MotionDraw: A Tool for Enhancing Art and Performance Using Kinect Technology

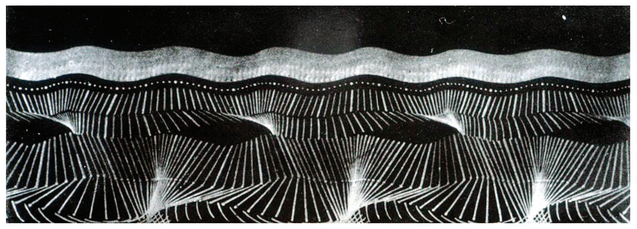
Danilo Gasques Rodrigues, Emily Grenader, Fernando da Silva Nos,
Marcel de Sena Dall'Agnol, Troels Ellegaard Hansen, Nadir Weibel

MotionDraw is a Kinect-based system that enables professional performers, students, and at-home users to create easily controlled visual interactive displays to be shown with their own dance performance.

1. Motivation

Inspired by the way photography began as a scientific tool, and later became an art form, we wanted to explore how the Kinect tool could be used to enable art and performance. In his photograph from 1884, shown below, Etienne-Jules Marey tracks a man's motion using multiple exposure photography. Even though the rationale of creating this photograph was a scientific study of motion, it is evident that the result is also an aesthetically interesting image.

To aid us in the design of our application and to develop our idea, we conducted small research interviews with seven dancers from UCSD's dance department. In order to validate our idea and ensure that our project would indeed be useful for performers, we conducted small research interviews with seven dancers from UCSD's dance department. We asked the dancers about their prior experience with the combination of dancing, projection and computing. All interviewees had seen large-scale performances that used technology and/or projection to enhance the performance, but none had been given the opportunity to work with computing or projection themselves, even though most of them expressed interest in doing so. Given that dancers usually have very little control over what is happening on the stage, we saw this as further motivation to develop an application that would be easy to use for performers, therefore changing the established system. From the beginning our ultimate goal was to enable professional performers, students, and at-home users to create visual interactive effects that they can easily control with intuitive gestures.



Chronophotographic Image Formed by Man Walking in Black Suit with White Stripes, Etienne-Jules Marey, 1884

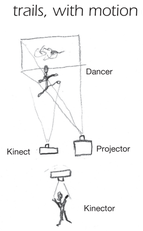
2. Related Work

Previous projects use visual displays in order to augment performance. For example, dancer and choreographer Merce Cunningham frequently worked with digital artists and musicians to place dancers into dynamic visual environments. In Hand-drawn Spaces, from 1998, Cunningham worked with the OpenEndedGroup to create displays of multiple abstract figures across many screens using motion capture. Since the release of the Microsoft Kinect device, there have been several efforts towards integrating art and motion-tracking technology. Super Mirror brings the low-cost possibility of using the Kinect into dance instruction, functioning as an augmented mirror. Alexiadis et al used Kinect technology for a similar project that provides real-time evaluation of dancers with visual feedback, and also supports evaluation of one dancer's performance against another. Both of these projects have instructional or evaluational purposes, and are not aimed at performances. Kinect has also been integrated into professional performances as in .cyclic, done in collaboration between University of Waikato and Stellaris dance in New Zealand, where choreographers and computer scientists joined efforts to create an interactive performance. This project depends highly on each individual collaborator's ability to complete a task, and remains inaccessible to everyday users.

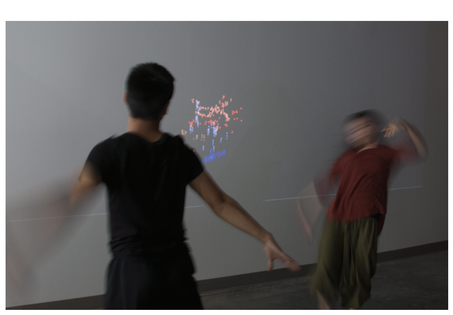
"Apparition" by Klaus Obermaier and Ars Electronica Futurelab

3. MotionDraw System

MotionDraw consists of a set-up with one or two Kinects, that record the position of joints of a person's body over time and draws the resulting trail on a 3D canvas. With a two-Kinect set-up, one Kinect points at the dancer while the other captures a "conductor" we named the kinectTor. The kinecTor is able to change the parameters of the projected drawing, such as color and width of the trails, with motion gestures. As soon as a defined gesture is recognized, the kinecTor software communicates with the MotionDraw software (on a separate computer, over a network) about the changes that should be applied. This integration is covered in more details below. With a one-Kinect set-up, MotionDraw provides a GUI-based control panel that enables a second user to perform the same controls as the kinecTor. The program also saves the Kinect stream to a file for further post-production.

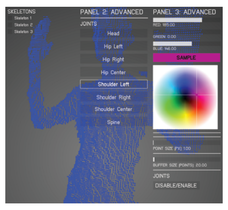


In a demo run of the system developed thus far, two dancers, Tony Ho and Abbie Wang, volunteered to perform for a few people in a short presentation, with a set-up containing music sung by our kinecTor, Fernando de Silva Nos, and a projector for visual effects. Although the event was informal, MotionDraw did indeed provide the performance with an additional level of interaction and visual effects, which both the audience and the dancers found to be pleasing.



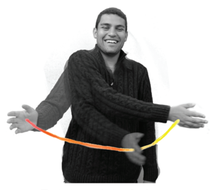
4. Technical Implementation

MotionDraw is designed to be simple and extensible. In our system, users are able to quickly learn how to control the interface. Its extensibility has two dimensions. The first is its file exporting interface, where a MotionDraw file format is used to save all data received from the Kinect device. This data can be used later by 3D modeling software. The second dimension relates to its ability to communicate with other softwares, namely the kinecTor users.



User Interface

The user interface is built using the OpenFrameworks framework, specifically, using a plugin named ofxUI. It is composed of one window, on which it is possible to switch from a Viewer perspective to a Control Box perspective, and vice-versa. A 3D drawing is presented in the Viewer window and users have options to select different Kinect devices in the Control Box window. Here, users are able to change options such as color, width, buffer size and visibility, for all skeleton joints. The 3D drawing on the viewer window is achieved by processing data received from a Kinect device using OpenGL.



Rotate point-of-view

Zoom in and out

Change paint color

Kinector

It is important to point out that the kinecTor, as developed by us, has involved several design decisions that need not be followed in different instances of the mtionDraw application. It is better described as a module that communicates with the main motionDraw application. We chose to work on one possible realization of this module: a program that uses a Kinect to read simple gestures and changes parameters in the main aplication accordingly (e.g. circular motion with the right hand changes the color of the trails, likewise with the left hand to change their width). Also, this communication is unidirectional, with no direct line from performer to conductor other than the performance itself. This is a design decision that was taken in order to present a working prototype of the system, but need not be so. There is no particular reason, other than keeping with the hands-free motion tracking paradigm, why the kinecTor could not be implemented as a smartphone app or a simple point-and-click computer interface (with bidirectional communication, possibly).

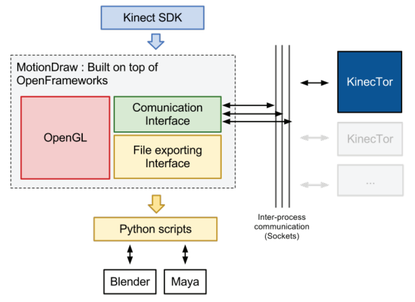
The GUI used in the one-Kinect set-up serves as a control panel from which any tunable parameter of the drawing may be accessed and modified; any kinecTor module may only be equally or less flexible. However, we envision situations where it may be desirable to compromise flexibility for ease of use (as in the kinecTor developed by us).

Post-Processing

MotionDraw allows the user to create a self-contained recording, enabling working on a recorded performance offline, using a log of tracked movements as recorded by the Kinect. In this way, MotionDraw makes it possible for the user to make use of different softwares for post-processing purposes, such as importing the recorded data into tools, like Blender or Maya. To this end, we defined a common file format and developed a Python script to convert this format into *.obj files, which may then be imported into the previously mentioned software, or any other that supports reading those files. Furthermore, more functionality may be added to the scripts; for instance, we developed basic filters, averaging points to reduce noise. This interface is flexible and extensible in the kind of processing that can be applied. A user perhaps would like to change the type of filter (e.g. median) or run different algorithms (i.e. clustering).

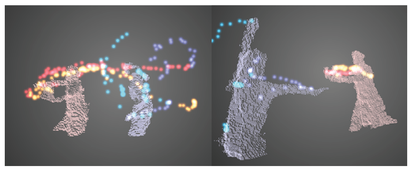
Inter-Processing Communication

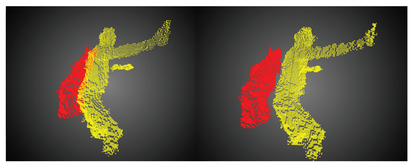
In order to support two, or more Kinect devices interacting with MotionDraw through the kinecTors, we developed a general Inter-Process Communication (IPC) interface that enables any outside applications (Kinect or non-Kinect based) to control the way MotionDraw processes the user's inputs. We based the IPC implementation on sockets and dedicated messages passed across machines interconnected through a network.



MotionDraw Architecture

The Kinects communicate directly with MotionDraw (the Kinect tracking the dancer) or over an IPC interface (one or more kinecTors, on the right). The OpenGL interface (on the left) is responsible for the graphical representation that is projected on the stage. MotionDraw also supports directly exporting the tracking of the joints to external tools such as Blender and Maya for post-processing, through dedicated Python scripts (bottom).

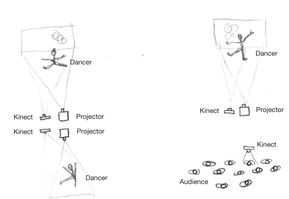




5. Future Work

We would like to further develop the graphical and interactive functionality in motionDraw. Some features we would like to implement are:

- Interactivity among dancers (e.g. touch of hands) as a trigger for changes in the display.
- Interacting with the actual projection - touching the projected drawing and influencing the projection.
- Relate the position of the projection to the body's position in order to create a shadow or completely cover the performer with the drawing.
- Improve real-time graphics with filters or point cloud effects.



Different set-ups with the audience as the kinecTor or two dancers as each other's kinecTors

6. Check us out

Email Us:

Danilo Gasques Rodrigues: danilogr@comp.ufscar.br
Emily Grenader: egrenade@ucsd.edu
Fernando da Silva Nos: fernando.nos@acad.pucrs.br
Marcel de Sena Dall'Agnol: marceldalagnol@gmail.com
Troels Ellegaard Hansen: troe@itu.dk
Nadir Weibel: weibel@ucsd.edu

Acknowledgements

We would like to thank people who were directly involved with our project: all of the dancers we interviewed especially Tony Ho and Abbie Wang, as well as Microsoft that donated the Kinect, the Experimental Drawing Studio, the Department of Computer Science and Engineering and the Moxie Center at UCSD, and the Brazilian Science Without Borders program, for providing us with the necessary support to begin development of this project.

MotionDraw Website

http://motiondrawstudio.com

MotionDraw Demo

http://vimeo.com
user: Motion Draw

1. M. W. Scott, et al. "Mixing dance realities: collaborative development of live-motion capture in a performing arts environment." Computers in Entertainment (CIE) 2.2 (2004): 12-12. 2. M. Raptis, D. Kirovski, and H. Hoppe. "Real-time classification of dance gestures from skeleton animation." Proc. SCA'11. 3. M. Zoe, et al. "Super Mirror: a kinect interface for ballet dancers." Proc. CHI'12. 4. D. S. Alexiadis, et al. "Evaluating a dancer's performance using kinect-based skeleton tracking." Proc. Multimedia'11. 5. S. Sentürk, et al. "Crossole: A Gestural Interface for Composition, Improvisation and Performance using Kinect." 6. D. Jung, et al. ".cyclic.: interactive performance combining dance, graphics, music and kinect-technology." Proc. NZ Chapter of the ACM's SIGCHI, 2012.