Robert Pless; Teaching Philosophy, 2005

I focus on teaching classical tools with applications to current problem domains. I have high expectations of my students in terms of the breadth, depth, and mathematical sophistication of their work. In return, I constantly seek new ways to teach challenging material, and work to define assignments and projects that are timely, relevant, and interesting to students. Over the last five years I have taught a rotation of three courses: graduate courses in Computational Geometry and Computer Vision, and the required, sophomore level course in Logic and Discrete Mathematics. Teaching each these classes presents different challenges but student response to each of them has been overwhelmingly positive. I believe this is due to the teaching philosophy that I sketch below.

"Time plus energy equals learning", wrote Chickering and Gamson in their *Seven principles for good practice in undergraduate education*. Students may be coerced into spending time in a class, but such time is rarely energetic. Instead of coercion, I try to guide the innate enthusiasm for learning through a combination of technology tools, assigned and student driven projects, and opportunities for extended topic exploration and research.

I believe that technology has an important role to play in education. Putting effective course materials online to be accessed "anytime, anywhere" is increasingly important as student time is less structured than in the past. But a more important role is in the ability of the web to support interactive visualizations within online course materials. This interaction, often in the form of Java applets, demands more attention of a students, and offers the chance to experiment and explore. Especially in computational geometry and computer vision classes, a few carefully designed applets can provide students with intuition about perspective projection or Delaunay triangulations that would be difficult to attain without active exploration.

One more extended example of this, applied to undergraduate teaching, is WUGLE (WUGLE User Guided Logical Exercises). Motivated by the reluctance of students to perform long, formal logical proofs, an undergraduate student (Jacob Perkins) and I worked to develop a user interface for a student to interact with a propositional expression. When the user highlights part of the expression, the allowable transformation rules, such as the distributive law, are presented as options to the user. A proof can be constructed by selecting the correct transformation rules (WUGLE then automatically applies the transformation), eliminating the potential for frustrating "copying" errors. The tool also generates random problems of the form "Prove the following is a tautology". This tool is open source and available on wugle.sourceforge.net, and led to larger work that comprised Jacob's Master's Thesis.

I believe that advanced classes ought to focus on projects. Computer vision — like many advanced computer science course — requires the synthesis of many intellectual components: programming and algorithm analysis, linear algebra, geometry, color, psychology, and optics. Projects are the only readily available tool to provide the motivation and intuition for integrating these far reaching subjects. My Computer Vision class pairs classical lectures with a series of projects that start simple and incrementally grow more complicated. These projects have increasing flexibility which students use to merge their interests in surprisingly disparate fields. Several of the most outstanding projects are maintained on my class project Hall of Fame (http://www.cs.wustl.edu/~pless/hallOfFame.html).

I believe that research and exploration plays a vital role in the undergraduate curriculum. Research opportunities set apart the undergraduate experience at premiere research universities and offer talented students the chance to use experimental technologies and contribute to world class research. I believe in directly mentoring these students; the substantial time investment in working with new students is offset by the infusion of novel and creative ideas that students bring to our projects. Several undergraduate students

have contributed significantly to my research program, and are co-authors on several publications [10, 11, 16, 8, 9].

Furthermore, I think that research opportunities play a role beyond including students in ongoing research projects. In my advanced classes, I offer the opportunity to work on open, unsolved questions in the field as a final project. Since it is unreasonable to expect students to solve long-standing open problems, I allow them to turn in a log-book of their efforts and meet with them several times during the project to suggest approaches that are reasonable to take. The two most successful of these class-based open research projects led to new understanding of problems in Computer Vision and Computation Geometry. A visualization of the "shortening distance regions", characterizes places where you can add a point to a point set to make the minimum weight triangulation have a *lower* total weight, and an open ended project on auto-stereograms (magic eye pictures) offered the first tool to make these images that are not exactly repeated patterns across the page (both of these projects are also linked on the Hall of Fame class projects page).

Finally, I believe that Computer Science is more than proofs, data types, and algorithms. There is no other intellectual field where so few can build so much, so quickly, for so many. And no field has had so many interesting ideas ignored by so many for so long. To have an impact, students of computer science need to learn a sense of style — to build proofs, to build interfaces, to build systems so that they can and will be used by others. The only way I know to teach a sense of style is with examples, tying small proofs in class to large ideas, relating social constructs to data structures, and with a consistent willingness to listen to students' ideas. This is what attracted me to Computer Science, and I hope to pass it on.

Summary of Teaching Evaluations

I have taught CS 201/240 (Logic and Discrete Mathematics), CS 506/546 (Computational Geometry) and CS 519/559 (Computer Vision). The following is a summary chart of the "Teacher Evaluation" scores in the student evaluation compiled by the university. The "Overall Course Evaluation" scores are substantially similar.

	Poor ..... Excellent										
Rating:	1	2	3	4	5	6	7	8	9	AVG	Compare *
CS 201, FA 2001	0	1	1	2	3	5	6	19	18	7.5	(7.0)
CS 201, FA 2002	0	0	0	0	1	1	4	9	16	8.2	
CS 240, FA 2003	0	0	0	0	0	1	6	12	14	8.2	
CS 240, FA 2004	0	0	1	0	0	1	4	10	18	8.2	
CS 506, SP 2001	0	0	0	0	1	0	4	5	20	8.4	(8.05)
CS 506, SP 2003	0	0	0	1	0	4	16	15	18	7.8	
CS 506, SP 2005	0	0	0	0	1	0	4	6	29	8.6	
CS 519, FA 2000	0	0	0	0	0	1	2	5	3	7.9	(7.94) **
CS 519, SP 2002	0	0	0	0	1	1	6	8	7	7.8	
CS 519, SP 2004	0	0	0	0	1	3	2	14	19	8.2	

* This is the average rating for all professors for this course from 1989--2003, as compiled by Professor Loui.

** I believe that I am on the only teacher of the class, this average is the weighted average of the scores that I have received teaching this class.

The following quotes are selected from student course evaluations. The first three of these are from the anonymous website "Ratemyprofessor.com", where my average rating for "overall quality" is is 4.4/5.0, the remainder of the comments are from the Washington University online course evaluation process (evals.wustl.edu).

Logic and Discrete Math

"Funny guy and good teacher. Homeworks can be difficult but Pless is willing to answer any questions you may have. Take him for CS201 if you can."

"Awesome lecturer. Extremely clear and easy to follow explanations for everything! Plus a quirky sense of humor. 241 with Pless is INCREDIBLE!"

"Beware: "Quirky" is a good adjective. But he's got a great sense of humor, and he really, really knows his stuff- take him if possible, but only if you're going to actually participate in class."

"Very enthusiastic, good examples!"

"The course material may not have been the most exciting but the professor was enthusiastic about it and the class was always fun to attend and ended up being one of my favorite classes I have taken."

"Professor Pless is one of the few great teachers at Wash U, I hope to have more classes with him someday"

Computational Geometry

"Great course, interesting subject, entertaining professor. You must take this class before you leave Washu"

"Take this course w/ Prof. Pless. Its tough but you'll get a lot out of it."

"He is a brilliant person. He can make anything interesting."