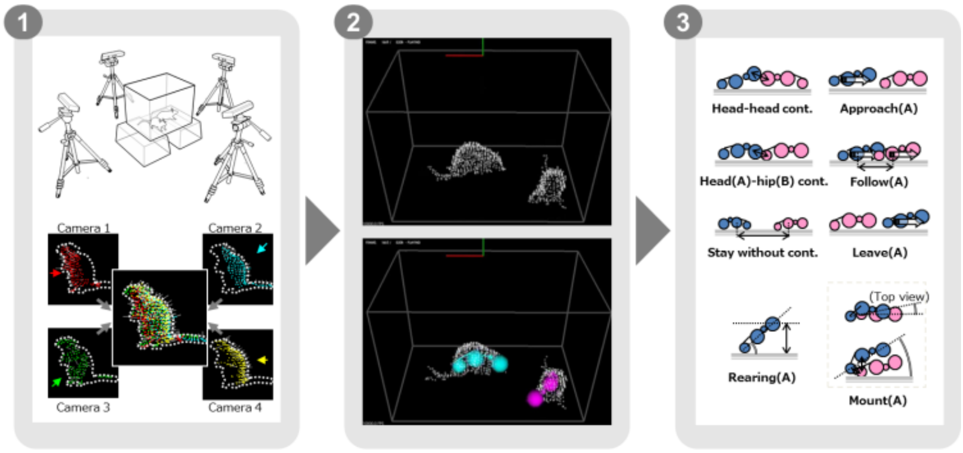
3DTracker-FAB Document

Introduction

3DTracker-FAB (for animal behavior) is an open source software for 3D-video based computerized behavioral analysis for laboratory animals. 3DTracker will help user to acquire and analyze 3D trajectory of body parts of animals, as shown below.



Matsumoto et al., 2013 (http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0078460)

[Fig 0-1]

First, a 3D image of animals is reconstructed from the images of multiple (~4) cameras (more specifically, depth cameras; Fig 0-1, left). Second, positions of body parts are estimated by fitting skeletal models to the 3D image (Fig 0-1, middle). Third, various behavioral parameters are calculated based on the trajectories of body parts (Fig 0-1, right).

In the actual workflow, the images from multiple cameras are calibrated and recorded, with "Recorder" software (Chapter 4). Then, the recorded images from the different cameras are processed off-line for integration to a full 3d image, down sampling, and etc., with "Preprocessor" software (Chapter 5). Then, skeletal models are fitted to the 3D image, with "Tracker" software (Chapter 6). Finally, the resultant 3D trajectories (csv file format) is analyzed with Python/MATLAB scripts (Chapter 7).

As of 2017 Nov, the present version of 3DTracker support analyzing only rats and mice (More species may be supported in the future), and the script for the post-processing of the trajectories are in preparation.

The 3DTracker-FAB is basically licensed under MIT license. So, note that the software is provided "as is", without warranty of any kind, express or implied.

Index

Chapter 1: Hardware preparation

Camera

Use Intel realsense R200 camera. A R200 contains a color camera and a depth sensor to obtain a RGB image and a depth image, respectively. R200 has an outstanding feature that there is almost no interference between cameras during multiple cameras are captured a same object simultaneously (see Chapter 9, Appendix for the detail of the camera).

[Fig 1-1]

# Unfortunately, R200 is now discontinued. However, there are still some stocks in the intel store as a part of a kit (https://click.intel.com/intelr-realsensetm-robotic-development-kit-2351.html). We are planning to support the new model, intel D435 or D415 in near future.

The original USB 3.0 cables of R200 is very short. So, the cable must be extended. For the extension, a "active" type cable (repeater cable), which amplify the signal, is recommended*. We got the best signal stability with the following extension cable: UGREEN USB 3.0 Active Extension Cable (5 meters long; https://www.amazon.com/dp/B01FQ88CE6)

*Because R200 uses almost all the transmission bandwidth of USB 3.0, the quality of the signal transmission is very important. People have reported that it is not so easy to extend the cable (https://software.intel.com/en-us/forums/realsense/topic/607341). So, we recommended to use the same product with us for the USB extension cables and the USB 3.0 expansion cards (shown below).

PC

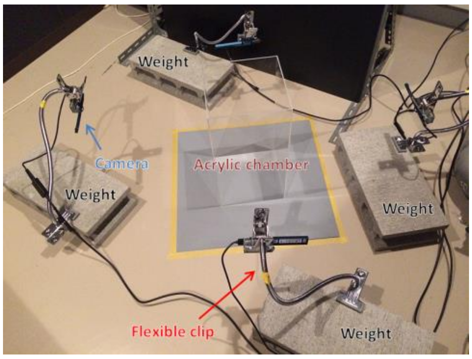
3DTracker (except for MATLAB/Python scripts) runs only on PC with Windows 10 64-bit. The CPU for the PC used for the recording is not necessary to be very fast, but the number of cores (or the number of parallel processes) in the CPU should be same as, or more than the number of cameras that you will use simultaneously, because the Recorder software processes each of the camera images in parallel. In addition, a large HDD should be installed to the PC, because the 3D video data is relatively big. The PC monitor resolution should be more than 1280x1024. The following example PC is what we have used for recording with four R200 cameras:

HP EliteDesk 800 G2 Tower PC; CPU: Core i7-6700 (3.4 GHz); RAM: 8GB; HDD: 500 GB + 6TB; Power unit: 400W 80PLUS; Three Empty PCIe slots.

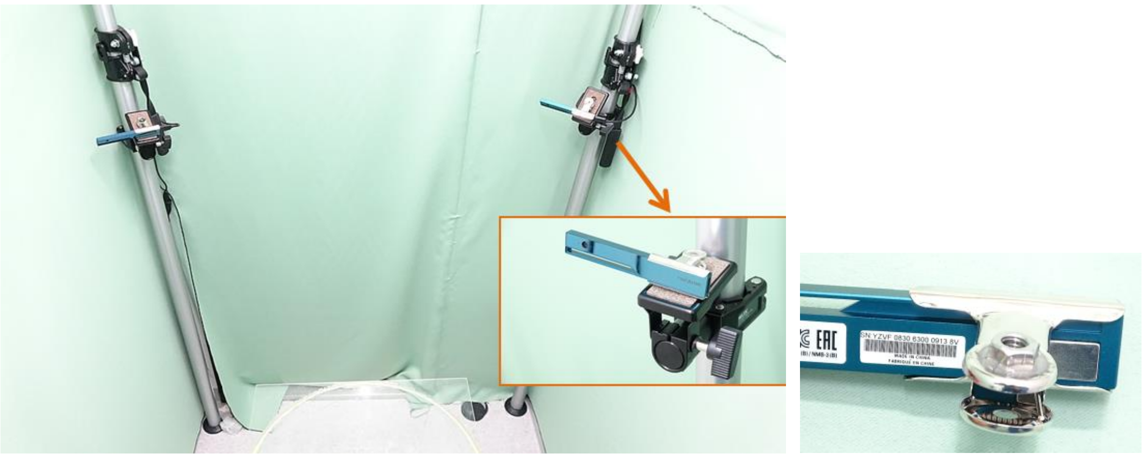
As noted above, R200 uses almost all the transmission bandwidth of USB 3.0. Thus, it is not recommended to connect more than one R200 to a USB controller. To avoid that, we added three of the following USB expansion cards to the PC and connected one R200 for each of the cards or the PC mother board:
Inateck Superspeed 4 Ports PCI-E to USB 3.0 Expansion Card - Interface USB 3.0 4-Port Express Card Desktop with 15 pin SATA Power Connector (https://www.amazon.co.jp/dp/B00B6ZCNGM)
The expansion card is highly evaluated for its signal stability. Also, the external power supply may be important to provide the enough power for the "active" USB extension cable.

Camera setup

Fix cameras surrounding the recording field. Below some exmaples are shown:



[Fig 1-2: A simple setup for prototyping]



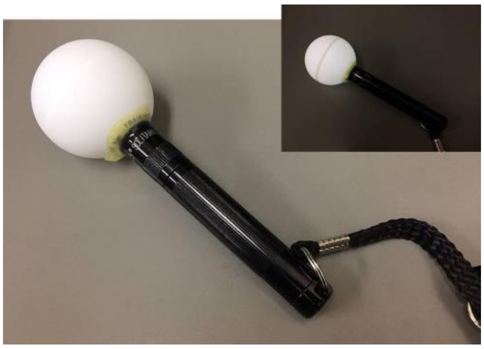
[Fig 1-3: A more sophisticated setup. A universal head was clamped to a pole fixed at the floor and the ceiling. R200 was fixed to the universal head with a handmade adaptor which was made by gluing a 1/4-inch nut to a metal clip. The green background is not necessary for normal recording.]

IMPORTANT NOTE:

+ The depth sensor of the camera measures the depth by emitting infrared (IR) light and capturing the IR reflection to objects (see Chapter 10 for the detail of the mechanism of the camera). So, two cameras should not be placed face-to-face, to prevent the infrared sensor from saturating because of directly received infrared light from the other. Similarly, shiny surfaces should not be placed face-to-face with the camera. Thus, we recommend making the cameras be slightly tilted downward (or upward) from the horizontal plane.

+ The depth sensor of R200 can't detect the depth < 40 cm. The horizontal and vertical field of view of R200 is 59 ± 5 and 46 ± 5 degrees, respectively. These constraints should also be considered in camera positioning.

Other



[Fig 1-4]

To calibrate relative positions between cameras, you need a special light pointer shown in Fig 1-4. This can be easily made by gluing a white ping-pong ball (diameter = 40 mm) on the tip of a white LED flash light with epoxy. The ping-pong ball works as a white light diffuser and a IR reflector (Fig 1-4 inset). By recording the trace of the pointer with each camera simultaneously, the relative positions between cameras are calibrated (see Chapter 4 for the detail of the calibration). We could get good results with the following LED flashlight:
MagLite Solitaire LED 1AAA Flashlight (https://www.amazon.com/dp/B00BIG5JQK/)

Chapter 2: Software preparation

Installing 3rd party libraries

3DTracker utilizes functions of many free software libraries. Please download and install the software below (you can also find the links in download section of our website: http://3dtracker.org/downloads.html):

1. OpenCV 3.2.0 (a library for image processing):
https://sourceforge.net/projects/opencvlibrary/files/opencv-win/3.2.0/opencv-3.2.0-vc14.exe/download
2. PCL 1.8.0 (a library for point cloud processing):
http://3dtracker.org/download/PCL-1.8.0-AllInOne-msvc2015-win64.exe
3. Visual C++ 2015 redistributable (a library for running the software made with Visual Studio C++ 2015)
https://www.microsoft.com/en-us/download/details.aspx?id=48145
4. Kinect for Windows Runtime v1.8 (a library for Kinect v1 camera; for backward compatibility)
https://www.microsoft.com/en-us/download/details.aspx?id=40277

# DO NOT install R200 driver provided from Intel. 3DTracker works without the driver. Further, the driver update firmware of R200 and some people reported that this firmware update may be problematic (see
https://github.com/IntelRealSense/librealsense/issues/163;
https://github.com/IntelRealSense/librealsense/issues/429).

After installing the above libraries, add the following folder to the environment variable "PATH"*:

C:¥opencv320¥build¥x64¥vc14¥bin (change "C:¥opencv320", to your install directory)
C:¥Program Files¥PCL 1.8.0¥bin
C:¥Program Files¥OpenNI2¥Redist

*How to add directories to “PATH”:
https://www.computerhope.com/issues/ch000549.htm

Installing main programs of 3DTracker

Download the prebuild version of 3DTracker-FAB from:
http://3dtracker.org/download/3DTrackerFAB_prebuild_v20171109.zip

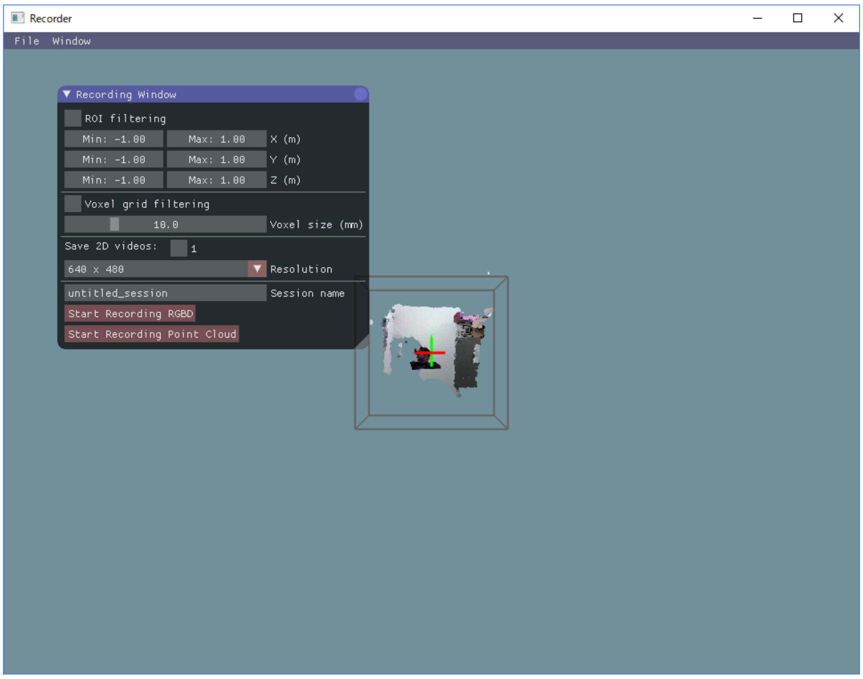
Extract the zip file. The extracted folder contains the software and data storage folder ("data" Fig 2-1). Because recorded data will be initially saved in the "data" folder, it is recommended to save and use the extracted folder in large HDD.

Chapter 3: Recoder usage

The Recorder is used for calibration of the multiple-camera setup and data acquisition.

Starting the Recorder

Connect the cameras to PC and start Recorder.exe. If Recorder.exe cannot start, the cameras may not be recognized well (see Chapter 7 for the trouble shooting).
Once the Recorder start correctly, the following screen will appear.



□ ROI filtering

Min: -1.00 Max: 1.00 X (m)

Min: -1.00 Max: 1.00 Y (m)

Min: -1.00 Max: 1.00 Z (m)

□ Voxel grid filtering

10.0 Voxel size (mm)

Save 2D videos: □ 1

640 x 480 Resolution

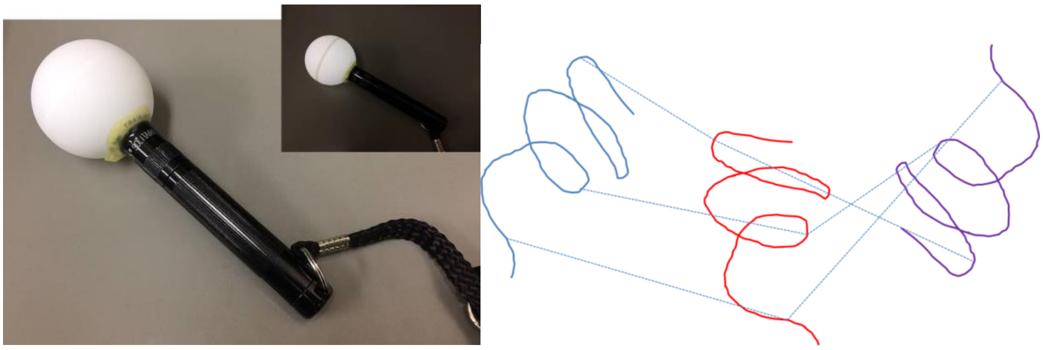
untitled_session Session name

[Fig 3-1]

There are a menu-bar and a 3D view where the camera images are projected as a point cloud (set of 3D points), on the top and in the other area in the screen, respectively. The 3D view can be controlled with the right button (R) dragging of a mouse, as follows: R drag -> Rotate the view; R drag + Ctrl key -> Zoom in and out; R drag + Shift key -> translate the view. Of the axes shown in the 3D view, the red, green and blue axes represent X, Y, and Z axes, respectively. Black small windows with various tools are shown overlapping on the 3D view. The different windows can be displayed from "Menu bar > Window".

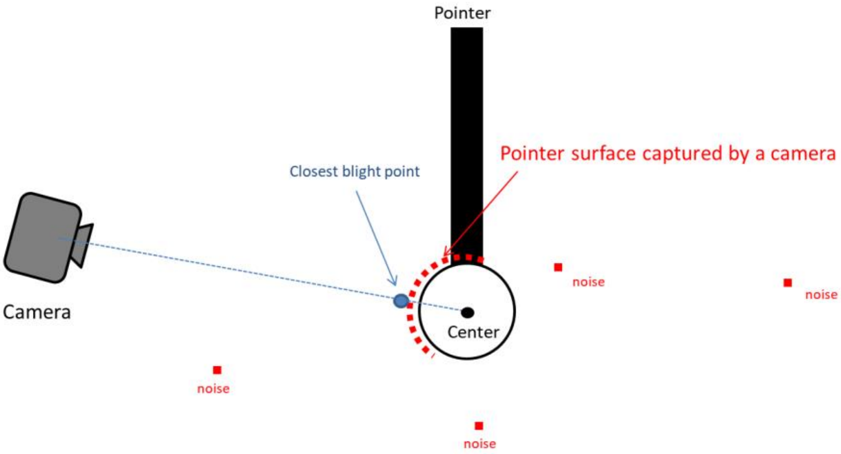
Camera calibration and configuration

If you look at the point cloud just after starting the Recorder, you will see that the points from different cameras are not correctly aligned (to more clearly discriminate between the points from different cameras, you can color the points by enabling "Mark cameras with different colors" in "Point cloud appearance" window). The misalignment is because of the lack of information of the relative positions of cameras. The Recorder can estimate the relative positions through semi-automatic calibration, as follows. In the calibration, user scans the recording space with the light pointer and the Recorder track the pointer. Then, the recorder calculates the camera relative positions to fit the trajectories observed from different cameras.



[Fig 3-2: left, the light pointer (see Chapter 2); right, the schematic showing the mechanism. the lines with different colors represent the pointer trajectories observed from different cameras]

The Recorder estimates the center of the pointer with the closest blight point from a camera and a radius of the ping-pong ball, as shown in Fig 3-3.



[Fig 3-3]

The protocol of the calibration is described below (see also a video:

https://youtu.be/04JgLTaYI2c).

Display the "Calibration window" from the "Window" menu. "Color filtering" setting section is used for detecting blight points which belong to the pointer. Find the good brightness threshold (the value of "Min Brightness" slider) with enabling "Test" check box. Usually the highest value (255) will work well for the enough blight pointer.
The "Outliner removal" setting section below the "Color filtering" section are used for removing small noise (i.e., small clusters of blight points because of reflection of room lights, etc.; Fig 3-3). For detail of the parameters of the Outlier removal, check here (http://pointclouds.org/documentation/tutorials/statistical_outlier.php). Usually changing only "Threshold" parameter is fine. You can also check the effect by enabling "Test" check box.
After adjusting filter setting for the pointer detection, click "Start pointer tracking" button on the bottom in the "Calibration" window. Then slowly move the pointer in the recording space to draw a trajectory for calibration (a spiral trajectory is recommended; Note that the pointer should be in the views of all the cameras. To check a view from each camera, display "Camera Monitor" window).
After drawing the trajectory, click "Stop pointer tracking" button, and filter the trajectories to remove noise in the trajectories in the following window. Then, click "Use the traces for calibration" to calculate the relative camera positions.
If the above operation has successfully finished, the point clouds from different cameras should be perfectly aligned. Confirm it by coloring the points by enabling "Mark cameras with different colors" in "Point cloud appearance" window. if the point clouds are not aligned well, repeat the above operation until you get a good result. Putting some objects (box, cylinder, etc.) will help to find such misalignment.

The above calibration using the pointer is only for the relative positions between cameras. So, the absolute positions of one of the cameras (the reference camera) should be input manually. Move sliders in "Reference camera position & pose" section on the top of the "Calibration" window. Set X axis rotation, Y axis rotation, Z axis rotation, and translation, RESPECTIVELY (e.g., rotation after translation make it hard to adjust the position and rotation; see also the video). When you set the absolute positions, make sure that Y axes corresponds to the vertical axis, because the Tracker assumes it for fitting. Lastly, set the region of interest (ROI) with the sliders on the top of the "Recording" window. You can also input the value directly by click + Ctrl key on slider.

A same setting is expected to be useful for the other recording sessions in a same apparatus (unless the camera won't be moved). To save and load the setting, use "File menu > save config" and "File menu > load config", respectively.

The calibration is easy to do, once you find tips for it. In our experience, it sometimes takes time for a person to get tips for each recording environment. See also Chapter 7 for the trouble shooting for the calibration.

Reference for the semi-automatic calibration:
T Svoboda, D Martinec, T Pajdla (2005) A Convenient Multi-Camera Self-Calibration for Virtual Environments:
http://cmp.felk.cvut.cz/ftp/articles/svoboda/Svoboda-Presence2005.pdf

Recording

To start recording, in the "Recording window", input the data file name in the "session name" text box and click "Start recording RGBD" or "Start Recording Point cloud" button. With "Start recording RGBD", raw data consists of RGB images and depth (D) images is stored. The size of the raw data is relatively big (around 300 MB/camera/min). With "Start Recording Point cloud", the point cloud data captured from the cameras is stored. Combining with the "ROI filtering" which remove the points outside the ROI and "Voxel grid filtering" which down sample the points, the size of the point cloud data become much smaller than that of the raw RGBD data. Use 5 and 10 mm for the voxel size for the voxel grid filtering for mice and rats, respectively. By enabling the "save 2D videos" check boxes for the cameras, you can also save the 2D RGB video, which is useful for tracking process (see Chapter 5), preliminary visual observation of behavior, etc. If you choose recording the raw data (RGBD), you can generate the 2D video offline. So, in this case, it is recommended not to check those boxes to reduce the computational load during recording.

IMPORTANT NOTE:
if you use the session name that has used previously, the original data will be overwritten.

After start recording, 3D view is disabled and RGB image is shown instead, to reduce computational load during recording. Click "Stop" button to stop recording. The recorded data is stored in the "data" folder in the folder containing the Recorder app ("Recorder.exe"). In the "data" folder, the recorded data is stored in the folder named as the "session name" that you have input before recording. The recorded data folder contains many files, each file contains data of different type and/or camera (see Chapter 9 Appendix for the detail of the output files).

Chapter 4: Preprocessor usage

The Preprocessor used for preprocessing the recorded data to generate merged 3D point clouds, down sample (removing overlapping points), calculate the surface normal vector at each point, and etc. The preprocessing is necessary for the pose estimation in the Tracker (Chapter 5).

When you start the Preprocessor, the app asks to select a folder. If you select a folder of a session, the app processes the session data. If you select a folder containing the multiple folders of different sessions, the app process all the data in the folder (the app does not search the data in more than one level below the selected folder; the data already preprocessed previously is skipped by default).

The Preprocessor is the app which is designed to be called from the command prompt, and preprocessing parameters are set by command line options. Because the use of command prompt is a bit complex, we have prepared following batch files which call the Preprocessor with predefined parameters:

- Preproc_rat.bat: Use this for rats （down sampling voxel size = 10 mm）

- Preproc_rat(f).bat: Same as the one above but no skip for the data already preprocessed previously.

- Preproc_mice.bat: Use this for mice （down sampling voxel size = 5 mm）

- Preproc_mice(f).bat: Same as the one above but no skip for the data already preprocessed previously.

The command line options [Advanced]

Example:

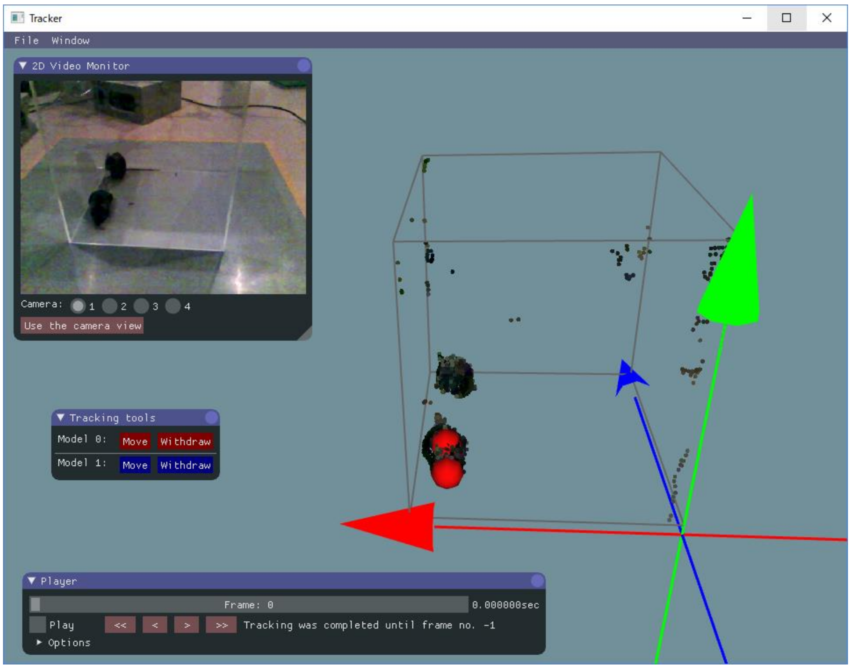
```
> preprocessor.exe -f -2dvid -vsize 0.01 -camuse 0011
```

-f: no skip for the data already preprocessed previously.
-2dvid: generate 2D video files from RGBD data.
-vsize: down sampling voxel size (in meter)
-camuse: select cameras for merging point clouds. 0=not use, 1=use. In the above example, the merged point cloud generated by using images from the camera no. 3 and 4.

Chapter 5: Tracker usage

Starting the Tracker

Start the Tracker and select the xml file in the data folder that you want to analyze, then the following screen will appear.



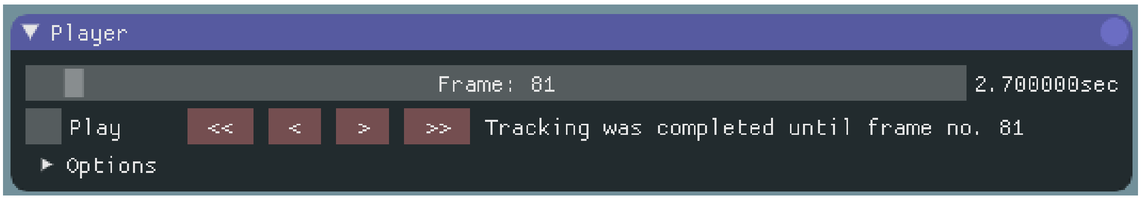
Frame: 0 0.000000sec

Tracking was completed until frame no. -1

[Fig 5-1]

Components in the screen of the Tracker are like those of the Recorder, i.e., a menu bar, a 3D view for point clouds, and tool windows. In addition, skeletal models represented by connected spheres are shown in the Tracker. The 3D view can be controlled with the right button (R) dragging of a mouse, as follows: R drag -> Rotate the view; R drag + Ctrl key -> Zoom in and out; R drag + Shift key -> translate the view. The tool windows can be displayed from "Menu bar > Window".

The following "Player window" is used for searching a video frame and fitting execution.



Frame: 81 2.700000sec

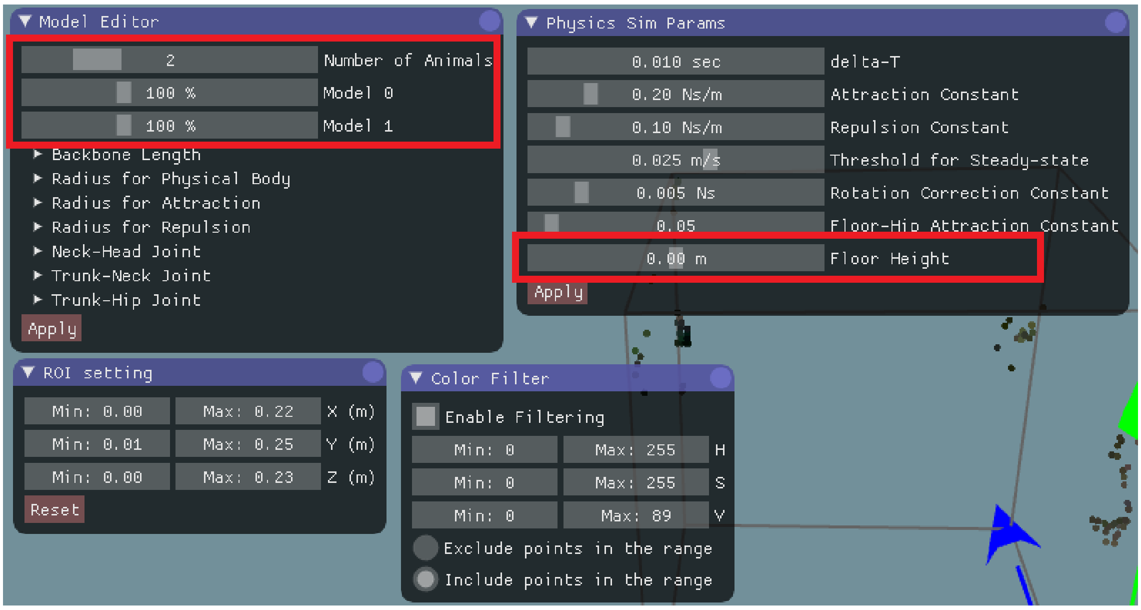
Tracking was completed until frame no. 81

[Fig 5-2]

The currently shown video frame is informed with the slider on the top of the window (frame id and time is shown). You can move the slider to search a video frame. The "Play" check box start the playback. During the playback fitting to the skeleton models (pose estimation) is also performed for the frames that have not been estimated (For the frames that have been estimated already, the result of the previous fitting is shown). The progress of the fitting process is displayed as "Tracking has been completed...". When you start the fitting process, start from the first frame (frame: 0) and progress in the order. DO NOT skip frames during the fitting, otherwise the skipping causes the malfunction in the fitting process (see below for detail of the fitting).

Importing and setting parameters

Before starting the fitting, you need to set some parameters for the process. The default parameters (e.g., dimensions of a skeletal model) are optimized for rats. So to analyze mice, import a predefined parameters for mice (e.g., Track_params_mouse_v20171104.xml) from "File Menu > Import params".
The parameters can be displayed and modified with "Model Editor window", "Physics sim params window", "ROI setting window", and "Color filter window".



Model Editor
2 Number of Animals
100 % Model 0
100 % Model 1
▸ Backbone Length
▸ Radius for Physical Body
▸ Radius for Attraction
▸ Radius for Repulsion
▸ Neck-Head Joint
▸ Trunk-Neck Joint
▸ Trunk-Hip Joint
Apply

Physics Sim Params
0.010 sec delta-T
0.20 Ns/m Attraction Constant
0.10 Ns/m Repulsion Constant
0.025 m/s Threshold for Steady-state
0.005 Ns Rotation Correction Constant
0.05 Floor-Hip Attraction Constant
0.00 m Floor Height
Apply

ROI setting
Min: 0.00 Max: 0.22 X (m)
Min: 0.01 Max: 0.25 Y (m)
Min: 0.00 Max: 0.23 Z (m)
Reset

Color Filter
□ Enable Filtering
Min: 0 Max: 255 H
Min: 0 Max: 255 S
Min: 0 Max: 89 V
○ Exclude points in the range
● Include points in the range

[Fig 5-3]

"Model Editor window" is for the parameters about the skeletal models. The slider on the top define the number of animals (skeletal models). The scale of an animal can change depending on their age, sex, etc. So, you may need to change the scales using the other sliders in the red rectangle in Fig 5-3. Detailed parameters outside the red rectangle (e.g., "Backbone Length", "Radius for Physical Body", etc.) are not necessary to change usually. After changing the parameters in "Model Editor window", click "Apply" button on the bottom, to apply the change.

"Physics sim params window" is for the parameters for fitting algorithm (the window is named like that because the fitting algorithm utilize a physics simulation). The most parameters here is also not necessary to change usually, but confirm that the "Floor height" indicates the exact height of the floor where animals were placed. The parameter is important for constraining the model movement, and the height may change depending on the environment and/or calibration.

The detail of the above parameters is described in Chapter 9 Appendix.

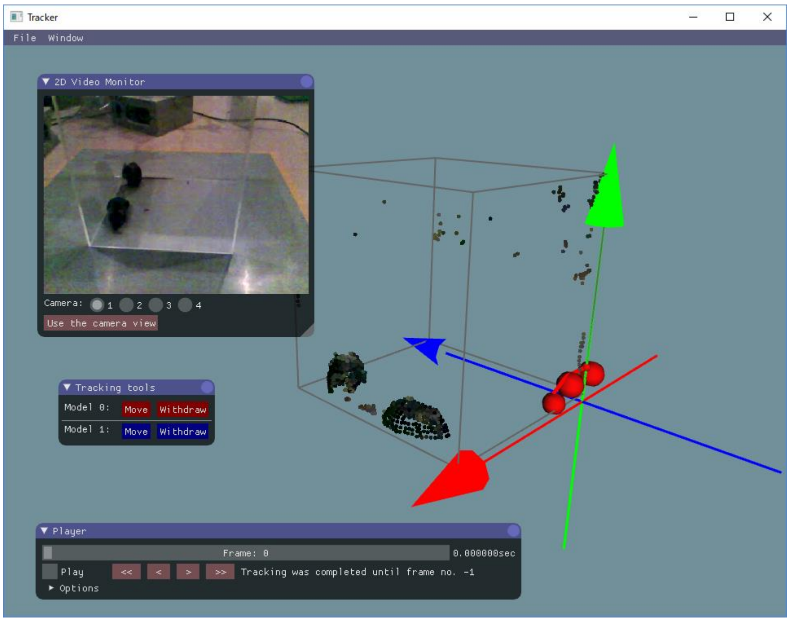
"ROI setting window" and "Color filter window" can be used to remove the points of non-animal objects. Removing such points will help successful fitting to the skeleton models to the point cloud. In the color filtering, set the range of colors to include/exclude in the HSV color space (https://en.wikipedia.org/wiki/HSL_and_HSV).

These parameters could be common if you use the same experimental system (same strain, age, apparatus, etc.). You can re-use the parameter by exporting and importing it from "File menu > Export params" and "File menu > import params", respectively.

Fitting

After finishing setting the parameters, start the fitting. Note that you cannot change the parameters in the middle of the fitting. If you change, the fitting result will be initialized.

Go to the first frame by moving the top slider in "Player window".



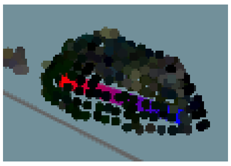
Frame: 0 0.000000sec

Tracking was completed until frame no. -1

[Fig 5-4]

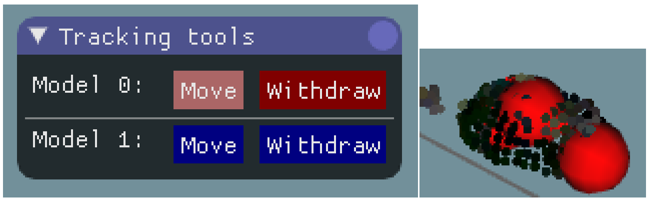
In the first frame, manually set the initial position of the skeletal models. With shiftkey pressed, click two points around a subject's hip and head in the order. You will see a line like the one in the figure below.

# If it is hard to know the direction of the subject in the 3D view, try using "2D video monitor window" (you need to have 2D videos to use this function; see Chapter 3).



[Fig 5-5]

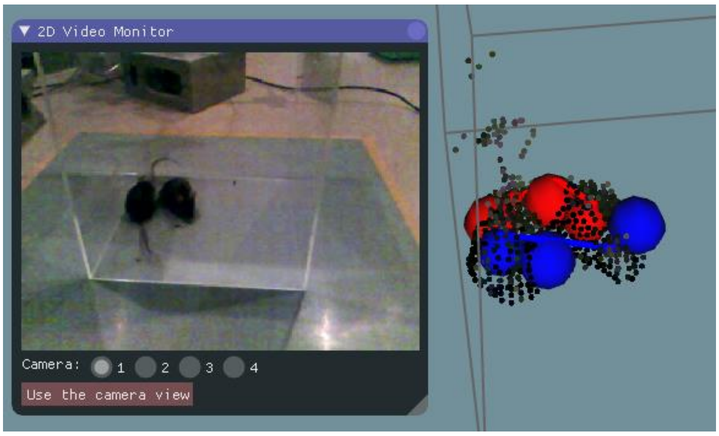
Then, click "Move" button in "Tracking tools window" to move the skeleton model to the place (Fig 5-6).



[Fig 5-6]

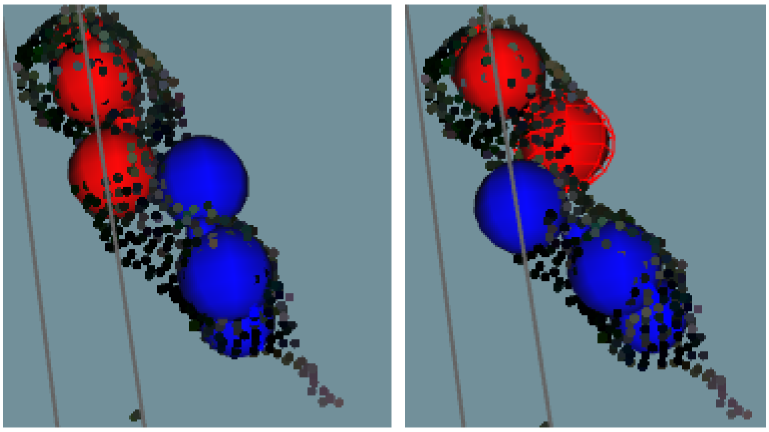
The model is fit automatically from the initial position. If you find the model is not fit perfectly, manually adjust it by dragging body parts (sphere) of the model with the left mouse button.

After placing all the skeleton models, start the fitting process by enabling "Play" check box in the "Player window". You can see the models follow the 3D image (point cloud) of animals. The positions shown during the play are the results of the fitting process. As fitting progresses, when you find an error in fitting (e.g., big part of a model is outside the 3D image, reversed direction, swapping between animals (Fig 5-7)), pause the fitting by disabling the "Play" check box.



[Fig 5-7: an example of the swapping error]

Then, by clicking "<<", "<", ">", and ">>" buttons in the "Player window", go back to the frame when the models start to get the error (Fig 5-8, left). Then, manually adjust the models by dragging body parts (Fig 5-8, right) and re-start fitting from that frame by enabling "Play" check box.



[Fig 5-8]

When you find an error again, pause and fix it in the same way. Repeat this and progress fitting to the last frame in the order.

The Tracker has many functions to support this semi-automatic fitting process:
+ The speed of playback can be changed in "Option" in the "Player window".
+ "Play" check box can be switched by pressing the space key.
+ Rotating mouse wheel during pausing change the current frame, according to the wheel rotating direction.
+ Rotating mouse wheel during playback changes the speed of playback.
+ "Use the camera view" button in the "2D video monitor window" makes the views of the 3D view and 2D video similar, making it easy to compare between the 2D and 3D image.

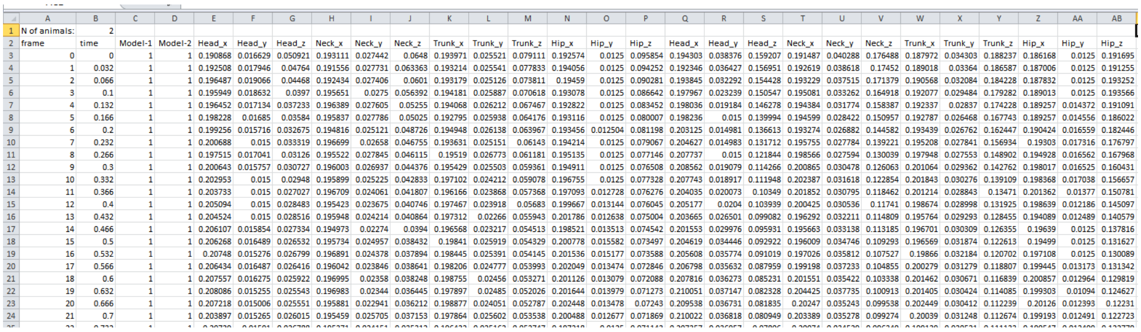
IMPORTANT NOTE:
Fitting process must be done in the order without skipping frames. Do not skip frames during the fitting, otherwise the skipping causes the malfunction in the fitting process. Even if you don't have any animal in the first frame, start from the first frame, not from the frame the animal appears. For the fitting during absence of the animals, you can withdraw the model with "Withdraw" button in "Tracking tools window".
Note that fitting process will be skipped if you move the frame using the slider on the top of the "Player window" or "<<", "<", ">", or ">>" button below. When you restart the fitting after the manual correction, make sure that it restarts from the frame that you have corrected (check the progress of fitting in "Tracking has been completed..." display).

Saving and Exporting the fitting result

Once you finished fitting, you can re-check the result by playback or moving the slider in the "Player window". The fitting result won't be changed just by replaying the frames.

You can save the fitting result from "File menu > Save result". The result will be automatically loaded when you open the data in the next time. You can also reset the result to the one saved last time from "File menu > Load result". These save/load function is also useful to interrupt the work.

For the data analysis on the trajectories of body parts, you can export the csv format file (Fig 5-9) from "File menu > Export result".



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	N of animals:	2																										
2	frame	time	Model-1	Model-2	Head_x	Head_y	Head_z	Neck_x	Neck_y	Neck_z	Trunk_x	Trunk_y	Trunk_z	Hip_x	Hip_y	Hip_z	Head_x	Head_y	Head_z	Neck_x	Neck_y	Neck_z	Trunk_x	Trunk_y	Trunk_z	Hip_x	Hip_y	Hip_z
3	0	0	1	1	0.190868	0.016629	0.050921	0.193111	0.027442	0.0648	0.193971	0.025521	0.079111	0.192574	0.0125	0.095854	0.194303	0.038376	0.159207	0.191487	0.040288	0.176488	0.187972	0.034303	0.188237	0.186168	0.0125	0.191695
4	1	0.032	1	1	0.192508	0.017946	0.04764	0.191556	0.027731	0.063363	0.193214	0.025541	0.077833	0.194056	0.0125	0.094252	0.192346	0.036427	0.156951	0.192619	0.038618	0.17452	0.189018	0.03364	0.186587	0.187006	0.0125	0.191255
5	2	0.066	1	1	0.196487	0.019066	0.04468	0.192434	0.027406	0.0601	0.193179	0.025126	0.073811	0.19459	0.0125	0.090281	0.193845	0.032292	0.154428	0.193229	0.037515	0.171379	0.190568	0.032084	0.184228	0.187832	0.0125	0.193252
6	3	0.1	1	1	0.195949	0.018632	0.0397	0.195651	0.0275	0.056392	0.194181	0.025887	0.070618	0.193078	0.0125	0.086642	0.197967	0.023239	0.150547	0.195081	0.033262	0.164918	0.192077	0.029484	0.179282	0.189013	0.0125	0.193566
7	4	0.132	1	1	0.196452	0.017134	0.037233	0.196389	0.027605	0.05255	0.194068	0.026212	0.067467	0.192822	0.0125	0.083452	0.198036	0.019184	0.146278	0.194384	0.031774	0.158387	0.192337	0.02837	0.174228	0.189257	0.014372	0.191091
8	5	0.166	1	1	0.198228	0.01685	0.03584	0.195837	0.027786	0.05025	0.192795	0.025938	0.064176	0.193116	0.0125	0.080007	0.198236	0.015	0.139994	0.194599	0.028422	0.150957	0.192787	0.026468	0.167743	0.189257	0.014556	0.186022
9	6	0.2	1	1	0.199256	0.015716	0.032675	0.194816	0.025121	0.048726	0.194948	0.026138	0.063967	0.193456	0.012504	0.081198	0.203125	0.014981	0.136613	0.193274	0.026882	0.144582	0.193439	0.026762	0.162447	0.190424	0.016559	0.182446
10	7	0.232	1	1	0.200688	0.015	0.033319	0.196699	0.02658	0.046755	0.193631	0.025151	0.06143	0.194214	0.0125	0.079067	0.204627	0.014983	0.131712	0.195755	0.027784	0.139221	0.195208	0.027841	0.156934	0.19303	0.017316	0.176797
11	8	0.266	1	1	0.197515	0.017041	0.03126	0.195522	0.027845	0.046115	0.19519	0.026773	0.061181	0.195135	0.0125	0.077146	0.207737	0.015	0.121844	0.198566	0.027594	0.130039	0.197948	0.027553	0.148902	0.194928	0.016562	0.167968
12	9	0.3	1	1	0.200643	0.015757	0.030727	0.196003	0.026937	0.044376	0.195429	0.025503	0.059361	0.194911	0.0125	0.076508	0.208562	0.019079	0.114266	0.200865	0.030478	0.126063	0.201064	0.029362	0.142762	0.198017	0.016525	0.160431
13	10	0.332	1	1	0.202953	0.015	0.02948	0.195899	0.025225	0.042833	0.197102	0.024212	0.059078	0.196755	0.0125	0.077328	0.207743	0.018917	0.111948	0.202387	0.031618	0.122854	0.201843	0.030276	0.139109	0.198368	0.017038	0.156657
14	11	0.366	1	1	0.203733	0.015	0.027027	0.196709	0.024061	0.041807	0.196166	0.023868	0.057368	0.197093	0.012728	0.076276	0.204035	0.020073	0.10349	0.201852	0.030795	0.118462	0.201214	0.028843	0.13471	0.201362	0.01377	0.150781
15	12	0.4	1	1	0.205094	0.015	0.028483	0.195423	0.023675	0.040746	0.197467	0.023918	0.05683	0.199667	0.013144	0.076045	0.205177	0.0204	0.103939	0.200425	0.030536	0.11741	0.198674	0.028998	0.131925	0.198639	0.012186	0.145097
16	13	0.432	1	1	0.204524	0.015	0.028516	0.195948	0.024214	0.040864	0.197312	0.02266	0.055943	0.201786	0.012638	0.075004	0.203665	0.026501	0.099082	0.196292	0.032211	0.114809	0.195764	0.029293	0.128455	0.194089	0.012489	0.140579
17	14	0.466	1	1	0.206107	0.015854	0.027334	0.194973	0.02274	0.0394	0.196568	0.023217	0.054513	0.198521	0.013513	0.074542	0.201553	0.029976	0.095931	0.195663	0.033138	0.113185	0.196701	0.030309	0.126355	0.19639	0.0125	0.137816
18	15	0.5	1	1	0.206268	0.016489	0.026532	0.195734	0.024957	0.038432	0.19841	0.025919	0.054329	0.200778	0.015582	0.073497	0.204619	0.034446	0.092922	0.196009	0.034746	0.109293	0.196569	0.031874	0.122613	0.19499	0.0125	0.131627
19	16	0.532	1	1	0.20748	0.015276	0.026799	0.196891	0.024378	0.037894	0.198445	0.025391	0.054145	0.201536	0.015177	0.073588	0.205608	0.035774	0.091019	0.197026	0.035812	0.107527	0.19866	0.032184	0.120702	0.197108	0.0125	0.130089
20	17	0.566	1	1	0.206434	0.016487	0.026416	0.196042	0.023846	0.038641	0.198206	0.024777	0.053993	0.202049	0.013474	0.072846	0.206798	0.035632	0.087959	0.199198	0.037233	0.104855	0.200279	0.031279	0.118807	0.199445	0.013173	0.131342
21	18	0.6	1	1	0.207557	0.016275	0.025922	0.196995	0.02358	0.038248	0.198755	0.02456	0.053271	0.201126	0.013079	0.072088	0.207816	0.036273	0.085231	0.201551	0.035422	0.103338	0.201462	0.030671	0.116839	0.200857	0.012964	0.129819
22	19	0.632	1	1	0.208086	0.015255	0.025543	0.196983	0.02344	0.036445	0.197897	0.02485	0.052026	0.201644	0.013979	0.071273	0.210051	0.037147	0.082328	0.204425	0.037735	0.100913	0.201405	0.030424	0.114085	0.199303	0.01094	0.124627
23	20	0.666	1	1	0.207218	0.015006	0.025551	0.195881	0.022941	0.036212	0.198877	0.024051	0.052787	0.202448	0.013478	0.07243	0.209538	0.036731	0.081835	0.20247	0.035243	0.099538	0.202449	0.030412	0.112239	0.20126	0.012393	0.12231
24	21	0.7	1	1	0.203897	0.015265	0.026015	0.195459	0.025705	0.037153	0.197864	0.025602	0.053538	0.200488	0.012677	0.071869	0.210022	0.036818	0.080949	0.203389	0.035278	0.099274	0.20039	0.031248	0.112674	0.199193	0.012491	0.122723

[Fig 5-9]

Chapter 6: Other software usage

Viewer

The software is made for checking the recorded data. The Viewer is something like the Tracker without fitting functions. To open the data in Viewer, no preprocessing is necessary, so it enables to check the data without waiting for the preprocessing.

Data analysis scripts (Python / MATLAB)

The scripts will help analyzing the trajectory data (the csv file) exported from the Tracker and the original point cloud data. The scripts are in preparation. We also encourage people to share such scripts. Please contact us if you make a useful script in the Forum (https://groups.google.com/forum/#!forum/3dtracker) or from the Contact form (http://3dtracker.org/contact.html).

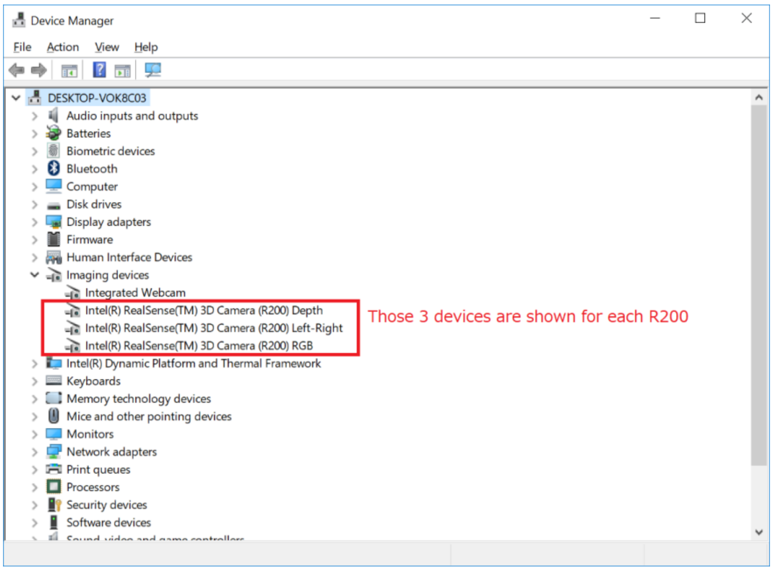
There are old MATLAB scripts that we previously published with a paper (File S1 in Matsumoto+ 2013, Plos one;
http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0078460#s6).
Although the scripts cannot be directly used with the new data format, the scripts would be a good starting point if you want to write your own scripts.

Chapter 7: Trouble shooting

If you cannot find the related topic or the solution here, ask the question in Forum (https://groups.google.com/forum/#!forum/3dtracker).

Recorder could not start (with R200 cameras)

The camera may not be recognized by the PC. Re-check the connection of the camera according to Chapter 2. If there is still the problem, check if the camera is listed in "Device manager" (Fig 7-1).

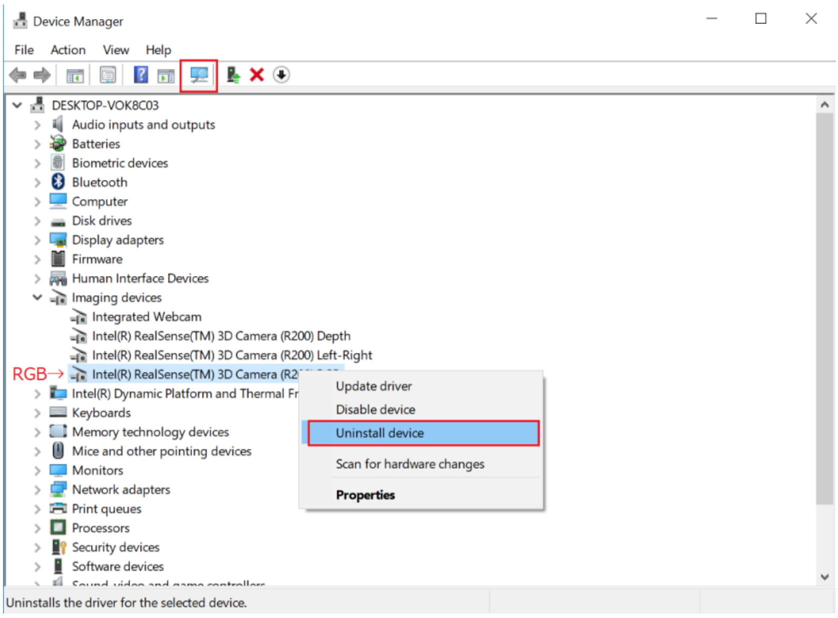


Those 3 devices are shown for each R200

[Fig 7-1]

If the camera is not listed, make sure that "USB selective suspend" is disabled (https://www.techsupportalert.com/content/how-fix-annoying-windows-usb-problem.htm).

If the camera is listed but there is still the problem, uninstall the device of RGB camera and re-install it by clicking "scan for hardware changes" icon (Fig 7-2). It is reported that such RGB camera driver error happens especially after "Windows 10 Creators Update" and that the above protocol fixes the problem (https://github.com/IntelRealSense/librealsense/issues/481).



Uninstalls the driver for the selected device.

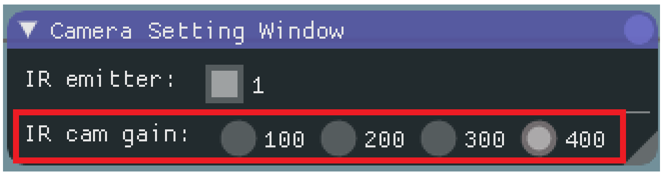
[Fig 7-2]

Calibration does not work well

We suggest trying followings:

+ Change infrared (IR) gain (R200)

If the camera is too close to the surfaces to record or number of cameras is large, the IR reflection become too strong to saturate the IR sensor. The data in the saturated part will be lost so the saturation can interfere the pointer tracking during the calibration.



Camera Setting Window

IR emitter: □ 1

IR cam gain: ○ 100 ○ 200 ○ 300 ● 400

[Fig 7-3]

Reduce "IR camera gain" in "Camera setting window" to avoid the saturation (see also Chapter 9, Appendix for the camera mechanism).

+ Check the area to scan with the pointer

Show "Camera monitor window" and make sure that the pointer is always inside the views of all the cameras. Also make sure the distance between the pointer and the camera is always more than 40 cm (the depth sensor of R200 can't detect the depth < 40 cm). Also make sure that the pointer light is not strongly reflected by the floor or the other objects.

+ Change the room illumination and filters for the pointer detection

Adjusting the "Color filter", "Outlier removal", and room illumination will improve the quality of the pointer tracking.

An object or an animal is not captured

+ Change infrared (IR) gain (R200)
see the same title section in above "Calibration does not work well"

+ Check obstacles in front of the camera (R200)

The RGB camera in R200 can be seen from outside. However, other two IR sensor and one IR emitter are behind the blue IR transmitting plastic. So make sure you have no obstacle on the whole surface of the front plastic (for the structure of R200, see: https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/realsense-camera-r200-datasheet.pdf).

+ Check the color of the object

Object surface without IR reflection cannot be captured (e.g., deep black colored plastic). In the case you need to reconsider the color of the object. Mice with black coat (e.g., C57BL/6) can be captured without any problem.

Drop frame during recording

Before you buy a faster PC, try the followings:
+ Wait enough after the PC boot (PC is often slow just after boot).
+ Change Windows setting to reduce the computational load for the other processes running in parallel (e.g., https://www.cnet.com/how-to/easy-ways-to-speed-up-windows-10/).

Menu bar invisible / mouse offset from view

The problem can happen with an old driver of Intel HD Graphics video card (https://developer.blender.org/T51521). Update the driver.

Chapter 8: Information for developers

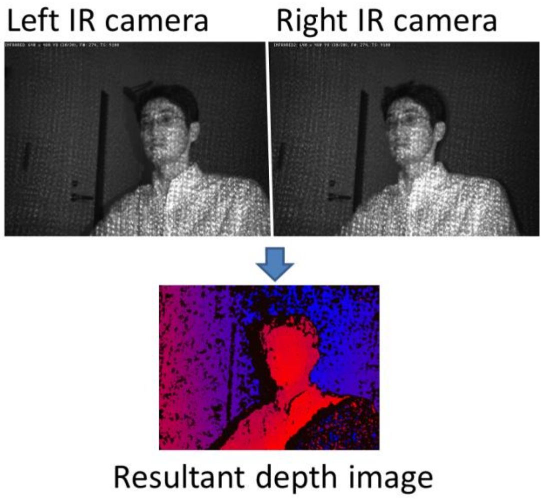
Coming soon…

Contact us in the Forum (https://groups.google.com/forum/#!forum/3dtracker).

Chapter 9: Appendix

R200 camera mechanism

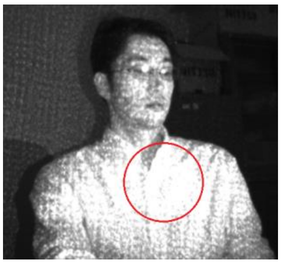
Intel realsense R200 contains one RGB camera, two IR camera, and one IR emitter. the IR emitter project a random IR pattern to the objects (Fig 9-1, upper pictures). Distances from the camera to the objects' surface are calculated using two IR camera images based on binocular disparity. As a result, a depth image (Fig 9-1, bottom picture), where each pixel in the image represent distance from the camera to the object surface, is generated.



[Fig 9-1]

In 3DTracker software, colored 3D point cloud is generated from the depth image and the RGB image captured by R200. Because R200 uses binocular disparity, the R200 cameras do not interfere each other. This is the excellent feature of R200 compared to the other 3D cameras (for example, ToF cameras, like Kinect v2, severely interfere each other).

Although basically R200 can work well with multi-camera setup, if many R200 exists or a recording object is close to a camera, some area recieve too much IR and the part of the IR camera image become saturated (red circle in Fig 9-2).



[Fig 9-2]

There is no binocular disparity in the saturated area. So the depth of the area can not be calculated. By reducing the IR camera gain (in "Camera setting window" of the Recorder app), the saturation can be avoided. For the detail of R200, see also the following datasheet:

Intel R200 data sheet:
https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/realsense-camera-r200-datasheet.pdf

Detail of the parameters in the Tracker

Coming soon…

See Matsumoto+ 2013, Plos one
(http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0078460).
Also feel free to ask questions in the Forum
(https://groups.google.com/forum/#!forum/3dtracker).

Output files

The data of a recording session saved as multiple files in the same folder. Each of the files contains data of different type and/or camera, as follows:

[Session_name].metadata.xml: metadata file, which stores recording configurations.
[Session_name].2dvideo.[N].avi: 2D RGB video of camera no. N.

[Session_name].camintrin.[N].bin: internal parameter of camera no. N.

[Session_name].rgbd.frame.[N].bin: RGBD data of camera no. N.
[Session_name].rgbd.ts.[N].txt: the timestamp of the RGBD data

[Session_name].pc.frame.[N].bin: Point cloud data of camera no. N.
[Session_name].pc.ts.[N].txt: the timestamp of the point cloud data

[Session_name].mrgpc.frame.bin: Merged point cloud data generated by Preprocessor
[Session_name].mrgpc.ts.txt: the timestamp of the merged point cloud data

[Session_name].trackparam.txt: Tracker parameters
[Session_name].trackresult.bin: Tracker result