

We recorded a total of 3 files independently and simultaneously:
2 via “Kinect Studio” (3D data), 1 video control.

20150319_140851_00.xef

20150319_141002_00.xef

(Video file)

We transferred files to a processing station, clustered them in a folder per dyad labelled: <participantA>_<participantB>_<day>-<month>-<hhmm>

AlbertEinstein_MarieCurie_04-03_1530

AlbertMarie1.xef

AlbertMarie2.xef

AlbertMarie.mp4

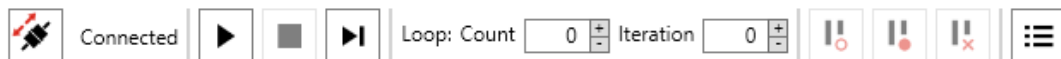
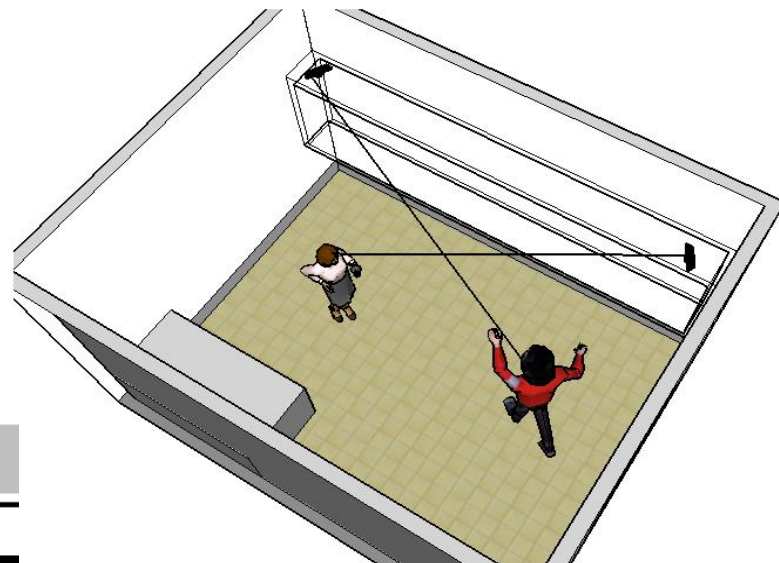
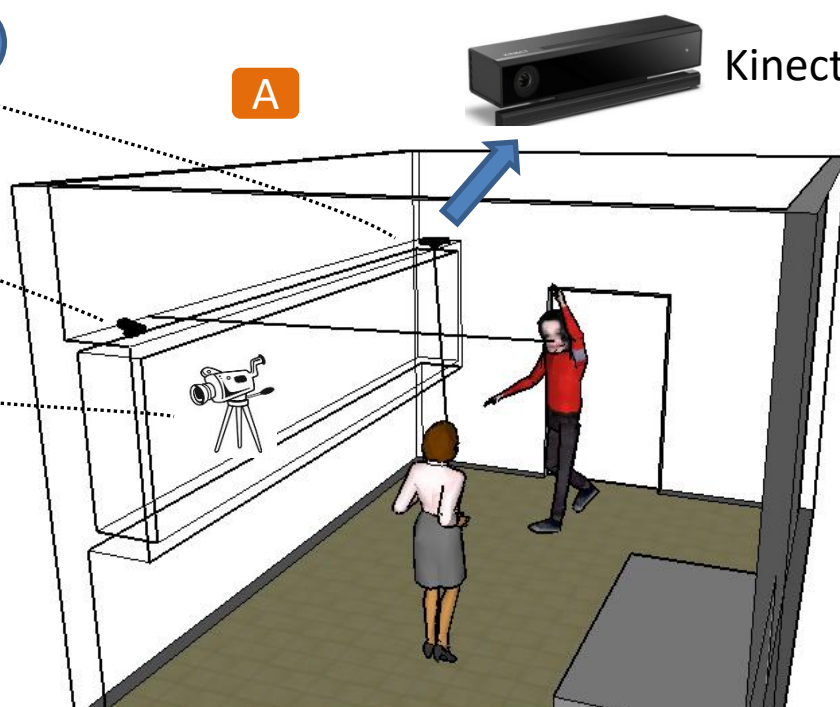


B

“Kinect Studio” was employed to *manually mark* time **sync-cue** in both files, e.g. a high five. Additionally, intervals of interest are marked by their starting and end times: **StartRound/EndRound**.

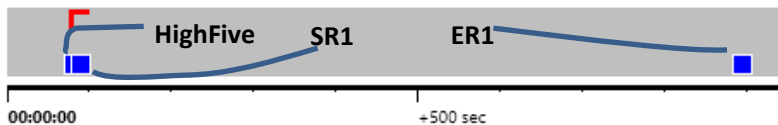
C

Kinect v2



NadavIdo2.xef [Read-Only]

0 Streams:
(0 visible, 0 hidden)



Nui Body Frame

Nui Calibration Data

Data which are acquired in two different coordinate systems are synchronized in space into a global coordinate system, by establishing a common anchoring point.

D

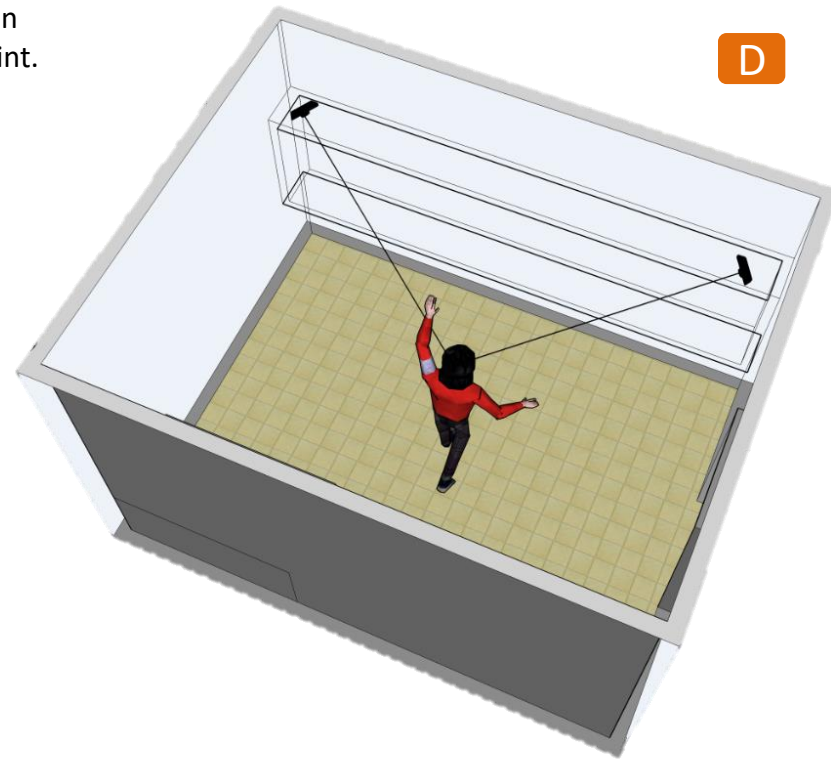
20150319_150851_00.xef

20150319_151002_00.xef

1

2

We recorded two files independently via “Kinect Studio”. In these recordings, an experimenter stands still for about 10 seconds facing the middle point between the cameras and stretching his arms sideways. This is later used to build a new coordinate system whose origin is on experimenter’s neck. From that point, a Cartesian system is formed, aligned with the stretched hand, spine direction, and the perpendicular vector to these.



AlbertEinstein_MarieCurie_04-03_1530

AlbertMarie1.xef

AlbertMarie2.xef

AlbertMarie.mp4

sync (directory)

Actual data files

AlbertMarie1.xef

AlbertMarie2.xef

We transferred files to the processing station under “sync” directory and renamed by same conventions used before.

space sync files



We extract the spatial transformations into the global coordinate system using the designated files.

E

```
private const string playbackDirPath = @"D:\Guy\Acquisitions\AlbertEinstein_MarieCurie_04-03_1530\sync\";
private const string playback1FileName = "AlbertMarie1.xef";
private const string playback2FileName = "AlbertMarie2.xef";
```

We use the files from the previous step (spatial sync) to build a new coordinate system whose origin is on experimenter's neck, and Cartesian axes are aligned with the stretched hand, spine direction and the perpendicular vector to those.

It is necessary to fix exactly 2 markers (label unconstrained) via "Kinect Studio" describing the confining interval, in which the experimentalist is tuned to the synchronization pose.

"CamTransExtractor.sln"

Running this file yields two spatial rigid transformations which allow transforming data acquired from the two independent sources into the global coordinate system.

AlbertEinstein_MarieCurie_04-03_1530



AlbertMarie1.xef

AlbertMarie2.xef

AlbertMarie.mp4

sync (directory)

AlbertMarie1.xef

AlbertMarie2.xef

spatial
transformations

Cam1toRef

Cam2toRef

We reserialize (store) the data available in the “xef” files into new tractable JSON format files. We term these “dumpfiles”.

F

```
private const string SyncCue = "HighFive";
private bool isFilterOn = true;
private static List<Tuple<String, String>> StandardSegments = new List<Tuple<String, String>> {
    new Tuple<String, String>("SR1", "ER1"),
    new Tuple<String, String>("SR2", "ER2"),
    new Tuple<String, String>("SR3", "ER3")
};

private static List<Tuple<String, String>> SpecificSegment = new List<Tuple<String, String>> {
    new Tuple<String, String>("SR3", "ER3"),
    //new Tuple<String, String>("test1", "test2"),
};

private const string playbackDirPath = @"D:\Guy\Acquisitions\AlbertEinstein_MarieCurie_04-03_1530\";
private const string outputPath = @"D:\Guy\Acquisitions\";
private static Dictionary<String, List<Tuple<String, String>>> SerializationProgram =
    new Dictionary<String, List<Tuple<String, String>>>()
{
    { "AlbertMarie1.xef", StandardSegments},
    { "AlbertMarie2.xef", StandardSegments},
};
```

“BodyBasicsPlaybackSerializer.sln”

Kinect Studio playback capability is used to run the “xef” files in the background and reserialize motion data in a lighter and tractable JSON format.

In this process data points’ timestamps are overridden by their elapsed time from the marked time-sync cue.

In addition, motion filtering techniques are employed in real time to smoothen the highly jittery signal.

Finally, this procedure focuses and processes data associated only with one body, the one facing it, and discards data belonging to other interlocutor, or misinterpreted reflections coming from the confining walls.



We make new copies of our JSON files where data is represented in the new, joint reference frame.



```
public const string Cam2ToCam1TransFileName = "Cam2toCam1";
public const string Cam1ToRefTransFileName = "Cam1toRef";
public const string Cam2ToRefTransFileName = "Cam2toRef";
// IMPORTANT: keep excerpts of same interaction adjacent in order to have correct time sync.
private static List<List<String>> data = new List<List<String>>() {
    new List<String>() {
        @"D:\Guy\Acquisitions\AlbertEinstein_MarieCurie_04-03_1530\",
        "AlbertMarie1_dumpfile_SR1_ER1",
        "AlbertMarie2_dumpfile_SR1_ER1",

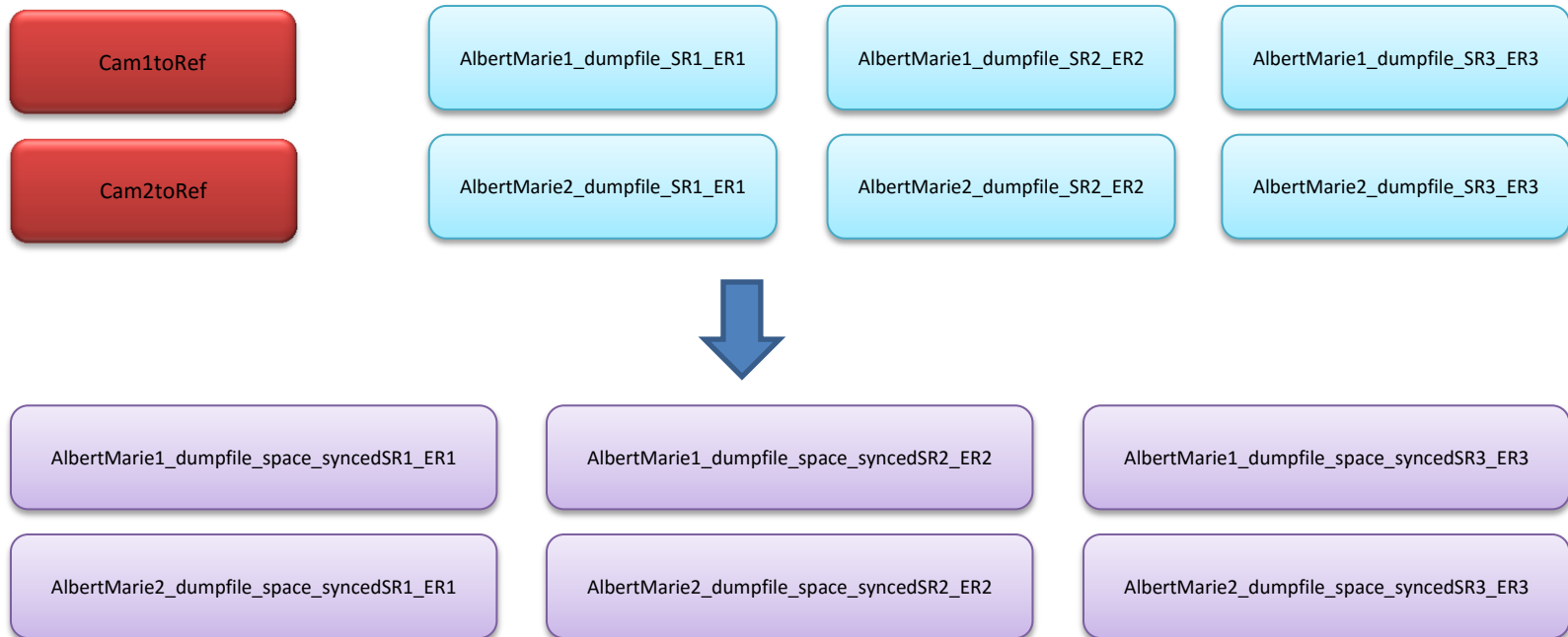
        "AlbertMarie1_dumpfile_SR2_ER2",
        "AlbertMarie2_dumpfile_SR2_ER2",

        "AlbertMarie1_dumpfile_SR3_ER3",
        "AlbertMarie2_dumpfile_SR3_ER3",
    }
};
```

"CoordinatesTransformer.cs"

This procedure outputs the same data type generated by the "serializing" procedure, but in the new reference frame created before.

It requires as input the JSON serialized files and the transformation files.



We save the data into CSV files



```
private const string playbackDirPath = @"D:\Guy\Acquisitions\AlbertEinstein_MarieCurie_04-03_1530\";
// IMPORTANT: keep excerpts of same interaction adjacent in order to have correct time sync.
private static List<List<String>> data = new List<List<String>>() {
    new List<String>() {
        @"D:\Guy\Acquisitions\AlbertEinstein_MarieCurie_04-03_1530\",
        "AlbertMarie1_dumpfile_space_syncedSR1_ER1",
        "AlbertMarie2_dumpfile_space_syncedSR1_ER1",

        "AlbertMarie1_dumpfile_space_syncedSR2_ER2",
        "AlbertMarie2_dumpfile_space_syncedSR2_ER2",

        "AlbertMarie1_dumpfile_space_syncedSR3_ER3",
        "AlbertMarie2_dumpfile_space_syncedSR3_ER3",
    }
};
```

"CSVMaker.sln"

Input: the synchronized JSON files.

Output: csv files.

*One can also run the procedure on the non-space-synced JSON files if synchronization is not required.

AlbertMarie1_dumpfile_space_syncedSR1_ER1

AlbertMarie1_dumpfile_space_syncedSR2_ER2

AlbertMarie1_dumpfile_space_syncedSR3_ER3

AlbertMarie2_dumpfile_space_syncedSR1_ER1

AlbertMarie2_dumpfile_space_syncedSR2_ER2

AlbertMarie2_dumpfile_space_syncedSR3_ER3



CSV files

AlbertMarie1_space_syncedSR1_ER1

AlbertMarie1_space_syncedSR2_ER2

AlbertMarie1_space_syncedSR3_ER3

AlbertMarie2_space_syncedSR1_ER1

AlbertMarie2_space_syncedSR2_ER2

AlbertMarie2_space_syncedSR3_ER3