Proceedings of the Third AAAI Symposium on Educational Advances in Artificial Intelligence

Pedagogical Explorations in Computational Perception for Performance

Keith J. O'Hara
Computer Science Program
Bard College
Annandale-on-Hudson, NY 12504
kohara@bard.edu

Abstract

Experience using computational perception within the context of art and performance is reported. Four different types of pedagogical projects are presented: a new non-majors introductory computing course, an upper-level course covering computer vision and graphics in an integrated manner, an interactive dance piece, and a peer-led tele-workshop outreach series.

Introduction

Thinking computationally isn't restricted to scientific or engineering disciplines. Recent projects such as Performatics (Heines et al. 2011) and Artbotics (Kim et al. 2007) have shown art to be an effective context for introducing a wide variety of students to computing. Researchers, technologists and artists have applied computational perception, robotics and AI to live performance (Pinhanez and Bobick 1998; Dils 2002; Breazeal et al. 2003; Martin and Egerstedt 2009; Murphy et al. 2011).

Within this tradition of interdisciplinary collaboration between technologists and artists, we undertook four pedagogical projects under the umbrella of *Computational Perception for Performance*. Each project explored the application of computer vision and graphics in an interdisciplinary manner. The first was a non-majors introductory class on interactive systems. The second took the form of a 300-level class on the computational image that was synchronized with a 300-level studio arts class. The third exploration was a semester long research project on computational perception that culminated in a series of live dance performances. Finally, the fourth involves a peer-led tele-workshop series. Although all four ventures involved similar topics, and often shared students, each had a very different purpose and format resulting in different advantages and disadvantages.

Beyond the Keyboard, Mouse and Monitor

Interactive computing systems give students an authentic, personally meaningful computing learning experience. The computer programs they write interact with their world in a real and useful way. After just using the keyboard and monitor to *develop* their programs, it often seems contrived to use that exact same interface to *use* their software. The program, and more importantly, the process, feels disconnected and artificial; *I'm writing programs purely for the sake of writing programs*. There are a variety of ways to make students' programs more connected and authentic (e.g. using real data sets, providing connection to the Internet), but few provide as much contrast in terms of interface as employing various sensors and actuators. The purpose of the program is quite clear, as is the distinction between developer and user. In our experience, students have found computational perception to be an engaging context for exploring computing at many levels.

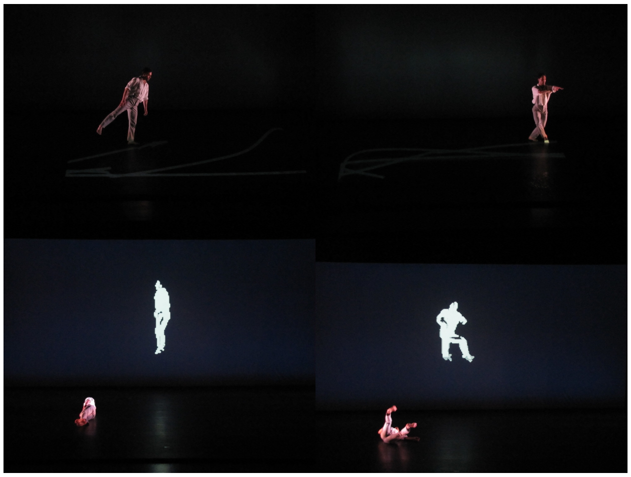

Figure 1: Photos from a performance of *Bird's Eye*

Interactive Systems

Observing the large arts and humanities student population at Bard, we decided to create a course targeted at computing for the arts. The idea being students create small computer programs that interact with the real world; computing with images, sound, and text. The class uses the Processing (Reas, Fry, and Maeda 2007) programming environment. The environment was developed to enable artists to explore computation on very pure terms — they write real programs. Class projects include an interactive self-portrait, a clock, and a digital photobooth.

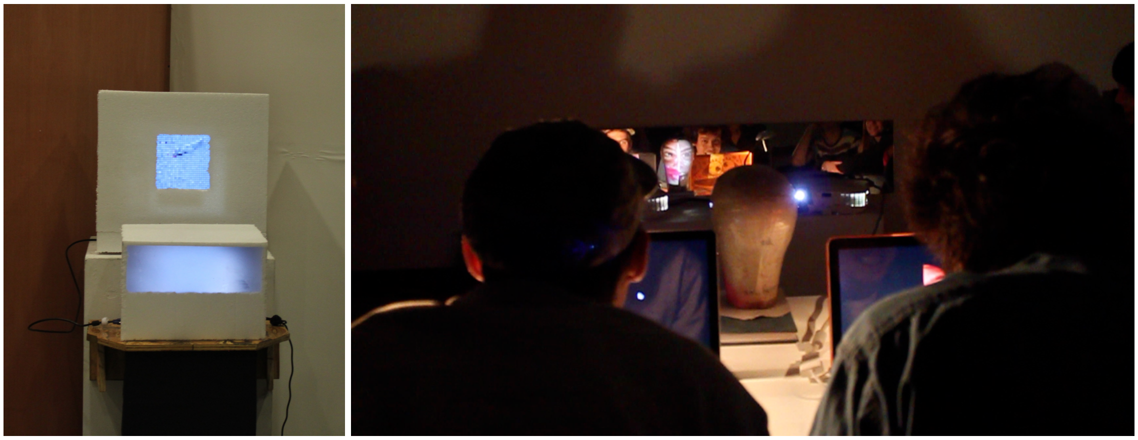

Figure 2: Two joint studio-art/computer-science projects from the *The Computational Image* and *Video Installation*.

The Computational Image

The Computational Image was a new 300-level course that introduced the fields of computer graphics and computer vision, the synthesis and analysis of digital images, in an integrated fashion. The course was synchronized with a course in Video Installation in the Studio Arts program, meaning they were offered at the same time and met several times throughout the semester for group workshops, but were two distinct classes. Although some aspects of the class would be improved by a team-taught course, synchronizing two distinct classes overcame problems with the number of prerequisites at the 300-level. Pairs of computer science and studio arts students worked together on a piece resulting in some interesting projects (see Fig. 2).

Bird's Eye: An Interactive Dance Piece

Birds Eye (see Fig. 1) was a collaboration between Computer Science and Dance. The interactive dance piece explored perspective, sensing, time and memory. The dance piece used a projector-camera system to track the dancer and provide live visualizations of the past, present and future states of the dance. The system was developed by undergraduates during a semester-long research project and was performed four times during the faculty dance concert.

A Creative Computation Tele-Workshop

We created an outreach mentoring program at Bard High School Early College centered around creative computation. The key aspects of the program are: art as a motivating context for computing, peer mentoring, and hybrid tele-workshops.

Learning about STEM via art (STEAM) is effective, but learning about STEAM from someone with whom you identify is even better. Our program is largely student run with assistance from faculty as needed, providing opportunities for peer mentoring and role modeling with a diverse set of undergraduate mentors.

The students primarily use teleconferencing software (e.g. Google Chat and Skype), file sharing (e.g. Dropbox) and cooperative coding software (e.g. sketchpad.cc) to facilitate the workshops. These tele-workshops will be punctuated by in-person workshops that ground the virtual relationships with physical interaction. The in-person workshops also allow for explorations of physical computing (e.g. Arduino, mobile robots), subjects difficult to explore remotely.

Acknowledgments

Bird's Eye was a collaboration with Peggy Florin and the Video Installation class was taught by Kristin Lucas.

References

Breazeal, C.; Brooks, A.; Gray, J.; Hancher, M.; McBean, J.; Stiehl, D.; and Strickon, J. 2003. Interactive Robot Theatre. *Commun. ACM* 46:76–85.

Dils, A. 2002. The Ghost in the Machine: Merce Cunningham and Bill T. Jones. *PAJ: A Journal of Performance and Art* 24(1):94–104.

Heines, J. M.; Greher, G. R.; Ruthmann, S. A.; and Reilly, B. L. 2011. Two Approaches to Interdisciplinary Computing+Music Courses. *Computer* 44:25–32.

Kim, H. J.; Coluntino, D.; Martin, F. G.; Silka, L.; and Yanco, H. A. 2007. Artbotics: Community-based Collaborative Art and Technology Education. In *ACM SIGGRAPH 2007 educators program*, SIGGRAPH '07.

Martin, P., and Egerstedt, M. 2009. Optimization of multi-agent motion programs with applications to robotic marionettes. In *12th International Conference on Hybrid Systems: Computation and Control*, HSCC '09, 262–275.

Murphy, R.; Shell, D.; Guerin, A.; Duncan, B.; Fine, B.; Pratt, K.; and Zourntos, T. 2011. A midsummer night's dream (with flying robots). *Auton. Robots* 30:143–156.

Pinhanez, C. S., and Bobick, A. F. 1998. "It/I": A Theater Play Featuring an Autonomous Computer Graphics Character. In *Sixth ACM international conference on Multimedia: Technologies for interactive movies*, 22–29.

Reas, C.; Fry, B.; and Maeda, J. 2007. *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press.