SLICE DISPLAY FOR HUMAN MODELS CONSTRUCTED WITH SPHERES

Norman I Badler, Barry B. Hunter, Jr.
and Cary Phillips

MS-CIS-88-54
GRAPHICS LAB 23

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

July 1988

Acknowledgements: This research is partially supported by Lockheed Engineering and Management Services, the Pennsylvania Benjamin Franklin Partnership, NSF grants MCS-82-19196-CER, IST-86-12984, DMC-85-16114, IRI84-10413-AO2 and ARO grants DAA29-84-9-0027, DAAG29-84-K-0061 including participation by the U.S. Army Human Engineering Laboratory.

Slice Display for Human Models Constructed with Spheres

Norman I. Badler
Barry B. Hunter, Jr.[1]
Cary Phillips

Computer Graphics Research Laboratory
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389

June 10, 1988

ABSTRACT

Bubblepeople are human figure models built from overlapping spheres. They provide a softer and more realistic representation of the human body than polygon-based models, require fewer primitives, and are easier to articulate. Unfortunately, displaying a body constructed entirely of spheres by conventional rendering techniques is relatively slow. A new slicing algorithm has been implemented which significantly decreases the time required to display solid shaded Bubblepeople as polygon models within a conventional scanline renderer.

1. Introduction

Creating natural and realistic human figure animations requires repeated generation of rendered images containing one or more human models per frame. Although a number of three-dimensional representations for human figures have been used including polygons, ellipsoids, and Bezier patches, all seem to have a rigid, robotic quality. The Bubblefigures, on the other hand, consist entirely of colored overlapping spheres and offer unusual qualities of softness, inherent curvature, and simple joint articulation [2].

Each Bubblefigure contains about 1700 spheres. When placed in a polyhedral scene a large proportion of the calculations in the rendering process were being used to shade the Bubblepeople. We therefore sought methods to substantially increase the efficiency of rendering scenes with Bubblefigures. An important attribute of the Bubblefigure display is that the spheres are *not* rendered to appear as true spheres (as in molecular displays), but are rather specially treated to produce more smoothly shaded surfaces.

2. Display algorithms

The original algorithm to display Bubblepeople used a special-purpose disk scan converter. Each sphere is displayed as a flat shaded disk, using simple depth priority and depth cueing to establish visibility and shading. We were reluctant to modify our existing scan-line visible surface system to accomodate the maintenance of multiple incremental copies of the disk drawing algorithm [1]. So methods which relied on existing polygon scan conversion processes within the scan-line system were favored.

The usual approach to sphere display in a scan-line environment is to tile it at some level of detail with flat polygons. This is extremely wasteful because of the great number of spheres and the small area of screen that will actually be covered by a particular sphere. Moreover, for Bubblefigures the display

---

[1] Present address: Computer Graphics Lab, New York Institute of Technology, P. O. Box 170, Old Westbury, NY 11568

advantage of shading the projected sphere as a disk obviates the tiling requirement. It suffices to project each sphere onto a plane perpendicular to the line of sight passing through the center of the sphere and decompose the projection into an *n* sided polygon approximating the projected disk. This results in rendering an *n* sided polygon as opposed to the entire sphere. This method is quicker than tiling but still requires one polygon per sphere.

From these techniques we can conclude that detail on the sphere level is not important since an individual sphere does not significantly contribute to the image; rather, the collection of the spheres on a whole is the important factor.

The technique to be used would have to process the spheres in a group and reduce the number of individual polygons to be rendered. One possible solution is to take each segment of the body, collapse each sphere into an *n* sided polygon perpendicular to the line of view (as described in the proceeding method), and collect each vertex of those generated polygons in a list. Use this list to create a convex-hull-polygon of those vertices in the list and send this convex-hull-polygon for each segment to be rendered. Since the renderer only accepts flat polygons (a polygon with a constant surface normal for every point on the surface), the mean z value (in the view transformed coordinate frame) of the segment is assigned to each vertex of the polygon. Finding the convex-hull-polygons drastically reduces the number of polygons to be rendered and results in a substantial time savings. However, the resulting image will appear flat or without depth and therefore look like a projection rather than a three-dimensional person. Further, since the depth of the Bubblepeople in relation to a normal environment would be small, the person would appear to be flat (constant) shaded.

Depth cueing is needed to improve depth perception. First, depth must be perceived between different body segments. For example if the bubble figure is aligned face toward the viewer with the right hand extended outward and the left hand straight down, the right hand is to be slightly brighter than the left. Second, depth must also be detectable within segments depending on their orientation. If a bubblefigure's arm is extended in some orientation toward or away from the viewer, the closer portion of the arm (from the viewer's perspective) should appear brighter.

To give the depth illusion between segments it was decided to render the polygons from Bubblefigures differently than other polygons in the scene. To differentiate these cases a field is added to the polygon record indicating which type of rendering is to be used on that polygon (similar to the shade tree notion of [reference: Cook]). Further, fields are added to indicate minimum and maximum z values (relative to the view) for the current Bubblefigure. Attenuation values determine the brightness for the closest point on the figure and the furthest. Rather than using a surface normal lighting model for these tagged polygons they are depth shaded. Intensity values are calculated by the selected shading model, but are then linearly attenuated depending on the current depth and attenuation value. The greater the difference between the back and front attenuation values the more noticable the depth shading. The attenuation values are determined symmetrically with respect to the body's center torso. Thus, as the figure moves it retains the same body shading relative to itself and does not grow brighter or dimmer as a whole.

Although the depth shading technique does add depth appearance the figures still look flat. The problem is that segments with depth will still appear as one polygon with the same z value. Since the mean z of the segment is assigned to each vertex of the convex-hull-polygon to insure the polygon is not curved. A solution to this problem is to divide the segments into slices. This is done by first passing a plane in the x-y axis through the segment's center of mass. Construct the front slices by adding a constant from the center of mass z component until that z value no longer is contained in the segment. Likewise construct the back slices by subtracting from the z component. The constant we selected is three centimeters in world coordinates. Next, for each slice collect all the spheres from that segment which intersect the current slice. This is an inexpensive operation since we need only compare the *z* value of the current slice and the *z* component of the sphere's center and its radius. If the sphere does not intersect the current slice it is ignored, otherwise the sphere is collapsed into a polygon, as described earlier, with its vertices placed in a list. When all the spheres have been checked in this slice the convex-hull-polygon is found from the list and assigned the *z* value from the current slice. This results in a segment having a varying number of slices depending on its orientation. For example an arm oriented straight down may only have three slices but an arm oriented toward the viewer may have as many as eighteen. This provides the depth shading desired and results in a realistic three-dimensional image of a Bubblefigure [Figures 3-1].

3. Conclusions

The slicing technique does produce a large number of polygons but still significantly less than the total number of spheres. Rather then rendering hundreds of spheres for each segment, only a few dozen polygons are created. The resulting polygons are also suitable for z-buffering hardware which further reduces the rendering overhead. Data for three positions of the Bubblefigure are listed in Table 3-1: A frontal view standing, a frontal view sitting, and a side view standing. The table shows the number of slices (polygons) per figure and the total number of vertices associated with those polygons. In comparison, collapsing each sphere into a polygon of eight vertices would generate 1700 polygons with a total of 13,600 vertices per figure for any position or view. Converting the spheres to icosahedra (a rather poor excuse for a sphere) would have created 34,000 triangles.

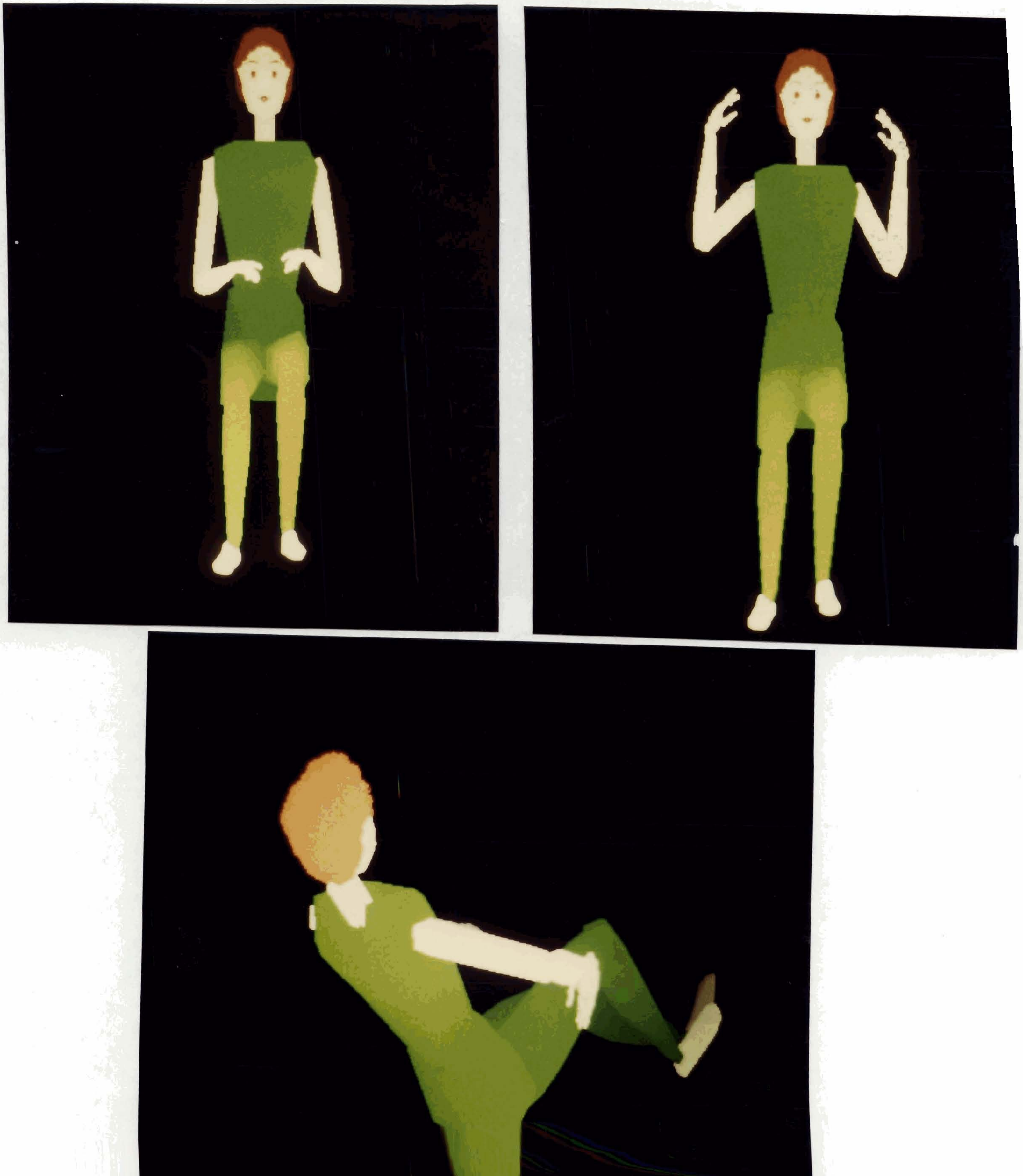
This technique only works with segment shapes that are convex because of the convex hull step. The figure's head and facial features, for example, are more correctly rendered directly from the spheres with the original depth cueing algorithm to preserve the concave structure.

While the convex hull step is not expensive (it is done by an *O(n log n)* algorithm, where *n* is the number of points), it is additional overhead between the transformation traversal figure positioning and the rendering. The entire process described here is therefore more advantageous in a frame buffer display environment where all processing is done in software anyway, than in a high performance workstation where pipelined polygon transformation and rendering is best left uninterrupted.

Table 3-1: Comparison between sliced polygon data generated from different representative body views and configurations

Position	Slices	Vertices
Standing	42	621
Side View	43	980
Sitting	69	1090

Figure 3-1: Three views of the Bubblewoman figure sliced and rendered with depth cued polygons.



References

1. Badler, Norman I. "Disk generators for a raster display device". *Computer Graphics and Image Processing 6* (1977), 589-593.

2. Badler, Norman I., Joseph O'Rourke, and Hasida Toltzis. "A spherical representation of a human body for visualizing movement". *IEEE Proceedings 67*, 10 (Oct. 1979), 1397-1403.