Tridimensional Reconstruction of Neurons using clear light microscopic images

G. Torrealba, M. Martin, A. Balandra

Computer Science Faculty – Autonomous University of Puebla

G. Flores

Physiology Institute – Autonomous University of Puebla

[1]uksa@first-third.edu, [2]mmartin@cs.buap.mx
[3]ponchobalandra@gmail.com, [4]gflores@siu.buap.mx

Abstract— **To visualize the tridimensional surface of a neuron and its dendrites, instead of a bi-dimensional representation of it, allows locating the morphological relations that make easier to characterize their structures. Due the size of the neurons these only can be observed with the help of a microscope, however through microscope only a bi-dimensional representation of the neuron can be seen, and this is why is necessary to take a series of images of the neuron on different depths for characterize it's 3D representation. To be able to visualize images taken of a neuron like these were just one image, help the different departments that study the neurons' morphological characteristics. Here are presented the algorithms used to build a 3D reconstruction of the neuron through a series of images in 2D taken with a clear light microscope.**

Keywords— **3D Reconstruction Algorithms, microscopic images, neurons, Golgi-Cox Technique, Morphological Neuron Studies.**

I. Introduction

The images taken though microscopy are produced by the reflection of light from the surfaces, or also from it transitions though translucent objects. The standard microscope of light has a reasonable high zoom and also a good field of vision, and this creates many problems to analyze surfaces. If the surface to analyze is not flat enough and perpendicular to the optic axis, this cannot be properly focused on the plane. The clear light microscopy has a good field of vision, and that difficult to extraction the surface length that is completely perpendicular to the focal plane, because of that the produced image will show parts of the object that are not in the focal plane.

In some projects realized in the BUAP's Physiology Institute, is necessary to carry out a count and characterize the neurons' dendritic structures contained in the cerebral tissue taken from rodents. –These tissue samples are treated using the Golgi-Cox Method [3]-. These procedures are executed using a clear light microscope equipped with a normal digital camera to capture stack of 2D images, which represent several depths in the 3D space occupied by the neuron.

With the finality to facilitate the count and characterization of the structures contained in the image stack an application had been developed to obtain the tridimensional reconstruction of the neuron.

These algorithms of surface reconstruction that start from a set of images have two fundamental steps [4,5]:

1. The first one obtains the closed boundary that envelope the object in every plane of the stack; this process is known as segmentation.

2. The second step is responsible of generate an approximated surface that join the continuous boundary sections.

This article presents the algorithms of segmentation and surface generation that can be implemented to obtain tridimensional reconstruction a neuron with clear light microscopy images.

II. Development

A. Outline

Due the nature of the problem, it had been divided in following modules:

1. Pre-processing Module.
2. Reconstruction Module.
3. Localization and Characterization Module.

The pre-processing module executes operations over the images contained in the package to obtain uniformity in the data and emphasize the structures of interest contained in the images, with the purpose to facilitate the reconstruction process.

The reconstruction module is responsible to build the tridimensional representation of the neuron. The final result of this process is a file with all data of the 3D model of the neuron; with this file the model can be loaded directly without running the reconstruction process again. This module cannot be executed if the image package hasn't been pre-processed with the pre-processing module.

In the last module is responsible to locate and characterize the dendritic structures of the neuron, using the obtained tridimensional module and the user interaction. This module

must generate a report of the obtained data from the previous modules. Fig 1.

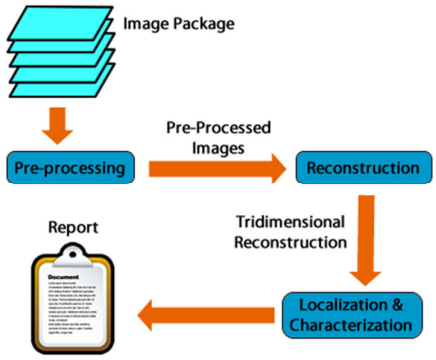


Fig 1 System's Module Diagram

B. *Pre-processing*

In the pre-processing module is where all the necessary previous processes to obtain the reconstruction are executed. The obtained images on this process will be the input data of the reconstruction model. Fig 2.

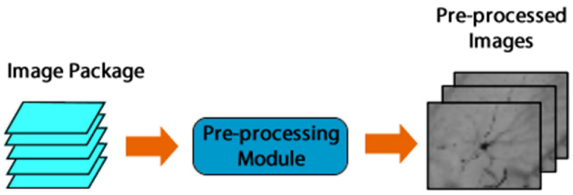


Fig 2 The input and output data of the pre-processing module

Every single image in the package will be represented using a tridimensional byte matrix that has all the levels of red, green and blue corresponding to every pixel channel of the image.

Once all the pixel data of every image in the package have been stored on the tridimensional matrix, the image processing algorithms are executed on the tridimensional matrix to emphasize the dendritic structures on every plane in the matrix.

Following, the applied process to every single image are listed [6,7,8]:

1. Gray Scale Conversion – A filter applied to work only with just one tone per pixel. The tone assigned to every single pixel is average of the three channels of the pixel. This can be also seen like this:

2. Zoom by Linear Interpolation – The data interpolation consists on estimate new intermediate values of a function, so a square is defined between the points *(i,j) y (i+1, j+1)* –of the original image-, that comes from another point *(p,q)*, which in turn goes from an exact point mapping *(i', j')* –of the amplified image-; then the contribution by the proximity of the known value of the function points *(i , j), (i+1 , j), (i , j+1), (i+1 , j+1)* must be calculated. A 300% zoom is applied to the images to slim down the dendritic structures, and these could be detected by the surface reconstruction algorithm.

Once the image have been transformed into gray scale and the magnified by 300%; the preprocessing stage has finished and the image must be stored on the disk. Now that the preprocessed images are safely stored, these will become the input data of the reconstruction module.

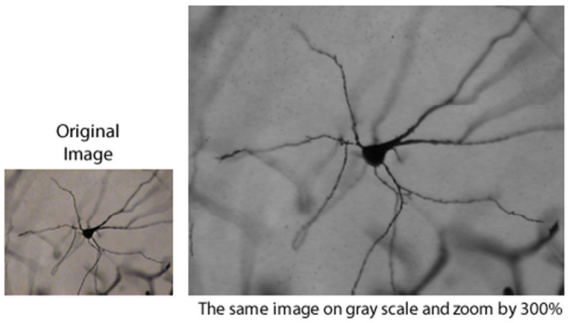


Fig 3 Original image and the same image after have been pre-processed.

C. *Reconstruction*

The objective of the reconstruction module is to generate the tridimensional view of the dendritic structures contained on an image package [7,8,9], and also to create a file that contains all the associated data to the tridimensional representation; to be able to visualize it continuously without processing the package again. The input data received in this module are the images obtained in the segmentation process and the output data is the tridimensional reconstruction and the file that represents it. Fig 4.

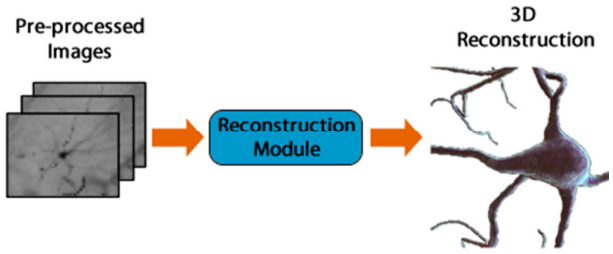


Fig 4 The input and output data of the reconstruction module

To carry out the tridimensional reconstruction the marching cubes algorithm have been implemented [9,10,11]. This algorithm is used to reconstruct a surface with volume contained on flat planes. To implement this algorithm is

necessary to know the value of the gray level that represents the boundary of the surface. Furthermore the pixels' gray tones are not the same for every image package, this is caused by different factors like: the light intensity of the microscope, the ambient lighting, the sample stain time, and other factors, so is necessary to obtain the boundary gray value from different image packages independently from these factors.

To determine the way the boundary value of the surface an analysis of the gray tones variation on every pixel in the images must be performed. From all the images in the package just one image plane is selected, and in that plane all the focused regions to reconstruct must be found. To solve this problem a simple graphic was built. This graphic contains the pixel gray tone variations located on a straight line that cut the focus surface from one side to the other; with the same focus region but now on a different focal plane the analysis is executed again but now this time on an unfocused region Fig 5. As the result of several analyses was observed that; the value that defines as the border must be less than any value on the unfocused region and higher than the any value from the focused region.

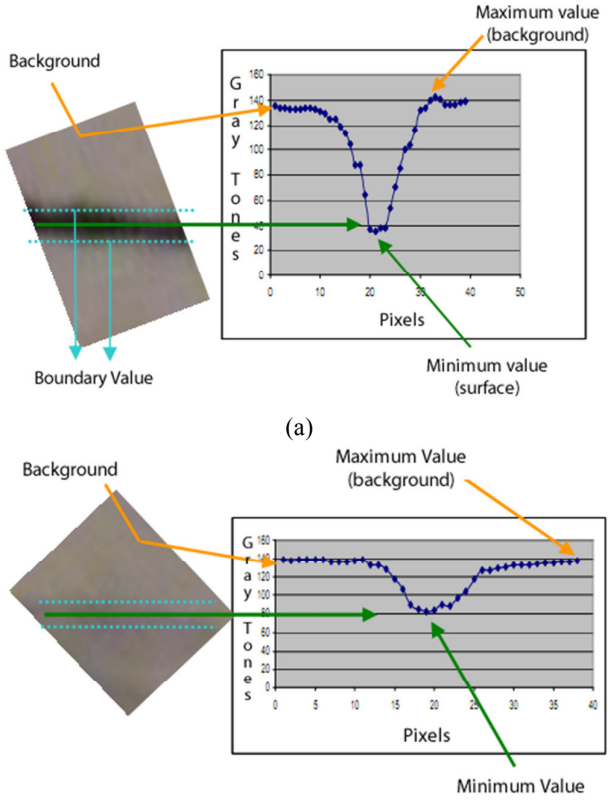


Fig 5 Gray tone variation analysis to obtain the boundary value. (a) Analysis on a focused area, (b) Analysis on an unfocused area.

So, to obtain the border value from an image package three zones focus zones and three unfocused zones are selected, and then the smaller value from the unfocused zones is selected and the smaller value from the focus regions. At last the final border value is purposed in the middle of the previous minimum values, and this also gives a good range of freedom.

Marching Cubes Algorithm

This algorithm generates a triangular wire structure from the border of the surface, obtaining these though the previous obtained boundary value. To obtain the point that will form the triangular wire structure is used an imaginary cube that is formed by eight vertexes with their own scalar values. This cube is goes around through the different planar sections that contain the surface to build. With the boundary value can be determined if the cube is inside the surface, outside of it, or if the surface cuts one or various sides of the cube; every possibility is determined by the amount and configuration of vertexes inside and outside the surface. If a vertex is inside the surface, and its adjacent vertex outside of it, is known that surface cuts the edge between those vertexes. The position of the intersection point over the edge is determined using linear interpolation, the boundary value, and the gray level of each vertex [9,8,10].

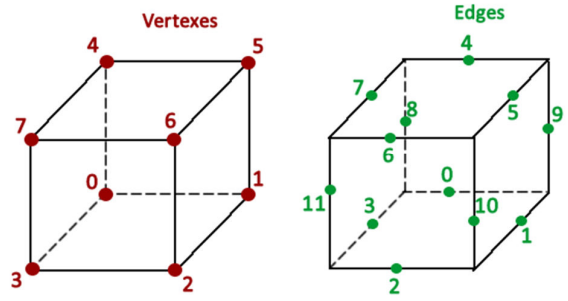


Fig 6 Representation of the tridimensional cell with the numerated vertexes and edges.

The first part of this algorithm calculates an 8 bit index, one per each vertex, which indicates the status of each vertex in the tridimensional cell. From that index and using an edge configuration table, which maps the all the vertexes configurations below the surface level, the intersected surface edges are determined. Now that the intersected edges were defined is necessary to obtain the intersection points of the edges though a lineal interpolation and the gray tone level of the edge's vertexes. The calculus of the previous mentioned index is conducted like this:

1. Is initialized the index variable on cero, index =0.
2. If the gray level on the vertex N=0 is below the boundary value, the first bit on the index variable turns on, index = index || 1.
3. If the gray level on the vertex N=2 is below the boundary value, the second bit on the index variable turns on, index = index || 2.
4. If the gray level on the vertex N=3 is below the boundary value, the third bit on the index variable turns on, index = index || 4.

N. If the gray level on the vertex N is below the boundary value, the umpteenth bit on the index variable turns on, index = index || (2 ^ (n-1)).

5. The process stops when N=9. Then if the gray level on the vertex N=9 is below the boundary value, the ninth bit on the index variable turns on, index=index || 128

If the previous index is used on the edge configuration table it returns a 12 bit number, in where every single bit represents an edge, if the bit is 0 the surface doesn't goes through that edge and if the bit is 1 the surface goes through the edge. If the surface doesn't go thought any cube's edges then the returned value is 0, and if all the cube's vertexes are inside de surface then the returned value will be 255.

The point where the surface intersects on edge should be calculated using lineal interpolation as:

Where:

, is the point to calculate.
and , are the edge's vertexes that have been intersected by the surface.
and are the restive gray levels of the points and .

Once all the intersection points were obtained, that means, all the boundaries of the surface were founded; is necessary to form the triangles that represent the surface that goes though that cell, using the different possibilities in where the surface can intersect the cube's edges. To obtain these triangles the process uses another table that harness the previous vertex index to get the sequence, in which all the vertexes must be connected to build the surface triangles on that cell.

So the tridimensional representation of the neuron will be constituted by a triangular mesh, which will be formed by a set of triangles obtained during the marching cubes process.

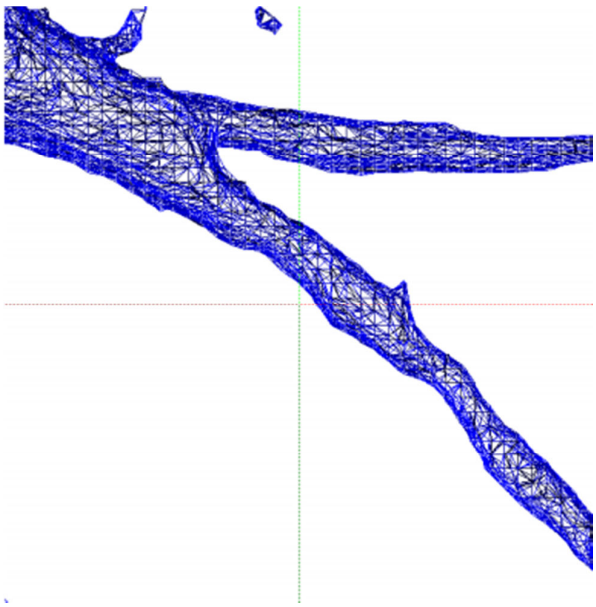


Fig 7 Representation of a 3D object with a triangular mesh.

D. Localization and Characterization

The location and characterization process of the dendritic structures of the neuron is carry out in a semiautomatic way, by means the tridimensional obtained model and with the user's assistance, the measurement of the 3D structures should be executed as: Every single structure to measure must be represented on the 3D scene with a path, which will form by a set of markers –spheres- on the 3D scene. The measures between those markers will be saved on a text file.

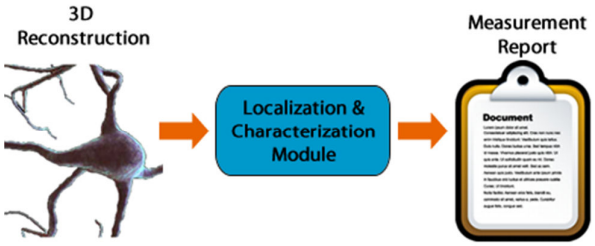


Fig 8 Representation of the input and output data in the localization and characterization module.

Once the reconstruction has been finished, to measure a complete dendritic structure, is necessary to create a path over the structure, putting several markers that will form the path to be measure.

Using the virtual length between the markers and the x,y,z resolution of the image package; the real length in microns of the dendritic structures are calculated using the linear relationship between the micron image package resolution and the distance between the sphere markers on the 3D scene, then a simple cross multiplication solves the problem.

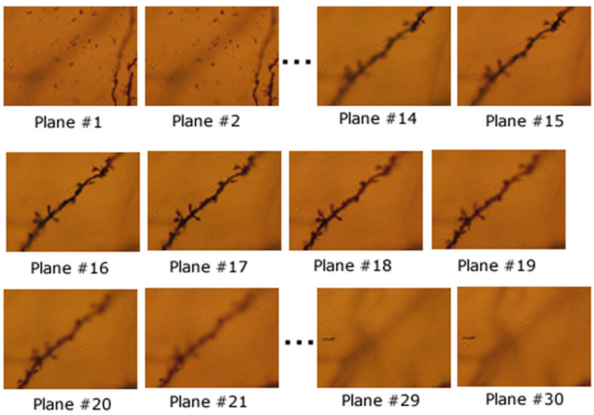
E. Testing and Results

The developed application that implements the presented algorithms was named Muni. It was developed in C# on Visual Studio 2005 using the Tao-Framework to get OpenGL support. This application opens the image packages with the native image libraries inside the .NET Framework, this gives the application the possibility to work with bmp, jpg, gif, png, and tiff formats.

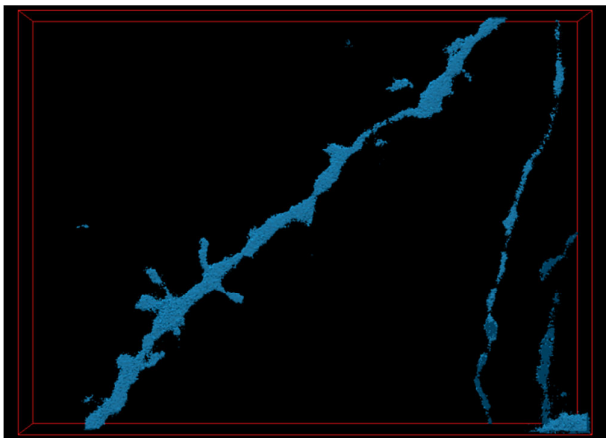
The application testing was carried out in a double processor Xeon server with 2.33 Ghz each and 4 Gb of RAM.

Following we present the testing results obtained with 3 image packages, on TIF format, with a 762 x 574 pixels size.

The Fig 9 shows the reconstruction obtained with an image package with 30 planes that show dendritic spines. The reconstruction process generated a triangular mesh with 171,714 triangles, in a total processing time of 42 sec.



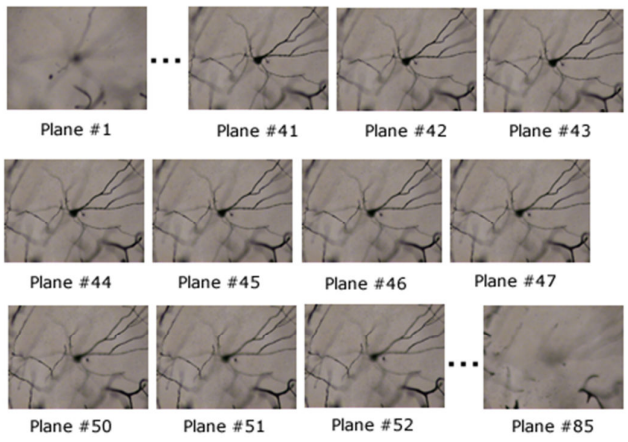
(a)



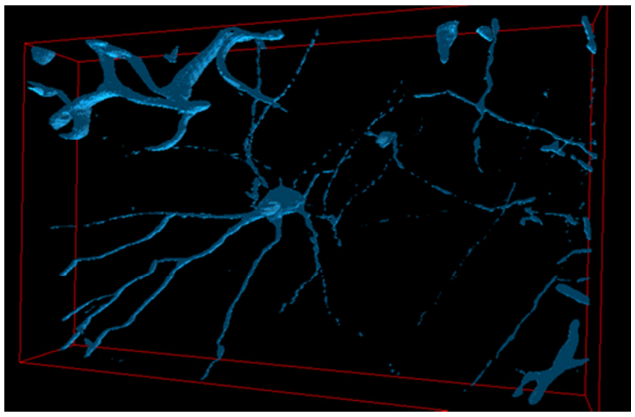
(b)

Fig 9 (a) Original dendritic spines image taken from an image package, (b) 3D reconstruction obtained from that image package.

The Fig 10 shows the reconstruction obtained from a complete photographed neuron with an image package with 85 planes. The reconstruction process generated a triangular mesh with 349,974 triangles, in a total processing time of 51 sec.



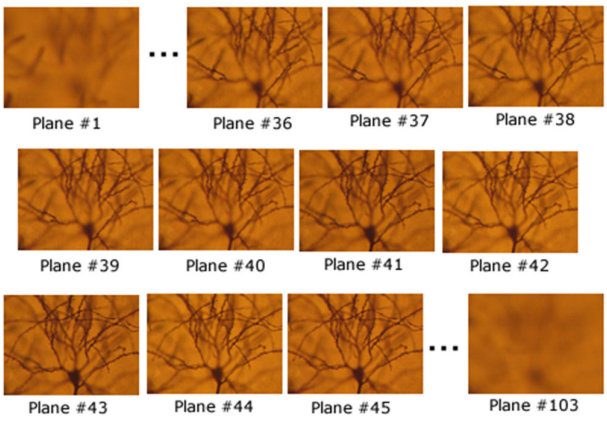
(a)



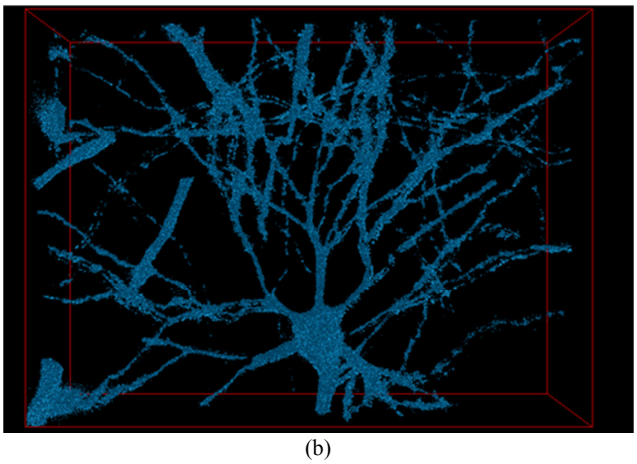
(b)

Fig 10 (a) Original dendritic spines image taken from an image package, (b) 3D reconstruction obtained from that image package.

The Fig 11 shows the reconstruction obtained from a complete photographed neuron with an image package with 103 planes. With a triangular mesh set formed by 2,374,895 triangles, in a total processing time of 198 sec.



(a)



(b)

Fig 11 (a) Original dendritic spines image taken from an image package, (b) 3D reconstruction obtained from that image package.

III. CONCLUSIONS

The technique used to get the tridimensional representation opens the possibility to carry out the location and characterization of the dendritic structures in a semi-automatic way. Furthermore there's also many work to do, to calibrate the x,y and z micron resolution of any image package taken from any type of microscope.

The method can also be applied to other structures from the same kind biomedical environment, like: bronchial tree inside the lungs, the tree structures inside the kidney, microscopic blood and lymphatic vessels, and many others.

The complementary phase of this work is the development of applications based on the presented algorithm. We are particularly interested on characterize neuronal structures like dendrites, spines and soma.

We thank CONACYT and VIEP-BUAP for all the given support to the accomplishment this project.

ACKNOWLEDGMENT

The presented application, models, and images packages are available in a Google Code repository http://code.google.com/p/muni-neuron/, and all this material have been released under the GNU/GPL licence.

REFERENCES

[1] Q.Wu, F.A. Merchant K.R. Castleman, "Microscope Image Processing", AP.
[2] D.Murphy, "Fundamentals of Light Microscopy and Electronic Imaging".
[3] R. Gibb, B. Kolb, "A method for vibratome sectioning of Golgi–Cox stained whole rat brain", J. Neurosci. Methods 79 1998.
[4] J. D. Pina y R. Alquézar "Nuevo método para la reconstrucción de superficies a partir de imágenes tomográficas".
[5] Nonato, A.Cuadros, R. Minghim, M.Oliveira,"Beta-connection: generating a family of models from planar cross sections".
[6] T. Acharya, A.Ray, "Image Processing, Principles and Applications", Wiley, 2005.
[7] I.Bankman, "Handbook of medical imaging", Academic Press.
[8] J.C.Russ, "The image Processing Handbook".
[9] R. Baldock, J. Graham, "Image processing and analysis, a particle approach",PAS.
[10] Chien-Chang Ho, Fu-Che Wu, Bing-Yu Chen, Yung-Yu Chuang, Ming Ouhyoung, "Cubical marching squares adaptative feature preserving surface extraction form volume data".
[11] Chien-Chang Ho, Chen-Han Tu, Ming Ouhyoung, "Detail Sculpting using cubical marching squares".
[12] S. Schafer, J. Warren, "Dual marching cubes: primal contouring of dual grids".