Chapter 21
Visualizing Content for Computational Geometry Courses

Christodoulos Fragoudakis
National Technical University of Athens, Greece

Markos Karampatsis
National Technical University of Athens, Greece

EXECUTIVE SUMMARY

The instructor overhead is a major obstacle to visualization technologies. Visualization is highly effective in two and three dimensions, and these are the dimensions where computational geometry occurs in practice. The authors present a hypertext system which creates e-content for computational geometry teaching. Their hypertext system provides geometric and visualization libraries that allow the quick creation of interactive visualizations of computational geometry algorithms. Inquiry-based learning is promoted as the learners have the opportunity to observe, interact, and experiment with the produced animations. Their system utilizes the inherent expressiveness of the Python programming language, which permits coding programs that look like pseudo code, whilst easily making advanced low importance details transparent. This is crucial for pedagogical use in computational geometry courses where the focus should be on the geometric algorithmic aspects, with low level details made abstract.

DOI: 10.4018/978-1-4666-0068-3.ch021

INTRODUCTION: MOTIVATION

A *Computational Geometry* course focuses on *Geometric Algorithms* which lie at the heart of many applications, ranging from protein modeling in biology and motion planning in robotics to virtual reality engines and computer graphics in games. A typical course introduces the main techniques from computational geometry like convex hulls, triangulations, Voronoi diagrams, visibility, art gallery problems and motion planning. In a computational geometry class, *visualization* is an effective tool in communicating ideas.

Visualization is highly plausible in two and three dimensions, and these are the dimensions where computational geometry action occurs in practice. Visualization can have a great impact in education. Watching and interacting with an algorithm can enhance understanding, give insight into geometry, and explain the intuition behind the algorithm. During the past decade there has been noticeable progress in the production of visualizations of geometric algorithms and concepts. Although the field of scientific visualization has received much attention and research, it has not been widely adopted by instructors. Algorithm designers want to visualize their algorithms but are limited by current tools. The design and creation of algorithm visualizations and it's integration with other learning materials, even for a simple task of three-dimensional geometric constructions is demanding a time consuming. Visualizations would be less rare if the effort to create them was little. Methods of designing algorithm visualization systems are still being explored. There are several algorithm animation systems, for example AGE, BALSA and TANGO. Each has several pros and cons, but unfortunately, all of them require a good deal of effort to implement a reasonable algorithm animation (Brown & Sedgewick, 1984; Stasko, 1990).

We propose the creation of a hypertext system that can create e-content for computational geometry teaching. The system should allow for the quick and simplified creation of two or three dimensional algorithm visualizations. Naps et al. (2002) point out that the instructor overhead is a major obstacle to visualization technology. The hypertext system should address this matter so that even highly complex geometric algorithms can be visualized with ease. This is an important consideration, since complicated algorithms gain the most from visualizations. The viewer should be able to observe, interact and experiment with the animations.

This chapter is organized as follows: In the next section the forms of learner engagement with computational geometry visualization technology is briefly introduced and specific effects of visualization in a computational geometry class is illustrated. We argue why Python is a suitable language for the programming of geometric algorithms and visualizations and illustrate a learning scenario, using GNOSIS, which is a prototype system for geometric algorithm visualization. In the