Proceedings of EDU+COMPUGRAPHICS'97, Algarve, Portugal, pp. 202-211, 1997.

AN ERROR ESTIMATION FOR ISOSURFACES

Pavel Baxa, Vaclav Skala, Roman Mouček[1]
{baxa, skala}@kiv.zcu.cz, moucek@hermes.zcu.cz

ABSTRACT

A new approach to the error estimation in isosurface construction using different tetrahedronization schemes in volume visualization will be presented. A special attention is devoted to small objects of voxel size. Error estimation is made for existing 5, 6 and 12 tetrahedra per a cube compared with new 12, 24 and 48 tetrahedra schemes. The theoretical error estimation with experimental results on CT and MRI images will be shown. The proposed approach brings better visual results of the final images and is convenient for parallel processing and hardware implementation.

Keywords: computer graphics, volume visualization, surface fitting, isosurface construction, tetrahedronization, imaging, CT, MRI.

1. INTRODUCTION

The volume visualization algorithms can be divided into two basic categories: Direct volume rendering and surface fitting.

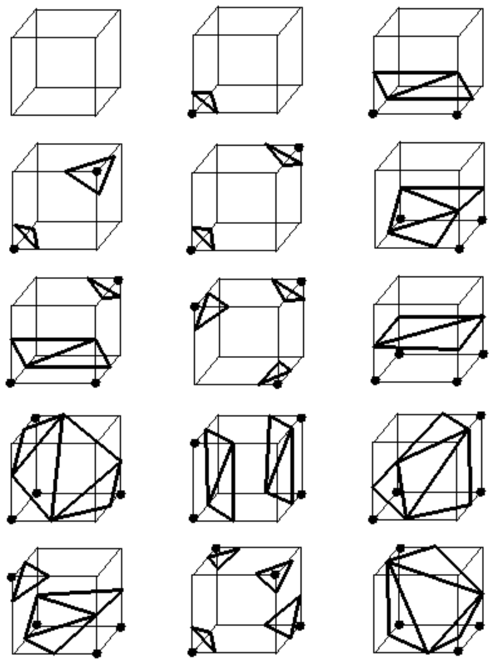
Direct volume rendering methods are characterized by direct mapping the scene elements to the display unit. The most commonly used algorithm belonging to this category is *ray casting*. It is based on casting a ray through each pixel of a 2D display device to the voxel scene. As the ray travels through the scene, it intersects a sequence of cells, each having defined density (or opacity), until it leaves the scene or until the sum of the densities reaches some limit value. The sum of densities defines the final color of the corresponding pixel. The main disadvantage of the direct volume rendering methods is that each time the image is to be rendered, the entire dataset must be traversed [Yagel92].

Surface fitting algorithms make use of a completely different idea. With a common assumption that all the voxels inside an object have greater value than those outside, the surface boundary of the object can be defined by an isosurface. An *isosurface* is a surface associated with a constant threshold value which separates voxels with values greater than or equal to the threshold from voxels with values less than it.

Marching cubes (MC) presented by [Lorensen-Cline87] is an example of a surface fitting algorithm. The basic notion is that for each eight neighbouring voxels in a dataset we define a cubic cell, vertices of which are represented by the eight voxels. If one or more voxels of the cell have values less than the user-defined threshold, and one or more of them have values greater than this value, we know the cell must contribute some component of the isosurface. There exist 256 different arrangements of voxels, since each of the eight cell vertices can lie on both sides of the isosurface. Due to geometrical symmetries (mirroring, rotations), this amount can be reduced to 15 basic configurations, see Fig. 1.

[1] This work was supported by The Ministry of Education of the Czech Republic - projects PG 97193, PG 97185 and VS 97155

By determining which edges of the cube cell are intersected by the isosurface, we can create triangular patches which divide the cube into regions within the isosurface and regions outside. By connecting the patches from all cubes on the isosurface boundary, we get a surface representation.



.

Marching cubes configurations

Fig. 1

One obvious problem with marching cubes is ambiguity in creating triangle patches in case when four or more intersection points are to create two or more neighbouring triangles. There is also a possibility of gaps in the triangle-represented isosurface.

Marching tetrahedra (MT) algorithm presented in [Doi-Koide91] is a modification of the marching cubes. The basic idea is that each cell is subdivided into several tetrahedra first, and then the intersections of the isosurface and tetrahedra edges are computed. This partialy solves the ambiguity problems, because maximum number of the intersections between the isosurface and tetrahedra edges is four, that is, at most two triangles will be created.

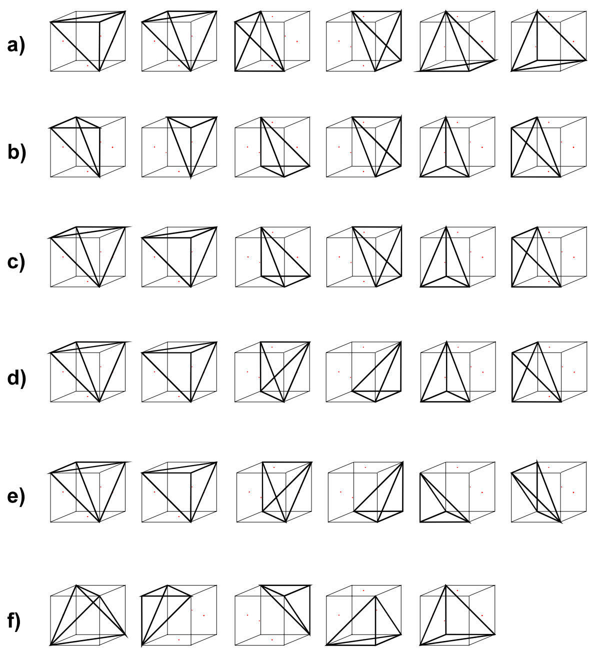
One of the most important features that any visualization algorithm must have is accuracy. The visualization process should represent the original data without any loss of information. If the vizualization algorithm uses any kind of approximation, the error of such approximation has to be minimized. This is extremely important when medical data are to be visualized. We decided to determine how the approximations used in MT algorithm affect the accuracy of vizualization, and find a way how to minimize the overall error of the results. The error is evaluated as a volume difference between the objects and their representation by MT algorithm. A special attention is devoted to small objects of voxel size.

The next section of the paper describes several commonly used tetrahedronization schemes and presents some new schemes, which were supposed to give good results. In section 3, the theoretical error estimation is shown in two-dimensional case for simplicity. The idea is then extended to 3D space in Section 4. Finally, the estimated values are compared with experimentally achieved errors and the test results are presented.

2. TETRAHEDRONIZATION METHODS

One of the essential elements of the MT algorithm is a tetrahedronization scheme, i.e. a way how the cell is subdivided into tetrahedra. The common schemes [Kolcun94] divide the cell into five or six tetrahedra, see Fig. 2.

Proceedings of EDU+COMPUGRAPHICS'97, Algarve, Portugal, pp. 202-211, 1997.



Commonly used tetrahedronization schemes

Fig. 2

The commonly used 5-6 tetrahedra schemes have several disadvantages. One of them is that the tetrahedra are uneven, mutualy different and asymmetric, result of which is that the generated boundary representation is not visually acceptable.

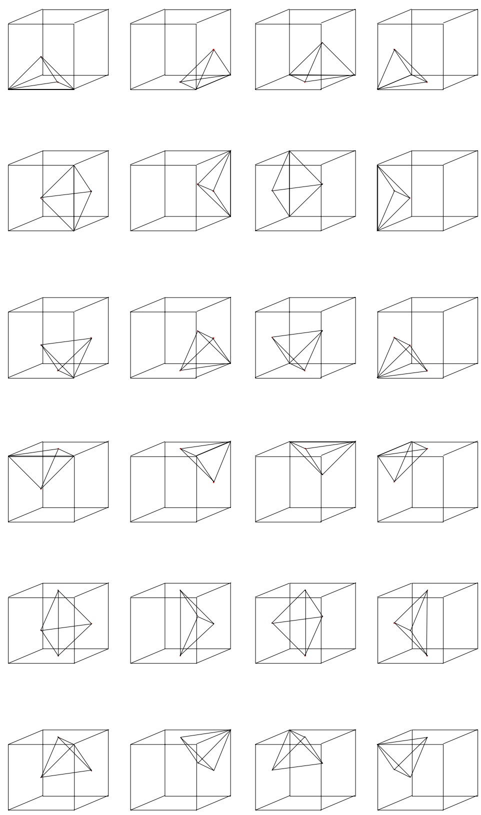
We propose three tetrahedronization schemes, that fulfil the following requirements:

- The subdivision is symmetrical.
- There is no information loss in comparison with the original voxel scene. This is very important in medicine, where no changes of original data are permitted.
- Tetrahedra are of at most two shapes: the less different shapes the tetrahedra have, the more simple is a possible hardware implementation.

The new subdivision schemes are shown in Fig. 3. In addition to the values in the original eight cell vertices, some new vertices are used. In Fig. 3a, the centers of each cell side are given a value that is an average value of the nearest 4 vertices values. In Fig. 3b, also the value of the cell's center of gravity is computed as an average value from the known 8 vertices as a result of trilinear interpolation in the cell. Subdivision into 48 tetrahedra can be achieved from the scheme in Fig. 3b by adding centers of the cell edges and thus splitting each tetrahedron in two.

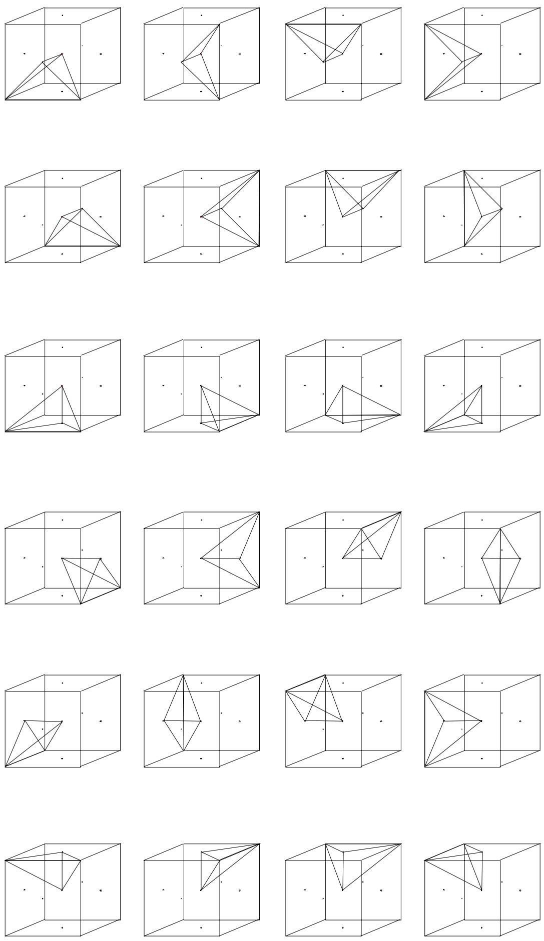
Such interpolations have no influence on the eight known values. This fact is important especially when this approach is used for medical data visualization purposes. Since we have no information about values between the cell vertices, linear interpolations seems suitable enough. Thus, instead of five or six tetrahedra schemes used in the original algorithm, we use 24 and 48-tetrahedral subdivision.

Two schemes of subdivision into



24 tetrahedra

Fig. 3a



Two schemes of subdivision into 24 tetrahedra

Fig. 3b

3. ERROR ESTIMATION IN 2D

Since the problem of error estimation in three-dimensional space is extremely difficult to present in a simple, comprehensive way, we decided to reduce it to 2D space, so that it can be easily understood from the drawings.

First, it is necessary to define assumptions needed for the estimation. Let the visualized o*bject* be defined as a circle with radius *r*. The *scene* is a continuous function *h* defined as a distance from the circle boundary.

$$h(x,y) = \sqrt{x^2 + y^2} - r$$

The *discretized scene* is a matrix of points having coordinates *i,j* and value $h_{i,j}$,

$$h_{ij} = \sqrt{i^2 + j^2} - r$$

The circle center has coordinates [0,0]. The MT algorithm[2] is used to find the boundary of the circle, i.e. isosurface with $h=0$. Then, the real circle surface area $S_R=\pi r^2$ is compared with the surface area S_D of the approximating polygon, where the relative error *e* is evaluated as

$$e = \frac{S_D - S_R}{S_R}$$

The most simple case to explain can be seen in Fig. 4a. The circle radius is less than 1, an object of one-voxel size is approximated. The MT algorithm uses a simple triangularization scheme where each 4-voxel cell is divided by one of its diagonals into two triangles. The aim is to find a point x_p such that $h(x_p, 0) = 0$. Using linear interpolation on the edge h_{00}-h_{10}, the value of x_p is computed as follows:

$$x_p = \frac{h_{00}}{h_{00} - h_{10}} = r$$

Now, it is possible to compute the surface area of both the real body (circle) and the approximated one. Also we can estimate an error *e*.

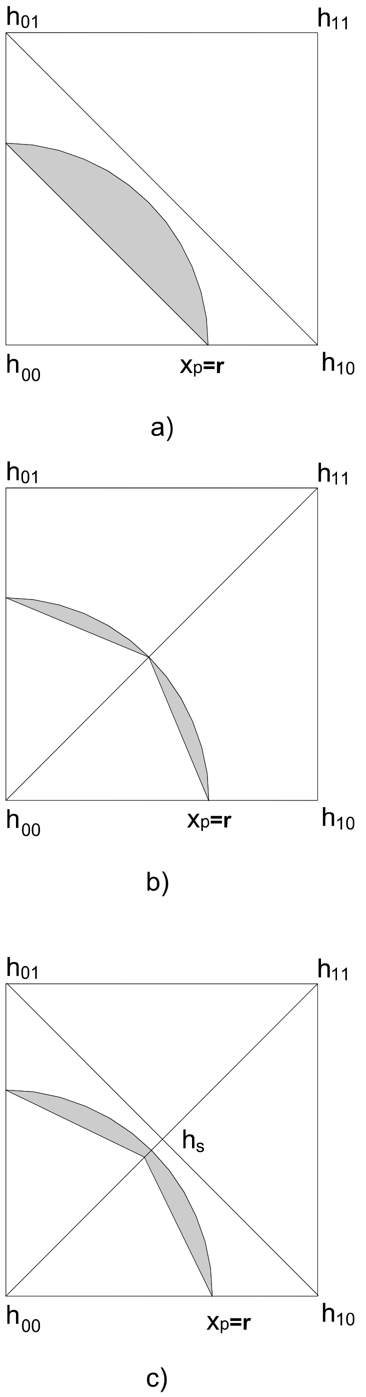
$$S_R = \pi r^2$$
$$S_D = 2r^2$$
$$e = \frac{S_D - S_R}{S_R} = \frac{2-\pi}{\pi} \approx -0.36$$

Fig. 4b shows similar situation, but the other diagonal is used for division this time. It is necessary to find a point u_p on the diagonal that approximates the intersection point between the circle and the diagonal, while x_p stays unchanged.

[2] MT algorithm is reduced to 2D case. A cube divided into tetrahedra is replaced with a square divided into triangles.

The *x*- and and *y*-coordinate of u_p is computed using linear interpolation on the edge h_{00}-h_{11}:

$$u_{px} = \frac{h_{00}}{h_{00} - h_{11}} = \frac{r}{\sqrt{2}}$$



Single-voxel sized object

Fig. 4

It is obvious that u_p lies on the circle boundary. The error in this case is

$$e = \frac{2\sqrt{2} - \pi}{\pi} \approx -0.1$$

Thus, the value of the error alternates between 10% and 36%, depending on used diagonal. We presumed that such asymmetry can be avoided when some different triangularization scheme is used. In 2D case, this reduces to scheme shown in Fig 4c. Both cell diagonals are used, so that the cell is subdivided into four triangles. New point *C* lies in the center of the cell and its value h_C is computed as mean value of the four voxels.

$$h_C = \frac{2 + \sqrt{2}}{4} - r$$

The coordinates of u_p has changed, since it is computed using linear interpolation on the line segment h_{00}-h_C.

$$u_{px} = \frac{h_{00}}{h_{00} - h_C} = \frac{2r}{2 + \sqrt{2}}$$

$$e = \frac{\frac{8}{2 + \sqrt{2}} - \pi}{\pi} \approx -0.254$$

This error value is less than the previous, still the average error is surprisingly greater than in the previous case.

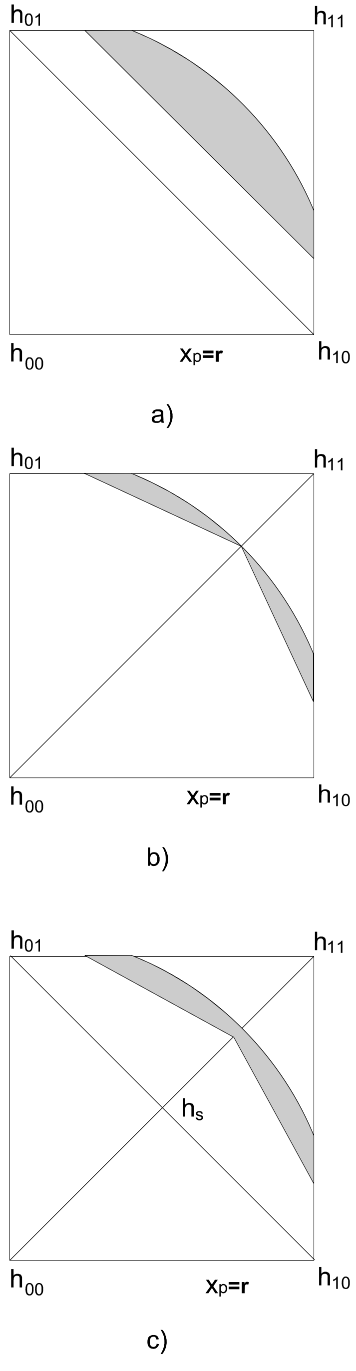
Let us suppose now that the circle radius lie in the interval $(1, \sqrt{2})$. In Fig. 5, the same cell is shown. The circle intersect the cell edges in points x_r, y_r. In this case,

$$x_p = y_p = \frac{r - 1}{\sqrt{2} - 1}$$

$$x_r = y_r = \sqrt{r^2 - 1}$$

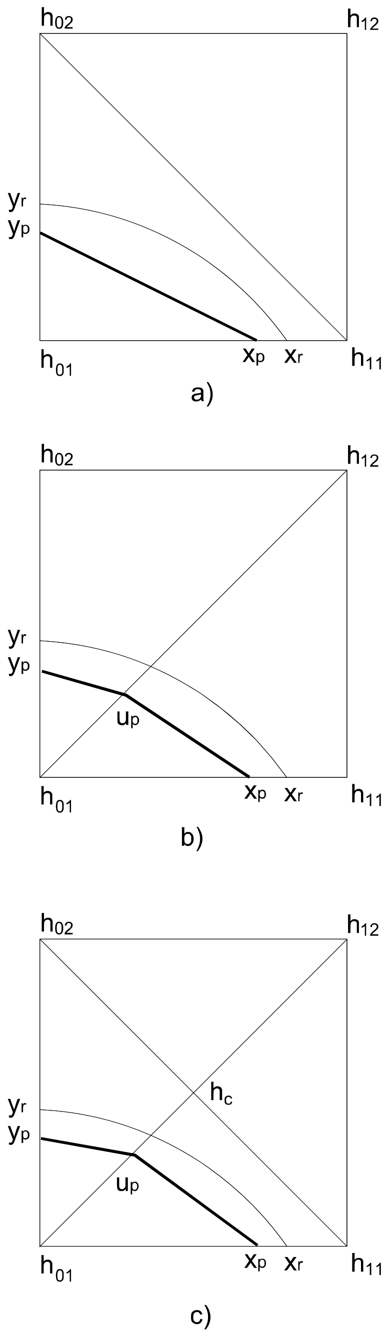
It can be shown that $x_p - x_r < 0$ for any *r* from the interval $(1, \sqrt{2})$. If the interpolation on the diagonal is computed, as in Fig. 5b, the intersection point u_p has obviously coordinates

$$u_p = \left(\frac{r}{\sqrt{2}}, \frac{r}{\sqrt{2}} \right).$$



Situation for r > 1

Fig. 5



Situation in neighbouring cell

Fig. 6

In order to determine the volume of the approximated sphere, also the neighbouring cells must be taken into account, as presented in Fig.6. Situation in Fig. 6a) again shows the most simple case, where the diagonal has no influence on the approximation.

In this case, the values in the four voxel vertices evaluate as follows:

$$h_{01} = 1 - r$$
$$h_{11} = \sqrt{2} - r$$
$$h_{02} = 2 - r$$
$$h_{12} = \sqrt{5} - r$$

It can be shown that

$$x_{r2} = \sqrt{r^2 - 1},\ y_{r2} = r,$$

$$x_{p2} = \frac{r-1}{\sqrt{2}-1},\ y_{p2} = r,$$
$$u_{px2} = \frac{r-1}{\sqrt{5}-1},\ u_{px3} = \frac{2(r-1)}{\sqrt{2}+\sqrt{5}+1}.$$

Even in such simple case that was shown here, the resulting error cannot be easily estimated, since it varies dependingly on the value of *r*. Still, there exist a common rule saying that

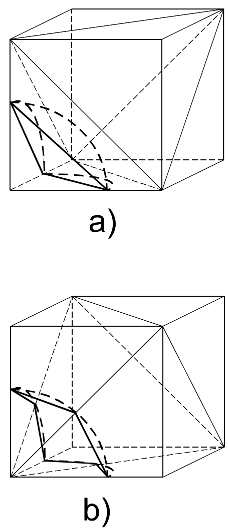
$$S_a < S_c < S_b < S_R.$$

S_a, S_b, S_c represent surface areas of the approximations shown in Fig. 5a, 6a, Fig. 5b, 6b and Fig. 5c, 6c, respectively.

4. 3D CASE

Since the particular situations in three-dimensional space are very difficult to imagine, let us have a look only at the easiest one. The radius of the visualized sphere is $r < \sqrt{2}/2$ and cube division into 5 tetrahedra is used. Since this scheme is assymetrical, both mutualy symmetrical cases have to be considered - see Fig. 7a,b. The schemes consisting of more tetrahedra could not be graphicaly presented here because of their complexity.

It can be proved that the approximation error is 68.2% in case a), 29.7% in case b) and finally 54.4% for selected 24 tetrahedra subdivision.



Simple 3D case for r<√2/2 and 5 tetrahedra subdivision

Fig. 7

5. TEST RESULTS

Regarding the volume error measuring, we generated 200 voxel scenes with sphere radius increasing from 0.0 to 20.0 units by 0.1 and the voxel values represented by both integer and real numbers. For each scene, the real sphere volume was computed, as well as the volume of the approximations. The relative error was computed as

$$e = \frac{S_D - S_R}{S_R}.$$

The results of the experiments are presented in the Chart 1 for integer voxel values and Chart 2 for real values of the scene voxels.

The results of the test tend to the same behaviour that was estimated theoreticaly in the previous sections. Concerning only on the volume accuracy, it proves useless to implement tetrahedronization schemes containing 24 or even 48 tetrahedra. The idea to add new points to the existing eight cell vertices does not seem good enough. The new points bring some error, since the linear interpolations are used to compute their values.

Still, the images generated by MT algorithm are visually obviously better for 24 and 48 schemes, as shown in section 6. The reason of this has to be sought in features of human vision. A human eye is not able to percept volume accuracy, but is sensitiveto edges and changing gradient of the surface. We supposed that some different criterion should explain why better results were achieved with 24 and 48 tetrahedra subdivision.

We implemented a similar test, but this time, we measured the surface area of the sphere and its approximations instead of volume. The results that we achieved can be seen in Chart 3 for integer data and Chart 4 for real voxel values. Now, the schemes having 24 or 48 tetrahedra per one cell provide for real voxel values evidently better results then the commonly used schemes.

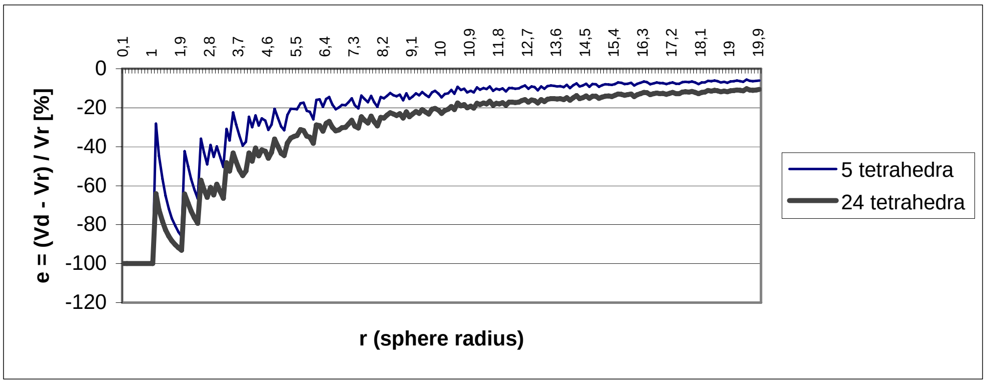


Chart1 - Relative volume error for integer data

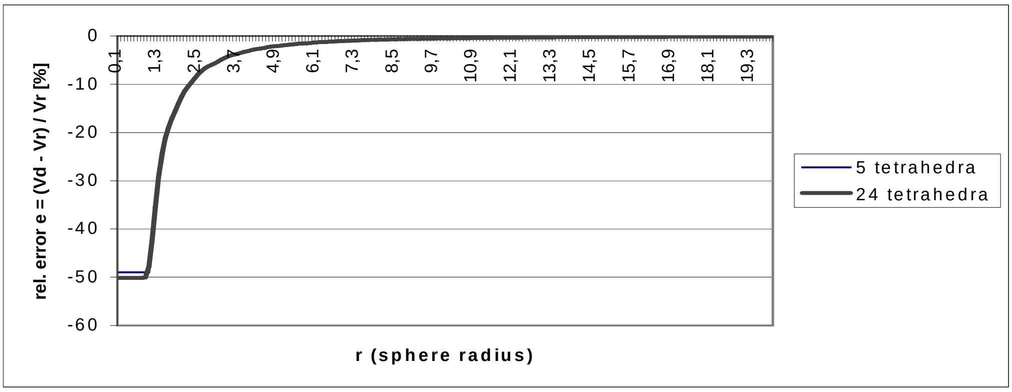


Chart 2 - Relative volume error for real data

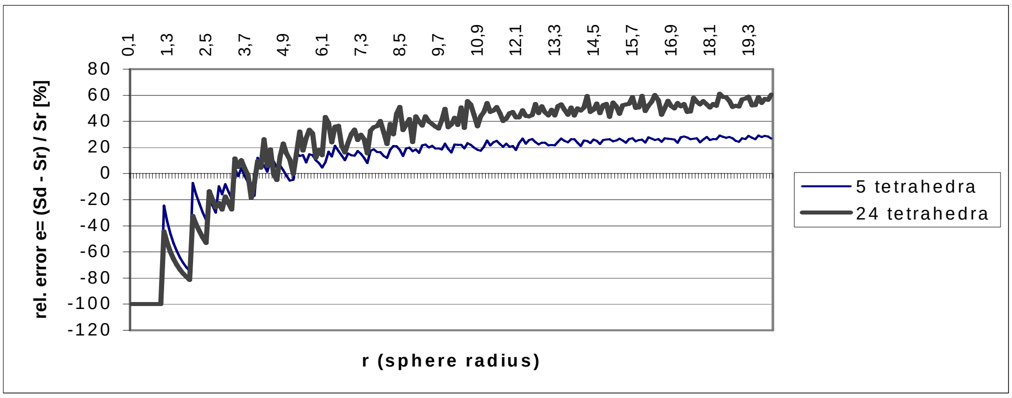


Chart 3 - Relative surface area error for integer data

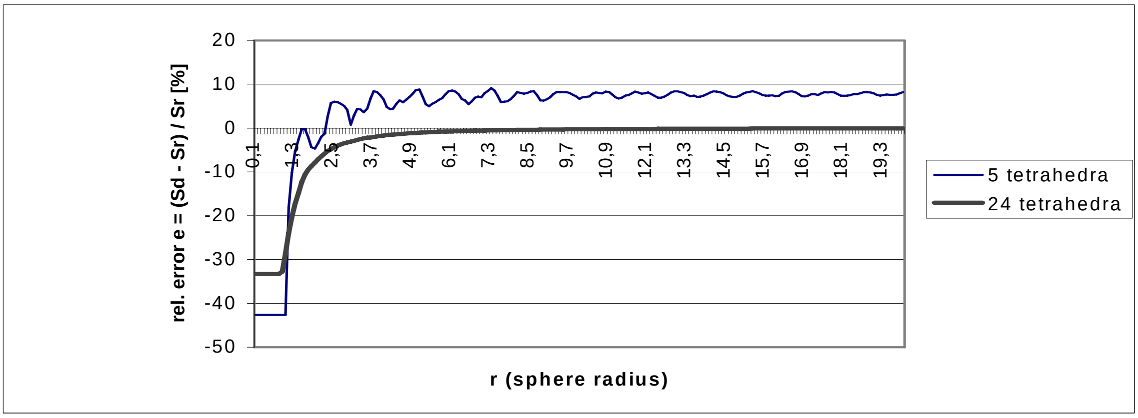
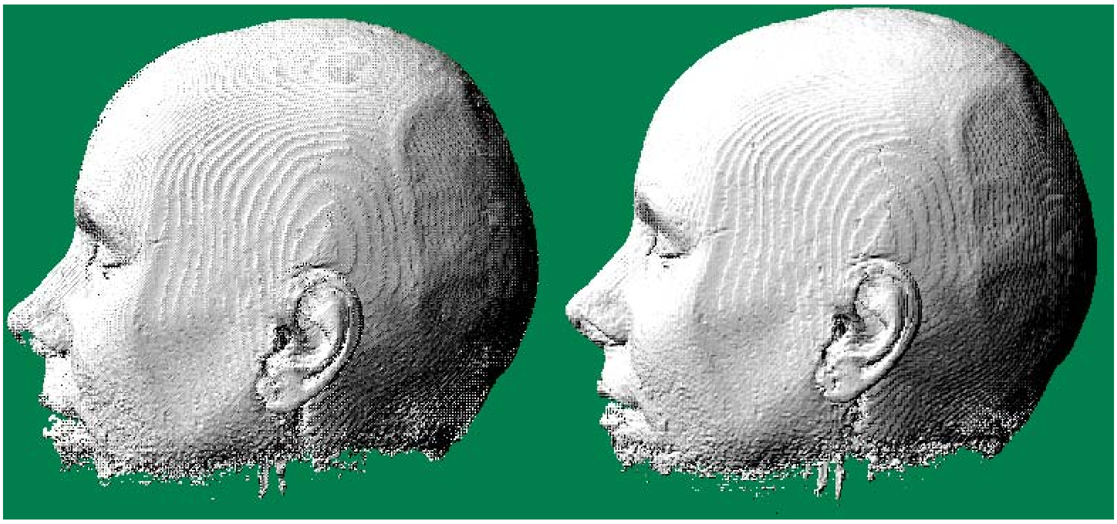


Chart 4 - Relative surface area error for real data

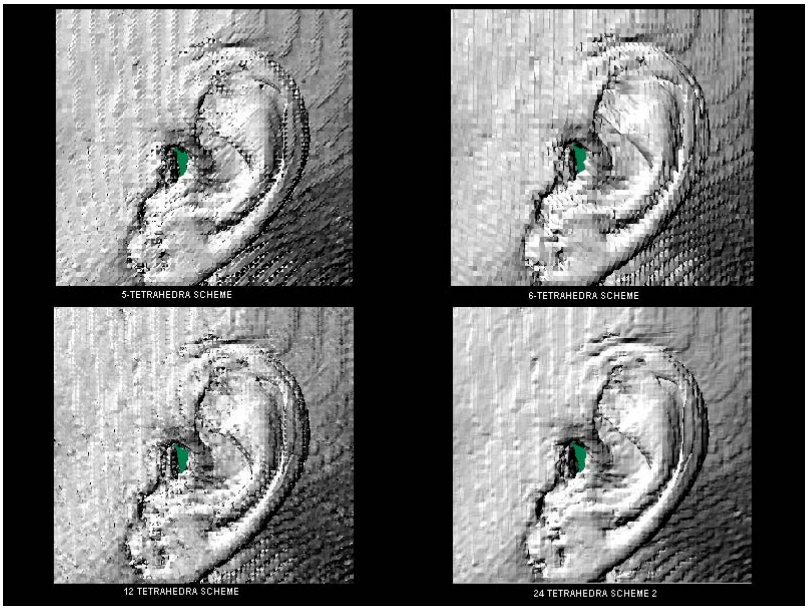
6. RESULTS FOR MRI DATA

The implemented MT algorithm was tested not only for volume and surface area accuracy, but also from the visual point of view. Fig. 8 and Fig. 9 show samples of 256x256x109 MRI scaned head visualized using different tetrahedronization schemes. The used 24 scheme gives results looking significantly better than commonly used schemes, especially when small details are rendered (see nose, ear and lips in Fig. 8 or detailed view of an ear in Fig. 9).



Two views of a MRI cadaver head. The left one was created using a scheme with 6 tetrahedra, the right one by 24-tetrahedra scheme

Fig. 8



Four views of an ear, detail from Fig. 8. The new schemes produce evidently better visual results.

Fig. 9

7. CONCLUSION

We presented a new approach to the estimation of errors in the construction of isosurfaces by surface-fitting algorithms. We proposed several tetrahedronization schemes that make use of added points to achieve better visual results. An error estimation was made for existing 5, 6 and 12 tetrahedra per a cube and compared with new 24 and 48 tetrahedra schemes. As a measure of the error we chose a relative volume error. Both theoretical estimation and experiments proved that it is useless to implement 24 or 48 tetrahedra schemes, when only the volume criterion is taken into account. We also showed that there is a coherence between visual quality of the resulting images and surface area accuracy. In the future work, an adaptive MT algorithm is to be proposed, which will approximate the objects more accurately both in volume and surface area.

8. ACKNOWLEDGEMENTS

Authors would like to thank Ms. Ivana Kolingerová for her suggestions and for making the final corrections to the English text.

9. REFERENCES

Doi A., Koide A., 1991, An efficient method of triangulating equi-valued surface by using tetrahedral cells, *IEICE Trans.*, E74, pp 214-224.

Kolcun A, 1994, Semiregular grids in 2D and 3D, *WSCG'94 Conference Proceedings*, January, pp 20-27.

Lorensen W.E., Cline H.E., 1987, Marching Cubes: a high resolution 3D surface reconstruction algorithm, *Computer Graphics*, Vol. 21, No. 4, July, pp 163-169.

Yagel R., Cohen D., Kaufman A., 1992, Discrete Ray Tracing, *IEEE Copmuter Graphics & Applications*, Vol. 15, No. 5, September, pp 19-28.