Maintaining Sharp Features in Surface Construction for Volumetric Objects

Robert Edward Loke
IIT-CNR, Pisa, Italy, robert.loke@iit.cnr.it

Frederik W. Jansen
EWI, TU Delft, The Netherlands, f.w.jansen@tudelft.nl

ABSTRACT

Discretized Marching Cubes (DMC) is a standard method in computer graphics and visualization for constructing 3D surfaces in data represented on a regular grid. After thresholding, it builds high-resolution surfaces by tiling surface patches halfway between objects and background in the data. This paper shows that if surfaces are built locally, in a high-resolution sub-grid of a cell instead of directly in a cell, sharp surfaces can be generated in order to preserve concave and convex object features. The main advantage is the improved geometric models that are extracted. This makes lower approximation errors and lower triangle counts possible.

Keywords: Computer Graphics: Curve, surface, solid, and object representations; Computer aided design (modeling of curves and surfaces); Computational geometry; Image processing.

1 INTRODUCTION

Volumetric models are defined on a regular data grid and can either be rendered with direct volume rendering techniques or with fast polygon rendering hardware. The latter only after extracting a surface model from the voxel data by surface construction algorithms like Marching Cubes [7]. In the last two decades a lot of research has been dedicated to improving the MC algorithm, both topologically [12, 14, 5] and in terms of reduced triangle counts [11, 13]. To reduce the number of degenerated triangles and to make it easier to merge smaller triangles into larger surface patches, the Discrete Marching Cubes (DMC) algorithm [8] fixes the position of the nodes of the triangulation to the midpoints of the cell edges. This reduces the number of orientations of the triangles to a limited set of discrete orientations, which facilitates merging the triangles into larger surface patches.

MC and DMC methods were originally designed for extracting and visualizing isosurfaces for gray shaded data, but they can also be advantageously used for (segmented, thresholded) binary voxel data, because they generate a surface in between the object and the background, and in this way they always create a manifold surface, also for isolated points, lines and thin planes. By constructing surfaces at a spatial resolution that is higher than the spatial resolution of the original voxel data we have room to build a manifold surface. Indeed, as was stated by the authors of MC, the spatial resolution of the surface is higher than that of the data [7].

Although DMC generates manifold surfaces and optimizes the triangle output, it has one remaining drawback: it creates oblique surfaces at object slopes but also at sharp edges and corners. In this paper we show that the standard surface mapping by DMC can be derived from a much simpler triangulation on the eight sub-cells of each cell. By locally refining the grid and applying a complementary surface model we can preserve sharp features such as convex and concave edges and corners.

The structure of the paper is as follows. First we introduce the necessary terminology in Section 2. In Section 3 we show how to construct a $3\times3\times3$ sub-grid for each cell by dividing the cell into eight sub-cells, classify the new intermediate nodes as object or background, and give the surface model which can be used on the sub-grid and which results in surfaces which are topologically equivalent to those generated by DMC. In Section 4 we show that if we know a priori which features are concave and convex, we can modify the sub-grid and refine the surface model of Section 3. The paper concludes with results in Section 5 and a discussion in Section 6.

2 BASIC NOTIONS

We assume that data are represented on a regular discrete cubic grid which is called the data grid, say of size $x\times y\times z$. The x, y and z-axes of the grid form an orthogonal space such that xy-planes are always perpendicular to xz- and yz-planes and xz-planes to yz-planes. The grid contains $x\times y\times z$ cubic elements which are called voxels. The 6-neighborhood (respectively, 18-, 26-neighborhood) of a voxel at data grid position (x,y,z) is comprised by those voxels for which $|x-a|+|y-b|+|z-c|\leq 1$ (2, 3), with (a,b,c) arbitrary voxel coordinates. Two voxels are n-adjacent if

 ISBN 978-80-86943-02-2

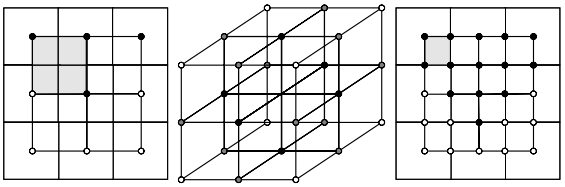

Figure 1: a) 3×3 data grid with 2×2 cuberilles b) 6-, 18- and 26-adjacency c) sub-grid with 4×4 cuberilles

they are *n*-neighbors. The neighborhood of a voxel is a 3×3×3 voxel block with the voxel itself in the center. In a voxel neighborhood the center voxel has 26 neighbors from which 6 voxels have a Manhattan distance of one, 12 voxels a distance of two, and 8 voxels a distance of three steps in orthogonal directions. At the data grid each voxel can be assigned a different data value. We limit ourselves to binary data values where 1 (black) denotes foreground/object and 0 (white) background.

Given the data grid we can define a cell grid with size $(x-1)\times(y-1)\times(z-1)$ in between the voxel centers such that the eight cell corners of each cell are at eight neighboring voxel centers (see Figure 1). The nodes at the cell grid uniquely correspond to the object and background voxels at the data grid. Therefore, the neighbor and adjacency definitions which have been defined for voxels also apply to nodes. Thus, we can speak of object (black) nodes and background (white) nodes. Configurations are unique patterns of black and white nodes in a cell. "Don't care" nodes will be used in configurations in order to indicate that these nodes may either belong to the object or the background. We define a higher resolution sub-grid by subdividing each cell into eight sub-cells (2×2×2). The new sub-grid nodes lay in between the grid nodes. There are eight unique sub-cells in the sub-grid of a cell. Original DMC space refers to the unique 2×2×2 nodes of cells. DMC sub-space refers to the unique 3×3×3 sub-grids of cells.

A path is a 6-path (respectively, 18-, 26-path) if it is a sequence of nodes $n_0...n_{n-1}$ on the cuberille grid such that n_i is 6-adjacent (18-, 26-adjacent) to n_{i-1} for $i=[1,n-1]$. A path is called a closed path if $n_0=n_{n-1}$. Two nodes are, respectively, 6-, 18- or 26-connected if there exists a 6-, 18- or 26-path between them. A component is a set of nodes and is, respectively, 6-, 18- or 26-connected if every pair of nodes in the component is 6-, 18- or 26-connected.

The DMC method generates a surface in between the object and background voxels through the nodes of the sub-grid. There are 256 possible patterns of black and white nodes in a cell which define 16 unique configurations, each with a corresponding triangulation. We call the set of these configurations a surface model. A triangulation is defined by the mapping of all cells at the cuberille grid with a surface model and yields a surface.

It is possible to generate many different topologically surface models, either based on (6,26), (6,18), (18,6) and (26,6) connectivity [5, 1] or hybrids. Kenmochi et al. [3] introduce a hybrid surface model that only uses 26-connectivity between nodes if it is part of a 3D-simplex, i.e. if there is an alternative 6/18-path between the nodes in a cell. In [6] we introduce a surface model that only has 18-connectivity if there exists an alternative 6-path between nodes in a cell. The Kenmochi model gives "priority" to the object and our model gives priority to the background. In [6] it is shown that the DMC triangulation is based on the background priority model.

In this paper we show that by performing the triangulation on the sub-grid (cuberille) level, we can combine Object Priority (OP) and Background Priority (BP) triangulation schemes within one object. This allows us to use different triangulation schemes for convex and concave edges and vertices. In this way we can maintain sharp features within objects.

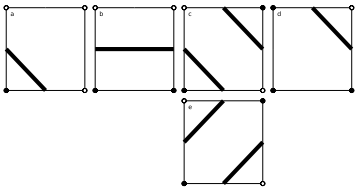

Figure 2: The original DMC configurations can be derived from the face configurations (*a-d*). The alternative "object priority" version can be derived with *e* instead of *c*

The triangulation patterns for the configurations of the Object- and Background-Priority models can be built from reduced sets of basic triangulation patterns that use don't care nodes for positions that can either be black or white. Figure 3 shows the basic patterns for the OP model and Figure 4 for the BP model. We note that in each configuration of Figure 3 with don't cares (K4-K7), at least one of the don't cares must be black.

3 SIMPLIFIED DMC SURFACE MODEL

In this section we first derive the connectivity model whereupon DMC is based. Then, we give a construction algorithm with which the sub-grid can be computed for any cell. Finally we show that the BP model can be used as a simplified surface model with which the DMC surface model can be generated on the sub-grid.

3.1 Interpretation of DMC

Color plate 1 shows the lookup table (lut) of the original DMC algorithm. If we don't pay attention to the gray and red spheres, the lut is equal to the one in DMC [8]. It contains 16 configurations. It shows all black-white combinations which are possible in a cell and their surface maps, except for inverse cases which

 ISBN 978-80-86943-02-2

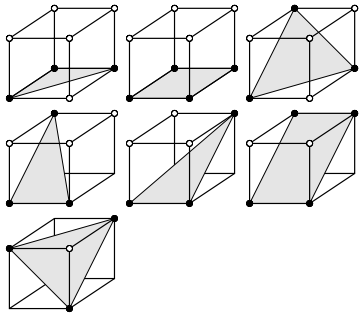

Figure 3: The seven configurations of the Object Priority model denoted with object nodes as black spheres, background nodes as white spheres and don't care nodes without spheres. The bottom left node in the back which is not always visible either belongs to the object (configuration 3) or is a don't care node (configurations 4 to 7)

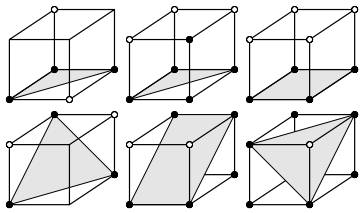

Figure 4: The six configurations $T1 - T6$ of the simplified surface or Background Priority model which can be applied in DMC sub-space. The bottom left node in the back now always belongs to the object. These configurations can be derived from the original DMC method (Color plates 1 and 2)

are obtained by swapping the black nodes to white ones and the white nodes to black ones. The inverse configurations of the configurations with four black and four white nodes (*h*-*m*) yield symmetric triangulations which can be found by rotation and projection of *h* to *m*. The inverse configurations of *a*, *b*, *d* and *e* also yield symmetric triangulations. These cases can be seen in Color plate 2. The inverse configurations of *c*, *f* and *g* are *o*, *p* and *n*, respectively. The triangulation of these cases is not symmetric: *g* and *n*, *c* and *o*, and *f* and *p* yield different triangulations for black-white compared to white-black patterns. This shows that DMC implements (6,18)-connectivity. For instance the (18,6)-connectivity model would connect the two 6-connected components which are 18-adjacent in configuration *n*.

We may observe that the authors of the DMC method gave priority to the background (6,18) instead of to the object, because otherwise the configurations would have been based on face configuration *e* in Figure 2 instead of on *c*. Of course, it is arbitrary whether we choose the one over the other, but it is necessary to be consistent in order to avoid ambiguous cases.

The connectivity is (6,18) and not (6,26) otherwise the inverse version of configuration *d* (Color plate 2: s) would have a white diagonal (tunnel) which is not the case. The (6,18)-connectivity of the DMC surface model reduces the chances that non-manifold situations do occur. For instance (18,6)-connectivity would create non-manifold situations in configurations *l* and *n*. It appears that the BP-triangulation on the high-resolution sub-grid complies with the (6,18)-node connectivity of the DMC model. However, it is clear that the BP model is not always the best choice, as in some situations the OP surface model would better represent the object. For example, if one wants to generate a convex edge in the triangulation of configuration *t*, the OP model would be a better choice.

In summary, with regard to the original DMC configurations (disregard gray and red spheres in Color plate 1), we can make the following observations:

- the DMC surface model is topologically equivalent to MC [9] (two shapes are topologically equivalent if they can be deformed into each other by a continuous, invertible mapping [2])
- in the DMC surface model, priority is given to the background instead of to the object. The a priori topology corresponds to (6,18)-connectivity, i.e. 6-connectivity for the object and 18-connectivity for the background
- object components which are 18- or 26-adjacent get partitioned into separate 6-connected components

Application of the DMC surface model yields a manifold surface for each 6-connected component of the object (recall the definition of a 6-connected component from Section 2). We see that DMC generates a surface in between the white and black nodes. The surface passes through the nodes of the sub-grid (red nodes). In this way, also for singular object nodes there always is a closed manifold surface around the node. So each 6-connected component is embedded in a manifold surface even if it is one voxel thin. In the DMC-method the sub-grid is implicitly used but never explicitly generated. We will propose a new surface model that generates the DMC configurations from the sub-grid explicitly. Before we can apply the new surface model to each individual sub-cell, we first have to classify the intermediate sub-grid nodes as being black or white.

3.2 Sub-grid construction

To create a classified sub-grid we first must initialize a sub-grid of size $3\times3\times3$ with white nodes. Then, we copy any black nodes at the cell to the corresponding nodes at the sub-grid. Last, we determine which nodes in the rest of the sub-grid must be set to black. The latter can be done in a two-pass process in which we first interpolate along the 6- and 18-edges/diagonals and then extrapolate from black nodes to 6-neighboring white nodes.

 ISBN 978-80-86943-02-2

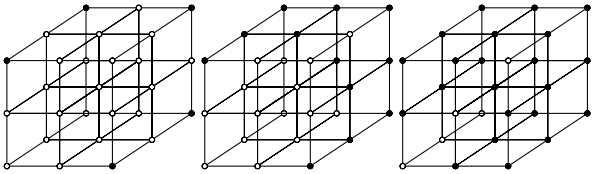

Figure 5: Example of the sub-grid construction algorithm: sub-grid after copying black nodes (a), interpolation (b) and extrapolation (c)

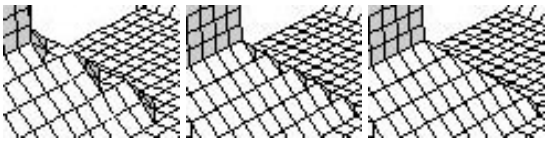

Figure 6: Example of surfacing variants at convex diagonal edges between 26-adjacent black nodes, from left to right: original surface, surface with BP model after extended interpolation, surface with OP model after extended interpolation

Figure 5 exemplifies the sub-grid construction for one configuration (a projected version of DMC-p). The construction algorithm sets for a cell with any input data the nodes at the sub-grid such that it exactly corresponds to DMC (as can be verified in Color plate 1).

3.3 Simplified surface model

Now we can define a new surface mapping in the high-resolution sub-grid of a cell. It turns out that we can do the mapping independently in each of the eight octants of the $3\times3\times3$ sub-grid and build the DMC triangulation by applying the BP surface model to each individual sub-cell [6]. Since Color plates 1 and 2 list all possible black-white combinations in a cell other configurations than those listed in Figure 4 are not possible in DMC sub-space.

Application of the BP surface model (Figure 4) in DMC sub-space yields surfaces which are topologically equivalent to the ones generated by the surface mapping with the DMC lut in original DMC space.

4 REFINED SURFACE MODEL

A drawback of the DMC model is that it always generates oblique surface patches also at edges and corners where a sharp convex or concave edge or corner are desirable. Another drawback is the stair casing at the missing 26-diagonals. The latter we can repair by locally applying the OP surface model (see Figure 6). Sharp edges can be introduced by extending the sub-grid in the convex case and refining the BP surface model in the concave case. Thus, we propose two extensions to standard DMC surfacing:

1. adding extra black nodes and applying the OP surface model for convex edges and corners
2. applying a refined BP surface model for concave edges and corners

We will consider three different types of object edges and two different types of object corners: (1) diagonal edges which are not aligned with the 3D grid between two 26-adjacent black nodes in a cell; (2) diagonal edges which are not aligned with the 3D grid between two 18-adjacent black nodes in a cell face; (3) right edges which are aligned with the 3D grid between two 6-adjacent black nodes on a cell edge; (4) right corners which are aligned with the 3D grid between three 6-adjacent black nodes; (5) diagonal corners which are not aligned with the 3D grid between three 18-adjacent black nodes.

There are geometrically three different situations in which the output surface can be adapted to become more concave or more convex: (A) At an edge node in a cell face, i.e. a black node which has one 18-adjacent white node in a face and two 6-adjacent black nodes in the same face, the surface can be modeled inwards the object to the edge node for a concavity, or outwards the object to the white node in order to make it more convex. (B) The same holds for a corner node in a cell, i.e. a black node with a 26-adjacent white node. See for example $T4$ and $T6$ in Figure 4. In $T6$ we might know that the (hidden) black node at the bottom left is a concavity in the object. In this case the standard triangulation could be adapted to form a sharp point towards that node. Also, if the don't care node at the top in $T4$ is black (and the other white) and we know that that node is a convexity of the object, the triangulation could be adapted similarly to form a sharp point outwards. (C) At a diagonal edge between two 18-adjacent black nodes in a cell face or a diagonal edge between two 26-adjacent black nodes in a cell, the two black nodes are never directly connected because in DMC preference is given to the white diagonal. However, if the two black nodes form a convex object edge they should be connected and the triangulation should be directed towards the black diagonal. Figure 6 exemplifies this.

In the concave case, we must refine the standard triangulation patterns ($T1$-$T6$) so that the surface passes correctly through the concave edge/corner nodes. In the convex case, we first must modify the sub-grid by propagating the convex edge/corner nodes and black diagonals on the grid and then substitute the standard triangulation patterns of the BP model by the convex patterns of the OP model. In both cases, under strict conditions it is allowed to locally deviate from the "default" BP surface model; see Sections 4.2 and 4.3.

In this section we will first define some local filters for detecting sharp object features. Then we describe the modifications which are necessary in the sub-grid construction and in the triangulation, first for the convex case and then for the concave case.

4.1 Feature detection

Feature detection can be supervised or unsupervised, depending on whether a user or an automatic method indicates which object edges and corners must be triangulated convex and concave instead of oblique. In our framework different algorithms can be incorporated because it does not depend on a specific edge and/or corner detection algorithm. In order to test our triangulation methods we developed some simple ad hoc filters for detecting convexity in the local neighborhood of a node on the sub-grid. We limited ourselves to testing for convexity in nine of the possible planes which can be defined in a 3D grid and which form themselves a regular rectangular 2D grid. The filters have the advantage that unwanted aliasing is not possible.

Two passes are needed: a first one for detecting convex diagonal edges and a second one for detecting convex right edges. Convex diagonal edges are detected with four filters; see the first two lines in Figure 7. The filters are asymmetrical with respect to the object and background, because the a priori topology corresponds to $(6,18)$-connectivity (recall Section 3.1). In the filters to the left the background is convex to the top and the object concave to the bottom. In the filters to the right the object is convex to the top and the background concave to the bottom.

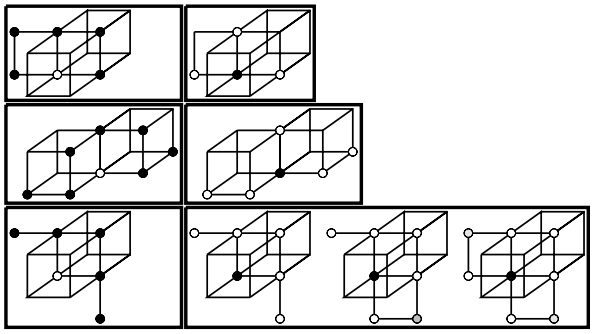

Figure 7: Feature detectors

Convex right edges are detected with four filters; see the third line of Figure 7. In the first filter the background is convex to the top-right and the object concave to the bottom-left. In the second, third and fourth filter the object is convex to the top-right and the background concave to the bottom-left. The first two filters in the third line are symmetrical with respect to the object and background: the object is 6-connected in the first filter and the background is 6-connected in the second filter. However, because the a priori topology is $(6,18)$-connectivity, two other filters are possible in which the background is 18-connected. In these filters, the gray nodes may not be black and form a diagonal edge with the black node in the middle, because then preference is given to the black diagonals (as detected with the filters for diagonal edges) and, consequently, the black node in the middle can not be a convex right edge.

After detecting convex diagonal and right edges, we use the following strategy for determining convex diagonal and right corners. Nodes which have been labeled as a convex right edge in three different faces of a single cell are labeled as convex right corners. Nodes which have been labeled as a convex diagonal edge in three different directions in the same cell are labeled as convex diagonal corners. Other combinations of convex right and diagonal edges are possible in a corner configuration, but we limit ourselves to these two cases.

In practice, we only store labels at convex nodes (black or white). The concavities are then represented by their convex counterparts. This makes the filters locally applicable in the $3\times3\times3$ neighborhood of a node.

4.2 Convexity

Object convexity is achieved by modifying the sub-grid construction algorithm such that the nodes on the sub-grid which correspond to convex black edges and corners are swapped from the background to the object. This is done by interpolating along convex diagonal edges and extrapolating at convex right edges and convex diagonal and right corners. Since extrapolation is only permitted if the topology remains unchanged, we use topology preservation masks, in order to guarantee that separate black 6-components cannot get connected and connected white 18-components cannot get separated.

For sub-cells with interpolated black diagonal edges or with extrapolated black diagonal corners, the object triangulation is done with the OP surface model and not with the BP model. The OP model yields convex triangulations for all configurations with black diagonal edges and corners. Mixing both models is allowed because the closed paths which are defined by the surface triangulations in all configurations are always equal in the BP and the OP surface model [6].

4.3 Concavity

Starting from the refined representation on the sub-grid of the object (black), the background (white), and the labeled white nodes (labels indicate the directions in which the background is convex), we can refine the simplified surface model as follows: at a concave right black edge, i.e. a convex right white edge, do not generate the standard oblique patch but a sharp one by forcing the surface patch into the object. Figure 8 illustrates the idea. A similar line of reasoning holds for concave black (diagonal or right) corners, i.e. convex white corners. For each configuration in the simplified surface model different surface patches are possible depending on the labels of the white nodes, i.e. on the directions in which the background is convex. If we order all cases systematically we come to the lut depicted in Color plate 3.

 ISBN 978-80-86943-02-2

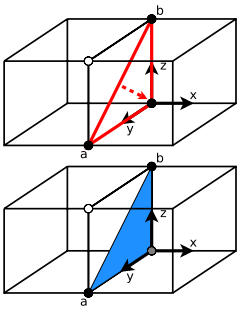

Figure 8: At a concave edge node in DMC sub-space (top figure) we want to force the triangulation inwards the concavity to $(0,0,0)$, i.e. build it not on the straight 18-path between *a* and *b* but on the hooked 6-path between *a* and *b* via $(0,0,0)$ (depicted in red). If there is no concavity detected in the *yz*-plane for the edge node at $(0,0,0)$, the triangulation in the two sub-cells will not be refined in the respective *yz*-quadrant and the blue area will remain inside the triangulated object volume (in this case the simplified surface model is applied without any changes). If a concavity is detected, the triangulation in each of the two sub-cells will be refined in the respective *yz*-quadrant and the blue area will be outside the object and the object volume will, depending on the settings in the sub-cells, decrease (in this case a refined surface model must be applied; see Color plate 3). The triangulation at an edge node can be made concave without altering the closed path in the two sub-cells which are fixed to the edge node

We can prove that the surfaces created by this refined surface model are topologically equivalent to the ones created by the simplified surface model. The proof consists of two parts. First, we prove that for each surface refinement, the closed path remains equal to that of the original surface. Color plate 3 shows that this is indeed the case. Thus, no gaps can occur and the topology is identical. Second, we prove that the output surface remains manifold. For the gray patches in Color plate 3 this holds because (due to the sub-grid construction) edge nodes on different sides of an object are always separated by two sub-cells in DMC sub-space. For the colored patches in Color plate 3 this does not hold, but for these patches we can prove that they can be removed from the refined surface model. See again Figure 8. When we regard the interface between two sub-cells, the blue triangle can always be removed without altering the closed path of the triangulation in the two cells. Thus, we can replace the lut of Color plate 3 by another lut in which all colored triangles are removed. This is the final surface model which can be applied in DMC sub-space in order to preserve concave right object edges and concave object corners.

5 RESULTS

Figures 9, 10, 11 and 12 show output surfaces generated by detecting the different types of edges and corners for various objects. The objects were grouped into three different classes: sharp objects with salient features (Figures 9 and 10), thin objects with parts which are only one voxel thick (Figure 11) and an object with diagonal edges and corners (Figure 12). In the displays, we note that each vertex corresponds to one node of the sub-grid and that, for clarity, we did not pack together any coplanar patches.

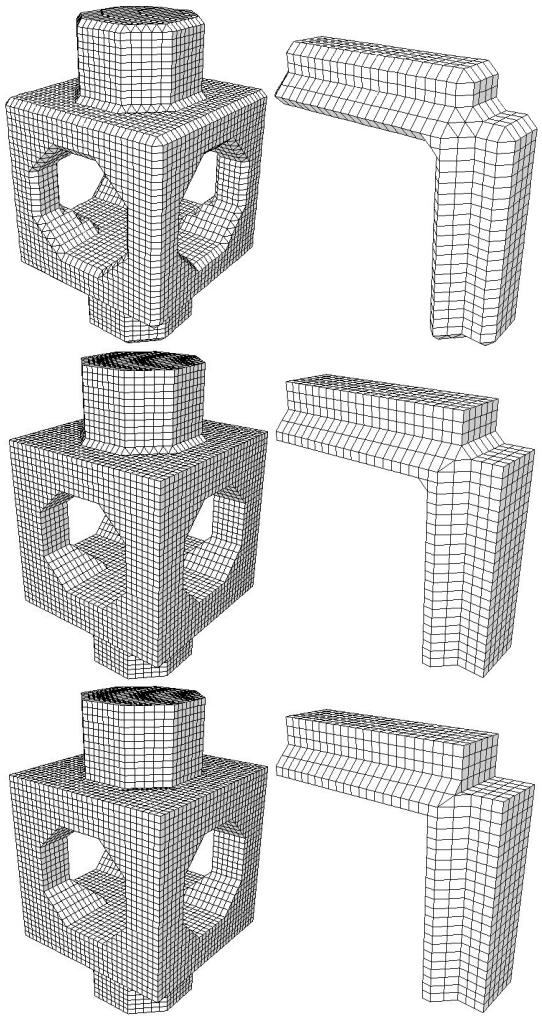

Figure 9: From top to bottom: original surface in DMC sub-space according to standard DMC; surface generated with the OP surface model at convex diagonal edges and corners; surface generated with the refined BP surface model at concave edges and corners

The surfaces at the top were obtained by applying the BP model at the sub-grid, without doing any feature detection. These surfaces are in agreement with those generated by the standard DMC method. The second rows show the surfaces when we do not always apply the BP model, but also the OP model at detected con-

 ISBN 978-80-86943-02-2

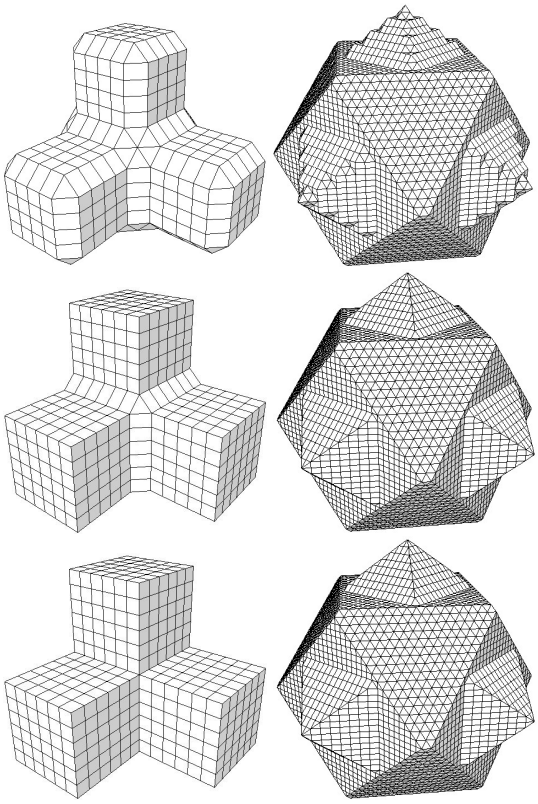

Figure 10: From top to bottom: original surface in DMC sub-space according to standard DMC; surface generated with the OP surface model at convex diagonal edges and corners; surface generated with the refined BP surface model at concave edges and corners

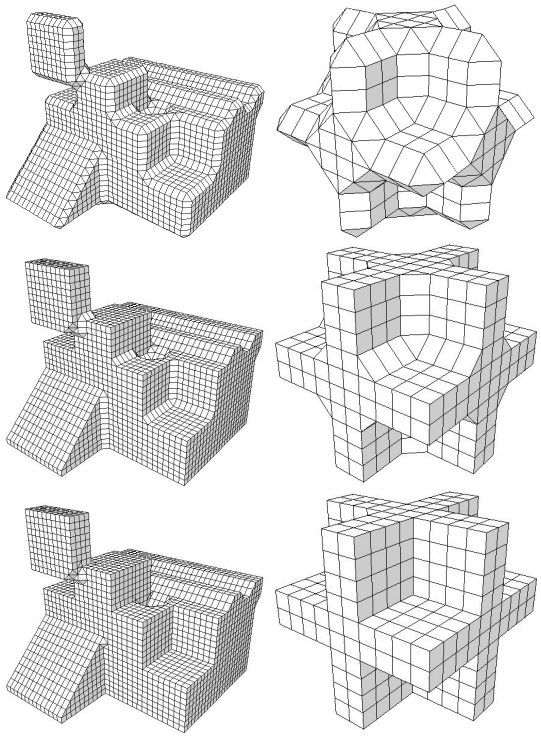

Figure 11: From top to bottom: original surface in DMC sub-space according to standard DMC; surface generated with the OP surface model at convex diagonal edges and corners; surface generated with the refined BP surface model at concave edges and corners

vex diagonal object edges and corners. The third rows show the surfaces when we also detect concave edges and corners and apply the refined BP model.

The figures show that the methods always create manifold output surfaces, also for the thin objects in Figure 11. Examples of the preservation of convex right edges and convex right corners can be seen in all figures. Examples of the preservation of concave right edges can be seen in Figures 9, 10 and 11. Note that for the first object in Figure 11 concave right edges are visible, but not in the middle where the concave right edge which comes from the bottom of the object ends. Here, the concavity could not be detected because there are only four black nodes which lie in the horizontal plane with the white node (this situation can be ambiguously interpreted as a concave edge or as an object slope) and the triangulation is accordingly adapted. Examples of the preservation of concave right corners can be seen in Figure 11. Examples of the preservation of convex diagonal edges can be seen in Figures 9, 10 and 12. Note that also for the second object in Figure 9 a convex diagonal edge was detected. Also note, by comparing the output surfaces at diagonal edges with the original surfaces in the top rows, that the aliasing at the edges has completely disappeared. Examples of the preservation of a concave and a convex diagonal corner can be seen in Figure 12. At the convex diagonal corner, the two 6-components of the object are connected to form the sharp point.

6 DISCUSSION

In this paper it has been shown that we can simulate DMC surfaces by applying a simplified surface model (the BP model) in DMC sub-space and can: (A) explicitly represent convex right and diagonal edges and corners on the sub-grid by using an extended sub-grid construction algorithm, (B) derive a refined surface model for concave right edges and corners and concave diagonal corners, and (C) deviate from (6,18)-connectivity and the BP model and locally use (18,6)-connectivity and the OP model to represent convex diagonal edges and corners. The advantage of working in DMC sub-space is that there is room to build surfaces (oblique or sharp) which are guaranteed to be manifold for any input data.

In practice, when working locally on a cell-by-cell basis, it is not needed to store the entire high-resolution data grid of size $(2x-1)\times(2y-1)\times(2z-1)$. Also, because the original DMC lut is computationally cheaper, this table should be run when there are no features whatsoever in a cell. Only in the case that a feature is present, the cell should be opened up into its eight

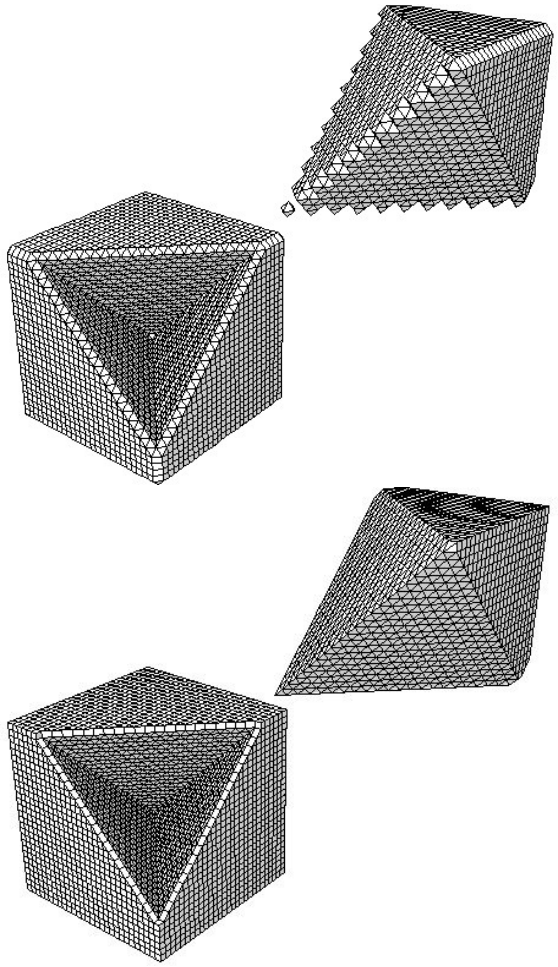

Figure 12: Original surface in DMC sub-space according to standard DMC (top) and surface generated with our method (bottom)

sub-cells. After building the sub-grid for the cell, the sub-cells are mapped with the BP model in the case of sub-cells without features and with the refined BP or OP model in the case of sub-cells with features.

Another important issue is the number of output triangles for an object, as this is a determinant factor for the fast rendering of surface models. The authors of the DMC algorithm show in a more recent paper [10] that the number of generated triangles can be optimized by using a pyramid structure in which all coplanar triangles are packed into single output polygons. Therefore, the main focus of this paper does not have to lie on minimizing triangle counts. In fact, as the sharpness of the object and the object topology is improved by our method, the number of output polygons will decrease, because oblique polygons at rounded edges and corners and incorrect polygons at aliased diagonal object edges will all be adapted and included into the relevant polygons which determine the true shape of an object. This can be illustrated by the very simple example of a cube which is aligned with the 3D grid. With our method the output surface will consist of six polygons, one for each face of the cube, without the incorrect polygons which are normally generated at the eight corners and the twelve edges of the cube.

In the future we will look at application of the framework for isosurfacing grayvalue data. Interpolated isosurfaces can be built by positioning nodes in DMC sub-space, not in the middle between node pairs, but, depending on the grayvalues which correspond to a node pair, along the straight trajectory between the nodes. Node pairs need not always correspond to two 6-neighbors (as in standard MC), but can also correspond, for right object edges, to two 18-neighbors, or, for object corners and for diagonal object edges, to two 26-neighbors. Nodes can be positioned along the entire trajectories without losing the guarantee of surface manifoldness.

REFERENCES

[1] Carlos Andujar, Pere Brunet, Antoni Chica, Isabel Navazo, Jarek Rossignac, and Alvar Vinacua. Optimal iso-surfaces. *J. of Computer-Aided Design and Applications*, 1(1-4):187–196, 2004.

[2] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. In *Proc. SIGGRAPH*, pages 339–346, 2002.

[3] Y. Kenmochi, A. Imiya, and A. Ichikawa. Boundary extraction of discrete objects. *Computer Vision and Image Understanding*, 71(3):281–293, 1998.

[4] Leif P. Kobbelt, Mario Botsch, Ulrich Schwanecke, and Hans-Peter Seidel. Feature sensitive surface extraction from volume data. In *Proc. SIGGRAPH*, pages 57–66, 2001.

[5] J. O. Lachaud and A. Montanvert. Continuous analogs of digital boundaries: A topological approach to iso-surfaces. *Graphical Models and Image Processing*, 62:129–164, 2000.

[6] R. E. Loke, F. W. Jansen, and J. M. H. du Buf. A background-priority discrete boundary triangulation method. In *Short communication proc. WSCG (Winter School on Computer Graphics)*, Plzen, Czech Republic, Jan.-Feb. 2006. UNION Agency-Science Press.

[7] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[8] C. Montani, R. Scateni, and R. Scopigno. Discretized marching cubes. In R. D. Bergeron and A. E. Kaufman, editors, *Proc. Visualization '94*, pages 281–287, Washington D.C., USA, 1994.

[9] C. Montani, R. Scateni, and R. Scopigno. A modified look-up table for implicit disambiguation of marching cubes. *The Visual Computer*, 10(6):353–355, 1994.

[10] C. Montani, R. Scateni, and R. Scopigno. Decreasing isosurface complexity via discrete fitting. *Computer Aided Geometric Design*, 17:207–232, 2000.

[11] H. Müller and M. Stark. Adaptive generation of surfaces in volume data. *The Visual Computer*, 9(1):182–198, 1993.

[12] G.M. Nielson and B. Hamann. The asymptotic decider: Resolving the ambiguity in marching cubes. In IEEE Computer Society Press, editor, *Proc. of Visualization '91*, pages 83–91, Los Alamitos (CA), USA, 1991.

[13] R. B. Shu, C. Zhou, and M. S. Kankanhalli. Adaptive marching cubes. *The Visual Computer*, 11(4):202–217, 1995.

[14] A. van Gelder and J. Wilhelms. Topological considerations in isosurface generation. *ACM Trans. Graphics*, 13(4):337–375, 1994.