Just as it is important for researchers to develop tools for solving real-world challenges, it is important for the scientific community to prepare the next generation of scientists and software engineers to create and apply these tools. In my opinion one of the key goals of education, especially in an engineering discipline such as computer science, is teaching general and transferable problem-solving skills. While students should learn a set of fundamental tools, terminology and facts, it is important that they also are able to address real-life challenges without constant mentorship. A teacher must help students develop the necessary knowledge base as well as a confidence and ability to work independently.

I think that one of the fundamentals of effective teaching is finding problems that motivate students and keep them engaged in their work as they develop their talents. I constantly seek examples to demonstrate that information conveyed in a lecture has fun or practical application. This principle is relatively easy to implement in the field of computer graphics, where students usually can choose a project ranging from computer games to medical imaging tools; and independent of that choice, have to learn a wide range of programming techniques and mathematical tools.

Motivating students is not the only challenge a teacher faces: identifying and conveying necessary basic concepts and state-of-the-art techniques in a clear and concise form is also essential. Furthermore, I deem it important to keep a high-level course structure in mind in order to identify long-term skills and techniques that students need to learn. Finally, outlining the main objectives of a course and maintaining coherency of ideas across all lectures is also important for having a lasting impact.

As a post-doctoral scholar at Stanford University, I lectured a course on data-driven shape analysis techniques in 2014. This course covered a wide range of mathematical tools and algorithms with focus on analyzing and leveraging collections of 3D models. The course was mostly designed for senior undergraduate or new graduate students and it initiated several successful research projects. I was especially pleased by the fact that the course attracted a range of students from different departments, with different backgrounds and interests. Discussions during the course and during my office hours allowed me to re-visit many concepts from a new, fresh perspective. I continue working with two students who took this course (MS and PhD), and they are currently pursuing research projects that involve data-driven analysis and processing of 3D data.

During my graduate studies at Princeton University, I was a Teaching Assistant for two very different courses. In 2010, I taught Precepts for Computational Universe class, which covered basics of computer science for non-computer science majors. Through use of a wide range of course topics and use of engaging equipment, students from a variety of disciplines acquired basic knowledge of algorithm design, machine learning, and image rendering, while developing algorithms to synthesize music, control mini robots, or render 3D scenes. Since most of those students were not computer science majors, it was important to regularly suggest how these skills can be useful in their future

1 http://graphics.stanford.edu/courses/cs468-14-spring/
careers. In 2011 I assisted in teaching Computer Graphics to a group of students who were primarily undergraduate upperclassmen. Having a course with a strong and open-ended project component provided tremendous motivation for the students. I quickly realized that students indirectly gave me immediate feedback on the quality of my precepts and my assistance in their project by their progress in accomplishing the tasks they set for themselves. This revealed to me that simply answering a question may not be as effective as teaching students how to formulate their problem, break it down to a manageable set of subproblems, and then independently seek a set of solutions with commonly available tools such as textbooks and the Internet.

I found that many skills that I have acquired as a teaching assistant are also suitable for mentoring graduate students. Similarly, the students need to find a motivating topic, learn fundamental methods and tools, and develop a coherent perspective on development of the field. There are several projects published at SIGGRAPH Asia\textsuperscript{2}, Eurographics\textsuperscript{3}, and SGP\textsuperscript{4} for which I extensively co-advised younger graduate students.

To conclude, I am guided by three main principles in teaching: (i) engage by providing motivating problems, (ii) teach the fundamental methods, terminology and field-specific facts that are necessary to formulate the problem and solve it independently, and (iii) ensure that a course is presented as a coherent story and has lasting impact on student’s understanding of the field.

\textsuperscript{2}Creating Consistent Scene Graphs Using a Probabilistic Grammar, T. Liu, S. Chaudhuri, V. G. Kim, Q. Huang, N. J. Mitra, and T. Funkhouser, SIGGRAPH Asia 2014

\textsuperscript{3}ShapeSynth: Parameterizing Model Collections for Coupled Shape Exploration and Synthesis, M. Averkiou, V. G. Kim, Y. Zheng, and N. J. Mitra , Eurographics 2014

\textsuperscript{4}Finding Surface Correspondences Using Symmetry Axis Curves, T. Liu, V. G. Kim, and T. Funkhouser, Symposium On Geometry Processing 2012