What Should We Teach in a Scientific Visualization Class?

Panel Organizer:
Jon D. Genetti, University of Alaska Fairbanks

Panelists:
Michael J. Bailey, Oregon State University
Jon D. Genetti, University of Alaska Fairbanks
David H. Laidlaw, Brown University
Robert J. Moorhead, Mississippi State University
Ross T. Whitaker, University of Utah

Introduction

Scientific Visualization (SciVis) has evolved past the point where one undergraduate course can cover all of the necessary topics. So the question becomes "how do we teach SciVis to this generation of students?" Some examples of current courses are:

- A graduate Computer Science (CS) course that prepares the next generation of SciVis researchers.
- An undergraduate CS course that prepares the future software architects/developers of packages such as vtk, vis5D and AVS.
- A class that teaches students how to do SciVis with existing software packages and how to deal with the lack of interoperability between those packages (via either a CS service course or a supercomputing center training course).
- An inter-disciplinary course designed to prepare computer scientists to work with the "real" scientists (via either a CS or Computational Science course).

In this panel, we will discuss these types of courses and the advantages and disadvantages of each. We will also talk about some issues that you have probably encountered at your university:

- How do we keep the graphics/vis-oriented students from going to industry?
- How does SciVis fit in with evolving Computational Science programs?
- Is SciVis destined to be a service course at most universities?
- How do we deal with the diverse backgrounds of students that need SciVis?

Position Statements

Michael J. Bailey

At the University of California San Diego, and now Oregon State University, I have treated scientific visualization courses as a chance for students to gain experience with different ways of displaying data to enhance insight. At times, we have dabbled with having the students work with canned visualization packages such as AVS and OpenDX, or using libraries such as VTk. But, we ended up realizing that a lot of a students understanding of visualization comes from doing the implementation themselves so that they have a better feel for exactly how the scene they see on the screen relates to the data they entered. Also, this class oftentimes is used as a springboard for the students graduate research in their own area of science or engineering. We have found that, in these cases, students need to integrate visualization into an existing simulation or analysis program, and thus it is more useful to be able to implement visualization algorithms themselves than it is to use standalone programs.

The course has evolved over the years and has taught methods such as hyperbolic geometry, data display using range sliders, contours, isosurfaces, volume rendering, stereographics, and Delauney triangulation. At times the students in the class have also used advanced graphics hardware such as the Volume Pro card, solid freeform fabrication machines, and the Perspecta true 3D display.

My philosophy is that visualization is really a big bag of tricks. When presented with a new dataset, an experienced user starts pulling tricks from the bag to gain the most insight possible. It is important, then, to acquaint the students with many techniques and give them the necessary hands-on experience to use them wisely.

Jon D. Genetti

At the University of Alaska Fairbanks, we have many research programs (mostly in the physical sciences) with unique visualization requirements. Examples include sea ice modeling, volcano plume modeling, ionospheric modeling, forest fire propagation, permafrost modeling, oil and gas development, and fisheries management. These efforts provide a diverse group of students *and* researchers for any type of visualization class. Our experience has been that they are not patient enough for a full semester course and prefer a one week (or even better, a one day!) training course to show them how to "do the vis."

So, if you can't train the researchers, then you will need to develop SciVis skills in students along with their ability to work with "real" scientists. The researchers generally prefer to fund a SciVis graduate student, as that doesn't detract from the work already in progress. The next goal is to keep these graduate students as full-time employees when they graduate, as they have the skills needed for collaborative visualization.

David H. Laidlaw

At Brown I have taught CS courses that fit into each of the four suggested categories. But my primary goal for a SciVis class is to prepare the next generation of SciVis researchers. In addressing this goal, students do learn how to use existing packages when they

IEEE Visualization 2004
October 10-15, Austin, Texas, USA
0-7803-8788-0/04/$20.00 ©2004 IEEE

are appropriate. And they work with “real” scientists on projects, learning how to develop research that is collaborative.

I’ll talk about three classes that I teach. One is structured around doing collaborative scientific visualization research by creating a grant proposal, getting it funded, doing the research, and reporting on it – all in a single semester. The second is a Brown course that is jointly listed at the Rhode Island School of Design. In this class half of the students are visual design students and half are computer science students. Together, they learn about virtual reality displays, about 3D time-varying fluid flow, and about working together. The class culminates in a set of group projects that bring visual design and software design together to facilitate science. Finally, I teach a senior-level software engineering class. How does that fit in? You’ll just have to come to the panel to find out.

Robert J. Moorhead

At Mississippi State University, I have taught “SciVis” courses to groups of students ranging from non-CS undergraduate to doctoral students using most of the approaches mentioned in the introduction.

I think the best way to teach SciVis to this generation of students is student-dependent. We must teach a plethora of ways. First, for those students who are interested in multi-disciplinary computational sciences, in the first course we need to introduce maybe one package, but require several programs dealing with classic scalar/vector techniques, addressing issues of data representation both from computational simulations, as well as to the graphics subsystem. We must teach them critical facts about color spaces, perception, and signal processing, and try to impart to them what exists, what works, and what deceives.

For those students who are interested in SciVis research as a career, we need to build on that knowledge with courses on graphics hardware, virtual reality, real-time rendering, geometric modeling, as well as novel visualization algorithms.

Ross T. Whitaker

Currently, scientific visualization is taught as a *craft*. There is little attention to principles and a great deal of emphasis on experience, aesthetics, and heuristics. As a result many courses in scientific visualization consist of merely of reading collections of papers (e.g. Vis Proceedings), experimenting with specific algorithms on specific data sets, or learning visualization software packages.

Because visualization relies on human perception and interaction, it will always require a certain level of visual design, creativity, and hands-on experience. Furthermore, it is sensible that one the basic tools for teaching visualization will be case studies. However, as the field matures it will start to look more like an engineering discipline. That is, scientific visualization will be an application of principles from a set of other related fields. The relevant fields are numerous, and not all researchers would agree on the ideal subset that should be addressed in the classroom. However, many people do agree that visualization relies heavily on the fields of computer graphics, discrete and differential geometry, numerical and scientific computing, topology, signal processing, visual perception, and graphic design.

The breadth of knowledge required to practice scientific visualization presents some difficult challenges when designing an introductory course. The strategy at the University of Utah is to focus on teaching these underlying technologies at a introductory level simultaneously with a toolbox of standard visualization techniques, which lead to direct applications on real data. The focus is on basic technologies and experience with real data using known algorithms, rather than hard-core programming. This makes the course accessible to people with a wide variety of moderately technical backgrounds including engineering as well as the physical social sciences.

Biographical Sketches

Michael J. Bailey
mjb@eecs.oregonstate.edu

Mike Bailey is a Professor in the School of Electrical Engineering and Computer Science at Oregon State University. Mike has been involved in various aspects of computer graphics research and teaching for the past 25 years. Mike’s areas of interest include scientific visualization, GPU-accelerated high performance computer graphics, solid freeform fabrication, geometric modeling, and computer aided design and analysis. Specific projects include hardware-accelerated volume rendering, solid models for visualization hardcopy, and augmented reality in the field. Mike received his PhD from Purdue University in Computer Graphics/Computer-Aided Design.

Jon D. Genetti
genetti@cs.uaf.edu

Jon D. Genetti is an Associate Professor of Computer Science at the University of Alaska Fairbanks and a joint faculty member with the Arctic Region Supercomputing Center. His current research interests are the design and implementation of custom rendering algorithms for large and novel datasets and collaborative research with the physical sciences unique to Alaska and the polar regions. His PhD in Computer Science is from Texas A&M University and he spent five years as a Computer Scientist at the San Diego Supercomputer Center.

David H. Laidlaw
dhl@cs.brown.edu

David H. Laidlaw is an Associate Professor in the Computer Science Department at Brown University. His research centers around applications of visualization, modeling, computer graphics, and computer science to other scientific disciplines including, archaeology, developmental neurobiology, medical imaging, orthopedics, art, cognitive science, remote sensing, and fluid mechanics to develop new computational applications and to understand their strengths and weaknesses. Particular interests include visualization of multi-valued multidimensional imaging data, comparisons of virtual and non-virtual environments for scientific tasks, and applications of art and perception to visualization. His PhD in Computer Science is from Caltech, where he also did post-doctoral work in the Division of Biology.

Robert J. Moorhead
rjm@erc.msstate.edu

Robert J. Moorhead II received the PhD degree in electrical and computer engineering and the MSEE degree from North Carolina State University in 1985 and 1982, respectively. He received the BSEE degree summa cum laude and with research honors from Geneva College in 1980. He is the Director of Visualization, Analysis, and Imaging Lab in the GeoResources Institute and a professor of electrical and computer engineering at Mississippi State University. He previously worked as a research staff member in the Imaging Technologies Department at the IBM T. J. Watson Research Center from 1985 to 1988. He has authored more than 90 papers or book chapters on visualization, image processing, and computer communications. He has received funding from ARPA, ONR, NRL, AFOSR, the Army Waterways Experiment Station (now ERDC), the Naval Oceanographic Office, NASA, Raytheon, CSC, and Logicon. He was the lead conference cochair for the IEEE Visualization '97 Conference, the chair of the IEEE Computer Society's Technical Committee on Visualization and Graphics in 1999 and 2000, and the lead papers cochair for the IEEE Visualization 2002 Conference.

Ross T. Whitaker
whitaker@cs.utah.edu

Ross T. Whitaker received the BS degree in electrical engineering and computer science from Princeton University in 1986 summa cum laude. From 1986 to 1988, he worked for the Boston Consulting Group, entering the University of North Carolina at Chapel Hill (UNC) in 1989. At UNC, he received the Alumni Scholarship Award, and received the PhD degree in computer science in 1994. From 1994-1996, he worked at the European Computer-Industry Research Centre in Munich, Germany, as a research scientist in the User Interaction and Visualization Group. From 1996-2000, he was an assistant professor in the Department of Electrical Engineering at the University of Tennessee. Since 2000, he has been at the University of Utah, where he is an associate professor in the College of Computing and a faculty member of the Scientific Computing and Imaging Institute.