Zeitschrift: Schweizer Ingenieur und Architekt
Herausgeber: Verlags-AG der akademischen technischen Vereine
Band: 104 (1986)
Heft: 7

Artikel: Computer Graphics Simulation in the 1990s
Autor: Greenberg, Donald P.
DOI: https://doi.org/10.5169/seals-76073

"Performing Arts Center" by *Phil Brock*, Program of Computer Graphics, Cornell University.
This high-resolution, ray-traced image of the new Performing Arts Center at Cornell University, designed by Stirling-Wilford Associates, but not yet constructed, was created using Cornell's testbed modeling system for realistic image synthesis. The light reflections, transparency, shade and shadows, and the contextual information illustrate how these simulation techniques can be used for architectural design evaluation.

«Performing Arts Center», Bild von Phil Brock (Program of Computer Graphics, Cornell University).

Dieses neue Zentrum der darstellenden Künste an der Cornell University ist von Stirling-Wilford Associates projektiert; es ist aber noch nicht gebaut. Das Bild wurde mit hoher Auflösung und mit Hilfe der Strahlen-Summentechnik für wirklichkeitsgetreue Bildsynthese auf der Versuchsanlage für Modeling an der Cornell University hergestellt

Computer Graphics Simulation in the 1990s

By Donald P. Greenberg, Ithaca, NY

Computer graphics simulation is a most powerful tool which can create realistic color photos of planned buildings before they exist. The samples shown here were created on an experimental system. Since October 1985 the Program of Computer Graphics at Cornell University, Ithaca N.Y., USA, is developing a graphics environment for a supercomputer, one of the four supercomputer centers recently established under a substantial research effort initiated by the US Government being located at Cornell University.

Observer position and illumination are chosen as desired, and each element of the image can be varied in shape, surface texture and color. The images created are helpful to the architect in designing, and they provide building committees and decision makers with realistic views, something which traditional plans and models can not convey.

The enormous increase in computer power offered by supercomputers will move simulation closer to the further goal of visualizing real time walks through an imagined building. *(ed.)*

Grafische Simulation mit Hilfe des Computers kann realistische Farbbilder geplanter Bauten liefern, bevor sie existieren. Die hier gezeigten Beispiele sind experimentell hergestellt. Dem Program of Computer Graphics an der Cornell University, Ithaca, N.Y., USA, steht seit Oktober 1985 ein Supercomputer zur Verfügung, der im Rahmen eines grossen Forschungsprogramms der amerikanischen Regierung zur Entwicklung und Anwendung von Supercomputern gebaut und in Betrieb genommen worden ist; eines der vier Supercomputer-Zentren wurde der Cornell University zugesprochen.
Der Beobachterstandort und die Beleuchtung sind frei wählbar. Jedes Bildelement kann in Form, Oberflächenstruktur und Farbe variiert werden. Wirksame Hilfe bietet dies nicht nur dem planenden Architekten, sondern vor allem auch den Bauherren und Baukommissionen, die eine wirklichkeitsgetreuere Vorstellung erhalten, wie sie herkömmliche Pläne und Modelle nicht vermitteln können. Die unerhörte Steigerung der Rechnerkapazitäten lässt das weitere Ziel eines Echtzeit-Rundganges in einem erdachten Gebäude näher rücken. (Red.)

Introduction

In this brief paper, I will try and first present an overview of computer graphics simulation in the United States today, including personal comments on existing hardware and software systems. I will then try and document a very brief history related to methods for graphical communication. After a description of the relationship between complexity and computer power, I will then describe some future hardware environments. Following this, one can characterize the types of problems that we are attempting to solve.

Since one can not verbalize computer graphics I use pictorial format to describe the current research effort at Cornell University and how we are approaching simulation in terms of realism, analysis and design. Although many of these techniques are experimental and can be conducted only in universities, they certainly will indicate what will be available in the early 1990s. Clearly, progress is not made without its attendant problems, and in concluding, I hope to present some of my opinions. Please note that my viewpoints are biased, and that I am only very familiar with some of the computer-aided design systems related to structural engineering and architecture. Thus, although I would like to present a broader overview of the present and the future simulation environments, this brief address is somewhat restricted to the fields with which I am most comfortable.

I also would like to add that I have been extremely fortunate in having the luxury of teaching at an excellent research university and of directing a very sophisticated computer graphics research laboratory. Thus, I have been able to experiment with many new techniques. Let me emphasize that these approaches are *experimental* and not yet cost effective. However, they are indicative of what will be available in the future.

Current Hardware and Software Technology

Let us look at where the computer hardware and software technology is today. In the United States, most engineering corporations, particularly in the aerospace, automotive and industrial sectors, are moving towards a distributed workstation environment, while still maintaining some large-scale centralized computing capability. A significant percentage of the large architectural firms are also moving to this workstation type of environment.

The hardware technology is now available and sufficiently economic to move in this direction. Workstations based on 32-bit processors, such as National Semi-conductor's 32032, Motorola's 68020, or DEC's Microvax II, each with 32-bit addressing capability and a floating point accelerator with 64-bit precision, are now common. Operating systems with virtual memory capability, window managers, and multi-tasking capabilities are becoming standard. Each workstation has its own local disk and perhaps up to eight or sixteen megabytes of fast memory, a tablet or mouse for interaction, and usually a high-resolution color display. In a minimal configuration, these devices are currently being sold in the United States for approximately $ 25 000 or less per station.

Although the price may seem high at first glance, let us look at the reasons and justification for moving in this direction. First of all, studies indicate that engineers really spend only twenty percent of their time doing engineering work, and eighty percent of the time in planning, administration, preparing documents, and perhaps worst of all, communication. Even when engineering work is being performed, such as a finite element analysis, studies indicate that eighty-five percent of the time is spent in the input tasks, such as defining the mesh necessary for the analysis, or in trying to interpret the results. In the architecture profession, the situation might be even worse. Probably less than twenty percent of an architect's time is spent designing, and the dominant portion of a person's workload is dealing with zoning, building codes, communications with owners, contractors, sub-contractors, and of course, the inevitable and excessive amount of drafting time.

What the workstation manufacturers are currently attempting to sell first is a product, not to improve the engineering tasks, but to improve the efficiency of the environment. Thus, many of the workstations support office automation which includes voice, data, and video transfer. I contend that the real purpose should be to improve the design methodology and the engineering process. Unfortunately, this is difficult to attain, because the software is frequently limited to restricted problems, is not sufficiently flexible or portable to other machines, can not be easily used with different systems since it does not interact with a common database, and last of all, is very difficult to use. Clearly, for those of you who have heard one of my presentations in the past, I believe that one of the major reasons for this is the poor communication channel between the user and the machine. *I believe that the language of communication which must be used is graphical* and that a picture is worth 1024 words. Let me briefly show how far software has progressed in computer-aided design and computer graphics.

Limitations of Historical Graphics Simulations

To create a synthetic scene, it is necessary to perform the following five steps:

1. *Three-dimensional model*
The entire geometry of the environment must be mathematically defined as well as the color of the surfaces.

2. *Perspective transformation*
Each vertex of the model is mathematically transformed to generate a true perspective picture on the image plane, as well as retaining the correct perspective depth information.

3. *Visible surface determination*
Surfaces remaining within the frustum of vision after the perspective transformation are sorted in depth so that only the elements closest to the observer are displayed.

4. *Light reflection model*
The color and spatial distribution of the light reflected from each surface in the environment must be simulated.

5. *Image display*
The final image is rendered by selecting the appropriate red, green, and blue intensities for each pixel in the visible scene.

Historically, because of the constraints of processing power, the unavailability of storage, and the high cost of memory, many shortcuts were taken. Simple environments were used, limited light reflection models were incorporated, and the separate steps of the image creation process were combined.

Today, much of the processing has now been imbedded in hardware. Transformations which use a 4×4 matrix multiplication and perform all of the perspective transformations, clipping, viewport mapping, and windowing are accomplished in hardware. Visible surface algorithms, and simplified shading models are also built in hardware. Furthermore, some parts of the process have been pipelined in more expensive display devices to provide increased speed in image generation.

Figures 1a, 1b, 1c. Finite Element Analysis by *T. Y. Han, Bruce Bailey, Jerry Hajjar,* Program of Computer Graphics, Cornell University.
These figures illustrate the modeling, pre-processing and post-processing stages of a finite element analysis of a screw propeller. The solid cone and blades were defined using parametric surfaces and lofting techniques (Fig. 1a). The visible surface image with diffuse shading reveals the solid nature of the object (Fig. 1b). Stress contours are shown on the three-dimensional surface without losing the perception of three dimensions (Fig. 1c).

Bilder 1a bis 1c. Finite Elememte-Analyse von T. Y. Han, Bruce Bailey und Jerry Hajjar (Program of Computer Graphics, Cornell University).
Am Beispiel eines Schraubenpropellers zeigen die Bilder die Phasen der Modellierung sowie der Vorbereitung und der Auswertung der Finite Elemente-Analyse. Die Körper des Nabenkonus und der Propellerblätter wurden (Bild 1a) mittels parametrischer Oberflächen und Anordnungstechniken dargestellt. Das Bild 1b vermittelt mit den sichtbaren Oberflächen und diffuser Schattierung einen plastischen Eindruck des Objektes. Im Bild 1c sind die Belastungsbereiche auf den dreidimensionalen Flächen dargestellt, wobei der räumliche Eindruck nicht verlorengeht

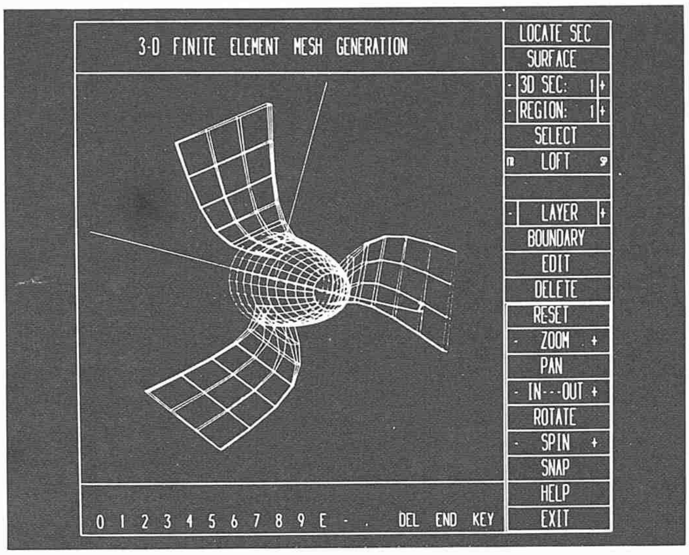


Figure 1a

However, there are still several fundamental flaws in most algorithms or systems which are used for image synthesis. These deficiencies include: poor light reflection models, incorrect description of material and surface properties, speed—particularly for dynamic sequences, and limitations on complexity.

Perhaps the most glaring weakness is the lack of global illumination effects. It is well known that the effect of intra-environment reflections can substantially enhance the quality of the image. Two new techniques which overcome this limitation will be described later.

Complexity and Implied Computational Requirements

The types of problems we must be able to handle are far more complex than those we have attempted to solve. The geometric modeling systems of today and the software currently being offered by turnkey vendors will not suffice for our future requirements. In order to solve real engineering problems, the scale of the problem definition must be changed. The physical world is not composed of simple two-dimensional parts nor are complex geometries built up from sticks and spheres. All objects are three-dimensional and most are very complex. In the future, we will not be satisfied with just analyzing a part, but we must understand the behavior of a set of parts, then sub-assemblages, and finally the behavior of entire systems.

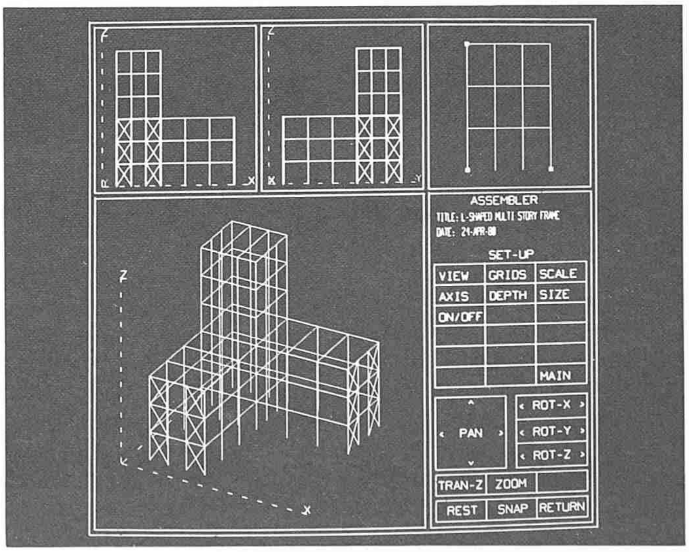
The scale of the problems we are investigating will inevitably be increased.

Let us look at the implications that the increase in problem complexity has on computational requirements in terms of engineering simulation as well as computer graphics. In most engineering problems, the simulation of physical behavior depends on solving a set of simultaneous equations. Obviously, as the scale of the problem gets larger, and the number of unknowns increases, the solution time increases. What is not so evident is how drastically the necessary computation time increases. Due to the three-dimensionality of typical problems, the number of unknowns increases as a cubic function. However, the solution time for a system of simultaneous equations is proportional to the cube of the number of unknowns. Thus, if the scale of the problem changes by a factor of ten, a billion (10^9) times more processing power is required.

Graphics problems are easier, but by no means are the solutions simple. Major computational tasks in trying to generate images of a simulated environment, fall into three categories: perspective transformations, visible surface computations and the intensity calculations. The time required for the perspective transformations varies linearly with the number of elements. The visible surface problem is basically a sorting problem, and varies approximately with the square of the number of elements. The intensity calculations are

Figures 2a, 2b, 2c by *Marcello Gattass* and *Carlos Pesquera,* Program of Computer Graphics, Cornell University, and 3D Eye, Inc., Ithaca, New York.
These figures illustrate how a three-dimensional frame can easily be interactively modeled and analyzed. Information necessary for analysis consists of the frame geometry, member sizes, support conditions and loads, all of which can be input graphically. Figures 2a and 2b show vector displays of a three-dimensional frame which has been subjected to dynamic loading. Figure 2c illustrates results of a commercial frame design software package, STEEL 3D, where color is used to depict the stress levels in each member.

Bilder 2a bis 2c. Räumliche Tragkonstruktion von Marcello Gattass und Carlos Pesqueras (Program of Computer Graphics, Cornell University; 3D/Eye Inc., Ithaca N. Y.).
Ein dreidimensionales Tragwerk kann ohne Mühe interaktiv aufgebaut und analysiert werden. Die für die Auswertung benötigten Angaben umfassen die Geometrie des Tragwerks, die Abmessungen der Elemente, die Lagerungsbedingungen und die Lasten; alle diese Werte können grafisch eingegeben werden. Die Bilder 2a und 2b zeigen Vektordarstellungen einer räumlichen Tragkonstruktion, die dynamischer Beanspruchung ausgesetzt ist. Das Bild 2c zeigt Anwendungsresultate eines kommerziellen Softwarepaketes, Steel 3D, bei welchem die Beanspruchung in jedem Element nach einer Farbskala dargestellt wird

Figure 2a


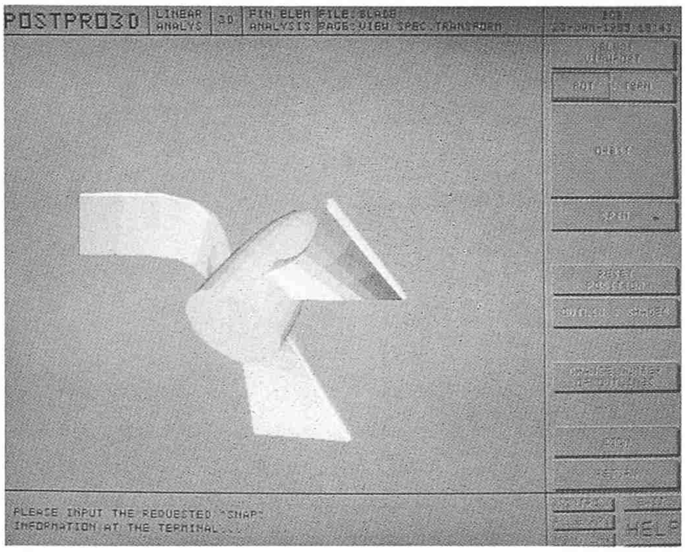


Figure 1b

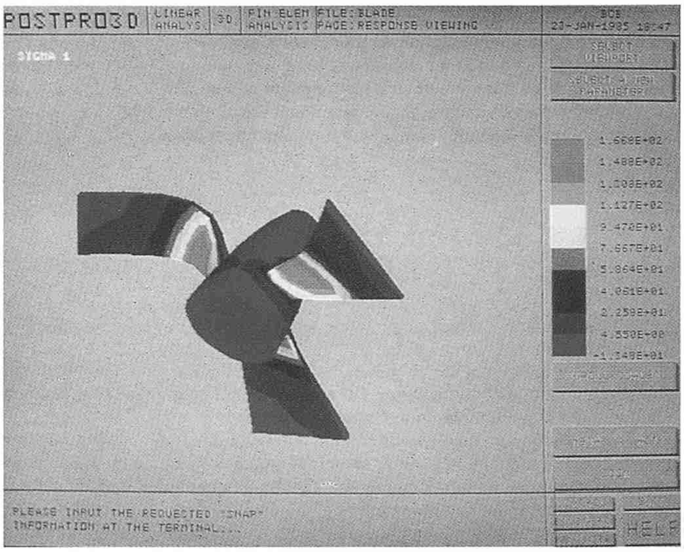


Figure 1c

basically a function of the complexity of the light reflection model, and although these models will have to become more sophisticated to model reality, the computational requirements are not exponential.

One of the major objectives will be real time image generation. The perception of motion has a great effect on our understanding of complex three-dimensional environments, and thus we will want to be able to generate images thirty to sixty times per second. Let me emphasize that these are images of models which are being used for design and engineering analysis, and the data might be continually changing. Therefore, I am not referring to the technology used for flight simulators in pilot training, where one is dynamically viewing a static environment. In order to perform these new simulation requirements, it is probable that we will need a million times more processing power for the graphics capabilities only. Note also that despite this large increase in requirements, it is still three orders of magnitude less than that needed for solving the engineering problems themselves.

Future Hardware Environments

It is clear from the previous discussion that several developments must follow. Future hardware environments will consist of networks of engineering workstations with very large computational nodes. The engineering/drafting table of the 1990s may still be a desktop, but the flat surface on which one works is liable to be the face of a computer display. The power which resides in the drawer will be vastly increased and many parallel processors will be used to provide the interactive environment. Specialized hardware will be incorporated to streamline the graphics operations and provide real time capabilities, and very high bandwidth networks in the gigabyte range will be available to transmit data to and from supercomputing nodes. This may all sound futuristic and much like a dreamworld, but let me try and briefly describe some of the efforts which are currently going on in the United States today, and what we are specifically doing at Cornell. The research is all experimental but is very exciting and becomes our first fundamental efforts to migrate to the computing environments necessary in the 1990s.

United States Supercomputer Research Effort and work at Cornell University

Recently, the United States Government initiated a substantial research effort both to advance the state-of-the-art in supercomputer production and in supercomputing, and to provide machine cycles for scientific exploration. After a year of evaluation of many proposals, four national centers were selected. Two of these are at universities, one at the University of Illinois and one at Cornell University.

Figure 2b

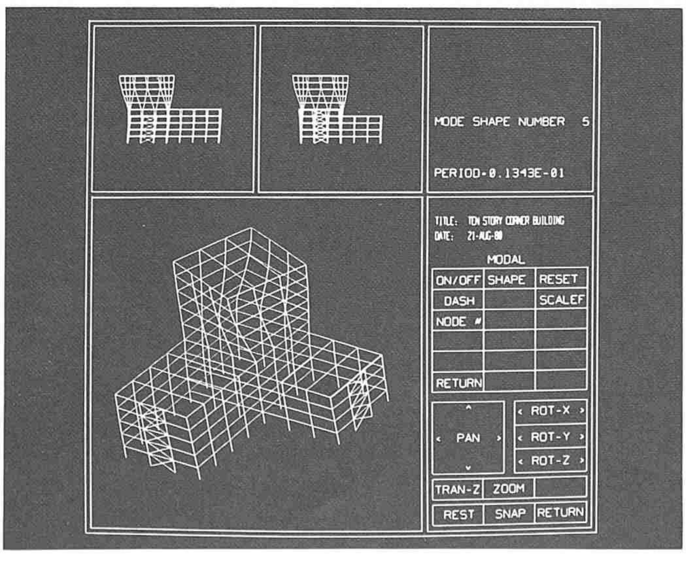
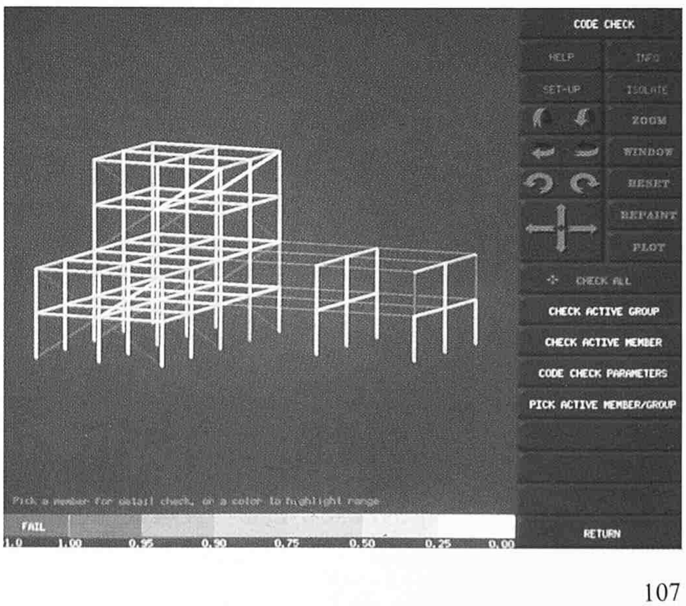


Figure 2c



Figures 3a, 3b, 3c. "Design Studies of a Conference Room" by *Alan Polinsky*, Program of Computer Graphics, Cornell University.
The power of the simulation techniques for interior design is illustrated by this sequence of three photos. The conference room was modeled and displayed using the ray-tracing technique. A new texture of the conference table is then substituted and the color motif changed in the subsequent images.

Bilder 3a bis 3c. Entwurfsstudien für ein Sitzungszimmer von Alan Polinsky (Program of Computer Graphics, Cornell University).
Die drei Aufnahmen veranschaulichen die Ausdruckskraft der Simulationstechniken. Das Sitzungszimmer wurde mittels Strahlen-Summentechnik dargestellt (Bild 3a). Dann wurde das Material des Tisches ausgewechselt und die Farbgestaltung verändert (Bilder 3b und 3c)

Figure 3a

This effort is just beginning and our computer center opened in mid-October 1985. Our mission at Cornell is threefold. The first is to immediately provide computing cycles to the scientific community in general. The second is to develop techniques for increasing processing power using parallel machines (the University of Illinois is researching methods to increase processing power using larger computers). The third objective is to improve the interactive graphics environment specifically for scientific investigation. In particular, my laboratory is responsible for the development of the advanced hardware/software/graphics subsystem which will surround the supercomputer.

The types of problems that we will be investigating on the supercomputer can be characterized by an increase in complexity, non-linear behavior, and time-dependency. Examples of these problems might be as follows: The airflow over the wing of an airplane with its turbulence and vortex shedding. The same solution to the Navier-Stokes equation would be useful to show blood flowing through the heart and arteries or fluid flow in hydraulics problems.

The membrane behavior and contraction of a neuron as the membrane receptor triggers an electrical response.

The synthesis of a protein molecule, built up according to the minimization of energy.

Using finite element methods, the transient flow or creep of material in a mold injection system or the crack propagation initiating with a brittle fracture can be predicted.

Other difficult but important engineering problems include the soil structure interaction or the vibration of a building subjected to earthquake and dynamic loading.

In the architecture profession, it might be the thermal response of a building to solar heat gain or loss, including the simulation of the heating, ventilating, and air-conditioning systems.

Lastly, it might be the aesthetic design evaluation of walking through a building, still in the design stage, with all the visual impressions of light, reflections, texture, and space.

These are the types of problems which we have not dared attack yet, but we wish to be able to solve. In essence, we intend to allow the scientist, engineer, architect or designer to *monitor the simulation as it is occurring*, to insert human creativity and expertise into the process, and thus be able to modify the experiment or the design.

Obviously I can not show you the results in 1985, but I can show some examples of the types of problems we are working on and what we hope to do. Since the Program of Computer Graphics consists of half engineering mechanics and structural engineering students, and half computer scientists, let me illustrate several examples related to structural engineering and architecture, all with graphic simulations.

Examples

It is not possible within this short presentation to fully describe the details of each example. However, I will use some pictures to illustrate these techniques and to show the enormous potential they have for the visualization of three-dimensional environments.

1. Finite element analysis

The dominant portion of time in a finite element analysis is spent in preparing the input mesh or in trying to interpret the results. This is particularly complex for arbitrary shaped surfaces in three dimensions. Using graphical preprocessors, one should be able to model the geometry and create the mesh very rapidly. To evaluate the results, we use color to represent the stress levels while still maintaining the three-dimensional perception. These examples of an airplane propellor illustrate the pre- and post-processing which are being used at our laboratory, particularly for brittle fracture. As the crack propagates in areas of high stress concentration, it is necessary to change both the topology and the geometry of the mesh at each analysis cycle. With the supercomputer capability, we hope to be able to witness these three-dimensional phenomena in real time, although it might be necessary to buffer the results of the simulation prior to display.

Frame analysis

This example illustrates a program called Steel 3D, which is currently available commercially. The three-dimensional frame is modeled interactively and after analysis, the results are also displayed in color. The real key to this sort of program is that the results of the analysis are then passed through a "building code design filter" to see whether the specifications are met. If not, the structure can be either interac-

Figures 4a, 4b. "Baroque Church" by *Dan Ambrosi*, Program of Computer Graphics, Cornell University.
These two figures illustrate another example of the ray-tracing technique. The roof and walls were modeled using quadric surfaces.

Bilder 4a und 4b. «Barocker Kirchenraum» von Dan Ambrosi (Program of Computer Graphics, Cornell University).
Ein weiteres Beispiel der Strahlen-Summentechnik; Decken und Wände sind mit Rechteckflächen modelliert.

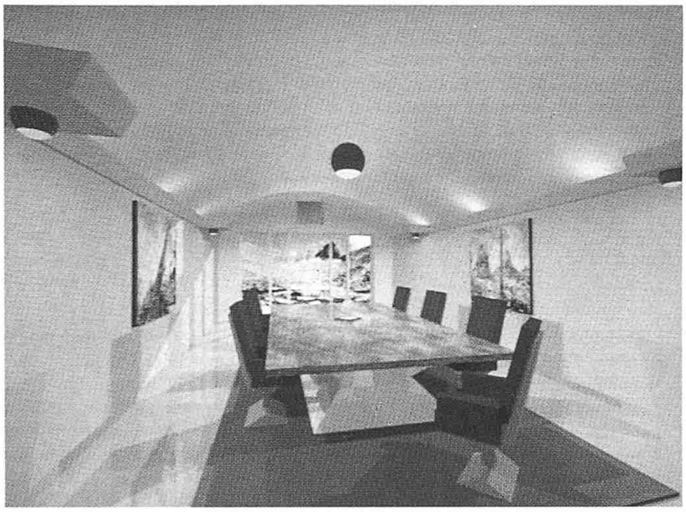


Figure 3b

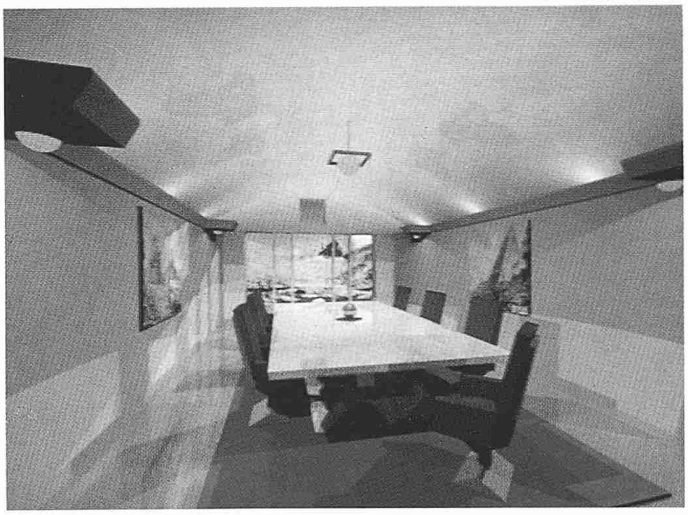


Figure 3c

tively or automatically resized to comply with the code requirements until a "optimal" design is reached. The program has been used by a number of companies involved with building design, and all indicate a significant reduction in steel weight. With supercomputer capability, we should be able to not only perform standard design, but evaluate the nonlinear behavior in such problems as soil structure interaction, or dynamic response to an earthquake.

Ray Tracing

In 1980, Turner Whitted [1] first introduced a new graphical simulation technique called ray tracing. In ray-tracing, a ray is sent through each pixel of the image plane into the environment. At each surface intersection, reflected and/or transmitted rays may be spawned. The final pixel color is obtained by combining the intensity contributions from all of the reflected and/or transmitted rays.

When using this approach, the computational demands are excessive. However, by using bounding volumes, hierarchical data structures, adaptive tree depths, and image coherence which substantially reduces the intersection calculations for the first ray, the times can be reduced by a factor of ten to twenty [2]. Thus, a ten-hour image of 1980 may only take one-half hour on a standard large minicomputer today. Since each image is dependent on the current observer position, it is costly to obtain dynamic sequences. However, despite the time requirements, the quality of the resulting images is impressive, particularly for specular environments.

Radiosity

The radiosity method, first introduced at Cornell [3], determines the light energy equilibrium of all surfaces in a diffuse environment. Surfaces are subdivided into small, homogeneous areas, and the interrelationships between elements can be described by a "form factor". A set of simultaneous equations is then generated and solved for the individual element intensities. Since this method is independent of the observer position, once the static environment has been computed in terms of its form factors and intensities, it is only necessary to render the image [4]. This can be shown dramatically by looking at the following sequence of images. Although the first image required approximately five hours of computation time, the lights could be turned out without having to recompute the form factors and geometric relationships. Thus the second image required only one-half hour for image generation. Even more dramatic is the third image, where the observer position is changed and only the rendering process has to be repeated. The computation time for rendering has been reduced to approximately fifteen minutes on our VAX 11/780.

We believe that the realism of these images is extremely important, both for the aesthetic evaluation of a design, or for the ability to portray the engineering parameters superimposed on the three-dimensional model. Our hope is that we can use these techniques to provide the scientists with the feedback necessary for an interactive session. With the additional processing power, which from October 1985 in our laboratory

Figure 4a

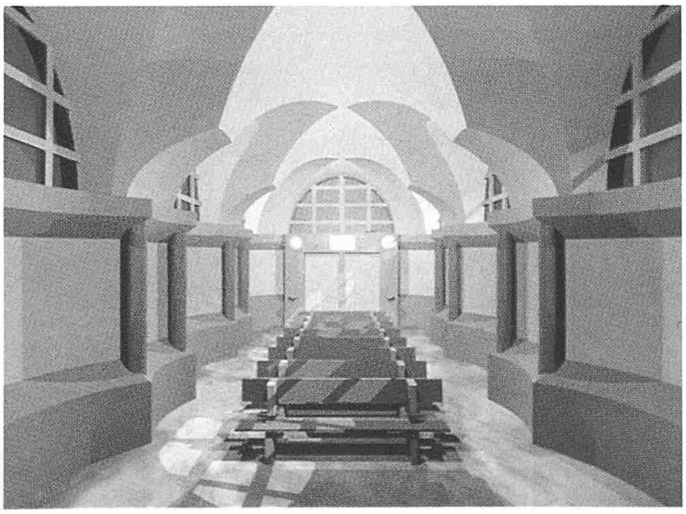
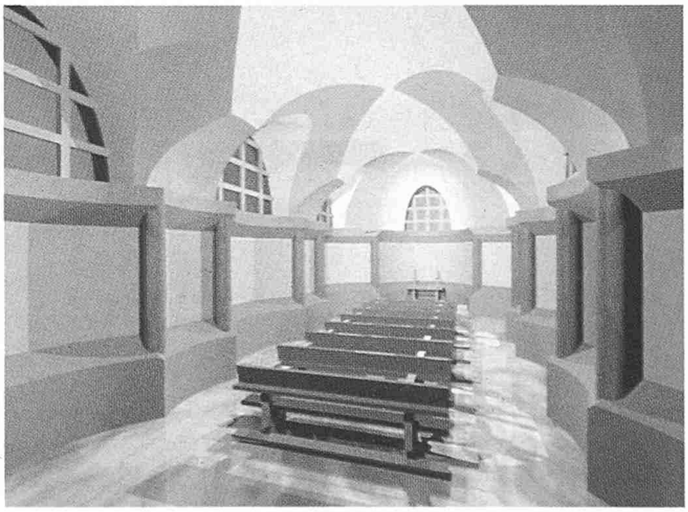
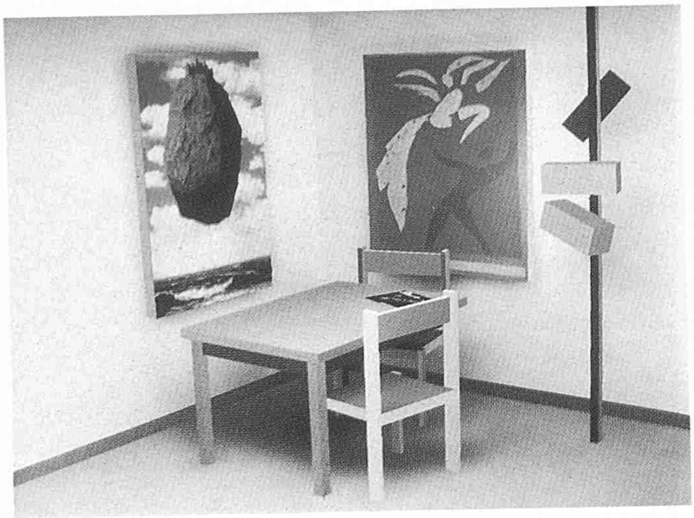


Figure 4b





Figures 5a, 5b, 5c. "Interior Rooms" by Prof. *Michael Cohen*, Program of Computer Graphics. Cornell University.
These three images were created using the radiosity approach. Although computation for the form-factors and surface intensities require several hours, since the algorithms are independent of observer position, subsequent images for dynamic motion can be generated in minutes. The approach is significant in that it correctly models the soft shadows of penumbras and the color-bleeding effects of diffuse environments.

Figure 5a

Bilder 5a bis 5c. «Innenräume» von Prof. Michael Cohen (Program of Computer Graphics, Cornell University), hergestellt nach dem Strahlungs-Prinzip. Obschon die Berechnung der Form-Faktoren und Oberflächenintensitäten mehrere Stunden beanspruchen, da die Algorhythmen nicht vom Beobachterstandort abhängen, können weitere Bilder für dynamische Bewegungen nachher innert Minuten erzeugt werden. Dieses Lösungsverfahren ist bedeutsam, da es die weichen Schattierungen im Halbschatten und die Farbauswascheffekte in diffuser Umgebung richtig darstellt

will be three-hundred times more powerful than what existed before, we expect to create images of this quality in the five-to-ten-second range. We hope that in a few years, as parallel processing techniques become available and as some of the algorithms can be implemented in microcode or hardware, this might approach the 1/30th of a second range necessary for dynamic simulation.

Conclusions

I hope that these topics I have briefly touched upon will indicate what is likely to happen in the future. The world is very complex and our quest for realism will not cease. Clearly some of this conjecture may seem unrealistic and there are many research problems ahead. Databases have not been completely worked out. Data transmission problems are not solved, and the software is woefully lacking behind the hardware.

How will it affect the architecture and engineering professions? Clearly, it will reduce the size of an office, and one hopes it will take the drudgery away from some of the tasks necessary to complete a project. If used properly, not only will there become an integration of architecture and engineering, partially because of the common database, but the emphasis on design creativity will be reestablished. With these powerful tools in the proper hands, our built environment can become substantially better. Expert systems and artificial intelligence methods will be incorporated in the software so the skill and knowledge of our past experiences can be embedded in new programs.

If I have one last request, it is that the people who are sitting in this room, most of whom have an architectural or engineering education, participate in the future software design process so that what becomes available is not specified by computer scientists, but by ourselves so that it is more suitable for our profession.

Authors address: Dr. *Donald P. Greenberg*, Dir. Program of Computer Graphics; Dir. Advanced Graphics, Theory Center; Jacob Gould Schurmann, Prof. of Computer Graphics, Cornell University, 120 Rand Hall, Ithaca, NY 14850-551, USA.

Acknowledgements
I would like to express my appreciation for the enormous support and participation of the entire staff and students of the Program of Computer Graphics. All of the systems developments, programming, and modeling was performed by this group of dedicated individuals. The support of the National Science Foundation and Digital Equipment Corporation is also gratefully acknowledged.

Edited version of paper presented to the SIA conference on Computer-Aided Design in Building held on September 11, 1985 at Basle.

Literature

[1] *Whitted, Turner:* An Improved Illumination Model for Shaded Display. Communications of the AMC, June, 1980

[2] *Weghorst, Hank; Hooper, Gary;* and *Greenberg, Donald P.:* Improved Computational Methods for Ray Tracing. AMC Transactions on Graphics, Jan., 1984, pp. 52-69

[3] *Goral, Cindy M.; Torrance, Kenneth E.; Greenberg, Donald P.; Battaile, Bennett:* Modeling the Interaction of Light Between Diffuse Surfaces. ACM Computer Graphics (Proceedings 1984), pp. 213-222

[4] *Cohen, Michael F.* and *Greenberg, Donald P.:* The Hemi-Cube: A Radiosity Solution for Complex Environments. ACM Computer Graphics (Proceedings 1985)

Figure 5b

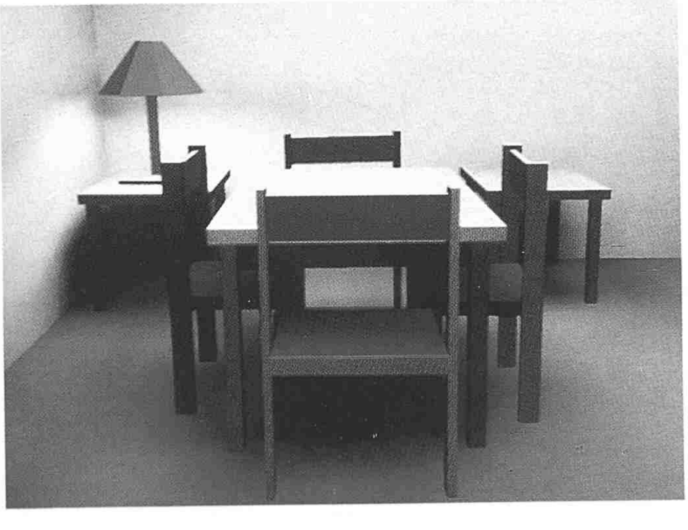


Figure 5c