Layering the Choreographic Process: Making Dance Work with Machine Learning

Kate Sicchio, PhD
New York University
Tandon School of Engineering
Brooklyn, NY 11237, USA
sicchio@nyu.edu

Proceedings of MICI 2017: CHI Workshop on Mixed-Initiative Creative Interfaces.

Abstract
This paper discusses a new performance by the author that was made in part by machine learning algorithms. Working with the t-SNE algorithm to visualize data, a choreographic score can become performed through a layer of images, live coding and an improviser performer. The final performance aims to produce new possibilities for live performance through using code to traverse clusters of media that the algorithm has produced.

Author Keywords
Choreographic score; machine learning; dance improvisation; t-SNE

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous

Introduction
When working with dance and technology it is common for work to be developed where the dancer and their movement is used to reflect the technology, or even demonstrate the technology. But when choreographers use machine learning techniques, there are interesting results when creating layered approaches, or mixed-

initiative creative interfaces to influence the movement vocabulary in new ways.

For example, Wayne McGregor has worked with the Choreographic Language Agent, an AI agent used to augment his process since 2004. The one iteration of this was *Becoming,* a tool developed with Marc Downie and Nick Rothwell. The aim is to provoke new movement creation in the dance studio, which is later shaped by McGregor to become a layer process in creating a performance output. Both of these pieces rely on the computation as part of the choreographic practice. This layered approach does not celebrate the computer as the choreographer but simply uses it in a way to inspire or create new movements that may be have been part of the choreographer or dancer's cannon.

Untitled Algorithmic Dance 2

Untitled Algorithmic Dance 2 is a dance performance where the choreography is generated through a series of interactions between humans and machines. It is a layered approach that creates a system with various places for interpretation, by both people and algorithms. This is no simple mapping of input to output, but several processes that layer to become a dance.

The first layer in this process is human. A dancer improvises to create movement. This becomes the content that is layer used to produce a choreographic score for the performance. Some of the movement material is just warming up, some is dancing for the sake of movement, some is purely gestural such as me grabbing a water bottle, and some responds to the prompt of creating movement for a camera. For now all of these types of movements remain in the piece, in the form of photographs.

The second layer is the camera. Time lapse photography was used to capture the original movement. During this process a GoPro photographed two photos per second. Over 3000 images from a dance improvisation session were captured. Some photos were clear shapes, others are bizarre half moments of movement. The movement is transformed in this layer without any machine learning just by trying to make it static.

Figure 1: Example of clustering of images as performed by the t-SNE.

The next layer is the actual layer of machine learning using Gene Kogan's code for the t-SNE algorithm (originally by van der Maaten) in OpenFrameWorks to in effect cluster the images. This is an untrained machine learning algorithm used to visual large amounts of data. The algorithm finds the similarities between

photos and places them into a grid. What you see right away is a new dance. A score made just through recognition and reorganization.

However, one does not have to simply use this grid, and certainly using the grid as a score is only one possibility with this re-imaged plot of movement. Another human layer becomes part of the system within this piece. The navigation of the clusters of captured movement is done live by the choreographer. In this first version of the piece it is very simply done through navigating the grid with a mouse. The choreographer controls the path of the score, the timing of the score and most importantly can respond to the dancer performing the score.

Which brings me to the final layer of the system, another human. There is an interpreter at the end of this system, which is a dancer. The dancer sees the image that the choreographer has currently selected and responds. For this performance, the dancer has agency to respond however they like – they may copy the shape of the body in the image, or make a contrapuntal shape. They may move in a direction indicated by the image or move in the opposite way. The dancer has the final say in the movement being generated. The human dancer has the ultimate control.

And while this layered approach works in creating a composition that we all recognize as a dance piece, it is infused throughout with new tools to push the choreographic possibilities of a single choreographer. This resonates with McGregor in that it is an influence on the movement and not the full development of the movement. Perhaps this also demonstrates ways of working with machine learning that are still human centered and uses this process as just a tool.

Conclusions

Untitled Algorithmic Dance 2 explores the use of the t-SNE algorithm to create live choreographic scores from a collection of time lapse photos. It works with a complex layering of human and machine interactions to create a performance in which machine learning influences the final piece. Through this process an interruption of a choreographic process via AI has begun and will be developed further with different initiatives such as motion capture.

References

1. Kate Sicchio. 2014. Hacking Choreography: Dance and Live Coding. Computer Music Journal. Vol. 38, Iss. 1 (March 2014) pp31 – 39.

2. Studio Wayne McGregor. 2004. Choreographic Language Agent. http://waynemcgregor.com/research/choreographic-language-agent

3. Laurens van der Maaten. 2014. Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research 15 (2014) pp 1-21.

4. Gene Kogan. 2016. https://github.com/genekogan/ofxTSNE