Topology Preserving Marching Cubes-like Algorithms on the Face-Centered Cubic Grid

Robin Strand
Centre for Image Analysis, Uppsala University,
Box 337, SE-75105 Uppsala, Sweden
robin@cb.uu.se

Peer Stelldinger
University of Hamburg
22527 Hamburg, Germany
stelldinger@informatik.uni-hamburg.de

Abstract

The well-known marching cubes algorithm is modified to apply to the face-centered cubic (fcc) grid. Thus, the local configurations that are considered when extracting the local surface patches are not cubic anymore. This paper presents three different partitionings of the fcc grid to be used for the local configurations. The three candidates are evaluated theoretically and experimentally and compared with the original marching cubes algorithm. It is proved that the reconstructed surface is topologically equivalent to the surface of the original object when the surface of the original object that is digitized is smooth and a sufficiently dense fcc grid is used.

1 Introduction

The Marching Cubes algorithm introduced in [5] is a well-known algorithm for surface extraction. The basic idea on the cubic grid is to consider $2 \times 2 \times 2$ configurations of grid points. Each of the local configurations results in a set of (planar) surface patches. The union of the surface patches is a representation of the surface of the object.

Algorithms based on the marching cubes algorithm (marching-algorithms) have been developed also for other grids than the cubic grid, for example the body-centered cubic (bcc) grid [1, 9] and the face-centered cubic (fcc) grid [8]. The packing density for the cubic, fcc, and bcc grid is 0.524, 0.741, and 0.680, respectively. Since the fcc and bcc grids are reciprocal (an fcc grid in spatial domain corresponds to a bcc grid in frequency domain), only (in higher precision) $0.524/0.741 = 0.7071$ and $0.524/0.680 = 0.7698$ of the number of samples in the cubic grid are needed for the bcc and fcc grids respectively to fulfill the Shannon sampling theorem. See also [4]. Since less samples are needed, using these grids in image analysis and processing is potentially more efficient. Since interpolating between different grids destroys information, it is preferable that the image is acquired, processed, and visualized using the same grid. The goal here is not to compete with the cubic grid, but to develop an image visualization method for the fcc grid as a step towards making it possible to do image processing directly on non-standard grids without any resampling to the cubic grid.

The surface generated by the original marching cubes-algorithm is not topologically consistent – some local configurations result in holes in the surface. Many approaches to avoid the holes have been presented. One efficient way to do so is to divide each cube resulting from the $2 \times 2 \times 2$ configurations into five tetrahedra [3]. (A tetrahedra decomposition on the bcc grid is presented in [2, 9].) The low number of configurations with tetrahedra does not include any critical cases, and the surface generated by marching tetrahedra does not contain any holes.

In this paper, three marching-algorithms (using three different sets of local configurations) on the fcc grid are presented. By assuming that the sampling grid is dense enough and that the boundary of the object is smooth some previous results from [7] guarantee that the surface generated by the marching-algorithms is topologically equivalent to the boundary of the original object.

2 Preliminaries

A set $A \subset \mathbb{R}^3$ is r*-regular* if, for each boundary point $x \in \partial A$, there exist two tangent balls of radius r to ∂A at x s.t. one lies entirely in A and the other lies entirely in the complement A^c of A. Let S be a countable set $S \subset \mathbb{R}^3$ and r' the Hausdorff distance between $\mathbb{R}^3$ and S, $r' = d_H\left(\mathbb{R}^3, S\right)$. In other words, r' is the maximal radius of a ball in $\mathbb{R}^3$ whose interior does not intersect a point in S. If $r' < \infty$, we denote such a set S an r'*-grid*. We use the following definition of an fcc r'-grid:

$$\{t + r' \cdot R \cdot (x, y, z) : \\ x, y, z \in \mathbb{Z} \text{ and } x + y + z \equiv 0 \pmod 2\},$$

where t is a translation vector and R is a rotation matrix. The voxels (Voronoi regions) of fcc grids are rhombic dodecahedra, see Figure 1 and 2. For a grid S and an object A, $S \cap A$ is the set of object grid points and $S \cap A^c$ is the set of background grid points.

A bijective function $f : A \to B$ is a *homeomorphism* if both f and its inverse are continuous. The sets A and B are *homeomorphic* if such a function exists. A homeomorphism $f : \mathbb{R}^3 \to \mathbb{R}^3$ is an r-*homeomorphism* of A to B if $f(A) = B$ and the Euclidean distance $d(x, f(x)) \leq r$ for all $x \in \mathbb{R}^3$.

The following result is derived in [7]: Given an r-regular object A and an fcc r'-grid with $\sqrt{2}r' < r$, the voxel representation of A is $2r'$-homeomorphic to A. Two voxels sharing only a vertex (see the dark grey voxels in Figure 1) is called a critical configuration. Because of the critical configuration in Figure 1, there is not enough information to decide whether the voxels represent one object or two (or more) objects. When $\sqrt{2}r' < r$, no critical configurations will result from the digitization.

Critical configurations are obviously problematic also when extracting surfaces from digitized objects. If a surface representation should be topologically equivalent to the original object, no critical configurations are allowed. The ambiguity for the fcc grid is considered for the fcc marching-algorithm in [8]. The solution there is to consistently divide the critical configurations into two parts, since this gives the least number of triangles in the resulting surface. In Section 4, it is shown that the marching-algorithms presented in this paper produce surfaces that are $2r'$-homeomorphic to the original object if $\sqrt{2}r' < r$.

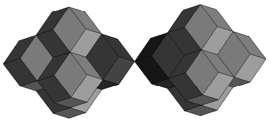


Figure 1. The dark grey voxels constitute a critical configuration.

3 Alternative Partitionings

When extracting surfaces with a marching-algorithm, local configurations of grid points are considered. From these configurations, it is decided which surface patch should be used to locally separate the background grid points from the object grid points. For the original marching cubes algorithm, the Delaunay mesh is used for the local configurations. This kind of dual is considered for configurations on the fcc grid in Section 3.1. This partitioning, consisting of tetrahedra and octahedra configurations, is used also in the marching-algorithm for the fcc grid in [8], see Figure 2(a) and (b). The Delaunay mesh of the bcc grid is considered for a marching-algorithm on the bcc grid in [1]. In Section 3.2, the octahedra in the Delaunay mesh are splitted into tetrahedra resulting in a partitioning with only tetrahedra configurations, see Figure 2(c) and (b). In the third partitioning presented in Section 3.3, a shifted fcc grid is used for the configurations resulting in rhombic dodecahedra configurations, see Figure 2(d).

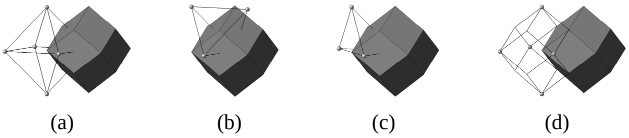


Figure 2. The configurations used in this paper. In this illustration, a voxel is used to represent one of the grid points in the configurations.

In the figures in this paper, black and white grid points correspond to object and background grid points, respectively. Observe that the complementary cases are not shown explicitly, but obtained by switching object grid points to background grid points and vice versa. The figures show all the possible configurations up to rotational symmetry, reflection symmetry, and complementarity.

3.1 Delaunay Mesh Partitioning

In this section, the Delaunay mesh is considered for the marching-algorithm. The Delaunay mesh of an fcc grid consists of octahedra and tetrahedra and the possible configurations are shown in Figure 3 and 4. This partitioning and the configurations in Figure 3 and 4 are also considered in [8].

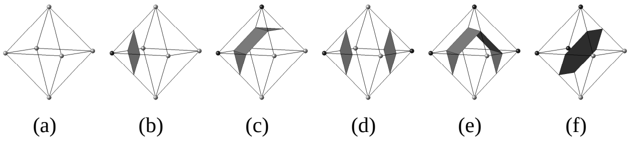


Figure 3. The octahedra cases in the Delaunay mesh.

3.2 Tetrahedra Partitioning

When using only tetrahedra configurations, the surface resulting from the marching-algorithm is topologically

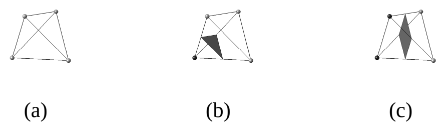


Figure 4. The tetrahedra cases in the Delaunay mesh cases and the tetrahedra partitioning.

sound. This is easy to verify due to the low number of possible configurations. Tetrahedra configurations have been examined for the cubic grid and the body-centered cubic grid, see [3] and [2, 9] respectively. One problem on the cubic grid is that in order to get a connected surface, neighbouring cubic configurations must be split into tetrahedra in different ways, see [3].

By splitting each octahedron in the Delaunay mesh into four tetrahedra, a marching-algorithm with only tetrahedra configurations is obtained. The splitting can be done in different ways leading to algorithms that extract different surfaces. Thus, different splittings lead to surfaces that are geometrically different. If case (d1) in Figure 6 is employed, the topology of the surface obtained by tetrahedra configurations is different from the surface obtained by octahedra configurations, see Figure 12 for an example. Here, the splitting shown in Figure 5 is used. The tetrahedra obtained are not regular (opposed to the tetrahedra in the Delaunay mesh). The local configurations are similar to the configurations in Figure 4. The different cases from this splitting are shown in Figure 6, where each configuration represents the four tetrahedra resulting from a octahedron are shown. Observe that, due to the splitting, different surface patches are extracted depending on the orientation of the configuration.

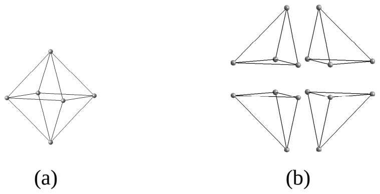


Figure 5. One octahedron divided into four tetrahedra.

3.3 Rhombic Dodecahedra Partitioning

A third partitioning is obtained by representing each set of six grid points forming octahedra in the Delaunay mesh by rhombic dodecahedra. In this way, no tetrahedron cases are needed since the rhombic dodecahedra are space-filling. The different cases are shown in Figure 7.

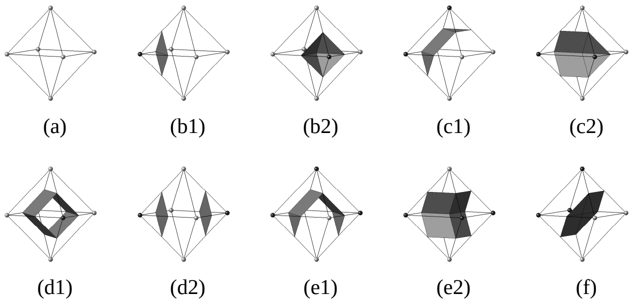


Figure 6. The octahedra cases when splitted into tetrahedra.

Observe that, extra rules must be applied to configuration (e) in Figure 7 in order to avoid non-manifold surfaces. Both vertex p_3 and p_6 correspond to object grid points. The non-manifold cases in Figure 8 are avoided if the angle between the vector $R(1,1,1)$ (R is the rotation matrix used in the definition of the fcc grid) and the normal of the rhombus with two object grid points whose interior intersects the surface is as small as possible. In Figure 7(e), either (the interior of) the rhombus including p_3 and p_6 or the rhombus including p_5 and p_6 can be choosen to intersect the surface. (Or, by complementarity, the rhombus including p_1 and p_2 or the rhombus including p_1 and p_4.)

In Figure 13(e2), an alternative surface is shown for case in Figure 7(e). This surface can, however, not be used for r-regular objects with $\sqrt{2}r' > r$. This is due to the fact that configurations similar to the ones in Figure 8 can not be avoided, so there would be cases leading to non-manifold surfaces. As we will see in Section 4, the surface patches in Figure 13(e2) can be used if $\sqrt{2}r' < r$.

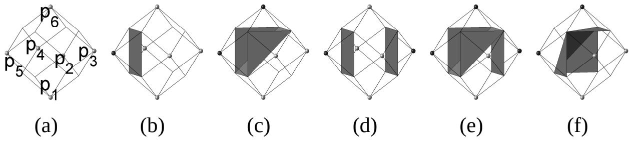
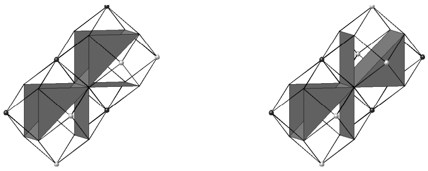


Figure 7. The rhombic dodecahedra cases.

4 Topology Preservation

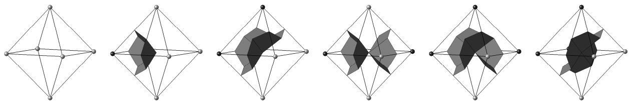
Theorem 1 *The Marching-algorithms presented here are all topology preserving reconstruction methods and the results of the algorithms are $2r'$-homeomorphic to the original object if A is r-regular and the sampling grid is an fcc r'-grid with $\sqrt{2}r' < r$.*



(a) (b)

Figure 8. These configurations lead to non-manifold surfaces. This is avoided by introducing an extra condition.

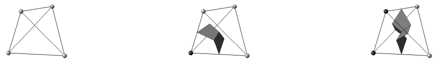
Proof: In [7], it is proved that the reconstruction of an r-regular object on an fcc r'-grid with $\sqrt{2}r' < r$ using voxel representation is $2r'$-homeomorphic to the original object. The surface obtained from the voxel representations of the configurations in Figure 3, 4, 6, and 7 are shown in Figure 9, 10, and 11. First of all, for an r-regular object in an fcc r'-grid with $\sqrt{2}r' < r$, the cases (d) can not occur [7]. It is easy to verify that for all the remaining cases, the surface patches in Figure 3, 4, 6, and 7 separate the grid points into object/background grid points in the same way as the surface patches in the corresponding cases in Figure 9, 10, and 11. Also, the local surface patches in the marching-algorithms presented here are such that the resulting surface is not self-intersecting *and* contains no holes between the configurations. Finally, all surface patches except (d) in Figure 3, 4, 6, and 7 are homeomorphic to the surface patches in the corresponding cases in Figure 9, 10, and 11. Thus, following the proof idea in [7], the surfaces obtained from the marching-algorithms presented here are $2r'$-homeomorphic to the original object *if* the surface obtained by the voxel representation is. This is the case when $\sqrt{2}r' < r$, so the theorem holds. □



(a) (b) (c) (d) (e) (f)

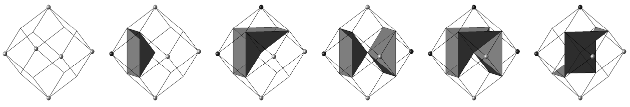
Figure 9. Surface reconstruction with voxels for the octahedra in the Delaunay and tetrahedra partitionings

The difference in topology of the reconstruction for an object that is not r-regular with $\sqrt{2}r' < r$ is shown in Figure 12, where the surface of the object with a critical configuration in Figure 1 is generated using the three different marching-algorithms. For the tetrahedra partitioning, the surface of a single connected object is generated. For the other two algorithms, the surface of two connected objects are extracted from the algorithm.



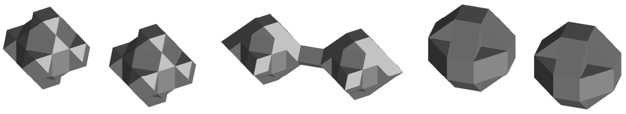
(a) (b) (c)

Figure 10. Surface reconstruction with voxels for the tetrahedra in the Delaunay and tetrahedra partitionings



(a) (b) (c) (d) (e) (f)

Figure 11. Surface reconstruction with voxels for the rhombic dodecahedra partitioning



(a) (b) (c)

Figure 12. Surface reconstruction of the object in Figure 1 using (from left to right) Delaunay partitioning, tetrahedra partitioning, and rhombic dodecahedra partitioning.

Without extra conditions, the pairs of configurations in Figure 8 would lead to self-intersecting surfaces (sharing an edge) resulting in a surface that is not a 2D-manifold. The transparent balls in Figure 13(a) and (b) have radii $\sqrt{2}r'$. The figures show that the surface of the ball intersects with both background and object grid points and since the object is assumed to be r-regular with $\sqrt{2}r' < r$, there can not be an object that has the black grid points in the interior without also having any of the white grid points in its interior. Therefore, these configurations can not occur when $\sqrt{2}r' < r$ and the extra condition introduced to avoid non-manifold surfaces is not needed. Also, since the configuration can not occur in pair as is shown in Figure 8 and 13(a) and (b) when $\sqrt{2}r' < r$, the surface patches (e2) shown in Figure 13 can be used instead of (e) in Figure 7.

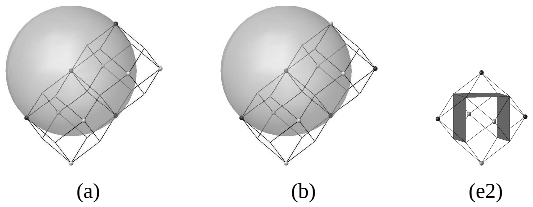


Figure 13. (a) and (b): These configurations lead to non-manifold surfaces. This is avoided by introducing an extra condition. (e2): An alternative configuration that can be used instead of (e) in Figure 7 when $\sqrt{2}r' < r$.

5 Expected Number of Triangles

In this section, following [1], an estimate of the number of triangles needed to represent the surface of an object is applied to the three partitionings. A finite subset of a grid is called an *image*. The number of grid points in an image on a cubic grid is denoted N. The results will be compared to the marching cubes algorithm on the cubic grid. Therefore, we include in the calculations the fact that only 0.7698 samples are needed on the fcc grid to fulfill the Shannon sampling theorem compared to the cubic grid. The number of cells N_{cells} (i.e. polyhedra in the partitioning scheme) that intersect the object surface is approximated by the formula used in [1]:

$$N_{cells} = O\left(N^{\frac{2}{3}}\right). \tag{1}$$

The formula

$$N_{tri} = N_{cells} \cdot N_{tri/cell} \tag{2}$$

will be used to estimate the total number of triangles.

5.1 Delaunay Partitioning

In the Delaunay partitioning, each grid point is the vertex of 8 tetrahedra and 6 octahedra. Each octahedron has 6 grid points as vertices and each tetrahedron has 4 grid points as vertices. The average number of cells per grid point is $\frac{1}{6} \cdot 6 = 1$ octahedron plus $\frac{1}{4} \cdot 8 = 2$ tetrahedra. The total number of cells are $(1 + 2) \cdot 0.7698N$ (observe that N is the number of grid points in a cubic grid). By using (1), $N_{cells} = ((1 + 2) \cdot 0.7698N)^{\frac{2}{3}} \approx 1.7472N^{\frac{2}{3}}$ of these intersect the surface.

The average number of triangles per octahedron is $\frac{12 \cdot 2 + 24 \cdot 4 + 6 \cdot 4 + 12 \cdot 6 + 8 \cdot 4}{62} = 4$ and the average number of triangles per tetrahedron is $\frac{8 \cdot 1 + 6 \cdot 2}{14} \approx 1.43$. The expected number of octahedra and tetrahedra per grid point is 1 and 2, respectively. Therefore, the expected number of triangles per cell is $N_{tri/cell} = \frac{2 \cdot 1.43 + 1 \cdot 4}{3} \approx 2.2857$. By (2), the expected number of triangles is now $N_{tri} = N_{cells} \cdot N_{tri/cell} = 1.7472N^{\frac{2}{3}} \cdot 2.2857 \approx 3.9935N^{\frac{2}{3}}$.

5.2 Tetrahedra Partitioning

By the octahedra configurations in Figure 6, the average number of triangles per octahedron is now 5.1613 which results in $N_{tri} \approx 4.6698N^{\frac{2}{3}}$.

5.3 Rhombic Dodecahedra Partitioning

Following the calculations above, $N_{tri} \approx (1 \cdot 0.7698N)^{\frac{2}{3}} \cdot 4 \approx 3.3598N^{\frac{2}{3}}$.

6 Experimental Results

For an image (a finite subset of the grid points) on the fcc grid with $N_{fcc} = 16384$ grid points, the number of triangles needed to represent the surface of a ball and a torus is computed. Balls and tori of 20 different radii and major radii (the minor radius is kept fixed for the tori), respectively, are considered. The average number of triangles needed to represent the balls and the tori are calculated for the three marching-algorithms. The numbers are shown in Table 1.

The resulting surfaces from the three algorithms when applied to the digitization of a potplant is shown in Figure 14.

7 Discussion and Conclusions

For the topology-preserving marching cubes algorithm presented in [6], the expected number of triangles is $N_{tri} = 2.85N^{\frac{2}{3}}$. This should be compared with the algorithms on the fcc grid presented here: $3.99N^{\frac{2}{3}}$, $4.67N^{\frac{2}{3}}$, and $3.36N^{\frac{2}{3}}$ for Delaunay, tetrahedra, and rhombic dodecahedra partitiong, respectively. Thus, when considering the Shannon sampling theorem, the algorithms we have presented for the fcc grid are less efficient compared to the marching cubes algorithm on the cubic grid. The conclusion from the estimation of the number of triangles needed for a marching-algorithm on the bcc grid in [1] is that also the bcc grid is, in this sense, less efficient compared to the cubic grid. Since the bcc grid needs only $0.680/0.741 \approx 0.92$ times the number of samples needed for the fcc grid [4] to fulfill the (Shannon) sampling theorem, the results for the bcc grid presented in [1] are slightly better than the results for the fcc grid.

If the topological sampling theorems for the fcc grid and the cubic grid instead are considered, an r-regular object implies that an fcc r'-grid with $\sqrt{2}r' < r$ ([7]) and a cubic

Table 1. Average number of triangles for balls and tori of different radii and major radii, respectively.

	Delaunay	Tetrahedra	Rhombic Dodecahedra
Expected	$2.2857 \cdot (3N_{fcc})^{2/3} \approx 3067$	$2.6728 \cdot (3N_{fcc})^{2/3} \approx 3586$	$4 \cdot (1N_{fcc})^{2/3} \approx 2580$
Ball	3207	3725	1944
Torus	3638	4089	2209

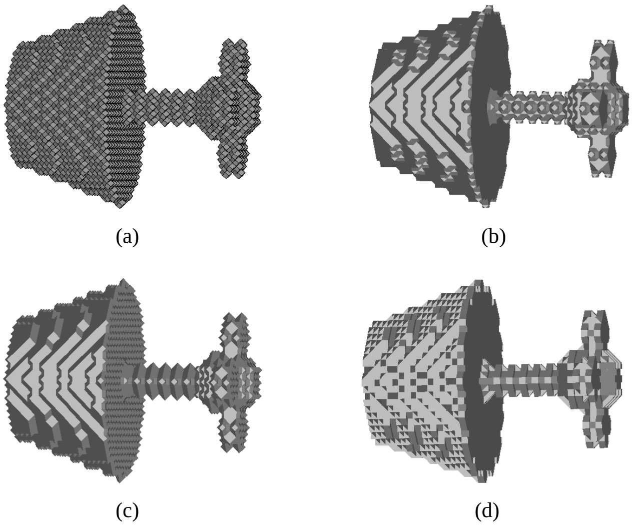


Figure 14. Triangulation of a (rotated) potplant. The voxel representation (4431 voxels) (a), triangulation using the Delaunay partitioning (12756 triangles) (b), tetrahedra partitioning (16164 triangles) (c), and rhombic dodecahedra partitioning (7372 triangles) (d).

r''-grid with $2r'' < r$ ([6]) are needed to preserve topology. With $2r'' = r = \sqrt{2}r'$ (i.e. using the limit of the allowed sparsity of the grids), a voxel in the cubic grid of unit volume has $r'' = \frac{\sqrt{3}}{2}$ and the volume of an fcc voxel with r' such that $2r'' = \sqrt{2}r'$ has volume $2 \cdot (2/3)^{(2/3)} \approx 3.67$. Thus, 3.67 times less sampling points can be used for the fcc grid compared to the cubic grid to guarantee topology preservation. Using the formulas (1) and (2) to compare with a cubic grid with $r'' = \sqrt{3}/2$, we get the expected number of triangles to be $2.85N^{\frac{2}{3}}$ for the marching cubes and $2.00N^{\frac{2}{3}}$, $2.33N^{\frac{2}{3}}$, and $1.68N^{\frac{2}{3}}$ for the Delaunay, tetrahedra, and rhombic dodecahedra partitioning, respectively.

From the values in Table 1, it is clear that the expected number of triangles for an image is a very rough estimation. It works as a tool here to analyze and compare the different marching-algorithms and shows that the algorithms presented here perform quite well compared to the original marching cubes-algorithm. The experiments verify that the marching-algorithm using rhombic dodecahedra partitioning gives the least number of triangles among the algorithms presented here.

We have presented a number of marching-algorithms designed for the fcc grid, all with the property of preserving the topology of the original object. This is the first paper describing topology-preserving marching-algorithms for the fcc grid. The major advantage with the fcc grid compared to the cubic grid in this context is its efficiency in topology preserving digitization. From this point of view, the marching-algorithms presented here outperform the traditional marching cubes algorithm on the cubic grid.

References

[1] H. Carr, T. Theussl, and T. Möller. Isosurfaces on optimal regular samples. In C. D. H. G.-P. Bonneau, S. Hahmann, editor, *Proceedings of the symposium on Data visualisation 2003*, pages 039–048. Eurographics Association, 2003.

[2] S. L. Chan and E. O. Purisima. A new tetrahedral tesselation scheme for isosurface generation. *Computers & Graphics*, 22(1):83–90, 1998.

[3] A. Gueziec and R. Hummel. Exploiting triangulated surface extraction using tetrahedral decomposition. *Visualization and Computer Graphics, IEEE Transactions on*, 1(4):328–342, 1995.

[4] L. Ibanez, C. Hamitouche, and C. Roux. Determination of discrete sampling grids with optimal topological and spectral properties. In *Proceedings of DGCI 1996, Lyon, France*, pages 181–192, 1996.

[5] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, July 1987. ACM Press.

[6] P. Stelldinger, L. J. Latecki, and M. Siqueira. Topological equivalence between a 3D object and the reconstruction of its digital image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):126–140, January 2007.

[7] P. Stelldinger and R. Strand. Topology preserving digitization with FCC and BCC grids. In *Proceedings of IWCIA 2006*, Lecture Notes in Computer Science, pages 226–240, 2006.

[8] T. Takahashi and T. Yonekura. Isosurface construction from a data set sampled on a face-centered-cubic lattice. In *Proceedings of ICCVG 2002*, volume 2, pages 754–763, 2002.

[9] G. M. Treece, R. W. Prager, and A. H. Gee. Regularised marching tetrahedra: improved iso-surface extraction. *Computers & Graphics*, 23(4):583–598, 1999.