On Properties of Contours of Trilinear Scalar Fields

Holger Theisel

Abstract. We study properties of contour surfaces of trilinear scalar fields, and give a classification based on how many unconnected surface parts they consist of. Furthermore, we introduce the concept of the segment number of a voxel. The segment number is a threshold-independent measure which estimates how complicated the contours inside the voxel are expected to be. Finally, we give necessary and sufficient conditions for a voxel to have a segment number of 1. These conditions are applied to analyze a computer tomography data set.

§1. Introduction

Contours (isosurfaces) of trilinear scalar fields are treated in a variety of applications. For instance, the data used in volume visualization usually consists of a number of scalars defined at certain grid points; between the grid points a piecewise trilinear interpolation of the scalar field is applied.

Given a voxel $\boldsymbol{V} = [0,1]^3$, the trilinear scalar field is defined by setting the values $c_{ijk}(i,j,k \in \{0,1\})$ of the field at the corners of $\boldsymbol{V}$. Then the scalar field is defined as

$$\begin{aligned} s(u,v,w) = {} & (1-u)\cdot(1-v)\cdot(1-w)\cdot c_{000} + (1-u)\cdot(1-v)\cdot w\cdot c_{001} \\ & + (1-u)\cdot v\cdot(1-w)\cdot c_{010} + (1-u)\cdot v\cdot w\cdot c_{011} \\ & + u\cdot(1-v)\cdot(1-w)\cdot c_{100} + u\cdot(1-v)\cdot w\cdot c_{101} \\ & + u\cdot v\cdot(1-w)\cdot c_{110} + u\cdot v\cdot w\cdot c_{111}. \end{aligned} \tag{1}$$

Figure 1a illustrates this. A contour of $\boldsymbol{V}$ is defined by $s(u,v,w) = r =$const for a certain threshold r. Figure 1b shows an example of a contour of (1).

There are a number of algorithms to produce a triangular approximation of a contour of (1). Of these, the Marching Cubes (MC) method ([3] and [4])

Curve and Surface Design: Saint-Malo 1999

Pierre-Jean Laurent, Paul Sablonnière, and Larry L. Schumaker (eds.), pp. 1–8.

ISBN 0-8265-1356-5.

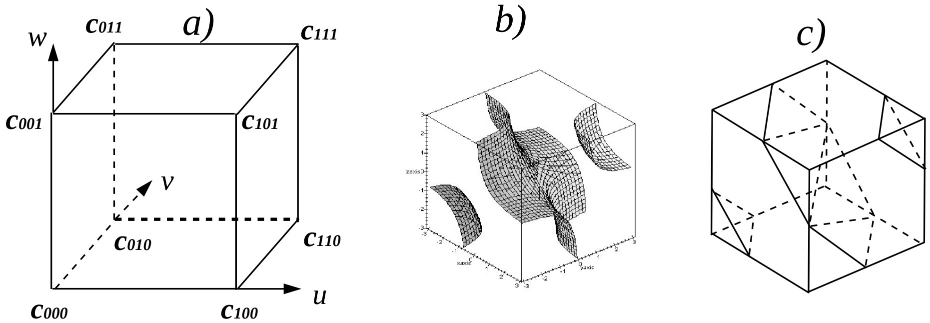

Fig. 1. a) Voxel $\boldsymbol{V}$; b) a contour in $\boldsymbol{V}$; c) result of MC.

is the most popular. Figure 1c shows the resulting triangular approximation of the contour shown in Figure 1b using the Marching Cubes method.

The Marching Cubes algorithm distinguishes several cases where some of them are harder to treat than others. In this paper we introduce a measure of how costly in terms of computing time the MC algorithm inside a certain voxel is expected to be. This characterization of a voxel – called segment number – is independent of a particular threshold. It estimates the costs of a Marching Cubes algorithm for varying thresholds.

As already stated in [2], the contour of (1) is a rational cubic surface. In [2] this surface is approximated by a collection of rational quadratic triangular patches.

Section 2 of this paper studies the contours of (1) in the domain $\mathbb{R}^3$. We give a classification based on how many unconnected surface parts the contours consist of. Sections 3 and 4 focus on contours of (1) inside a certain voxel. Section 3 introduces the concept of segment number as a measure of how simply a voxel can be treated by an MC algorithm. In Section 4, necessary and sufficient geometric conditions for a voxel to have a segment number of 1 are shown. In Section 5, the number of voxels with a segment number of 1 are computed for a real volume data set.

§2. Classification of the Contour in $\mathbb{R}^3$

In this section we consider the contour of (1) not in a particular voxel but in the domain $\mathbb{R}^3$. In general, the contour consists of a number of surface parts which are not connected to each other. Before we classify the contours of (1) by the number of unconnected surface parts, we apply a translation of the coordinate system as shown in Figure 2. Choosing

$$p = c_{001} + c_{010} + c_{100} + c_{111} - c_{000} - c_{011} - c_{101} - c_{110}$$

$$\boldsymbol{p}_0 = \frac{1}{p} \cdot \begin{pmatrix} c_{000} + c_{011} - c_{001} - c_{010} \\ c_{000} + c_{101} - c_{001} - c_{100} \\ c_{000} + c_{110} - c_{010} - c_{100} \end{pmatrix}$$

we obtain for (1)

$$s = a \cdot u + b \cdot v + c \cdot w + d \cdot u \cdot v \cdot w + e \tag{2}$$

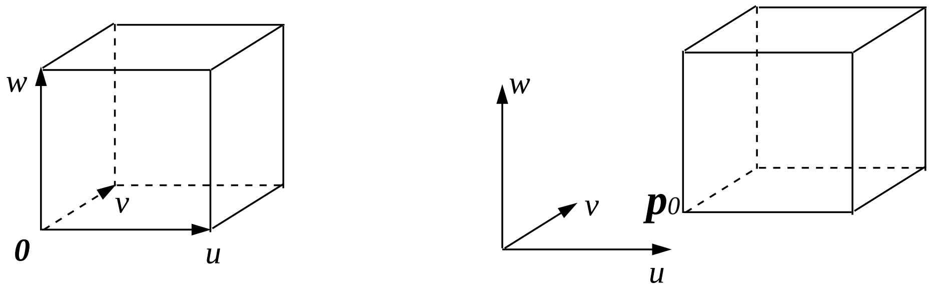

Fig. 2. Translating the coordinate system of a voxel.

with

$$
\begin{aligned}
a &= \frac{(c_{111} - c_{011}) \cdot (c_{100} - c_{000}) - (c_{110} - c_{010}) \cdot (c_{101} - c_{001})}{p} \\
b &= \frac{(c_{111} - c_{101}) \cdot (c_{010} - c_{000}) - (c_{110} - c_{100}) \cdot (c_{011} - c_{001})}{p} \\
c &= \frac{(c_{111} - c_{110}) \cdot (c_{001} - c_{000}) - (c_{101} - c_{100}) \cdot (c_{011} - c_{010})}{p} \\
d &= p,
\end{aligned}
$$

where e is a certain constant. Thus, we only have to analyze

$$s(u, v, w) = a \cdot u + b \cdot v + c \cdot w + d \cdot u \cdot v \cdot w = r = const \tag{3}$$

in $\mathbb{R}^3$. A classification of (3) can be achieved by rewriting (3) as $w = \frac{r - a \cdot u - b \cdot v}{c + d \cdot u \cdot v}$ and comparing the zeros of the numerator and denominator function. The zeros of the numerator function form a line in the $u - v-$plane, whereas the zeros of the denominator function give a hyperbola. Studying their interplay gives the following classification:

- case 1: $abcd \leq 0, d \neq 0$:
  - case 1.1: $r^2 \geq -\frac{4abc}{d}$: (3) gives 3 unconnected surface parts
  - case 1.2: $r^2 < -\frac{4abc}{d}$: (3) gives 2 unconnected surface parts
- case 2: $abcd < 0$: (3) consists of 1 connected part
- case 3: $abcd = 0, d \neq 0$:
  - case 3.1: $r \neq 0$:
    - case 3.1.1: $ab \neq 0, c = 0$: (3) gives 2 unconnected surface parts
    - case 3.1.2: $a \neq 0, b = c = 0$: (3) gives 3 unconnected surface parts
    - case 3.1.3: $a = b = c = 0$: (3) gives 4 unconnected surface parts
  - case 3.2: $r = 0$:
    - case 3.2.1: $ab \neq 0, c = 0$: (3) gives 3 unconnected surface parts
    - case 3.2.2: $a \neq 0, b = c = 0$: (3) gives 3 parts intersecting each other
    - case 3.2.3: $a = b = c = 0$: (3) gives 3 perpendicular planes.

Figure 3 illustrates these cases.

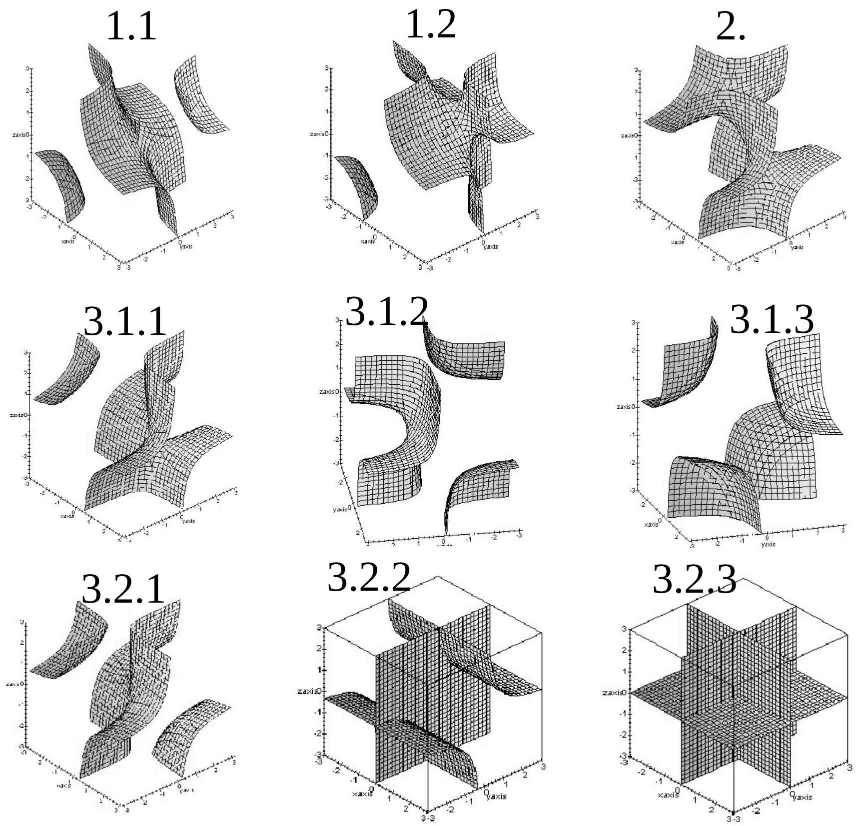

Fig. 3. Classification of the contours of (3) in $\mathbb{R}^3$.

§3. Segment Number of a Voxel

We now study the contour of (3) in a particular voxel $\boldsymbol{V} = [u_0, u_0 + 1] \times [v_0, v_0 + 1] \times [w_0, w_0 + 1]$. Unfortunately, the results of Section 2 are not directly applicable here because one connected surface part may intersect $\boldsymbol{V}$ more than once.

Varying the threshold r in (3), the contours change. So does the number of unconnected surface parts of the contour.

Definition 1. *Given the trilinear scalar field $s(u, v, w) = a \cdot u + b \cdot v + c \cdot w + d \cdot u \cdot v \cdot w$ in the domain of the voxel $\boldsymbol{V} = [u_0, u_0 + 1] \times [v_0, v_0 + 1] \times [w_0, w_0 + 1]$, the* segment number *$S(\boldsymbol{V})$ of $\boldsymbol{V}$ is the maximal number of unconnected surface parts of the contour $s(u, v, w) = r =$const in $\boldsymbol{V}$ for any threshold r.*

Figure 4 gives an example of a voxel $\boldsymbol{V}$ with $S(\boldsymbol{V}) = 1$. Increasing the value of r, the isosurface "moves" through the voxel. It consists of at most one connected part for any r. Figure 5 shows a voxel with $S(\boldsymbol{V}) = 4$. Here the contours consist of up to 4 unconnected parts.

The segment number is a threshold-independent characterization of a voxel $\boldsymbol{V}$. For any $\boldsymbol{V}$ we get $S(\boldsymbol{V}) \in \{1, 2, 3, 4\}$. For visualization purposes, voxels with a segment number 1 are of particular interest. As shown in the example of Figure 4, they have a nice behavior while varying r. In fact, for

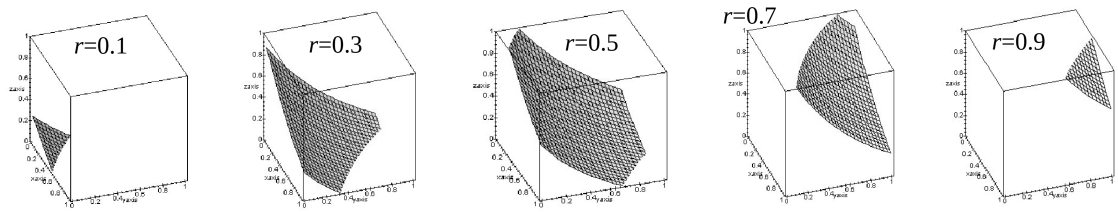

Fig. 4. Contours of a voxel with $S(\boldsymbol{V}) = 1$.

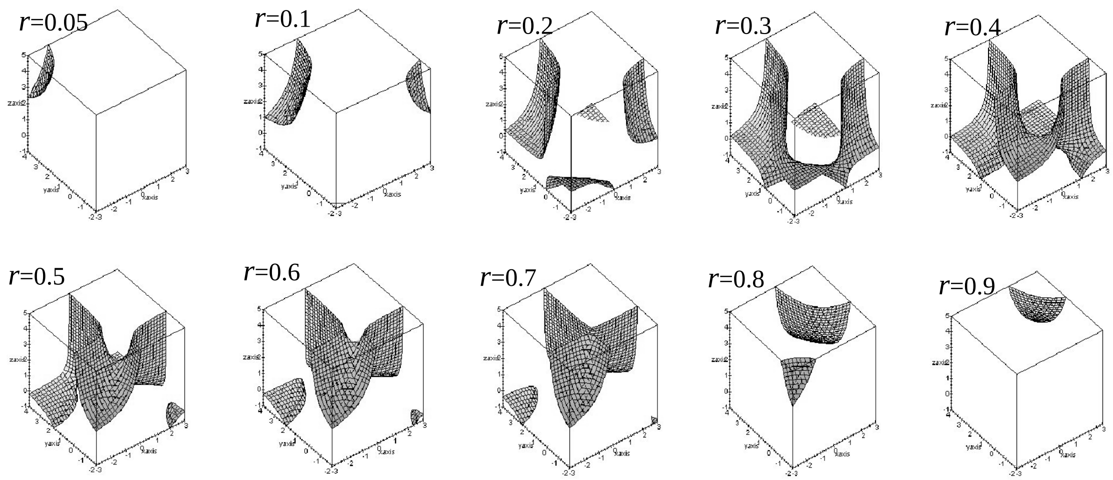

Fig. 5. Contours of a voxel with $S(\boldsymbol{V}) = 4$.

any r the contour consists of only one connected surface part inside $\boldsymbol{V}$. Thus, accelerated Marching Cubes methods may apply to them. Moreover, adjacent voxels with $S(\boldsymbol{V}) = 1$ may be merged to form one bigger voxel before applying Marching Cubes methods. So it makes sense to search for geometric conditions for a voxel $\boldsymbol{V}$ to have $S(\boldsymbol{V}) = 1$.

§4. Geometric Conditions for $S(\boldsymbol{V}) = 1$

In this section we give necessary and sufficient geometric conditions for a voxel to have $S(\boldsymbol{V}) = 1$. Again, we consider the contour of (3) in the voxel $\boldsymbol{V} = [u_0, u_0 + 1] \times [v_0, v_0 + 1] \times [w_0, w_0 + 1]$.

To formulate the conditions for $S(\boldsymbol{V}) = 1$, we need to introduce the concept of characteristic hyperbolas. The first characteristic hyperbola $\boldsymbol{h}_1$ in $\mathbb{R}^3$ is defined by the condition $s_{vw}(u, v, w) = 0$ in (3). $\boldsymbol{h}_1$ can be written as a rational quadratic Bezier curve described by two control vectors $\boldsymbol{b}_0^1$, $\boldsymbol{b}_2^1$ and a control point $\boldsymbol{b}_1^1$ (see [1]). For $\boldsymbol{h}_1$ we obtain

$$\boldsymbol{b}_0^1 = \begin{pmatrix} (-4bc)/d \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{b}_1^1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{b}_2^1 = \begin{pmatrix} 0 \\ 1/b \\ 1/c \end{pmatrix}, \quad w_1^1 = 1,$$

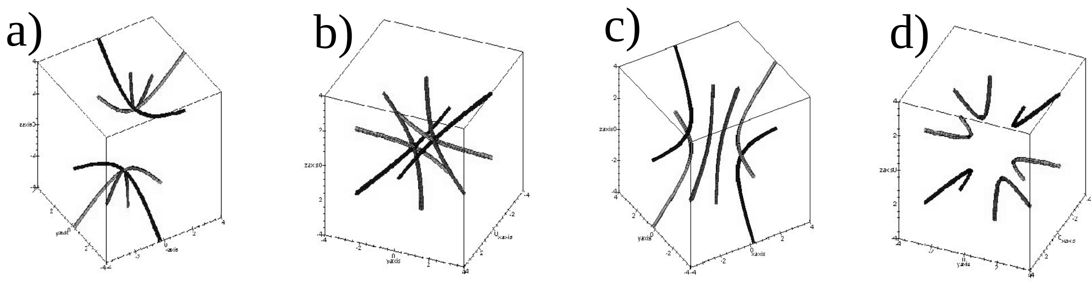

Fig. 6. Location of characteristic hyperbolas; a),b): $abcd < 0$; c),d):$abcd > 0$.

where w_1^1 is the weight of $\boldsymbol{b}_1^1$. Then we obtain

$$\boldsymbol{h}_1(t) = \frac{\boldsymbol{b}_0^1 B_0^2(t) + w_1^1 \boldsymbol{b}_1^1 B_1^2(t) + \boldsymbol{b}_2^1 B_2^2(t)}{w_1^1 B_1^2(t)}.$$

In a similar way we define the characteristic hyperbola $\boldsymbol{h}_2$ by $s_{uw}(u,v,w) = 0$, and $\boldsymbol{h}_3$ by $s_{uv}(u,v,w) = 0$. The Bezier point $\boldsymbol{b}_1^2$ with the corresponding weight w_1^2 and the control vectors $\boldsymbol{b}_0^2, \boldsymbol{b}_2^2$ describing $\boldsymbol{h}_2$ are

$$\boldsymbol{b}_0^2 = \begin{pmatrix} 0 \\ (-4ac)/d \\ 0 \end{pmatrix}, \quad \boldsymbol{b}_1^2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{b}_2^2 = \begin{pmatrix} 1/a \\ 0 \\ 1/c \end{pmatrix}, \quad w_1^2 = 1.$$

$\boldsymbol{h}_3$ is described by

$$\boldsymbol{b}_0^3 = \begin{pmatrix} 0 \\ 0 \\ (-4ab)/d \end{pmatrix}, \quad \boldsymbol{b}_1^3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{b}_2^3 = \begin{pmatrix} 1/a \\ 1/b \\ 0 \end{pmatrix}, \quad w_1^3 = 1.$$

If $a \cdot b \cdot c \cdot d < 0$ then $\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3$ intersect in two common points. Figures 6a and b illustrate this situation from two different viewpoints. If $a \cdot b \cdot c \cdot d > 0$ then $\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3$ do not have any intersections. Figures 6c and d show this from different viewpoints. The degenerate case $a \cdot b \cdot c \cdot d = 0$ is omitted here.

To formulate conditions for $S(\boldsymbol{V}) = 1$, we have to classify the faces of $\boldsymbol{V}$. Given the voxel $\boldsymbol{V} = [u_0, u_0+1] \times [v_0, v_0+1] \times [w_0, w_0+1]$, let $\boldsymbol{f}_1 = \{(u,v,w) \in \boldsymbol{V} : u = u_0 \vee u = u_0 + 1\}$, $\boldsymbol{f}_2 = \{(u,v,w) \in \boldsymbol{V} : v = v_0 \vee v = v_0 + 1\}$, and $\boldsymbol{f}_3 = \{(u,v,w) \in \boldsymbol{V} : w = w_0 \vee w = w_0 + 1\}$. See Figure 7 for an illustration of the faces.

Theorem 1. *Let $\boldsymbol{V} = [u_0, u_0+1] \times [v_0, v_0+1] \times [w_0, w_0+1]$ be a voxel in the scalar field defined by (3). Then the condition $S(\boldsymbol{V}) = 1$ is equivalent to the three conditions $\boldsymbol{h}_1 \cap \boldsymbol{f}_1 = \emptyset$ and $\boldsymbol{h}_2 \cap \boldsymbol{f}_2 = \emptyset$ and $\boldsymbol{h}_3 \cap \boldsymbol{f}_3 = \emptyset$.*

Figure 8 illustrates the idea of the proof. Suppose $\boldsymbol{h}_3$ intersects $\boldsymbol{f}_3$ as shown in Figure 8a. Figure 8b is a magnification of the voxel and $\boldsymbol{h}_3$ in

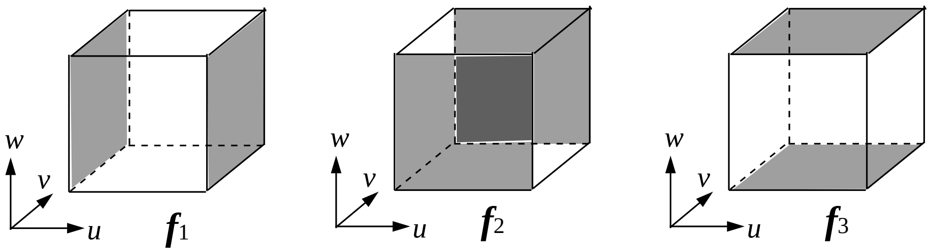

Fig. 7. The faces $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3$ of a voxel.

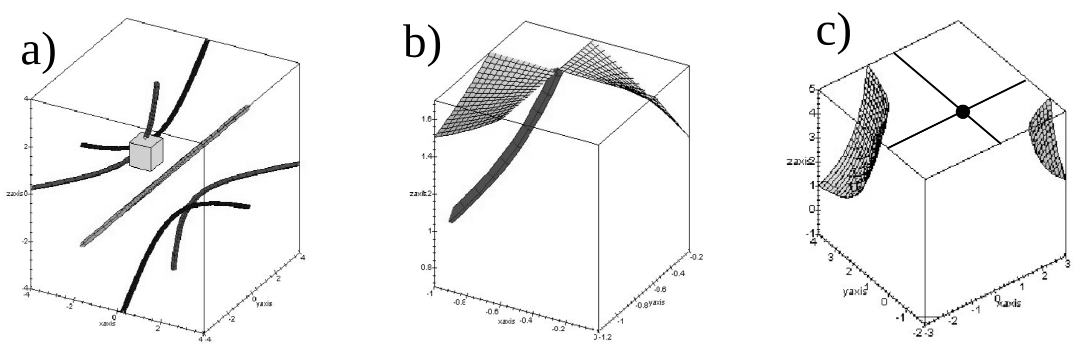

Fig. 8. Proof idea of Theorem 1.

Figure 8a. We compute the intersection point of $\boldsymbol{h}_3$ and $\boldsymbol{f}_3$, and consider the contour passing through this point. As shown in Figure 8a, this contour consists of at least two surface parts.

For the proof of the converse statement of Theorem 1, we assume that for a certain threshold r the contour consists of at least two unconnected surface parts. Then we can find a face of $\boldsymbol{V}$ which has two intersection curves with the contour. (In the worst case we have to vary r to find such a face). (Figure 8c shows two surface parts of the contour which produce two intersection curves in the upper face of $\boldsymbol{f}_3$). Then we can find a point on this face which is the intersection point with the corresponding characteristic hyperbola. (In Figure 8c, the marked point on the upper part of $\boldsymbol{f}_3$ is the intersection with $\boldsymbol{h}_3$).

§5. Results and Future Work

We have tested the voxels of a CT test data set for the property $S(\boldsymbol{V}) = 1$. The data set consists of $255 \times 255 \times 108 = 7,022,700$ voxels. Figure 9 shows a slice through the data set.

In the raw data we found 1,978,711 voxels with $S(\boldsymbol{V}) = 1$ (28 %). After some noise reducing filter operations on the data, we detected 4,833,063 voxels with $S(\boldsymbol{V}) = 1$ (69 %). This shows that there is a reasonable number of voxels with $S(\boldsymbol{V}) = 1$ to pay special attention to them.

In the future we plan to develop algorithms to merge voxels with $S(\boldsymbol{V}) = 1$ to form bigger voxels before starting the Marching Cubes algorithm.

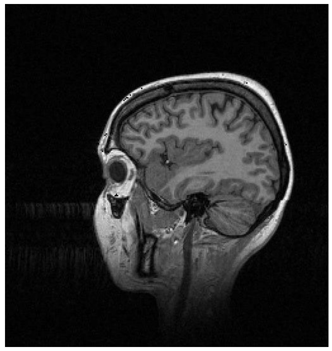

Fig. 9. Slice through the test data set.

Acknowledgments. The author would like to thank Prof. Heidrun Schumann from the University of Rostock for her constant support and encouragement of this work.

References

1. Farin, G., *NURB Curves and Surfaces*, A K Peters, Wellesley, 1995.
2. Hamann, B., I. J. Trotts, and G. Farin, On approximating contours of the piecewise trilinear interpolant using triangular rational quadratic Bezier patches, IEEE Transactions on Visualization and Computer Graphics, **3(3)** (1997), 215–227.
3. Lorensen, G. M., and H. E. Cline, Marching Cubes: a high resolution 3D surface reconstruction algorithm, Computer Graphics, **21** (1987), 163–169.
4. Nielson, G. M., and B. Hamann, The asymptotic decider: resolving the ambiguity in marching cubes, proceedings IEEE Visualization 91, 1991, 83–91.

Holger Theisel
University of Rostock, Computer Science Department
PostBox 999, 18051 Rostock, Germany
`theisel@informatik.uni-rostock.de`
`http://wwwicg.informatik.uni-rostock.de/~theisel/`