Animation and Visualization Tools: From Undergraduate Projects to Pedagogical Aids

David J. Russomanno and J. Caleb Goodwin

The University of Memphis

I. INTRODUCTION

The University of Memphis is not unlike many other doctoral/research extensive institutions in that many of our undergraduates do not participate in a service-learning experience that is directly linked to their program of study. One goal of our interdisciplinary data visualization course, established as part of a Course, Curriculum, and Laboratory Improvement (CCLI) grant funded by the National Science Foundation (NSF), is to make improvements in this situation. Students of our course are encouraged to propose, design, and implement a final project that supports one of two objectives: 1) the development of an application that may have utility in the high-school classroom; or 2) an application that may support a research experience for undergraduates. In some instances, students choose to improve or extend the functionality of a previous term project based on end-user feedback. The course is an elective and not part of the capstone design experience in the engineering or computer science curricula.

Not many high-school teachers or students have significant exposure to animation or visualization software used or developed by engineers and scientists. As stated by Beattie et al. [1]: "Visualization of scientific processes offers a new dimension in communicating the excitement of science and vividly introducing K-12 educators to both the richness and perplexing complexity of natural and technological processes." Our discussions with teachers from our local high schools, who are also serving as consultants to our NSF grant, confirm this statement, and subsequently, the high-school teachers themselves are becoming active participants and researchers in our project.

Although there are extensive, ongoing research and development efforts at other institutions to transform STEM education by the use of modern tools as pedagogical aids [2, 3]; our project is unique in that we are encouraging the proposal, design, and implementation of tools by undergraduate students rather than by professional software teams. An argument can be made that in some cases equivalent or superior tools to those being developed by our undergraduate students are available via academic or commercial products [4]. However, having our undergraduates participate in a collaborative development process, with high-school teachers and their students viewed as clients, provides valuable software engineering experience in joint application design and incremental product improvement. Also, it is more likely that teachers will experiment with the applications given that they had some input on the tools' requirements specification. The teachers provide critical feedback and have confidence that their suggestions will appear in subsequent tools since they are locally developed and maintained.

Although anecdotal, undergraduate students appear to be satisfied knowing their applications may have utility in the local community after the semester ends. Moreover, our students appear to be more effectively incorporating software engineering best practices in their work, knowing that their designs and implementations may be subsequently extended by their peers. Hopefully, besides enhancing the overall learning experience, an additional outcome of our project will be that more high-school students in our region will consider careers in engineering, computer science, or other STEM fields, given examples of the types of tools they could develop or research problems they could encounter as part of their undergraduate program of study.

It is clear that new educational technology, including animation and visualization tools, provides a means for infusing enthusiasm and excitement into traditional paper-pencil techniques that are often perceived as boring. Unfortunately, many students appear to lose an interest in STEM fields at an early age and *write off* careers in these fields before they are engaged by the subject matter [5]. Therefore, it is critical to provide strategic opportunities for students to use technology that they perceive as exciting, including computer games, cell phones, iPods [6], etc., as well as animation

Abstract

Animation and visualization tools are being designed, implemented, and maintained primarily by undergraduate students at The University of Memphis as part of an interdisciplinary data visualization course established via an NSF/CCLI grant. Currently, the tools consist of modules in projectile motion, conservation of momentum, and elementary airplane dynamics, as well as tools for visualizing large medical domain data. Students are involved in a requirements elicitation process, with input provided by high-school teachers, to develop applications that may have pedagogical utility in high schools in the Mid-South area. Teachers are using the tools in the classroom and acquiring data to assess the tools effectiveness as pedagogical aids, while undergraduate students are receiving feedback regarding their software designs.

and visualization tools, as early as feasible into their education; to bring such tools into K-12 students perception of science, and to facilitate their development of a passion for and the confidence to pursue careers in science and engineering. The demographics of the Mid-South region provide a large pool of underrepresented students; therefore, our efforts have the potential to increase the percentage of these students who consider and subsequently pursue STEM careers.

II. METHODS

Our course teaches animation and visualization techniques and related applications [7]. An additional outcome is that students develop the pragmatic skills that are required to process scientific and engineering data and build custom software applications using the Visualization Toolkit (VTK) [8] in an interdisciplinary, team environment. Since one of our primary objectives is to develop and maintain a local repository of animation and visualization tools for the high-school classroom, and to support a variety of research endeavors, we chose to use VTK rather than a turn-key visualization package.

By using VTK rather than a variety of shrink wrap packages, engineering and science students not only acquire understanding of basic animation and visualization techniques, but they also gain competency using a software development platform to create/modify/test algorithms and to design user interfaces while contributing to the implementation of custom applications within a unified environment.

Using VTK to implement animation and visualization applications fits well within the object-oriented software engineering process of requirements elicitation, analysis, system design, object design, implementation, test, and maintenance. Furthermore, a key objective of the NSF/CCLI Adaptation and Implementation (A&I) program is to improve undergraduate STEM education through adaptation and implementation of effective materials, techniques, and practices to result in a positive change at an institution [9]. We adopted VTK based in-part on its widespread use in other visualization courses throughout the world [10-13]. In addition, VTK maintains an open source code repository, along with extensive web-based documentation and discussion groups, which further lends itself well to custom development. Moreover, VTK is available free of charge.

The key concepts of our data visualization

Fig. 1: Data Visualization Applications (Integrated)

course are implemented using the scripting language Tcl/Tk (pronounced by many as tickle tee-kay) [14] along with visualization functionality provided within the VTK framework. Tcl/Tk is also available free of charge. It has been our experience that Tcl/Tk is relatively easy for students to learn on their own provided they already know at least one high-level programming language. Since Tcl/Tk is an interpreted language, which does not require compilation and linking processes before execution, it is more straightforward than C/C++ or Java for many students to use for experimenting with animation, visualization, and user interface design techniques, and requires significantly fewer lines of code than C/C++ or Java for most problems. Finally, Tcl/Tk provides a neutral "playing field" for an interdisciplinary course since students in our STEM curricula have most likely not had any prior experience with it.

III. APPLICATIONS

As part of their coursework, our undergraduate students have developed animation and data visualization applications that may have utility in the high-school classroom. Some of these applications, including modules on projectile motion, conservation of momentum, airplane dynamics, etc., have been integrated into a single application as shown in Fig. 1. Other applications provide facilities for visualizing large-scale data produced from biomedical engineering research that may also be interesting to high-school students and serve as a potential recruitment mechanism. Undergraduate and graduate students, as well as engineering faculty members, are visiting local high-school physics classes to discuss animation and visualization applications in more detail and to also discuss how their research builds upon topics currently being studied by

the high-school students. The descriptions of representative applications are discussed in more detail in the following subsections.

A. Projectile Motion Module

Using Cartesian coordinates, users (i.e., teachers or students) can set the initial position <x,y,z> of a projectile. Since the concept of a vector's components is to be emphasized, users can specify the projectile's launch in a variety of modes including: i) in terms of the projectile velocity vector's components; or ii) in terms of projectile orientation, launch angle, and velocity magnitude. Details such as the maximum height, horizontal distance traveled, and total flight time is displayed for each launch as shown in Fig. 2. The user can step through the entire sequence and optionally view the associated resultant velocity vector, or view the vector components scaled by magnitude at each point in time or at specified intervals during the projectile's flight. Acceleration values are also provided. In addition, the equations governing the horizontal and vertical motion can be turned on/off as desired along with the various parameter values, which are updated during the projectile's trajectory.

High-school teachers recommended the start/stop and variable slow-motion features for the animation as it provides the opportunity to discuss the vectors and other values associated with each point of the projectile's trajectory. After a few iterations to improve the functionality of the application, teachers have evaluated it as superior to several similar, widely available tools due to the flexibility to control various animation parameters to better integrate the tool with the teachers' pedagogical approach.

B. Conservation of Momentum Module

The conservation of momentum module animates two-dimensional (2D) collisions between billiard balls. The topic was chosen since ball collisions are a classic high-school physics topic. The module can animate two scenarios for 2D elastic collisions: (1) a collision involving one stationary ball; and (2) a collision involving two moving balls. The user may input the angle of impact, as well as the mass and velocity of each ball, respectively. The user can also step through the entire animation sequence which produces traces showing each ball's movement. The simulation is built using the physics relationships that are used in a typical high-school course. A view of the underlying equations can be optionally turned on/off by the user during the animation. Given that the

Fig. 2: Projectile Motion Module

parameters for the balls may be changed, the instructor can provide both an interactive and educational experience facilitating discussion and demonstration of the various collision scenarios in the classroom.

C. Airplane Dynamics Module

The airplane dynamics module animates, using basic vector mathematics, how an airplane is affected by headwinds, tailwinds, and crosswinds. Through the use of algebra and
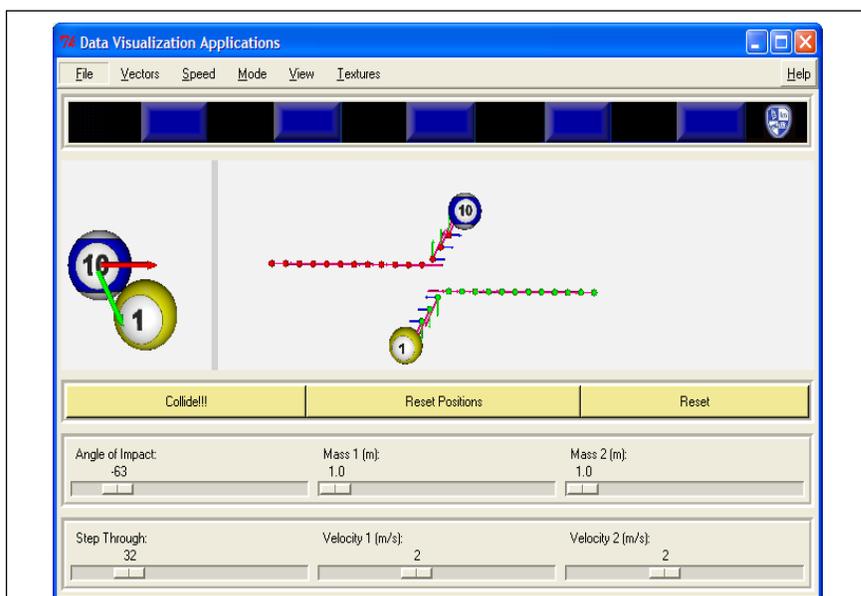
Fig. 3: Conservation of Momentum Module

trigonometric functions, it is shown how a plane must correct for crosswinds, how the resulting ground speed of the airplane differs from its airspeed, and how a plane can fly in a direction different from where the nose of the plane is pointed. Waypoints are also calculated and optionally depicted in the animation to illustrate where the plane needs to be pointed to reach its intended destination, as well as the location where the plane would arrive if it did not correct for a crosswind. The plane can be launched with or without correction for crosswinds. The user can also enter the various parameters including the headwind/tailwind, crosswind, airspeed, and distance to travel. Finally, the user can turn on/off the equations used to generate the simulation.

Although the module is an extension to some of the projectile motion concepts, the animation provides concrete examples of vector forces in a problem that is familiar to the majority of high-school students. Moreover, the module is a good example of how undergraduate students are extending the work of their peers to address refined requirements obtained from the high-school teachers.
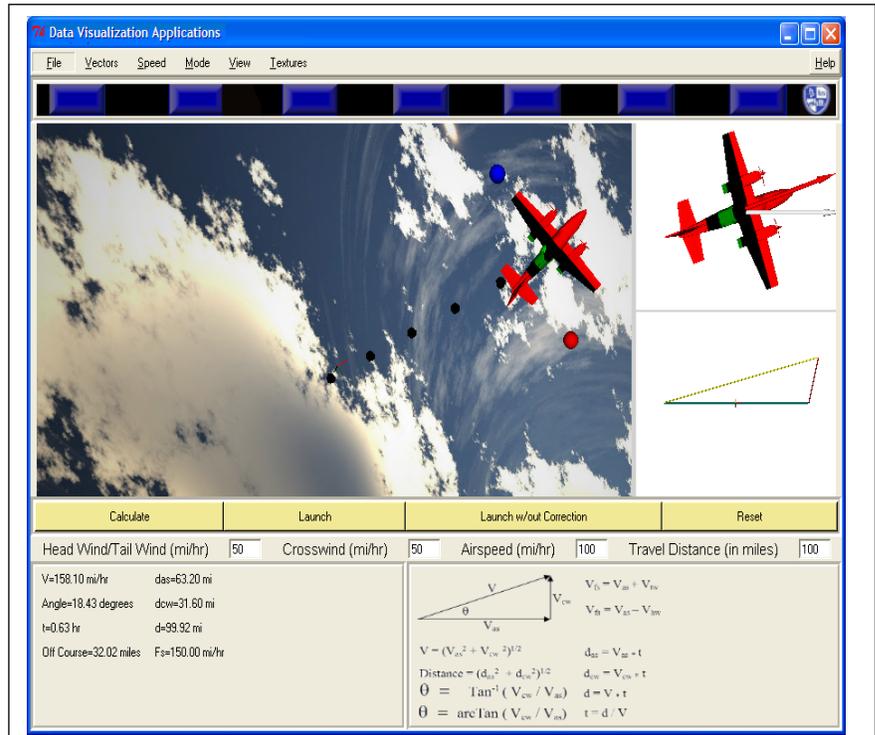
D. Medical Data Visualization Applications

The projectile motion, conservation of momentum, and airplane dynamics modules are primarily animation tools rather than true data visualization applications. Although not integrated with the other modules, several additional applications have been developed as part of our project to visualize large medical data sets. These applications have focused primarily on complementing other on-going research efforts within the Herff College of Engineering at The University of Memphis but these applications may also have some utility in high-school physics, anatomy, and possibly several other classes.

Curry et al. [15-17] have developed a three-dimensional (3D) anatomically realistic, finite element model of the human torso to generate defibrillation simulations. The model supports varying the parameters for conductivity and anisotropy for regions of the torso, including the skeletal muscle, lungs, epicardium, left atrium, right atrium, left ventricle, right ventricle, and various vessels of the heart. The model computes electric field distribution in the torso given electrode locations, applied voltage, and tissue conductivities. Some students of our data visualization course have developed applications to view the various tissues that

Fig. 4: Airplane Dynamics Module

comprise the 3D human torso. Other students have developed software modules to visualize the data produced from this simulation system. One such module provides a graphical user interface for the interactive change of the shape, size, and position of a defibrillator's electrodes and the magnitude of the applied shock as
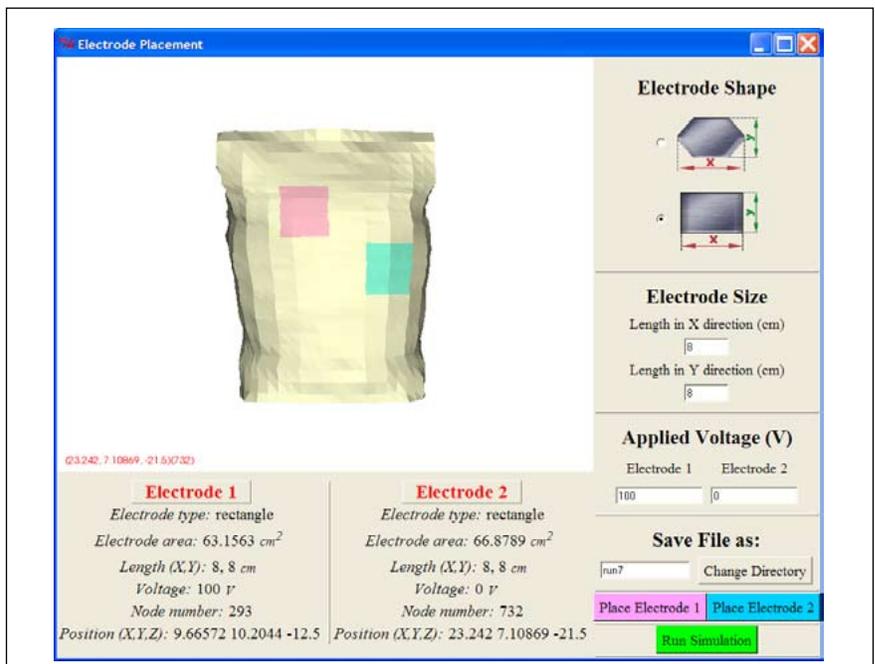
Fig. 5: Electrode Placement for Defibrillation Simulation

shown in Fig. 5 [18]. Another module provides functionality for visualizing slices through the myocardium such that the distribution of the magnitude of the electric field can be inspected as shown in Fig. 6. These various modules have been integrated into a comprehensive defibrillation simulation and visualization tool.

Ultimately, the primary goal of the defibrillation simulation and visualization tool is to enhance the understanding of how electrode parameters relate to the change of the electric field values throughout the heart which may enhance the design of both external and internal defibrillators. Besides the research utility of the defibrillation simulation and visualization tool, an additional use is to introduce high-school students to applications of electromagnetic field theory in "real world" engineering problems and to illustrate the role that undergraduate students can have in research projects.

Our high-school teachers have reported that too often their students say that the theoretical physics topics that they study in the classroom lack the context of application. Some high-school physics students had formed the impression that since the material they were studying had long ago been discovered, it no longer has any significant bearing on the practical design of modern-day products. Demonstration of the defibrillation simulation and visualization tool to high-school physics students provides a STEM recruitment opportunity while also providing context and motivation for the students to learn the fundamentals currently being presented in their classroom.

Although anecdotal, these applications appear to be correcting some student misconceptions and enhancing their perceptions of STEM careers. Other applications developed as part of this project but not yet incorporated into the high-school experience include:

1. A utility to clip, cut, and turn on/off views of torso tissues, as well as to change their respective view parameters, such as transparency, as shown in Fig. 7.
2. A utility to select MRI slices for visualization using a geometric rendering of the torso for slice selection, as shown in Fig. 8. This particular project combines both geometric and volumetric rendering techniques within a single application.
3. Marker gene selection and evaluation in brain tumor patients using multi-dimensional data visualization techniques [19]. Ultimately, if marker genes can be precisely located then tumor prediction

may be dramatically improved before the disease manifests itself.

4. Visualization and graphics-based approaches to computer security by limiting automated access to websites via completely automated public tests to tell computers and humans apart (Captchas) [20].

A website is maintained for both the data visualization course and the overall NSF project. The course website is located at http://engronline.ee.memphis.edu/eece4731/ and it is updated during each semester in which the course is offered. The NSF/CCLI project website is located at http://engronline.ee.memphis.edu/nsf_ccli_0410290/. The project website is
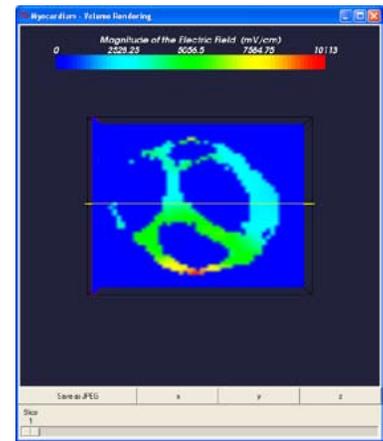
Fig. 6: Myocardium Slice after Shock
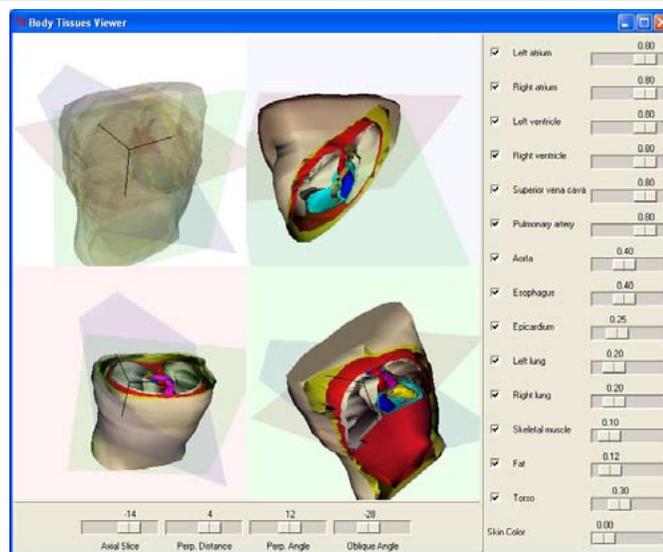
Fig. 7: Body Tissue Viewer
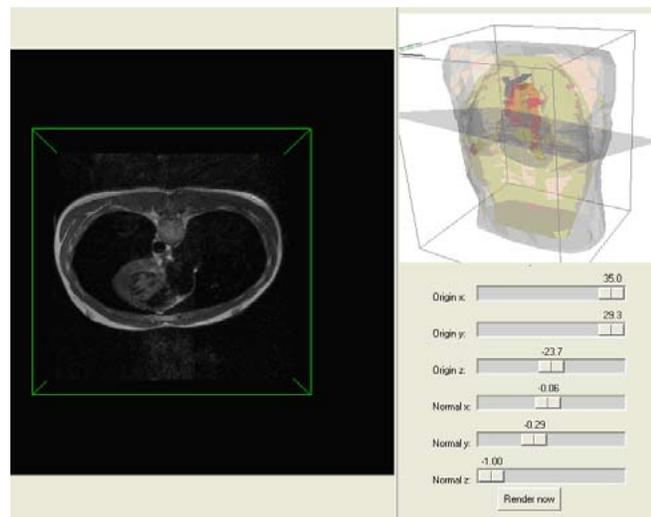
Fig. 8: MRI Viewer with Geometric Torso Slice Selector

updated with meeting minutes and other notes from workshops held with teachers and other project participants. Some project suggestions, such as an adaptive periodic table, elicited from high-school teachers are not yet implemented and these materials may be suitable for adoption by others.

Our project began September 1, 2004. The data visualization course was first offered during the spring semester 2005 and it will be offered once per year for the foreseeable future; therefore, significant data for a meaningful assessment is ongoing and results will be disseminated as they become available. An outline of the learning objectives, measurable activities, and assessment instruments is discussed in our overview paper of the project [7], which also discusses various aspects of the visualization tools, but not at the level of detail provided in this paper.

IV. CONCLUSIONS

This paper has emphasized the animation and visualization tools that have been implemented as part of an ongoing NSF/CCLI sponsored project to develop an interdisciplinary data visualization course with a significant community-based experience for the undergraduate students who take the course. The project is unique in-part because it strives to team undergraduate students with high-school teachers to define and implement animation and visualization applications that may have utility as pedagogical aids in the high-school classroom.

Developing tools that add more realism to the theory and topics studied in high-school mathematics, physics, or anatomy classes (e.g., magnetism, potential, voltage, motion, organs, tissues, etc.) integrated with practical, data-intensive applications may bring undergraduates' work closer to contemporary challenges that might interest them while also providing a forum for the recruitment of high-school students. Finally, high-school teachers are attempting to determine if the activities that use the tools provide context and motivation for high-school students to consider pursuing careers in STEM fields.

ACKNOWLEDGMENTS

REFERENCES

[1] C. Beattie, R. Kriz, and C. Ribbens, "Scientific Modeling and Visualization Classroom Planning Document," Virginia Tech, Blacksburg, VA, http://www.sv.vt.edu/future/vizclass.html, 1995 (current Sep., 2006).

[2] C. Wieman and K. Perkins, "Transforming Physics Education," *Physics Today*, pp. 36-41, Nov., 2005.

[3] K. Perkins, W. Adams, M. Dubson, N. Finkelstein, S. Reid, C. Wieman, and R. LeMaster, "PhET: Interactive Simulations for Teaching and Learning Physics," *The Physics Teacher*, vol. 44, no. 1, pp. 18-23, 2006.

[4] W. Christian and M. Belloni, *Physlet Physics: Interactive Illustrations, Explorations and Problems for Introductory Physics*, Pearson Education Inc., Upper Saddle River, NJ, 2004.

[5] T. Smith, B.A. Young, Y. Bae, S. Choy, and N. Alsalam, "Women in Mathematics and Science," *The Condition of Education 1997* (NCES 97-388), National Center for Education Statistics, http://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=97982, 1997 (current Oct., 2006).

[6] L. Shallcross, "The POD Squad," *ASEE Prism*, Sep., 43-45, 2006.

[7] D.J. Russomanno, D. Franceschetti, A. de Jongh Curry, and A. Phillips-Lambert, "An Interdisciplinary Data Visualization Course with an Ongoing Community-Based Project Component," *Computers in Education Journal*, vol. 16, no. 3, 29-39, 2006.

[8] W. Schroeder, L.S. Avila, and W. Hoffman, "Visualizing with VTK: A Tutorial," *IEEE Computer Graphics and Applications*, 20-27, September/October, 2000.

[9] Anonymous, "Supplemental Information for Principal Investigators and Applicants to NSF's Course, Curriculum, and Laboratory Improvement Program," *NSF 00-117*, Jun., 2000.

[10] Anonymous, "Visualization Course Website," http://www.cs.iupui.edu/~sfang/cs552/cs552.html (current Sep., 2006).

[11] J. Kraak, "Visualization Course Website," http://www.rug.nl/rc/hpcv/visualisation/VTK/notes/notes.html (current Sep., 2006).

[12] Anonymous, "Visualization Course Website," http://www.cs.unc.edu/Courses/comp215-s06/ (current Sep., 2006).

[13] G. Watson, "Visualization Module," The School of Informatics, The University of Edinburgh, http://www.dcs.ed.ac.uk/teaching/cs4/www/visualisation/course.html (current Sep., 2006).

[14] B. Welch and K. Jones, *Practical Programming in Tcl and Tk*, Prentice Hall PTR, 2003.

[15] A. L. de Jongh, E. G. Entcheva, J. A. Replogle, R. S. Booker, B. H. Kenknight, and F. J. Claydon, "Defibrillation efficacy of different electrode placements in a human thorax model," *Pacing Clin Electrophysiol*, vol. 22, pp. 152-157, 1999.

[16] L. Hunt and A. L. de Jongh Curry, "Electrode placement significantly affects transthoracic atrial defibrillation thresholds," *Cardiovascular Engineering*, vol. 10, no. 1, pp. 62-68, 2005.

[17] L. Hunt and A. L. de Jongh Curry, "Finite element computer modeling of atrial defibrillation," *Proceedings of 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2004.

[18] G. Atanasova, D.J. Russomanno, A. de Jongh Curry, and L. Hunt, "Interactive visualization tool for electrode placement and assessment of transthoracic defibrillation thresholds," *Proceedings of the 38th Southeastern Symposium on Systems Theory*, IEEE Press, Cookeville, Tennessee, pp. 61-65, 2006.

[19] A.U. Islam, K. Iftekharuddin, and D.J. Russomanno, "Marker Gene Selection Evaluation in Brain Tumor Patients Using Parallel Coordinates," *Proceedings of the 2006 International Conference on Bioinformatics and Computational Biology*, CSREA Press, Las Vegas, Nevada, pp. 172-173, 2006.

[20] M. Hoque, D.J. Russomanno, and M. Yeasin, "2D Captchas from 3D Models," *Proceedings of the IEEE SoutheastCon*, Memphis, TN, pp. 165-170, Mar., 2006.

David J. Russomanno is Ballard Professor and Chairman of The Department of Electrical and Computer Engineering at The University of Memphis. He received the B.E.E. degree in Electrical Engineering from Auburn University in 1986, the M.E. in Electrical and Computer Engineering and the Ph.D. in Computer Engineering from the University of South Carolina in 1989 and 1993, respectively. Prior to joining U of M, Russomanno was employed by Intergraph Corporation, Pratt and Whitney Aircraft, and Michelin Tire Corporation. Russomanno's interests include engineering and computer science education, intelligent sensors, knowledge representation for the Semantic Web, logic programming applications, and data/knowledge migration and visualization applications. He is a senior member of IEEE, and also a member of ASEE and ACM.

J. Caleb Goodwin is a graduate student in the Department of Electrical and Computer Engineering. He received his B.S. degree in Computer Engineering from The University of Memphis in 2005. His interests include intelligent sensors, artificial intelligence, and data visualization.