MotionBank: Kinect Setup
Recording Performances of Jonathan Burrows & Matteo Fargion

Fraunhofer IGD, Svenja Kahn, 05/2012

Introduction

In May 2012, Jonathan Burrows and Matteo Fargion performed selected parts of their dances "Both Sitting Duet", "Speaking Dance", "Counting to 100", "The Quiet Dance", "The Cow Piece" and "Cheap Lecture" at the LAB of the Forsythe dance company in Frankfort, Germany. These performances were recorded as part of the MotionBank research project. Furthermore, the recorded performances are analyzed and processed in this research project.

In order to capture the movements of Jonathan Burrows and Matteo Fargion, the dances were recorded with Kinect cameras. Then, the movements were estimated with the Microsoft Kinect SDK. This research was made possible by the Microsoft Connect / Kinect for Windows Testing and Adoption program, which provided the possibility to record Kinect data first and then to estimate the human poses with the Microsoft SDK skeleton tracking in a post processing step.

Camera Setup

To capture the performances of Jonathan Burrows and Matteo Fargion, we used a multi-camera setup of two Kinects and several 2D HD camcorders. The first Kinect was used to capture the movements of Jonathan Burrows, the second one to capture the movements of Matteo Fargion. Two 2D camcorders were used to capture HD close-ups of both Jonathan Burrows and Matteo Fargion; the other ones were used to capture sequences of the whole stage. Amongst others, we use the latter recordings to reconstruct 3D positions (for example of the heads) from the recorded 2D image streams via reconstruction by triangulation.

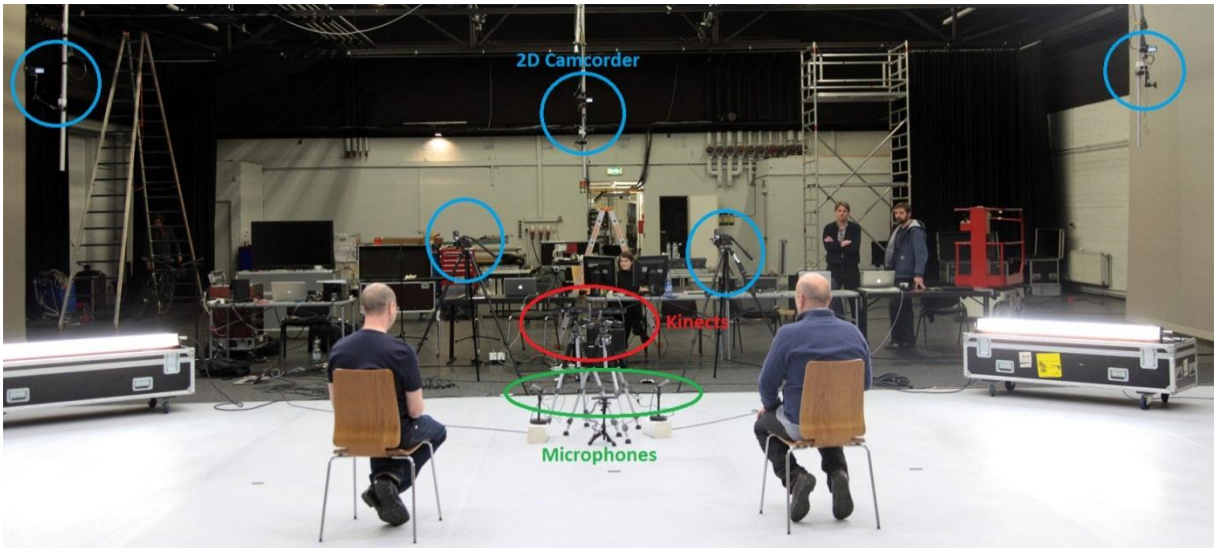

Figure 1: Camera setup for capturing dances of Jonathan Burrows & Matteo Fargion (May 2012).

Spatial Calibration (Camera Pose Estimation)

For the estimation of the camera poses (position and orientation), we place spherical markers on the stage and measure their positions with a laser pointer. The center of a marker which is visible in a camera image provides a 2D-3D correspondence between a 3D point in the world and its projection onto the 2D camera image. Then, the position and orientation of each camera is calculated from the set of 2D-3D correspondences which are visible in the camera image.

The markers are placed on the stage each time the position or orientation of a camera was changed for adapting the filming setup to a different dance. Then, a snapshot is taken with each camera and the markers are removed before the performances are recorded.

For the Kinect cameras, the markers are only captured with the color camera, not with the depth camera. The reason for this choice is that capturing the markers with the depth camera (or, more precisely, the infrared camera) would require covering the projector and illuminating the stage with light which is visible for the infrared camera. This would pose the risk that the Kinect cameras might slightly be shifted when covering or uncovering the projector. Furthermore, such a large stage cannot be illuminated easily with infrared light in a spectrum visible for the infrared camera.

This is why we capture the markers only with the Kinect color cameras. Then, the pose of a Kinect depth camera is calculated from the pose of the color camera by the relative transformation between the color and the depth camera of a Kinect (which was calculated in an offline calibration procedure, similar to the estimation of the intrinsic parameters of the depth and color cameras).

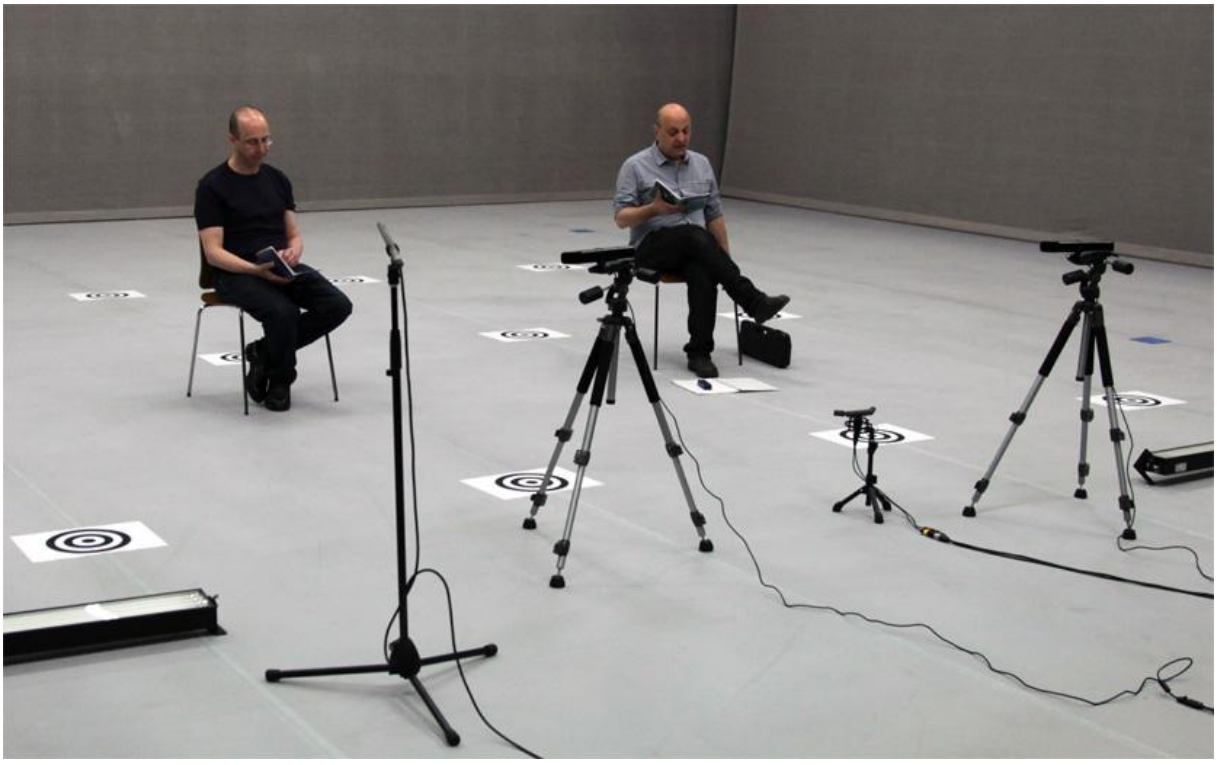

Figure 2: Spherical markers for camera pose estimation.

Temporal Synchronization

A clapperboard is used to synchronize the 2D cameras, the sound recording and the Kinect cameras. The sound and the cameras are synchronized by detecting the frame in which the clapperboard is shut for each captured video stream and audio recording.

The 2D cameras capture video sequences with a guaranteed, constant frame rate. This is why all the frames captured by all 2D cameras are automatically synchronized for a recorded take if the captured videos are synchronized at a single time instance. In contrast to the 2D camcorders, the Kinect cameras capture sequences with varying frame rates. However, their recordings can be synchronized with the other Kinect and with the 2D cameras by the time stamp which is stored with each captured image, in combination with the known common frame in which the clapperboard was shut.

Figure 3: Clapperboard used for camera synchronization.

In addition to the clapperboard, we also use a QR code which encodes the current date and time (in milliseconds) of a reference system. A short video or a snapshot of the QR code displaying the reference time was captured with each camera. By detecting the QR code in the captured videos, all recordings can be temporally aligned to the common reference time.

Figure 4: QR code used for camera synchronization.

Motion Capture Results (Without Occlusions)

Figure 5 and Figure 6 visualize tracked poses of Jonathan Burrows and Matteo Fargion, estimated with the Microsoft Kinect SDK.

In the dance “Both Sitting Duet”, Jonathan Burrows and Matteo Fargion sit on a chair. Nevertheless, the skeleton tracking works quite well in the default skeleton tracking mode (actually much better than in the “seated” skeleton tracking mode). This is why we consistently use the default skeleton mode to estimate the movements of Jonathan Burrows and Matteo Fargion with the Kinect SDK.

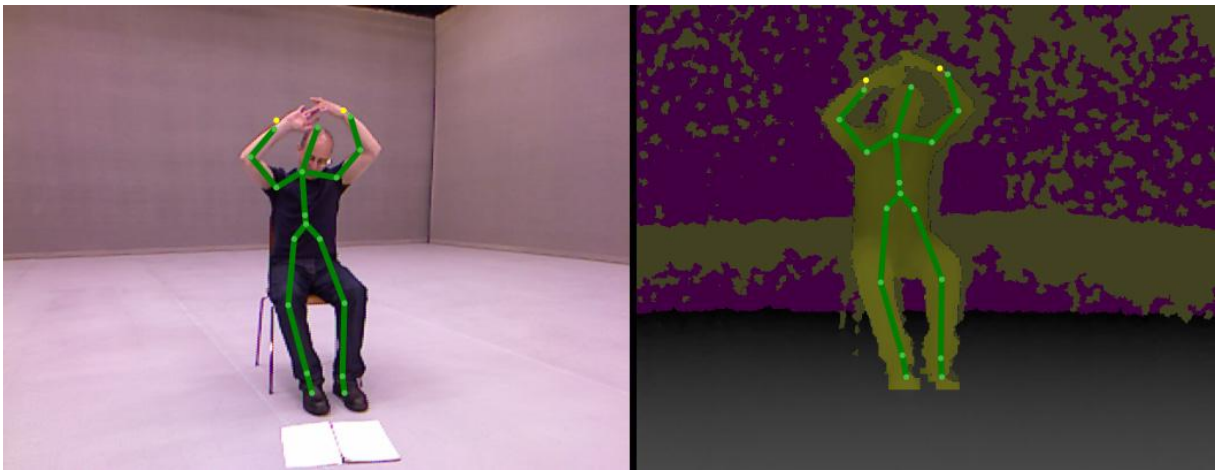

Figure 5: Motion capture of Jonathan Burrows in the dance “Both Sitting Duet” (estimated with default=standing skeleton mode).

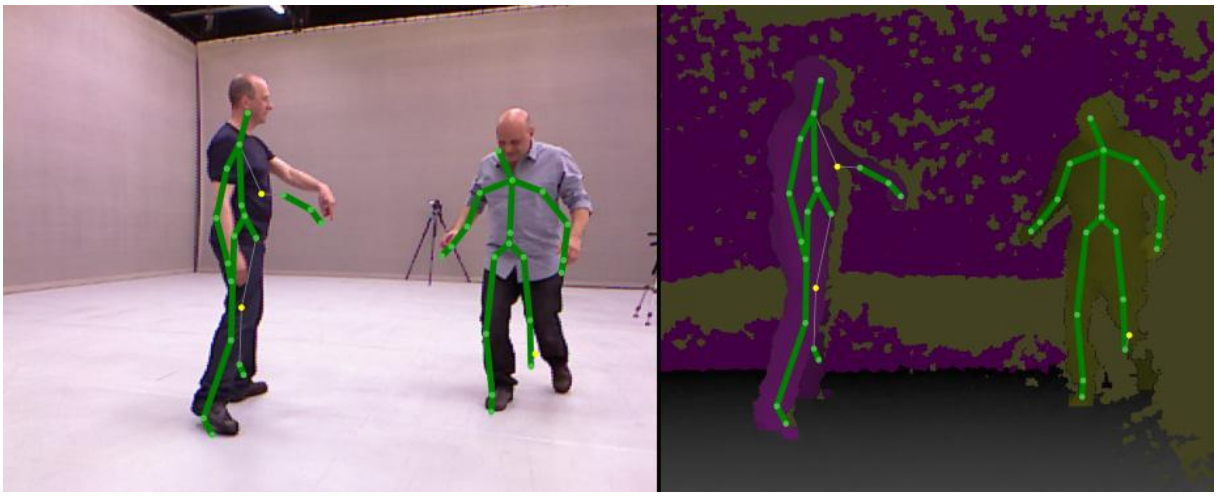

Figure 6: Motion capture of Jonathan Burrows and Matteo Fargion in the dance “Quiet Dance”.

Motion Capture Results (With Occlusions)

The next two images visualize tracked poses in situations, in which part of the body was occluded by another object.

In the dance “The Cow Piece”, Jonathan Burrows stands behind a table. Initially, we were a bit worried that this might cause problems for the skeleton tracking. However, if the Kinect is positioned such that the depth camera records the depth parallel to the table, the skeleton estimation is surprisingly stable, even in spite of the partial occlusion. The tracking is stable as long as the knees are not occluded by the table, which can easily be avoided by positioning the Kinect parallel to the table.

The dance “Cheap Lecture” poses more difficult challenges for the skeleton tracking in view of occlusions: In this dance, Jonathan Burrows and Matteo Fargion hold a sheet of paper in front of their body. In most frames, the Kinect skeleton tracking mistakes the sheet of paper for the arms and thus cannot estimate the arm positions correctly.

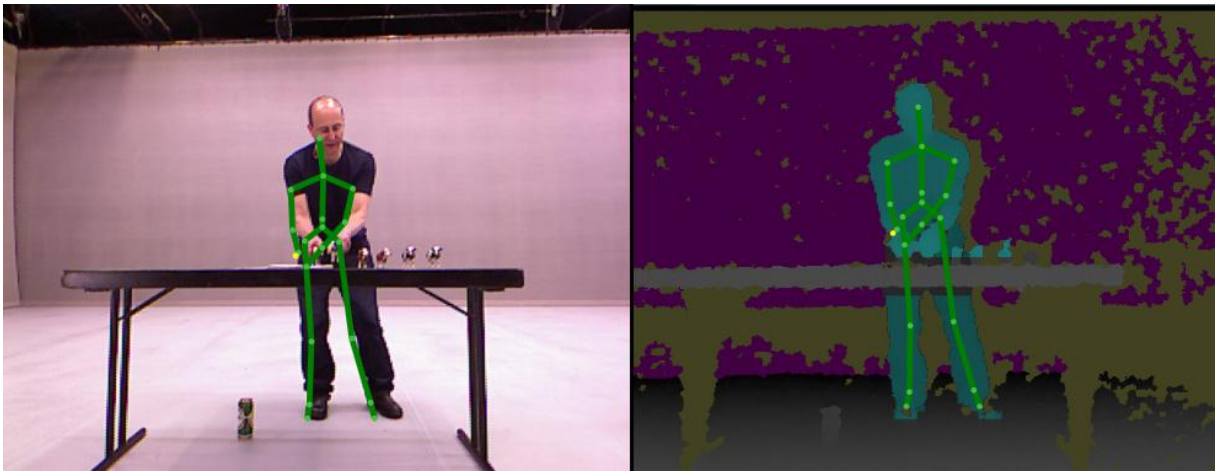

Figure 7: Motion capture of Jonathan Burrows in the dance “Cow Piece”.

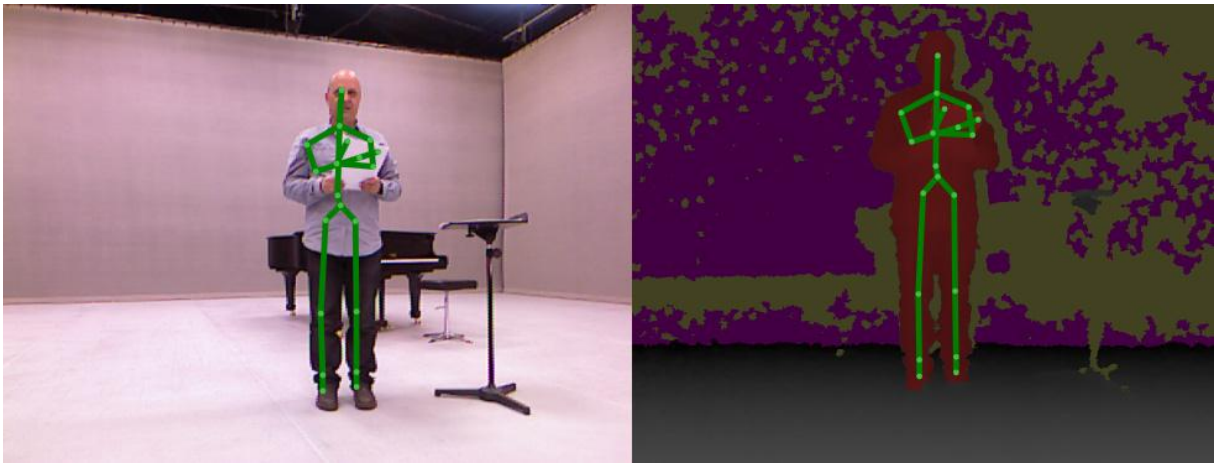

Figure 8: Motion capture of Matteo Fargion in the dance “Cheap Lecture”.

Conclusion & "Wish List" for Post Processing Skeleton Tracking with the Microsoft SDK

Kinect skeleton tracking with the Microsoft SDK provides great means to capture and to analyze the movement of persons, such as the movements of Jonathan Burrows and Matteo Fargion in their modern dance performances. The Kinect Studio makes it possible to record the performances first and then to estimate the movements of the recorded performances in a post processing step.

Currently, the approach of recording movements first with Kinect Studio and then analyzing them in a post processing step has two limitations:

1. Kinect Studio replays the recorded sequences with the same speed as they were recorded (about 28-30 frames per second). However, the skeleton tracking is calculated with a lower frame rate than the sequences were recorded (about 20 frames per second). As Kinect Studio does not "wait" for the skeleton tracking to complete until it outputs the next frame, about one third of the captured frames gets lost (which means that no skeleton pose is estimated for these frames).
    - => It would be great to have some means to estimate the skeleton for each captured depth image (for example by slowing down the playback speed such that no frames get lost, or by an explicit call to "please get the next frame").

2. As far as we know, currently it is not possible to change the depth image before the skeleton tracking is estimated. We believe that this would provide the possibility to enhance the accuracy of the skeleton pose estimation with self-written algorithms, for example if some part of the tracked person is occluded.
    - For example, then we could write an algorithm for detecting the sheet of paper which occludes Matteo Fargion in Figure 8 and then replace each depth measurement in this part of the depth image with an estimation of the distance of this pixel to his upper body. This would make it possible to track the poses of his arms in spite of the sheet of paper.
    - We have implemented such approaches (which change the depth image before the skeleton is estimated) in OpenNI. With these depth image adjustments, it becomes possible to track the poses of persons in recorded sequences that cannot be correctly analyzed otherwise. For example, we followed a dancer moving around a large stage and recorded her movements with a hand-held Kinect depth camera. Then we removed the ground floor and the background walls from the recorded depth sequence and stored the changed depth sequence in a new file. Whereas it was not possible to estimate the skeleton of the dancer in the original file, this was possible with the new file, which contains the changed depth images.
    - => It would be great to have some similar way to change the depth images (which are input into the skeleton tracking) with the Microsoft SDK, too.