Comparison of Surface Rendering Techniques for 3-D-Tomographic Objects

U. TIEDE, K.H. HOHNE, M. RIGNER

Institute of Mathematics and Computer Science in Medicine (IMDM), University Hospital Eppendorf, Hamburg, F.R.G.

Introduction

In the past few years several algorithms of producing 3D-images from sequences of cross-sectional images (MRI and CT) have been developed. In this paper we compare the three most important algorithms (polygon approximation, binary voxel and gray scale voxel method) and demonstrate their properties with the same data set.

Method

Fig. 1 shows an overview of the applied methods together with a set of typical results. A method known from CAD is the representation by triangles (Triangulation). For its application contours of the object to be visualized have to be extracted from the single slices. These contours are then connected by triangular surfaces through heuristic algorithms [1]. The main problems are the connection of corresponding contours and the optimal placing of the triangles. For the latter problem we have developed a special algorithm [2].

A class of algorithms based on surface representation by surface voxels has been pioneered by G.T. Hernan [3]. In this binary voxel method the gray level image sequence is interpolated to give a gray scale volume with cubic voxels. From this volume a binary volume is produced containing the object to be displayed (mostly bone). This volume can then be converted into special data structures (e.g. octree) from which projections can be produced. Alternatively the surface images can be produced from the volume directly. We have developed a "back to front" algorithm that scans the binary volume without needing the time consuming bit access.
Fig. 1: Overview of the most important 3D reconstruction methods
In order to get a shaded surface, the surface normal has to be computed from the visible surface elements. For triangulated surfaces generally Phong interpolation is used [4]. For the voxel method a distance image is computed first, which contains the distance of surface voxels nearest to the image plane. Several algorithms for the estimation of the surface normals from this distance image have been described [5]. We have developed an algorithm that also produces shadows from the distance image [6].

In the gray scale voxel method the gray scale volume is also interpolated but no a-priori segmentation is performed. The surface normals are here estimated from the gray level gradients at the surface [7]. The images of this paper have been produced with an improved version of this algorithm.

Results

The different algorithms have been run on a VAX-11/780 using the program system Voxel-XAN, which is written in Pascal. The relative speed of the different methods is listed in table 1. It turns out that for diagnostic purposes the triangulation approach (Fig. 1 first row, Fig. 2a) is not very suitable. Only simple object surfaces such as the outer skin can be generated automatically. Only surfaces, but no volumes and cut planes can be visualized.

The binary voxel method (Fig. 1 center row, Fig. 2b,c) does not have these drawbacks. The quality of surface reconstruction, however, depends strongly on the slice distance, which leads to artifacts at slice distances of more than 4 mm. Because of the reasonable computation times, however, it is the method of choice for routine applications.

The gray scale voxel method yields images with the highest smoothness and dynamic range of shading (Fig. 1 bottom row, Fig. 2d). Yet it is restricted to surfaces defined by interfaces of materials of basically uniform density (bone-soft tissue, soft tissue-air). In addition the quality is paid for with a higher cost of computation time. It will certainly replace the binary method as soon as more computation power is available. Other properties of this method are discussed in [8].
Fig. 2: Results of surface rendering for different surface representations using the Phong illumination model.
Table 1: Processing time of different surface rendering algorithms on a VAX-11/780 computer using Pascal implementation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Preprocessing</th>
<th>Projection</th>
<th>Shading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contour: 24 min.</td>
<td>37440 vertices</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triangulation (skin surface)</td>
<td>7.49 min.</td>
<td>75133 triangles</td>
<td></td>
</tr>
<tr>
<td>Voxel method (bone surface)</td>
<td>Linear interpolation</td>
<td>8,059,319 voxels</td>
<td></td>
</tr>
<tr>
<td>1 bit res.</td>
<td>Thresholding: 15 min</td>
<td>Back-to-front:</td>
<td></td>
</tr>
<tr>
<td>8 bit res.</td>
<td>Trilinear interpolation</td>
<td>Ray-trace:</td>
<td>18 min</td>
</tr>
<tr>
<td>Gray scale volume*</td>
<td>Rotation: &gt; 2 h</td>
<td>Gray level gradient:</td>
<td>9 min</td>
</tr>
</tbody>
</table>

* Experimental version

References


