

Teaching Statement

Crane Chen

The Johns Hopkins University

Overview

In addition to research, I've been developing my personal teaching philosophy and approaches in preparation for my future position at the lectern. I categorize teachers into three levels: The least effective teachers merely recite Wikipedia-based slides, providing no value to students. Slightly more effective teachers introduce high-level ideas but jump to derivations without context. Best teachers teach intuitions, which cannot be googled. I am dedicated to striving for excellence and emulating the teaching style of best teachers.

Teaching Experience

At the Johns Hopkins University, I completed my TA responsibilities before finalizing my thesis topic, so unfortunately missed the chance to TA my favorite course, Computer Graphics. I TA-ed 530.646 Intro to Robotics twice, where I graded assignments, answered questions on Piazza, hosted office hours, and mentored labs and projects involving the UR5 robot. I also TA-ed 601.661 Computer Vision once, designing weekly Jupyter notebook exercises, grading coding homework, and supporting students on Piazza and in office hours. For both courses, I mentored student-proposed projects focused on open-ended research ideas, with some submitted to top conferences for publication. Through the TReND in Africa charity, I lectured on Python programming for medical school students. Additionally, I delivered mini-lectures and invited talks at the Johns Hopkins University on surface reconstruction and other topics, hosted by various faculty members.

Teaching Philosophy

In terms of style, I consider myself as a fusion of old-school and new-school.

The Old-school Crane

Intuition is more important than implementation details

This is the most important teaching philosophy I got from my Ph.D. advisor prof. Misha Kazhdan. His memorable C++ intermediate programming course at Hopkins included a simple, clear explanation of segmentation faults: *"If you ran into this error, nine out of nine times it's related to memory. You are touching memories that's not yours"*. Simple and clear. I'll probably remember this lifetime. To elaborate the same concept, a bad teacher would just copy the Wikipedia definition *"In computing, a segmentation fault (often shortened to segfault) or access violation is a fault, or failure condition, raised by hardware with memory protection, notifying an operating system (OS) the software has attempted to access a restricted area of memory (a memory access violation)."*

Interacting with students effectively is nontrivial

This philosophy was again passed to me from my Ph.D. advisor Misha Kazhdan. I believe interaction with students is something sounds trivial but difficult to implement. Misha is such a good teacher, who would try to catch every single face that looks confused in the classroom and go after them. It is dangerous to say "Understood" in his classroom when he asks "Understand?". Admittedly, it was a bit intimidating at the beginning. But it has been very beneficial for students. That way, I stopped pretending a know-it-all, and learned more.

Learning happens when coding happens

I appreciate heavy coding courses as a student. Talk is cheap in this field. The plan is to do the same for my students. Even for research, strong coding skills help you implement your ideas faster, that way you would be more productive.

The New-school Crane

Less is more

This philosophy was passed to me from my father prof. Hongzhi Chen, amazing teacher in Electrical Engineering. I will keep in mind that not every single student in the classroom of Computer Graphics aims at becoming an expert in this narrower field or publishing SIGGRAPH papers. When planning my lessons, I contemplate which single concept I would like my students to retain if they could only remember one thing. For instance, in my rendition of Computer Graphics, the key takeaway from the entire lecture series would be the Laplacian. For my least motivated students, this serves as the baseline expectation. For those with a bit more drive, I strive to identify one enduring concept per lecture that they can commit to memory for a lifetime.

Using YouTube to find signature courses

As a student, I am a fast learner. The way to do it is to actively search for online courses (most often, YouTube). These are sometimes of higher quality than the same course from my own institution. Because these are what I call "signature course". Usually, if you are second best in the world in teaching this, there would be no need to upload to YouTube, because everyone would be watching the best.

Teaching Plan

Promoting diversity equality and inclusion in pedagogy

At Hopkins, I attended a four-semester seminar series on Diversity and Inclusion in Computer Science and Engineering led by my Ph.D. advisor, Misha Kazhdan. Promoting diversity, equity, and inclusion (DEI) requires more than just broadening admissions to include underrepresented backgrounds; it also involves supporting their success through tailored educational approaches. Limited past accessibility has left underrepresented students overwhelmed in heavy coding or math courses. Research shows that "CS Unplugged" [Bell and Vahrenhold(2018)] encourages their involvement. I aim to develop bridging lectures in an unplugged style, to motivate these students. Moreover, as students often spend more time with TAs than professors, I will strive to involve more women and people of color as TAs, ensuring every student has a TA they feel comfortable approaching.

Upgrading existing courses

I benefited from Discrete Differential Geometry (Keenan Crane, CMU), Shape Analysis (Justin Solomon, MIT), and CS231N Deep Learning (Serena Young and Justin Johnson, Stanford). This inspires course development. If my university requires me to teach a course with the same title of the above signature courses, I would take those existing resources as reference, and incorporate more intuitions from myself. Picasso said, good artists copy, great artists steal. I am a great artist.

Developing a signature course

One of my dreams is to develop my own "signature course" some day. I would like to open-source my courses (with permissions) to benefit society beyond my school. I believe this will also be a good practice to develop the brand of my school and research lab.

References

- [Bell and Vahrenhold(2018)] Tim Bell and Jan Vahrenhold. 2018. CS unplugged—how is it used, and does it work? *Adventures between lower bounds and higher altitudes: essays dedicated to Juraj Hromkovič on the occasion of his 60th birthday* (2018), 497–521.