Locally Adaptive Marching Cubes through Iso-Value Variation

Michael Glanznig   Muhammad Muddassir Malik   M. Eduard Gröller

Vienna University of Technology, Institute of Computer Graphics and Algorithms
Karlsplatz 13, A-1040 Vienna, Austria

michael.glanznig@student.tuwien.ac.at   mmm@cg.tuwien.ac.at   groeller@cg.tuwien.ac.at

ABSTRACT

We present a *locally adaptive marching cubes* algorithm. It is a modification of the marching cubes algorithm where instead of a global iso-value each grid point has its own iso-value. This defines an iso-value field, which modifies the case identification process in the algorithm. The marching cubes algorithm uses linear interpolation to compute intersections of the surface with the cell edges. Our modification computes the intersection of two general line segments, because there is no longer a constant iso-value at each cube vertex. An iso-value field enables the algorithm to correct biases within the dataset like low frequency noise, contrast drifts, local density variations, and other artefacts introduced by the measurement process. It can also be used for blending between different isosurfaces (e.g., skin, veins, and bone in a medical dataset).

Keywords
marching cubes, contouring, iso-value field, isosurface correction, blending between isosurfaces

1. INTRODUCTION

The extraction of a constant density surface from a dataset (contouring) is used in many disciplines. In medical environments it has opened new ways for radiologists and physicians to visualize and interact virtually with the human body. In metrology (the science of measurement) industrial Computed Tomography (CT) scanners are used for specimen measurements and non-destructive-testing (NDT).

The most well-known algorithm for extracting a polygonal representation (e.g., set of triangles) of a constant density surface (isosurface) from a 3D dataset is *marching cubes* and was published by LORENSEN and CLINE in 1987 [Lor87]. The marching cubes algorithm works on the *divide and conquer* principle. The volumetric dataset is divided into cells and the isosurface within each cell is calculated. It uses a global threshold called iso-value to determine the interior and exterior of the isosurface in each cell.

A volumetric dataset often is subject to biases. Such biases can be low frequency noise, contrast drifts, lo-

cal density variations and other artefacts introduced by the measurement process. Examples are noise-induced streaks, aliasing, beam-hardening, scattered radiation effects, cupping etc. [Hei07]. When performing contouring on a biased dataset with a global iso-value the result may not be satisfactory. The isosurface may contain holes, thinned regions, or regions where volume is added. An iso-value field where every vertex has its own iso-value enables the contouring algorithm to compensate for flaws. We want to locally modify the iso-value and with it the surface in those regions of the dataset where measurement errors would cause an unsatisfactory surface to be generated. Additionally an iso-value field supports blending between various isosurfaces by linearly interpolating between iso-values.

Our presented *locally adaptive marching cubes* algorithm modifies the marching cubes algorithm and allows the specification of an iso-value field. The modifications to the marching cubes algorithm include a slightly different case identification process and a changed computation of intersections between the surface and cell edges. Marching cubes uses linear interpolation to compute intersections. In the modified algorithm we intersect two general line segments that are defined by two density values and two iso-values.

In the following we list some related work on isosurface correction. Then we provide a brief overview on the marching cubes algorithm and present our modifications to transform it into the *locally adaptive marching cubes* algorithm. In the implementation section we

present a reference application that supports the specification of iso-value fields on volumetric datasets. Another application visualizes the surface in the interior of a single cell. We conclude with discussing simple and more complex iso-value fields on various volumetric datasets.

2. RELATED WORK

Common approaches to correct isosurfaces are either to modify the input dataset and perform contouring with a global iso-value or optimize the isosurface after generation.

HEINZL ET AL. [Hei07] use Dual Energy Computed Tomography (DECT) and image fusion to reconstruct a dataset before extracting an isosurface. As input images they use a high-energy macro focus image, which is blurry but less affected by artefacts, and a low-energy micro focus image, which is precise but artefact-prone. The overall structure of the specimen is taken from the low precision dataset while the sharp edges are derived from the high precision dataset. For contouring a local surface extraction approach is used. First an isosurface is extracted from the dataset, which describes the real surface in good approximation. In the next step the isosurface is corrected by moving each surface vertex along the direction of the surface normal until the gradient magnitude in the volume reaches a maximum. The gradient magnitude is computed from the low energy dataset.

For variance comparison HEINZL ET AL. [Hei06] developed a pipeline, which uses a watershed filter on the gradient information of the smoothed volumetric data to create a binary dataset. This dataset is then taken to create the surface model by using constrained elastic surface nets. With this pipeline they try to compensate for artefacts that are present in the dataset.

KOBBELT ET AL. [Kob01] developed an extended marching cubes algorithm to compensate for alias artefacts at sharp features of the extracted surfaces. First, their algorithm detects those grid cells that are intersected by a sharp feature of the surface. Then it computes additional sampling points that lie on the feature and inserts them into the mesh. Additional sampling points are retrieved from a local distance field and its gradient.

MATYAS ET AL. [Mat05] present an automated contouring approach with locally changing iso-values to contour anatomic branching structures. They first segment the dataset and determine the appropriate iso-value in each segment. Iso-values do not vary continuously but in discrete steps, which requires to blend together the extracted isosurfaces of the segments. They named the resulting surface "metasurface".

ŠEREDA [Ser07] proposes local transfer functions to adapt to locally modified data values in the dataset. A transfer function (TF) is used to transform density values into optical properties such as color and opacity. Typically a global TF is defined. In his work he discusses how the definition of TFs can be automated and how local TFs produce better results on datasets with local data variations.

3. LOCALLY ADAPTIVE MARCHING CUBES

In the following subsections we first briefly describe the marching cubes algorithm and then present the necessary modifications to transform it into our *locally adaptive marching cubes* algorithm.

3.1 The Marching Cubes Algorithm

The algorithm as published by LORENSEN and CLINE [Lor87] subdivides the volumetric dataset into cubical cells and processes each of them separately. To identify the triangulation of the isosurface inside a cell, the density value at each vertex is compared with the global iso-value to check if the vertex lies inside or outside the isosurface. The possible triangulations of a cell are stored in a case lookup table. There are 256 possible cases, which can be reduced by topological and rotational symmetry to 15 cases. In the original algorithm there exist ambiguous cases where more than one triangulation is possible. This can lead to holes in the isosurface. A method to overcome this problem is the *asymptotic decider* which was proposed by NIELSON and HAMANN [Nie91]. A simple solution adds six additional cases to the case table to avoid holes [Sch02]. After retrieving the triangles for a case from the lookup table, the positions where the triangle vertices lie on the cell edges have to be calculated. This is done with linear interpolation.

Roughly four steps are performed for each cell:

1. identify the appropriate index case

2. look up triangles in the case lookup table

3. interpolate intersection points along the cell edges

4. compute surface normals for triangle vertices (required for shading)

3.2 Modified Algorithm: Support of Iso-Value Fields

Our *locally adaptive marching cubes* algorithm uses one iso-value per cell vertex instead of a single global iso-value. Each cell has eight iso-values, one at each

vertex of the cube. Trilinearly interpolating these varying iso-values produces a continuous iso-value field. This alters how the index cases are identified and also how the intersection points along the cell edges are calculated.

In the case identification process the marching cubes algorithm compares every cell vertex value to the global iso-value. In the modified algorithm every vertex has its own iso-value to which the cell vertex value is compared. It is then decided whether the vertex is considered to be zero or one in the index, i.e., inside or outside. The remaining part of the case identification process is similar to the original algorithm.

The marching cubes algorithm uses linear interpolation to interpolate triangle vertices along the edges. Since both vertices of an edge have now different iso-values, the intersection point of two arbitrary line segments has to be found (Figure 1).
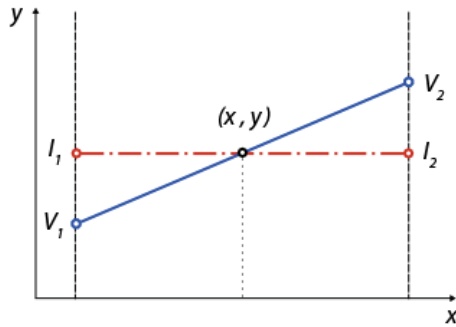
(a) linear interpolation
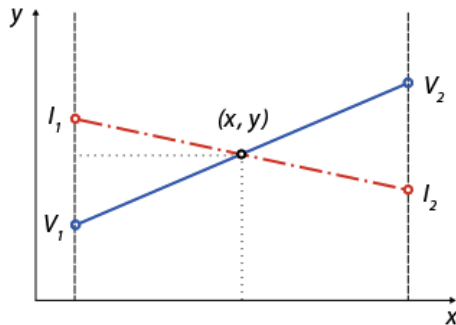
(b) arbitrary line intersection

$-\cdot-\cdot-$ iso-values
$\textemdash$ density values

Figure 1: Linear interpolation in the original algorithm (a) and arbitrary line intersection in the modified algorithm (b). In (a) the slope of the iso-value line is always zero. In (b) it can have any value.

We assume that the line segment of density values is defined by the two cube vertices V_1 and V_2 and the line segment of iso-values is defined by I_1 and I_2. In the following x corresponds to the spatial position along an edge and y corresponds to density values or iso-values.

Line equations are of the form $A_j \cdot x + B_j \cdot y = C_j$, $j = 1, 2$. V_j^x refers to the x-component of V_j.

We assign:
$$A_1 = V_2^y - V_1^y$$
$$B_1 = V_1^x - V_2^x$$
$$C_1 = A_1 \cdot V_1^x + B_1 \cdot V_1^y$$

A_2, B_2 and C_2 are derived in the same way.

When solving our linear system:
$$A_1 \cdot x + B_1 \cdot y = C_1 \tag{1}$$
$$A_2 \cdot x + B_2 \cdot y = C_2 \tag{2}$$

we get:
$$x = \frac{C_1 \cdot B_2 - C_2 \cdot B_1}{A_1 \cdot B_2 - A_2 \cdot B_1}$$

$$y = \frac{C_2 \cdot A_1 - A_2 \cdot C_1}{A_1 \cdot B_2 - A_2 \cdot B_1}$$

If $A_1 \cdot B_2$ equals $A_2 \cdot B_1$ the lines are parallel and do not intersect. Even if an intersection point between the two lines exists, it may not lie on the line segments themselves. Hence we have to check if the intersection point lies between the line segment's endpoints:
$$\min(V_1^x, V_2^x) \leq x \leq \max(V_1^x, V_2^x)$$

It is sufficient to check only the x-values of one line segment to decide if the intersection point is valid since the line segments are correlated (see Figure 1).

4. IMPLEMENTATION

We implemented the algorithm with the proposed changes and created two basic applications to show the algorithm's possibilities. One application (*Locally Adaptive Marching Cubes*, Figure 2) allows the specification of iso-value fields on volumetric datasets. The second one (*Cube Insight*) visualizes the interior of a single cubical cell using trilinear interpolation.

The graphics and visualization part of the applications is handled by *The Visualization Toolkit* (VTK) [Sch96]. Using VTK for the reference applications has several advantages. First, the toolkit is popular in the visualization community and is open source. Second, an implementation of the marching cubes contouring filter is already available and can be easily modified.

The generation of iso-value fields requires user interaction. There exist various possibilities to alter the iso-value field. Our application supports fields with linear iso-value gradients and isotropic 3D Gaussian distributions of iso-values centered at specific sampling points of the dataset (Figure 3).
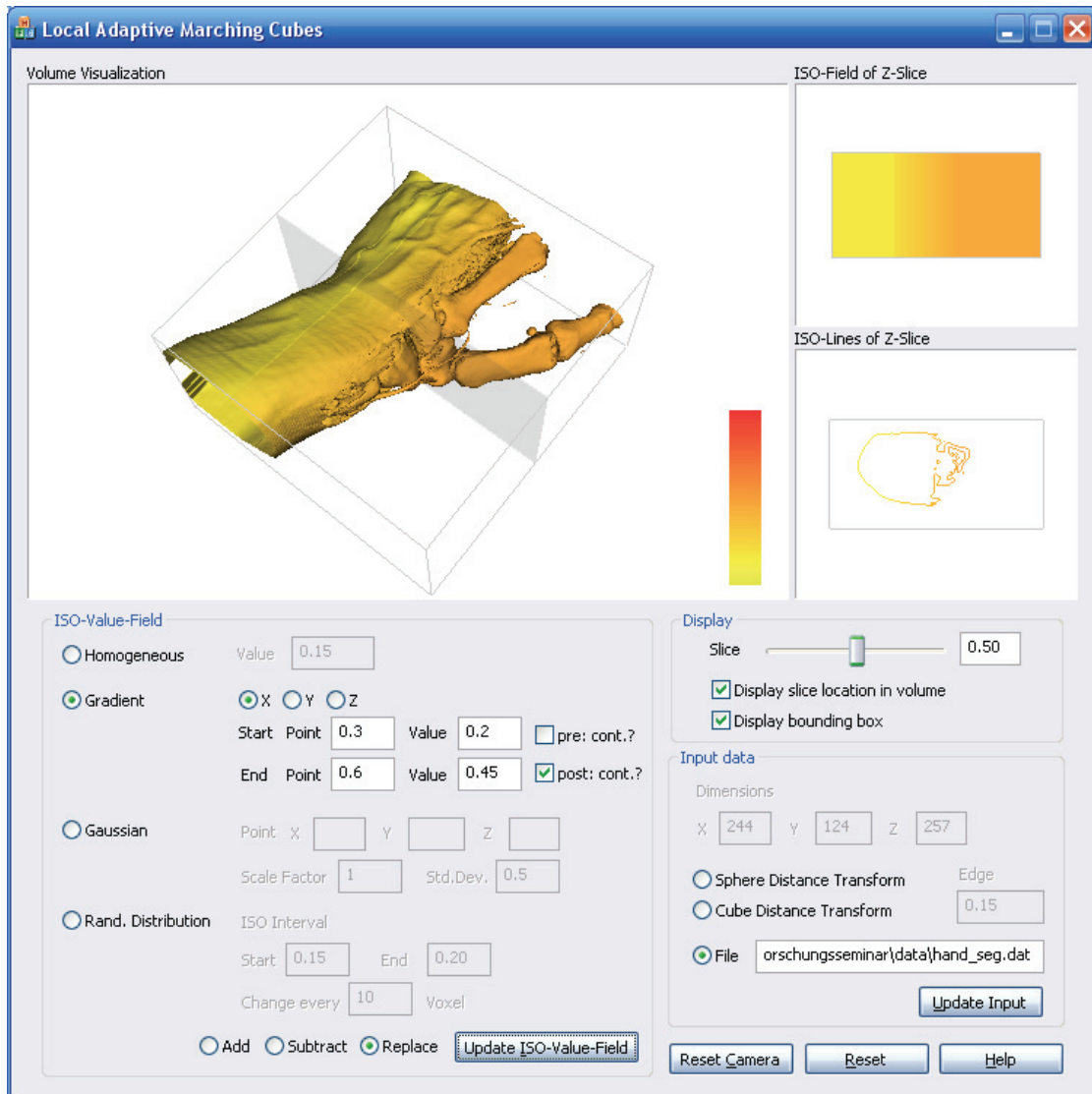
Figure 2: Locally Adaptive Marching Cubes allows the specification of iso-value fields on volumetric datasets. It is possible to specify homogeneous iso-value fields, fields with linear iso-value gradients in coordinate directions, and isotropic 3D Gaussian distributions of iso-values centered at specific sampling points of the dataset.
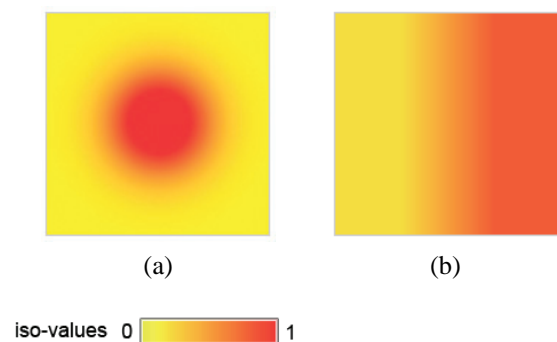

iso-values 0 ▭ 1

Figure 3: (a) Isotropic Gaussian distribution added to a homogeneous iso-value field. (b) Linear gradient in x-direction with range [0.3, 0.7].

5. RESULTS

We now discuss several iso-value fields on various volumetric datasets. First, we show how iso-value fields impact the appearance of simple objects. Then we present a biased dataset and discuss how the surface can be corrected with our method. Furthermore we show how our technique supports blending between different isosurfaces and discuss how the surface inside a cell is modified when iso-values at the cell vertices are altered.

5.1 Simple Objects

In Figure 4 the volume is a Euclidean distance transform of a sphere. The distance transform is generated by computing the Euclidean distance of each sample point from the centre of the dataset. Distance trans-
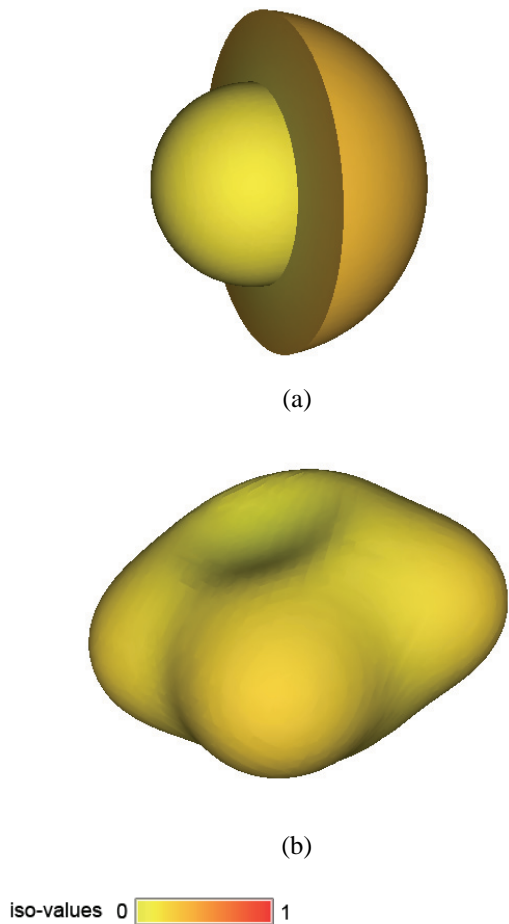
(a)

(b)

iso-values 0 ▭ 1

Figure 4: Euclidean distance transform of a sphere ($64 \times 64 \times 64$). In (a) the iso-value field is divided into two halves. In (b) it has been modified with Gaussian distributions of standard deviation 5 at four locations.

forms of simple objects like a sphere are well suited to show how different iso-value fields change the appearance of the surface. Continuous iso-value modifications must lead to continuous changes in the surface of the sphere (bulges and dents). Since we have a distance field, increasing iso-values lead to an increased sphere radius, decreasing iso-values lead to a decreased radius of the resulting surface. In Figure 4a the iso-value field is divided into two halves, one with a lower and one with a higher iso-value which leads to two half-spheres. Figure 4b shows the sphere with a more complex iso-value field. Gaussian distributions of standard deviation 5 have been added to a homogeneous iso-value field at three locations and one Gaussian distribution has been subtracted from it. This leads to three bulges and one dent in the surface. Modifying the iso-value field in such a way can be considered as digitally painting in the 3D iso-value field, where the brush is given by the Gaussian distribution. Various extensions and

variations to this painting metaphor and brush types are easily conceivable.

5.2 Correcting the Isosurface of a Biased Volumetric Dataset
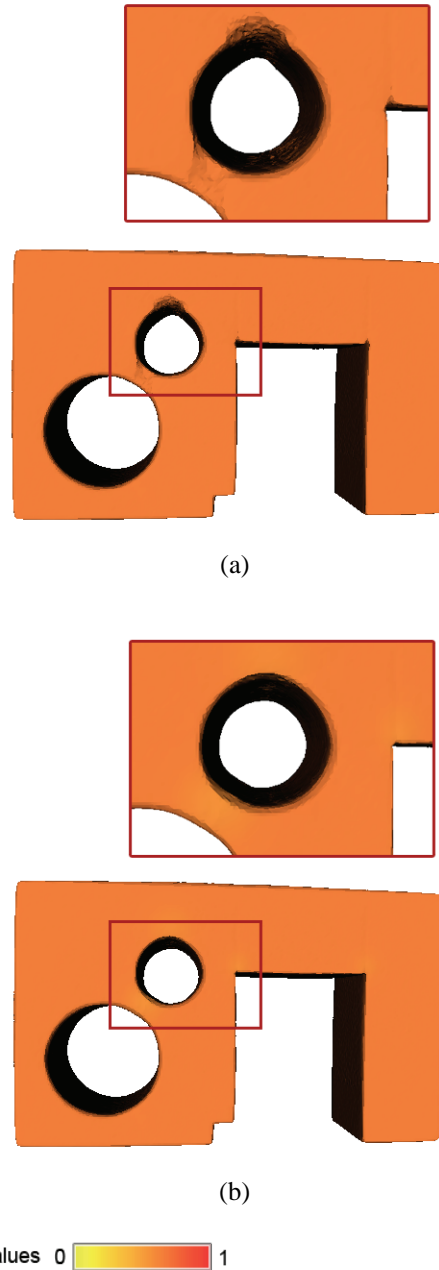

(a)

(b)

iso-values 0 ▭ 1

Figure 5: A volumetric dataset ($164 \times 263 \times 90$) that contains artefacts. (a) shows the isosurface extracted with a global iso-value of 0.65. There are several flaws visible in the surface especially at the smaller hole. (b) depicts the surface that was extracted with a modified iso-value field and has most of the flaws corrected.

The specimen depicted in Figure 5 has severe scattered radiation and beam hardening artefacts in the area of

the drill holes and also at the rectangular milling. The isosurface in Figure 5a shows several errors especially at the smaller hole where the geometry of the object is changed. Given that there exists an iso-value field where the surface is appropriate in that area of the dataset, it is possible to correct the surface with our method. The iso-value field is modified in those areas where flaws are present. To assure a continuous transition between iso-values an isotropic 3D Gaussian distribution is used. Figure 5b shows the resulting surface. Several Gaussian distributions of various kernel sizes have been subtracted from the homogeneous iso-value field of value 0.65. The flaws at the drill holes and at the corners of the milling are thus corrected.

The approach described here requires user interaction to generate the iso-value field. The user introduces his knowledge about the object geometry into the process. One can think also of automated approaches to surface correction using iso-value fields. By generating special (e.g., data-dependent) iso-value fields the information that corrects artefacts can be passed to the extracting algorithm as input instead of changing/correcting the extracted surface afterwards.

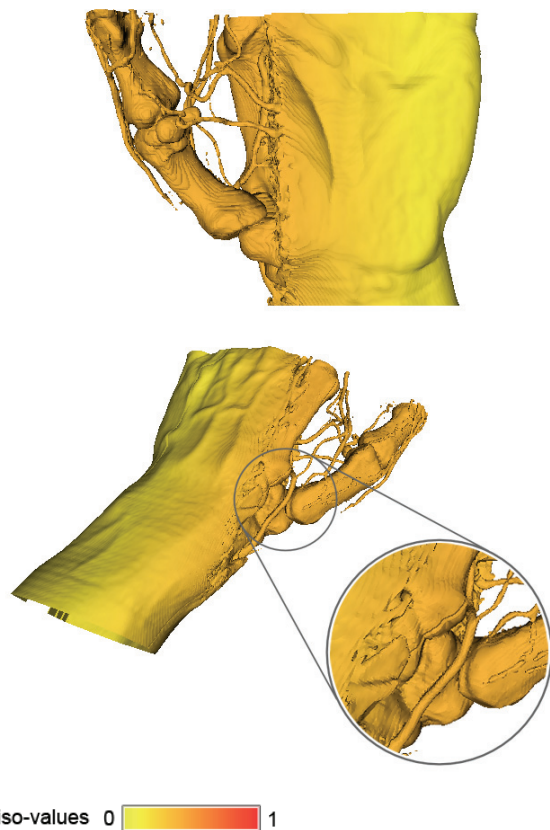## 5.3 Blending Between Isosurfaces

iso-values 0 ■■■■■ 1

Figure 6: A human left hand ($244 \times 124 \times 257$) with linear blending between the isosurfaces of skin and bone.

Linear gradients in iso-value fields can be used to blend between different isosurfaces in one processing step. The interpolation of iso-values in a dataset region leads to a continuous transition between isosurfaces. In Figure 6 a human left hand is shown. A linear iso-value gradient in the x-range [0.25, 0.6] with iso-value range [0.15, 0.4] is used to visualize the surface of skin on one half and the surface of bones and veins on the other half of the dataset. A contrast medium has been injected into the veins, since there is almost no difference between their density value and the density value of bone. In the result image it becomes visible how the skin gets thinner, dissolves and then exposes the bones in a smooth way.

(a)   (b)
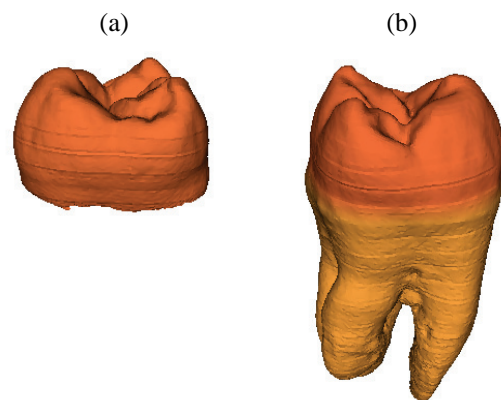

iso-values 0 ■■■■■ 1

Figure 7: A human tooth ($256 \times 156 \times 161$). (a) With a global iso-value of 0.7 only the enamel part of the tooth is displayed. (b) An iso-value field with a linear gradient in the z-range [0.7, 0.75] is used to blend between the isosurface of the dentine part and the isosurface of the enamel part of the tooth.

In Figure 7 a human tooth is depicted which roughly consists of the enamel (top part of the tooth) and the dentine (bottom part of the tooth). We assume that the enamel should be visualized with an iso-value of 0.7. With that value it is not possible to visualize the dentine because it starts to dissolve at iso-values greater than 0.6 (Figure 7a). A solution for this problem could be to generate isosurfaces for the dentine with iso-value 0.5 and for the enamel with iso-value 0.7 and then combine both isosurfaces. With our method this is not necessary. A linear iso-value gradient in the z-range [0.7, 0.75] with iso-value range [0.5, 0.7] can be used to visualize the entire tooth in one processing step (Figure 7b) without the need to combine two generated isosurfaces.

5.4 Surface in the Interior of a Cell

Figure 8 shows the surface in the interior of a cell for three index cases. Generally modifications of iso-

values result in the movement of intersection points on the cell edges. There can also be topology changes and the index case might change as well. Increasing the iso-value at the circled vertex in Figure 8a alters the index case. An opening in the surface appears. In Figure 8b the index case is not changed, but intersection points are moved. First both "wings" of the surface have equal height. After increasing the iso-value at the circled vertex one "wing" is lower than the other. The modification of the iso-value in Figure 8c connects both components of the surface and changes its topology while leaving the index case unchanged.

6. SUMMARY AND CONCLUSIONS

This paper describes *locally adaptive marching cubes*, a modification of the marching cubes algorithm. It allows the usage of an iso-value field instead of a single global iso-value. Iso-value fields are generated by user specification and are independent from the volume data. Surfaces are extracted, which are a continuous blend between various isosurfaces. The necessary modifications to the algorithm are simple. First, the case identification process has to be altered since we now have a different iso-value for each cell vertex. Second, when finding the intersection points of the surface with the cell edges, we now have to compute the intersection of two general line segments. Modifications of iso-values can impact the surface in the interior of a cell in three ways. First, the modification of iso-values can change the index case as in Figure 8a. When the index case is not changed, the modification of iso-values moves intersection points of the surface with the cell edges as in Figure 8b. There can also be topology changes as in Figure 8c.

Locally adaptive marching cubes can be used for the correction of isosurfaces with flaws like low frequency noise, contrast drifts, and local density variations. The iso-value field is appropriately modified in those regions of the dataset where flaws are present (Figure 5). Our technique also supports blending between different isosurfaces by specifying iso-value gradients in the iso-value field. This is useful when the entire dataset can not be visualized with one global iso-value (Figure 7), or when different isosurfaces should be visualized in different regions of the dataset (Figure 6). Blending between isosurfaces normally requires to first extract all isosurfaces and then combine them in an additional processing step. Our algorithm supports blending directly in the contouring process.

Two reference applications have been presented. One application allows the specification of iso-value fields and volumetric datasets. A second application visualizes the interior of a single cell. Possibilities to enhance our system include improving the user interface and improving the process of defining modifications to the iso-value field. When correcting isosurfaces with flaws due to artefacts there may be automated ways to specify the iso-value field.

7. REFERENCES

[Hei06] C. Heinzl, R. Klingesberger, J. Kastner and E. Gröller. Robust Surface Detection for Variance Comparison and Dimensional Measurement. In *Eurographics/IEEE-VGTC Symposium on Visualization,* 75–82, 2006.

[Hei07] C. Heinzl, J. Kastner and E. Gröller. Surface Extraction from Multi-Material Components for Metrology using Dual Energy CT. *IEEE Transactions on Visualization and Computer Graphics,* 13(6):1520–1527, 2007.

[Kob01] L. P. Kobbelt, M. Botsch, U. Schwanecke and H.-P. Seidel. Feature Sensitive Surface Extraction from Volume Data. In *SIGGRAPH '01,* 57–66, 2001.

[Lor87] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH '87,* 163–169, 1987.

[Mat05] N. M. Matyas, L. Linsen and B. Hamann. Metasurfaces: Contouring with Changing Isovalue. In *VMV 2005,* 147–154, 2005.

[Nie91] G. M. Nielson and B. Hamann. The asymptotic decider: resolving the ambiguity in marching cubes. In *VIS '91,* 83–91, 1991.

[Ser07] P. Šereda. *Facilitating the Design of Multidimensional and Local Transfer Functions for Volume Visualization.* PhD thesis, Eindhoven University of Technology, 2007.

[Sch96] W. J. Schroeder, K. M. Martin and W. E. Lorensen. The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In *VIS '96,* 93–101, 1996.

[Sch02] W. J. Schroeder, K. M. Martin and W. E. Lorensen. *The Visualization Toolkit, An Object-Oriented Approach To 3D Graphics, Third Edition.* Kitware, Inc., 154–159, 2002.
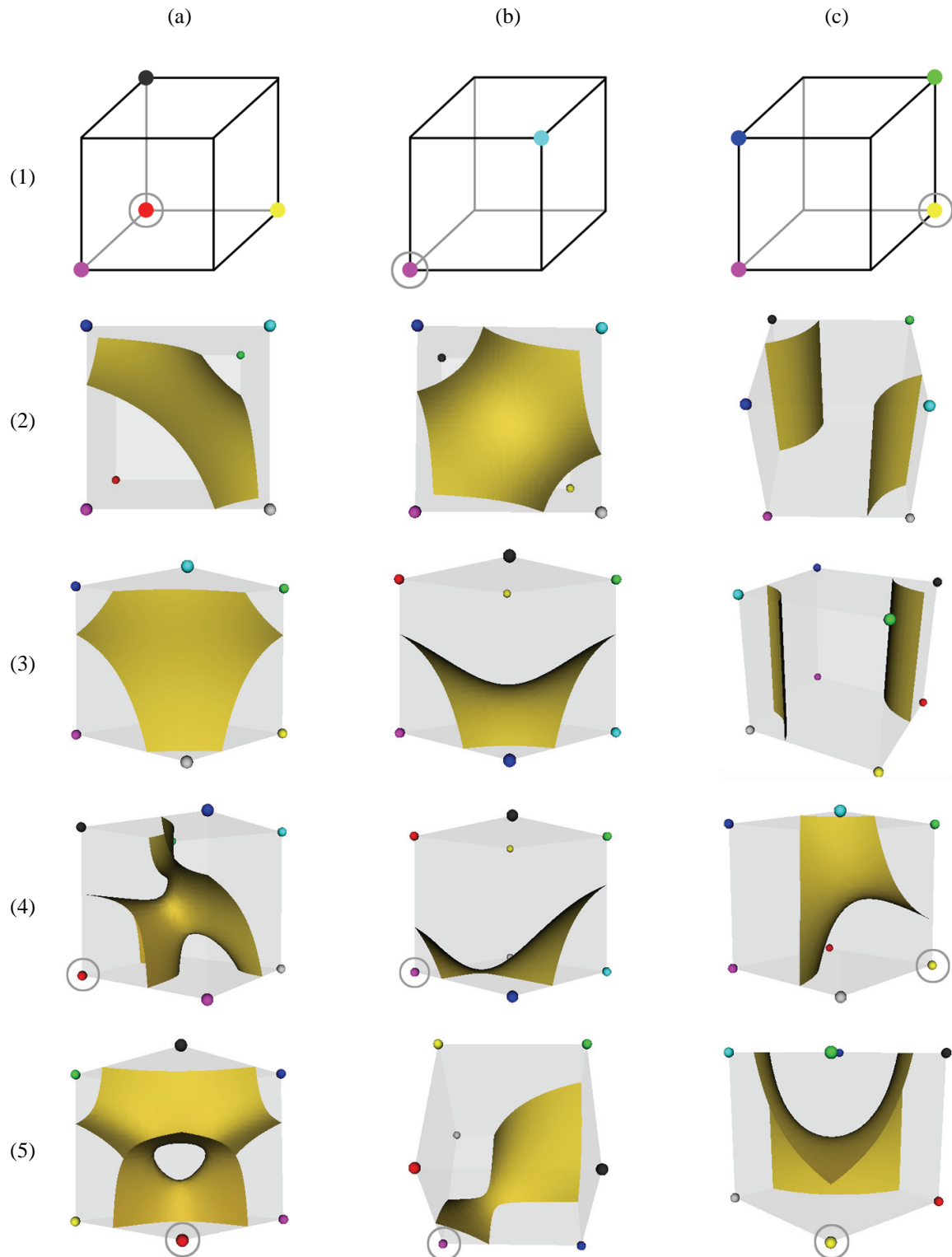
Figure 8: Visualization of the surface in the interior of a cell for three index cases. (1) depicts the index case. A cell vertex marked with a dot indicates that the density value exceeds its iso-value.The surface intersects those edges where one vertex is marked with a dot while the other one is not. (2) and (3) show the case from two different viewpoints while (4) and (5) show corresponding views where the iso-value of a vertex is modified. Vertices with modified iso-values are circled. Density values are set to 0.1 for non-marked vertices and are set to 0.25 for marked vertices. The iso-values are set to 0.15.