Interactive Tracking Manual

This document serves as a guide on how to use the Interactive Tracking software to obtain high precision tracking of objects recorded from multiple Kinect cameras. The system requirement is as follow:

- MATLAB R2014a (or later version)
- MATLAB Computer Vision Toolbox
- MATLAB Image Processing Toolbox
- MATLAB Statistics and Machine Learning Toolbox

The operation of the system is divided into four components. Here we give a brief overview of these components (more details will be given later).

1. Data preprocessing:
   Extraction of color images from raw .mat Kinect recording files. This also reduces the size of the data to roughly 25% of the original size.

2. Camera synchronization and reunion event identification
   Manual inspection of the videos to synchronize the cameras and to identify the start/end of the reunion.

3. Interactive tracking
   An interactive process to track target objects in videos.

4. Data extraction
   Automated process to extract 3d location of the objects as well as deriving the approach/avoidance data from the result of interactive tracking.

Now we describe into more details how to run each of the above components. Basic understanding of MATLAB commands is required to be able to understand the rest of this document. Before we begin, make sure to set the current matlab directory to the directory that contains the interactive tracking code, and type the following in the matlab console:

```
addpath(genpath(path_to_InteractiveTracking));
```

We are now ready to begin.

Data preprocessing

The first step that we need to do is to extract the color images from raw .mat Kinect recording as well as structuring the data into a format that is usable by the Interactive Tracking software. To do this, first identify the .mat files directories of the 4 Kinect cameras (K1, K2, K3 and K4), and create a matlab cell variable that points to those directories. For example:

```
datapath{1} = 'F:\FN55-K3-EXPORT';
datapath{2} = 'F:\FN55-K4-EXPORT';
datapath{3} = 'F:\FN55-K1-EXPORT';
datapath{4} = 'F:\FN55-K2-EXPORT';
```

Important: note how datapath{1} points to K3, datapath{2} to K4 and so on. **Please follow this convention.**

For the case when there is no data from a particular Kinect (which sometimes happen due to recording failure), we can assign empty variable to the datapath cell for that particular kinect. For example, assuming we don't have data from K1, the datapath variable then becomes the following:

```
datapath{1} = 'F:\FN55-K3-EXPORT';
datapath{2} = 'F:\FN55-K4-EXPORT';
datapath{3} = [];
datapath{4} = 'F:\FN55-K2-EXPORT';
```

Next, we create a vector in matlab to identify which kinects are present. In the previous example (missing data from K1), the vector should be as follow:

```
idx = [1 2 4];
```

Note how '3' is missing from the variable `idx` (remember the earlier convention where datapath{3} points to K1).

Finally, create a string that points to the target directory for saving the preprocessed data. For example, if we want to save the preprocessed data to 'F:\Preprocessed\FN055', type the following in matlab:

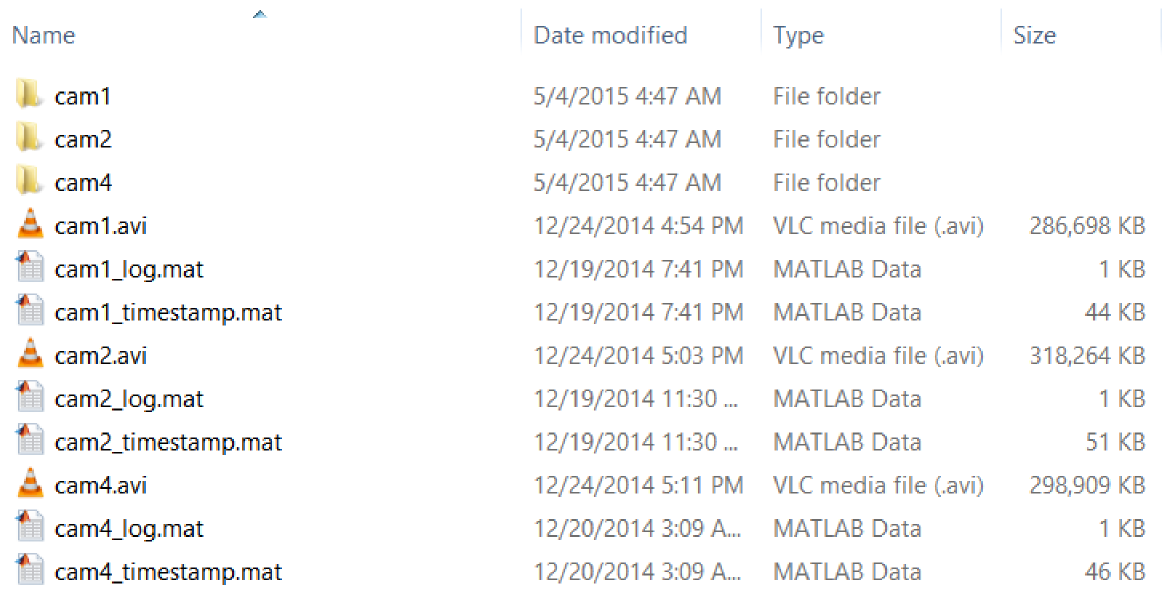
```
savepath = 'F:\Preprocessed\FN055';
```

Important: the naming convention for the Strange Situation is FNxxx and the naming convention for the Exploration protocol if FNxxxE (notice the 'E' and the end). **Please follow this convention**.

After creating all the above 3 variables in matlab (`datapath`, `idx` and `savepath`). We are ready to run the data preprocessing step. Simply type the following command in matlab:

```
ExtractImagesFromMat(datapath, savepath, idx)
```

This process will take several hours to finish. It is highly recommended to do run this overnight. Once the data preprocessing is completed, the savepath directory will look like the following:



Name	Date modified	Type	Size
cam1	5/4/2015 4:47 AM	File folder	
cam2	5/4/2015 4:47 AM	File folder	
cam4	5/4/2015 4:47 AM	File folder	
cam1.avi	12/24/2014 4:54 PM	VLC media file (.avi)	286,698 KB
cam1_log.mat	12/19/2014 7:41 PM	MATLAB Data	1 KB
cam1_timestamp.mat	12/19/2014 7:41 PM	MATLAB Data	44 KB
cam2.avi	12/24/2014 5:03 PM	VLC media file (.avi)	318,264 KB
cam2_log.mat	12/19/2014 11:30 ...	MATLAB Data	1 KB
cam2_timestamp.mat	12/19/2014 11:30 ...	MATLAB Data	51 KB
cam4.avi	12/24/2014 5:11 PM	VLC media file (.avi)	298,909 KB
cam4_log.mat	12/20/2014 3:09 A...	MATLAB Data	1 KB
cam4_timestamp.mat	12/20/2014 3:09 A...	MATLAB Data	46 KB

Figure 1. Preprocessing results.

Now we are ready to move to the next step.

Camera synchronization and reunion event identification

Now we need to synchronize the different Kinect cameras). The goal here is to compute the ***timing offset*** of one Kinect with respect to a ***target*** Kinect. The basic convention here is that **whenever cam2 is present, we use it as the target**. Otherwise, other cam can be used as the target. First, we create a .txt file called `maincam.txt` in the savepath directory and the content of that file is simply a single number indicating the target camera (in our example, it is 2).

Next, we need to figure out the timing offset from the various kinects to the target Kinect. To do this, look for moments that can be used to synchronize the videos (e.g. frames that contain the clapper). For example, the clapper was used on frame 633 for cam2 and frame 1410 for cam4.

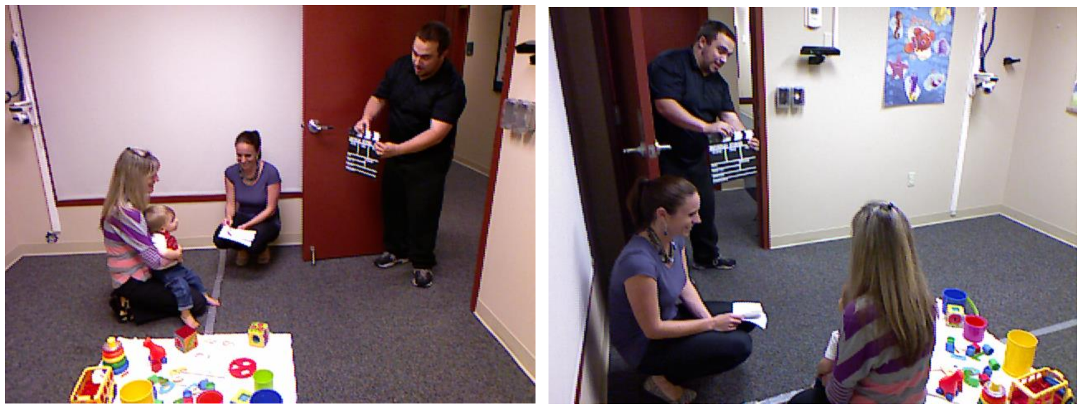


Figure 2. Clap for synchronization. Left is frame 633 for cam2, right is frame 1410 for cam4.

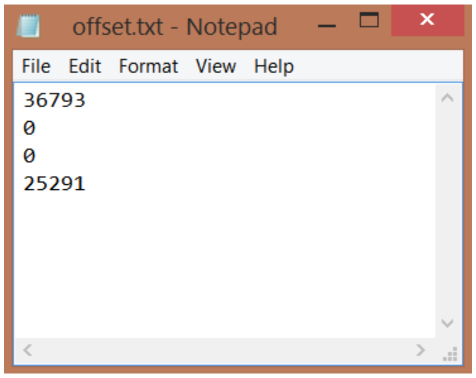
Given a frame number, we can compute the timing offset by typing the following command to matlab:

ComputeTimeOffset(savepath, 4, 1410, 2, 633)

MATLAB will then give the following output

ans = 25291

Create a file named offset.txt inside the savepath directory, and write down this offset number into the file. The file offset.txt should contain **exactly** four lines where each line describe the timing offset of that camera to the target camera. In our example above, line 4 of offset.txt should be 2591, line 3 should be 0 since there is no cam3, line 2 should also be 0 since cam2 is the target camera. The value of line 1 should be computed the same way we compute the value of line 4 (i.e. by finding sync event in the video). The following is an example of offset.txt:



```
36793
0
0
25291
```

Figure 3. Content of offset.txt. Target cam is cam2. Cam3 has no data.

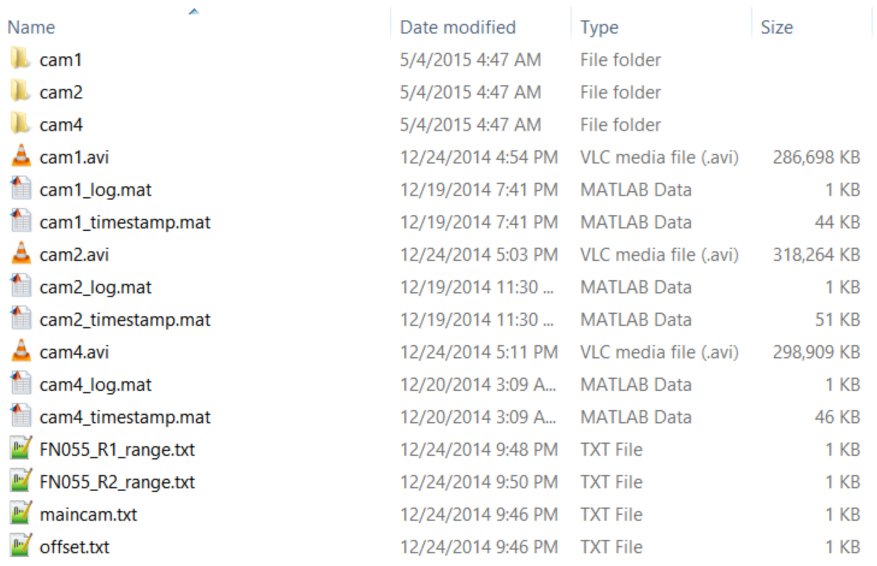
For the last step, we need to identify that start/end of the reunion **in the target camera**. In our example above, the target camera is cam2, and after looking at the images, we identify that R1 begins on frame 14750 and ends on frame 20450. Create a .txt file called 'FN055_R1_range.txt' in the savepath directory, and the content of the file should be the following:

```
14750:20450
```

Important: note the use of colon ':' in between the two numbers.

Do the same for R2 and create a file called 'FN055_R2_range.txt'.

Once all the above process is done, the savepath directory should look like the following:



Name	Date modified	Type	Size
cam1	5/4/2015 4:47 AM	File folder	
cam2	5/4/2015 4:47 AM	File folder	
cam4	5/4/2015 4:47 AM	File folder	
cam1.avi	12/24/2014 4:54 PM	VLC media file (.avi)	286,698 KB
cam1_log.mat	12/19/2014 7:41 PM	MATLAB Data	1 KB
cam1_timestamp.mat	12/19/2014 7:41 PM	MATLAB Data	44 KB
cam2.avi	12/24/2014 5:03 PM	VLC media file (.avi)	318,264 KB
cam2_log.mat	12/19/2014 11:30 ...	MATLAB Data	1 KB
cam2_timestamp.mat	12/19/2014 11:30 ...	MATLAB Data	51 KB
cam4.avi	12/24/2014 5:11 PM	VLC media file (.avi)	298,909 KB
cam4_log.mat	12/20/2014 3:09 A...	MATLAB Data	1 KB
cam4_timestamp.mat	12/20/2014 3:09 A...	MATLAB Data	46 KB
FN055_R1_range.txt	12/24/2014 9:48 PM	TXT File	1 KB
FN055_R2_range.txt	12/24/2014 9:50 PM	TXT File	1 KB
maincam.txt	12/24/2014 9:46 PM	TXT File	1 KB
offset.txt	12/24/2014 9:46 PM	TXT File	1 KB

Figure 4. Content of savepath directory. Note the 4 additional .txt files compared to Figure 1.

We are now ready to begin the interactive tracking process.

Interactive tracking

To start the tracking process, run the interactive tracking tool in matlab by typing the following in matlab console:

```
InteractiveTracking
```

You will be presented with the following user interface:

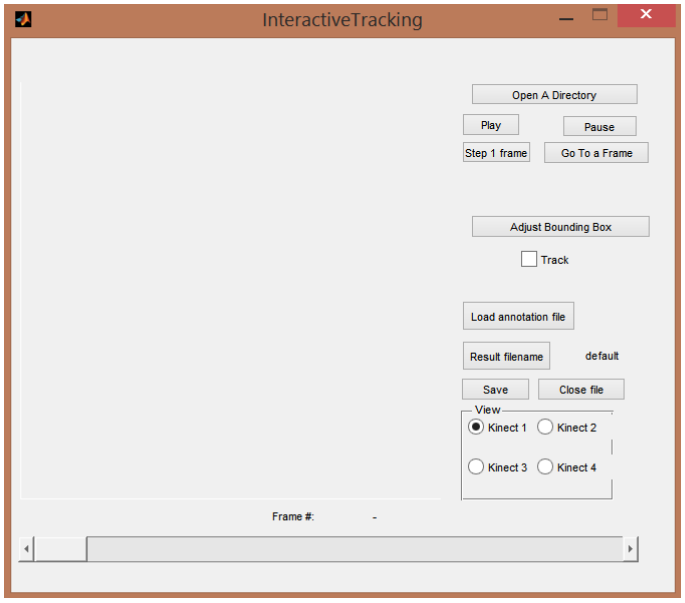


Figure 5. UI of the InteractiveTracking software

Next, open a directory containing a preprocessed Kinect data by pressing the ‘Open A Directory’ button. After opening a directory, you should be able to see an image from one of the videos. You can navigate the video (play, pause, jump to a specific frame) by using the highlighted navigational buttons below:

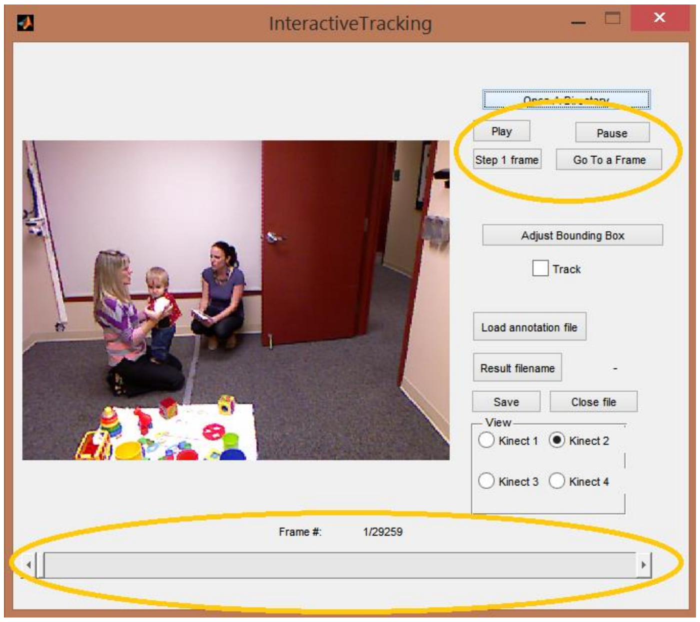


Figure 6. Navigational buttons.

You can also switch between the different Kinect by clicking the radiobuttons inside the ‘view’ section.

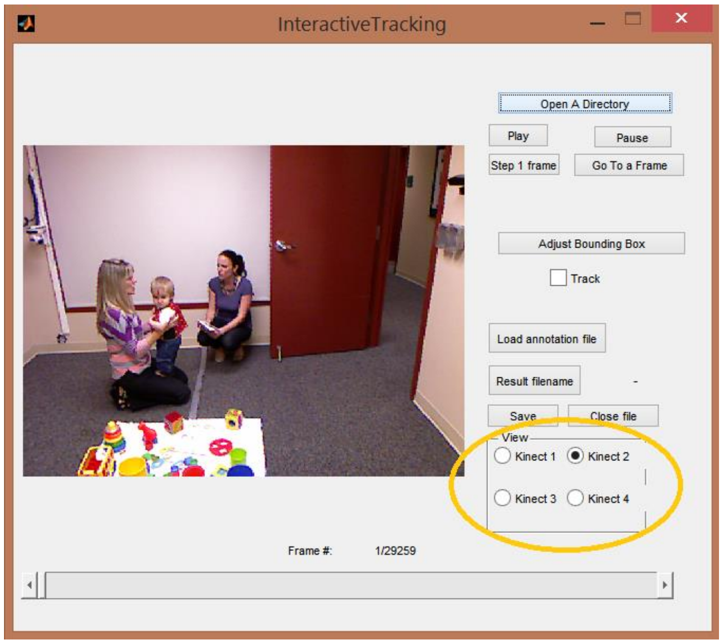


Figure 7. Switching between kinects.

You can load previous annotations by clicking the 'Load annotation file' button. After loading an annotation file, you should see a green bounding box on frames which have been annotated.

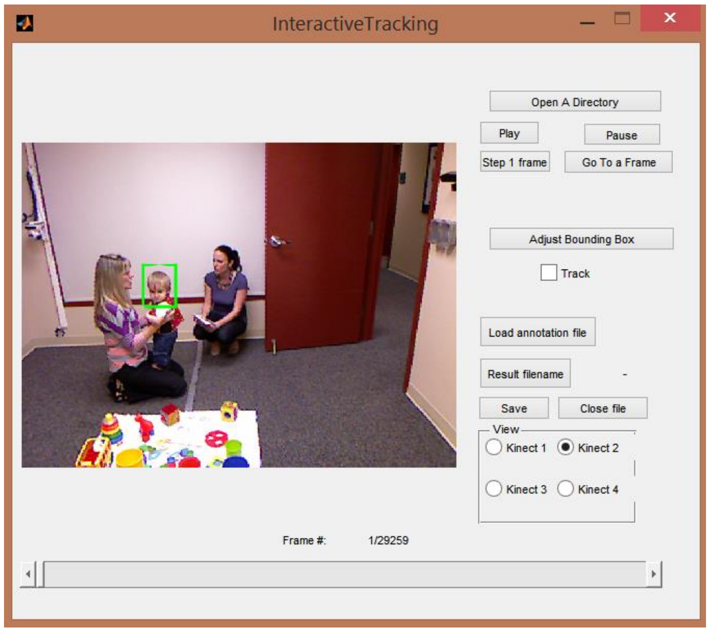


Figure 8. Bounding box after loading the annotated file.

The interactive tracking process can be started from scratch or from previous annotation work (by first loading an annotation file). Once you are ready to begin the annotation process, make sure the software is in the tracking mode by checking the 'Track' button:

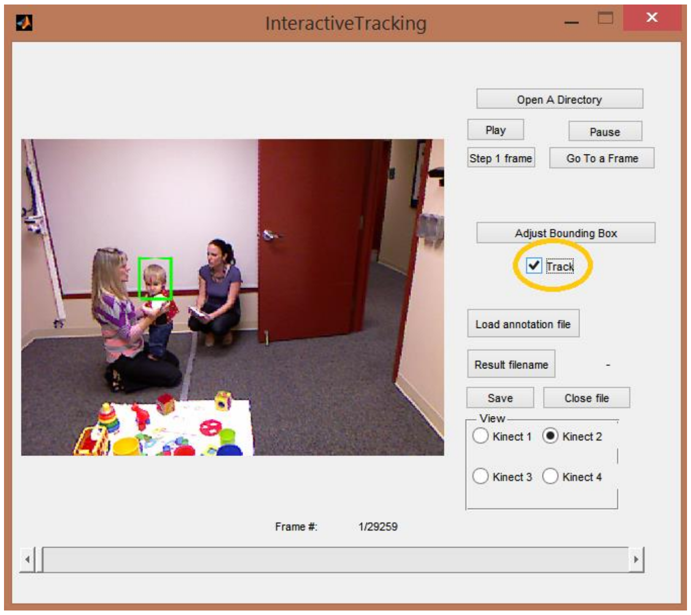


Figure 9. Check 'Track' to switch to tracking mode.

Before we begin tracking, make sure to input a proper filename for the result by clicking the 'Result filename' button. **Important**: for the strange situation, the naming convention for the filenames are baby1 for the baby in R1, baby2 for the baby in R2, mom1 for the mother in R1, and so on. For the exploration protocol, toy1 for first exploration, toy2 for the second, etc.

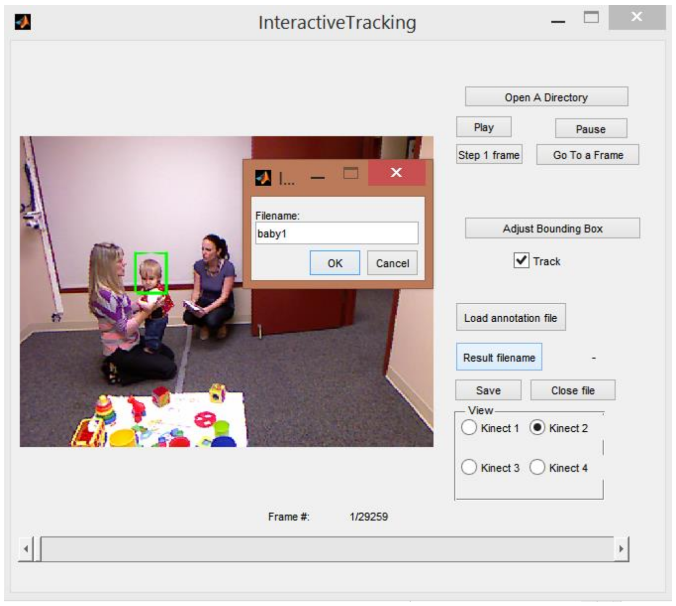
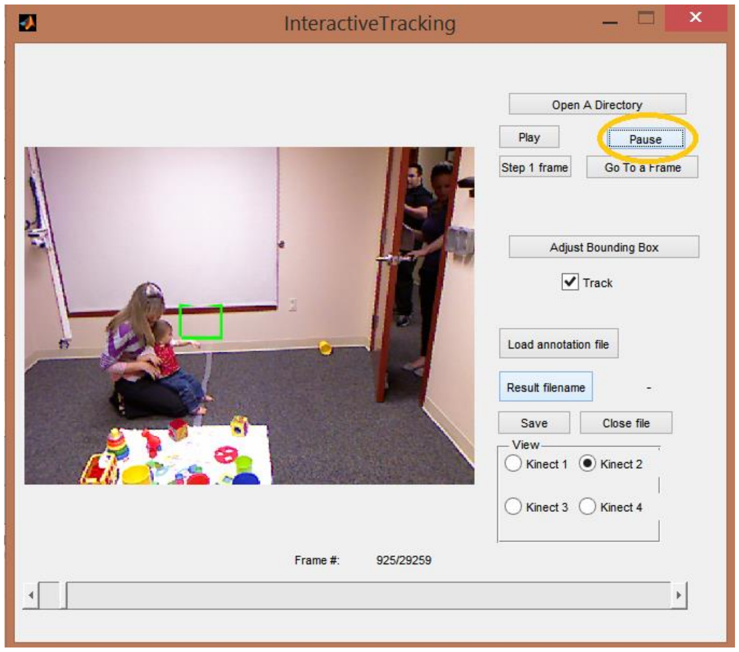


Figure 10. Assigning name to the result file. Please adhere to the naming convention for SS and Exploration.

Now we are ready to begin tracking. Jump to the frame the marks the start of the reunion (or exploration) that have been identified through the previous step. Press 'Adjust bounding box' button and annotate the target object bounding box by clicking the **top left** and **bottom right** corner of the target object in the image. You should see the target contained inside the bounding box in green. Now, click 'Play' to begin the tracking process. Note that as the tracking progresses, drift will occur (i.e. the bounding box started drifting from the target object). Whenever this happens, just hit the 'Pause' button and adjust the bounding box by pressing the 'Adjust bounding box' delineating the correct bounding box for the target.



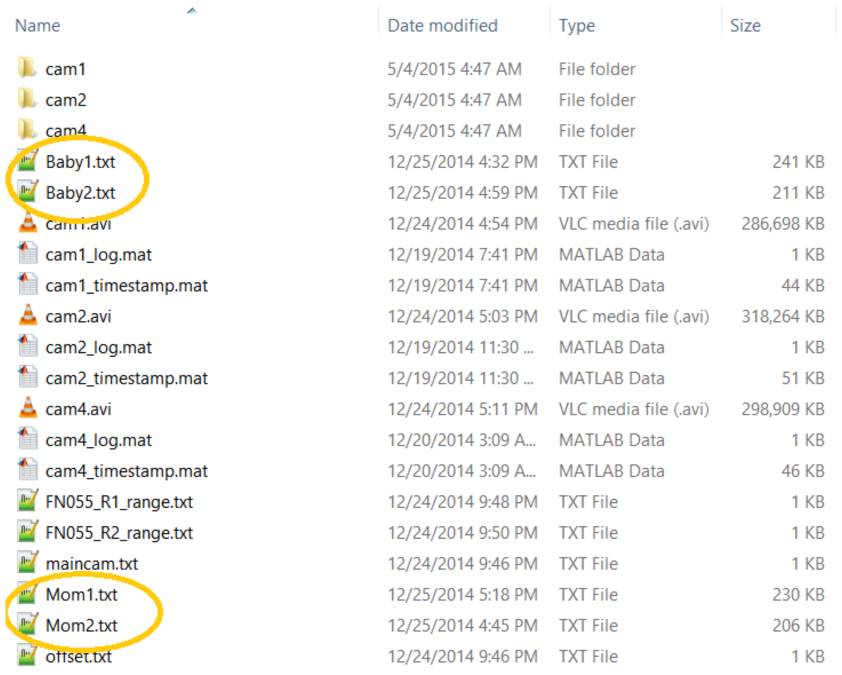
[✓ Track]

View: Kinect 1 [✓ Kinect 2] Kinect 3 Kinect 4

Frame #: 925/29259

Figure 11. Target drift. Hit pause and adjust the bounding box.

Continue this interactive tracking process until the end of the reunion. Adjust the bonding box whenever necessary. Also switch view to the appropriate camera whenever the target object goes out of view. Once done, press the 'Save' button to save the tracking result and finally press the 'Close file' button to end the tracking session. Do this interactive tracking process for each target for each of the reunions. After this is done, the savepath directory should contain 4 additional .txt files (for the strange situation):



Name	Date modified	Type	Size
cam1	5/4/2015 4:47 AM	File folder	
cam2	5/4/2015 4:47 AM	File folder	
cam4	5/4/2015 4:47 AM	File folder	
Baby1.txt	12/25/2014 4:32 PM	TXT File	241 KB
Baby2.txt	12/25/2014 4:59 PM	TXT File	211 KB
cam1.avi	12/24/2014 4:54 PM	VLC media file (.avi)	286,698 KB
cam1_log.mat	12/19/2014 7:41 PM	MATLAB Data	1 KB
cam1_timestamp.mat	12/19/2014 7:41 PM	MATLAB Data	44 KB
cam2.avi	12/24/2014 5:03 PM	VLC media file (.avi)	318,264 KB
cam2_log.mat	12/19/2014 11:30 ...	MATLAB Data	1 KB
cam2_timestamp.mat	12/19/2014 11:30 ...	MATLAB Data	51 KB
cam4.avi	12/24/2014 5:11 PM	VLC media file (.avi)	298,909 KB
cam4_log.mat	12/20/2014 3:09 A...	MATLAB Data	1 KB
cam4_timestamp.mat	12/20/2014 3:09 A...	MATLAB Data	46 KB
FN055_R1_range.txt	12/24/2014 9:48 PM	TXT File	1 KB
FN055_R2_range.txt	12/24/2014 9:50 PM	TXT File	1 KB
maincam.txt	12/24/2014 9:46 PM	TXT File	1 KB
Mom1.txt	12/25/2014 5:18 PM	TXT File	230 KB
Mom2.txt	12/25/2014 4:45 PM	TXT File	206 KB
offset.txt	12/24/2014 9:46 PM	TXT File	1 KB

Figure 12. Savepath directory after tracking the 2 subjects on 2 reunions. Highlighted are the result file from interactive tracking.

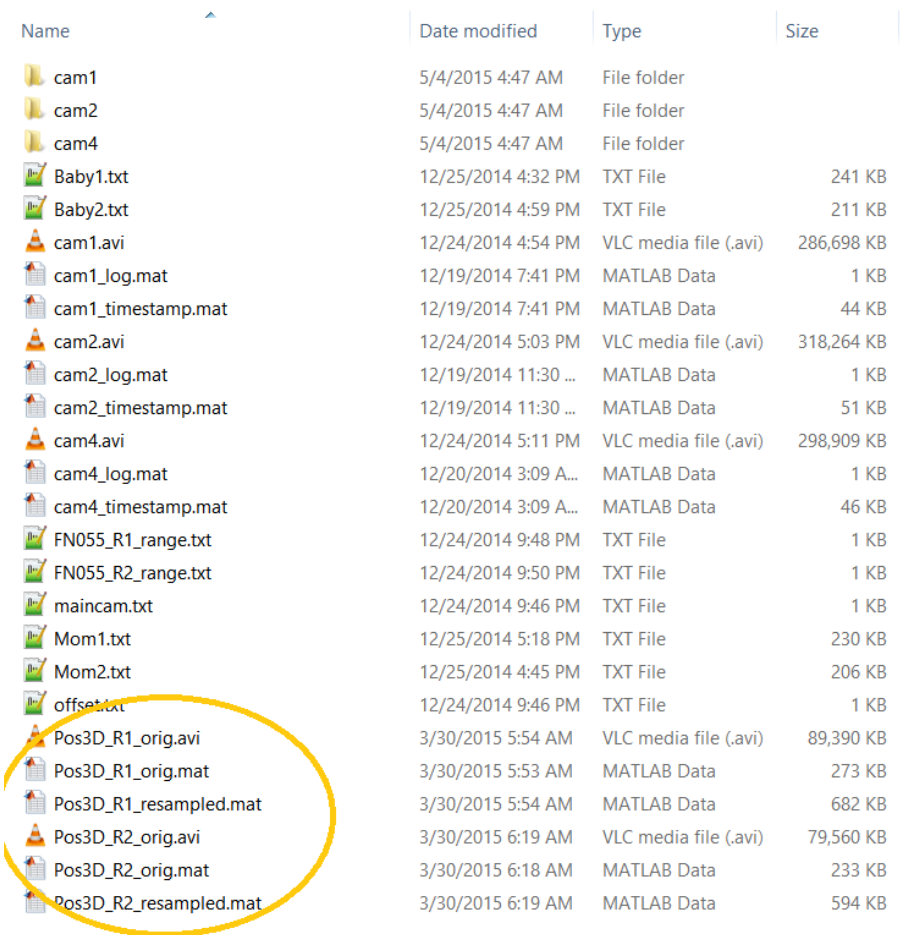
After the interactive tracking step, we now can begin the data extraction process.

Data extraction

Now, we can extract the 3d location of the objects as well as deriving the approach/avoidance data by using the result of interactive tracking. To do this, simply run the following command in matlabL

```
Extract3DPos(savepath)
```

This process will take a few hours to finish. By the end of it, you should see 3 new files for each of the reunions (2 .mat files and 1 .avi file):



Name	Date modified	Type	Size
cam1	5/4/2015 4:47 AM	File folder	
cam2	5/4/2015 4:47 AM	File folder	
cam4	5/4/2015 4:47 AM	File folder	
Baby1.txt	12/25/2014 4:32 PM	TXT File	241 KB
Baby2.txt	12/25/2014 4:59 PM	TXT File	211 KB
cam1.avi	12/24/2014 4:54 PM	VLC media file (.avi)	286,698 KB
cam1_log.mat	12/19/2014 7:41 PM	MATLAB Data	1 KB
cam1_timestamp.mat	12/19/2014 7:41 PM	MATLAB Data	44 KB
cam2.avi	12/24/2014 5:03 PM	VLC media file (.avi)	318,264 KB
cam2_log.mat	12/19/2014 11:30 ...	MATLAB Data	1 KB
cam2_timestamp.mat	12/19/2014 11:30 ...	MATLAB Data	51 KB
cam4.avi	12/24/2014 5:11 PM	VLC media file (.avi)	298,909 KB
cam4_log.mat	12/20/2014 3:09 A...	MATLAB Data	1 KB
cam4_timestamp.mat	12/20/2014 3:09 A...	MATLAB Data	46 KB
FN055_R1_range.txt	12/24/2014 9:48 PM	TXT File	1 KB
FN055_R2_range.txt	12/24/2014 9:50 PM	TXT File	1 KB
maincam.txt	12/24/2014 9:46 PM	TXT File	1 KB
Mom1.txt	12/25/2014 5:18 PM	TXT File	230 KB
Mom2.txt	12/25/2014 4:45 PM	TXT File	206 KB
offset.txt	12/24/2014 9:46 PM	TXT File	1 KB
Pos3D_R1_orig.avi	3/30/2015 5:54 AM	VLC media file (.avi)	89,390 KB
Pos3D_R1_orig.mat	3/30/2015 5:53 AM	MATLAB Data	273 KB
Pos3D_R1_resampled.mat	3/30/2015 5:54 AM	MATLAB Data	682 KB
Pos3D_R2_orig.avi	3/30/2015 6:19 AM	VLC media file (.avi)	79,560 KB
Pos3D_R2_orig.mat	3/30/2015 6:18 AM	MATLAB Data	233 KB
Pos3D_R2_resampled.mat	3/30/2015 6:19 AM	MATLAB Data	594 KB

Figure 13. Savepath directory after data extraction. Note the additional files containing the tracking results.

The Pos3D_R1_orig.avi and Pos3D_R2_orig.avi files are the videos the tracking results. The Pos3D_R#_resampled.mat files contains 16 variables:

- bInterpolationType: this is an indicator variable for each row of bPos3DResampled. 1 indicates the baby position is interpolated before the first valid datapoint, 2 indicates it is interpolated between two valid datapoints, 3 indicates it is interpolated after the last valid datapoint.
- mInterpolationType: same as above, but for mom position data.
- bPos3D: baby position in 3D (original data from kinect).
- bPos3DResampled: baby position in 3D after resampling to make sure there is a constant time interval between two datapoints (40ms between datapoints).
- mPos3D: mom (toy in the exploration protocol) position in 3D (original data from kinect).
- mPos3DResampled: mom (toy in the exploration protocol) position in 3D after resampling to make sure there is a constant time interval between two datapoints (40ms between datapoints).
- timestampResampled: the timestamp for both bPos3DResampled and mPos3DResampled (the two are perfectly synchronized since they are sampled from the same timeline)
- bTimestamp: the timestamp for the data in bPos3D (in millisecond)
- mTimestamp: the timestamp for the data in mPos3D (in millisecond)
- babyApproach: baby approach speed derived from the resampled variables (in meters/second)
- momApproach: mom approach speed derived from the resampled variables (in meters/second)
- dist3d: 3d distance between the baby and the mom derived from the resampled variables.
- bHeightFromFloor: baby's head height w.r.t. the ground plane (derived from the resampled variables)
- mHeightFromFloor: mom's head height w.r.t. the ground plane (toy's height in the exploration protocol) (derived from the resampled variables).
- isBabyCloseToMom: binary variable indicating whether the baby is close to the mom (toy in the exploration protocol). This is done by thresholding the dist3d variable. Currently a threshold of 800mm is used. (derived from the resampled variables)
- isBabyOnTheFloor: binary variable indicating whether the baby is on the floor or being picked up by the mom. This is done by thresholding the bHeight variable. Currently a threshold of 900mm (3 feet) is used since I assume the baby is less than 3 feet tall (no giant baby). (derived from the resampled variables)

Now we are done.