Teaching portfolio

Fuhua (Frank) Cheng (Fall, 2006)

A. Teaching Evaluation

1. Reflective Statement

My teaching has three goals: (1) to make sure that students understand the course materials well, (2) to make sure that students know how to use/apply the materials they learn in class, and (3) to make sure that the students are evaluated fairly.

To achieve the first goal,

- I use a motivation-driven approach in my lecture, i.e., I give the background and applications of the result first, and then explain the theory that leads to the result.
- I give many examples in my notes (see, e.g., my CS535 and CS633 notes).
- I encourage the students to be involved and active during lectures. (However, for those who find it difficult to do so, I welcome them to ask questions or make comments after class.)
- I make all my class notes available on line so that, instead of copying my notes in class, they can closely follow my lectures on course materials. (My class notes such as CS633, CS631, CS535, and CS321 have been used by some of my students and colleagues in their own classes.)

To achieve the second goal,

- I give applications for each covered result.
- I provide students with sample programs to help them initiate their work (see my web pages).
- I encourage students to share their ideas.
- I award students with extra credit if they have new ideas on assignments.

To achieve the third goal,

- I always let the students know at the outset of the course exactly what is expected. I clearly specify the requirements of the course such as materials to be covered, grading policy, program requirements (see, e.g., my CS535 and CS633 Programming Requirements), late penalty, and numerical scale to be used in the evaluation, on the first day of class.
- I provide students with solution sets for all homework assignments and exams (see my webpages) so they would not only know the solutions to the questions, but also know if their works are graded fairly.

I have different expectations for graduate and undergraduate students though. For a undergraduate or programming-extensive course, the students are evaluated based on two subjects: programming assignments and tests. I usually put equal weight on both sides so the effort of the students can be evaluated fairly. However, I encourage students to do critical thinking and they get extra credit if they do so such as providing comments or improvement on existing techniques. For a seminar course or advanced topics, I evaluate the students mainly based on the quality of the work, i.e., I will follow the numerical scale, but a student with good ideas will get more extra credit than the ones who don't.

2. **List of Courses taught**

Term	Course	Title	Enrollment
Spring 05	CS275	Discret Mathematics	34
Fall 05	CS535	Intermediate Computer Graphics	24
Spring 06	CS633	Computer Animation	9
Fall 06	CS535	Intermediate Computer Graphics	8
	CS335	Graphics and Multimedia	13

Discrete Mathematics (CS275): This course covers topics in discrete math aimed at applications in Computer Science. For fundamental principles, we cover: set theory, induction, relations, functions and Boolean algebra. For techniques of counting, we cover: permutations, combinations, recurrences, and algorithms to generate them. This course will also do an introduction to graphs and trees.

Students will develop a knowledge of a variety of mathematical tools applicable in computer science. Specifically, students will be able to (1) construct inductive proofs, (2) apply set algebra, (3) apply elementary logic, (4) enumerate combinatorial objects, and (5) solve recurrence relations.

Intermediate Computer Graphics (CS535): This course covers 2D graphics such as rasterization of lines/polygons/curves, clipping, anti-aliasing and 3D graphics such as modeling, viewing, lighting, shading, hidden line/surface removal. More advanced topics such as solid modeling, curves and surfaces, advanced raster graphics architecture and algorithms, advanced modeling techniques, and animation will also be covered. The supporting graphics system used in this course is OpenGL.

Computer Animation (CS633):

This course presents algorithms and programming techniques for specifying and generating motion for graphical objects. It addresses practical issues and provides accessible techniques and straightforward implementations. It is not intended for animators using off-the-shelf animation software, nor does it address the issue of computer-assisted animation, i.e., the computerization of conventional hand-drawn techniques. This course is primarily concerned with 3D computer animation.

Motion specification techniques in two categories: interpolation and basic technique and advanced algorithm, are studied and discussed. The interpolation and basic techniques category consists of ways in which the computer is used to fill in the details of the motion once the animator specifies the required information, such as key framing and path following. Advanced algorithms generate motion using a set of rules or constraints that specify what is to be done instead of how is to be done. Model-specific applications are also surveyed. These are grouped into two general areas: natural phenomena and figure modeling. The graphics library used in this course is OpenGL.

Graphics and Multimedia(CS335): This course focuses on the graphical human-machine interface, covering the principles of windowing systems, graphical interface design and implementation, and processing graphical data. There is an emphasis on medium-scale programming projects with graphical user interfaces using an object-oriented programming language such as Java.

3. **Course Syllabi**

(see attachment)

4. **Student Evaluation**

		2005		2006
		Spring	Fall	Spring
		275	535	633
	Enrollment	34	24	9
	Number of answers	23	17	5
1	Material/grading outlined	3.3	3.6	3.8
2	Textbook	2.9	3.4	3.6
3	Supplemental reading	3.3	3.6	4.0
4	Exams reflection	3.6	3.7	4.0
5	Grading fair	3.6	3.6	4.0
6	Distributing assignments evenly	3.5	3.8	4.0
7	Assignments graded promptly	3.7	3.6	4.0
8	Grading including comments	2.9	3.5	4.0
9	presentation	3.3	3.5	4.0
10	Knowledge of subject	3.6	3.8	4.0
11	Availability	3.5	3.6	3.8
12	Answer questions	3.4	3.5	4.0
13	Stimulate interest	2.8	3.5	4.0
14	Encourage participation	3.4	3.5	3.8
15	Respect viewpoints	3.2	3.5	3.8
16	Ability to analyze	3.3	3.5	4.0
17	Solve problems	3.6	3.2	4.0
18	Understand concepts	3.2	3.8	4.0
19	Read further	2.6	3.6	4.0
20	Value of course	3.2	3.6	4.0
21	Quality of teaching	3.5	3.8	4.0

5. **Class Notes**

(CS535, CS633, CS335 and CS275 can all be accessed from my website)

B. Advising Evaluation

1. Reflective Statement

My goal in advising a project or a thesis is to ensure that the student knows how to set up a target and how to develop a strategy to reach that target. The target must be very specific and the strategy must be practical. The idea is to let the student know how to play a game by him/her-self and to what extent that he/she should keep trying before giving up. I help the student with the technical part initially after he/she has successfully performed background study, target selecting, and strategy design.

My advising in pre-registration meetings with the students will ensure that (1) students understand the requirement of a computer science major in addition to the college and university requirements, and (2) each student develops an appropriate course plan for each semester. This will be achieved by going through a checklist with the student and showing him/her the best combination for the semester.

2. Numbers of Students Advised

	2005		2006	
	Spring	Fall	Spring	Fall
Undergraduate Program Advisees	32	29	21	27
Graduate Program Advisees	5	5	2	2
MS Committees*	0	0	0	0
PhD Committees**	3	3	3	4

** Yun Lin, George Landon, Wei Su(ECE), Ning Cao

3. PhD Students

- **Shuhua Lai**

  Area of Research: *Subdivision surface based one-piece representation*

  Starting Date: January 2003

  Supported Period: January 2003 - August 2006 (supported by NSF grants DMS-0310645 and DMI-0422126).

  Publication: five journal papers, three conference papers, one submitted

  Graduation Date : August 2006.

  Current Status: *Assistant Professor at Virginia State University*

- **Gang Chen**

  Area of Research: *Feature Generation for Subdivision Surfaces*

  Starting Date: January 2005

  Current Status: *passed no breadth and depth exams yet*

  Publication: *two conference papers*.

  Supported period: January 2005 - present (supported by NSF grant DMI-0422126)

  Anticipated Graduation Date : December 2008.

- **Qi Chen**

  Area of Research: *Shadow Generation based on Angular Representation and One-Pass Z-Buffering*

  Starting Date: August 2004

  Supported period: August 2004 - August 2005 (supported by NSF grant DMI-0422126)

  Publication: *none*.

  Current Status: *quit PhD program (switched to MS program) in August 2005*

- **Fengtao Fan**

  Area of Research: *Shape Reconstruction using Subdivision Surfaces*

  Starting Date: August 2006

  Supported Period: August 2006 - present (CS Department TAship).

  Current Status: *passed no breadth and depth exams yet.*

  Publication: None.

  Anticipated Graduation Date : May 2010.

4. **MS Students**

- **zhaohui Ren**

  Masters Thesis: *Distance Evaluation for Catmull-Clark Subdivision Surfaces*

  Supervising Period: September 2004 - August 2005

  Current Status: *left in August 2005 for a job in New York*

- **Ping Du**

  Masters Project: *Catmull-Clark Subdivision Surface Interpolation*

  Supervising Period: September 2004 - August 2005

  Current Status: *left in August 2005 for a job in Lexington*

5. **Student Activities Summary**

- **Summer project**

  Student advised: Samir Malla

  Advising period: summer, 2006

Attachments

1. **Syllabi**

   CS535 - Intermediate Computer Graphics
   CS633 - Computer Animation
   CS275 - Discrete Mathematics
   CS335 - Graphics and Multimedia

2. **Materials prepared for teaching activities** (all can be accessed from my website)

   CS535 - class notes, programming assignments (with sample programs), homework assignments (with solution sets)
   CS633 - class notes, programming assignments (with sample programs), homework assignments (with solution sets)
   CS335 - class notes, programming assignments (with sample programs), homework assignments (with solution sets)
   CS275 - class notes, programming assignments (with sample programs), homework assignments (with solution sets)