Pergamon

Comput. & Graphics, Vol. 19, No. 3, pp. 461–465, 1995

0097-8493/95 $9.50+0.00

0097-8493(95)00017-8

Education

COMPUTER GRAPHICS EDUCATION AT DELFT UNIVERSITY OF TECHNOLOGY

FREDERIK W. JANSEN and PETER R. VAN NIEUWENHUIZEN

Faculty of Technical Mathematics and Informatics, Julianalaan 132, 2628 BL Delft, The Netherlands

Abstract—Computer graphics nowadays covers a large number of quite different subjects and techniques. Moreover, these techniques are applied within a large variety of application domains. Not surprisingly, many of these subjects also appear prominently in courses taught by other disciplines. This has brought us, from a computer science point of view, to specialize on three sub-themes: rendering (3-D computer graphics), scientific visualization, and geometric modeling. These topics are taught in three advanced courses, where we try to keep a good balance in the attention for the basic physical phenomena (light reflection, digital sampling), their embedding in efficient geometric algorithms and data structures, and their implementation in graphics programs and systems.

1. INTRODUCTION

Computer graphics is a rapidly expanding field of science and engineering. But unfortunately, this rapid evolution has also led to a strong fragmentation of the field into a large number of sub-disciplines such as rendering, visualization, animation, interactive modeling, user interfaces, multimedia, virtual reality, not to mention the overlap with strongly related disciplines such as image processing and telecommunication. Many of these subjects in their turn integrate a large variety of disciplines such as physics, psychology, mathematics, electrical engineering, *etc.* To make matters worse, these subjects are all applied within a large variety of application domains. The massive weight of new textbooks on computer graphics are a daily proof of this ongoing process.

In fact, computer graphics is nowadays considered to be so crucial for many disciplines and so attached to computer applications in general, that computer graphics techniques are taught in a large number of courses by a broad spectrum of disciplines. Within Delft University of Technology, for instance, the Applied Physics department offers courses on image processing, pattern recognition and computer vision, the Electrical Engineering department on image analysis and compression, the Surveying department on GIS, the Architectural, Civil Engineering, Mechanical Engineering, and Industrial Design departments on 3-D CAD with emphasis on modeling and graphics. In particular, the Industrial Design department is quite active in offering a large variety of courses ranging from realistic visualization, sketch input, fast proto-typing, virtual reality, perceptual issues, user interface design to information ergonomics.

Even within the computer science department, our Computer Graphics and CAD/CAM group is not the only section doing research and offering courses on computer graphics. The Information Systems section, for instance, offers courses on user interfaces and highly interactive systems (including virtual reality, hyper media, *etc.)*, the Knowledge-based Systems section is active on (multi-media) sensor interpretation, and the integration of images and text may soon attract the interest of the Data Bases and Operating/Network Systems sections.

Given this fanning out of media technologies—more than ever "media is the message"—the question for our section a few years ago was how to position our courses within this spectrum in a way that would be both of interest for computer science students as well as for students of other faculties. Given our own research interests, we decided to focus the courses on 3-D graphics and modeling and scientific visualization, subjects that have also direct relevance for other engineering disciplines. Even then, the variety of topics was so large that it could not easily be packed into one course or a series of courses with gradually more advanced topics, as we did before. This has led us to develop a course program with one introductory course (fourth semester) and three advanced courses (fifth and sixth semester), one on 3-D computer graphics (rendering), one on scientific visualization, and one on geometric modeling. The advanced courses include 14–21 h of lecturing and 40 h of programming exercises/assignments and can be taken completely independently of each other.

The advantage of having the advanced courses in line with our main research themes is that we did not have to acquire any additional expertise, and the courses can be kept up-to-date with relatively low effort. PhD students working on our research projects can take part in the lecturing, can provide visual examples and case studies, can help in supervising the programming exercises, and can provide new tools. An additional positive side effect is that students get in touch with our research topics and may eventually choose a master thesis subject on one of these subjects.

While the lectures are mainly intended to present the methods and techniques together with the necessary background material, the aim of the program assignments is to train students in graphics programming. This is in line with the idea that most of the graduates will come to work in CAD/CAM and graphics system programming. A problem with the programming assignments is that these exercises are not very accessible for students from other departments. They can enter the courses, but lack the programming background (C, UNIX, X-windows) that is provided in the first semesters of the computer science curriculum. This means that we will have to give them extra help and support. Still, we think it is worthwhile for them as well to have some hands-on experience with programming. Overall, the course program is quite successful and the courses are highly appreciated by the students.

In the following we will give an overview of the contents of the courses and the programming assignments, and discuss some related aspects (mainly with respect to the 3-D Computer Graphics course) as course format, lecturing style, course materials, sample programs. We end with a short discussion.

2. THE COURSES

The course program consists of an introductory course, which is compulsory for students in the second year of the computer science curriculum, and three advanced courses, which can be selected by students for their third year program. Students who specialize in computer graphics will generally take all three, students with other specializations may take only one or two. The section Computer Graphics and CAD/CAM also teaches two advanced CAD/CAM courses, Product Modeling and Industrial Automation, which are not discussed here.

a415 Introduction Computer Graphics and CAD/CAM

The computer graphics part of this introductory course is based on the book of Hearn and Baker and covers graphics hardware, 2-D graphics algorithms and simple 3-D display algorithms. The complete course consists of 14 h of lectures and 20 h of practical exercises. The course is given in the fourth semester. After the fourth semester, the computer science students have to choose a direction of specialization. The introductory course therefore is important as a first acquaintance with the subject. We assume that students from other departments will have followed a similar type of introductory course at their own faculty (which is however not always the case).

The advanced courses are:

a471 Geometric Modeling

This course deals with input, representation and display of 3-D surface and solid models. It covers the following subjects:

—curve representations
—parametric representations
—Bezier, B-spline
—surface representations
—solid modeling
—boundary representations
—CSG
—conversions
—input representations
—sweeping
—cross-sectional design
—constraints
—display of models
—direct display of CSG models
—surface scanning algorithms
—kinematic analysis and interference detection
—finite element mesh generation.

Practical assignments include the implementation of an isoparametric line drawing, a subdivision rendering algorithm for bicubic patches, and a 2-D boundary evaluation algorithm for polygons.

a476 3-D Computer Graphics

This course deals with the most important techniques for visualizing 3-D objects and scenes. The emphasis is on realism.

—3-D object modeling
—viewing transformations
—visible-surface algorithms
—color models
—local reflection models
—shadow, texture
—ray tracing
—ray intersection algorithms
—spatial subdivision
—adaptive stochastic ray tracing
—radiosity
—two-pass algorithms
—texture mapping
—procedural models
—animation
—display architectures
—parallel rendering.

The practical assignments include three programming exercises: the implementation of the viewing pipe-line in a wire-frame program, Gouraud shading in a scanline algorithm, and stochastic shadow ray tracing for area light sources in a ray tracing program.

a477 Scientific Visualization

This course deals with methods and techniques for visualizing multi-dimensional scalar, vector and tensor fields. The emphasis is on flow visualization.

—introduction 3-D graphics
—data visualization process
—perceptual issues, color
—volume data
—surface extraction
—volume rendering
—hybrid rendering

—vector fields
—vector field visualizations
—flow visualization
—particle tracing
—stream ribbons and surfaces
—topological analysis
—case studies.

The practical assignments are done with AVS. After some small introductory exercises, a vector field visualization and two volume visualizations have to be made.

3. THE FORMAT OF THE COURSES

Each advanced course encompasses 2 or 3 h of lecturing (45 min each) during 7 weeks. The lectures are given in a reasonable tempo, without spending too much time on details. The students are assumed to spend another 3–4 h on self-study for each lecturing hour. The self-study is done with the recommended books or with the supplied course material. The lectures are given using overhead sheets that correspond with the provided course material. For the 3-D computer graphics course, copies of the overhead sheets are made available.

Twice a year for each course a written examination is held. This examination takes 3 h and consists of six open questions (with 3 or 4 sub-questions) that require elaborate answers (1 page for each question).

The programming exercises are done in groups of two students on UNIX workstations. There is a comprehensive written instruction available for each exercise. The exercises are very much pre-programmed. The students have to fill in the parts that are left open for the assignment. The exercises are, therefore, quite structured and there are no "open" assignments. The semester structure of our curriculum unfortunately prohibits small projects or to use a more compact and intensive course format. As students have to do several courses simultaneously, it is difficult to do projects with small groups. Also students are too easily carried away with open-ended projects. See also the discussion in ref. [1].

Now and then we are able to experiment with a more compact format when one of our courses is given at another university. Then the course is normally packed in a period of 2 weeks; for instance, five mornings with lectures and five afternoons with preprogrammed exercises during the first week. Then there is a week left for a small project. For the 3-D Computer Graphics course, for instance, a nice project that we recently did is to extend the public domain ray tracer Rayshade [2] with indirect diffuse reflection (either by a radiosity pre-processing, path tracing, particle shooting, or any other technique that improves on ambient shading). This kind of project is, of course, very stimulating.

4. COURSE MATERIALS

For the two advanced courses Geometric Modeling and Scientific Visualization, material is made available in the format of tutorial notes, supplemented with a set of scientific papers. The advantage of this type of notes is that they are cheap for the student ($5) and well tuned to the course contents. For the staff it is initially some effort to create the notes, but updates and modifications are later easily made. Disadvantages are the low quality of color reproductions (although this is quickly improving) and the lack of uniformity in the added papers. Papers are not easy reading, however, we think it is worth the effort as students may not encounter a scientific paper otherwise.

For the 3-D Computer Graphics course the book of Watt [3] is used. The advantages are that the book matches quite well with what the course tries to convey (with the exception of a large and detailed chapter on surface modeling that is covered in our Geometric Modeling course) and that the English is very readable for Dutch students. Drawbacks are the weak first chapters and the weak overall structure of the book. Simple raster operations appear only in the middle of the book, anti-aliasing is tucked away in a separate chapter, *etc.* (see the course contents above for our preferred order).

For the introductory course, the first edition of Hearn and Baker [4] has been used so far. This served our purpose well. It is good to read and the concept and structure are clear. Unfortunately, the new edition [5], although improved on details, is a much more elaborated text book with new chapters added on ray tracing, radiosity, texture mapping, *etc.* The size has almost doubled from 350 to 650 pp. and so has the price. The book contains more material than needed for the introductory course. Worse, for the advanced courses, the material is too superficial and the students will have to buy other books or notes that will largely overlap with the book they already have.

In fact, for the introductory course, the students have to buy two books, one for the Computer Graphics and one for the CAD/CAM part of the course. This means *ca* $100 for a 2 h/semester course, which is far too much for the second year. What we really need are two simple introductory texts for Computer Graphics and CAD/CAM for a total price below $50. Given the fact that these are not available, the choice is to either make a simple text for CG&CAD/CAM ourselves and to use books for the advanced courses, or to use books for the introductory course and use notes for the advanced courses (the latter will probably be the case).

5. SAMPLE PROGRAMS

As one of the aims of our courses is to make the student familiar with graphics programming, it is important to have sample programs to work with. For the 3-D Computer Graphics course, we use three simple programs, a wire frame, scan-line and ray tracing program. For the exercises the students have to add a given functionality to the program. By doing this they learn the structure and mechanics of the whole program.

In the early eighties the first author used the wire-frame program by Crow [6]. It is a simple and very elegant hidden-line program based on the depth-priority algorithm and the Sutherland–Hodgman clipping algorithm. As an exercise, Gouraud shading was added, as well as the depth-buffer algorithm. Other exercises were to implement the setting of the viewing transformation matrix, object rotations (*e.g.*, yaw, pitch and roll), and local light sources. Watt [3] also provides a simple wire frame modeler with his book. Currently we are using a wire frame modeler embedded in an X-windows environment. This makes the program structure more complex and difficult to understand, but the students get some acquaintance with X-windows.

For the shading exercise we use a simple version of the scan-line program PLAMO [7].

For the third exercise we used for some time a simple version of our own ray tracing program RAYMO, developed by van Wijk [8]. Exercises were to add shadows, reflections, and more complex primitive intersections. Now we are using the program Rayshade. Typical exercises are the calculation of refracted rays and jittered shadow sampling of area light sources (the existing functionality is first removed).

As mentioned before, Rayshade provides a valuable starting point for small projects and more research oriented assignments. It is built well in an object oriented way and understanding the structure is worth the effort. It is also fast. We have used it extensively for (master thesis) projects on spatial subdivision structures and ray traversal methods.

6. DISCUSSION

Teaching of computer graphics involves three aspects. First of all there are the concepts of the physical phenomena underlying the graphics (*e.g.*, global and local light reflection for rendering, computational fluid dynamics for flow visualization, perceptual issues, *etc.*). Very important for graphics also is the theory of sampling and filtering, issues closely related to signal and image processing. Once these issues have been sorted out, it comes to the efficient embedding in (geometric) algorithms and data structures, and their efficient implementation in graphics programs.

However, we do not consider the physics (of light and color) as such to be that fundamental for computer graphics, as for instance in refs [9–11]. Nor do we feel it necessary to reverse the order of 2-D and 3-D subjects during the course. In the advanced courses we go through the 2-D basics very quickly, and we are then left with plenty of time for the more interesting 3-D material. We have never spent much time on graphics standards and libraries. The use of graphics subroutine packages is more or less integrated in the programming exercises.

Rather we see the development of efficient geometric algorithms and data structures as the more fundamental theoretical contribution in teaching computer graphics. In that sense we are inclined to include more material from the field of computational geometry in our introductory course. This would make the introductory course not only a good starting point for computer graphics, but also a valuable base for a broader range of technical applications, such as product modeling, robot programming, numerical control, geographic databases, and other forms of spatial reasoning.

As mentioned before, our exercises and programming assignments are very much precanned. We do not give open-ended assignments. In fact, there is no time available in the curriculum, nor do we have the ability to supervise large numbers of students. Nevertheless, we are well aware that this type of project can be very stimulating and a good learning experience. Currently we are, together with several other Dutch groups on image processing and advanced computer architectures, in the process of establishing a graduate research school for Advanced Systems in Computing and Imaging (ASCI). This school will bring together research on image processing, computer vision and computer graphics with research on advanced (parallel) computing (systems). Computer graphics will be one of the subjects for the course program of this school. This will open for us new opportunities for advanced courses that may involve more research-directed projects. A compact course format in combination with open-ended assignments seems to us the preferred format for this type of education.

7. CONCLUSIONS

During the last three years we have developed a new course program for computer graphics at Delft University of Technology. We have chosen three advanced courses on quite different topics, in addition to a more general and basic introductory course. We try to keep a good balance between the basic (physical) principles, their embedding in geometric algorithms and data structures, and their implementation in graphics programs. As an engineering school we emphasize the experimental and engineering attitudes (*cf.* [12]).

Although graphics textbooks already fill a large shelf, we are still looking for a simple introductory text for computer graphics and CAD/CAM; a textbook on scientific visualization would be welcome as well.

Acknowledgements—The course program was developed by the staff of the Computer Graphics and CAD/CAM section, which includes Wim Bronsvoort (a471 Geometric Modeling), Erik Jansen (a476 3-D Computer Graphics), Frits Post (a477 Scientific Visualization) and Peter van Nieuwenhuizen and Matthijs Sepers (a415 Introduction Computer Graphics and CAD/CAM) and several PhD students.

REFERENCES

1. D. Schweitzer and L. Northrop, Getting to the "graphics" in a graphics exercise. *Computer Graphics* **25**, 151–157 (1991).

2. C. E. Kolb, *Rayshade User Guide and Reference Manual* (1992) (ftp://princeton.edu/pub/Graphics/rayshade.4.0).
3. A. H. Watt, *Three-Dimensional Computer Graphics*, 2nd edn. Addison Wesley (1993).
4. D. Hearn and M. P. Baker, *Computer Graphics*, Prentice Hall, NJ (1986).
5. D. Hearn and M. P. Baker, *Computer Graphics*, 2nd edn. Prentice Hall, NJ (1994).
6. F. C. Crow, Three-dimensional computer graphics, parts 1 & 2. *Byte Mar* (Apr., 1981).
7. W. F. Bronsvoort, F. Klok, and F. H. Post, Plamo: an interaction-oriented solid modelling system with shaded images, Report 85-25, Faculty of Technical Mathematics and Informatics, Delft University of Technology (1985).
8. J. J. van Wijk, On new types of solid models and their visualization with ray tracing. PhD thesis, Delft University Press (1986).
9. G. S. Owen, M. M. Larrondo-Petrie, and C. Laxer, Computer graphics curriculum: time for a change? *Computer Graphics* **28**, 183–185 (1994).
10. Z. Xiang, A nontraditional computer graphics course for computer science students. *Computer Graphics* **28**, 186–188 (1994).
11. G. S. Owen, Teaching image synthesis as a physical science. *Computers & Graphics* **18**, 305–308 (1994).
12. G. S. Owen, Teaching computer graphics as an experimental science. *Eurographics Workshop on Graphics and Visualization Education* (1994).