Chen, W. et al. (Eds.) (2017). Proceedings of the 25th International Conference on Computers in Education.
New Zealand: Asia-Pacific Society for Computers in Education

The Journey to Improve Teaching Computer Graphics: A Systematic Review

Thomas SUSELO*, Burkhard C. WÜNSCHE & Andrew LUXTON-REILLY
University of Auckland, New Zealand
*tsus609@aucklanduni.ac.nz

Abstract: Computer graphics is often regarded an exciting and enjoyable subject due to it combining technology, art and creativity. The past few years have seen a rapid evolution of the field with novel consumer-level devices (e.g. head-mounted displays) and media (e.g. 3D videos on YouTube) enabling a much wider section of the population to experience and create 3D content. However, teaching computer graphics can be challenging due to it requiring a diverse range of skills such as mathematics, physics, programming, spatial reasoning, problem solving, and art and design. Several researchers have acknowledged this problem and have attempted to make computer graphics teaching easier and more effective. However, so far no consensus seems to exist about the key problems teachers need to overcome and what concepts and methodologies might help with this. In this paper, we address this issue by conducting a systematic literature review identifying reported challenges, methodologies, and approaches for teaching computer graphics. Our research offers practitioners new insight into computer graphics teaching, which we hope will be useful for curriculum design, developing more effective tools and support for struggling students, and suggesting avenues for future research.

Keywords: computer graphics, systematic review, teaching, learning, teaching methodology

1. Introduction

Computer graphics is a challenging subject to teach and learn (Du and Shu, 2011; Han et al., 2008; He and Zhao, 2012; Peternier et al., 2006; Sueyasu et al., 2010; Zhou et al., 2010) because it involves a wide variety of skills such as mathematics, programming, physics, cognitive psychology, spatial reasoning, problem-solving, human-computer interaction, and art and design.

Over the past decade, many researchers have tried to address this problem and several specialised teaching tools have been developed. For example, Spalter and Tenneson (2006) present an interactive Java-based application called Graphics Teaching Tool (GTT) that allows students to study objects and understand what transformations have been applied to them and what the resulting transformation matrix is. Sueyasu et al. (2010) developed a Simplified Language for Graphics Programming (SLGP) and claimed it improves students' productivity in computer graphics developments significantly. However, no consensus seems to exist what specific issues make computer graphics difficult to teach, and what teaching methodologies and concepts should be used to address this.

We could not find previous work reviewing research on teaching and learning of computer graphics. In this paper, we will systematically evaluate the literature in order to identify the challenges in teaching computer graphics and identify promising concepts and methodologies for addressing them.

2. Methodology

We use the protocol by Kitchenham et al. (2010) as a framework for identifying relevant literature on teaching and learning of computer graphics and for describing challenges, methodologies, and approaches. Our study goals are captured by the following research questions:

- Q1. What common issues are reported in teaching and learning computer graphics?
- Q2. What approaches in the research articles are used to improve teaching computer graphics?

2.1 Searching Process

For our systematic literature review, we used four well-known digital libraries: ACM Digital Library, IEEExplore, SpringerLink, and ScienceDirect. A detailed search was run on all four databases on a single day (16 April 2017). The following search criteria were employed: (1) the keywords "3D" or "teaching" or "learning" are contained in the metadata (such as title, abstract, keywords); (2) the keywords "graphics" or "computer graphics" are mentioned in the article title. As not all the databases supported advanced searching in the same way, the searching process was adapted where required to obtain equivalent results. The publication date of articles was restricted to the years 2000 and 2017 since around that time consumer-level graphics hardware started to become common and consequently the importance of low-level algorithms (such as rasterization and visibility determination) reduced.

2.2 Data Collection

A total of 622 articles were collected from the four databases with 165 articles being obtained from the full-text collection of the ACM Digital Library and 230 articles from IEEE Explore using the automated filtering configuration. Searching SpringerLink and ScienceDirect resulted in 154 and 73 articles, respectively.

A two stage filtering was used to produce the primary study data set of relevant articles:

1. Exclude papers not about computer graphics teaching (115 articles remaining),
2. Exclude duplicates (45 articles remaining).

In order to analyse the remaining 45 articles, we identified relevant attributes and themes based on key issues discussed in those papers and the data required to answer our research questions. The following attributes/themes were collected for each paper: (1) type of publication, (2) publication venue (journal or conference name), (3) year of publication, (4) identified problems with teaching computer graphics, (5) proposed teaching approaches and methodologies (6), utilised and/or proposed teaching tools (programming language, API, specialised software), and (7) how the effectiveness of the new teaching approach was measured.

3. Results

From the 45 identified papers 15 were published in journals, and 30 were published in conference proceedings. 22 articles were published in venues related to computing education, 14 articles were published in computer graphics venues, and 9 in general Computer Science venues. 10 articles were published between 2000 and 2005, 22 articles between 2006 and 2011, and 13 articles in 2012 or later.

3.1 Issues in Teaching and Learning Computer Graphics

We found that problems related to teaching and learning of computer graphics can be categorised into four key issues as summarised Table 1.

The first issue is insufficient background, especially inadequate skills in mathematics and programming (Cunningham, 2000; Du and Shu, 2011; Glvez et al., 2008; Han et al., 2008; He and Zhao, 2012; Hitchner and Sowizral, 2000; Hui et al., 2012; Papagiannakis et al., 2014; Santos, 2001; Schweitzer et al., 2011; Talton and Fitzpatrick, 2007). According to Elyan (2012), mathematical algorithms and procedures are important in computer graphics, especially when used to calculate transformations and projections. Programming skills are needed to implement, understand, and experiment with algorithms. Several researchers report that in their studies students were not sufficiently prepared (Haitao et al., 2012; Lowther and Shene, 2000; Papagiannakis et al., 2014).

The second issue is difficulties in understanding geometric concepts such as transformations, projections and 3D modelling (Du and Shu, 2011; Elyan, 2012; Santos, 2001; Schweitzer et al., 2011; Seron et al., 2008; Sung and Shirley, 2004). Sung and Shirley (2004) suggest that these issues arise because students have little visual experience and comprehension of geometric modelling.

The third issue is difficulties in solving logical problems and making the connection between theory, programming, application and final visual effects (Santos, 2001; Schweitzer et al., 2011; Talton and Fitzpatrick, 2007). Seron et al. (2008) observed that for topics such as global shading and inverse kinematics students struggled most with the technical complexity of the implementations.

The fourth issue is that many students are passive learners and don't interact much with peers and teachers (Gao and Zhang, 2014; Hui et al., 2012; Li, Huang and Gu, 2009). One suggestion addressing this issue is to use a top-down approach involving group projects. This resulted in increased attention to learning activities, more autonomous learning, and improved teamwork and communication skills.

Table 1: Learning Issues in Computer Graphics.

#	Issues	Solutions	Results
1	Insufficient knowledge of mathematics (Glvez et al., 2008; Hui et al., 2012; Zhou et al., 2010) and basic programming (Lowther et al., 2000; Papagiannakis et al., 2014)	Top-down: learn the foundation and the structure of computer graphics knowledge whilst practising the tools (Sung and Shirley, 2004). *Methods: Supervised lab activities.*	Students' mathematical concepts increased (Hitchner and Sowizral, 2000) and Students' programming skills improved (Elyan, 2012)
		Hybrid: Using individual programming assignments linked to a set of traditional lecture units with a balance of implementation, algorithms, and mathematical concepts (Lewis, 2012; Guo et al. 2010). *Methods: Programming assignments.*	
2	Difficulties in understanding transformations, projections and 3D geometric modelling (Elyan. 2012).	Top-down: Implement interactive 3D demos to show concepts and techniques in computer graphics (Kadam et al., 2013). *Methods: Specialised web-based 3D graphics learning tools for self-study.*	Improvement in geometrical structures understanding (Sueyasu et al., 2010).
3	Difficulties in solving logical problems (Hitchner and Sowizral, 2000; Talton and Fitzpatrick, 2007) and making the connection between theory, programming, application and the visual effects (Stephenson and Taube-Schock. 2009).	Top-down: Using a teaching platform with a set of compact applications to demonstrate computer graphics techniques and algorithms (Ganovelli and Corsini, 2009; Spalter and Tenneson, 2006). *Methods: Supervised lab activities, Project-based learning.*	Using modern high-level APIs improved students' mathematics, problem-solving and logical thinking skills (Hitchner and Sowizral, 2000).
		Hybrid: Set up a systematic practice-based learning process for students consisting of concept validation, project design, and project training (Schweitzer et al., 2011; Reina et al., 2014). *Methods: Specialised web-based 3D graphics learning tools for self-study.*	
4	Students have become passive learners and don't interact much with peers and teaching staff (Peternier et al., 2010, Marti et al. 2006).	Top-down: Pay attention to the relationship between theory and application, and use practical applications to promote communication between students and teacher (Li et al., 2009; Raikar et al., 2015). *Methods: Case-based Teaching, Specialised web-based 3D graphics learning tools for self-study, Project-based learning.*	Students' attention in their learning activities increased, and students became autonomous learners (Gousie, 2000). Teamwork and communication skills improved (Tori et al., 2006).
		Hybrid & Bottom-up: Motivating and guide students to be active learners. (Yang and Sanver, 2002; Taxén, 2004; Zhao et al., 2005). *Methods: Individual & pair class activities, Supervised lab activities, Specialised web-based 3D graphics learning tools for self-study.*	

3.2 Teaching Approach and Methodology

We identified three common approaches of teaching computer graphics. Of the 45 reviewed papers 34 described a top-down approach, seven a hybrid approach, and four a bottom-up approach.

The bottom-up approach is regarded as most traditional (Sung and Shirley, 2004) and seems to be favoured in text books for teaching computer graphics and is also most popular based on our personal teaching experience. The approach presents first foundations such as transformations and rendering of simple objects, before more complex topics are introduced (Taxén, 2004; Bouvier, 2002; Cunningham, 2000; Lowther and Shene, 2000; Shanshan et al., 2008; Sung and Shirley, 2004).

The top-down approach starts with a moderately complex problem or case study, e.g. a simple game, and then breaks it down into simpler problems (functional modules) (Shirley et al., 2015; Sung and Shirley, 2004, Talton and Fitzpatrick, 2007). This can help students absorbing the foundations and structure of graphics applications while practising (visible rather than mathematical based) application-level understanding and skills (Gousie, 2000; Santos, 2001; Seron et al., 2008; Song et al., 2009; Sueyasu et al., 2010; Sung and Shirley, 2004; Tori et al., 2006; Yang and Sanver, 2002). The top-down approach often involves using high-level tools and can promote self-learning and increase students' motivation and knowledge of fundamental computer graphics concepts (Nishino et al., 2011, Song et al., 2009, Sueyasu et al., 2010). Using high-level development tools, such as game engines, can enable students with insufficient mathematics skills to understand computer graphics concepts and produce attractive results (Elyan, 2012; Nishino et al., 2011; Raikar et al., 2015; Shanshan et al., 2008; Sueyasu et al., 2010; Tori et al., 2006).

Several researchers have combined the top-down and bottom-up approach into a hybrid approach (Glvez et al., 2008; He and Zhao, 2012; Hitchner and Sowizral, 2000; Hui et al., 2012; Schweitzer et al., 2011; Tori et al., 2006). The motivation is to support the learning of practical skills while simultaneously improving the knowledge base. Teachers can combine theory with graphics programming and use of graphics software to foster students' abilities in solving practical problems (Andújar and Vázquez, 2006; Reina et al., 2014; He and Zhao, 2012; Hui et al., 2012; Schweitzer et al., 2011; Tori et al., 2006).

4. Discussion

Our research identified four key issues encountered when teaching computer graphics. Three of the issues were related to insufficient skills and/or a difficulty of applying skills related to mathematics, programming, and spatial reasoning. The fourth key issue was that students have become passive learners. The results suggest that more care must be taken in defining appropriate prerequisites for entering a computer graphics course, or that more emphasis needs to be placed on teaching fundamental skills not necessarily directly related to computer graphics.

Our review suggests that a top-down or hybrid approach might be able to address some of these issues by demonstrating the importance and usefulness of fundamental concepts, applying them in a practical context, and giving students an increased motivation to study them (e.g. in order to create a game or other fun application). However, it is unclear whether such a top-down approach will result in the same breadth of knowledge of fundamental concepts as a bottom-up approach, and so far no quantitative comparison of these approaches exists.

Another possibility might be to exploit that students have different skills. We haven't found any research paper suggesting this approach, but we are aware of many tertiary training institutes, e.g. animation and design schools, combining students with an arts/design and programming background in the same class, but offering different assessment tasks to them.

It is somehow surprising that the vast majority of papers suggest using a top-down or hybrid approach, whereas the traditional bottom-up approach still seems to be most common in practice. One issue might be the fact that a top-down approach is often project-based, e.g. developing a simple game (Ganovelli and Corsini, 2009). This is likely to be attractive to students, but requires a large amount of supervision and might be difficult to implement for very large classes or instances where students' abilities differ substantially. None of the papers we reviewed was about teaching a large class (say, 200 or more students).

There seems to be a general consensus that increased interaction, e.g. via supervised labs, group work or peer work is useful. This corresponds to suggestions made in the computing and general education literature, e.g. the "blended learning" and "flipped classroom" concepts (Lage et al., 2000).

5. Conclusion and Future Works

Our research had identified four key issues making teaching and learning of computer graphics difficult. We discussed these issues and offered suggestions to address them. The most common solution offered in the literature is to employ a top-down or hybrid approach and promote more practical work and more interactions. While these suggestions are common in other educational fields as well, we believe they are particularly important in computer graphics due to the wide variety of skills involved and the fact that its output are images and models. Students can easily relate to visual output and often have an intuitive understanding of it, as compared to, say, the underlying mathematical and physical concepts such as transformation matrices and illumination equations. In future work, we would like to analyse tools and technologies for teaching computer graphics and investigate more formally what type of skills are predictors for success in computer graphics.

References

Andújar, C. & Vázquez, P. (2006). A programmable tutor for OpenGL transformations. *EG Education Papers.*

Angel, E., Cunningham, S., Shirley, P., & Sung, K. (2006). Teaching computer graphics without raster-level algorithms. *SIGCSE Bull, 38*(1), 267. doi:10.1145/1124706.1121423

Angel, E. & Shreiner, D. (2012). Introduction to modern OpenGL programming. *ACM SIGGRAPH 2012 Courses, 2*, 109. Los Angeles, California. doi:10.1145/2343483.2343485

Bouvier, D. J. (2002). From pixels to scene graphs in introductory computer graphics courses. *Computers & Graphics, 26*(4). doi:10.1016/S0097-8493(02)00109-7

Cunningham, S. (2000). Powers of 10: The case for changing the first course in computer graphics. *SIGCSE Bull, 32*(1). doi:10.1145/331795.331809

Cunningham, S., Hansmann, W., Laxer, C., & Shi, J. (2004). The beginning computer graphics course in computer science. *ACM SIGGRAPH Computer Graphics, 38*(4). doi:10.1145/1039140.1039154

Du, H. & Shu, L. (2011). Teaching computer graphics in digital game specialty. *Information Computing and Applications: Second International Conference*. Springer Berlin Heidelberg.

Elyan, E. (2012). Enhanced interactivity and engagement: Learning by doing to simplify mathematical concepts in computer graphics and animation. *Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON),* 1-8. doi:10.1109/EDUCON.2012.6201073

Ganovelli, F. & Corsini, M. (2009). ENVyMyCar: A multiplayer car racing game for teaching computer graphics, *28*(8). *Computer Graphics Forum*. doi:10.1111/j.1467-8659.2009.01425.x

Gao, S. & Zhang, C. (2014). Some reflections on undergraduate computer graphics teaching. *Frontier and Future Development of Information Technology in Medicine and Education*, 3175-3180. Springer Netherlands.

Glvez, A., Iglesias, A., & Corcuera, P. (2008). An introductory computer graphics course in the context of the European space of higher education: A curricular approach. *International Conference on Computational Science, 5102,* 715-724. doi:10.1007/978-3-540-69387-1_83

Gousie, M. B. (2000). Teaching computer graphics in a small department. *Proceedings of the Fifth Annual CCSC Northeastern Conference on the Journal of Computing in Small Colleges, 15,* 194-202. Consortium for Computing Sciences in Colleges.

Guo-wu, Y., Dan, X., & Yang, Z. (2010). Teaching reform of computer graphics experiments. *5th International Conference on Computer Science & Education,* 658-662. Hefei doi:10.1109/ICCSE.2010.5593525

Haitao, W., Hanrong, L., Jiajun, X., Yan, G., Jinsheng, H., & Caihua, W. (2012). Building computer graphics into an independent sub-discipline. *Knowledge Discovery and Data Mining*, 673-680. Springer Berlin Heidelberg. doi:10.1007/978-3-642-27708-5_93

Han, H., Zhang, C., & Feng, J. (2008). Research on several questions in CG case teaching system. *IEEE International Symposium on IT in Medicine and Education,* 193-197. doi:10.1109/ITME.2008.4743850

He, Y., & Zhao, Y. (2012). Reform and exploration of the computer graphics. *Services Computing Conference,* 403-405, IEEE Asia-Pacific. doi:10.1109/APSCC.2012.61

Hitchner, L. E., & Sowizral, H. A. (2000). Adapting computer graphics curricula to changes in graphics. *Computers & Graphics, 24*(2). doi:10.1016/S0097-8493(99)00162-4

Hui, P., Shanmei, L., Xiu, J., & Ruifang, Z (2012). Ability-oriented teaching reform for computer graphics. *7th International Conference on Computer Science & Education,* 1908-1911.

Kadam, K., Sahasrabudhe, S., Iyer, S., & Kamat, V. (2013). Integration of blender 3D in basic computer graphics course. *Proceedings of the 21st International Conference on Computers in Education (ICCE),* 483-486.

Kitchenham, B., Pretorius, R., Budgen, D., Pearl Brereton, O., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering - A tertiary study. *Information and Software Technology, 52*(8), 792-805. doi:10.1016/j.infsof.2010.03.006

Lewis, R. R. (2012). Coaster: Teaching computer graphics with a comprehensive project--work in progress. *Journal of Computing Sciences in Colleges, 28*(1).

Lage, M., Platt, G., & Treglia, M. (2000). Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment. *The Journal of Economic Education, 31*(1), 30-43. doi:10.2307/1183338

Li, L., Huang, Y., & Gu, Y. s. (2009). Research and discuss in teaching of computer graphics. *First International Workshop on Education Technology and Computer Science, 1*(3), 22-24.

Lowther, J. L., & Shene, C. (2000). Rendering + modeling + animation + postprocessing = computer graphics. *Journal of Computing Sciences in Colleges, 16*(1), 20-28.

Marti, E., Gil, D., & Julia, C. (2006). A PBL experience in the teaching of computer graphics. *Computer Graphics Forum, 1*(25). doi:10.1111/j.1467-8659.2006.00920.x

Nishino, H., Sueyasu, S., Kagawa, T., & Utsumiya, K. A CG learning tool with guided programming environment. (2011). *14th International Conference on Network-Based Information Systems, 1*, 423-432.

Papagiannakis, G., Papanikolaou, P., Greassidou, E., & Trahanias, P. (2014). glGA: An OpenGL geometric application framework for a modern, shader-based computer graphics curriculum. *Eurographics - Education Papers.* doi:10.2312/eged.20141026

Peternier, A., Thalmann, D., & Vexo, F. (2006). Mental vision: A computer graphics teaching platform. *International Conference on Technologies for E-Learning and Digital Entertainment, 3942,* 223-232.

Raikar, M. M., Giraddi, S. G., & Naragund, J. G. (2006). Open end activity (OEA): An experience in computer graphics laboratory. *Proceedings of the International Conference on Transformations in Engineering Education*, 405-413. doi:10.1007/978-81-322-1931-6_47

Reina, G., Müller, T., & Ertl, T. (2014). Incorporating modern OpenGL into computer graphics education. *IEEE Computer Graphics and Applications, 34*(4), 16-21. doi:10.1109/MCG.2014.69

Santos, M. P. D. (2001). Computer graphics in the scope of informatics engineering education. *Computers & Graphics, 25*(5). doi:10.1016/S0097-8493(01)00131-5

Schweitzer, D., Boleng, J., & Graham, P. (2010). Teaching introductory computer graphics with the processing language. *Journal of Computing Sciences in Colleges, 26*(2).

Schweitzer, D., Boleng, J., & Scharff, L. (2013). Interactive tools in the graphics classroom. *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, 113-117.

Seron, F. J., Cerezo, E., & Baldassarri, S. (2008). Computer graphics: Problem based learning and interactive embodied pedagogical agents. *Proc. Eurographics,* 173-180. doi:10.2312/eged.20081004

Shanshan, G., Caiming, Z., Rui, Z., & Jing, C. (2008). Design and implementation of case in CG case teaching. *IEEE International Symposium on IT in Medicine and Education,* 332-335.

Shesh, A. (2013). Toward a singleton undergraduate computer graphics course in small and medium-sized colleges. *ACM Transactions on Computing Education (TOCE), 13*.

Shirley, P., Ashikhmin, M., & Marschner, S. (2015). *Fundamentals of computer graphics*: CRC Press.

Song, M., Mokhov, S. A., & Grogono, P. (2009). Designing an interactive OpenGL slide-based presentation of the softbody simulation system for teaching and learning of computer graphics techniques. *Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering.*

Spalter, A. M., & Tenneson, D. K. (2006). The graphics teaching tool. *ACM SIGGRAPH 2006 Educators Program.* doi:10.1145/1179295.1179337

Stephenson, B., & Taube-Schock, C. (2009). QuickDraw: Bringing graphics into first year. *SIGCSE Bull., 1*(1).

Sueyasu, S., Nagano, T., Nishino, H., Kagawa, T., & Utsumiya, K. (2010). An interactive CG learning system through 3D authoring and programming. *Parallel, Grid, Cloud and Internet Computing (3PGCIC),* 115-122.

Sung, K. & Shirley, P. (2004). A top-down approach to teaching introductory computer graphics. *Computers and Graphics (Pergamon), 28*. doi:10.1016/j.cag.2004.03.005

Talton, J. O. & Fitzpatrick, D. (2007). Teaching graphics with the OpenGL shading language. *SIGCSE Bull., 39*(1). doi:10.1145/1227504.1227402

Taxén, G. (2004). Teaching computer graphics constructively. *Computers & Graphics, 28*.

Tian, F., Zhou, D., Wang, M., Zhou, S., & Gu, C. (2011). The application of thematic teaching method in CG teaching. *International Conference on Control, Automation and Systems Engineering (CASE),* 1-3.

Tori, R., Bernardes J. L., & Nakamura, R. (2006). Teaching introductory computer graphics using java 3D, games and customized software: A Brazilian experience. *ACM SIGGRAPH 2006 Educators Program, 12*.

Werner, M. (2013). Teaching graphics programming on mobile devices. *Journal of Computing Sciences in Colleges, 28*.

Yang, L. & Sanver, M. (2002). Web-based interactive 3D visualization for computer graphics education. *Proceedings of the First International Conference on Advances in Web-Based Learning,* 423-432.

Zhao, Y., Zhang, M., Wang, S., & Chen, Y. (2005). Exploring constructivist learning theory and course visualization on computer graphics. *Computational science and its applications (ICCSA): International conference, 4*, 1-9. Springer Berlin Heidelberg. doi:10.1007/11424925_1

Zhou, J., Ye, M., & Huang, C. L. (2010). Reform of computer graphics teaching method. *International Forum on Information Technology and Applications.* doi:10.1109/IFITA.2010.192