ANL/IFR/CP--82835
Conf-950311--5

COMPUTER IMAGING OF EBR-II FUEL HANDLING EQUIPMENT

by

G. G. Peters and L. H. Hansen

Argonne National Laboratory-West
P.O. Box 2528
Idaho Falls, ID 83403-2528

BNES Conference - Fuel Management and Fuel Handling

March 20-22, 1995

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Abstract - This paper describes a three-dimensional graphics application used to visualize the positions of remotely operated fuel handling equipment in the EBR-II reactor. A three-dimensional (3D) visualization technique is necessary to simulate direct visual observation of the transfers of fuel and experiments into and out of the reactor because the fuel handling equipment is submerged in liquid sodium and therefore is not visible to the operator. The system described in this paper uses actual signals to drive a three-dimensional computer-generated model in real-time in response to movements of equipment in the plant. This paper will present details on how the 3D model of the in-tank equipment was created and how real-time dynamic behavior was added to each of the moving components.

Introduction

EBR-II is a liquid metal fast reactor operated by Argonne National Laboratory for the U.S. Department of Energy and is located approximately 35 miles west of Idaho Falls, Idaho. An aggressive fuel unloading program has been initiated in preparation for the reactor's ultimate decommissioning. Reactor fuel and experiments are being exchanged with reflector assemblies in the reactor vessel via remotely operated fuel handling equipment which is submerged in a 26 foot diameter tank containing approximately 86,000 gallons of 700°F liquid sodium. The fuel handling equipment consists of all the components necessary to insert or remove subassemblies from the reactor core and insert or remove them one at a time from the primary tank. This equipment includes the core gripper which can be rotated over any one of 637 subassembly locations in the reactor core and the transfer arm which transfers a subassembly between the core gripper and a temporary storage basket. After cooling in the storage basket for approximately two months, spent fuel is transferred from the storage basket to the transfer arm which transfers the fuel to the fuel unloading machine gripper which in turn pulls the fuel out of the primary tank through a nozzle in the top of the tank.

The liquid sodium in the EBR-II primary tank has prevented direct observation of fuel handling operations since the reactor was constructed in the early 1960's. A three-dimensional (3D) visualization technique is necessary to simulate direct visual observation of the transfers of fuel and experiments into and out of the reactor. Operators currently rely on lights, meters and digital LED displays to determine the positions of the fuel handling equipment. In the new system, however, the operators will be provided with an additional three-dimensional computer-generated display of the moving equipment. This display will improve the operators' understanding of the current state of the system as well as provide a means of troubleshooting mechanical or misalignment problems. It is hoped that this system will decrease the probability of damage to in-tank equipment due to operator error.

Initial Work Tasks

The first task was to create a detailed 3D computer model of the major components located within the primary tank. This model was first created in AutoCad™ using the original construction drawings and as-built dimensions taken before the tank was filled with sodium. The AutoCad model requires approximately 30 MB of storage and contains over 1.5 million polygons (Figure 1).

The I-DEAS™ Solid Modeling software from Structural Dynamics Research Corporation was then used to import and display the 3D computer model. The I-DEAS software is a powerful 3D modeling package capable of creating and modifying complex geometries, performing interference checks, viewing and manipulating 3D models, and other similar tasks. The original intent was to use this software to animate the model in real-time from actual data. This approach, however, presented some difficulties that are common to many CAD oriented programs. For example, CAD applications are intended for design of mechanical systems and therefore are able to show how mechanical parts move and fit together but are not usually designed with real-time animation in mind with interfaces to actual data.

Work supported by the U.S. Department of Energy, Nuclear Energy Programs, under Contract No. W-31-109-Eng-38.

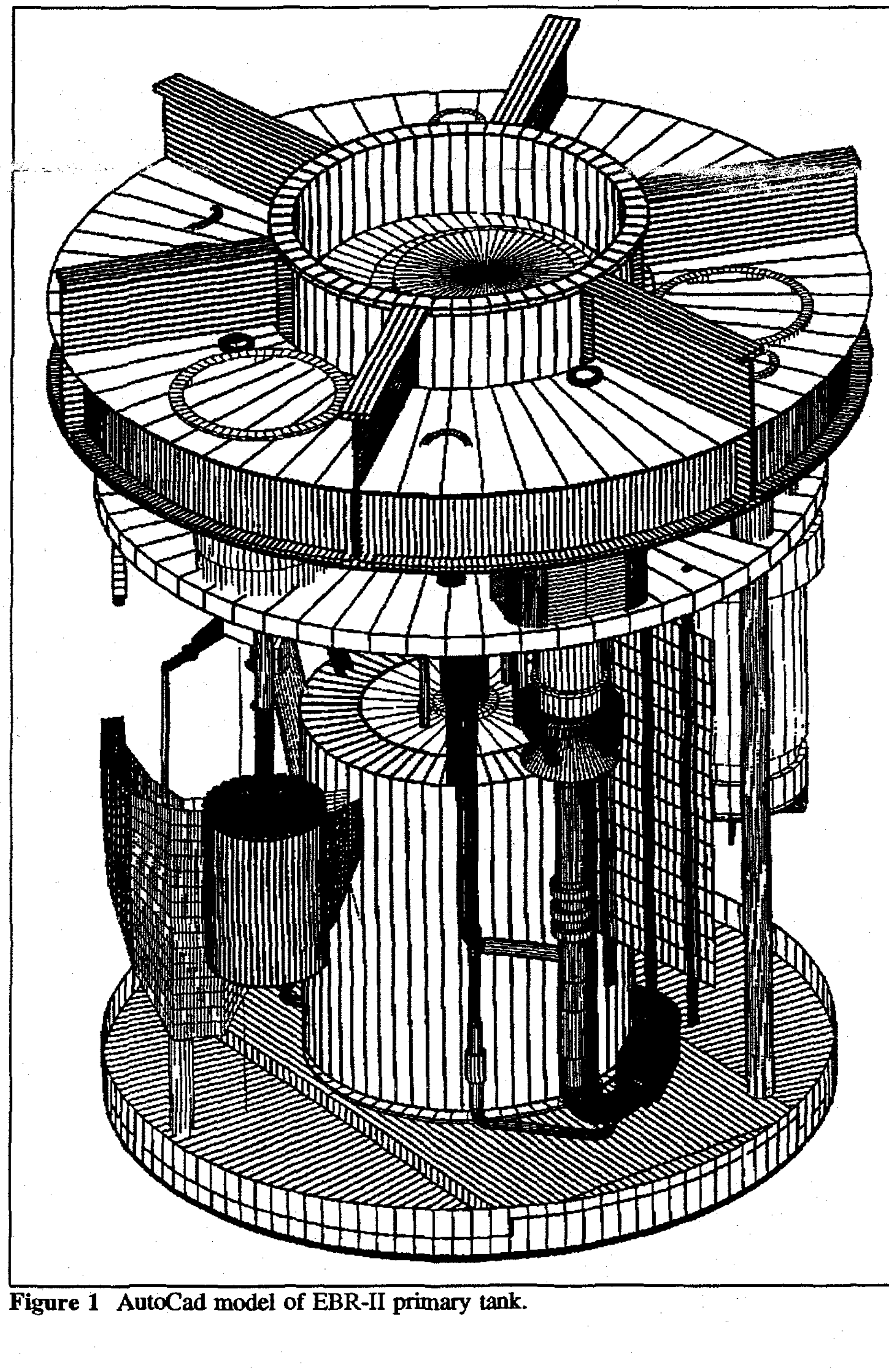

Figure 1 AutoCad model of EBR-II primary tank.

Another difficulty encountered was that the overall structure and extremely high polygon count of the AutoCad model prevented real-time display. The model was highly detailed and could not be trimmed down without any loss of information.

New Approach

In order to solve these problems, a new approach was required. Functional requirements were established in the search for a high-performance graphics software application. Among the important criteria was the data interface from shared memory. Another key specification was real-time visual simulation that would replicate the fuel handling operations as they were occurring during fuel handling sequences. Levels of detail switching was also an important consideration so polygons too small to be seen when zoomed out would be eliminated. Also considered was the ability to run the graphics outside the development application and without the window manager as an operator interface and the flexibility of producing C source code to interface with the graphics application.

The software selected for the new approach was Designer's Workbench (DWB) by Coryphaeus Software, Inc. DWB is a high-performance graphics software application that meets all of the above functional requirements including interfacing directly with shared memory and supporting varying levels of detail in the model. Although DWB can import model files from AutoCad, optimum performance could only be achieved by reconstructing the fuel handling equipment using the new software. This is because DWB is graphics-oriented rather than CAD oriented. A graphics-oriented application is more conducive to polygon control and structure detailing. For example, objects that interface with other objects in the model contain polygons at the interfaces which are known to be hidden from view. Therefore, these polygons may be eliminated from the model since they will never be seen. In a CAD environment, however, objects are generally treated as geometric entities such as cylinders and spheres rather than as groups of polygons so such optimizations are not possible.

Performance Optimizers for Real-Time Display

A prime advantage of using an application for both modelling and animation development is the amount of control maintained by the developer. This ensures predictable and optimal results during construction and execution. When the 3D computer model of the primary fuel handling components was constructed, several factors were considered during the development phase. Some of the factors included how to best construct a particular component, how to structure the hierarchical tree, and what type of shading to use. Each determines the level of realism and real-time performance achieved.

Component Construction

There were many important factors which had to be considered when constructing the 3D computer model. One consideration was the number of sides or facets to apply when creating a cylindrical object. A large number of sides improves the smoothness of the curve but results in a degradation in performance during animation. When proper shading is applied, very few sides are required in order to achieve a cylindrical appearance, especially when the object is relatively small. Large objects required more facets in order to appear smooth. We found that many of the objects in our model required as few as 5 to 7 sides in order to appear round. This resulted in a significant reduction in the number of polygons required to create the model as compared to the AutoCad version. A partial model consisting of the major fuel handling components was reduced from 500,000 polygons in the AutoCad version to 1700 polygons in the DWB model.

Constructing 3D objects using polygons involves a number of other important performance considerations. For example, polygons should be limited to three or four vertices. This restriction greatly improves the graphics performance since triangles and 4-sided polygons will draw several times faster than polygons containing five or more vertices. This is because the Silicon Graphics computers are designed with dedicated graphics engines that are optimized for triangles and 4-sided polygons. It is also advantageous to use only planar polygons which have vertices that are all in the same plane. Non-planar polygons are difficult to render properly and can slow down the graphics speed considerably.

Another modelling consideration concerns the use of convex versus concave polygons (Figure 2). Concave polygons tend to cause problems with some computer graphics algorithms such as shading and therefore should be avoided. When the model required the use of a concave shape, a concave polygon of the desired shape was first constructed and then reduced to a mesh of triangles which guarantees that the polygons are convex and planar. Figure 3 shows an example of this for the gripper jaws.

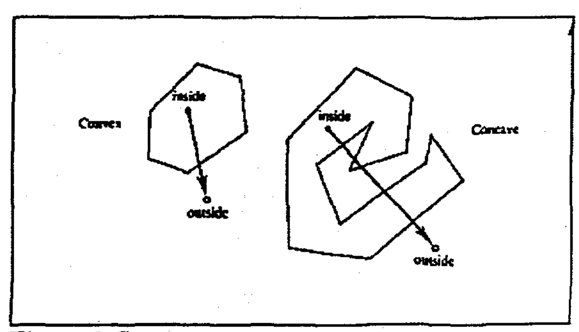

Figure 2 Convex vs. concave polygons.

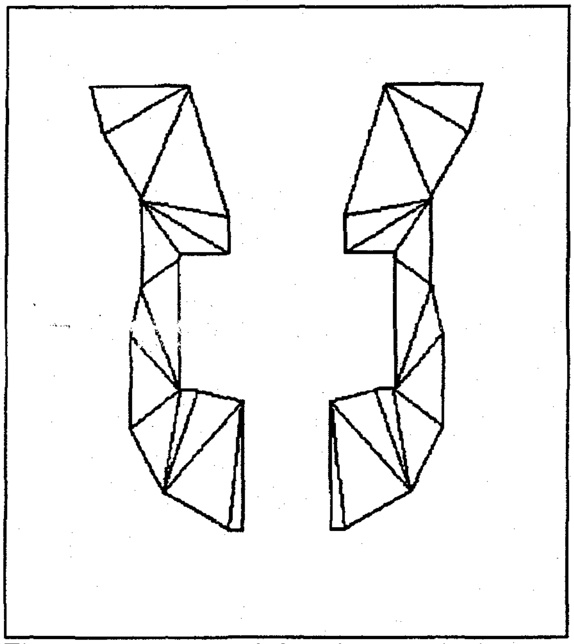

Figure 3 Triangle mesh for gripper mechanism.

Also during construction, care was taken in attaching the polygons. T-vertices occur when a vertex is attached to an edge where there is not a vertex. These should be avoided since cracks can appear in the display where the background color comes through. Several methods were used to combat this problem. One solution is to create a vertex on the edge for the other vertex to attach or place the object slightly inside the edge.

Structuring the Database Hierarchy

DWB automatically generates a database hierarchy in which groups of polygons, vertices, strings, levels of detail, etc. are placed at different levels of the hierarchy tree. These groupings allow special attributes or behavior to be applied to entire groups of polygons. Levels of detail switches, for example, were placed on small details of the model, so that these polygons are drawn only when the eyepoint is zoomed in close enough to see them. Thus, the same object was sometimes modeled two or three times, each with a different amount of detail. When the switch points are placed correctly, the observer is not able to detect the transition points and the model transitions smoothly as the zoom scale is increased.

This database hierarchy is also important when adding dynamics to the model. Polygons can be grouped together so that a common dynamic behavior is applied to an entire object. Since link dynamics can be at any level of the hierarchy, the structure of the tree controls the particular section affected. Some objects, such as the core gripper structure, have an entire hierarchy of objects at different levels. This allows the gripper jaws to open and close, the gripper shaft and jaws to raise and lower, and the entire gripper structure to move with the rotating plugs.

Shading

DWB supports a number of different techniques for creating shaded views of 3-D objects. In general, more realistic shading effects require more number crunching and therefore result in a decrease in speed. Although flat shading provides the fastest graphics performance, the resulting images will appear flat or faceted rather than smooth. As stated before, cylindrical objects may appear round with as few as 5 to 7 sides when proper shading is applied. This can be done using smooth Gouraud shading. Smooth Gouraud shading averages the surface normals of any adjacent polygons which results in a common light intensity along any shared edge between two polygons. Most of the fuel handling components were either illuminated or Gouraud shaded for more realistic effects.

Conclusion

An approach for visualizing 3D environments in response to real-time data has been presented. Critical components directly related to fuel handling operations were modelled. Dynamic behavior of the moving components was then incorporated into the model by creating links to database elements that define the translational and rotational movements of each component. Data from shared memory can be used to animate the model in real-time. Data can also be accessed from a file for replaying particular transfer sequences.

Several performance optimizations have been described which significantly improve the graphics performance of the system. These include eliminating unnecessary polygons from the model, using only 3 or 4 sided convex polygons, incorporating levels of detail switches, and using shading techniques that minimize the number of polygons required to achieve rounded surfaces. Tradeoffs between rendering time and level of realism must always be considered.

Future Work

Future work will include using texture maps to achieve a higher degree of realism and incorporate more detail in certain portions of the model. The core area in particular is very difficult to model efficiently in that it consists of 637 subassemblies which each require at least 100 polygons. Texture mapping allows the software to project a two-dimensional image, or texture map, onto a surface in the three-dimensional model. Using this technique, a computer system can render relatively simple geometric models with a great amount of apparent complexity. Photographs of the reactor core and storage basket funnels will be scanned, edited, and imported into DWB where they will be incorporated into the 3D model.

The final goal of this project is to incorporate the 3D real-time display of the fuel handling equipment into an upgrade of the operator control console to improve fuel handling surveillance. Instrumentation located outside of the primary tank will sense the positions of moving components. New position sensors for the storage basket and transfer arm have been designed and will measure the elevation and rotation of each component. Signals from these instruments will interface directly with a SUN Sparc 20 workstation via A/D boards installed in the backplane of the computer. These signals will be indicated as readouts on a 2D graphics display and sent over the ethernet to a Silicon Graphics Indigo2 computer for updating the 3D model.