Marching Cubes Algorithm for Transforming Images

Angélica Hernández Rayas, Delia Irazú Hernández Farías, Eduardo Pérez Careta, Rafael Guzmán Cabrera, José Francisco Gómez-Aguilar, José Zacarías Huamaní, and Teodoro Cordova Fraga

***Abstract*—A Marching Cubes algorithm modification is presented in this work. This proposal is used to make from surface image discretized on volumetric pixels, called voxels, a three-dimensional representation from a flat image is raised. To achieve this effect, each component is connected continuously (without gaps) regardless of the values of the image surface. The aim is to achieve this effect, optimizing computational resources. The ambiguous elimination cases as well as the identification of non-trivial cases are among the main features attended. Once identified, the modelling process is optimized, seeking congruence with the edges of the volume, applying techniques that allow connecting the edges adjacent to the corners, generating the triangulation of the cube, and introducing a new set of edge equivalence classes and cube faces. The obtained results allow to observe the effective performance of the implemented modification.**

***Index terms*—Image processing, marching cubes algorithm, surface reconstruction.**

1. Introduction

THREE-DIMENSIONAL surface reconstruction is an important topic in various application areas, such as quality inspection and reverse engineering. The goal of image-based 3D reconstruction is to infer the 3D geometry and structure of objects and scenes from one or multiple 2D images. Many image-based reconstruction methods have been proposed, based on photometric as well as geometric principles. For example, in [1] and [2] the authors used ray casting to find the surface and shading it with hue gradients or grayscales. A limitation of these ray casting algorithms is that they result in approximate shading with an unnormalized gradient. In [3], the authors present an approximation started on the contour surface and progressively connected triangles on the surface between adjacent slices. In [4], the authors proposed a network, which takes as input an RGB image and a target viewpoint. Other techniques estimate multiple depths maps from pre-defined or arbitrary viewpoints, for example, in [5] and [6] the authors followed the same approach but predict the depth maps, along with their binary masks, from pre-defined viewpoints. In [7], additional vertices are added to account for these sharp features. However, a subsurface with more than one component would cause ambiguities for connecting contours.

The standard surface reconstruction algorithm is Marching Cubes (henceforth MC). In the original MC algorithm, sharp edges create undesirable artifacts. The simplest technique is to draw the voxels directly on the screen with a simple 2D square. When the MC algorithm was first published, the fact that there were a few ambiguities in some of the vertex configurations was not completely known. The basic principle of the MC algorithm [8] is to subdivide space into a series of small squares, i.e., to sample the space on a regular grid. This algorithm examines the data in groups of eight, with each group forming a small cube. For each corner of the cube, the algorithm decides whether it is on the inside or the outside by using the provided region definitions, see Fig. 1.

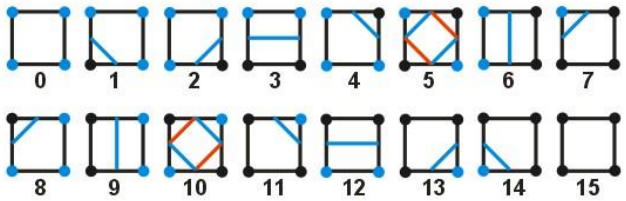


Fig. 1. The 16 square combinations of the MC algorithm.

When we process a single cube, from the sequence of cubes, we first classify each of the eight corner vertices of the cube identified as positive values if they are within the image surface and as negative values if they are outside the

Manuscript received on July 8, 2019, accepted for publication on September 17, 2019, published on December 30, 2019.

Angélica Hernández Rayas, Delia Irazú Hernández Farías, José Zacarías Huamaní, and Teodoro Cordova Fraga are with División de Ciencias e Ingenierías, Universidad de Guanajuato Campus León, Loma del Bosque 103, Lomas del Campestre, CP 37150, León, GTO, Mexico.

Eduardo Pérez Careta and Rafael Guzmán Cabrera are with División de Ingenierías, Universidad de Guanajuato Campus Irapuato-Salamanca, Carretera Salamanca-Valle de Santiago km 3.5 + 1.8, Comunidad de Palo Blanco, 36885, Salamanca, GTO, Mexico.

José Francisco Gómez-Aguilar is with CONACyT-Tecnológico Nacional de México / CENIDET, Interior Internado Palmira S/N, Col. Palmira, 62490, Cuernanvaca, MOR, Mexico.

Corresponding author: Rafael Guzmán Cabrera (e-mail: guzmanc@ugto.mx)

TABLE I
THE MC ALGORITHM RECOGNIZED
15 EQUIVALENCE CLASSES.

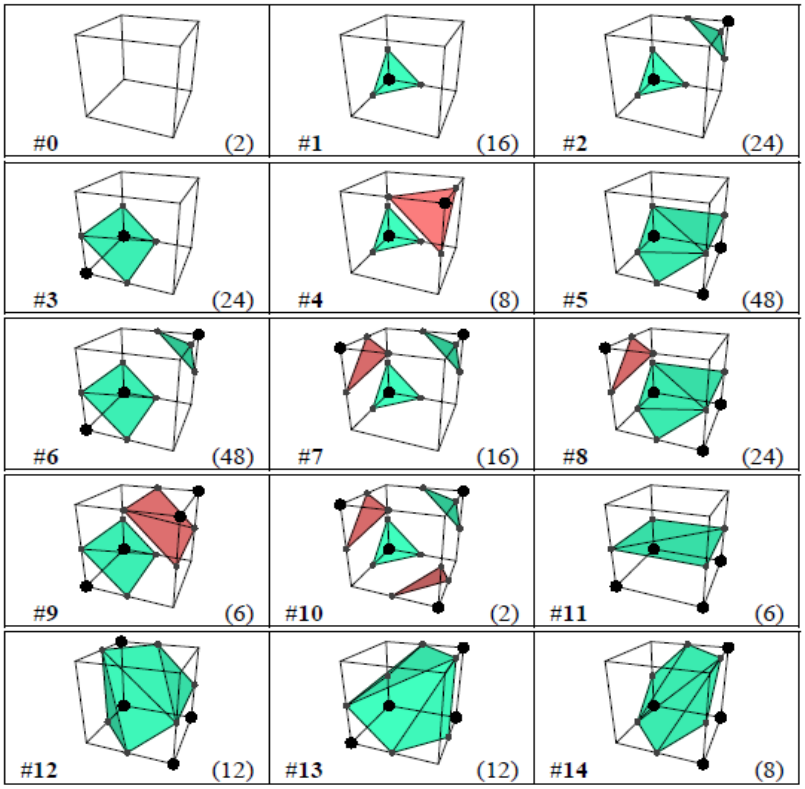


image surface. The vertices belonging to the internal triangulation of a cube when they are at the edge or one end can be computationally classified as: "outside" or empty space, and on the other hand "inside" of the image. The classifications vary between “inside / outside” successively of the eight corners of the cube. There are exactly 256 different possible combinations for each cube. The original MC algorithm represents these 256 cases, 28 in 15 equivalent cases, as shown in Table 1. Two cases can remain in the same class when we rotate a cube around an axis that passes through the center so that it can coincide with another case exactly.

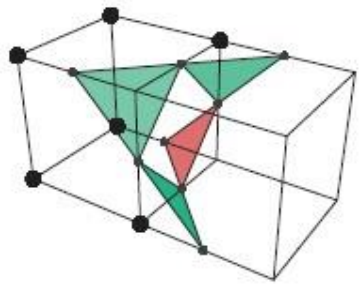


Fig. 2. Two cubes presenting an ambiguous face.

As shown in Table 1, the number in the lower-left corner of each entry is the class index, and the number in the lower right corner is the number of cases that belong to the equivalence class of the 256 total cases. A black dot in the corner indicates that it is within the volume of the solid and the corner without a black dot indicates that it is outside the volume of the solid. Green triangles indicate that they are the front concerning the point of view and red triangles indicate that they are hidden faces behind the cube material. If we choose to enforce a case described in Table 1, then problems are created in some situations with two cubes that share the same face, this can be ignored in some occasions, but if not, one or the other has to be inverted cube for proper image formation using the “inside/outside” option according to the cube situation. There is a result of a mismatch between two cubes that generates a kind of rectangle hole in the final triangulation of the two cubes, in Figure 2 it can be observed that the cube on the left must be inverted to match the respective equivalency of the image cell. The edges of the resulting triangulations do not fit along the shared face and a large rectangular hole is created. Green polygons are located on the outside of the cube and red polygons are located on the inside of the cube.

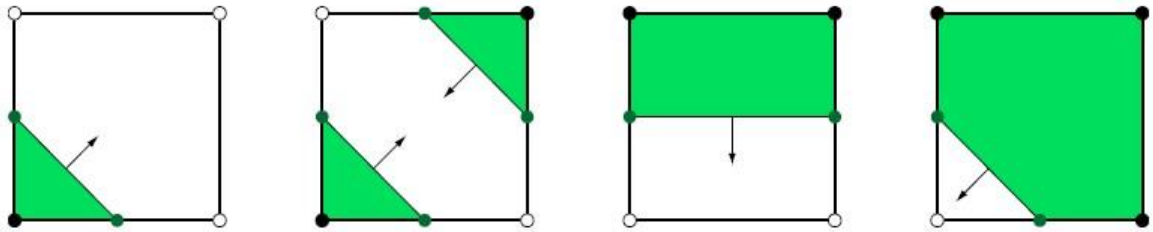


Fig. 3. The image volume edge congruence requirement

2. Proposed Changes to the Marching Cubes Algorithm.

When generating the image surface based on voxels, our most important interest is that the resulting surface is accurate, that is, each component is connected continuously, without gaps, regardless of the values of the image surface. Assuming that this requirement is achieved, our main objective among the remaining precautions is a maximum performance using the minimum computational resource to obtain results in short times. The main objective of the algorithm modifications' is to eliminate ambiguous cases without adding calculation per cube, to achieve this, the cases are considering the possible computational states in Table 1, it is divided into two groups: "inside / outside" out of the four corners of a face of a given cube, there are only four non-trivial cases after we eliminated the ones that are equivalent through the rotation.

The image volume edge congruence requirement can be satisfied when there is an ambiguous case as long as we connect vertices at adjacent edges that share a bottom corner, see Figure 3, they are the only non-trivial edge settings allowed by our algorithm (and a fifth configuration is a trivial face that has no edge because the four corner faces have the same state inside or outside).

Our modified MC algorithm does not allow the edge configuration, and therefore the triangulation of the left cube in Figure 2 is not considered to be a contaminant element. This circumstance requires the introduction of a new set of triangulation equivalency classes that require edge-by-face configuration, as shown in Figure 3, the only four non-trivial edge configurations that can be introduced in the face of a cube by the modification of the MC. This fact is due to a rotation of the cube. Vertices with solid points are classified inside and vertices with points are classified as outside. The green region represents solid space, and the arrows represent the normal surface direction outward along the edges.

Can be solved if the inversion of the equivalence relation is done for the classes that have an ambiguous face, leaving only the rotation. Next, it is set up three new equivalence classes to contain cases that have more than four interior corners in the image volume from the original equivalence class #2, #6, and #7 shown in Table 2. The result is the set of the 18 equivalence classes shown, where three new classes are shown in the last row. Classes #15, #16, and #17 are made up of the inverse representation of cases #2, #6, and #7, and their respective rotations.

The number in the lower-left corner of each entry is the class index, and the number in the lower right corner is the number of cases that belong to the equivalence class of the 256 total cases. A black dot in the corner indicates that it is within the volume of the image, and the corner without a dot indicates that it is outside. Green triangles are in front of the cube face and red triangles are behind the cube material ions.

If chooses to enforce a case described in Table 1, then problems are created in some situations with two cubes that share the same face, this can be ignored in some occasions, but if not, one or the other cube has to be inverted to the adequate formation of the image using the "inside/outside" option according to the situation of the cube. There is a result of a mismatch between two cubes that generates a kind of rectangle hole in the final triangulation of the two cubes as shown in Figure 4.

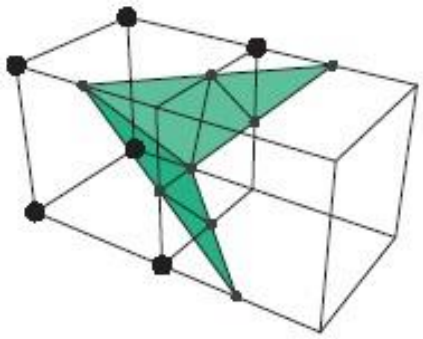


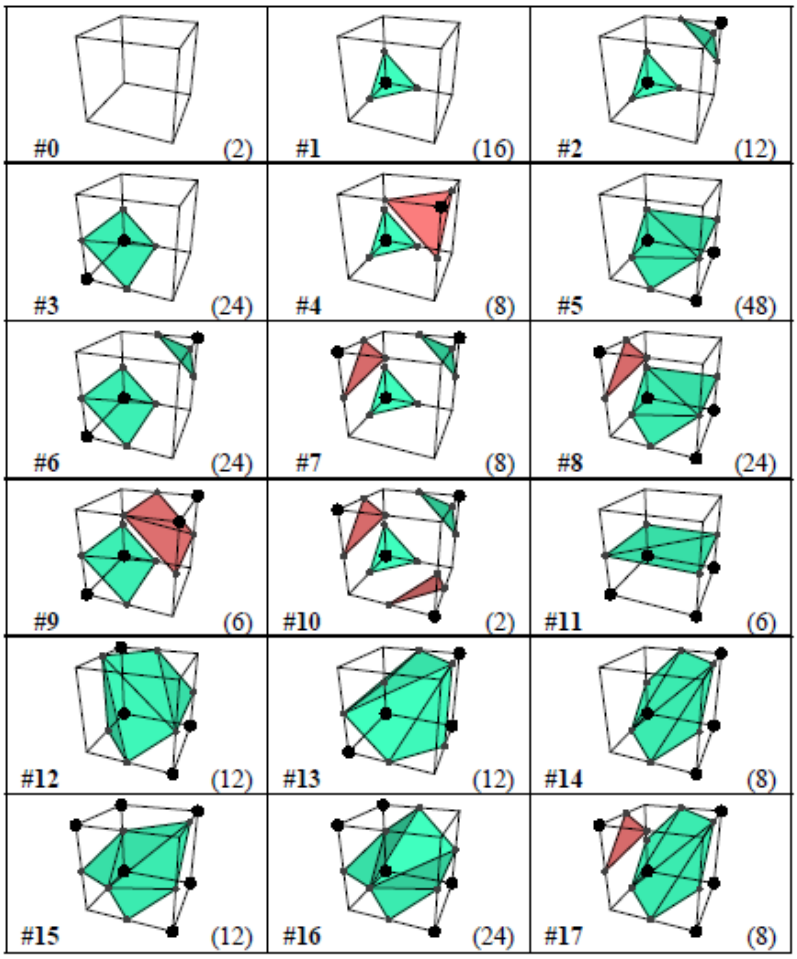
Fig. 4. Two cubes sharing an ambiguous face.
The left side belongs to one of the new equivalence classes

A. Marching Cube Algorithm

Modifications to the Marching Cube algorithm were designed considering the following points:

- Possibility to edit small surface regions without rebuilding the whole image surface.

TABLE II
MC ALGORITHM WITH 18 EQUIVALENCE CLASSES IDENTIFIED.



- The previous presentation of the surface to show the details of the image.
- The resulting mesh is defined with no imperfections between triangles or regions without the rise that arose the surface.
- Vertices it is shared as efficiency between each neighboring cube and take full advantage of the hardware to transform artifacts and decrease the storage requirements of the resulting surface.
- Minimize the use of memory and computational resources by generating meshing on the image surface.
- Dividing the total image into relatively small voxels of equal size helps to achieve these goals.

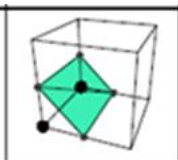
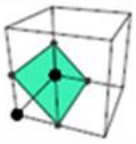
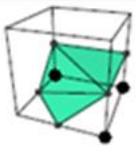
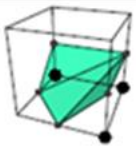
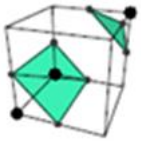
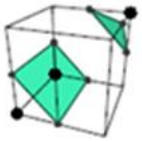
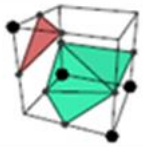
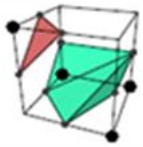
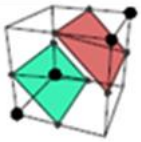
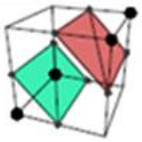
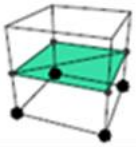
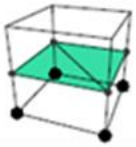
So that the level of detail of analysis of the total volume of the solid is high, blocks of 16×16×16 voxels are divided for their respective analysis. Each block has a coordinate axis regardless of the division it is, and these are x, y, and z from one to any block joint set.

3. RESULTS

Table 3(a) shows the 12 equivalence classes for which multiple triangulations are possible, the two triangulations shown are the only ones that test us for the better division at high-resolution voxel data, they are chosen to be as different as possible in terms of their main directions of curvature, the second column, represent the total triangulations are possible for each equivalence class.

Since the lowest levels of detail are normally processed only when the pixel intensity is low, the triangulation we assign to any cube only makes a subtle difference that can be observed directly on the surface mesh. However, there are two indirect ways in which the ability to choose between triangulations makes an important difference. The first way is to increase the level of detail to a few sets of cubes on the surface, visualizing a noticeable "jump" that occurs in the mesh when the level of detail is abruptly changed, but the visible magnitude of this transition is

TABLE III
THE 12 EQUIVALENCE CLASSES FOR WHICH
MULTIPLETRIANGULATION: POSSIBLE -1.

#3			$C_2 = 2$
#5			$C_3 = 5$
#6			$C_2 = 2$
#8			$C_3 = 5$
#9			$C_2^2 = 4$
#11			$C_2 = 2$

reduced when the low resolution of the triangulation is a better division for the high-resolution surface .

Table 4 shows appearance of shading is the possibility -2. A very noticeable artifice can appear when a poor choice of triangulation is made because it can create a concavity that produces unwanted shadow when aligned in a way unfavorable to the surface view. Choosing the best triangulation from Tables 3 and 4 helps eliminate these defects by creating a smoother low-resolution surface mesh.

We refer to the meaning of transition to be one of the forms shown in Figure 5, which is influenced by nine empty voxels on one side. The face opposite the full resolution face is called the half resolution face, and the sample values at its four corners are identical to that of the corresponding corners of the full resolution face. The width of a transition cube (that is, the distance between full resolution and average face resolution) is a freely adjustable parameter on the image surface.

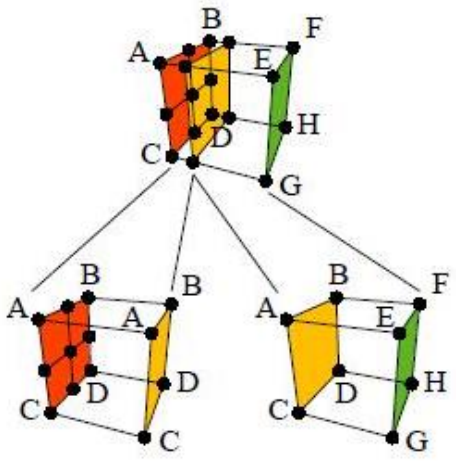
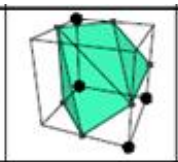
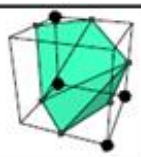
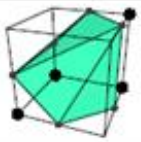
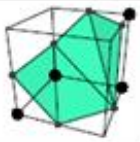
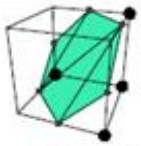
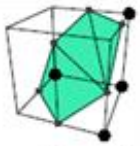
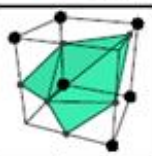
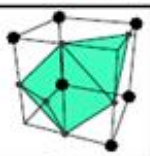
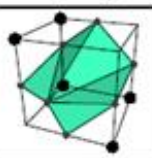


Fig. 5. A transition cube is divided into two parts along a plane parallel to the boundary face between maximum resolution block and medium resolution

TABLE IV
THE 12 EQUIVALENCE CLASSES FOR WHICH
MULTIPLE TRIANGULATIONS: POSSIBLE -2

#12			$C_4 = 14$
#13			$C_4 = 14$
#14			$C_4 = 14$
#15			$C_4 = 14$
#16			$C_5 = 42$
#17			$C_4 = 14$

Our new algorithm triangulates the left side using sample values that only appear on the full resolution side. The right part is triangulated with the conventional modified MC algorithm, see Figure 5.

4. CONCLUSION

The transition cubes always have the same configuration regarding the number and location of our voxels, and this is true even when a medium resolution block is bordered by full resolution of two- or three-part blocks. An equivalence class is a trivial class that contains the two cases in which the interior or exterior state of the nine sample values is the same.

The modifications proposed to the MC algorithm in this work allow us to overcome some of the deficiencies that the original algorithm has, allowing us to make the recovery of surfaces without gaps, which allows us to obtain a 3D representation of a flat image in a more efficient.

ACKNOWLEDGMENTS

Authors wishing to acknowledge at the University of Guanajuato for the partial support in this project under grant DAIP2019/59023. Also José Francisco Gómez Aguilar acknowledges the support provided by CONACyT: Cátedras CONACyT para jóvenes investigadores 2014.

REFERENCES

[1] E. J. Farrell. Color display and interactive interpretation of three-dimensional data. IBM J. Res. Develop 27, pp 13-19, August 1983.

[2] K. H. Hohne and R. Bernstein. Shading 3D images from CT using gray-level gradients. IEEE Transactions On Medical Imaging MI-5, pp. 45-47, March 1986.

[3] E. Keppel. Approximating complex surfaces by triangulation of contour lines. IBM J. Res. Develop 19, pp. 2-11, January 1975.

[4] M. Tatarchenko, A. Dosovitskiy, and T. Brox, “Multi-view 3D models from single images with a convolutional network,” in ECCV, 2016, pp. 322–337.

[5] A. A. Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum, “Synthesizing 3D shapes via modeling multi-view depth maps and silhouettes with deep generative networks,” in IEEE CVPR, 2017, pp. 1511–1519.

[6] C.-H. Lin, C. Kong, and S. Lucey, “Learning Efficient Point Cloud Generation for Dense 3D Object Reconstruction,” AAAI, 2018.

[7] L. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel. Feature sensitive surface extraction from volume data. In Proceedings of SIGGRARPH 2001, pp. 57-66, 2001

[8] W. E. Lorensen and H. E. Cline, *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*, Computer Graphics 21(4): 163-169, 1987.