

# 3DTrackerFAB documentation

## 3DTrackerFAB-doc

- 1. Hardware preparation
  - 1.1. Cameras
  - 1.2. Extension cables
  - 1.3. PC
  - 1.4. Camera mounting
- 2. Software preparation
  - 2.1. Installing 3rd party libraries
  - 2.2. Installing main programs of the 3DTracker-FAB
- 3. Calibrator and recorder usage
  - 3.1. Camera calibration using the Calibrator
  - 3.2. Recording 3D video using the Recorder
- 4. Preprocessor usage
- 5. Tracker usage
  - 5.1. Starting the Tracker
  - 5.2. Importing and setting parameters
  - 5.3. Fitting
  - 5.4. Saving and Exporting the fitting result
  - 5.5. Measurement tools
- 6. Other software usage
  - 6.1. Viewer
  - 6.2. Data analysis scripts (Python/MATLAB)
- 7. Trouble shooting
  - 7.1. Calibrator/Recorder does not start (with R200 cameras)
  - 7.2. Calibration does not work well
  - 7.3. Objects or animals are not captured
  - 7.4. Dropping frames during recording
  - 7.5. Menu bar invisible / mouse offset from view
- 8. Information for developers
  - 8.1. Preparing the development environment
  - 8.2. Programming Style
  - 8.3. More Information
- 9. Appendix
  - 9.1. R200 camera mechanism

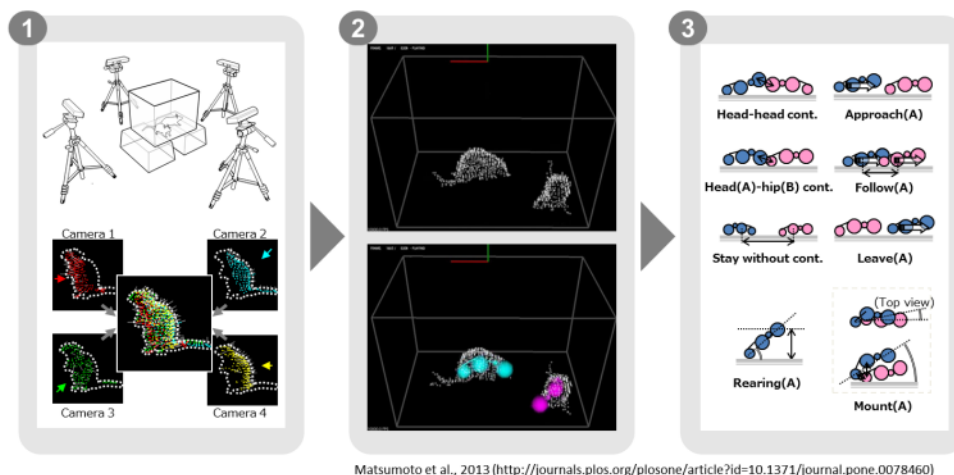
- 9.2. Details of the parameters in the Tracker
- 9.3. Output files
- 9.4. Synchronizing with other systems
- 9.5. Online tracking (under development)

## 3DTracker-FAB documentation¶

The document was written by Dr. Jumpei Matsumoto and Dr. Marcelo Aguilar-Rivera. We thank Dr. Laleh Quinn for her proof reading. *This version is still under refinement.*

## Introduction¶

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



[Fig 0-1]

First, animals 3D images are reconstructed from depth cameras (~4) images (Fig 0, left). Second, body positions are estimated by fitting skeletal models to the 3D image (Fig 0, middle). Third, several behavioral parameters are calculated based on body part trajectories (Fig 0, right).

In the current workflow, a set of software modules are used. The images from multiple cameras are calibrated and recorded, with the “Calibrator” and “Recorder” modules respectively (Chapter 3). Then, the recorded images from different cameras are processed off-line for their integration into a full 3D image, down sampling, etc., with the “Preprocessor” module (Chapter 4). Then, the skeletal models are fitted to the 3D image, with the “Tracker” module (Chapter 5). Finally, the resultant 3D trajectories (csv file format) are analyzed with Python/MATLAB scripts (Chapter 6). The present version of the 3DTracker-FAB only supports the analysis of rats and mice. Other species could be supported in the future.

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

## 1. Hardware preparation¶

### 1.1. Cameras¶

Multiple Intel Realsense R200 or D400 cameras (Fig 1-1) are required for recording with the 3DTracker system. These cameras are chosen because of their outstanding performance characterized by the lack of interference among cameras when multiple units are simultaneously capturing the same object (see [Appendix](#) for details of the camera). A R200 or D400 is composed of a color camera and a depth sensor to obtain RGB and depth images, respectively.



[Fig 1-1: Intel realsense cameras]

You can get R200 and D400 from the Intel website:

<https://click.intel.com/realsense.html>

D400 has two models: D415 and D435. The software supports both of them. For new users, we recommend using D400, because it is an improved version of R200. Both D435 and D415 are good enough for tracking with the present system. For the detailed comparison, see: <https://realsense.intel.com/compare/>

### 1.2. Extention cables¶

Given that the original USB 3.0 cable of the R200 or D400 is too short, it is necessary to extend it or to replace with a longer cable. We have confirmed the following extentions/cables work:

- **For R200:**
  - [UGREEN, USB 3.0 Active Extension Cable](#)
- **For D400:**
  - [Anker, PowerLine 10ft USB-C to USB 3.0](#)
  - [NewNex, UL-A01C12-8M](#)

### 1.3. PC¶

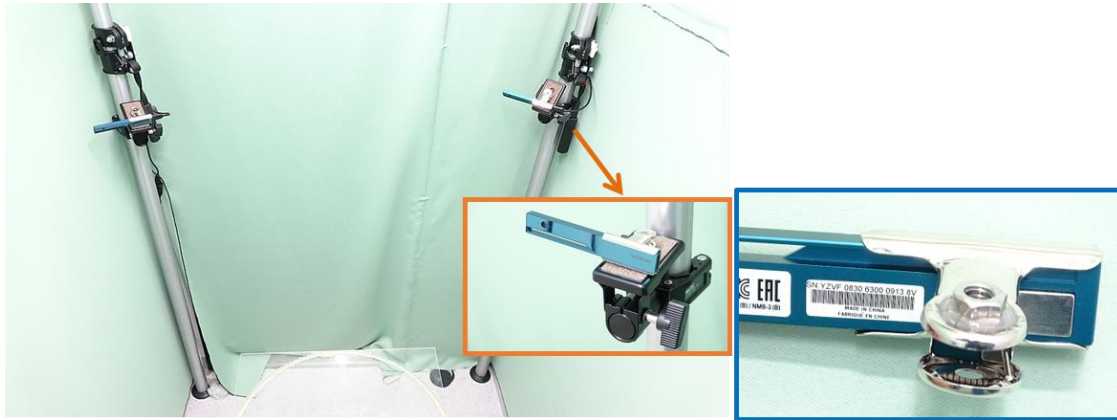
The 3DTracker (except for MATLAB/Python scripts) runs only on Windows 10 (64-bit). The CPU does not need to be very fast. However, the number of cores (number of parallel processors) in the CPU should be more than the number of cameras used, because the Recorder software processes each camera data in parallel. In addition, a large HDD should be installed in the PC, because of the big size of 3D video. The PC monitor resolution should be more than 1280x1024. We have been recording from four cameras with a HP EliteDesk 800 G2 Core i7-6700 (3.4 GHz), 8GB RAM, 500 GB + 6TB HDD, and a power unit of 400W 80PLUS.

The camera uses almost the whole transmission bandwidth of an USB 3.0 port. So, more than one camera cannot be connected to the same USB controller. For this reason, we added three expansion cards (<https://www.amazon.co.jp/dp/B00B6ZCNGM>) to get in total four USB controllers (including the one embedded in the mother board), thus having a camera connected per controller. The expansion cards have to be hooked to the PC power supply to feed enough power into the “active” USB extension cable, beside of be inserted in the mother board slots.

#### Note

We strongly recommend using the same components suggested above to get a high-quality signal. We have tried several different products and those previously described are the best ones based on our experience.

#### 1.4. Camera mounts¶



[Fig 1-2]

Example of four depth cameras setup. Each R200 camera was hooked to a post fixed to the floor and the ceiling (insert orange frame), through a custom adaptor made by gluing a 1/4-inch nut to a metal bulldog clip (insert blue frame). The main picture shows two of the four cameras.

#### Hint

The depth sensor of the camera measures the depth by emitting infrared (IR) light and capturing the IR reflection to objects (see [Appendix](#) 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 in front of a camera. Thus, we recommend placing the cameras slightly tilted downward from the horizontal plane. Also check the camera specification for 1) the minimum distance between the camera and objects for depth sensing (e.g., 40 cm for R200), and 2) the field of view. These constraints should be considered during camera setting.

## 2. Software preparation¶

### 2.1. Installing 3rd party libraries¶

The 3DTracker-FAB utilizes functions of many free software libraries. Please download and install the software below. You can download the software from our website at <http://3dtracker.org/downloads.html>

1. OpenCV 3.2.0 (library for image processing)
2. PCL 1.8.0 (library for point cloud processing)
3. Visual C++ 2015 redistributable (library for running the software that was written on Visual Studio C++ 2015)
4. Kinect for Windows Runtime v1.8 (library for Kinect v1 camera and backward compatibility)

#### Warning

**DO NOT install the R200 driver** provided by Intel because the 3DTracker doesn't need it. Also, some users have reported that updating the R200 firmware could trigger the malfunctioning of the R200. See links below:

<https://github.com/IntelRealSense/librealsense/issues/163>  
<https://github.com/IntelRealSense/librealsense/issues/429>

After installing the libraries listed above, 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

See how to add directories to the "PATH":

<https://www.computerhope.com/issues/ch000549.htm>

### 2.2. Installing main programs of the 3DTracker-FAB¶

Download the prebuilt version of the 3DTracker-FAB from our website

<http://3dtracker.org/downloads.html>

Uncompress the downloaded zip file. The extracted folder contains the software and data folder.

Hint

Because recorded data will be initially saved in the data folder, it is recommended to place the extracted data folder in a large HDD.



### 3. Calibrator and recorder usage¶

The Calibrator and the Recorder are modules for calibration and recording of the multiple-cameras setup of the 3DTracker system, respectively.

#### 3.1. Camera calibration using the Calibrator¶

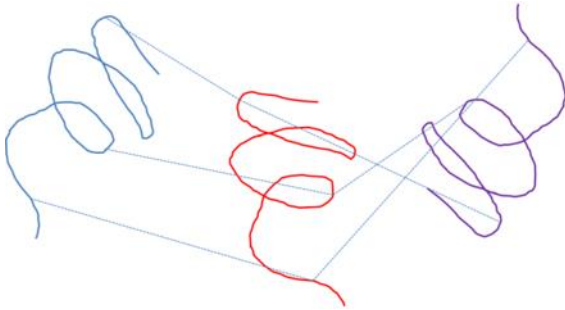
The video instruction: [https://youtu.be/IE\\_ZDVuzYC8](https://youtu.be/IE_ZDVuzYC8)

Calibration for the relative positions of the cameras are necessary for integrating images captured by different cameras and then to re-construct a full 3D image. The Calibrator is a software wizard for the multi-camera calibration.

Connect the cameras to the PC and start Calibrator.exe. If Calibrator.exe doesn't start, the cameras may have not been recognized (see [Trouble shooting](#)). In the start page (step 0) of this software wizard, you will see a 3D view display (center) where camera images are projected as a cloud of points, and a wizard window (bottom). In the 3D view, the red, green, and blue arrows correspond to X, Y, and Z axes, 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. To discriminate clearly between the points from different cameras, you can color the points by enabling "Mark cameras with different colors" in the top left window. The 3D view can be switched to the 2D view where the image of each camera is shown, by clicking the "Switch to 2D" button on the top left window. Click "Next" in the wizard window to proceed to the next step.

In the step 1, move cameras to cover the space where you want to record while checking the 2D view. Also select the reference camera (the point clouds of the other camera will be aligned to the one of the reference camera). Note that in the 2D view, camera images are ordered in left-to-right and top-to-bottom according to the camera IDs.

In the following steps (2-6), the user scans the recording space with a light pointer and records the pointer track. Then, the software calculates the relative camera positions fitting the observed trajectories recorded from the different cameras during the calibration process (Fig 3-1, left). For the pointer we recommend using incandescent MAG-LITE (e.g., <https://www.amazon.com/dp/B00002N9ET>) in the candle mode, by removing the lid covering the light bulb (Fig 3-1, right)



[Fig 3-1: left, schematic showing the mechanism. The lines with different colors represent the pointer trajectories observed simultaneously from 3 different cameras in this example; right, light pointer as a candle]

In the step 2, set the camera exposure for pointer tracking. Put the pointer in the recording space and turn off the room light. Then, set the exposure using the slider in the wizard window, to enhance the contrast between the pointer light and the background.

In the step 3, Set the brightness threshold for the pointer detection and tracking.

In the step 4, 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 view of all the cameras.

In the step 5, filter the trace (to remove some noise). You can use the temporal trimming and the outlier removal to remove the noise in the traces. For the detail of the parameters of the outlier removal, see:

[http://pointclouds.org/documentation/tutorials/statistical\\_outlier.php](http://pointclouds.org/documentation/tutorials/statistical_outlier.php).

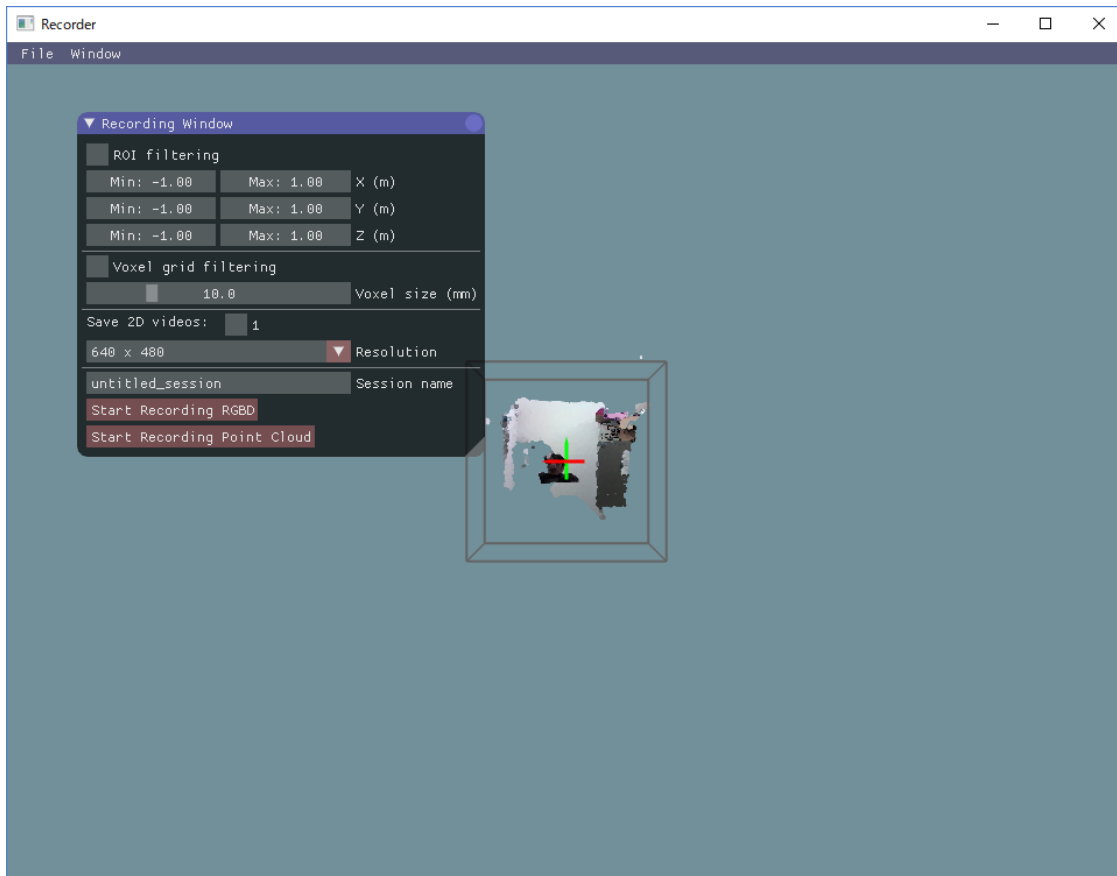
In the step 6, first, check if the point clouds from different cameras are perfectly aligned. To check the alignment, place some objects in front of the cameras and color the points by enabling “Mark cameras with different colors” in the top left window. Then, look at the corners and the overall shapes of the objects in the screen. If point clouds with different colors are not well aligned, go back to the previous steps and repeat the operation described above until a good result is achieved.

The calibration using the pointer is only for the relative positions among cameras. So, the absolute position of one of the cameras (the reference camera) should be input manually. Move the sliders in the wizard window to set the position and the rotation of the reference camera. Set X axis rotation, Y axis rotation, Z axis rotation, and translation. Rotation after translation makes it hard to adjust the position and rotation (see also the video link above). When you set the absolute position, make sure that the Y axis corresponds to the vertical axis, because the Tracker assumes it for fitting purposes.

Lastly, in the step 7, set the region of interest (ROI) with the sliders. You can also input the value directly by clicking + Ctrl key on slider. Then, click “Next” and “Save Config and Quit”, to finish the calibration.

### 3.2. Recording 3D video using the Recorder¶

Start Recorder.exe. Once the Recorder starts, the following screen will appear:



[Fig 3-2]

You will see a menu-bar, and a 3D view display which is same as the one in the Calibrator (see above). The various tool windows can be displayed from the menu tab > ‘Window’.

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 consisting of RGB images and depth (D) images is stored. The size of the raw data is big (around 300 MB/camera/min). With “Start Recording Point cloud”, the point cloud data captured from the cameras are stored. Combined with the “ROI filtering” which removes the points outside the ROI,

and “Voxel grid filtering” which down samples the points, the size of the point cloud data becomes 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 (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.

#### Note

For D400 cameras, in addition to point cloud and RGBD file format, recording with rosbag file format (<http://wiki.ros.org/rosbag>) is also supported. Using the format, you can select higher spatial and temporal resolution, although the file size will be large (the compression algorithm in rosbag format is faster but less efficient than the one in RGBD format). Click “Start Recording Bag” in the “Recording window” for the rosbag recording. Before the preprocessing (Chapter 4), convert the rosbag to RGBD format, by using Rosbag2RGBD.exe.

After starting recording, 3D view is disabled and an RGB image is shown instead, reducing the computational load during recording. Click “Stop” button to stop recording. The recorded data is stored in the “data” folder containing the Recorder app (“Recorder.exe”) in the folder named as the “session name” that you have input before recording. See Chapter 9 for details about the output files.

## 4. Preprocessor usage¶

This module is for preprocessing the recorded data, for merging the 3D point clouds, down sampling (removing overlapping points) the data, calculating the surface normal vector at each point, etc. The preprocessing is necessary for pose estimation in the Tracker (Chapter 5). When you start the Preprocessor, it asks to select a folder to processes the session data in. If you select a folder containing multiple sub-folders from different sessions, the module will process the data in these sub-folders if they don't contain additional sub-levels. This is because the app does not search more than one level below the folder selected. The data preprocessed previously is skipped by default. The Preprocessor is called from the command prompt, and the preprocessing parameters are set by command line. We have prepared the following batch files that call the Preprocessor with certain predefined parameters as follows:

- Preproc\_rat.bat: Use this for rats (down sampling voxel size = 10 mm).
- Preproc\_rat(f).bat: Same as the previous one but it does not skip the data already preprocessed.
- Preproc\_mice.bat: Use this for mice (down sampling voxel size = 5 mm).
- Preproc\_mice(f).bat: Same as Preproc\_rat(f), but for mice.

### Example of a command line (Advanced):

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

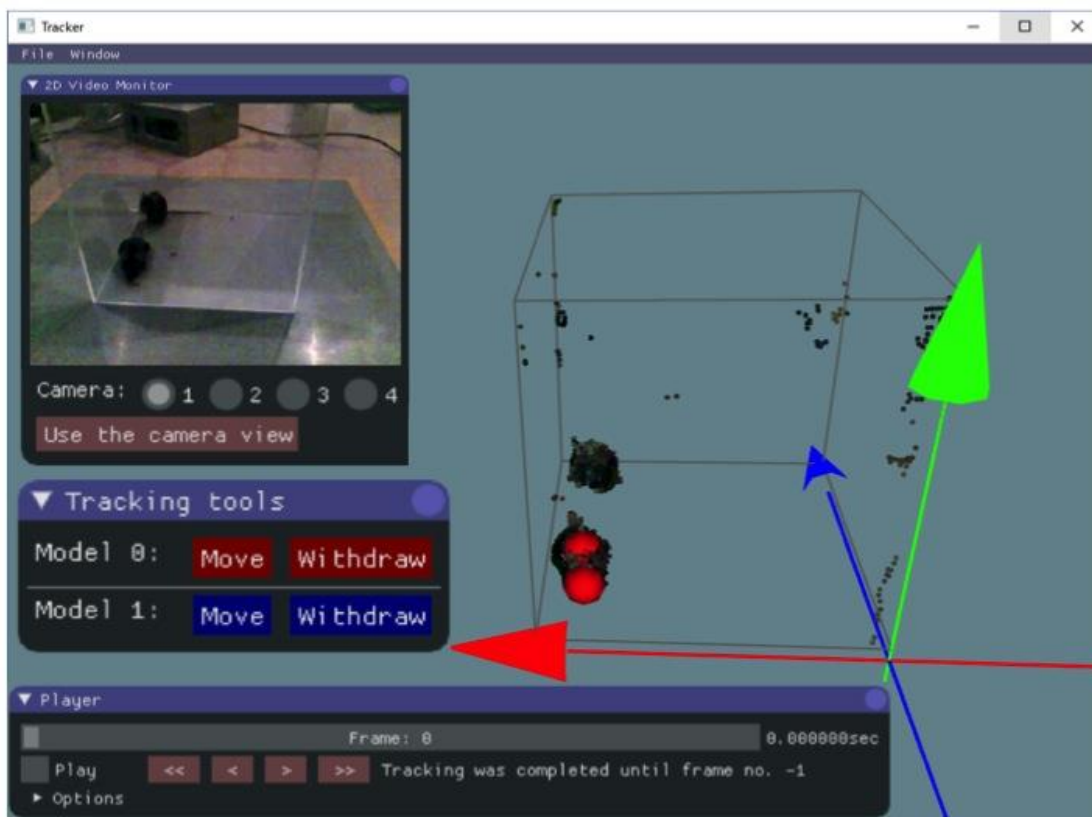
- **-f**: no skip for the data preprocessed previously
- **-2dvid**: generate 2D video files from RGBD data
- **-vsize**: down sampling voxel size (in meters)
- **-camuse**: selects cameras for the merging of the point clouds. 0=not use, 1=use.

In the example above, the merging points cloud is generated by using images from the camera #3 and #4.

## 5. Tracker usage¶

### 5.1. Starting the Tracker¶

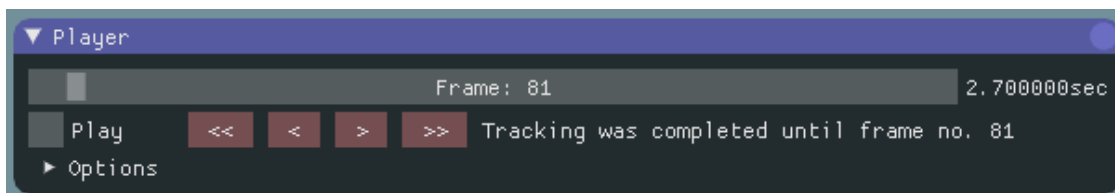
Start the Tracker and select the xml file in the data folder that you want to analyze. Then, the following screen will appear.



[Fig 5-1]

The windows on the screen are those you can launch by clicking the tabs in the menu Window. In addition, the skeletal models represented by connected spheres are shown in the Tracker screen. The 3D view can be controlled with the right button (R) by dragging the mouse as follows: R drag -> Rotate the view; R drag + Ctrl key -> Zoom in and out; R drag + Shift key -> translate the view.

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

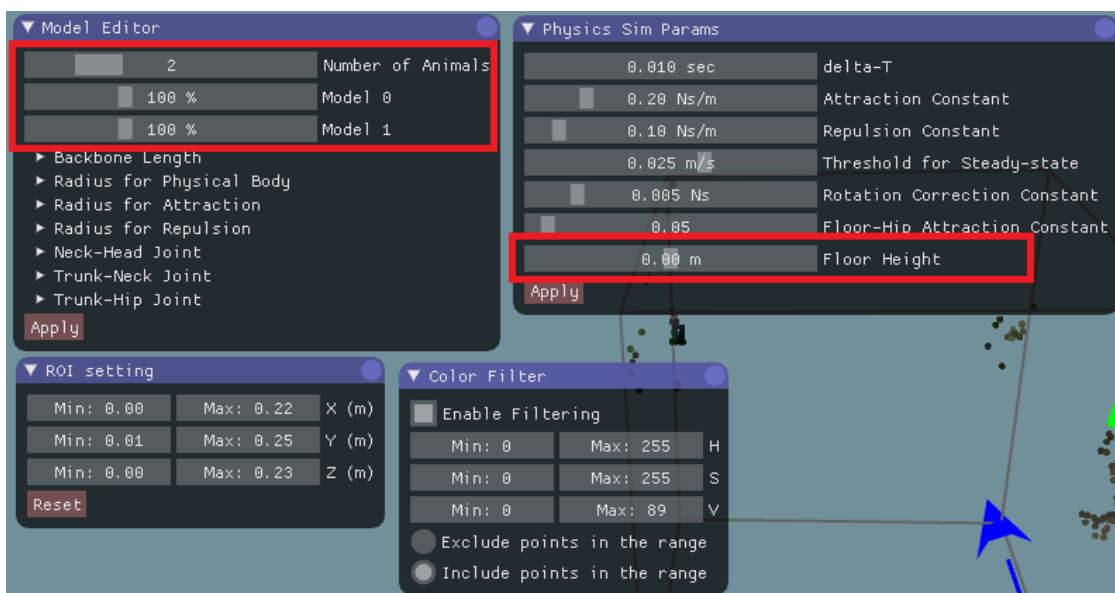


[Fig 5-2]

The numbers inside, and at the right side of the top slide bar, indicate the current # of the video frame and time, respectively. You can move the slide bar to search a specific video frame. By checking the box “Play” you can start the playback. The fitting of the skeleton models for pose estimation during playback is for the frames where it was not estimated. For the frames where it has already been estimated, the result of fitting is shown. The progress during the fitting process is displayed as “Tracking that has been completed...”. When starting the fitting process, you have to do it from the first frame and then continue with the next one. **DO NOT skip frames** during the fitting since it can result in an irreversible corrupted physics simulation.

## 5.2. 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 predefined parameters for mice (e.g., Track\_params\_mouse\_v20171104.xml) from “File Menu > Import params”. The parameters can be displayed and modified with the “Model Editor window”, “Physics sim params window”, “ROI setting window”, and “Color filter window”.



[Fig 5-3]

The “Model Editor window” is for the skeletal models. The slider on the top defines the number of animals (skeletal models). The scale of an animal can change depending on its age, sex, etc. So, you may need to change the scales using the other sliders in the red rectangle as shown in Fig 5-3. Detailed parameters outside the red rectangle (e.g., “Backbone Length”, “Radius for Physical Body”, etc.) don’t need to be changed frequently. After changing the parameters in the “Model Editor window”, click “Apply” button on the bottom, to apply the change.

The “Physics sim params window” is to parametrize the fitting algorithms, and usually it is necessary to change just some of them. The value of the “floor height” should be set to the height of the floor where the animal is standing in the global coordinate system. This parameter is important to constrain the movement of the model. The details of the parameters mentioned above are described in Chapter 9.

“ROI setting window” and “Color filter window” can be used to remove the points of non-animal objects. Removing such points will help fitting the skeleton models to the points 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](https://en.wikipedia.org/wiki/HSL_and_HSV)). These parameters could be repeatedly used for the same experimental setup regarding rodent strain, age, apparatus, etc. You can re-use the parameters by exporting and then importing them from the “File menu > Export params” and “File menu > import params”, respectively.

### 5.3. Fitting¶

After the parameters have been set, start the fitting. Note that you cannot change the parameters during the fitting, otherwise the fitting will be re-initialized. In the first frame, manually set the initial position of the skeletal models. With shift key pressed, click two points from hip to head. You will see a line like the one in the figure below.



[Fig 5-4]

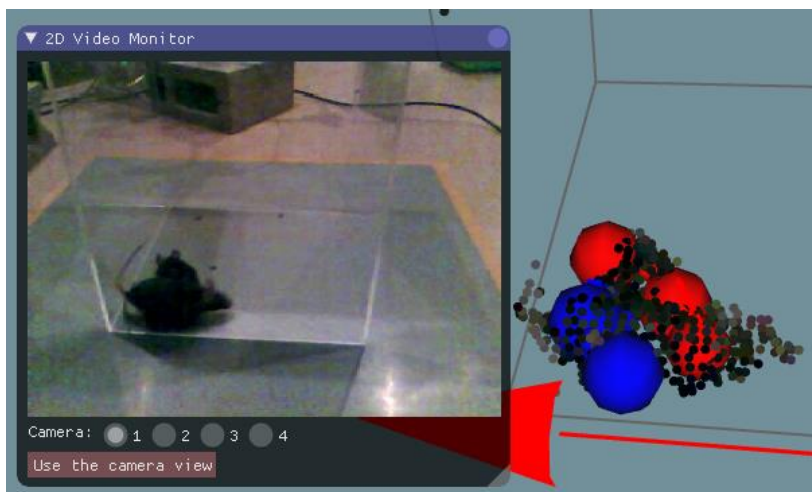
If it is hard to know the direction of the subject in a 3D view, try using “2D video monitor window” as shown in Fig 5-1. Then, click the “Move” button in “Tracking tools window” to change the skeleton model to the correct place (Fig 5-5).





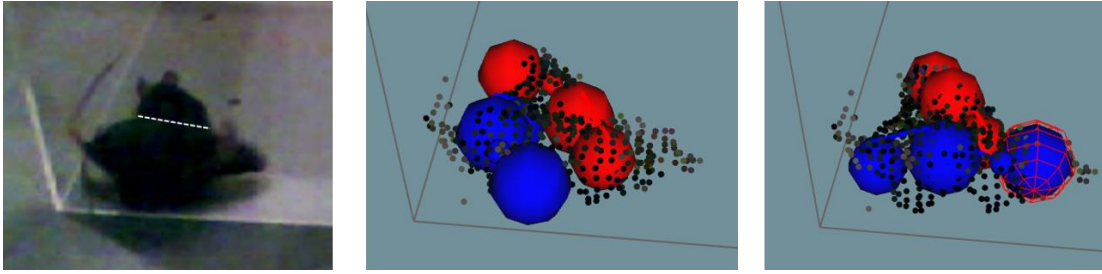
[Fig 5-5]

The model is automatically fit from the initial position. If you find that the model was not fitted properly, manually adjust it by dragging body parts (sphere) of the model with the left mouse button. After placing the skeleton models, start the fitting process by enabling the “Play” check box in the “Player window”. The positions shown during the play are the result of the fitting process. As the fitting progresses, when you find an error in fitting, such as when a big part of a model is outside the 3D image, reversed direction, or swapping between animals (Fig 5-6), pause the fitting by disabling the “Play” check box.



[Fig 5-6: an example of 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 5-7, middle). Then, manually adjust the models by dragging body parts (Fig 5-7, right) and re-start fitting from that frame by enabling “Play” check box.



[Fig 5-7: correction of the swapping error shown in Fig. 5-6. Left: Actual mice positions, middle: before correction, right: after correction.]

If you find an error again, pause and fix it following the instructions described above. Repeat it and then progress with the fitting up to the last frames.

#### Hint

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.
- By rotating the mouse wheel during the video pausing you can change the video frame.
- By rotating the mouse wheel during playback it changes the playback speed.

#### Note

The fitting process must be done without skipping frames to get a successful result. Even if you don't have an animal in the first frame, start from the first frame and not from the frame when the animal appears. For the fitting during the absence of animals, you can withdraw the model with “Withdraw” button in “Tracking tools window”. Note that the 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).

#### 5.4. Saving and Exporting the fitting result ¶

Once you have finished the 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 the next time you open the data. You can also reset the results to the one saved last time from “File menu > Load result”. This save/load function is also useful to interrupt the work. For the data analysis on the trajectories

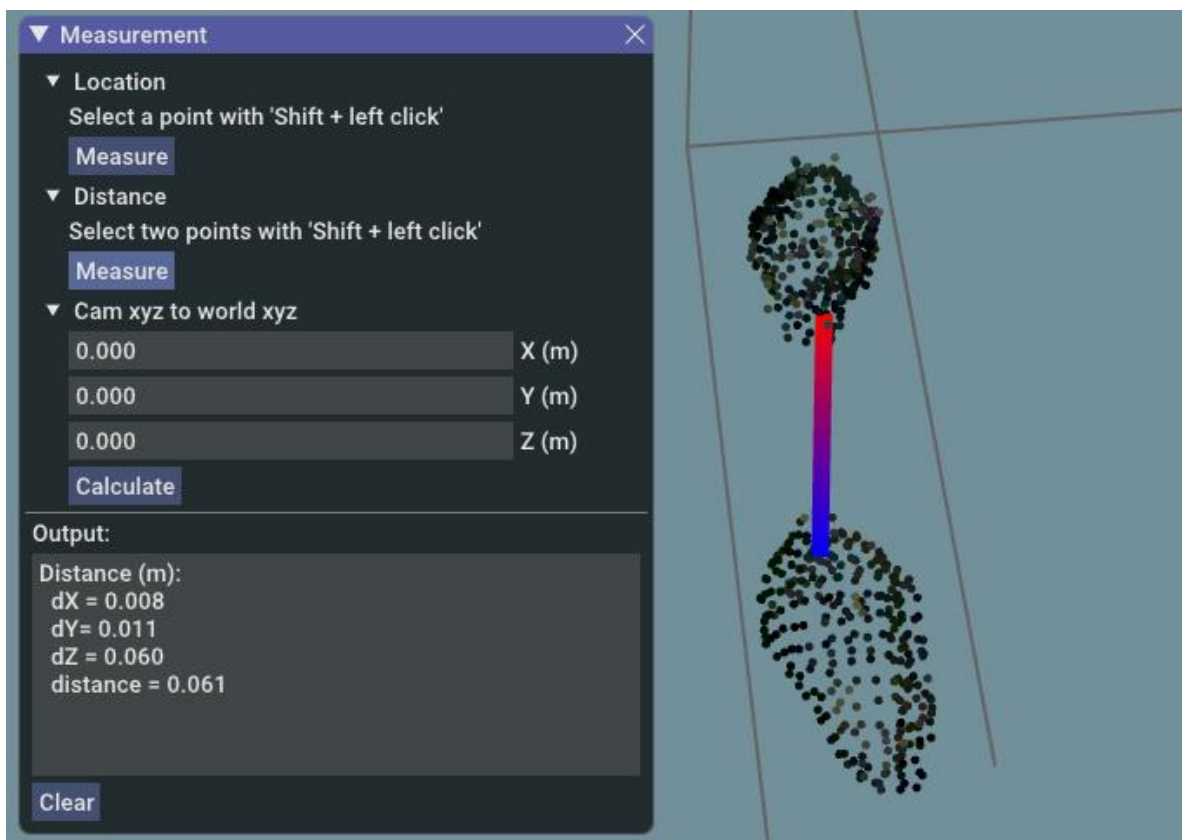
of body parts, you can export the csv format file (Fig 5-8) from “File menu > Export result”.

	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	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
2	Frame	0	0	1	1	0.190868	0.016929	0.050921	0.193111	0.027442	0.063963	0.193971	0.025321	0.079311	0.190574	0.0125	0.095854	0.194303	0.038176	0.193207	0.191487	0.040288	0.196488	0.187972	0.034303	0.186168	0.0125	0.191695
3		1	0.032	1	1	0.192508	0.017946	0.04764	0.191556	0.027731	0.063963	0.193214	0.025541	0.077833	0.194056	0.0125	0.094252	0.192346	0.036427	0.195691	0.192619	0.038618	0.174532	0.189018	0.03364	0.186587	0.0125	0.191255
4		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	0.0125	0.193125
5		3	0.1	1	1	0.195949	0.018632	0.0397	0.195651	0.0275	0.056392	0.194181	0.025387	0.070618	0.193078	0.0125	0.088642	0.197967	0.032329	0.150547	0.195081	0.033262	0.164918	0.192077	0.029484	0.179282	0.0125	0.193586
6		4	0.132	1	1	0.196432	0.017134	0.037233	0.196389	0.027605	0.05253	0.194068	0.026212	0.067467	0.193822	0.0125	0.088452	0.198036	0.031844	0.146278	0.194384	0.031774	0.158387	0.192337	0.02837	0.174238	0.0125	0.193091
7		5	0.166	1	1	0.198228	0.01885	0.03584	0.195837	0.027786	0.05025	0.192795	0.025938	0.064176	0.193116	0.0125	0.080007	0.198236	0.03	0.139994	0.194599	0.028422	0.150937	0.192787	0.024646	0.167743	0.0125	0.186022
8		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	0.0125	0.182446
9		7	0.232	1	1	0.200688	0.015	0.033319	0.196699	0.02658	0.046755	0.195631	0.025151	0.06143	0.194214	0.0125	0.079067	0.204627	0.014883	0.131712	0.195755	0.027784	0.139221	0.195208	0.027841	0.156934	0.0125	0.176797
10		8	0.266	1	1	0.201515	0.017041	0.031236	0.195322	0.027845	0.046115	0.19519	0.025773	0.061181	0.195135	0.0125	0.077148	0.207737	0.015	0.121844	0.198566	0.027594	0.130039	0.197948	0.027553	0.148902	0.0125	0.167968
11		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	0.0125	0.160431
12		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	0.0125	0.166657
13		11	0.366	1	1	0.203733	0.015	0.027027	0.196709	0.024061	0.041807	0.196166	0.023888	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	0.0125	0.150781
14		12	0.4	1	1	0.205094	0.015	0.023483	0.195423	0.023675	0.040746	0.197407	0.023919	0.05683	0.196667	0.013144	0.076045	0.205177	0.0204	0.103939	0.200425	0.030536	0.11741	0.198674	0.028998	0.131025	0.0125	0.145097
15		13	0.432	1	1	0.204524	0.015	0.028518	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	0.031211	0.114809	0.195764	0.029293	0.128455	0.0125	0.140579
16		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	0.201533	0.029976	0.095931	0.195663	0.031318	0.113185	0.196701	0.030309	0.126355	0.0125	0.137816
17		15	0.5	1	1	0.206268	0.016489	0.026532	0.195734	0.024957	0.038432	0.19841	0.025919	0.054529	0.200778	0.015582	0.073497	0.204619	0.034446	0.092922	0.196009	0.034746	0.109293	0.196569	0.031874	0.122613	0.0125	0.131627
18		16	0.532	1	1	0.207486	0.015276	0.024799	0.196091	0.024378	0.037894	0.198445	0.025591	0.054145	0.201536	0.015177	0.073588	0.205608	0.035774	0.091019	0.197026	0.038812	0.107327	0.19686	0.032184	0.120702	0.0125	0.130899
19		17	0.566	1	1	0.208434	0.016487	0.026418	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	0.104835	0.200279	0.031179	0.118807	0.0125	0.131342
20		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	0.0125	0.129819
21		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	0.0125	0.124627
22		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	0.0125	0.12231
23		21	0.7	1	1	0.208997	0.015365	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	0.202039	0.031248	0.112874	0.0125	0.122723

[Fig 5-8]

## 5.5. Measurement tools

When you want to know the coordinates of some points in addition to the coordinates of body parts, “Measurement” window (Fig 5-9) is useful.



[Fig 5-9: Measurement tools. In this example, the distance between manually selected two points (a nose of a mouse and a hip of another mice) is calculated.]

The tool can be used to estimate three types of data: (1) location, this is the coordinate of a selected point, (2) distance between two selected points, and (3) an absolute coordinate converted from a relative coordinate to a camera.

For (1) and (2), select point(s) and click the “Measure” button, then the result will appear in the “Output” text box.

For (3), input the relative coordinate in the text boxes and click the “Calculate” button, then the result will appear in the “Output” text box. This function is useful, for example, when you want to know the coordinate of another sensor (microphone, high-speed camera, 4K camera, etc) mounted on the 3D cameras. If you input  $(X, Y, Z) = (0, 0, 0)$ , you can also know the camera positions.

## 6. Other software usage¶

### 6.1. Viewer¶

The software can be used as a viewer to check the recorded data but without the fitting functions. To open the data in the Viewer, no pre-processing is necessary.

### 6.2. Data analysis scripts (Python / MATLAB)¶

We provide Python and MATLAB scripts for various processing with the trajectory data (csv file) and raw 3D video data (point cloud data, RGBD data) in the following site:

<https://github.com/ThreeD-Tracker-FAB/3DTrackerFAB-script>

We also encourage people to share their own scripts. You can do it through our github site, the forum (<https://groups.google.com/forum/#!forum/3dtracker>) or by contacting us through the form for such purpose (<http://3dtracker.org/contact.html>). If you have a useful code you would like to share, please let us know!

## 7. Trouble shooting¶

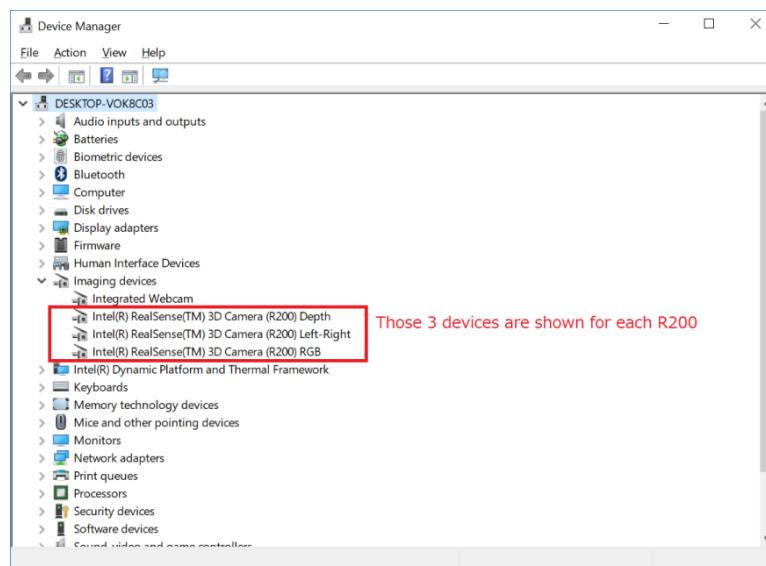
If you cannot find a specific topic associated to the 3DTracker or the solution for some related issue, ask your question in our forum

(<https://groups.google.com/forum/#!forum/3dtracker>).

### 7.1. Calibrator/Recorder does not start (with R200 cameras)¶

The camera may not be recognized by the PC. Rule out any possible issue associated with cables between the PC and the camera. Check the connectors at the camera is not damaged.

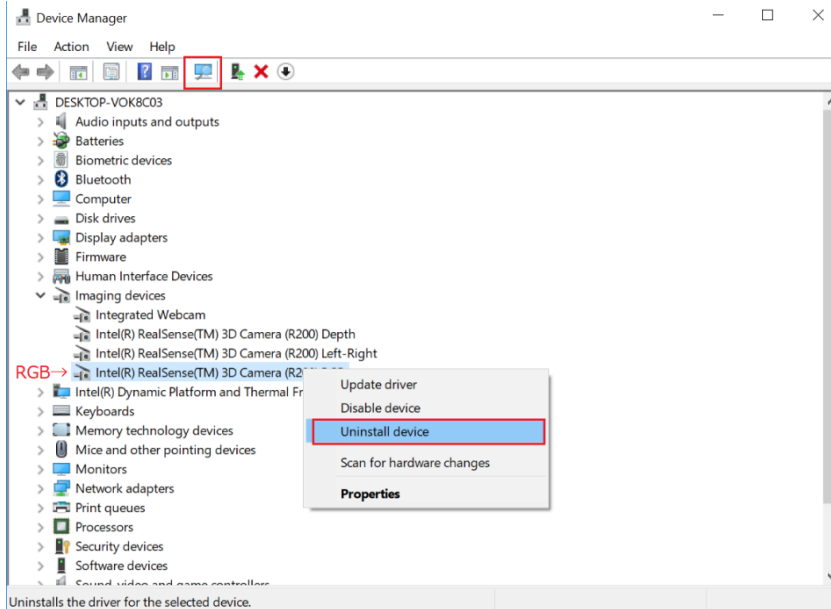
Re-check the connection of the camera according to Chapter 2. If the problem persists, check whether the camera is listed in the “Device manager” (Fig 7-1).



[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 the problem persists, uninstall the device of RGB camera and re-install it by clicking “scan for hardware changes” icon (Fig 7-2). It has been reported that an RGB camera driver error happens after “Windows 10 Creators Update”, and the steps above fix the problem.

(<https://github.com/IntelRealSense/librealsense/issues/481>).



[Fig 7-2]

## 7.2. Calibration does not work well¶

We suggest trying the following:

- Check the area to scan with the pointer. In the 2D view of the Calibrator make sure the pointer is always inside the view of all the cameras, and the distance between pointer and cameras is always more than the minimum distance for the depth sensing of the camera (e.g., 40 cm for R200). Also make sure the pointer light is not strongly reflected by the floor or other objects.

## 7.3. Objects or animals are not captured¶

- Change the infrared (IR) gain. If the camera is too close to the surface recorded or number of cameras is big, the IR reflection become too strong saturating the IR sensor. The data in the saturated part will be lost, interfering the pointer tracking during the calibration. Reduce “IR camera gain” in “Camera setting window” to avoid the saturation (see also [Appendix](#) for the camera mechanism).
- Check the color of the object. Surfaces without IR reflection cannot be captured well (e.g., deep black colored plastic). In the case you need to consider changing the color of the object. Black fur mice (e.g., C57BL/6) can be captured without any problem.

## 7.4. Dropping frames during recording¶

Before you buy a faster PC, try the following:

- Wait long enough after the PC boots (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/>).

#### **7.5. Menu bar invisible / mouse offset from view¶**

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



## 8. Information for developers¶

### 8.1. Preparing the development environment¶

Use Visual Studio 2017 for the IDE.

First, download and install following libraries (You can download from our website at <http://3dtracker.org/downloads.html>):

1. OpenCV 3.2.0 (library for image processing)
2. PCL 1.8.0 (library for point cloud processing)
3. Kinect for Windows SDK v1.8 (library for Kinect v1 camera and backward compatibility)
4. 3DTracker Developer Tools (= librealsense, intel R200 camera library; Bullet physics, physics simulation library; freeglut, free opengl utility toolkit)
5. librealsense 2 SDK (library for using intel D400 camera)
6. (optional) NIDAQmx (National Instrument device library; used for TTL signal)

Second, add the following folder to the environment variable “PATH”:

- C:\Program Files\3DTrackerDevTools\bin
- C:\opencv320\build\x64\vc14\bin
- C:\Program Files\PCL 1.8.0\bin
- C:\Program Files\OpenNI2\Redist

### 8.2. Programming Style¶

For this, follow this link: <https://google.github.io/styleguide/cppguide.html>

### 8.3. More Information¶

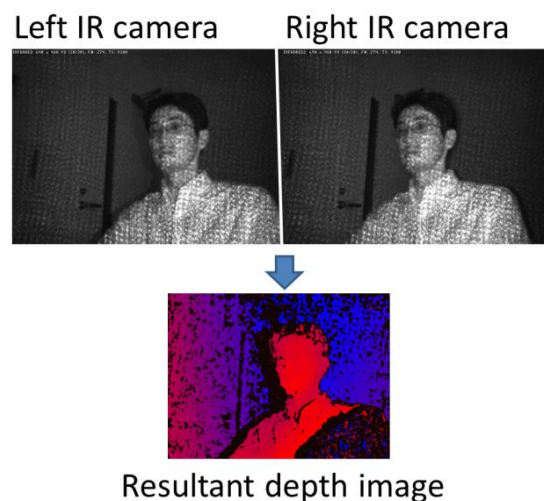
Contact us through the forum

(<https://groups.google.com/forum/#!forum/3dtracker>).

## 9. Appendix¶

### 9.1. R200 camera mechanism¶

Intel realsense R200 contains one RGB camera, two IR cameras, and one IR emitter. The IR emitter projects 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 represents distance from the camera to the object surface, is generated.



[Fig 9-1]

In 3DTracker software, a 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 an excellent feature of R200 compared to the other 3D cameras (for example, ToF cameras, like Kinect v2) that severely interfere with each other. Although the R200 can work well in a multi-camera setup, if many of them are recording an object closely, the R200 would receive too much IR light causing some saturation (red circle in Fig 9-2).



[Fig 9-2]

There is no binocular disparity in the saturated area, so the depth of the area cannot be calculated. By reducing the IR camera gain (in “Camera setting window” of the Recorder app), the saturation can be avoided. For the details of R200, see the Intel datasheet:

<https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/realsense-camera-r200-datasheet.pdf>

## 9.2. Details of the parameters in the Tracker¶

For reference see Matsumoto et al., 2013:

<http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0078460>

## 9.3. Output files¶

The data of a recording session is 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

## 9.4. Synchronizing with other systems¶

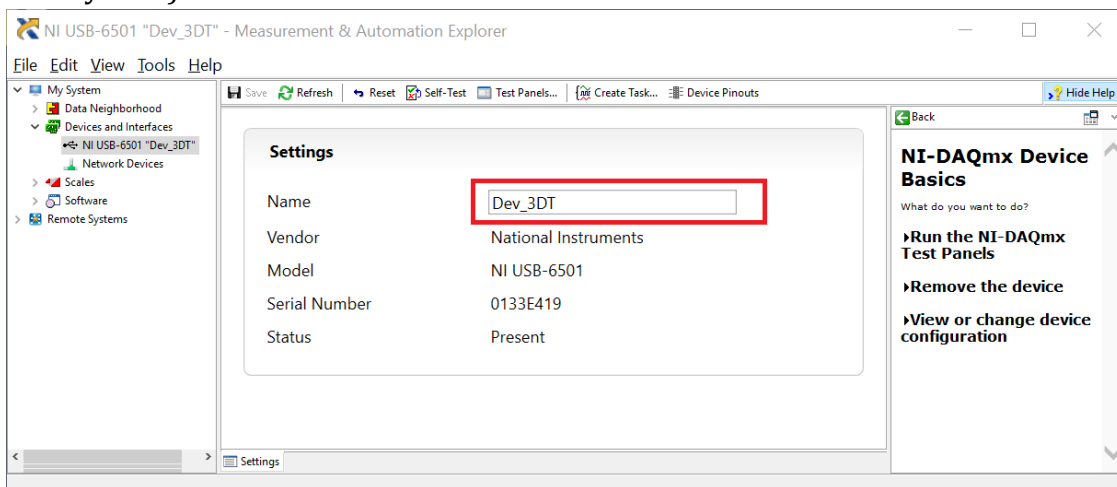
To synchronize the 3D video recording with other data acquisition systems such as an ephys recording system, you can use the Recorder\_TTL.exe. Specifically, the app has the following functions:

- TTL (5V) signal output at each video frame acquisition.
- Starting and stopping recording by TTL trigger inputs.

Hardware and software preparation before using the Recorder\_TTL.exe:

1. Prepare a National Instrument DIO device (e.g., USB-6501)

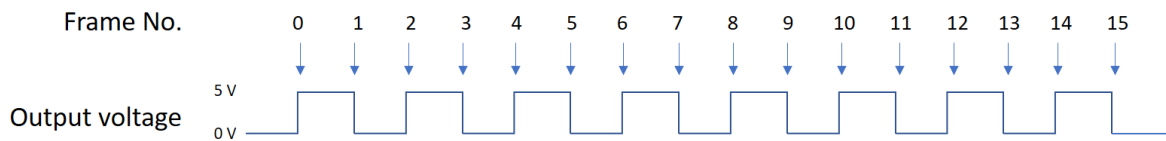
2. Install NIDAQmx software (driver of National Instrument devices; you can find a link in <http://3dtracker.org/downloads.html>)
3. Connect the DIO device to the PC and set the device name as “Dev\_3DT” (Fig 9-3) using NI MAX software.
4. The pins P0.0, P1.0 and P1.1 of the DIO device are for the video frame acquisition timing (output), starting recording and stopping recording (input), respectively. Connect the pins (and the ground pin of the DIO device) to appropriate pins of the other systems (e.g., the frame timing pin to a digital input of a data acquisition system).



[Fig 9-3]

In Recorder\_TTL.exe, check “Trigger Recording by TTL” in the “Recording Window”, to enable triggering recording by the TTL inputs.

During the recording, at each video frame acquisition, the output pin voltage flips transiently from 0 to 5 V and 5 to 0 V. Thus, each of the 5 V onsets and offsets corresponds to a frame acquisition timing (Fig 9-4).



[Fig 9-4: the relation between frame acquisition timings (arrows) and the output pin voltage]

#### Note

The USB data transmission from the camera involves the compression of the data, thus it has a delay (usually 100 - 300 ms). So, the TTL timing (when the PC acquires a video frame) is the actual frame capture timing + the delay. Each user has to measure and correct the delay of the camera. For example, capture an Arduino blinking its LED in front of the camera while the LED pin is also connected to a digital input of a data acquisition system. Then measure the delay, by comparing the timing of the TTL signal of the video frame with the LED onset and the timing of the LED pin signal. Repeat this to estimate the average delay.

#### 9.5. Online tracking (under development)¶

The Tracker can also work in online, directly using the data streaming from the cameras. For the online tracking, connect the cameras to the PC and run Calibrator.exe to calibrate (if calibration has not been done). Then, run Tracker\_online\_mode.bat.

**Date last modified: 2018.10.31**

© Copyright 2017-2018, Jumpei Matsumoto, Marcelo Aguilar-Rivera, et al..