3DTrackerFAB-doc Documentation

Jumpei Matsumoto, Marcelo Aguilar-Rivera, et al.

Sep 24, 2018

Contents

1 Introduction **3**

2 Index **5**

2.1 Hardware preparation 5
2.2 Software preparation 7
2.3 Recorder usage 7
2.4 Preprocessor usage 10
2.5 Tracker usage 11
2.6 Other software usage 16
2.7 Trouble shooting 16
2.8 Information for developers 18
2.9 Appendix 18

The document was written by Dr. Jumpei Matsumoto and Dr. Marcelo Aguilar-Rivera. We thank Dr. Laleh Quinn for her proof reading.

CHAPTER 1

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.

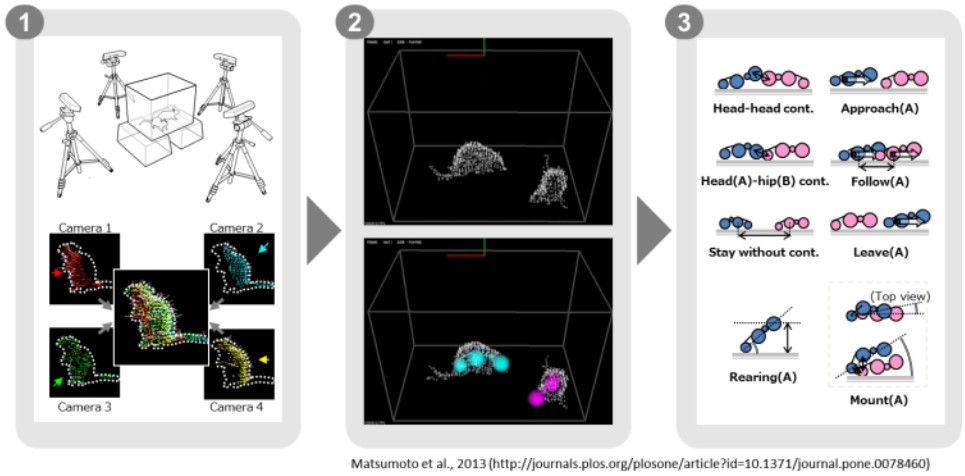

Fig. 1: [Fig 0-1]

First, animals 3D imagines 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 "Recorder" module (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) is analyzed with Python/MATLAB scripts (Chapter 6). The present version of the 3DTracker-FAB only supports the analysis of rats and mice. The scripts for post-processing of the trajectories are in preparation. 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.

CHAPTER 2

Index

2.1 Hardware preparation

2.1.1 Cameras

We suggest using Intel realsense R200 cameras (Fig 1-1) 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 is composed of a color camera and a depth sensor to obtain RGB images and depth images, respectively.

Fig. 1: [Fig 1-1: Intel realsense R200 camera]

Unfortunately, R200 is now discontinued. However, there are still some stock in the intel store as part of a kit (https://click.intel.com/intelr-realsensetm-robotic-development-kit-2351.html) for robotics. In addition, sometimes it is possible to find them at Mouser Electronics. Given that the original USB 3.0 cable of the R200is too short, it is necessary to extend it. We suggest using an "active" repeater cable as found in the link (https://www.amazon.com/dp/B01FQ88CE6)

2.1.2 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 processers?) in the CPU should be => 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 R200 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.

R200 uses almost the whole transmission bandwidth of an USB 3.0 port. So, more than one R200 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 connecting one R200 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 they are 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 shown above are the best ones based on our experience.

2.1.3 Camera mounts

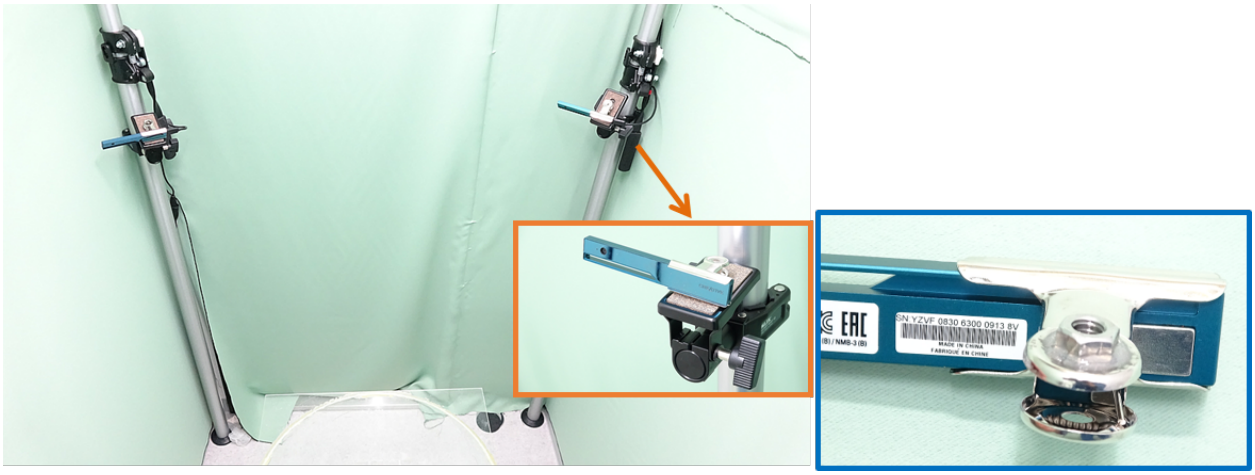

Fig. 2: [Fig 1-2]

Example of four depth cameras setup. Each 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. The depth sensor of R200 can't detect a depth < 40 cm. The horizontal and vertical field of view of R200 is 59 $\pm$ 5 and 46 $\pm$ 5 degrees, respectively. These constraints should be considered during camera installation.

2.2 Software preparation

2.2.1 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 https://github.com/IntelRealSense/librealsense/issues/163; https://github.com/IntelRealSense/librealsense/issues/429).

After installing the libraries 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

2.2.2 Installing main programs of the 3DTracker-FAB

Download the prebuilt version of the 3DTracker-FAB from the 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.

2.3 Recorder usage

The Recorder is a module for calibration of the multiple-cameras setup of the 3DTracker system.

2.3.1 Starting the Recorder

Connect the cameras to the PC and start Recorder.exe. If Recorder.exe doesn't start, the cameras may have not been recognized (see Chapter 7 for the trouble shooting). Once the Recorder starts correctly, the following screen will appear:

You will see a menu-bar, and a 3D view display where camera images are projected as a cloud of points. 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 red, green and blue axes shown in the

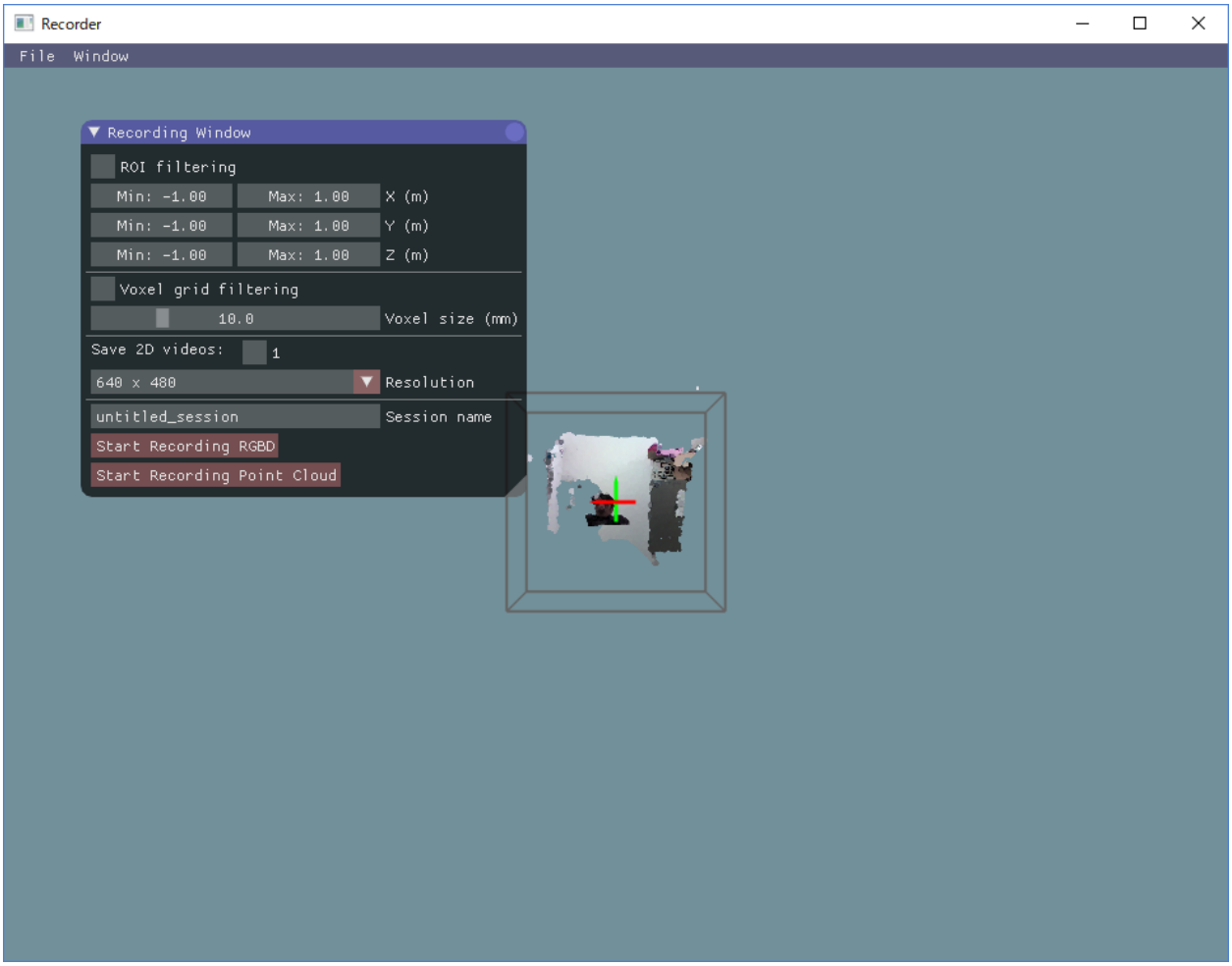
Recorder
File Window
Recording Window
ROI filtering
Min: -1.00
Max: 1.00
X (m)
Min: -1.00
Max: 1.00
Y (m)
Min: -1.00
Max: 1.00
Z (m)
Voxel grid filtering
10.0
Voxel size (mm)
Save 2D videos:
1
640 x 480
Resolution
untitled_session
Session name
Start Recording RGBD
Start Recording Point Cloud

Fig. 3: [Fig 3-1]

3D view represent X, Y, and Z axes, respectively. The remaining tool windows can be displayed from the menu tab > 'Window'.

2.3.2 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 discriminate clearly between the points from different cameras, you can color the points by enabling "Mark cameras with different colors" in the "Point cloud appearance" window. The misalignment is because of the lack of information about the relative positions of the cameras. The Recorder can estimate the relative positions through a semi-automatic calibration, as follows. During calibration the user scans the recording space with a light pointer, and records the pointer track on the "Calibration window". Then, the recorder calculates the relative camera positions fitting the observed trajectories recorded from the different cameras during the calibration process. The pointer can be made by epoxying a white ping-pong ball of 40 mm of diameter (Fig 3-2), onto the tip of a white LED flashlight (https://www.amazon.com/dp/B00BIG5JQK/). The ping-pong ball works as a white light diffuser and an IR reflector.

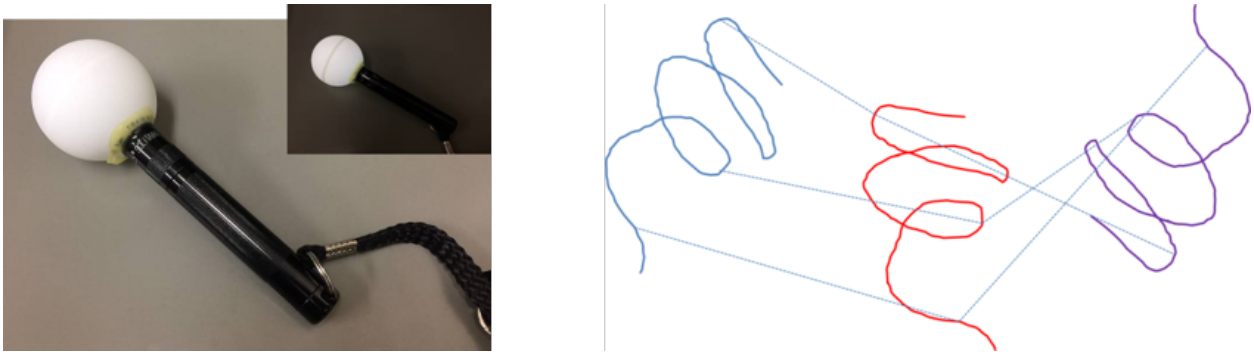

Fig. 4: [Fig 3-2: left, light pointer; right, schematic showing the mechanism. The lines with different colors represent the pointer trajectories observed simultaneously from 3 different cameras in this example.]

The Recorder estimates the center of the pointer based on the closest blight point from a camera and a radius of the ping-pong ball, as shown below.

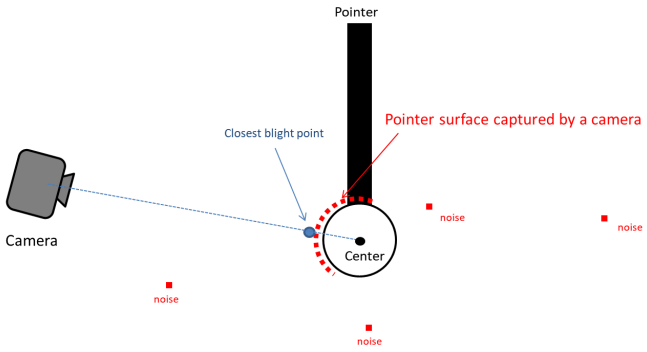

Fig. 5: [Fig 3-3]

Display the "Calibration window" from the "Window" menu. "Color filtering" setting section is used for detecting bright points that belong to the pointer. Find the best brightness threshold (the value of "Min Brightness" slider) by enabling "Test" check box.

The "Outlier removal" setting section below the "Color filtering" section is used to remove minor noise referred here as small clusters of blight points because of light reflections, etc. For details about the parameters of the Outlier

removal, check here (http://pointclouds.org/documentation/tutorials/statistical_outlier.php). Usually, changing only the "Threshold" parameter is fine. You can also check the effect by enabling the check box "Test".

After adjusting filter settings for the pointer detection, click the "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 view of all the cameras. To check the 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 this operation has been successfully done, the point clouds from different cameras should be perfectly aligned. To check the alignment, place some objects in front of the camera and color the points by enabling "Mark cameras with different colors" in the "Point cloud appearance" 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, 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 "Reference camera position & pose" section on the top of the "Calibration" window. 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, set the region of interest (ROI) with the sliders on the top of the "Recording" window. You can also input the value directly by clicking + Ctrl key on slider. The same setting is expected to be useful for the coming recording sessions run with the same setup (unless the camera has been moved). To save and load the setting, use "File menu > save config" and "File menu > load config", respectively. See Chapter 7 for trouble shooting calibration. For more information about calibration see the video at https://youtu.be/04JgLTaYI2c

2.3.3 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 consisting 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 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.

Warning: if you work with the session name that was used previously, the original data will be overwritten.

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.

2.4 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 processes 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.

2.5 Tracker usage

2.5.1 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.

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.

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 physic simulation.

2.5.2 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".

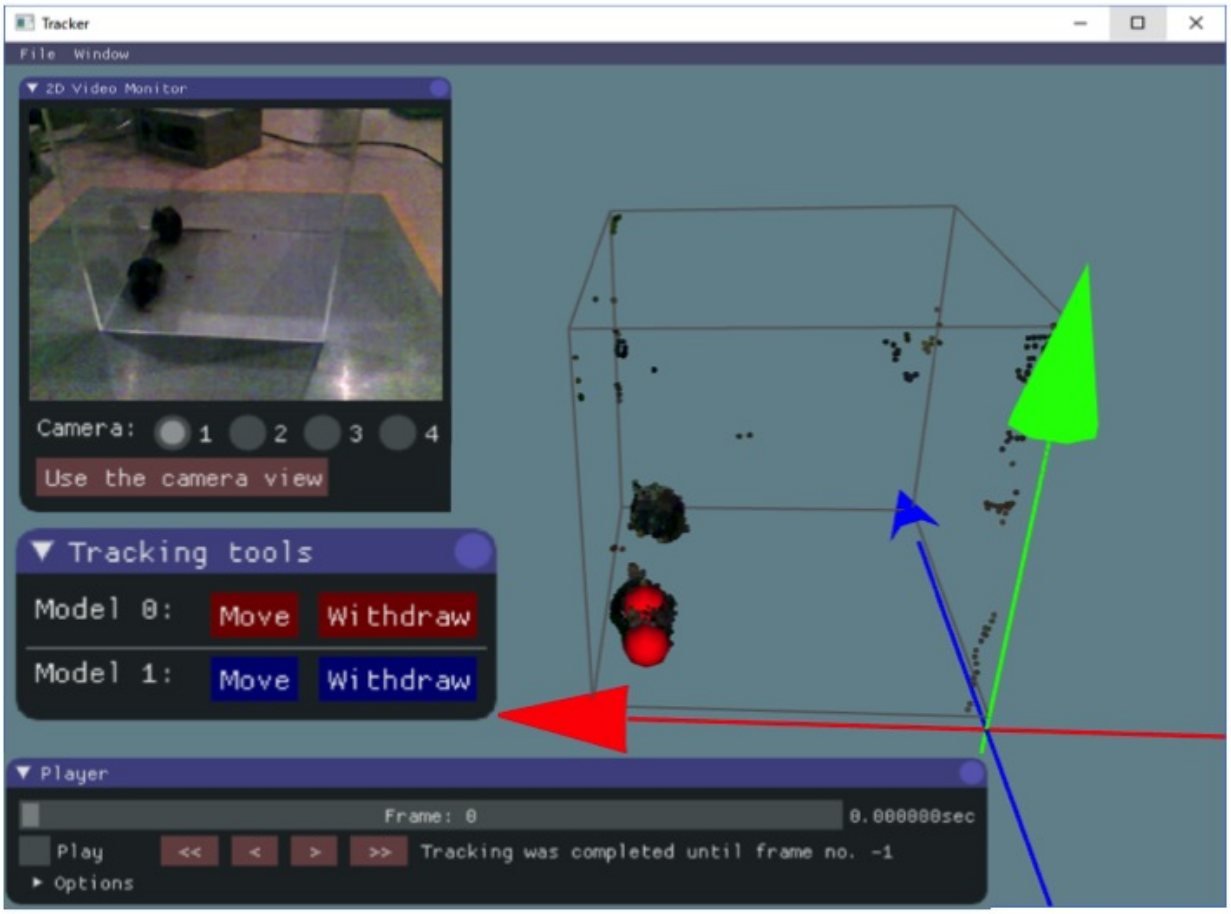

Fig. 6: [Fig 5-1]

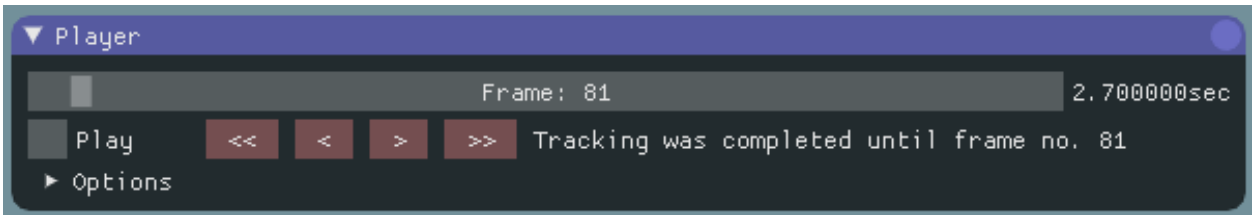

Fig. 7: [Fig 5-2]

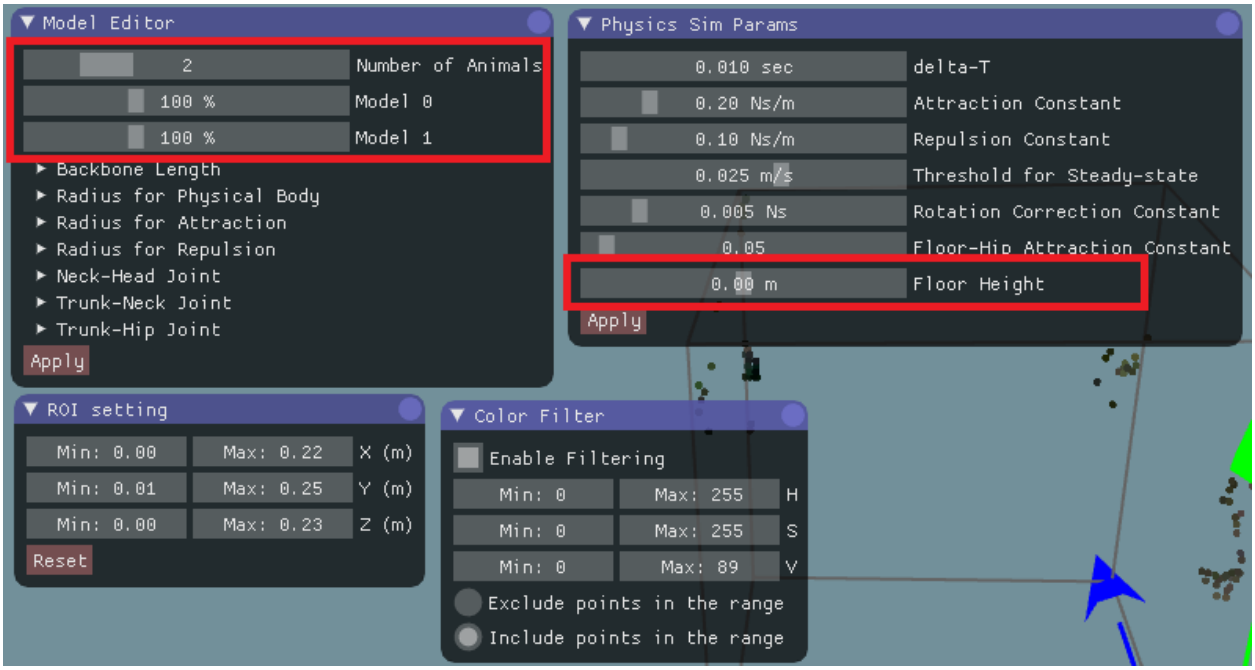

Fig. 8: [Fig 5-3]

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). 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.

2.5.3 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.

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).

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

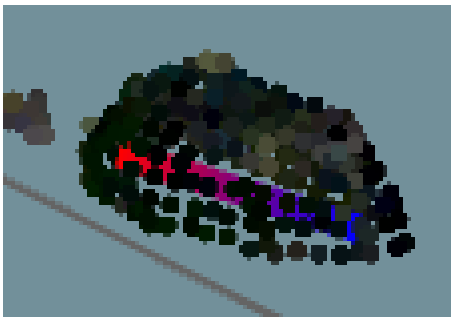

Fig. 9: [Fig 5-4]

Fig. 10: [Fig 5-5]

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.

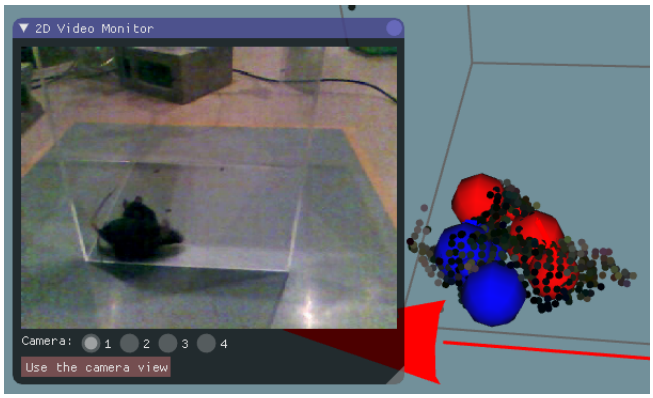

Fig. 11: [Fig 5-6: an example of swapping error]

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.

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.

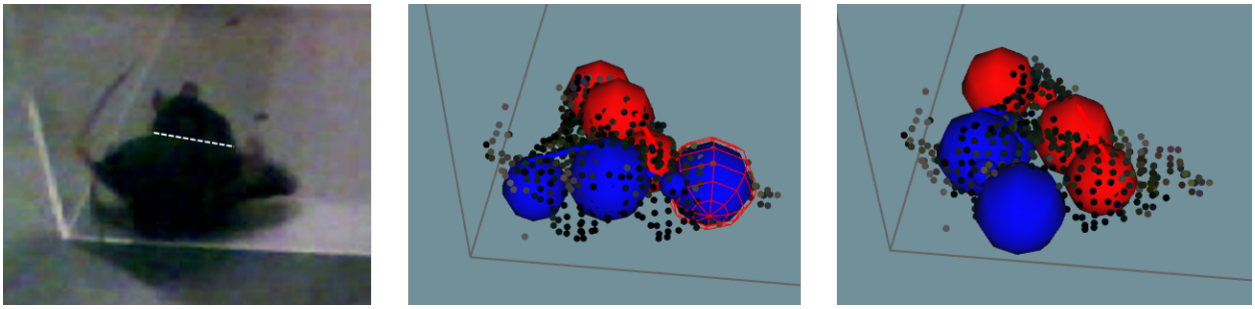

Fig. 12: [Fig 5-7: correction of the swapping error shown in Fig. 5-6. Left: Actual mice positions, middle: before correction, right: after correction.]

- 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).

2.5.4 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:	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. 13: [Fig 5-8]

2.6 Other software usage

2.6.1 Viewer

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

2.6.2 Data analysis scripts (Python / MATLAB)

These scripts are in preparation.

These scripts will be for analyzing the trajectory data (csv files exported from the Tracker) and the original point cloud data.

We encourage people to share their own scripts. Please let us know through the forum (https://groups.google.com/forum/#!forum/3dtracker) or the contact form (http://3dtracker.org/contact.html) if you have a useful code you would like to share. Also, you can find the old MATLAB scripts that were previously published with our paper, Matsumoto et al., 2013, (http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0078460#s6). Even though these old scripts cannot be directly used with the new data format, they would be a good starting point to write your own codes.

2.7 Trouble shooting

If you cannot find a specific topic associated to the 3DTracker or the solution for related some issue, ask your question in our forum (https://groups.google.com/forum/#!forum/3dtracker).

2.7.1 Recorder does 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 the problem persists, check whether the camera is listed in the "Device manager" (Fig 7-1).

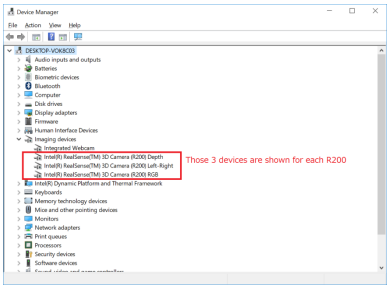

Fig. 14: [Fig 7-1]

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 a RGB camera driver error happens after "Windows 10 Creators Update", and the steps above fix the problem. (https://github.com/IntelRealSense/librealsense/issues/481).

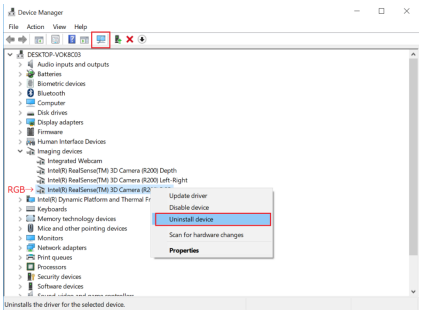

Fig. 15: [Fig 7-2]

2.7.2 Calibration does not work well

We suggest trying the following:

- Change the infrared (IR) gain of the R200. 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 (Fig 7-3; see also *Appendix* for the camera mechanism).

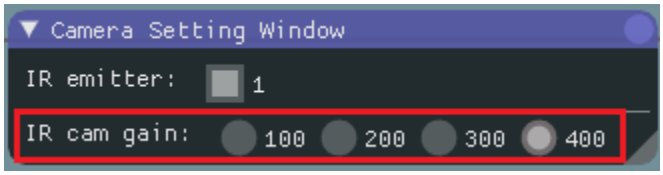

Fig. 16: [Fig 7-3]

- Check the area to scan with the pointer. Show "Camera monitor window" and make sure the pointer is always inside the view of all the cameras, and the distance between pointer and cameras is always more than 40 cm (R200 sensor can't detect a depth < 40 cm). Also make sure the pointer light is not strongly reflected by the floor or 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.

2.7.3 Object or animals are not captured

- Change the infrared (IR) gain of the R200 as it is described above.
- Check for objects in front of the camera (R200). The RGB camera in the R200 can be seen from outside. However, the other two IR sensors and the IR emitter are behind the blue case. For information about the R200 structure see the link: 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. Surfaces without IR reflection cannot be captured (e.g., deep black colored plastic). In the case you need to consider changing the color of the object. Mice with black coats (e.g., C57BL/6) can be captured without any problem.

2.7.4 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/).

2.7.5 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.

2.8 Information for developers

2.8.1 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 realsense camera library; Bullet physics, physics simulation library; freeglut, free opengl utility toolkit)
5. (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

2.8.2 Programming Style

Basically follow https://google.github.io/styleguide/cppguide.html

2.8.3 More Information

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

2.9 Appendix

2.9.1 R200 camera mechanism

ntel 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.

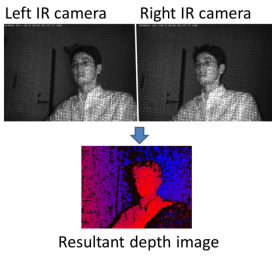

Fig. 17: [Fig 9-1]

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. 18: [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

2.9.2 Details of the parameters in the Tracker

Coming soon.

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

2.9.3 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

Date last modified: 2018.01.22