Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 745–749
ISBN 978-83-60810-51-4

Graph-Based Volumetric Data Segmentation on a Hexagonal-Prismatic Lattice

Mihai Popescu, Razvan Tudor Tanasie
Department of Software Engineering
Faculty of Automatics, Computers and Electronics
University of Craiova, Romania
Email: {mpopescu, razvan_tanasie}@software.ucv.ro

***Abstract*—In this paper we present a graph-based volumetric data segmentation method based on a 3D hexagonal prismatic lattice. We evaluate the advantages and disadvantages of using this lattice in contrast with classic ones. One of the main advantages are high isoperimetric quotient, near equidistant neighbours (ability to represent curves better, resulting in a better segmentation) and high connectivity. Disadvantages are due to the main stream lack of interest in this area and thus data sets must be converted back and forth from rectangular to hexagonal latices both in acquisition and visualization processes.**

Keywords- volumetric, segmentation, hexagonal, lattice.

I. Introduction

Recent advances in the study of the human visual system have shown that the implicit arrangement of the sensor array is hexagonal in nature. This implies that all the other systems, wherever they may lie in the cortical hierarchy depend on this arrangement. This fascinated people including researchers and scientists, for a long time. The classic example is that Pappus of Alexandria stated the Honeycomb conjecture that says that the best way to partition a plane into regions of equal area is a regular hexagon. This conjecture have recently been proven by Hales [1].

In this paper we examine the possibility and the effects of changing the classical sampling lattice from cubic to hexagonal. Switching to the new lattice will influence data aquisition, indexing, storage and processing algorithms. Based on it's geometry the lattice has advantages and disadvantages that are also discussed.

We propose a pseudo-hexagonal prismatic lattice and for it a volumetric segmentation algorithm is approached theoretically. Experimental results cannot be provided at this stage.

II. Volumetric Data Sampling Lattice

Volumetric data can be obtained mainly from two sources:

1) Acquisition of discrete representations of real objects with medical imaging technologies like:
   a) Computed Tomography (CT)
   b) Magnetic Resonance Imaging (MRI)
   c) 3D Ultrasound
2) Computer generated 3D textures:
   a) Polygonal object voxelization
   b) Interactive voxel editors
   c) Procedural objects defined by 3D functions

A. Classic lattice systems

Usually the volumetric data is defined by a scalar 3D function discretized into a set of cells or points using a regular lattice. Formally, a lattice is a discrete subgroup of $\mathbb{R}^n$ that can be generated from a vector basis by linear combination with integer coefficients: $L = \{\sum_{i=1}^{n} a_i v_i | a_i \in \mathbb{Z}\}$, where v_i is a generation vector from the vector basis.

The most frequently used lattice is the Cubic Cartesian lattice with generation vectors $X = (1, 0, 0)$, $Y = (0, 1, 0)$ and $Z = (0, 0, 1)$. The lattice depends of course on the modality of the volume data. While CT and MRI scans use the CC lattice, the volumetric data for 3D ultrasound uses a non-Cartesian grid called *acoustic grid*, similar to a truncated pyramid or a cone.

Other lattice systems such as Body-Centred Cubic (BCC), Face-Centred Cubic (FCC) and Hexagonal Close Packing (HCP) also can be used. Kepler conjecture states that the highest average density of sphere packing is $\approx 74\%$. Hales shows that only FCC and HCP are the only lattices that offer this property [2].

Most research is focused on rectangular lattices because this is the most common lattice system used by the acquisition devices (2D, 3D or 4D) and so, hexagonal lattices are left in the shadow even if they present interesting properties. One of the major advantages of the hexagonal lattice is higher connectivity and the neighbouring points are equidistant, while on a rectangular lattice the distance between neighbours depends on whether the neighbour is directly adjacent or diagonal. The fact that hexagons are rounder than squares make image processing algorithms, such as edge detection or image segmentation, provide better results when they are applied on a hexagonal lattice.

On the other hand, it's impossible to build a regular 2D hexagonal lattice right on top of a rectangular lattice. The new sampling points will not match the original points and thus the reconstructed lattice will have to suffer in resolution.

B. The proposed lattice

To overcome these problems we introduce a new pseudo-hexagonal lattice which is created by simply shifting the even scan-line with half a unit. This is the 3D extension of the ”Brick Wall” lattice first proposed by Fitz and Green [3] for hexagonal image latices. It’s possible to achieve this shift by sampling at midpoints in every even scan-line or it can be implemented in the input device hardware itself. We denote this lattice as the *Solid Brick Wall* lattice or SBW.

In 2D the compactness factor is determined by the isoperimetric quotient $Q = \frac{4\pi A}{L^2}$, where A is the area and L is the length or object perimeter. The compactness factor of a brick wall element is $Q = \frac{3\pi}{(1+\sqrt{5})^2} \approx 89,99\%$ higher than the one if the lattice was rectangular $Q \approx 78,54\%$ or if the lattice was pseudo-hexagonal build using the rectangular lattice sample points $Q \approx 85,74\%$ disregarding two inner pixels as in [4].

An equivalent quotient can be expressed for 3D as well: $Q = \frac{36\pi V^2}{S^3}$, where V is the volume and S is the surface of the polyhedron. The 3D SBW lattice cell with the generation vectors $X = (1,0,0)$, $Y = (\frac{1}{2},1,0)$ and $Z = (0,0,1)$ has the compactness factor of $Q \approx 52,47\%$. Nonetheless, the highest compactness factor known can be realized with the Weaire-Phelan structure ($Q \approx 76,5\%$) [5] but our structure is simpler and has a slightly better quotient than that of a cubic lattice cell with $\Delta Q \approx 0,11\%$. We can increase the compactness factor even further up to $Q \approx 59,99\%$ if we were to allow the Z axis generation vector to be $Z = (0,0,\frac{6}{1+\sqrt{(5)}})$ but as it is better to have rational sampling points locations rather than real values we will remain at our first variant.

It can be easily be shown that the 3D Delaunay triangulation can be generated if we decompose an SBW cell into six triangular prisms and each prism into three tetrahedrons. Each tetrahedron point will be circumscribed by the same sphere if it belongs to the same prism and the sphere will contain no other point within.

III. Practicality of using SBW Lattice

A new lattice system that rejects the common square lattice must face first some serious considerations on which one must ponder upon.

1) *Data acquisition* Hardware capable of sampling volumetric data from the real world directly onto the a hexagonal lattice are not generally available for use. Therefore, data resampling on the standard cubic-latticed volumetric is required before any processing can be performed.
2) *Indexing and Storage* Before any processing, the volumetric data must be indexed to be able to address individual voxels from the SBW lattice. The voxel form is a pseudo-hexagonal prism (rather than cubes) being the Voronoi cell of the proposed lattice. Storing

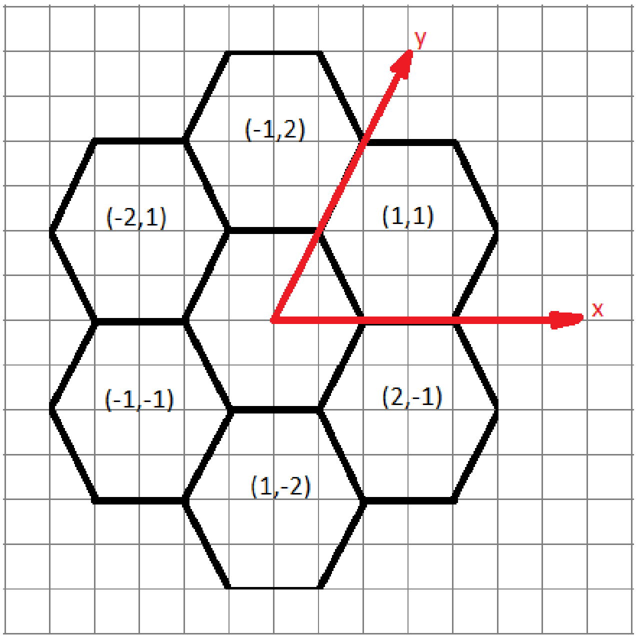

Figure 1. 2D brick wall lattice

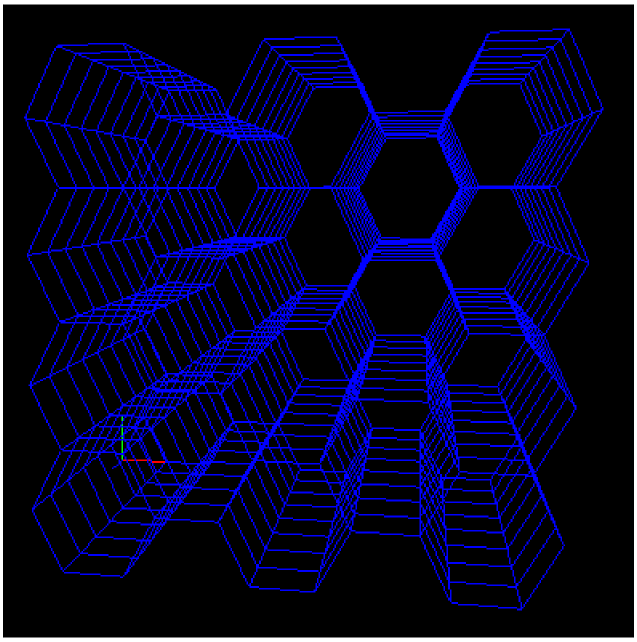

Figure 2. 3D solid brick wall lattice

should be made also in hexagonal form to avoid conversion every time data is accessed. The indexing system should allow simple arithmetic functions to be able to have random access on volumetric data.

3) *Data Processing* Volumetric data processing algorithms must be redesigned or adapted to be able to exploit the strengths of the new pseudo-hexagonal lattice system, particularly the high node connectivity, high compactness quotient and near equidistant voxel nodes.
4) *Visualization* Nowadays, volumetric data visualization methods are specialized for cubic lattices. Therefore hexagonal 3D images must suffer another conversion prior to the visualization so that it can benefit from the display hardware features such as fast 3D texture sampling. This conversion process depends on the indexing method used: reversing the original conversion process or possibly a convolution.

A. Data acquisition

As stated earlier, hardware based acquisition can be unavailable but existing hardware systems can easily be modified by shifting alternate scan-lines. Using the proposed lattice, the modification can be cost-effective. Staunton [6] designed an architecture capable of sampling hexagonal images on the brick wall lattice, injecting a delay to alternate lines; his work being extensible to 3D scanners as well. Interestingly, hexagonal sensors are already being implemented in medical imaging hardware [7] but the applications are limited.

Software based resampling can be a more straightforward alternative. To resample volumetric data for a SBW lattice, first we have to apply a weighted average and a downsampling to halve the resolution on X and Y axes and then we apply the half unit shift on alternate rows.

B. Indexing and Storage

Due to the nature of prismatic hexagonal lattices, the points are not aligned in three orthogonal directions as for cubic lattices. Because the hexagon is only 2D, we can use only two skewed axes as in figure 1. Those axes are used in literature by Snyder and Watson [8], [9].

For storage and indexing we can use a simple two-dimensional array treating lattice cells in a zig-zag pattern, row by row. With this method we can achive fast random access to lattice cells.

Another interesting approach is a 7^L layered indexing system named HIP developed by Middleton in [10], [11]. Layer 0 is defined by a single hexagon from which the whole lattice is generated. Layer 1 is simply made out of the surrounding hexagons of Layer 0 laid down in counter-clockwise order. Layer 2 is a Layer 1 tile surrounding Layer 1, and so on for each layer. It's easy to see that there are 7^L hexagons in a layer L tile. Using HIP indexing system it's possible to uniquely index every hexagon in a tile using an L-digit septenary numbers. The diagram in figure 3 demonstrates this. For volumetric data, this indexing system can easily be extended by adding another number for the Z axis and the lattice can be stored as a grayscale image.

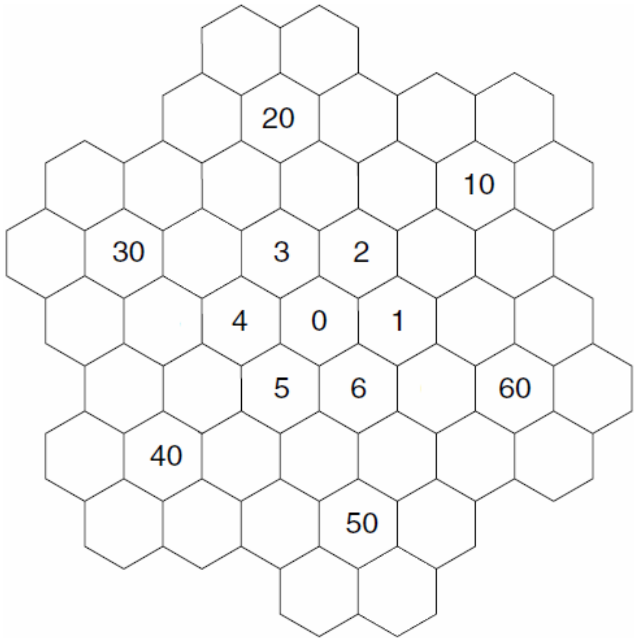

Figure 3. HIP Indexing

IV. Volumetric Data Segmentation

As mentioned earlier, volumetric data processing algorithms need to be redesigned in order to take full advantage of the new lattice.

In this paper we present a volumetric segmentation technique modified to use the SBW lattice. The whole idea is to create a 3D graph using the new lattice and extend the segmentation algorithm presented in [4] to volumetric data.

But first of all we can apply a manual volumetric segmentation using a transfer function to remove noise and to select the region of interest by selecting the hysteresis threshold values. We perform this using cubic splines on the alpha channel and use the output data for the visualization module.

A. Transfer functions

Histograms are useful for analysing which ranges of values are important in the data and show distribution of data but omit spatial distribution of samples in the volume. However, for smooth transitions a cubic spline can be used which can be controlled by control points called knots.

A cubic spline is a spline constructed of piecewise third-order polynomials $Y_i(t) = a_i + b_i * t + c_i * t^2 + d_i * t^3$, where $t \in [0, 1]$ and $i = \overline{0..n-1}$. The cubic spline passes through a set of (n+1) control points $(y_0, y_1, ..., y_n)$. The second derivative of each polynomial is commonly set to zero at the endpoints, since this provides a boundary condition that completes the system of (n-1) equations. This produces a so-called *natural* cubic spline and leads to a simple tridiagonal system which can be solved easily to give the coefficients of the polynomials [12].

$$\begin{bmatrix} 2 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 2 \end{bmatrix} \mathbf{D} = 3 \begin{bmatrix} y_1 - y_0 \\ y_2 - y_0 \\ y_3 - y_1 \\ \vdots \\ y_{n-1} - y_{n-3} \\ y_n - y_{n-2} \\ y_n - y_{n-1} \end{bmatrix}$$

where $\mathbf{D}$ is $D = (D_0, D_1, \cdots, D_n)^T$ and represent the end points derivatives of $Y(t)$.

We then apply these equations for each RGBA colour channel, using in fact a 4D cubic spline transfer function.

For example, we have these 5 colour knots:

```
0.18 0.81 0.81 0
0.3 0.75 0.25 50
0.93 0.93 0.07 100
0.8 0.32 0.2 200
0.0 0.0 0.0 255
```

and 7 alpha knots:

```
0.0 0
0.0 40
0.2 60
0.05 63
0.0 80
0.9 82
1.0 255
```

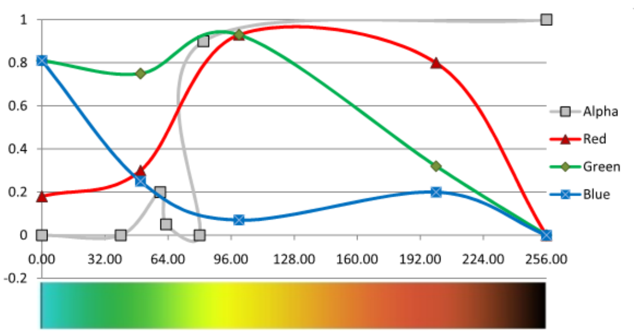

Figure 4. 4D Cubic Spline Spline Transfer Function

The transfer function for the knots is depicted in figure 4. We can observe that we have separated alpha knots from color knots so that we can have a separate control function for opacity. Usually, the visualization algorithm, take the

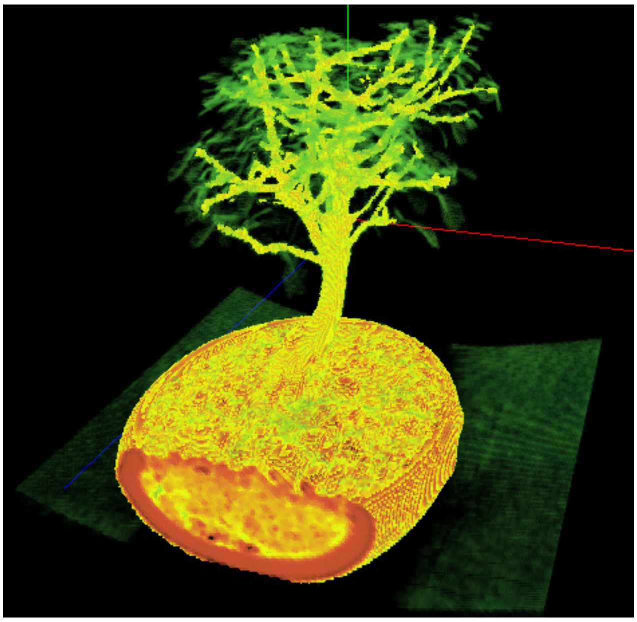

Figure 5. Transfer function applied on volume data

intensity of the voxel as it's density and use it as opacity. With this, a pre-segmentation step is realized in order to remove unwanted background noise. In figure 5 we can see the rendering result after applying the transfer function on the bonsai volume data set (made publicly available by the Visible Human Project).

B. Segmentation on a Graph

We create a weighted undirected graph $G = (V, E)$ between every lattice cell as depicted in figure 6. The weight is determined computing the distance between the cell's average intensities.

$$D = \frac{1}{14} | \sum_{1}^{14} V_1 - \sum_{1}^{14} V_2 |$$

where V_1 and V_2 represent the two lattice cells and each of them containing 14 points. Adjacent cells will share some points, 4 or 7 if the neighbouring cells are on Z axis.

Having the 3D graph we then apply a modified thresholded version of Kruskal's algorithm to find a forest of minimum spanning trees. Each tree corresponds to a different object in the segmentation. The algorithm is described in Algorithm 1.

V. Conclusion and Future Work

In this paper we present the advantages and disadvantages of using the Solid Brick Wall lattice system to acquire, index, store, process and visualize volumetric data in contrast to the classical cubic lattice. We we have extended a hexagonal graph-based image segmentation algorithm to the new 3D

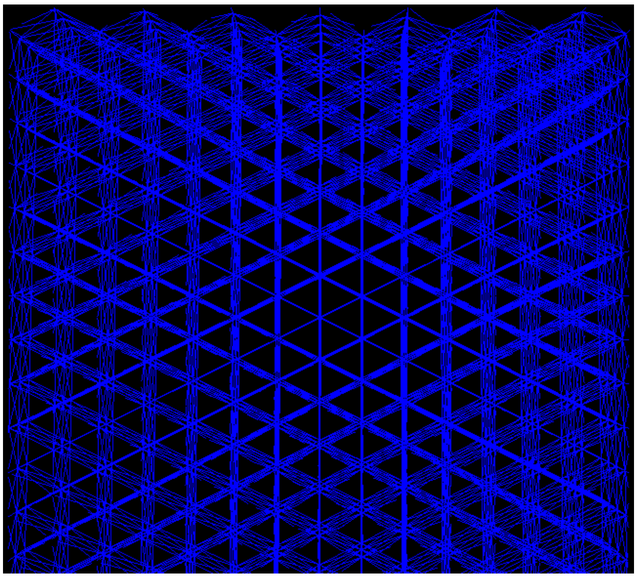

Figure 6. Lattice cell graph

Algorithm 1 Graph-based segmentation

Require: Graph G, Threshold T
Ensure: MinimumSpanningTree MST
  $L \leftarrow 0$
  $S^L = \{\{h_1\}, ..., \{h_{|V|}\}\}$
  Sort E in order of increasing weight
  for $k = 1 \rightarrow |E|$ **do**
    Let $e_k = (h_i, h_j)$
    $t_i \leftarrow FINDSET(h_i)$
    $t_j \leftarrow FINDSET(h_j)$
    if $t_i \neq t_j$ **then**
      if $weight(h_i, h_j) \leq T$ **then**
        $UNION(t_i, t_j, S^L)$
        $L = L + 1$
      end if
    end if
  end for

lattice [4]. The algorithm extension to the three dimensional domain is straightforward as no planarity condition is required for the graph in order the algorithms to work.

The segmentation is now semi-automatic as it depends on the selected transfer function and on the fixed threshold. Future work will include adaptive transfer functions based on histograms and adaptive threshold based on the difference between internal and external contrast at which we add the average of color distances and the standard deviation [13]. We are targeting to develop an automatic graph-based segmentation scheme for volumetric data and we hope to have the same performances as in 2D.

To avoid transforming the volume back to the cubic lattice to perform rendering we will consider using this lattice in a volume global illumination technique as in [14].

This paper represents research at an early stage and so experimental results will be provided in a future work.

References

[1] T. C. Hales, *The Honeycomb Conjecture*. Springer Verlag, vol. 25, pp. 1–22.

[2] ——. (2002) An overview of the Kepler conjecture. [Online]. Available: http://arxiv.org/abs/math/9811071v2

[3] A. P. Fitz and R. J. Green, "Fingerprint classification using a hexagonal fast Fourier transform," *Pattern Recognition*, vol. 29, no. 10, p. 1587 1597, 1996.

[4] D. Burdescu, M. Brezovan, E. Ganea, and L. Stanescu, "A new method for segmentation of images represented in a hsv color space," in *Advanced Concepts for Intelligent Vision Systems*, ser. Lecture Notes in Computer Science. Springer Berlin, 2009, vol. 5807, pp. 606–617.

[5] D. Weaire and R. Phelan, "A counter-example to Kelvin's conjecture on minimal surfaces," *Philosophical Magazine Letters*, vol. 69, no. 2, pp. 107–110, 1994.

[6] R. C. Staunton and N. Storey, "A comparison between square and hexagonal sampling methods for pipeline image processing," in *Proc. SPIE*, vol. 1194, 1989, p. 142 151.

[7] W. Neeser and et. all, "DEPFET pixel bioscope," *IEEE Transactions on Nuclear Science*, vol. 47, no. 3III, p. 1246 1250, 2000.

[8] H. Q. W. Snyder and W. Sander, "A coordinate system for hexagonal pixels," *Proceedings of the International Society for Optical Engineering*, vol. 3661, p. 716 727, 1999.

[9] A. B. Watson and A. J. Ahumada, "A hexagonal orthogonal-oriented pyramid as a model of image representation in the visual cortex," *IEEE Transactions on Biomedical Engineering*, vol. BME, no. 36, p. 97 106, January 1989.

[10] L. Middleton and J. Sivaswamy, *Edge detection in a hexagonal-image processing framework*. Springer Verlag, December 2001, vol. 19, no. 14, p. 1071 1081.

[11] ——, *Hexagonal Image Processing : A Practical Approach*. Springer Verlag, 2005.

[12] J. C. Bartels, R. H.; Beatty and B. A. Barsky, "Hermite and cubic spline interpolation," in *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*. San Francisco, CA: Morgan Kaufmann, 1998, ch. 3, p. 9 17.

[13] E. G. Marius Brezovan, Dumitru Dan Burdescu and L. Stanescu, "Boundary-based performance evaluation of a salient visual object detection algorithm," in *IPCV'10*, 2010, pp. 694–700.

[14] F. Qiu, F. Xu, Z. Fan, N. Neophytos, A. Kaufman, and K. Mueller, "Lattice-based volumetric global illumination," 2007.