Teaching Accomplishments and Objectives

Robert E. Bodenheimer, Jr.

Introduction

As an educator, I have high expectations for the quality of my students' work ("Expect more and you will get more" [Chickering and Gamson, 1987]). I believe in trying to motivate and channel students' enthusiasm for a subject through projects, both student-devised and assigned, technological scaffolding, and research opportunities ("Time plus energy equals learning" [Chickering and Gamson, 1987]). I have found that students are quite energetic and I try to harness that energy in the intellectual discipline of a classroom. I strongly believe that the best teaching occurs when learners understand underlying principles and can apply them to problems in new contexts [Bransford et al., 1999].

At Vanderbilt, I have taught courses in the areas of Computer Graphics, Computer Animation, Image Processing, and Human-Computer Interaction. The creation and display of synthetic images is one of the most striking things about the modern computer. Although my courses are electives in our Computer Science curriculum, in addition to providing breadth, they provide important fundamentals for many other areas of Computer Science that increasingly use visualization. All the courses I taught I developed myself, from the ground up.

Because of its visual nature, Computer Graphics can serve as an attractive gateway to the broader field of Computer Science and, more generally, to those theories of computation that are increasingly becoming common in other disciplines. To foster this gateway, I have been a lecturer in the interdisciplinary course *Introduction to the Visual System* (EECE 225/Psy 236) since 2001, and in the Computer Science module of *Introduction to Engineering* (ES 140).

A summary of my teaching activities at Vanderbilt is:

Curriculum Development Since there was no computer graphics or animation program at Vanderbilt prior to my arrival, I have developed and taught five new courses, four of which are now part of the regular CS curriculum (the fifth has only been taught once as part of the new graduate curriculum in Computer Science). Two are undergraduate (*Introduction to Graphics*, CS 258; *Introduction to Animation*, CS 259), and three are graduate (*Advanced Animation*, CS 351; *Human-Computer Interaction*, CS 352; *Advanced Topics in Computer Graphics and Image Processing*, CS 396). I have also developed lecture material for my parts of the Visual System course and the Introduction to Engineering course.

Evaluations My evaluations in these courses have been above the School of Engineering average in most cases (12/15), and some are significantly above the average. I expect a lot from students in these courses and evaluations of requirements in my courses have also been above the School of Engineering average in most cases (10/15), again sometimes significantly. My most recent evaluations for the undergraduate courses *Introduction to Graphics* (CS 258) and *Introduction to Animation* (CS 259) were above the school average (4.33 for CS 258, school average 3.83; 4.0 for CS 259, school average 3.79).

Advising I was the undergraduate advisor for the Computer Science Class of 2005 (22 students). I have supervised three undergraduates as part of part of Vanderbilt University's Summer Research Program (all of whom have published refereed conference papers or are co-authors on journal articles), and six undergraduates in independent studies. I have supervised four graduate students in theses programs (Jing Wang and Christina de Juan have completed Ph.D.'s, Betsy Williams is on track to complete a Ph.D. in Spring 2007, and Elizabeth Seward will complete her M.S. degree in 2006). I have supervised four other graduate students in research projects. These students graduated with M.S. degrees in Computer Science.

Undergraduate Education

I believe that advanced classes in Computer Science ought to focus on projects. Few intellectual fields outside of Computer Science enable juniors and seniors to build complex systems of significant capability in a single semester. *Introduction to Graphics* (CS 258) is a project-based, rigorous introduction to the fundamentals of graphics. It covers image processing operations as linear systems, and includes the 2D Fourier Transform, anti-aliasing, 2D and 3D transformations, polygon scan conversion, clipping, ray-tracing, and morphing. Projects are completed in C or C++ using a simplified OpenGL library. These projects require the students to translate abstract and mathematical concepts into working code with the tools that they will employ over the course of their professional careers. Typically, the projects generate a TIFF image reader/writer, do anti-aliased minification and magnification, implement 3D transformations with perspective, and implement a ray-tracer (examples are shown in Figure 1). These projects teach the fundamentals of computer graphics, but they also do more. In coding the TIFF image reader/writer, the students must read and implement a complex software specification approximately 60 pages long (a strict subset of the complete specification [Adobe Developers Association, 1992]). Many of them have never had to read a software specification before, and I introduce them to this important component of their professional development.

CS 258 is an elective course in the undergraduate curriculum. Throughout the time I have taught the course, I have modified the subject material to reflect interesting developments in the field of computer graphics, although its core is based on fundamentals. In the rapidly evolving fields of computer graphics and animation, fundamentals imply that **students should be able to translate the underlying equations of image creation and manipulation into efficient algorithms.**

The course has been received well by students, although it is known to be one of the more difficult courses in the Computer Science curriculum (the requirements rating has always been more than one standard deviation above the school average). The overall course ratings have been above the school average every year except the first (I have taught the course four times). In the first year, I did not properly anticipate the level of programming skill that the undergraduates had, and it was necessary to adjust my lectures to address this point. Individual instructor ratings have increased every year that I have taught the course and they were above the school average the last time I taught it. Comments on CS 258 include *“Teacher is always very helpful outside of class,” “Projects very interesting, professor always relates lectures to actual graphics work,” “lots of work,”* and *“teacher is patient and nice.”*

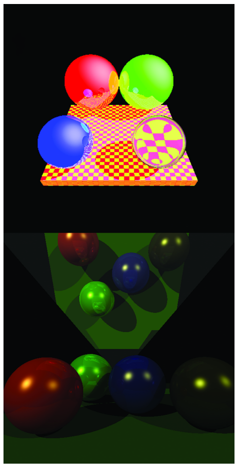

Figure 1: Example images generated by students in *Introduction to Graphics* (CS 258).

I believe that allowing students to develop a sense of what they have learned by allowing them to tackle problems of their own conception provides an important self-motivating framework for them to learn successfully. This principle guided my development of *Introduction to Animation* (CS 259), a popular elective course in the CS curriculum. This course is also project-based. It was designed so that the students were divided into groups, and each group was asked to produce a two-to-three minute computer animated film over the course of the semester. This task was divided into components that included implementing the models, animating the models, rendering the models, editing, and adding sound effects. The feature of the course that students find most exciting is that they get to invent the animation project that they will construct in consultation with me (examples are shown in Figure 2). I believe this feature contributes to the success and interest the students express in the course. The lectures introduce the students to state-of-the-art computer animation techniques that they explore as part of their animation projects. Most recently, I incorporated material on the automatic generation of motion transitions, research by my Ph.D. student Jing Wang, into the lectures on motion capture and human figure animation. The student projects have significantly exceeded my expectations with the quality of the animations produced.

The response of the students to this course has been very enthusiastic, although we have sometimes had infrastructure problems regarding the rendering speed the school computers can support. The overall course ratings have been above the school average every time the course has been taught. Project screenings at the end of the semester are always well attended, and usually include a number of faculty and students not enrolled in the class. Student projects have also been shown in animation festivals, which implies that the projects are of high quality. There has been interest in broadcasting them on VTV, the student television station.

Student comments on CS 259 include *“good teacher, good communication with class,” “the instructor stayed very current, always kept us up with the material, and seemed genuinely excited about the material,”* and *“great lecture*

Figure 2: Example images from *Introduction to Animation* (CS 259) class projects.

style and course setup."

CS 258 and CS 259 give students a view of modern engineering and software development. In modern engineering practice, the scope and magnitude of large projects require that they be carried out by teams of engineers and computer scientists (the days of building a best-selling video game in someone's garage are over). I believe that we should therefore expose students to group projects, as CS 259 does, but also instill in students the confidence and the ability to get a complicated system working on their own, as CS 258 does. These courses teach students to be designers of complex software systems, and to develop algorithms and visual media in an increasingly animated and digital world.

Additionally, both of these courses provide students with examples of the use of modern graphics and animation. I was privileged to attend two seminars sponsored by the Center for Teaching given by Prof. Leonard Folgarait, who demonstrated quite compellingly the interest and motivation that could be generated by a proper use of examples in a lecture. I have tried to incorporate a similar technique into my own lectures. Students have recognized this effort in their written comments, saying, for instance, *"Examples at the end of class help show what computer graphics can do,"* and *"Excellent demos of lecture topics."*

The students' evaluations I have received, both from undergraduate and graduate students, attest to my sincere efforts to help students learn and master a subject. These evaluations represent only part of the story, however. I participate in *Introduction to the Visual System* (Psy236/EECE225), a multi-disciplinary course covering the the human visual system and how it allows us to interface with the world. I participated in a committee to evaluate and re-design the freshman course *Introduction to Engineering* (ES 140) into its current modular format. I have been a lecturer in the Computer Science module of ES 140. I also was the advisor for the twenty-two students in the Computer Science Class of 2005, and used that opportunity to help those students excel in their time at Vanderbilt.

Finally, critical to the health of Computer Science as an intellectual endeavor is encouraging undergraduates to participate in research. **I believe research plays an important role in undergraduate education, and allowing students to see the entire process, from problem conception to paper writing, plays a critical role in enticing students into research careers.** I have supervised six undergraduate students in independent studies, and three in a Vanderbilt Summer Undergraduate Research Program (VUSRP). An example of such a student project involving the automatic generation of multiple perspectives in a single image is shown in Figure 3. In all these studies I have mentored students on research problems and all projects have delivered items of research interest. The VUSRP fellows were all able to develop their projects into research publications. All three students have had refereed conference papers or are co-authors on journal articles. Three of these nine students have gone on to graduate studies in Computer Science, and three have joined startup companies.

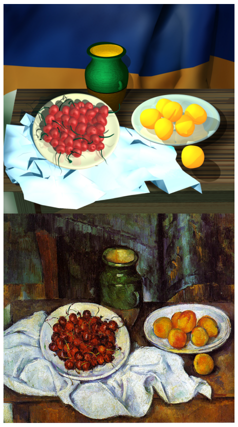

Figure 3: Example independent student project: automatic multi-perspective image synthesis (top) after Cezanne (bottom).

Graduate Education

I believe that graduate instruction in classroom settings should foster the transition of a promising undergraduate student into a researcher, and my graduate courses are built around this idea. In addition to advanced concepts, I believe that graduate instruction must teach students to find, understand, evaluate, and present current research. This section covers my graduate coursework at Vanderbilt; the success of my graduate advising is covered in my Research Statement.

My first course at Vanderbilt was *Special Topics in Computer Graphics*, CS 395, a graduate version of what has become *Introduction to Graphics* (CS 258). The course was an advanced version of CS 258 and added one project, to implement the Beier-Neely morphing algorithm [Beier and Neely, 1992]. This project required the students to read and implement a published research paper. The paper is carefully chosen to be implementable in a reasonable time frame, but the experience of actually implementing a research result, and struggling with translating the published results into reality, is a critical step in teaching graduate students how to conduct research. This course was received well by the students. The overall course rating was 3.88 (school average 3.64). Students wrote comments such as *"the most enjoyable class I have taken at Vandy,"* and *"very good projects — I learned a lot."*

In Spring 2002, several graduate students approached me to ask if I could teach a course in human-computer interaction (HCI). This area of Computer Science is related to but not my main area of expertise. Consequently, I developed and taught a graduate level course, *Human-Computer Interaction* (CS 352). It serves as an an introduction to this important research area in Computer Science. Taught as a seminar, we covered the background material sufficient for an understanding of approximately twenty-five recent HCI papers. As part of the coursework, students divided into groups and proposed projects involving a novel application of HCI. They presented their work in a poster session and wrote a final report in a format suitable for submission to a conference. Several of these projects have been accepted at peer-reviewed conferences. The course was well-attended, popular, and now has become a regularly taught course. Recently, this course has been taught by Prof. Julie Adams.

The overall rating for this course was at or slightly below the school average the two times I taught it. Appraising these evaluations, my conclusions were that the seminar format is not the optimal format for beginning graduate students to learn this material, particularly in an area where I may lack sufficient experience to give broad context and motivation to the material. Subsequently, Prof. Adams has changed the format of the course significantly. However, the written comments I received for this course were generally favorable, and include *"Instructor was very knowledgeable about subject," "The papers gave an added dimension to the course,"* and *"I really enjoyed the format of the class."*

I have also taught an advanced seminar in computer animation for graduate students, *Advanced Animation* (CS 351). This course covers the state-of-the-art in computer animation techniques. Students individually choose and implement one of the papers that we cover, often with enhancements suggested by me. The students have found this course very exciting and it had an overall course rating of 4.38 (school average 3.78). Student comments include *"got a good feel of what the field is about,"* and *"being forced to give a presentation on a paper really pushes you to fully understand the paper, which really helped broadening my horizon in the field of animation."*

In Spring 2006 I developed a seminar course, *Advanced Topics in Computer Graphics and Image Processing* (CS 396), that fits as a much-needed depth course for the new Computer Science graduate curriculum (I was a member of the committee that designed the new curriculum). This course focused on modern tools in graphics and image processing, and particularly focused on teaching students to understand, evaluate, and present current research. An example project, to develop level set segmentation techniques applicable to vascular data, is shown in Figure 4. In this course, I was able to teach some of the core principles that motivate my research, such as the need for perceptual studies to inform and validate the design of animation algorithms, and show the students some of our recent work in virtual environments. Student response to this course was very favorable, with an overall course rating of 4.06 (school average 3.79). Student comments include *"course was great and so was the instructor," "one of the most practically useful courses I have taken at Vanderbilt," "made the class interactive and interesting,"* and *"the things I like about [Dr. Bodenheimer] are that he is . . . excited about sharing tips and is very approachable."*

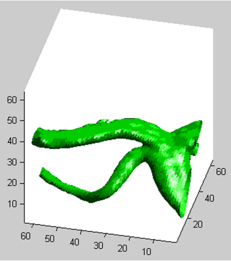

Figure 4: Example graduate student project.

Future Plans

My future plans for teaching are guided by the following observations. Computer Science is more than data structures and programming. The field changes rapidly in response to faster and more sophisticated computational resources. Consequently, Computer Science students must master fundamental principles of learning that will enable them to pursue their careers effectively. In Computer Graphics, the theories of computation that enable the computer to perform difficult calculations quickly and efficiently are structured to the specific result of creating visual media. Providing examples of new and improved visual media, and the methods of creating them, is a major attraction of the field for me, and I strive to pass it on to my students.

As mentioned previously, Computer Graphics can serve as a portal to the theory of computation and the broader discipline of Computer Science. I intend to continue to promote the field in this way. My lectures in the Visual System class are one example in this direction, and animation can serve as another direction. Animation is a fascinating art form, and historically has tied its presentation very closely to a soundtrack, more closely than film, for example. I have begun conversations with Prof. Daniel Landes in the Blair School of Music, who teaches course in Computer Music, about a direct, interdisciplinary exploration of the two through our respective classes.

References

[Adobe Developers Association, 1992] Adobe Developers Association (1992). *Tiff Revision 6.0*. Adobe Systems, Inc., Mountain View, CA. http://partners.adobe.com/asn/developer/PDFS/TN/TIFF6.pdf.

[Beier and Neely, 1992] Beier, T. and Neely, S. (1992). Feature-based image metamorphosis. In *Computer Graphics (Proceedings of SIGGRAPH 92)*, volume 26, pages 35–42, Chicago, Illinois. ISBN 0-201-51585-7.

[Bransford et al., 1999] Bransford, J. D., Brown, A. L., and Cocking, R. R., editors (1999). *How People Learn: Brain, Mind, Experience, and School*. National Academy Press, Washington, D.C.

[Chickering and Gamson, 1987] Chickering, A. W. and Gamson, Z. F. (1987). Seven principles for good practice in undergraduate education. *American Association of Higher Education Bulletin*, 39(7):3–7.