Permuting connections: software for dancers

DeLahunta, S.

Original citation & hyperlink:
DeLahunta, S. (2008) 'Permuting Connections: Software for Dancers' In Paul D. Miller (Aka DJ Spooky)(Ed). *Sound Unbound: Sampling Digital Music and Culture* (pp: 265-272). Cambridge, Massachusetts, USA: MIT Press.
http://mitpress.mit.edu/books/sound-unbound

No. 11

‘Permuting Connections: Software for Dancers’

in *Sound Unbound - an anthology on Sound Art, Digital Media and new Compositional Strategy*. ed. P. D. Miller. Cambridge: MIT Press. 2008, pp. 265-272.

In March 2002, I was invited by Paul D. Miller (aka Dj Spooky) to write a chapter for his planned anthology on ‘Sound Art, Digital Media and new Compositional Strategy’. He had seen a presentation I gave that involved some of the human figure animation work of Michael Girard’s [see Critical Appraisal footnote 12] and was interested to see me include some of the “movement and flocking” material I showed in my lecture with a “little historical overview”.[1] To construct the historical background I thought best for Girard’s work, I reused some of the material from the ‘Periodic Convergences: Dance and Computers’ [No. 8] at the start.

The book was initially being published by Routledge, but something disrupted that plan and it took six years before eventually being published by MIT with the support of Doug Sery, well-known acquisitions editor for new media related topics at MIT Press. The list of other authors in the publication include: Manuel DeLanda, Cory Doctorow, Frances Dyson, Brian Eno, Moby, Hans Ulrich Obrist, Pauline Oliveros and Bruce Sterling.

[1] Email to the author, 18 Mar 2002.

Permuting Connections: Software for Dancers 25

Scott deLahunta

One can trace connections between computers and dance back to the 1960s when early computer artists, often mathematicians and computer scientists by training, were experimenting with algorithmically generated graphic images and patterns and forms of computer creativity. Working with Bell Labs in Murray Hill, New Jersey, A. Michael Noll began to explore the possibility of combining digital computers and the visual arts by studying three-dimensional computer graphics and computational aesthetics. In a crude approximation of the Turing test in which human and machine intelligence are compared, Noll invented the algorithms that would instruct a computer to generate an image that would mimic in its patterns and structure Piet Mondrian's *Composition with Lines* (1917).[1]

In 1965, Noll created a work of computer animation he titled *Computer-Generated Ballet*, reported to be the first such use of a "digital computer to create an animation of stick figures on a stage."[2] But perhaps his most significant contribution to the convergence of computers and dance was in January 1967 when he published an article in *Dance Magazine* entitled "Choreography and Computers," in which he described a software program he was creating that would indicate stage positions of stick figures and could potentially be of use to choreographers. In the same issue, Ann Hutchinson-Guest (an authority on dance notation) penned "A Reply" to Noll's speculations, in which she writes that the computer will "never replace" the facility a choreographer has for composing movement with the dancer. However, she does concede that the computer might assist in the overall outlining and editing of a score for a dance.[3]

Around the same period, John Lansdown, an architect by training, was pursuing a different vision of integrating dance and the computer. Based in London, Lansdown was particularly interested in the possibilities for "artificial creativity," to use the computer to contribute to a creative process as an autonomous composer, rather than to support or augment an existing one. In his introduction to "Artificial Creativity," a paper given in 1995, Lansdown describes the computer's ability to make "decisions according to rules." He traces the history of the use of related "regulatory" systems in music composition, architecture, and painting and distinguishes between two types: those that are randomized and those that are rule-based. Contemporary choreographers have used similar systems.[4] Merce Cunningham's and John Cage's well-known experiments with random methods were explored further by experimental choreographers in the early 1960s. In the 1970s Trisha Brown devised "dance making machines"—rule-based systems that generated particular performances such as *Accumulation* and *Locus*.[5] William Forsythe's use of algorithmic structures in the 1990s with the Ballet Frankfurt is well documented.[6]

Back in 1968, Lansdown had begun to experiment for the first time with "computer-generated" dances. He first attempted to use the computer to create all the instructions a dancer would require, but soon determined that "a more satisfactory method" was to provide a looser framework within which there was some room for interpretation by the dancers.[7] He developed the concept of generating "peaks" of movements rather than the movements themselves and allowed the dancers to fill in the material between. Using these methods, Lansdown contributed to many performances between 1968 and 1993 with various dance companies, including London-based Another Dance Group, the Royal Ballet School, and The One Extra Company of Sydney, Australia.

In the 1960s and 1970s access to computers was extremely limited and programming a slow tedious process. A. Michael Noll states optimistically in his 1967 *Dance Magazine* article: "The computer and graphic output equipment might be centrally located and time-shared with many users. Anyone could apply this technology to produce this form of 'dance notation typewriter.'"[8] Perhaps Noll thought the conditions he describes would generate more convergence between computers and dance. It did, but not surprisingly it was choreographers working at academic institutions with access to computer science departments who were best able to explore the possibilities, and the mainstream of contemporary dance practice tended to be unaware of, or uninterested in, the outcomes of this work. Computing aids (either generative or supporting) for choreographic compositions have not proliferated to a large degree as perhaps Noll and Lansdown would have predicted. The piece of technical equipment that has become ubiquitous in the rehearsal studio is clearly the video camera and television monitor, but there has been little incorporation of computer technologies into this setup.[9]

In the 1980s, we saw the emergence of "interactive performance systems" such as Canadian artist David Rokeby's *Very Nervous System* (VNS).[10] The VNS uses a video camera as an "eye," the cable to the computer as an "optic nerve," and the computer as the "brain" to create an interactive "seeing" space in which the movements of one's body triggers sound and/or music.[11] There are several other similar systems available today for performance artists wishing to explore interactive systems, including the BigEye software at the Studio for Electro-Instrumental Music (STEIM) in Amsterdam and EyeCon by the Palindrome Inter-media Performance Group based in Nürnberg.[12]

The classes of input devices for interactive systems can be extended beyond those that are video based to include haptic (touch), for example, pressure and flex sensors, and nonhaptic (distance), for example, ultrasound. However, a "seeing" space—video-based technology like VNS—requiring only a camera and software and being relatively easy to set up, is an attractive option for choreographers and dancers who wish to experiment with an interactive system in performance.[13] In all these systems, performer movement or action triggers some sort of event (sonic, visual, robotic, and the like) in the space around or in some proximity to the performer. The connection between the "input," the performer action that oscillates the data stream, and the output event is determined by "mapping" the input to the output in the computer software.[14]

The concept of mapping is a topic of creative interest and a focus of artistic practice in the field of electronic music in particular. In a paper entitled "Towards a Model for Interactive Mapping in Expert Musical Interaction," Marcelo Wanderley and Ross Kirk review the ways "performer instrumental action can be linked to sound synthesis parameters."[15] They describe two main "mapping" directions: (a) the use of generative mechanisms (e.g., neural networks) to perform mapping, and (b) the use of explicit mapping strategies.

Once completed, however, the instructions that make up the mapping itself are relegated to the invisibility of computation. It is the manifestation of mapping, the performer-triggered event, which enters the field of perception of the viewer–listener, not the mapping itself. This poses a challenge to those artists integrating "interactive performance systems" on the stage (in the conventional sense of a space for performance separated from an audience). Some "hide" the interactive systems, placing the emphasis on what is visible; others prefer to expose their workings.

Many accounts of interactive systems shift the focus of discussion to those occasions in which the viewer becomes a player or participant, rather than questioning circumstances and issues surrounding the more traditional performer–audience separation. An alternative to this could be to reorient a set of questions toward the notion of the performer again, but in the condition of training, practice, or rehearsal rather than in performance.

In the interactive computer music improvisation duo *Interface*, Dan Trueman and Curtis Bahn, build their own technologically augmented stringed instruments that are "extended, surrounded, and obscured . . . with a variety of technologies."[16] These "composed instruments," combine idiosyncratic sensor designs with equally idiosyncratic speaker configurations that encourage the development of new playing techniques. Finding a way to practice these techniques outside of the live performance context has produced some innovative strategies. Trueman has developed a method for fine-tuning his playing through the recording of a reduced amount of gesture-derived data that can be played back as a trace of the live performance. He analyzes this "recorded sketch" for the types of adjustments that might be made to the interactive system. In his own words:

> So, what I do is take these "recordings" of me playing the instrument and spend hours developing and refining mappings of the sensor data to audio (and video, to a lesser extent) signal processing and synthesis algorithms. This technique could be used in exactly the same manner with dancers, and could offer a better way for dancers/choreographers and composers/electronic musicians to collaborate, compose, and choreograph "offline." It can be so tedious to "mode switch" all the time between playing a mapping and actually composing the mapping. This way, you can sit down, look at the recording of the "performing body," and develop the instrument, away from the instrument.[17]

A similar fascination with the recording or tracking of movement can be traced to the "precursor to film" technologies of the late nineteenth century, for example in the work of French physiologist and instrument inventor Étienne Jules-Marey. But the computer-based technology we commonly refer to today as motion capture has a briefer historical trajectory that is closely associated with the development of advanced computer graphics. In the early 1980s, the MIT Architecture Machine Group and the New York Institute of Technology Computer Graphics Lab experimented with an optical tracking system for human body.[18] Toward the end of that decade, motion capture had evolved into a robust means for recording human (or animal) motion in a simulation of three-dimensional space and using this motion either for analysis (sports science/ergonomics) or to animate a variety of forms (film entertainment industry). With an obvious attraction to those working with movement as their material, several dance artists became involved in using these systems during the 1990s.[19]

Motion capture is a form of sampling, but computer animation is equally concerned with the synthesis of motion using a variety of computational approaches. For example, it has become increasingly possible to instruct animations to perform tasks, to develop behaviors, and to maneuver autonomously in differentiated environments. Computer scientists have developed a variety of classification systems for these approaches. Nadia and Daniel Thalmann (directors of MIRALab in Geneva) have developed a classification in three parts: (1) *locally controlled* motions driven by data via either motion capture or key frame animation; (2) *dynamic simulation* where motions are controlled using equations relating to forces, torques, and constraints; and (3) *behavioral animation* in which motions emerge within an environment in which all objects act in relation to other objects.[20] As these tools increasingly rely on dynamic and behavioral computation to generate motion, the role of the animator shifts toward defining the conditions within which these motions or gestures take place, the environments and the tasks involved.

Susan Amkraut and Michael Girard are software developers and multimedia artists who have had a significant impact on these developments in the field of computer animation through their exploration of human and animal figure animation when they were working with Ohio State University's Computer Graphics Research Group in the 1980s. They contributed to the development of the spline and inverse kinematics approaches and provided key input toward the development of dynamic simulation and behavioral animation. They are now developing software tools for commercial animators, most recently producing a software extension called Crowd, which integrates local control (motion capture data) with dynamic simulation and behavioral animation.[21] Crowd supports the organization of the behavior of large numbers of animated figures by drawing on some of the principles of bird flocking systems. Amkraut is credited with some of the early work in the mid- to late 1980s on flocking systems. Flocking systems use three simple rules: (1) separation: steer to avoid crowding local flockmates; (2) alignment: steer toward the average heading of local flockmates; and (3) cohesion: steer to move toward the average position of local flockmates.[22]

Imagine you have motion captured someone jogging. Make twenty-five copies of this figure. Place these twenty-five figures in the space and give them the following instructions: keep running toward the center of the space and avoid collisions by turning right or left. From just these two simple rules complicated patterns of self-organizing movement emerge on the screen. With Crowd you can also define the terrain or the environment the figures are moving in. Although the rules are entirely deterministic, the emergent behavior appears to be undetermined, something one could not have predicted beforehand. In addition, this would be nearly an impossible task if we had to animate each of these figures individually.

Currently, the amount of time and effort it takes to learn to work with the software and render these complex animations effectively distances this as a creative activity from the choreographer who works with dancers in physical space. However, with eventual advances in computer hardware and software, nearly instantaneous processing should be possible as the gap between adjusting the parameters and rules and seeing the result becomes negligible. Combine this with software interfaces that are easier to use, and we may see the reconvergence of computers and dance in the practice of choreography echoing the creative possibilities A. Michael Noll and John Lansdown set out to explore in the 1960s.

Acknowledgments

Thank you to Susan Rethorst for editing assistance and to Dr. Söke Dinkla and Dr. Martina Leeker for permission to use an extract from the following essay:

S. deLahunta, "Periodic Convergences: Dance and Computers," in *Tanz und Neue Medien* (book and CD-ROM/DVD), ed. Dr. Söke Dinkla and Dr. Martina Leeker, pp. 66–84. Berlin: Alexander Verlag, 2002.

Notes

1. A. Michael Noll, "Human and Machine—A Subjective Comparison of Piet Mondrian's *Composition with Lines* 1917, and a Computer-generated Image," *Psychological Record* 16 (January 1966), 1–10.

2. This quote of A. Michael Noll's is available under the "computer art" heading at http://www.citi.columbia.edu/amnoll/ (accessed April 5, 2002).

3. A. Michael Noll and Ann Hutchinson, "Choreography and Computers" and "A Reply," *Dance Magazine* (January 1967), 43–46, 81–82.

4. John Lansdown, "Artificial Creativity." A version of this paper was given at the Digital Creativity Conference, Brighton, April 1995: http://www.cea.mdx.ac.uk/CEA/External/Staff96/John/artCreat.html (accessed April 5, 2002).

5. Trisha Brown's dance making instructions can be found published in different locations, e.g.: *Locus* in *Contemporary Dance: An Anthology*, ed. Anne Livet (Abbeville Press, 1978), 54–55, and *Accumulation* in *The Drama Review: The Postmodern Dance Issue* 19, no. 1, (March 1975), 29.

6. E.g., in the interview of William Forsythe by Paul Kaiser in "Dance Geometry: William Forsythe in Dialogue with Paul Kaiser," *On Line: Performance Research* 4, no. 2 (summer 1999), 66–69, and in the interview of Dana Caspersen by Senta Driver in "It Starts from Any Point: Bill and the Frankfurt Ballet," *Choreography and Dance* 5, part 3 (2000), 24–39.

7. John Lansdown, "Computer-Generated Choreography Revisited," in *Proceedings of 4D Dynamics Conference*, ed. A. Robertson (De Montfort University, Leicester, 1995), 89–99. See http://www.dmu.ac.uk/ln/4dd/guest-jl.html (accessed April 5, 2002).

8. Noll, "Choreography and Computers," 44.

9. This issue was recently addressed at the *Software for Dancers* London-based research project taking place from September 24 to October 6, 2001, aimed to develop concepts for software rehearsal tools for dance makers. See http://huizen.dds.nl/~sdela/sfd (accessed April 5, 2002).

10. For an online historical account of the development of the "interactive performance system" see Söke Dinkla's seminal paper, "The History of the Interface in Interactive Art," presented at ISEA 1994, available at http://www.isea.qc.ca/symposium/archives/isea94/pr208.html (accessed April 5, 2002).

11. Interview of David Rokeby by Douglas Cooper, "Very Nervous System," *Wired* 3.03 (March 1995): http://www.wired.com/wired/archive/3.03/rokeby.html (accessed April 5, 2002).

12. More information is available from the following sites: Palindrome Intermedia performance group, http://www.palindrome.de/; STEIM: http://www.steim.nl/ (accessed April 5, 2002).

13. Two examples of dance and electronic composers/digital artists developing wearable sensor systems for use in interactive spaces are Troika Ranch based in New York City and DIEM based in Aarhus, Denmark: http://www.troikaranch.org and http://www.daimi.aau.dk/~diem/digitaldance.html (accessed April 5, 2002).

14. There are several options for software for mapping input to output, but Mark Coniglio's Isadora offers one of the best for the nonprogrammer to experiment with: http://www.troikaranch.org/troikatronix/isadora.html (accessed April 5, 2002).

15. The Wanderley/Ross PDF is available at http://www.ircam.fr/equipes/analyse-synthese/wanderle/Gestes/Externe/Hunt_Towards.pdf (accessed April 5, 2002).

16. More information about Interface can be found at http://www.arts.rpi.edu/crb/interface/interface.htm (accessed April 5, 2002).

17. From email correspondence between the author and Dan Trueman on March 10, 2002.

18. David J. Sturman, "A Brief History of Motion Capture for Computer Character Animation," SIGGRAPH 94: http://www.css.tayloru.edu/instrmat/graphics/hypgraph/animation/motion_capture/history1.htm (accessed April 5, 2002).

19. For a description of motion capture technologies and some work involving dance artists, see Scott deLahunta, "Coreografie in bit e byte: motion capture, animazione e software per la danza," in *La Scena Digitale: Nuovi media per la danza*, ed. Armando Menicacci and Emanuele Quinz (Venice: Marsilio Editori s.p.a., 2001), 83–100. An English version is available at http://www.daimi.au.dk/~sdela/bolzano/ (accessed April 5, 2002).

20. N. Magnenat Thalmann and D. Thalmann, "Computer Animation," in *Handbook of Computer Science* (CRC Press, 1996), 1300–1318.

21. The *Crowd* extension comes with Character Studio, one of the 3D Studio Max animation software applications that is specifically designed to work with motion capture data.

22. Simple rules leading to complex results—emergent and unpredictable—is a conception underlying the computer-aided study of chaos and other complex systems. "Artificial Life" refers to the field of modeling and study of such systems using the computer.