Overview of Three-Dimensional Computer Graphics

DONALD H. HOUSE

Visualization Laboratory, College of Architecture, Texas A&M University

Three-dimensional computer graphics is called three-dimensional for several reasons. The graphics are generated by constructing a virtual 3D model which is then imaged, employing a physical simulation of illumination in three-dimensional space. Much of the research in the field is aimed at creating an illusion of three-dimensionality. Devices such as perspective, physically realistic shading, focus, atmospherics, and 3D motion for animation have been investigated [Foley et al. 1990; Watt and Watt 1992; Glassner 1995].

ORGANIZATION OF A 3D GRAPHICS SYSTEM

A three-dimensional graphics system can be organized into three major components: scene specification, rendering, and image storage and display. Figure 1 gives a schematic view of the process used in three-dimensional graphics and shows the role of each component.

Scene Specification. *Scene specification* provides an internal 3D representation of the virtual scene to be imaged. Its interface may be interactive or programmable. Scene specification requires a concept of geometric coordinate system and ways of describing geometry, virtual materials, and lighting.

Each *local coordinate system* used to describe the geometry of the scene is defined with respect to an external reference system. These might themselves be specified relative to other external systems, allowing models to be described in a *hierarchical fashion*, as shown in Figure 2.

Coordinate systems are often defined by 4D *homogeneous transformation matrices*, each specifying a transformation from the local system to its reference system. The 4D homogeneous form allows the unification of the basic transformations of translation, scale, rotatation, and shear in a single representation. Complex affine transformations are built up by matrix multiplication.

For *geometric modeling*, the simplest geometric primitives are points, orientation vectors, lines, and polygons. These are arranged together to form more complex objects.

More sophisticated modelers provide *parametric surfaces* and/or *implicit surfaces*. Parametric surfaces are defined via an underlying piecewise polynomial formulation [Bartels et al. 1987; Rogers and Adams 1990]. A typical formulation is a *biparametric surface*, defined in Figure 3. An *implicit surface* is the set of points of satisfying a mathematical expression of the form $F(x, y, z) = 0$. Implicit techniques can be generalized to describe highly complex forms [Ebert 1994].

Volume data is geometric data in the form of a set of scalar values distributed in a three-dimensional field. A typical example is data from medical scanners.

The *material* is the attribute of a geometric object that describes how its surface reflects and refracts light. This is usually specified by providing parameters to a generalized function. A material specification also usually includes provision for a set of maps that modify basic color or geometry information as a function of parametric position on a surface.

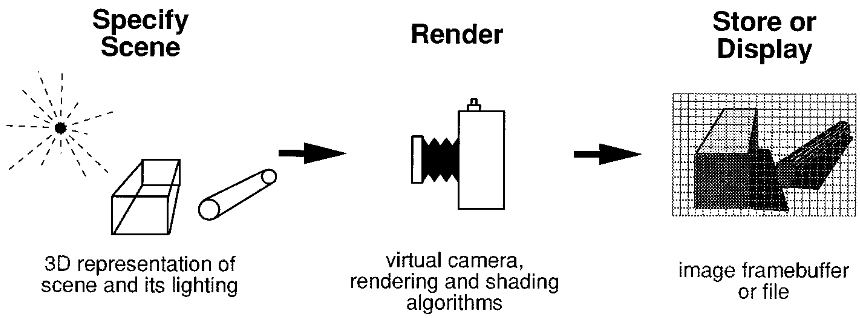

Fig. 1. The 3D graphics process.

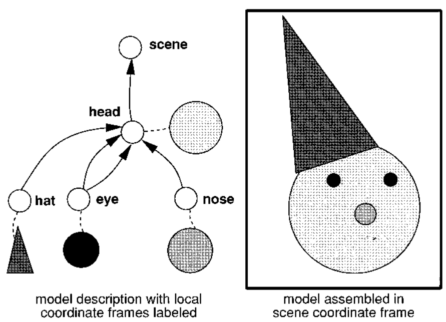

Fig. 2. A clown described by a hierarchy of local coordinate frames.

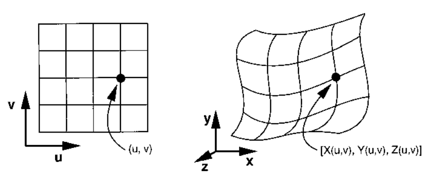

Fig. 3. A biparametric surface.

Lights provide the illumination source for simulated shading calculations. Some rendering algorithms work with light sources with geometric properties such as shape and area, but most systems work with variants of *infinite* and *point* lights. These and some of their variants are illustrated in Figure 4.

Rendering. Rendering is the process of transforming a 3D scene description into a 2D image. Making this process both physically realistic and algorithmic, so that it can be done efficiently and accurately, is the essence of the traditional rendering problem.

The *renderer* is the engine driving the image-making process. Simple renderers consider only the direct interaction of lights with surfaces. More sophisticated renderers attempt to solve the *global illumination* problem, taking into account interreflections among objects.

The key steps in rendering, not necessarily completed in this order, are the following:

(1) Orienting the 3D scene to the camera's position.
(2) Projecting points in the 3D scene into their images on a 2D virtual image plane.
(3) Deciding which of the surfaces projected onto the image plane are visible.
(4) Fixing a set of sample points on the 3D geometry.
(5) Determining the shade or color reflected or transmitted to the screen for the visible sample points.
(6) Constructing the image using the calculated sample shades.

The *virtual camera* provides both a point of view from which to render an image and the basic parameters of the mathematical projection used to form the virtual image.

The *shader* is the algorithm that uses the information collected by the renderer about a point sample on the scene

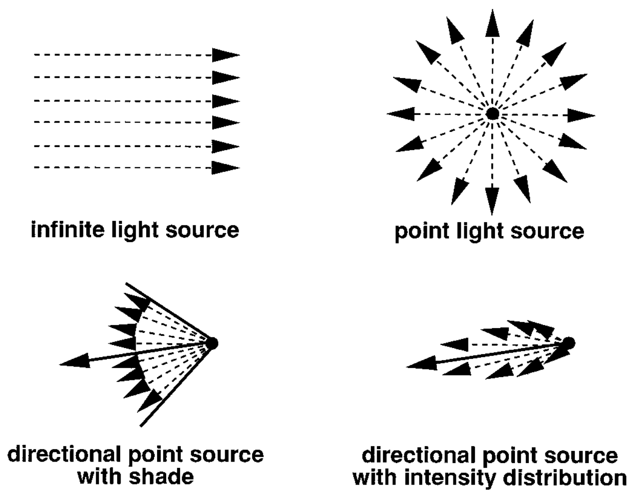

Fig. 4. Infinite and point light sources and variants.

geometry, its material attributes, and the available lighting to calculate a color for the sample.

A *digital image* is constructed from a set of shaded samples by some form of low-pass filtering and resampling [Wolberg 1990]. In simplest terms, the digital image pixel grid is superimposed over the virtual screen, then each pixel is colored with a weighted average of the shaded samples in the pixel's vicinity.

Image Storage and Display. Rendered results must be turned into tangible images that can be viewed, stored, and transmitted. Thus, a three-dimensional graphic system is organized around digital image data structures, a model of display technology, and file formats.

The *pixmap* is the basic data structure for the storage of digital images. It is simply a 2D array of pixel values, each stored in units of one or more bits. The number of bits per pixel determines how many distinct colors or other attributes an image may have.

The most frequently used display device is the color *cathode ray tube* (CRT), interfaced to the pixmap data structure via a hardware *framebuffer*. The framebuffer is simply an array of computer memory that holds the pixmap and electronics providing controls to draw the image frame as a raster on the CRT.

Image *file storage* efficiency is an important issue due to the very large number of pixels in an image. Fortunately, various forms of internal image coherency lend themselves to exploitation in the development of file-compression techniques. For these and also for commercial reasons, the number of image file formats in common use is quite large [Murray and vanRyper 1994].

RESEARCH ISSUES

Several research areas are timely and important to the development of the field. In rendering, they are improving global illumination techniques and developing nonphotorealistic techniques. In modeling, they are exploring volume methods, improving interactive techniques, and effectivly modeling natural forms. In computer animation, important research areas are physically based modeling, applications of AI and ALife, and interactive techniques. Finally, there is the related field of virtual reality.

REFERENCES

BARTELS, R., BEATTY, J., AND BARSKY, B. 1987. *An Introduction to Splines for Use in Computer*

Graphics and Geometric Modeling. Morgan Kaufmann, Los Altos, CA.

EBERT, D. S., ED. 1994. *Texturing and Modeling: A Procedural Approach*. AP Professional, Boston, MA.

FOLEY, J., VAN DAM, A., FEINER, S., AND HUGHES, J. 1990. *Computer Graphics Principles and Practice*. Addison-Wesley, Reading, MA.

GLASSNER, A. 1995. *Principles of Digital Image Synthesis*. Morgan Kaufmann, Los Altos, CA.

MURRAY, J. AND VANRYPER, W. 1994. *Encyclopedia of Graphics File Formats*. O'Reilly, Sebastopol, CA.

ROGERS, D. AND ADAMS, J. 1990. *Mathematical Elements for Computer Graphics*. McGraw-Hill, New York.

WATT, A. AND WATT, M. 1992. *Advanced Animation and Rendering Techniques*. Addison-Wesley, Reading, MA.

WOLBERG, G. 1990. *Digital Image Warping*. IEEE Press, Los Alamitos, CA.