Proceedings of ALGORITMY 2000
Conference on Scientific Computing, pp. 368 - 378

PRECISSION OF ISO-SURFACE EXTRACTION FROM VOLUME DATA AND VISUALIZATION

VÁCLAV SKALA[1]

Abstract. There are various methods for extracting iso-surfaces from volumetric data. Many specific techniques have been developed to increase the speed of computation and decrease memory requirements. A comparison of the selected methods was made in different aspects. Surprisingly, experiments proved, that there is no direct relation between precision and human perception of the extracted iso-surface.

Although a precision of the iso-surface extraction is very important as well, it is not mentioned usually. After the extraction, the surface (usually the triangular mesh) is displayed by a standard package that uses algorithms like constant, Gouraud or Phong shading, for shading polygonal meshes. Unfortunately the Gouraud and Phong shadings cannot be used in some cases as these techniques prefer the "smoothness" to "precision" of the displayed surface. In cases, when the colour of the triangle of the triangular mesh is given in a representative inner point for each triangle, both methods can lead to very misleading results. The S-shading method that respects the smoothness and precision in some extent can be used.

Keywords: volume data visualization, marching cubes, marching tetrahedra, iso-surface extraction, algorithm complexity, computer graphics, shading models, polygonal mesh shading.

1. Introduction. Volume data sets are often acquired by scanning a material of interest by using Magnetic Resonance Imaging (MRI), Computed Aided Tomography (CT), Positron Emission Tomography (PET) etc. Particular data acquisition devices, such as CT and MRI scanners, are good at sampling a specific characteristic of a substance. This characteristic is usually different for each kind of device used, e.g. MRI scan shows soft tissue, while CT scan shows hard material, like bones etc.

The advantage of the volume data representation is that all volume data sets can be treated similarly even though they are generated by diverse means. The amount of processed data is based on application, but typical experimental data size are 512 x 512 x 256 voxels, 2 Bytes per voxel. A resolution in x, y and z axes is generally different, but in many applications, especially in medical applications, the resolution in x and y axes is the same, see Fig.1.1.

[1] This work was supported by the Ministry of Education of the Czech Republic - project VS 97155, Academy of Sciences of the Czech Republic project - A2030801
Department of Computer Science, University of West Bohemia, Univerzitní 8, Box 314, 306 14 Plzeň, Czech Republic, e-mail: **skala@kiv.zcu.cz** **http://iason.zcu.cz/~skala**

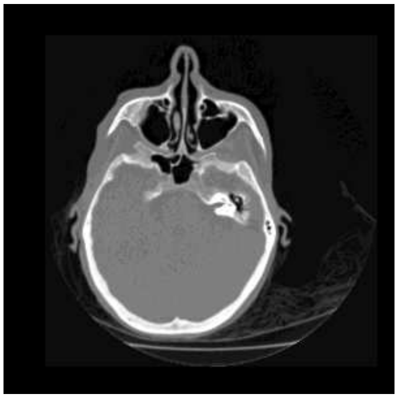

FIG. 1.1 *Typical CT scan for x-y plane*

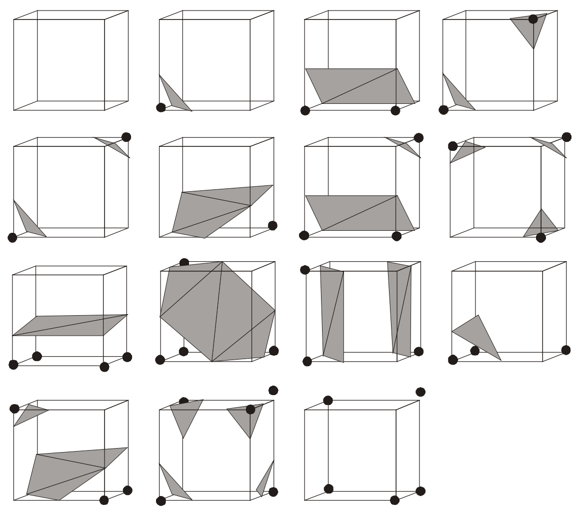

FIG. 2.1. Fundamental cases in MC method

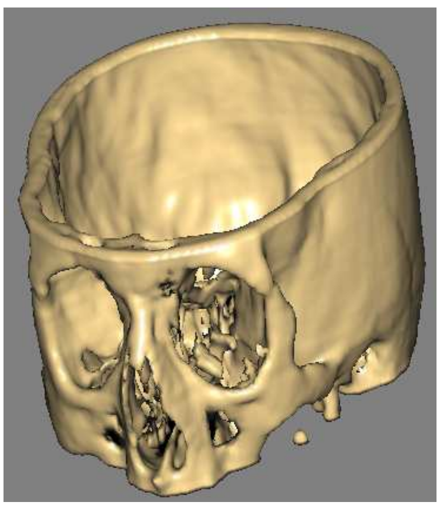

FIG. 2.2 *MC method - 0.65[s], P233MHz 93 395 vertices, 185 535 polygons*

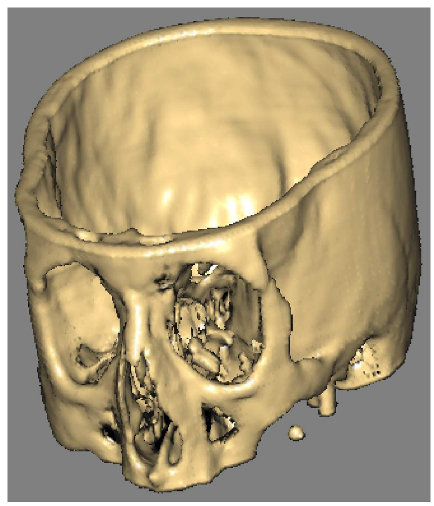

FIG. 2.3 *MT5 method - 1.19[s], P233MHz 230 015 vertices, 460 068 polygons*

Volume data visualization algorithms can be divided into two main groups:

- Direct Volume Rendering (**DVR**) algorithms map elements directly to the image space. A fundamental disadvantage of DVR method is that the entire data set must be traversed each time the image is to be rendered.
- Surface-Fitting (**SF**) algorithms use surface primitives such as polygons (typically triangles) or patches to represent the iso-surface.

Generally the SF methods are usually faster than DVR methods as SF methods traverse the volume only once to extract surfaces. After extracting the surfaces, rendering hardware and rendering methods can be used to fast render the selected iso-surface. It is necessary to point out that changing the SF threshold value means all the cells to be processed again to extract an iso-surface.

A very important property of the iso-surface extraction is the precision of the found surface or volume as it is not equivalent to the human perception of the “quality” of final picture ".

2. Marching Cubes and tetrahedra. Marching Cubes (MC) method is based on voxel vertices classification. It is known that there are 256 possible cases but they can be reduced to 15 fundamental (enlarged vertex means that the voxel value is above a threshold), see Fig.2.1.

The advantage of this method is that it generates generally less triangles than Marching Tetrahedra method, see Fig.2.2 and Fig. 2.3 for time and memory comparisons.

Marching Tetrahedra (**MT**) method is based on a different approach based on a decomposition of the voxel space to tetrahedronal space. It gives generally only two fundamental cases as the tetrahedron can be intersected by an iso-surface in 3 or 4 points. It means that the iso-surface can be represented by two triangles, generally. The advantage of this approach is that the final image is generated principally without "holes" that might happen in MC algorithms.

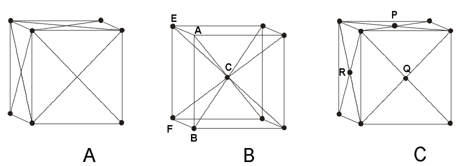

FIG. 2.4

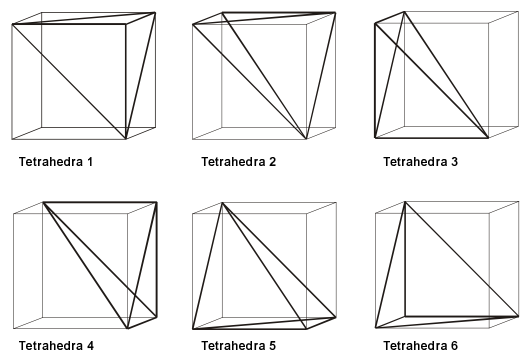

FIG. 2.5 *Possible decomposition to 6 tetrahedra*

Generally three cubic lattices are used, see fig.2.4, and following fundamental cases can be distinguished: cubic primitive (A), body-centred cubic (B), face-centred cubic (C). It is well known that five tetrahedrons is the smallest number how a cube can be partitioned. There are several conform cube decomposition, for details see [Kol94a].

There are two fundamental decomposition:

- 5 "corner" tetrahedra scheme gives 4 identical tetrahedra and one different - centred. It is twice as big as the corner tetrahedron,
- 6 tetrahedra scheme gives the advantage of common faces for tetrahedra of neighbour voxels, see fig.2.5. This is very important feature for the iso-surface generation.

If we use body-centred and face-centred cubic lattices, we can get 6, 12, 24, 36, 48 tetrahedrons per voxel. This approach generates more triangles, so it has very high computational and memory requirements. The result of experiments proved that error of iso-surface extraction is not related to the picture perception, see Fig.2.8 [Bax97a]. It can be seen that the 24 tetrahedron schemes gives better result in some details than 6 tetrahedra scheme, but computational time and memory requirements are unacceptable. There are methods based on body-centred cubic lattice that construct tetrahedrons using two neighbouring voxels [Chan98a], see Fig.2.7. It can be expected that different decomposition schemes give different properties of the final iso-surface.

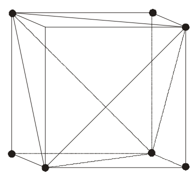

FIG. 2.6 *Classical 5/6 tetrahedra scheme*

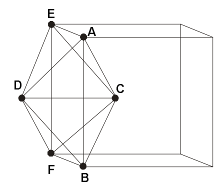

FIG. 2.7 *New 12 tetrahedra per voxel scheme*

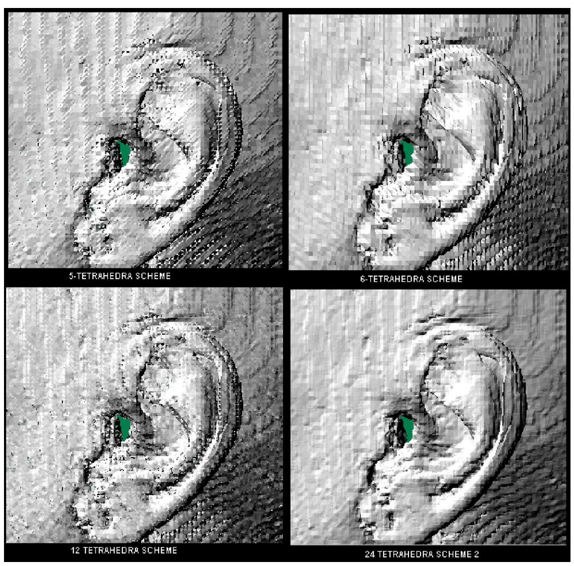

FIG. 2.8 *Comparison of 5, 6, 12, 24 decomposition on ear detail*

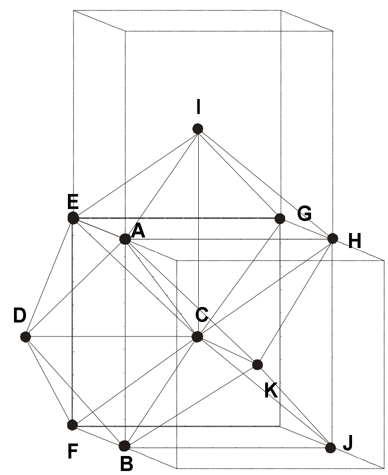

FIG. 2.9 *Tetrahedra that must be inspected when one voxel is evaluated*

This new scheme has an advantage that all tetrahedra are the same (edges are a, $a\sqrt{3}$, a, $a\sqrt{3}$ if $2a$ is the length of the voxel's edge). Nevertheless it is necessary to consider more neighbours for iso-surface extraction than in the body-centred tetrahedronal scheme.

3. Experimental results. The properties of the standard 12 - 48 tetrahedra schemes are known, see [Bax97a]. Therefore the classical 6 tetrahedra and new 12 tetrahedra schemes have been compared according to time, number of generated triangles, radius, area (surface) and volume errors, for details see [Bru99a]. Also perception of image comparison has been tested. It is necessary to point out that error measurement in some cases is not standardised or defined, so this measurement has been defined for the purpose of the comparative study, [Bru99a].

An algorithm with decomposition to 6 tetrahedra will be referenced as the First algorithm and tetrahedral decomposition of space from the centred cubic with 12 tetrahedra lattice will be referenced as the Second algorithm.

An important aspect when comparing two methods is the time for generating the iso-surface and number of generated triangles. Table 4.1 presents some experimental results.

Image		Time [s]		Number of triangles	
Name	Size	First alg.	Second alg.	First alg.	Second alg.
Syn_64	64x64x64	2	4	171276	269184
Engine	256x256x110	53	159	5744340	8333868
3dhead	256x256x109	104	215	12713532	17652630

TABLE 4.1 *Iso-surface extraction time from volumetric data and number of triangles extracted*

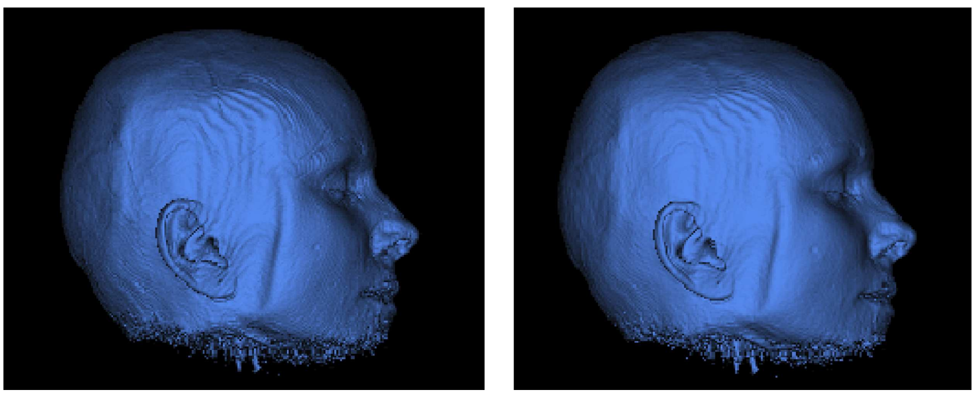

FIG. 4.1 *3dhead image, images are generated by the First and the Second algorithms*

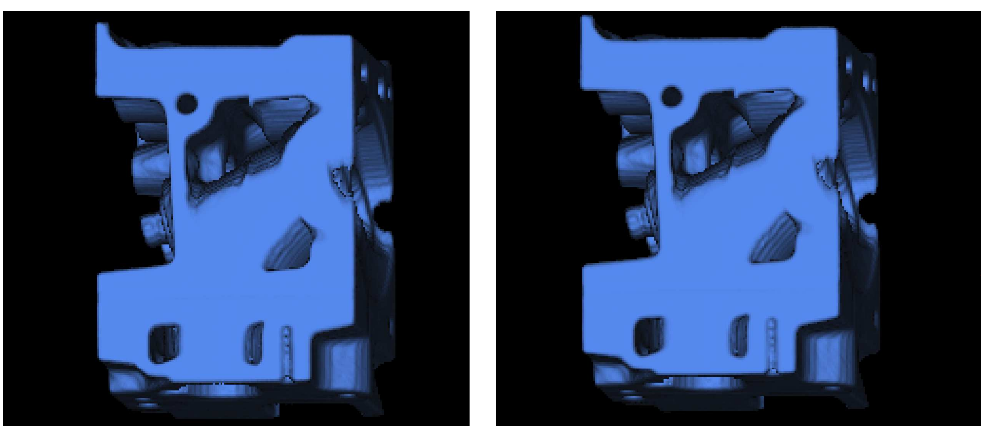

FIG. 4.2 *Engine image, images are generated by the First and the Second algorithms*

Generally it has been proved that the second algorithm gives better visual perception in spite of the fact that all measurements, e.g. radius error from centre of gravity, area error, volume error proved that the First algorithm is more precise. Detailed description of all experiments can be found in [Bru99a]. Nevertheless it is necessary to point out that the Second algorithm is more complex and difficult to speed it up meanwhile the First algorithm is easy to implement with a high efficiency of the iso-surface extraction.

4. Shading models. In the computer graphics applications polygonal mesh shading algorithms are used, especially the constant, Gouraud and Phong algorithms [Fole92a], [Gour71a], [Phon75a], that have been developed recently. Nevertheless only the first two are mostly used because they are also supported by the OpenGL standard. In the constant shading model all the triangles are filled with a constant colour according to the triangle normal. The Gouraud model gives better results, because for each vertex normal a colour is computed and then interpolated over the triangle. Phong model relies on the normal interpolation over the triangle and colour is computed for each pixel according to the interpolated normal. All these methods use the local illumination method, because the Phong model is computationally expensive and OpenGL does not generally support it.

There are some variations of the Gouraud and Phong shading models that mostly use a different vertex normal computation taking into consideration different properties of the triangular mesh, e.g. the area of triangle or the angle of the triangle wedge which is incident with the vertex [Thur98a]. There are also some attempts how to speed up the Phong shading by using different schemes for the normal computation [Ouya00a].

There are some applications, like radiosity methods, that produce a triangular mesh with the known colour in a representative point of this triangle, usually the centre of gravity, and we need smooth shading. **The question is how this type of triangular mesh should be displayed?**

Let us consider a "synthetic" scene that consists of green and red triangles. The use of the constant shading model is natural, see fig.4.3. Nevertheless this approach does not give smooth transition of colours or the smoothness of the surface.

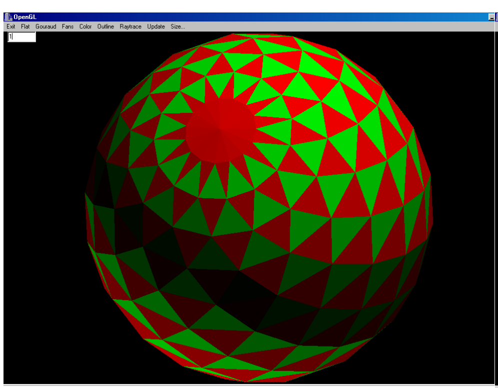

FIG. 4.3 *Triangular mesh - constant shading*

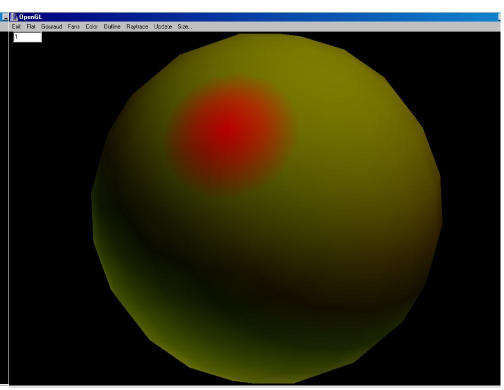

FIG. 4.4 *Triangular mesh - Gouraud shading*

There is a possibility to use the Gouraud shading model, of course, but there is a disadvantage if colours are used. It can be seen that the colour in all vertices of the triangular mesh (the poles are excluded) is to be "an average colour in the vertex", i.e. in our case the yellow colour as Red + Green = Yellow. Also the shape perception can be misleading.

This is a very critical point: we want to display a picture that is nice but generally the colours that have been computed (sometimes using a very intensive computations) **are not in the picture at all**. In this case we obtain nice result, see fig.4.4, but it is generally incorrect because all pixels are set to dark yellow (excluding the part with poles) and no red or green pixels occur at the picture.

In this sense we cannot speak about realistic rendering, of course. Unfortunately, the Phong shading does not solve this problem either. We cannot compute the colour to be displayed according to the local illumination model as the colour was determined by another principle, like radiosity etc. There are well known problems with interpolated shading, like:

- **polygonal silhouette** - the silhouette edge of the mesh is clearly visible as polygonal,
- **perspective distortion** - interpolation is performed after perspective transformation,
- **orientation dependency** - shading depends on how the triangle is situated in 2D plane - rotated triangle is shaded differently,
- **shared vertices problems** - discontinuities may occur if a vertex lies on an edge, fig.4.5,
- **vertex normal does not represent adequately** the surface.

5. S-shading algorithm. The proposed Smooth shading (S-shading) method is based on the following presumptions:

- Each triangle has a representative point in which the colour is determined differently from the local illumination model
- Colours in the representative points must be displayed in the final picture
- Triangles are given by their vertices and the neighbours of all triangles are known
- The smooth shading is needed
- The algorithm must use the standard OpenGL operations and acceleration as much as possible.

These requirements are very crutial according to recently stated properties of the Gouraud and Phong shading models. It is possible to connect the representative points of triangles to a different mesh, which is in some sense "dual", so that each polygon contains the only one vertex of the original mesh, see fig.5.1.

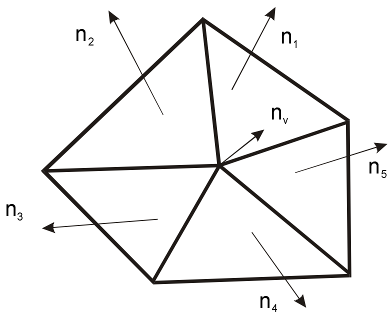

FIG. 4.5 *Vertex normal computation*

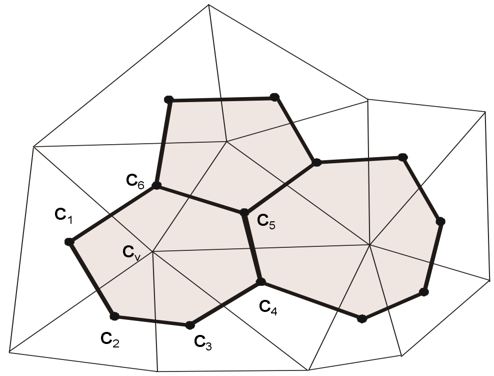

FIG. 5.1 *Dual representation of triangular mesh by fans*

The colours c_1 - c_6 must be displayed as they have been computed or determined somehow recently. It is necessary to fill in such non-planar polygons smoothly so that all vertices of the polygon will have the required colour in vertices. It is a problem a little bit due to the fact that filling is made actually in 2D after rotation and perspective transformation. Nevertheless if the surface is smooth, without sharp edges, it is reasonable to make a similar step as is usually made in the vertex normal computation and somehow estimate a colour c_v in the vertex from the colours c_1 - c_6. The simplest way is to compute (similarly as in Gouraud shading) just and an average colour (if the RGB colour system is used) as

$$c_v = \frac{1}{k}\sum_{i=1}^{k} c_i$$

Modifications like influence of the triangle surfaces or triangle angle in a vertex etc. can be applied similarly as in the Gouraud or Phong shadings. Now the polygon can be shaded as a FAN using GL_TRIANLE_FAN mode by the standard Gouraud shading under OpenGL mode, see fig.5.2.

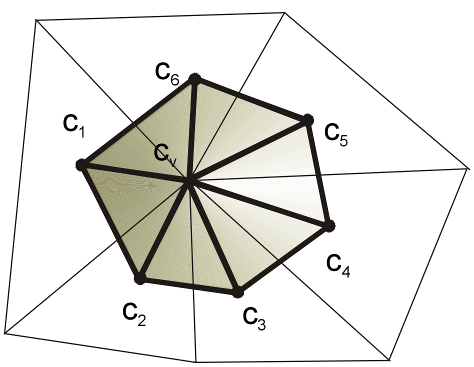

FIG. 5.2 *Shading a FAN*

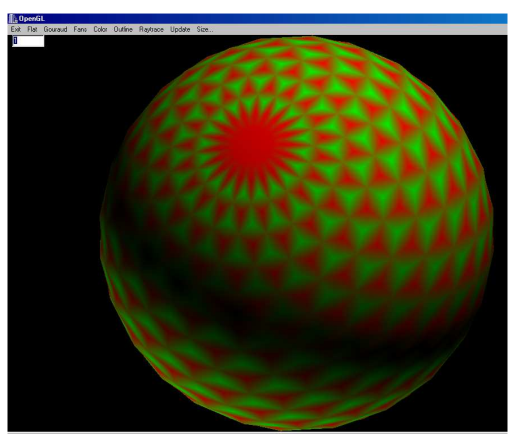

FIG. 5.3 *S-shaded triangular mesh - colour case*

This approach leads to some inefficiency, as we need to draw three times more triangles than before when we draw triangles of the original mesh in GL_TRIANGLE mode; because now each triangle is actually split into three triangles. The drawing of triangular mesh can be significantly speeded up by using the GL_TRIANGLE_STRIP mode, but finding of optimal strips is a quite difficult problem.

It is generally true that approximately three times more triangles needs to be drawn but due to the GL_TRIANLE_FAN mode the actual speed of drawing is reasonably high. It is necessary to display *n* triangles in the original picture. If the GL_TRIANGLE mode is used it is necessary to process *3n* vertices while in the GL_TRIANLE_FAN mode it is necessary to process *n+1* vertices for the *n*-sided fan. It means that it is necessary to process *3*(n+1)* vertices and it means that the expected relative speed ν of drawing can be estimated as

$$\nu = \frac{n}{n+1}$$

It is well known that there are about 6 triangles that share a vertex in average case in regular triangular mesh. Therefore it can be expected that it can be reached about 85,7% of the original drawing speed. This model enables to respect the sharp edges, too [Ska00a].

6. Experimental results. The proposed S-shading algorithm was used to display simple and complex surfaces after triangular mesh reduction. The results have proved that the proposed S-shading algorithm is applicable and that the S-shading algorithm gives a chance to see the actual surface of the object while the Gouraud shading removes many artefacts, in some cases very important. This might be even dangerous, especially in visualization of physical phenomena as the expert must interpret obtained results correctly. A mesh after mesh reduction is used to show the differences on rough meshes.

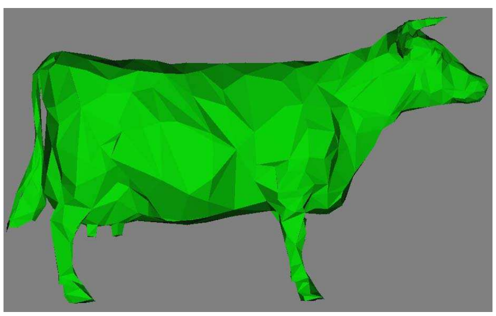

FIG. 7.1 *Constant shading - model of a cow*

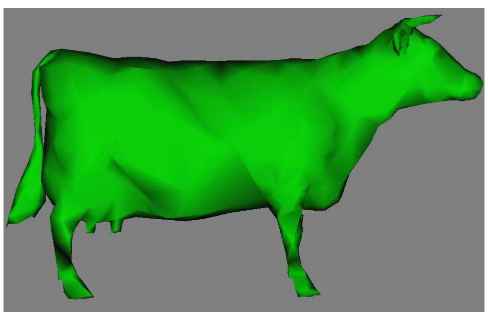

FIG. 7.2 *Gouraud shading - model of a cow*

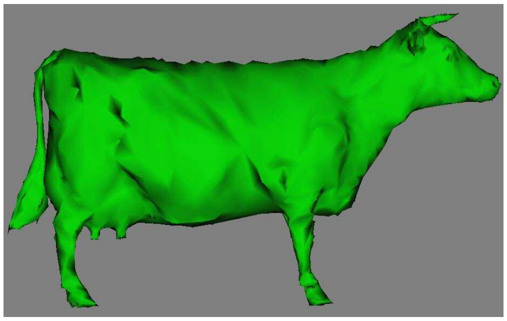

FIG. 7.3 *S-shading - model of a cow*

Fig.7.1 presents an object with the constant shading, while Fig.7.2 presents results of the Gouraud method. Fig.7.3 shows the same object if the proposed S-shading method is used. The differences between Gouraud and S-shading can be seen on fig.7.2 and fig.7.3 that present a model of a cow after triangular mesh decimation. It can be seen that the S-shading gives a smooth surface but also clearly shows the parts where triangular mesh after decimation is rough with some artefacts.

7. Conclusion. A comparison of the MT algorithm with the decomposition into six tetrahedra and MT with a tetrahedral decomposition of the space from the cubic lattice has been tested. When is the time of generation or number of triangles measure of quality, the decomposition to 6 tetrahedra gives better results. Distance error from each vertex of the triangles, distance error from the centre of gravity, area and volume error were studied. The visual impression is better when a complex model is generated with the Second algorithm with the decomposition using cubic lattice. These results are similar in comparison with [Bax97a], where five and six tetrahedra decomposition were compared with 24 and 48. A similar explanation can be given about the better image results with the second decomposition: **human eye is sensitive to edges and gradient changes of a surface**. Also a new S-shading algorithm was used to display some features that are not visible if Gouraud shading model is used and it:

- enables to display surfaces with respect to the precision,
- benefits from the hardware acceleration under the OpenGL graphical interface,
- proved the advantage for applications where the final shape is preferred to the "nice" picture but the smoothness of the final picture is required.

Many published papers that deal with triangular mesh reduction based on several approaches with nice pictures demonstrate an efficiency of the algorithm using shaded pictures. Fig.7.2 and fig.7.3 show that it is very difficult to evaluate quality of the given mesh as the people's perception strongly depends on the shading model. It is necessary to note that the S-shading method could bring some artefacts when triangular mesh with long thin triangles is to be displayed.

It is necessary to point out that today's Intel based systems with memory up-to 2 GB RAM, up to 1GHz processor (Pentium III & Athlon) and development of 4,5 GHz processor has already been announced. Also the high performance graphics cards capable to display more than 10^6 triangles per second are affordable. It moves some recently considered problems away and others become more important.

8. Acknowledgment. The author would like to thanks to all who contributed to this work, especially to MSc. and PhD. students at the University of West Bohemia in Plzen who have stimulated this thoughts and development of this new algorithm. This paper benefits from several discussions with them a lot. Special thanks belong to P.Baxa, M.Benda, A.Brusi, M.Franc and P.Sládek for careful reference implementations, to anonymous reviewers of this paper as they shared some valuable insights on this problem solution. Theirs invaluable critical comments and suggestions improved the manuscript significantly.

REFERENCES

[Bar99a] Bartoš,P.(supervisor Skala,V.): Volumetric Data and Surface Models (in Czech), MSc. Thesis, Univ.of West Bohemia, Plzen, Czech Republic, 1999.

[Bax97a] Baxa,P., Skala,V., Moucek,R.: Error Estimation for Iso-surfaces, In Proceedings of COMPUGRAPHICS '97, 1997.

[Ben96a] Bentum,M.J., Barthold,B., Lichtenbelt,A., Malzbender,T.: Frequency Analysis of Gradient Estimators in Volume Rendering, IEEE Transactions on Visualization and Computer Graphics, Vol.2, N.3, 1996.

[Bru99a] Brusi,A.(supervisor Skala,V.): Comparison of tetrahedron Subdivision Schemes of Volumetric Data, MSc.Thesis, Univ.of West Bohemia, Plzen, Czech Republic, 1999.

[Cha98a] Cahn,S.L., Purisima,E.O.: A new Tetrahedral Tessellation scheme for Isosurface Generation", Computer and Graphics, Vol. 22, No 1, pp. 83-90, 1998.

[Chu95a] Chuang,J.-H., Lee,W.Ch.: Efficient Generation of Isosurfaces in Volume Rendering, Computer & Graphics, Vol. 19, No 6, pp 805-813, 1995.

[Fole92a] Foley,J.D., van Dam,A., Feiner,S.K., Huges,J.F.: Computer graphics - Principles and practice, Addison - Wesley Publ. Comp., 1992

[Gour71a] Gouraud,H.: Continuous shading of curved surfaces, IEEE Trans.on Computers, Vol.20, pp.623-629, 1971.

[Kol94a] Kolcun,A.: Conform decomposition of cube, SSCG'94 proceedings, Comenius University, Slovakia, 1994.

[Ouya00a] Ouyang,S., Maynard,D.E.: Phong shading by binary interpolation, Computers&Graphics, Vol.20, No.6, pp.839-848

[Phon75a] Phong,B.T.: Illumination for computer generated pictures, CACM, Vo.18, pp.311-317, 1975.

[Ska00a] Skala,V.: S-Shading Algorithm - A new Smooth Shading Algorithm for Triangular Meshes, Technical report, Univ.of West Bohemia, Plzen, 2000

[Thur98a] Thürmer,G., Wüthrich,Ch.A.: Computing vertex normals from polygonal facets, J.of Graphics Tools, Vol.3., No.1, pp. 43-46, 1998.

[Tod92a] Todd,T.E.: A Survey of Algorithms for Volume Visualization, SDSC Labs., 1992.