Next Frame

Advances current frame to the next frame.

Timeline FPS

Sets how many frames per second should be used for export.

Note that clips themselves store data according to timestamps and can have a different frame rate.

The app will internally for each clip load the frame with the closest possible timestamp, if the export frame rate is higher than recorded in the clip frames will be duplicated, if it is lower frames will be skipped.

NumTracks

Number of tracks in the timeline, this will automatically be raised when loading additional clips.

Snap To Frames

When enabled snaps clips, when dragging them, to whole frames.

When disabled allows clips offset to be fractions of whole frames.

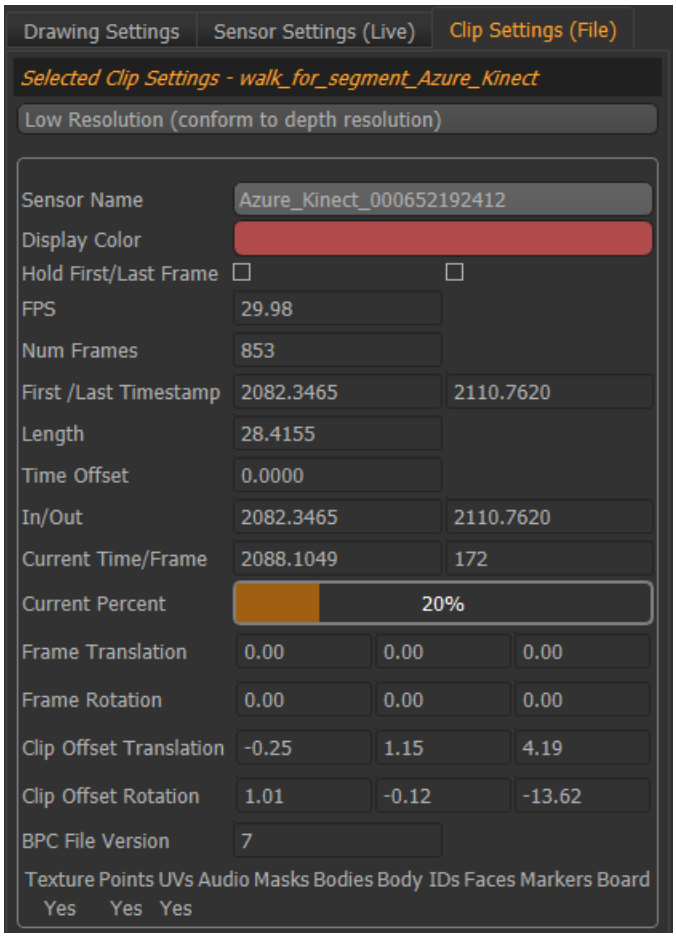
Allow Clip Overlap

When enabled allows clips to overlap when dragged onto the same track.

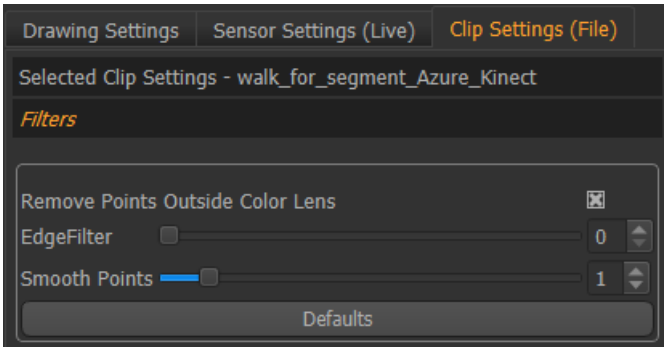
When disabled tries to prevent collisions from overlapping clips.

CLIP SETTINGS

This tab (top left) will display information and allow you to change various things regarding the currently selected clip.



FILTERS



Remove Points Outside Color Lens

On many depth sensors there are 2 different lenses, one for color and another for depth. There may be points that fell outside of the area the color sensor covered.

Enabling this option removes those points as they would otherwise result in black areas with no color information.

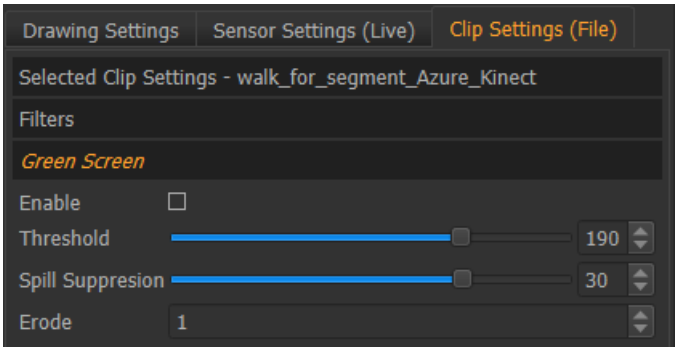
EdgeFilter

Helps to clean up edges in areas of large differences in depth, for example between a person and the background.

Smooth Points

Smooths out pointcloud data to remove surface noise.

GREEN SCREEN



Enable

When enabled this removes points with a green color from the pointcloud.

You can use this with a “green screen” to isolate people from a green background which is a common technique in visual effects (also typically used for a TV weather person standing in front of a map for example).

Threshold

Controls the cutoff point for when a point is determined to belong to the greenscreen.

Slide this to remove more or fewer points depending on the exact pixel color intensities depending on how your greenscreen was lit.

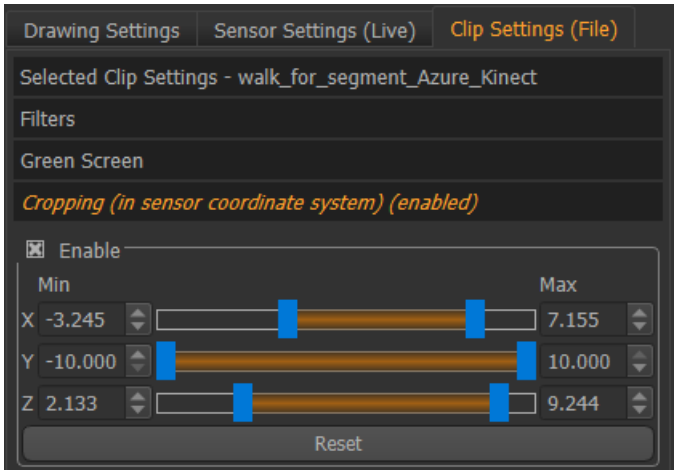
Spill Suppression

Percentage of color correction to apply to get rid of green reflection from the green screen onto the subject.

Erode

Amount of pixels to shrink the mask to clean up the edges.

CROPPING (IN SENSOR COORDINATE SYSTEM)



Enable

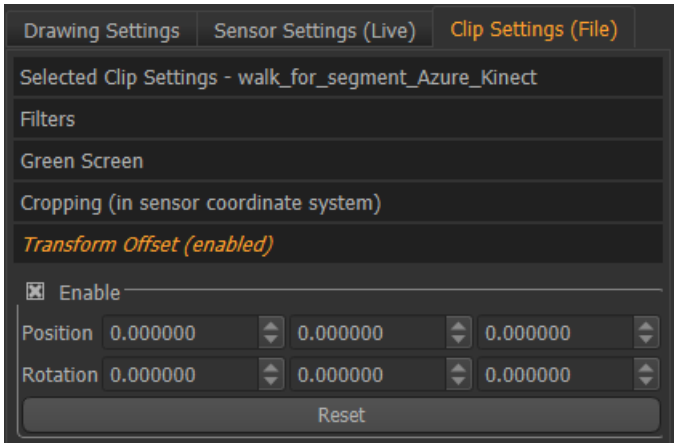
When enabled removes points that fall outside the min/max settings.

Note that everything is in the sensor coordinate system so Z for example is distance from the sensor and not the Z world axis in the 3D viewport.

Min/Max

Minimum/Maximum for each axis, points outside these ranges will be cropped out.

TRANSFORM OFFSET



Allows you to offset the pointcloud/sensor by manually specifying the position and rotation.

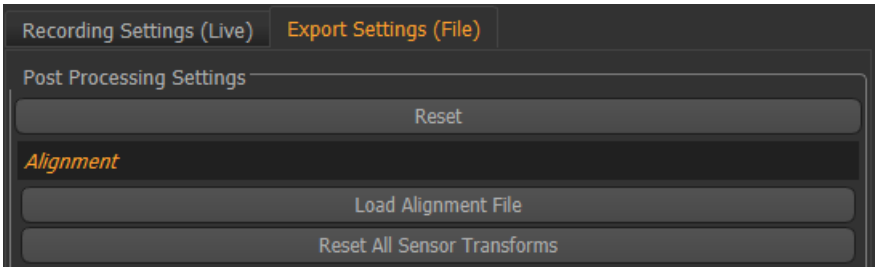
POST PROCESSING SETTINGS

On this tab on the right you will find settings that apply to all the pointclouds simultaneously.

Reset

Sets all settings on this tab to their defaults

ALIGNMENT



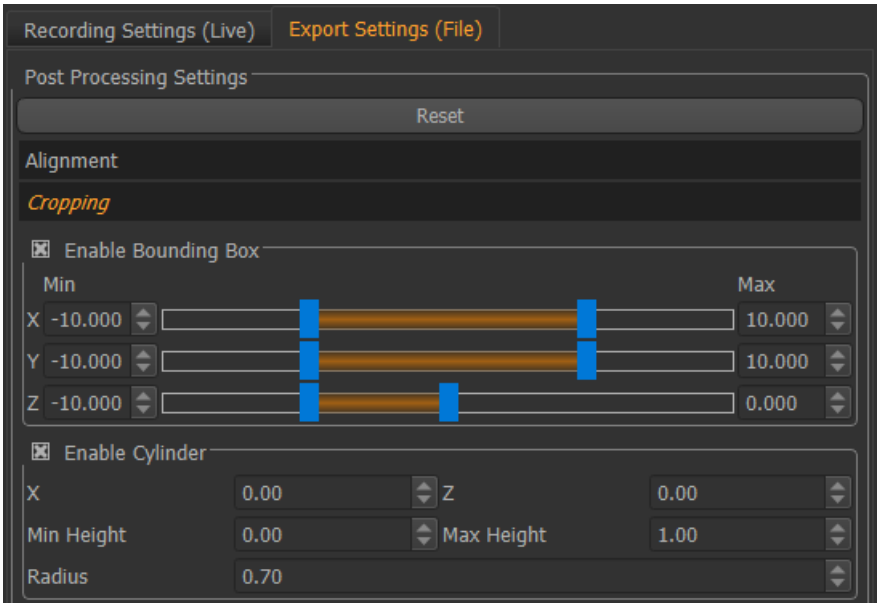
Loads Alignment File

Allows you to load an alignment file created from PointCloud v3 in Live mode or from Body v3.

Reset All Sensor Transforms

Resets alignment of all sensors.

CROPPING



Enable Bounding Box

When enabled points outside this box (defined by the min/max settings) will be removed.

The X/Y/Z axis align with the global coordinate system as displayed in the 3D viewport

Enable Cylinder

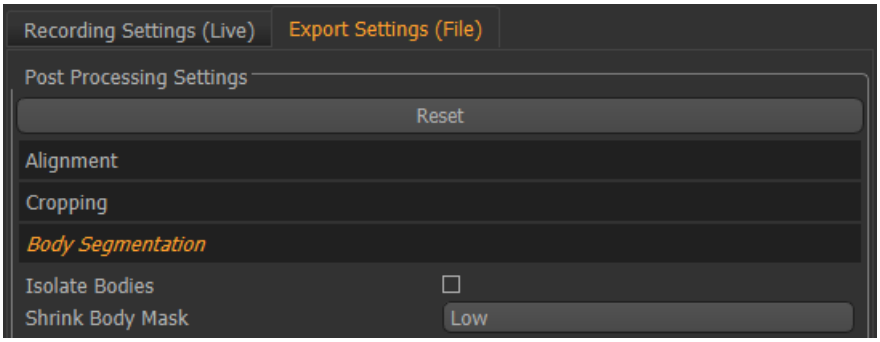
When enabled points outside the defined cylinder will be removed.

XZ define the centerline of the cylinder.

Min/Max Height define the bottom and top in the Y direction.

Radius defines how large the cylinder is.

BODY SEGMENTATION



Isolate Bodies

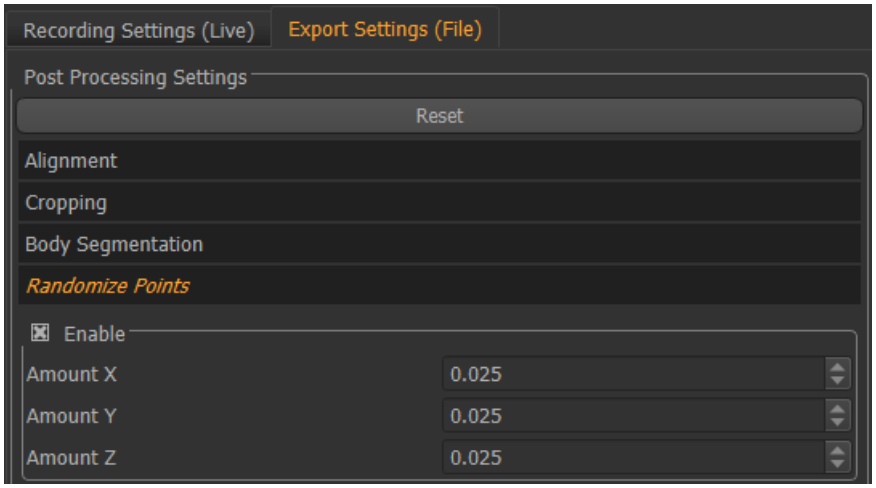
When enabled the software uses a deep learning method to automatically identify humans in the pointcloud data and remove all background points.

Shrink Body Mask

Since there may be some ambiguity on the edges for points belonging to an identified human or the background it is generally a good idea to shrink the mask a bit to clean things up.

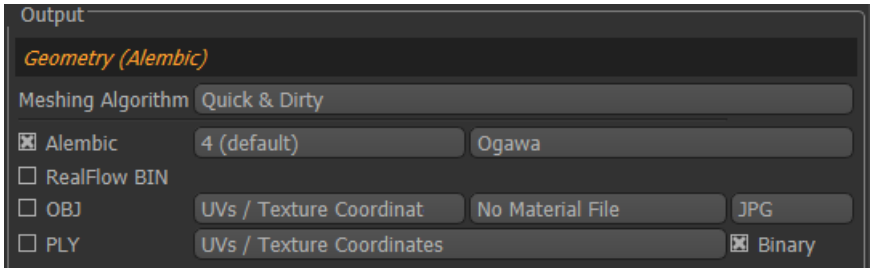
This setting defines the amount of cleanup used.

RANDOMIZE POINTS



Applies noise to the points based on the amounts set for each axis.

GEOMETRY / MESH OUTPUT



Specifies which mesh cache formats to write to disk during export.

Meshing Algorithm

Specifies which algorithm to use to generate meshes from the pointcloud data. (more options may be added in the future with different quality/speed/compute characteristics)

Alembic

Exports all frames to a single Alembic (.ABC) file, this is one of the most efficient file formats for meshes, natively supported by pretty much all 3D apps out there. This is generally the most efficient and easy option to work with meshes.

Alembic Compression

Alembic files can be compressed (lossless) to save disk space, this sets the level of compression.

Higher values produce smaller files but will be slower to export.

Alembic Compression Type

Not all 3D apps support both HDF5 and Ogawa compression types, you may want to experiment what's best for you regarding support in your favorite 3D app, file size and compression/decompression speed.

RealFlow BIN

Exports meshes to a sequence of Realflow BIN files.

You can download free "Connectivity Plug-ins" for most 3D applications from the Realflow website.

OBJ sequence

Exports meshes to a sequence of Wavefront OBJ files.

These files can be read by most 3D applications, but not all applications may be designed to handle sequences of OBJ files efficiently. Due to the lack of compression for OBJ files they generally are rather large.

OBJ color settings

- No UVs / Vertex Colors exports only mesh information.
- UVs / Texture Coordinates exports UV data embedded into the OBJ files so that textures can be mapped correctly.
- Vertex Colors exports RGB color data per vertex.

OBJ material file settings

This specifies if you want to export an MTL material file along with each OBJ file. (most applications don't need this but some like MegaCache for Unity may do).

You can select which texture format to write into the MTL file. (note that this will not enable texture export automatically, it just writes the path into the MTL file).

PLY sequence

Exports meshes to a sequence of PLY files.

PLY color settings

- No UVs / Vertex Colors exports only mesh information.
- UVs / Texture Coordinates exports UV data embedded into the PLY files so that textures can be mapped correctly.
- Vertex Colors exports RGB color data per vertex.

PLY Binary

When enabled will export binary files which are smaller but may not be supported by all applications.

When disabled will export in the ascii text file format which are larger but more generally supported.

SenseXR - https://www.senseofspace.io

Export mesh and texture to a format compatible with Sense of Space's Sense XR Studio.

You can select either JPG or PNG for textures, JPG's be smaller on disk but the lossy compression may introduce some artifacts and are a bit slower than PNG.

You can simply import the asset using "File > Import > Alembic (.abc)" in Sense XR Studio.

It should automatically find the associated texture sequence.

SenseXR Studio can compress and package everything into a single file that can be used with their Unity game engine playback integration.

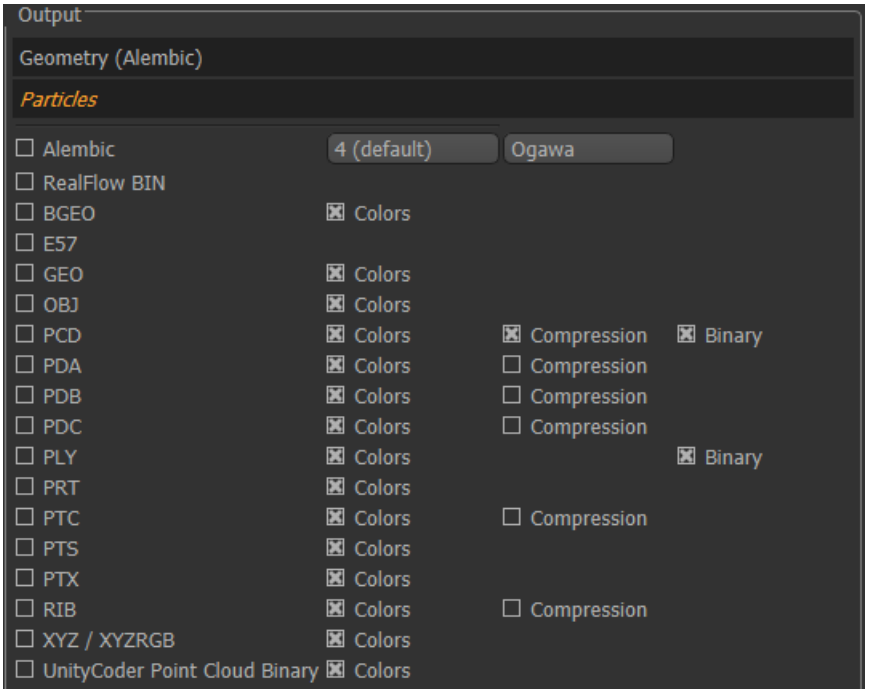
Holo CatchLight - https://assetstore.unity.com/packages/add-ons/holo-catchlight-plugin-pro-177405

Export mesh and texture to a format compatible with Holo CatchLight by Prometheus Vision.

You can select either JPG or PNG for textures, JPG's be smaller on disk but the lossy compression may introduce some artifacts and are a bit slower than PNG.

This can be used with CatchLight's "ObjSeqConverter" application to convert it to an MP4 file that can be used with their Unity game engine integration.

PARTICLE / POINT OUTPUT



Alembic

Exports all frames to a single Alembic (.ABC) file, instead of writing meshes this export data as points.

Almost all 3D apps out there support Alembic files but not all of them support point data from Alembic, some will only accept meshes.

Alembic Compression

Alembic files can be compressed (lossless) to save disk space, this sets the level of compression.

Higher values produce smaller files but will be slower to export.

Alembic Compression Type

Not all 3D apps support both HDF5 and Ogawa compression types, you may want to experiment what's best for you regarding support in your favorite 3D app, file size and compression/decompression speed.

RealFlow BIN

Exports points to a sequence of Realflow BIN files.

You can download free "Connectivity Plug-ins" for most 3D applications from the Realflow website.

BGEO

Exports points to a sequence of Houdini BGEO files.

E57

Exports points to a sequence of E57 pointcloud files.

GEO

Exports points to a sequence of Houdini GEO files.

OBJ

Exports points to a sequence of Wavefront OBJ files.

PCD

Exports points to a sequence of PCD (PointCloud Library) files.

PDA

Exports points to a sequence of PDA files (for use in Maya).

PDB

Exports points to a sequence of PDB files (for use in Maya).

PDC

Exports points to a sequence of PDC files (for use in Maya).

PLY

Exports points to a sequence of PLY files.

PRT

Exports points to a sequence of PRT files, for use with Krakatoa.

PTC

Exports points to a sequence of PTC files, for use with Renderman compliant renderers.

PTS

Exports points to a sequence of Leica PTS pointcloud files.

PTX

Exports points to a sequence of Leica PTX pointcloud files.

RIB

Exports points to a sequence of RIB files, for use with Renderman compliant renderers.

XYZ/XYZRGB

Exports points to a sequence of XYZ pointcloud files, or XYZRGB when color is enabled.

UnityCoder Point Cloud Binary

Exports to the native file format of the UnityCoder plugins for Unity.

https://assetstore.unity.com/packages/tools/utilities/point-cloud-viewer-and-tools-16019

TEXTURE OUTPUT

JPG

Exports textures as JPEG files with the set amount of compression quality.

PNG

Exports textures as PNG files with the set amount of compression.

TGA

Exports textures as Targa files with or without compression.

TIFF

Exports textures as Tiff files with various types of compression.

Video

Exports textures as video files with the set amount of bitrate (higher bitrate is better quality but larger files)

OpenEXR

Exports raw depth to OpenEXR file format, preserving the full bitdepth and detail. (since this is bigger than any 8bit file format can store), optional Compression is lossless.

The value is a multiplier that allows you to keep the depth values in a user specified range.

PNG

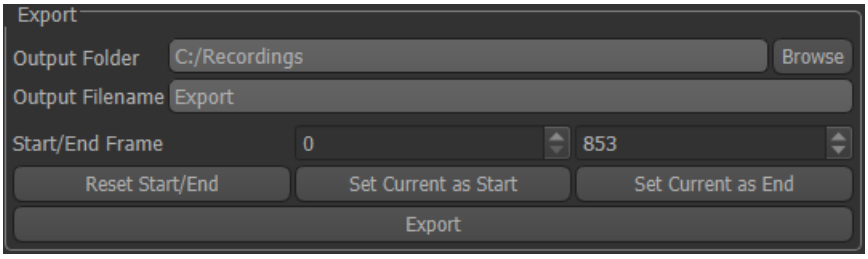
Exports raw depth to 16-bit PNG files.

SENSOR TRANSFORM OUTPUT

FBX

Exports the sensor transforms as FBX files using the specified file version and ascii/binary type.

EXPORT



Output Folder

Main folder to which to export the files, note that file sequences will be put in their own subfolder which will automatically be created if needed.

Output Filename

The base filename for the exported files, note that some file formats may append some additional information and/or frame numbers to the files.

Start/End Frame

Depicts which range of the timeline will be exported.

Export Start/End frames will be displayed on the timeline as transparent blue vertical lines for reference.

Reset Start/End

Resets the Start & End so the all the available frames will be used for export.

Set Current as Start

Sets the current frame as the start frame.

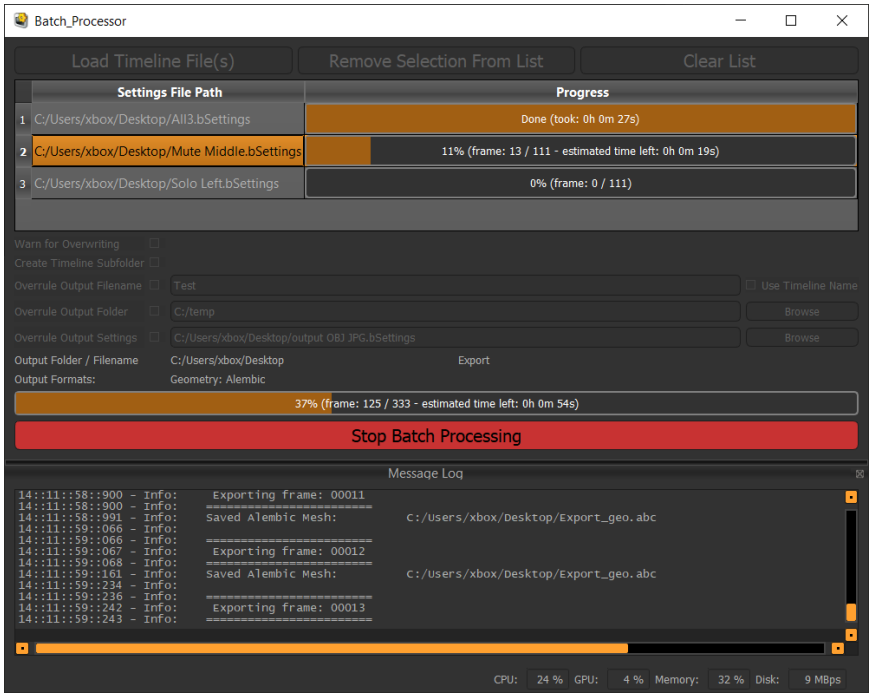
Set Current as Ends

Sets the current frame as the end frame.

Export

Will go through all frames based on the timeline FPS, fuse the data and export data to the various selected file formats.

BATCH PROCESSOR



The Batch Processor is an external program (which can be started from the button in the main app or from the Windows start menu shortcut for example) that can process multiple timeline files and export them to the desired output file formats.

Since the full GUI and visualization is not used this may be slighly quicker than invoking “Export” from the main app, it also has the advantage of processing multiple files all at once.

Load Timeline File(s)

This will bring up a file browser in which you can select one or multiple timeline files (“.bSettings” files saved from the main app using the “Save Timeline” option from the Timeline window).

These contain settings on what clips were on the timeline, post processing settings as well as the output formats to which to export to.

Loaded timeline files will appear in the table in the GUI.

You can also drag & drop “.bSettings” onto the table in order to add them.

Remove Selection From List

Removes selected items from the list.

Clear List

Clears all items from the list.

Warn for Overwriting

This will stop processing and display a warning when output files have the risk of overwriting existing files

Create Timeline Subfolder

This will create a subfolder (using the timeline filename) into which the various output files will be stored, this can be handy to prevent file overwriting when processing multiple files with the same output name.

Overrule Output Filename

This will overrule all output names to what is set in the GUI, instead of using the output filename specified in the timeline file's settings.

Use Timeline Name

This will overrule the output filename to match the timeline's filename.

Overrule Output Folder

This will overrule the output folder to what is set in the GUI, instead of using the output folder specified in the timeline file's settings.

Overrule Output Settings

This will overrule the output settings (which file formats to export to) from a specified ".bSettings" file, instead of using the timeline file's settings.

This works both with ".bSettings" files saved with the "Save Timeline" as well as "Save Output Settings" options from the main app.

Output Folder / Filename / Output Formats

These are read-only fields that display the respective information from a selected item from the table.

Start/Stop Batch Processing

Starts the actual processing, pushing the button during processing cancels the operation.

During processing the progress bars will display the percentage done, current frame, number of frames, estimated time left and total elapsed time when done. Both for individual items in the list as overall when processing multiple items.

LOADING DATA INTO THE UNITY GAME ENGINE

There are several options to load exported data into the Unity game engine.

Your best choice may depend on the look you desire (meshes/pointcloud) and which platform(s) you need to export to, so a bit of experimentation may be needed.

Meshes:

- Export to Sense of Space – Sense XR Studio format, load it into that app and export a compressed file for use with their Unity playback integration: https://www.senseofspace.io
- Export to Holo CatchLight format and use their converter and Unity playback integration: https://assetstore.unity.com/publishers/49805
- Export to OBJ and use MegaCache: https://assetstore.unity.com/packages/tools/modeling/mega-cache-26522
- Export to Alembic and use the Alembic plugin that Unity themselves provide.

Particle / PointCloud look:

- Export to UnityCoder file format and use MGear's Point Cloud Viewer and Tools plugin: https://assetstore.unity.com/packages/tools/utilities/point-cloud-viewer-and-tools-16019
- Export to Alembic and use the Alembic plugin that Unity themselves provide

LOADING DATA INTO UNREAL ENGINE 4

UE4 comes with Alembic file format support.

Although the PointCloud tools in UE4 do support LAS, XYZ & XYZRGB formats from the Brekel app it currently does not provide support for playback of sequences yet, only static pointclouds.

LOADING DATA INTO A 3D ANIMATION (OR OTHER) APPLICATION

Most 3D animation applications support Alembic files these days, which are convenient since they are compressed, a single file per sequence and generally the most efficient.

They can also contain UVs so you can directly map one of the texture outputs to them.

Alembic files with vertex colors and Alembic files with particles may not be supported by all apps.

For apps that are supported by one of the (free) RealFlow Connectivity plugins this may be an efficient option for both meshes & pointclouds as these files (like Alembic) are compressed and a single file per sequence.

For other apps and workflows there is a multitude of file formats to choose from.