

Three Dimensional Image Processing in Hexagonal Prism Lattice of Z^3 Grid

¹Mohd. Sherfuddin Khan, ²E. G. Rajan, ³Vijay H. Mankar

¹Research Scholar, G.H Raison College of Engineering
Nagpur, Maharashtra, India

²President, Pentagram research Centre
Hyderabad, Telangana State, India

³HOD (ECE), Government Polytechnic
Ahmednagar, Maharashtra, India

sherfuddin.phd@gmail.com; rajaneg@yahoo.co.in; vhmankar@gmail.com

ABSTRACT

A 2D hexagonal image is an array of what are called pixels which are the coordinates of the hexagonal lattice points decided by the linear horizontal rows and the nonlinear vertical zig-zags. A 2D hexagonal image could also be informally called as image slice or a matrix of pixel values arranged in a hexagonal array. A 3D hexagonal image is viewed as an ordered sequence of 2D hexagonal image slices arranged in the z-direction and the 3D arrangement of voxel values is called as a prism of voxel values. Most of the 3D hexagonal image processing operations are similar to those of 2D hexagonal image processing. 3D hexagonal images are processed with the help of 3D hexagonal scanning windows, whereas 2D hexagonal images are processed with the help of 2D hexagonal scanning windows. For instance, a 3D hexagonal image processing operation like 3D surface detection is carried out using analogous 2D edge detection algorithm on every image slice and the processed slices assembled to visualize 3D surface detected image. In fact, 2D contours of an image slice are called superficial features and closed surfaces of a 3D image are called volumetric features. One can always obtain surface detected version of a 3D hexagonal image by processing the 2D hexagonal image slices using 2D edge detection operation, and consequently the 3D surface detection operation is termed as 2.5D hexagonal image processing. One could also process the 3D hexagonal image data using a 3D surface detection algorithm, in which case it is termed as 3D hexagonal image processing. This is not the case with the operation of skeletonization. One cannot make use of 2.5D skeletonization operation of 2D image slices in order to get skeletonized version of the corresponding 3D image. In fact, one would come across discrepancies and differences when 2.5D skeletonization of 2D hexagonal image slices of a 3D hexagonal image is carried out instead of the direct 3D skeletonization of the 3D hexagonal image. This paper highlights certain 3D algorithms for processing 3D hexagonal images.

Keywords: Hexagonal Lattice Grid, Hexagonal Lattice Grid Images, Hexagonal Lattice Grid Image Processing

1 Introduction

Regular hexagons could be tiled up as a tessellation of a hexagonal lattice in order to display digital images with the idea of visualizing images with better curvilinear property. The extraction of features

DOI: 10.14738/aivp.53.3241

Publication Date: 23rd June, 2017

URL: <http://dx.doi.org/10.14738/aivp.53.3241>

on Rectangular lattice is done by some researches [1]-[27] in two and three dimensional grids but extraction of features on two dimensional hexagonal grid is done by [28]-[53] but extraction of features on hexagonal lattice prism is done no where in the literature.

The technique of realizing a hexagonal lattice in Z^2 grid is briefly explained in the following.

1.1 Two Dimensional Image Visualization in Hexagonal Lattice of Z^2 Grid

A sample Z^2 grid of size 14x25 is a two dimensional array of nodes and it is shown in Fig. 1. The addressing scheme of those lattice points (nodes) is also demonstrated in Fig. 1. Hexagonal lattice in a Z^2 grid is a two dimensional array of nodes, which is shown in Fig. 2. The addressing scheme of those lattice points (nodes) is also demonstrated in Fig. 2. As per Fig. 2, the coordinates of the hexagonal lattice points are decided by the linear horizontal rows and the nonlinear vertical zig-zags.

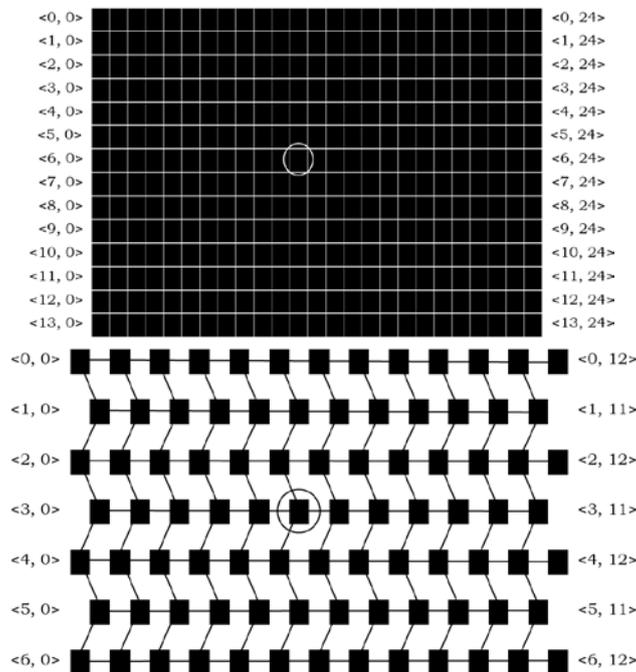


Figure 1: Two dimensional rectangular lattice points in Z^2 grid Fig. 2: Hexagonal lattice points decided by rows and zig-zags

Now, each node in the hexagonal lattice (Fig. 2) is related to the corresponding node in the rectangular lattice (Fig. 1) by the following formulas: (i) for even row elements of the hexagonal lattice (0^{th} row is an even row), $\langle Hx_i, Hy_j \rangle = \langle Rx_{2i}, Ry_{2j} \rangle$ and (ii) for odd row elements of the hexagonal lattice $\langle Hx_i, Hy_j \rangle = \langle Rx_{2i}, Ry_{2j+1} \rangle$. Fig. 3 shows a sample image displayed on rectangular and hexagonal lattice points in Z^2 grid.

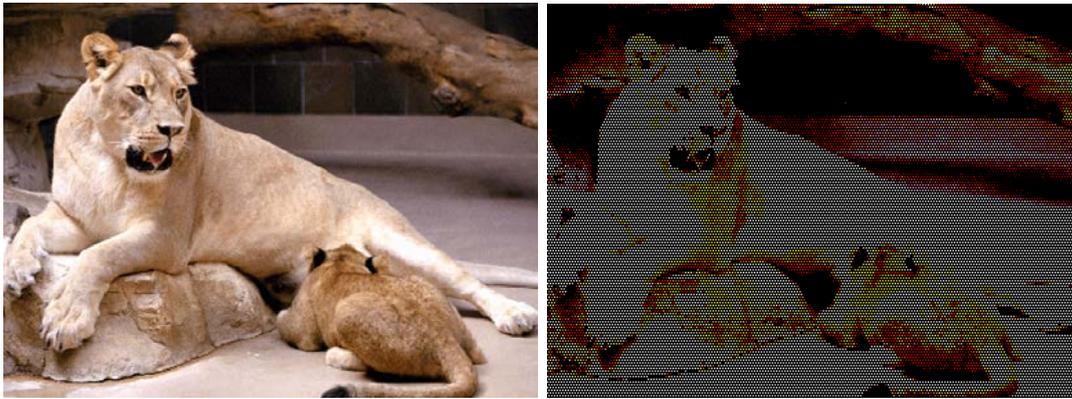


Figure. 3: Sample image displayed on rectangular and hexagonal lattice points in Z^2 grid

1.2 Three Dimensional Image Visualization in Hexagonal Lattice of Z^3 Grid

Nodes in a two dimensional array are called pixels. Similarly, nodes in the three dimensional array are called voxels. A voxel in a 3D lattice is shown in figure 4.

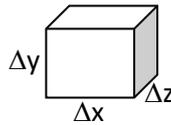


Figure. 4: Voxel in a 3D lattice

Figure. 5 shows a $10 \times 10 \times 10$ array of voxels. Each voxel could be represented by a gray tone or a color in terms of Red, Green and Blue. If Red, Green and Blue values are identical then the voxel is a gray level voxel.

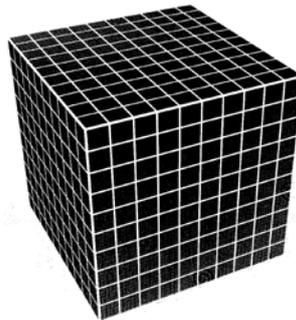


Figure. 5: Simple 3D array of voxels of size $10 \times 10 \times 10$

It is not possible to surmise a three dimensional hexagonal lattice which has a symmetry in all directions. One may try to construct a 3D hexagonal lattice using six regular hexagons as it is done in the case of 3D rectangular lattice. In fact, one can think of a hexagonal unit prism consisting of 12 nodes for tiling which would evolve a three dimensional hexagonal prism lattice. Fig. 6 shows one such hexagonal unit prism which could act as a fundamental building block in the construction of such a lattice and two such units concatenated.

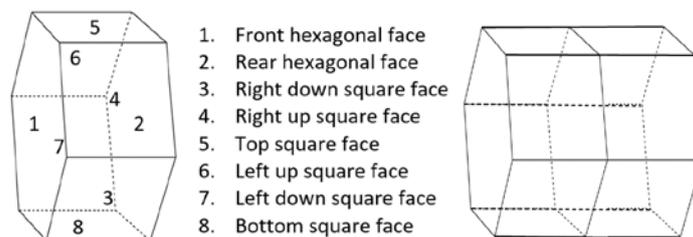


Figure. 6: Hexagonal unit prism consisting of 12 nodes and two such units concatenated

Figure. 7 shows a 3D hexagonal prism lattice of voxels obtained from a 3D rectangular lattice of voxels. Now, each node in the hexagonal lattice (Fig. 7) is related to the corresponding node in a rectangular lattice by the following formulas: (i) For even row elements of k^{th} voxel array in hexagonal lattice (Note: 0^{th} row is even row), $\langle Hx_i, Hy_j, Hz_k \rangle = \langle Rx_{2i}, Ry_{2j}, Rz_{2k} \rangle$, and (ii) For odd row elements of k^{th} voxel array in hexagonal lattice $\langle Hx_i, Hy_j, Hz_k \rangle = \langle Rx_{2i}, Ry_{2j+1}, Rz_{2k} \rangle$. As per Fig. 7, the coordinates of the hexagonal lattice points in a voxel array are decided by the linear horizontal rows and the nonlinear vertical zig-zags.

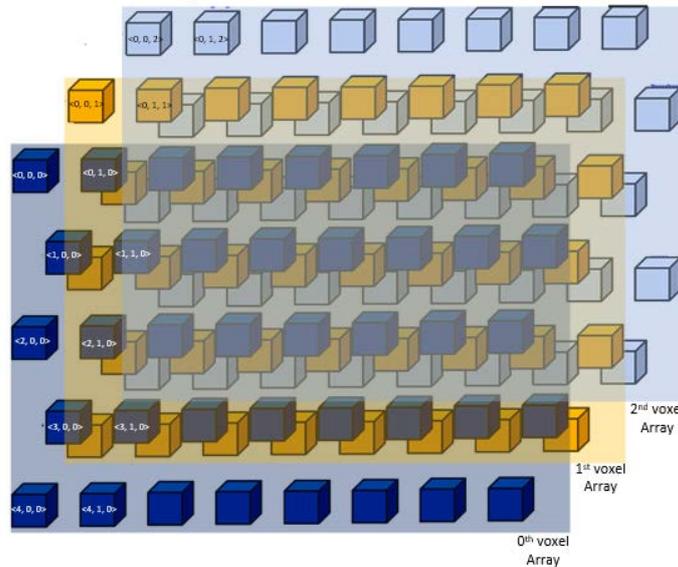


Figure. 7: Hexagonal lattice points depicted in Z^3 grid

Figure 8 shows a sample 3D image displayed on a regular rectangular lattice points in Z^2 grid and the same image displayed on a simulated 3D hexagonal symmetric prism lattice.

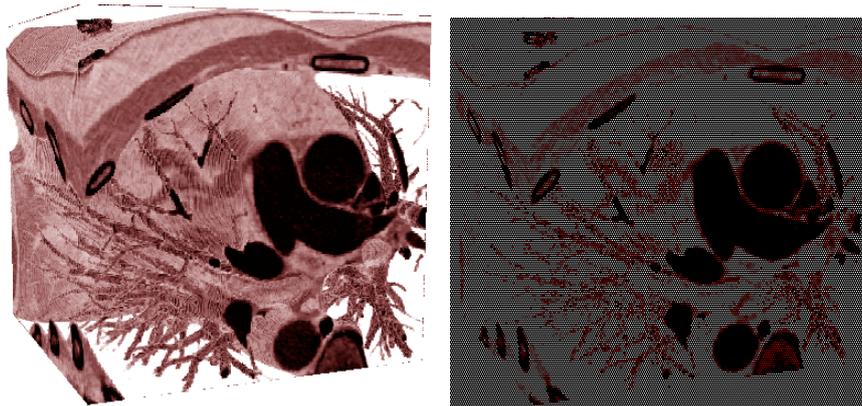


Figure. 7: Sample image displayed on rectangular lattice in Z^3 grid and same image displayed on hexagonal lattice in Z^3 grid

After visualizing 3D images on a hexagonal lattice, one would try to process the images as per requirement. 3D image processing is basically a complex operation even in the case of rectangular lattice based images. It is more complex in the case of hexagonal lattice based 3D images. This paper presents results of an extensive research carried out to process 3D hexagonal prism lattice based images and it consists of two sections, (i) processing of 3D hexagonal images using 2D algorithms and (ii) processing of 3D hexagonal images using 3D algorithms.

2 Processing of 3D hexagonal images using 2D algorithms

Cellular logic array processing of digital images is the basic paradigm that is used to process 2D and 3D hexagonal images. The hexagonal image to be processed is treated here as the initial configuration of a cellular automaton and the processed hexagonal image as its final configuration. The updating rule of the cellular automaton is viewed here as the desired hexagonal image processing operation. In this framework of cellular logic array processing one can develop algorithms for implementing morphological operations and also traditional operations like binarization, segmentation, and edge detection on hexagonal images. Most of the conventional image processing operations are carried out using numerical calculations, and so use of pattern directed algorithms for their implementation may lead to more complexity and more CPU time. So, we make use of less complex numerical operations like addition and subtraction wherever it is necessary in our fast operating pattern-directed algorithms so that we achieve a high throughput image processing system. Given a 3D hexagonal image, one can use a 2D algorithm to process it slice by slice in order to achieve an overall processing effect on the image.

This type of processing 3D images using 2D algorithms is called '2.5D processing' of 3D images. As a test case study, the operations of 2.5D edge detection and 2.5D skeletonization are carried out on two real time MR images and results shown with the idea of highlighting the significance of cellular logic array processing.

2.1 Two dimensional scanning windows for 2.5D processing of 3D hexagonal images

The given hexagonal image is scanned by any of the 18 structuring elements shown in Fig. 8. In fact, one would use five structuring elements $C_{1,4}$, $C_{2,5}$, $C_{3,6}$, $D_{1,3,5}$ and $D_{2,4,6}$ which are called 'basis patterns' for this purpose. Fig. 8 also shows the 5-level lattice structure consisting of the 18 patterns with the relation 'contained in' denoted by the symbol \subset . For instance, the relation $D_{1,3,5} \subset C_{1,3}$ denotes that the pattern $D_{1,3,5}$ is contained in the pattern $C_{1,3}$. Note that there are five basis patterns $C_{1,4}$, $C_{2,5}$, $C_{3,6}$, $D_{1,3,5}$ and $D_{2,4,6}$ (shaded black) in Fig. 8. All these five basis patterns have to be used while scanning a hexagonal image.

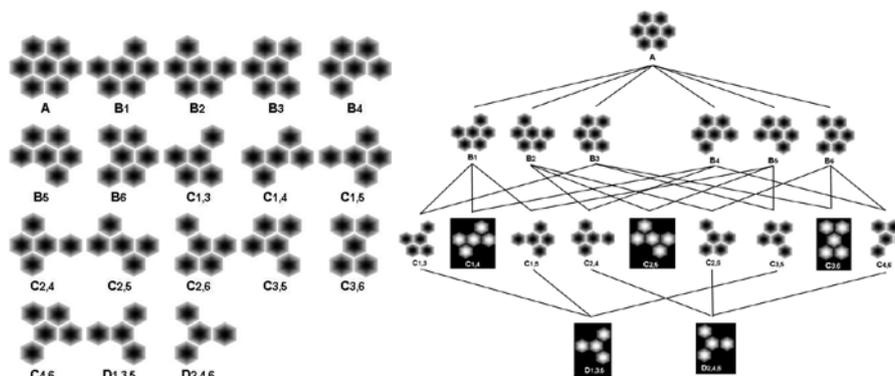


Figure 8: Eighteen convex polygons over the basic polygon A and its lattice structure

Based on the above argument, it is sufficient to make use of the neighbourhood structures $C_{1,4}$, $C_{2,5}$, $C_{3,6}$, $D_{1,3,5}$ and $D_{2,4,6}$ (masks) for processing a 2D hexagonal image because checking for the presence of these five neighbourhoods in a hexagonal image ensures checking of all possible 18 convex patterns in the image.

2.1.1 Edge detection in 2D hexagonal images using 2D algorithm

A region that appears to have a single gray-level may really contain many adjacent gray-levels. They appear to be the same because of the effect of visual quantization exercised by an observer. Segmentation is a process of partitioning a given gray-level image into disjoint regions each of which appears to an observer to have a single gray-level. So, a given image is first segmented using a threshold-based-quantization method and after that the boundaries of the quantized regions are detected. On every move, the sub image enclosed by any of the five neighbourhood windows is examined to see whether the D (gray-distance), which is the difference between the maximum and the minimum gray-values analogous to that sub image, is less than or equal to a threshold value T . If D is equal or less than to T , then the gray-value 0 is assigned to central cell; otherwise the value in central cell left unchanged. This procedure is repeated till the whole hexagonal image is scanned. The final outcome is that the boundaries of different regions in the given hexagonal image, that appear to be uniform, are retained and their interior parts are removed thus giving us the edge detected edition of the original image. This operation is carried out by pattern directed if-then rules. In the case of color images, the following method is used to detect the edges.

Repeat sliding the five structuring elements over the image

```
{
  Examine all pixels of a nonzero neighbourhood;
  Get the maximum Red color value Red_max;
  Get the minimum Red color value Red_min;
  Then find the difference between these two values  $D_{Red}$ ;
  Get the maximum Green color value Green_max;
  Get the minimum Green color value Green_min;
  Then find the difference  $D_{Green}$ ;
  Get the maximum Blue color value Blue_max;
  Get minimum Blue color value Blue_min;
  Then find the difference  $D_{Blue}$ ;
  When all the values  $D_{Red}$ ,  $D_{Green}$  and  $D_{Blue}$  are less than or equal to a threshold value  $T$  then substitute the
  central pixel value with  $R = G = B = 0$ , else slide the five structuring elements
}
```

until the structuring elements span whole of the image.

Fig 9 shows a sample hexagonal 2D image and its edge detected version.

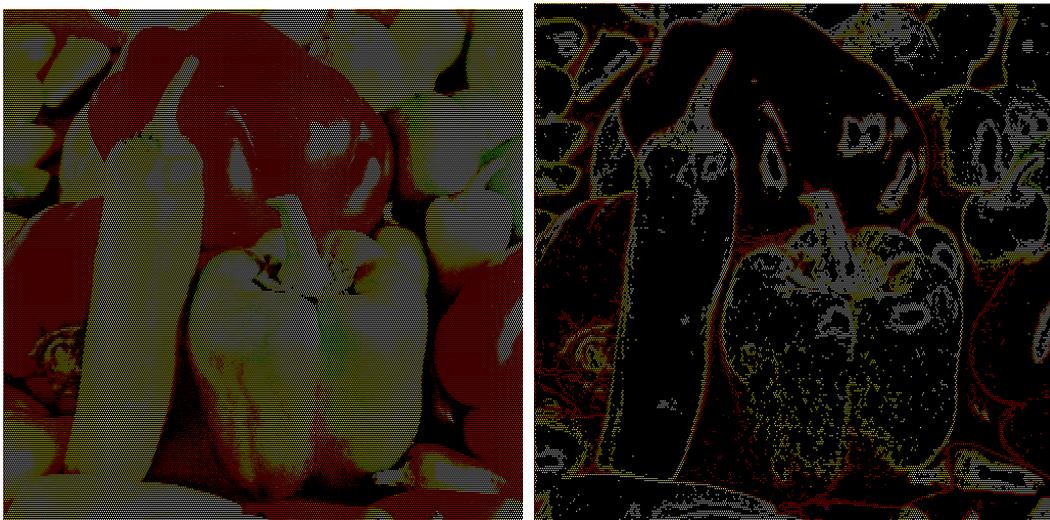


Figure. 9: Sample hexagonal colour image and its edge detected version

2.1.2 Skeletonization of 2D hexagonal images using 2D algorithm

On every move, the sub image enclosed by any of the five scanning windows is examined to see whether the D (gray-distance), which is the difference between the maximum and the minimum gray-

values analogous to that sub image, is less than or equal to a threshold value T . If D is equal or less than to T , then the gray-value of the central cell is preserved and the boundary pixel removed. This procedure is repeated till the whole image is scanned. The final outcome is that the boundaries of different regions in the given image, that appear to be uniform, are removed and their interior parts are retained thus giving us the skeleton edition of the original image. Note that this operation is carried out by pattern directed if-then rules. In the color images case, the following method is used to detect the edges.

Repeat sliding the five structuring elements over the image

```
{
    Examine all pixels of a nonzero neighborhood;
    Get the maximum Red color value Red_max;
    Get the minimum Red color value Red_min;
    Then find the difference between these two values  $D_{Red}$ ;
    Get the maximum Green color value Green_max;
    Get the minimum Green color value Green_min;
    Then find the difference  $D_{Green}$ ;
    Get the maximum Blue color value Blue_max;
    Get minimum Blue color value Blue_min;
    Then find the difference  $D_{Blue}$ ;
    When all the values  $D_{Red}$ ,  $D_{Green}$  and  $D_{Blue}$  are less than or equal to a threshold value  $T$  then retain the
    central pixel value, else slide the five scanning windows
}
```

until the structuring element scans the entire image; repeat this procedure till there is no boundary detected.

Figure. 10 shows a sample hexagonal 2D image and its skeletal form.

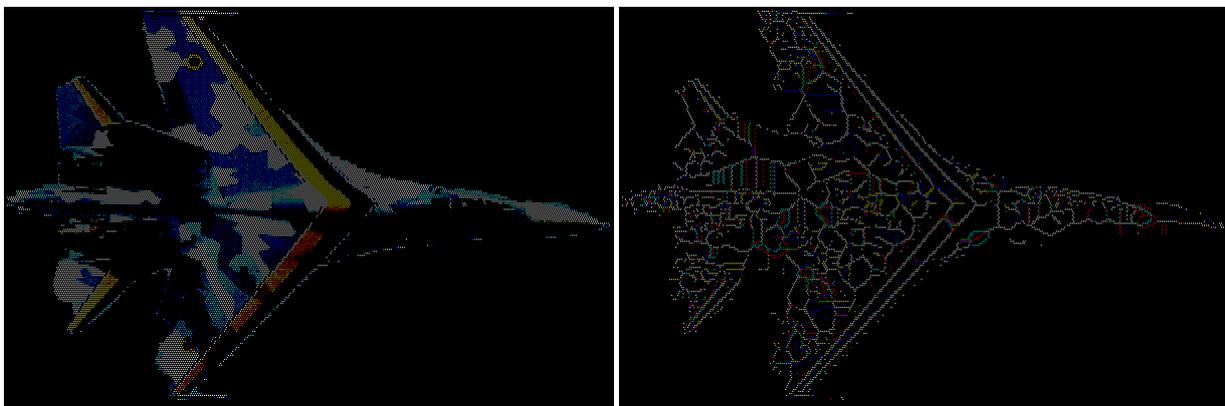


Figure 10: Sample hexagonal 2D color image and its skeletal form

2.2 Processing of 3D images using 2.5D algorithms

One can process 3D hexagonal images using 2D algorithms slice by slice and integrate the processed 2D slices as a 3D image. ***In this case also, the 2D algorithms are called 2.5D algorithms because they are applied to a set of 2D slices of a 3D image.*** Fig. 11 shows an MR image of a human heart, its 2.5D edge detected and 2.5D skeletonized versions.

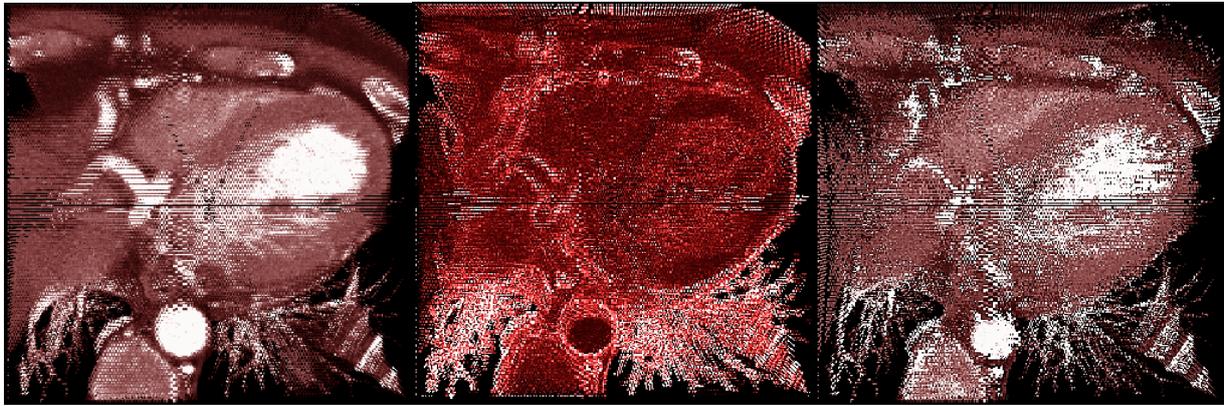


Figure 11: Sample MR image of a human heart, its 2.5D edge detected and 2.5D skeletonized versions

With reference to Fig. 11 and to an in depth study by carrying out edge detection and skeletonization operations on several varieties of 3D hexagonal images, it has been observed that the result of processing 3D hexagonal images using 2.5D algorithms slice wise does not yield precise results. Alternatively, it has been found that the results of processing 3D hexagonal images using 3D algorithms yield better results.

3 Processing of 3D hexagonal images using 3D algorithms

The notion of three dimensional scanning windows for 3D processing of 3D hexagonal images is briefly described in this section. 3D algorithms make use of one of 324 unique convex 3-D hexagonal polyhedrons (hexagonal prisms).

3.1 Three dimensional scanning windows for 3D processing of 3D hexagonal images

Consider a 21- neighbourhood structure in the three-dimensional hexagonal grid as in Fig. 12.

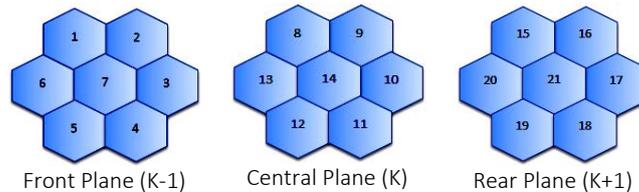


Figure. 12: 21- neighbourhood structure

One can construct a total of 324 convex polyhedrons (prisms) in the 21-neighborhood structure. These are listed below under seven groups among which 25 of them are basis convex hexagonal polyhedrons.

Group A: No voxel is eliminated we get 1 combination

$$A = \{1,2,3,4,5,6,15, 16,17,18,19,20\}$$

Group B: Eliminating one voxel we obtain 12 combinations

$$B_1 = \{2,3,4,5,6,15,16,17,18,19,20\}; B_2 = \{1,3,4,5,6,15,16,17,18,19,20\}; B_3 = \{1,2,4,5,6,15,16,17,18,19,20\};$$

$$B_4 = \{1,2,3,5,6,15,16,17,18,19,20\}; B_5 = \{1,2,3,4,6,15,16,17,18,19,20\}; B_6 = \{1,2,3,4,5,15,16,17,18,19,20\};$$

$$B_{15} = \{1,2,3,4,5,6,16,17,18,19,20\}; B_{16} = \{1,2,3,4,5,6,15,17,18,19,20\}; B_{17} = \{1,2,3,4,5,6,15,16,18,19,20\};$$

$$B_{18} = \{1,2,3,4,5,6,15,16,17,19,20\}; B_{19} = \{1,2,3,4,5,6,15,16,17,18,20\}; B_{20} = \{1,2,3,4,5,6,15,16,17,18,19\};$$

Group C: Eliminating two voxels we obtain 54 combinations

$$C_{1,3} = \{2,4,5,6,15,16,17,18,19,20\}; C_{1,4} = \{2,3,5,6,15,16,17,18,19,20\}; C_{1,5} = \{2,3,4,6,15,16,17,18,19,20\};$$

$$C_{1,15} = \{2,3,4,5,6,16,17,18,19,20\}; C_{1,16} = \{2,3,4,5,6,15,17,18,19,20\}; C_{1,17} = \{2,3,4,5,6,15,16,18,19,20\};$$

$$C_{1,18} = \{2,3,4,5,6,15,16,17,19,20\}; C_{1,19} = \{2,3,4,5,6,15,16,17,18,20\}; C_{1,20} = \{2,3,4,5,6,15,16,17,18,19\};$$

$$C_{2,4} = \{1,3,5,6,15,16,17,18,19,20\}; C_{2,5} = \{1,3,4,6,15,16,17,18,19,20\}; C_{2,6} = \{1,3,4,5,15,16,17,18,19,20\};$$

$$C_{2,15} = \{1,3,4,5,6,16,17,18,19,20\}; C_{2,16} = \{1,3,4,5,6,15,17,18,19,20\}; C_{2,17} = \{1,3,4,5,6,15,16,18,19,20\};$$

$$C_{2,18} = \{1,3,4,5,6,15,16,17,19,20\}; C_{2,19} = \{1,3,4,5,6,15,16,17,18,20\}; C_{2,20} = \{1,3,4,5,6,15,16,17,18,19\};$$

- | | | |
|---|---|--|
| C _{3,5} = {1,2,4,6,15,16,17,18,19,20}; | C _{3,6} = {1,2,4,5,15,16,17,18,19,20}; | C _{3,15} = {1,2,4,5,6,16,17,18,19,20}; |
| C _{3,16} = {1,2,4,5,6,15,17,18,19,20}; | C _{3,17} = {1,2,4,5,6,15,16,18,19,20}; | C _{3,18} = {1,2,4,5,6,15,16,17,18,19,20}; |
| C _{3,19} = {1,2,4,5,6,15,16,17,18,20}; | C _{3,20} = {1,2,4,5,6,15,16,17,18,19}; | C _{4,6} = {1,2,3,5,15,16,17,18,19,20}; |
| C _{4,15} = {1,2,3,5,6,16,17,18,19,20}; | C _{4,16} = {1,2,3,5,6,15,17,18,19,20}; | C _{4,17} = {1,2,3,5,6,15,16,17,18,19,20}; |
| C _{4,18} = {1,2,3,5,6,15,16,17,19,20}; | C _{4,19} = {1,2,3,5,6,15,16,17,18,20}; | C _{4,20} = {1,2,3,5,6,15,16,17,18,19}; |
| C _{5,15} = {1,2,3,4,6,16,17,18,19,20}; | C _{5,16} = {1,2,3,4,6,15,17,18,19,20}; | C _{5,17} = {1,2,3,4,6,15,16,18,19,20}; |
| C _{5,18} = {1,2,3,4,6,15,16,17,19,20}; | C _{5,19} = {1,2,3,4,6,15,16,17,18,20}; | C _{5,20} = {1,2,3,4,6,15,16,17,18,19}; |
| C _{6,15} = {1,2,3,4,5,16,17,18,19,20}; | C _{6,16} = {1,2,3,4,5,15,17,18,19,20}; | C _{6,17} = {1,2,3,4,5,15,16,18,19,20}; |
| C _{6,18} = {1,2,3,4,5,15,16,17,19,20}; | C _{6,19} = {1,2,3,4,5,15,16,17,18,20}; | C _{6,20} = {1,2,3,4,5,15,16,17,18,19}; |
| C _{15,17} = {1,2,3,4,5,6,16,18,19,20}; | C _{15,18} = {1,2,3,4,5,6,16,17,19,20}; | C _{15,19} = {1,2,3,4,5,6,16,17,18,20}; |
| C _{16,18} = {1,2,3,4,5,6,15,17,19,20}; | C _{16,19} = {1,2,3,4,5,6,15,17,18,20}; | C _{16,20} = {1,2,3,4,5,6,16,17,18,19}; |
| C _{17,19} = {1,2,3,4,5,6,15,16,18,20}; | C _{17,20} = {1,2,3,4,5,6,16,16,18,19}; | C _{18,20} = {1,2,3,4,5,6,16,16,17,19}; |

Group D: Eliminating three voxels we obtain 112 combinations

- | | | |
|---|---|--|
| D _{1,3,5} = {2,4,6,15,16,17,18,19,20}; | D _{1,3,15} = {2,4,5,6,16,17,18,19,20}; | D _{1,3,16} = {2,4,5,6,15,17,18,19,20}; |
| D _{1,3,17} = {2,4,5,6,15,16,18,19,20}; | D _{1,3,18} = {2,4,5,6,15,16,17,19,20}; | D _{1,3,19} = {2,4,5,6,15,16,17,18,20}; |
| D _{1,3,20} = {2,4,5,6,15,16,17,18,19}; | D _{1,4,15} = {2,3,5,6,16,17,18,19,20}; | D _{1,4,16} = {2,3,5,6,15,17,18,19,20}; |
| D _{1,4,17} = {2,3,5,6,15,16,18,19,20}; | D _{1,4,18} = {2,3,5,6,15,16,17,19,20}; | D _{1,4,19} = {2,3,5,6,15,16,17,18,20}; |
| D _{1,4,20} = {2,3,5,6,15,16,17,18,19}; | D _{1,5,15} = {2,3,4,6,16,17,18,19,20}; | D _{1,5,16} = {2,3,4,6,15,17,18,19,20}; |
| D _{1,5,17} = {2,3,4,6,15,16,18,19,20}; | D _{1,5,18} = {2,3,4,6,15,16,17,19,20}; | D _{1,5,19} = {2,3,4,6,15,16,17,18,20}; |
| D _{1,5,20} = {2,3,4,6,15,16,17,18,19}; | D _{1,15,17} = {2,3,4,5,6,16,18,19,20}; | D _{1,15,18} = {2,3,4,5,6,16,17,19,20}; |
| D _{1,15,19} = {2,3,4,5,6,16,17,18,20}; | D _{1,16,18} = {2,3,4,5,6,15,17,19,20}; | D _{1,16,19} = {2,3,4,5,6,15,17,18,20}; |
| D _{1,16,20} = {2,3,4,5,6,15,17,18,19}; | D _{1,17,19} = {2,3,4,5,6,15,16,18,20}; | D _{1,17,20} = {2,3,4,5,6,15,16,18,19}; |
| D _{1,18,20} = {2,3,4,5,6,15,16,17,19}; | D _{2,4,6} = {1,3,5,15,16,17,18,19,20}; | D _{2,4,15} = {1,3,5,6,16,17,18,19,20}; |
| D _{2,4,16} = {1,3,5,6,15,17,18,19,20}; | D _{2,4,17} = {1,3,5,6,15,16,18,19,20}; | D _{2,4,18} = {1,3,5,6,15,16,17,19,20}; |
| D _{2,4,19} = {1,3,5,6,15,16,17,18,20}; | D _{2,4,20} = {1,3,5,6,15,16,17,18,19}; | D _{2,5,15} = {1,3,4,6,16,17,18,19,20}; |
| D _{2,5,16} = {1,3,4,6,15,17,18,19,20}; | D _{2,5,17} = {1,3,4,6,15,16,18,19,20}; | D _{2,5,18} = {1,3,4,6,15,16,17,19,20}; |
| D _{2,5,19} = {1,3,4,6,15,16,17,18,20}; | D _{2,5,20} = {1,3,4,6,15,16,17,18,19}; | D _{2,6,15} = {1,3,4,5,16,17,18,19,20}; |
| D _{2,6,16} = {1,3,4,5,15,17,18,19,20}; | D _{2,6,17} = {1,3,4,5,15,16,18,19,20}; | D _{2,6,18} = {1,3,4,5,15,16,17,19,20}; |
| D _{2,6,19} = {1,3,4,5,15,16,17,18,20}; | D _{2,6,20} = {1,3,4,5,15,16,17,18,19}; | D _{2,15,17} = {1,3,4,5,6,16,18,19,20}; |
| D _{2,15,18} = {1,3,4,5,6,16,17,19,20}; | D _{2,15,19} = {1,3,4,5,6,16,17,18,20}; | D _{2,16,18} = {1,3,4,5,6,16,17,19,20}; |
| D _{2,16,19} = {1,3,4,5,6,16,17,18,20}; | D _{2,16,20} = {1,3,4,5,6,16,17,18,19}; | D _{2,17,19} = {1,3,4,5,6,15,16,18,20}; |
| D _{2,17,20} = {1,3,4,5,6,15,16,18,19}; | D _{2,18,20} = {1,3,4,5,6,15,16,17,19}; | D _{3,5,15} = {1,2,4,6,16,17,18,19,20}; |
| D _{3,5,16} = {1,2,4,6,15,17,18,19,20}; | D _{3,5,17} = {1,2,4,6,15,16,18,19,20}; | D _{3,5,18} = {1,2,4,6,15,16,17,19,20}; |
| D _{3,5,19} = {1,2,4,6,15,16,17,18,20}; | D _{3,5,20} = {1,2,4,6,15,16,17,18,19}; | D _{3,6,15} = {1,2,4,5,16,17,18,19,20}; |
| D _{3,6,16} = {1,2,4,5,15,17,18,19,20}; | D _{3,6,17} = {1,2,4,5,15,16,18,19,20}; | D _{3,6,18} = {1,2,4,5,15,16,17,19,20}; |
| D _{3,6,19} = {1,2,4,5,15,16,17,18,20}; | D _{3,6,20} = {1,2,4,5,15,16,17,18,19}; | D _{3,15,17} = {1,2,4,5,6,16,18,19,20}; |
| D _{3,15,18} = {1,2,4,5,6,16,17,19,20}; | D _{3,15,19} = {1,2,4,5,6,16,17,18,20}; | D _{3,16,18} = {1,2,4,5,6,15,17,19,20}; |
| D _{3,16,19} = {1,2,4,5,6,15,17,18,20}; | D _{3,16,20} = {1,2,4,5,6,15,17,18,19}; | D _{3,17,19} = {1,2,4,5,6,15,16,18,19,20}; |
| D _{3,17,20} = {1,2,4,5,6,15,16,18,19}; | D _{3,18,20} = {1,2,4,5,6,15,16,17,19}; | D _{4,6,15} = {1,2,3,5,16,17,18,19,20}; |
| D _{4,6,16} = {1,2,3,5,15,17,18,19,20}; | D _{4,6,17} = {1,2,3,5,15,16,18,19,20}; | D _{4,6,18} = {1,2,3,5,15,16,17,19,20}; |
| D _{4,6,19} = {1,2,3,5,15,16,17,18,20}; | D _{4,6,20} = {1,2,3,5,15,16,17,18,19}; | D _{4,15,17} = {1,2,3,5,6,16,18,19,20}; |
| D _{4,15,18} = {1,2,3,5,6,16,17,19,20}; | D _{4,15,19} = {1,2,3,5,6,16,17,18,20}; | D _{4,16,18} = {1,2,3,5,6,15,17,19,20}; |
| D _{4,16,19} = {1,2,3,5,6,15,17,18,20}; | D _{4,16,20} = {1,2,3,5,6,15,17,18,19}; | D _{4,17,19} = {1,2,3,5,6,15,16,18,20}; |
| D _{4,17,20} = {1,2,3,5,6,15,16,18,19}; | D _{4,18,20} = {1,2,3,5,6,15,16,17,19}; | D _{5,15,17} = {1,2,3,4,6,16,18,19,20}; |
| D _{5,15,18} = {1,2,3,4,6,16,17,19,20}; | D _{5,15,19} = {1,2,3,4,6,16,17,18,20}; | D _{5,16,18} = {1,2,3,4,6,15,17,19,20}; |
| D _{5,16,19} = {1,2,3,4,6,15,17,18,20}; | D _{5,16,20} = {1,2,3,4,6,15,17,18,19}; | D _{5,17,19} = {1,2,3,4,6,15,16,18,20}; |
| D _{5,17,20} = {1,2,3,4,6,15,16,18,19}; | D _{5,18,20} = {1,2,3,4,6,15,16,17,19}; | D _{6,15,17} = {1,2,3,4,5,16,18,19,20}; |
| D _{6,15,18} = {1,2,3,4,5,16,17,19,20}; | D _{6,15,19} = {1,2,3,4,5,16,17,18,20}; | D _{6,16,18} = {1,2,3,4,5,15,17,19,20}; |
| D _{6,16,19} = {1,2,3,4,5,15,17,18,20}; | D _{6,16,20} = {1,2,3,4,5,15,17,18,19}; | D _{6,17,19} = {1,2,3,4,5,15,16,18,20}; |
| D _{6,17,20} = {1,2,3,4,5,15,16,18,19}; | D _{6,18,20} = {1,2,3,4,5,15,16,17,19}; | D _{15,17,19} = {1,2,3,4,5,6,16,18,20}; |
| D _{16,18,20} = {1,2,3,4,5,6,15,17,19} | | |

Group E: Eliminating four voxels we obtain 105 combinations

- | | | |
|---|---|---|
| E _{1,3,5,15} = {2,4,6,16,17,18,19,20}; | E _{1,3,5,16} = {2,4,6,15,17,18,19,20}; | E _{1,3,5,17} = {2,4,6,15,16,18,19,20}; |
| E _{1,3,5,18} = {2,4,6,15,16,17,19,20}; | E _{1,3,5,19} = {2,4,6,15,16,17,18,20}; | E _{1,3,5,20} = {2,4,6,15,16,17,18,19}; |
| E _{1,3,15,17} = {2,4,5,6,16,18,19,20}; | E _{1,3,15,18} = {2,4,5,6,16,17,19,20}; | E _{1,3,15,19} = {2,4,5,6,16,17,18,20}; |
| E _{1,3,16,18} = {2,4,5,6,15,17,19,20}; | E _{1,3,16,19} = {2,4,5,6,15,17,18,20}; | E _{1,3,16,20} = {2,4,5,6,15,17,18,19}; |
| E _{1,3,17,19} = {2,4,5,6,15,16,18,20}; | E _{1,3,17,20} = {2,4,5,6,15,16,17,20}; | E _{1,3,18,20} = {2,4,5,6,15,16,17,19}; |
| E _{1,4,15,17} = {2,3,5,6,16,18,19,20}; | E _{1,4,15,18} = {2,3,5,6,16,17,19,20}; | E _{1,4,15,19} = {2,3,5,6,16,17,18,20}; |
| E _{1,4,16,18} = {2,3,5,6,15,17,19,20}; | E _{1,4,16,19} = {2,3,5,6,15,17,18,20}; | E _{1,4,16,20} = {2,3,5,6,15,17,18,19}; |
| E _{1,4,17,19} = {2,3,5,6,15,16,18,20}; | E _{1,4,17,20} = {2,3,5,6,15,16,18,19}; | E _{1,4,18,20} = {2,3,5,6,15,16,18,19}; |
| E _{1,5,15,17} = {2,3,4,6,16,18,19,20}; | E _{1,5,15,18} = {2,3,4,6,16,17,19,20}; | E _{1,5,15,19} = {2,3,4,6,16,17,18,20}; |
| E _{1,5,16,18} = {2,3,4,6,15,17,19,20}; | E _{1,5,16,19} = {2,3,4,6,15,17,18,20}; | E _{1,5,16,20} = {2,3,4,6,15,17,18,19}; |
| E _{1,5,17,19} = {2,3,4,6,15,16,18,20}; | E _{1,5,17,20} = {2,3,4,6,15,16,18,19}; | E _{1,5,18,20} = {2,3,4,6,15,16,17,19}; |
| E _{1,15,17,19} = {2,3,4,5,6,16,18,19}; | E _{1,16,18,20} = {2,3,4,5,6,15,17,19}; | E _{2,4,6,15} = {1,3,5,16,17,18,19,20}; |
| E _{2,4,6,16} = {1,3,5,15,17,18,19,20}; | E _{2,4,6,17} = {1,3,5,15,16,18,19,20}; | E _{2,4,6,18} = {1,3,5,15,16,17,19,20}; |
| E _{2,4,6,19} = {1,3,5,15,16,17,18,20}; | E _{2,4,6,20} = {1,3,5,15,16,17,18,19}; | E _{2,4,15,17} = {1,3,5,6,16,18,19,20}; |
| E _{2,4,15,18} = {1,3,5,6,16,17,19,20}; | E _{2,4,15,19} = {1,3,5,6,16,17,18,20}; | E _{2,4,16,18} = {1,3,5,6,15,17,19,20}; |
| E _{2,4,16,19} = {1,3,5,6,15,17,18,20}; | E _{2,4,16,20} = {1,3,5,6,15,17,18,19}; | E _{2,4,17,19} = {1,3,5,6,15,16,18,20}; |
| E _{2,4,17,20} = {1,3,5,6,15,16,18,19}; | E _{2,4,18,20} = {1,3,5,6,15,16,17,19}; | E _{2,5,15,17} = {1,3,4,6,16,18,19,20}; |
| E _{2,5,15,18} = {1,3,4,6,16,17,19,20}; | E _{2,5,15,19} = {1,3,4,6,16,17,18,20}; | E _{2,5,16,18} = {1,3,4,6,15,17,19,20}; |
| E _{2,5,16,19} = {1,3,4,6,15,17,18,20}; | E _{2,5,16,20} = {1,3,4,6,15,17,18,19}; | E _{2,5,17,19} = {1,3,4,6,15,16,18,20}; |

- | | | |
|--|--|---|
| $E_{2,5,17,20} = \{1,3,4,6,15,16,18,19\};$ | $E_{2,5,18,20} = \{1,3,4,6,15,16,17,19\};$ | $E_{2,6,15,17} = \{1,3,4,5,16,18,19,20\}$ |
| $E_{2,6,15,18} = \{1,3,4,5,16,17,19,20\};$ | $E_{2,6,15,19} = \{1,3,4,5,16,17,18,20\};$ | $E_{2,6,16,18} = \{1,3,4,5,15,17,19,20\}$ |
| $E_{2,6,16,19} = \{1,3,4,5,15,17,18,20\};$ | $E_{2,6,16,20} = \{1,3,4,5,15,17,18,19\};$ | $E_{2,6,17,19} = \{1,3,4,5,15,16,18,20\}$ |
| $E_{2,6,17,20} = \{1,3,4,5,15,16,18,20\};$ | $E_{2,6,18,20} = \{1,3,4,5,15,16,17,19\};$ | $E_{2,15,17,19} = \{1,3,4,5,6,16,18,20\}$ |
| $E_{2,16,18,20} = \{1,3,4,5,6,15,17,19\};$ | $E_{3,5,15,17} = \{1,2,4,6,16,18,19,20\};$ | $E_{3,5,18} = \{1,2,4,6,16,17,19,20\}$ |
| $E_{3,5,15,19} = \{1,2,4,6,16,17,18,20\};$ | $E_{3,5,16,18} = \{1,2,4,6,15,17,19,20\};$ | $E_{3,5,16,19} = \{1,2,4,6,15,17,18,20\}$ |
| $E_{3,5,16,20} = \{1,2,4,6,15,17,18,19\};$ | $E_{3,5,17,19} = \{1,2,4,6,15,16,18,20\};$ | $E_{3,5,17,20} = \{1,2,4,6,15,16,18,19\}$ |
| $E_{3,5,18,20} = \{1,2,4,6,15,16,17,19\};$ | $E_{3,6,15,17} = \{1,2,4,5,16,18,19,20\};$ | $E_{3,6,15,18} = \{1,2,4,5,16,17,19,20\}$ |
| $E_{3,6,15,19} = \{1,2,4,5,16,17,18,20\};$ | $E_{3,6,16,18} = \{1,2,4,5,15,17,19,20\};$ | $E_{3,6,16,19} = \{1,2,4,5,15,17,18,20\}$ |
| $E_{3,6,16,20} = \{1,2,4,5,15,17,18,19\};$ | $E_{3,6,17,19} = \{1,2,4,5,15,16,18,20\};$ | $E_{3,6,17,20} = \{1,2,4,5,15,16,18,19\}$ |
| $E_{3,6,18,20} = \{1,2,4,5,15,16,17,19\};$ | $E_{3,15,17,19} = \{1,2,4,5,6,16,18,20\};$ | $E_{3,15,18,20} = \{1,2,4,5,6,16,17,19\}$ |
| $E_{4,6,15,17} = \{1,2,3,5,16,18,19,20\};$ | $E_{4,6,15,18} = \{1,2,3,5,16,17,19,20\}$ | |

Group F: Eliminating five voxels we obtain 36 combinations

- | | | |
|--|--|---|
| $F_{1,3,5,15,17} = \{2,4,6,16,18,19,20\};$ | $F_{1,3,5,15,18} = \{2,4,6,16,17,19,20\};$ | $F_{1,3,5,15,19} = \{2,4,6,16,17,18,20\}$ |
| $F_{1,3,5,16,18} = \{2,4,6,15,17,19,20\};$ | $F_{1,3,5,16,19} = \{2,4,6,15,17,18,20\};$ | $F_{1,3,5,16,20} = \{2,4,6,15,16,17,19\}$ |
| $F_{1,3,5,17,19} = \{2,4,6,15,16,18,20\};$ | $F_{1,3,5,17,20} = \{2,4,6,15,16,18,19\};$ | $F_{1,3,5,18,20} = \{2,4,6,15,16,17,19\}$ |
| $F_{1,3,15,17,19} = \{2,4,5,6,16,18,20\};$ | $F_{1,3,16,18,20} = \{2,4,5,6,15,17,19\};$ | $F_{1,4,15,17,19} = \{2,3,5,6,16,18,20\}$ |
| $F_{1,4,16,18,20} = \{2,3,5,6,16,18,19\};$ | $F_{1,5,15,17,19} = \{2,3,4,6,16,18,20\};$ | $F_{1,5,16,18,20} = \{2,3,4,6,15,17,19\}$ |
| $F_{2,4,6,15,17} = \{1,3,5,16,18,19,20\};$ | $F_{2,4,6,15,18} = \{1,3,5,16,17,19,20\};$ | $F_{2,4,6,15,19} = \{1,3,5,16,17,18,20\}$ |
| $F_{2,4,6,16,18} = \{1,3,5,15,17,19,20\};$ | $F_{2,4,6,16,19} = \{1,3,5,15,17,18,20\};$ | $F_{2,4,6,16,20} = \{1,3,5,15,17,18,19\}$ |
| $F_{2,4,6,17,19} = \{1,3,5,15,16,18,20\};$ | $F_{2,4,6,17,20} = \{1,3,5,15,16,18,19\};$ | $F_{2,4,6,18,20} = \{1,3,5,15,16,17,19\}$ |
| $F_{2,4,15,17,19} = \{1,3,5,6,16,18,20\};$ | $F_{2,4,16,18,20} = \{1,3,5,6,15,17,19\};$ | $F_{2,5,15,17,19} = \{1,3,4,6,16,18,20\}$ |
| $F_{2,5,15,17,20} = \{1,3,4,6,16,18,19\};$ | $F_{2,6,15,17,19} = \{1,3,4,5,16,18,20\};$ | $F_{2,6,16,18,20} = \{1,3,4,5,15,17,19\}$ |
| $F_{3,5,15,17,19} = \{1,2,4,6,16,18,20\};$ | $F_{3,5,16,18,20} = \{1,2,4,6,15,17,19\};$ | $F_{3,6,15,17,19} = \{1,2,4,5,16,18,20\}$ |
| $F_{3,6,16,18,20} = \{1,2,4,5,15,17,19\};$ | $F_{4,6,15,17,19} = \{1,2,3,5,16,18,20\};$ | $F_{4,6,16,18,20} = \{1,2,3,5,15,17,19\}$ |

Group G: Eliminating six voxels we obtain 4 combinations

- | | | |
|--|--|---|
| $G_{1,3,5,15,17,19} = \{2,4,6,16,18,20\};$ | $G_{1,3,5,16,18,20} = \{2,4,6,15,17,19\};$ | $G_{2,4,6,15,17,19} = \{1,3,5,16,18,20\}$ |
| $G_{2,4,6,16,18,20} = \{1,3,5,15,17,19\}$ | | |

Out of these 324 convex 3D hexagonal polyhedrons, 25 convex hexagonal polyhedrons (prisms).form the basis polyhedrons.

Basis Polyhedrons from E Group

- $E_{1, 4, 15,18}; E_{1, 4, 16,19}; E_{1, 4, 17,20}; E_{2, 5, 15,18}; E_{2, 5, 16,19}; E_{2, 5, 17,20}; E_{3, 6, 15,18}; E_{3, 6, 16,19}; E_{3, 6, 17,20}$

Basis Polyhedrons from F Group

- $F_{1, 4, 15,17,19}; F_{1, 4, 16,18,20}; F_{2, 5, 15,17,19}; F_{2, 5, 16,18,20}; F_{2, 5, 15,17,19}; F_{2, 5, 15,17,19}; F_{1, 3, 5,15,18}; F_{1, 3, 5,16,19}; F_{1, 3, 5,17,20}; F_{2, 4, 6,15,18}; F_{2, 4, 6,16,19}; F_{2, 4, 6,17,20}$

Basis Polyhedrons from G Group

- $G_{1,3,5,15,17,19}; G_{1,3,5,16,18,20}; G_{2,4,6,15,17,19}; G_{2,4,6,16,18,20}$

Three dimensional (3-D) convex hexagonal polyhedrons discussed above can be used as masks in the traditional processing and as structuring elements for morphological processing of three dimensional hexagonal images. Operationally 3-D morphological operations are identical to those of 2-D, with the difference that the image is traced with a 3-D structuring element where as in 2-D morphological operations, 2-D structuring element is used.

Three dimensional (3-D) structuring elements play a vital role in the processing of three dimensional volumetric images like 3-D medical images. These structuring elements are used in 3-D mathematical morphological operations such as erosion, dilation etc. Any of these 324 hexagonal prisms could be used to process a 3D hexagonal image. Usually one uses one of the 25 basis hexagonal structuring elements to do the job. For example, algorithms of edge detection and skeletonization are outlined in the following.

3.1.1 Edge detection in 3D hexagonal images using 3D algorithm

A region that appears to have a single gray-level may really contain many adjacent gray-levels. They appear to be the same because of the effect of visual quantization exercised by an observer. Segmentation is a process of partitioning a given gray-level image into disjoint regions each of which appears to an observer to have a single gray-level. So, a given hexagonal image is first segmented using a threshold-based-quantization method and after that the boundaries of the quantized regions are detected. The given hexagonal image is scanned by the 25 basis patterns discussed above. It is

sufficient to make use of these 25 neighbourhood structures (masks) for processing a 3D hexagonal image, especially in the case of edge detection because checking for the presence of these 25 neighbourhoods in a hexagonal image ensures checking of all possible 324 convex 3D hexagonal patterns in the hexagonal image. On every move, the sub image enclosed by any of these 25 neighbourhood windows is examined to see whether the D (gray-distance), which is the difference between the maximum and the minimum gray-values analogous to that sub image, is less than or equal to a threshold value T. If D is equal or less than to T, then the gray-value 0 is assigned to central voxel; otherwise the value in central cell left unchanged. This procedure is repeated till the whole 3D hexagonal image is scanned. The final outcome is that the boundaries of different regions in the given image, that appear to be uniform, are retained and their interior parts are removed thus giving us the edge detected edition of the original 3D hexagonal image. Note that this operation is carried out by pattern directed if-then rules. In the case of color images, the following method is used to detect the edges.

Repeat sliding the 25 structuring elements over the 3D hexagonal image

```
{
    Examine all voxels of a nonzero neighborhood;
    Get the maximum Red color value Red_max;
    Get the minimum Red color value Red_min;
    Then find the difference between these two values  $D_{Red}$ ;
    Get the maximum Green color value Green_max;
    Get the minimum Green color value Green_min;
    Then find the difference  $D_{Green}$ ;
    Get the maximum Blue color value Blue_max;
    Get minimum Blue color value Blue_min;
    Then find the difference  $D_{Blue}$ ;
    When all the values  $D_{Red}$ ,  $D_{Green}$  and  $D_{Blue}$  are less than or equal to a threshold value T then substitute the central pixel value with  $R = G = B = 0$ , else slide the neighbourhood windows
}
```

 until the structuring element spans whole of the image.

3.1.2 Skeletonization of 3D hexagonal images using 3D algorithm

On every move, the sub image enclosed by any of the 25 neighbourhood window is examined to see whether the D (gray-distance), which is the difference between the maximum and the minimum gray-values analogous to that sub image, is less than or equal to a threshold value T. If D is equal or less than to T, then the central voxel is retained; otherwise the value in central voxel is not considered. This procedure is repeated till the whole 3D image is scanned. The final outcome is that the boundaries of different regions in the given image, that appear to be uniform, are retained and their interior parts are removed thus giving us the edge detected edition of the original 3D image.

Note that this operation is carried out by pattern directed if-then rules. In the color images case, the following method is used to detect the edges.

Repeat sliding the 25 structuring elements over the 3D hexagonal image

```
{
    Examine all voxels of a nonzero neighbourhood;
    Get the maximum Red color value Red_max;
    Get the minimum Red color value Red_min;
    Then find the difference between these two values  $D_{Red}$ ;
    Get the maximum Green color value Green_max;
    Get the minimum Green color value Green_min;
    Then find the difference  $D_{Green}$ ;
    Get the maximum Blue color value Blue_max;
    Get minimum Blue color value Blue_min;
    Then find the difference  $D_{Blue}$ ;
    When all the values  $D_{Red}$ ,  $D_{Green}$  and  $D_{Blue}$  are less than or equal to a threshold value T then retain the central voxel value else slide the seven neighbourhood
}
```

 until the structuring elements span whole of the image.

3.2 Processing of 3D images using 3D algorithms

One can process 3D hexagonal images using 3D algorithms. Basically one would use 25 basis 3D convex hexagonal polyhedrons as scanning windows to process 3D hexagonal images. Fig. 13 shows an MR image of a human heart, its 3D edge detected and 3D skeletonized versions.

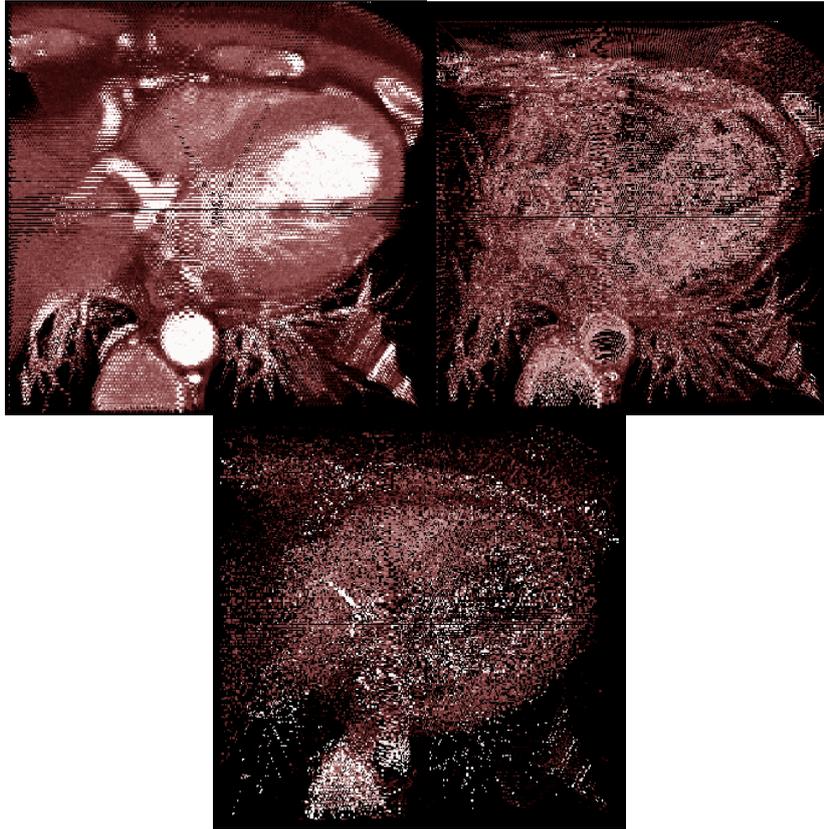


Figure 13: Sample MR image of a human heart, its 3D edge detected and 3D skeletonized versions

With reference to Fig. 11 and Fig. 13, one would observe that the results of processing 3D hexagonal images using 3D algorithms yield better results when compared to those of processing 3D hexagonal images using 2.5D algorithms.

4 Conclusions

Feature extraction from 3D images displayed over rectangular lattices has been tried by some researches but not from those displayed over hexagonal lattices. This paper provides certain preliminary techniques for visual display of 3D images over hexagonal prism lattice, a novel effort made in this direction, and extraction of certain features like 3D edges and skeleta from them. Cellular Logic Array Processing is the logico-mathematical framework in which all the relevant algorithms have been developed. A real time MR image of a human heart has been used for testing the algorithms. It has been experimentally verified that use of 3D algorithms instead of 2.5D algorithms yield better results while processing 3D hexagonal images.

ACKNOWLEDGEMENT

The authors acknowledge the extensive support offered by the managing director of Pentagram Research Centre Private Limited, Hyderabad, India for carrying out this innovative research in their

premises. The technical and programming support extended by Mr. G. Sathya, Don Jagoda Associates Inc., Melville, New York, USA is gratefully acknowledged.

REFERENCES

- [1] H. Blum, "A Transformation for Extracting New Descriptors of Models for the Perception of Speech and Visual Form", W. Wathen-Dunn, ed., US: MIT Press, Cambridge. pp 362-380. 1967
- [2] L.R.Nackman and S. M. Pizer, "Three dimensional axis transform I: Theory", IEEE Trans. On Pattern analysis and Machine Intelligence, Vol. 7, no. 2, pp. 187-202, March 1985.
- [3] Cornea, N. D., Min, P., "Curve-skeleton properties, applications, and algorithms", IEEE Transactions on Visualization and Computer Graphics, vol. 13, No. 3, pp. 530-548. May-Jun, 2007
- [4] Kalman Palagyi, Erich Sorantin, Emese Balogh, Attila Kuba, Csongor Halmai, Balazs Erdohelyi, and Klaus Hasegger, "A Sequential 3D Thinning Algorithm and Its Medical Applications," in Proc. 17th international Conf. IPMI, vol. 2082, 2001, pp. 409-415.
- [5] Kalman Palagyi and Attila Kuba, "A 3D 6-subiteration thinning algorithm for extracting medial lines," Pattern Recognition Letters, vol. 19, 1998, pp. 613-627.
- [6] Kalman Palagyi and Attila Kuba, "Directional 3D thinning using 8 subiterations," in Proc. 8th international Conf. DGCI, vol. 1568, 1999, pp. 325-336.
- [7] Ta-Chih Lee, Rangasami L. Kashyap and Chong-Nam Chu, "Building skeleton models via 3-D medial surface/axis thinning algorithms," CVGIP : Graphical Models and Image Proc., vol. 56, 1994, pp. 462-478.
- [8] Gunilla Borgefors, "Distance transformations in digital images," Computer Vision, Graphics, and Image Processing, vol. 34, 1986, pp. 344-371.
- [9] Gunilla Borgefors, "Distance transformation in arbitrary dimensions," Computer Vision, Graphics, and Image Processing, vol. 27, 1984, pp. 321-345.
- [10] Gunilla Borgefors, "On digital distance transforms in three dimensions," Computer Vision and Image Understanding, vol. 64, 1996, pp. 368-376.
- [11] Frank Y. Shih and Christopher C. Pu, "A skeletonization algorithm by maxima tracking on Euclidean distance transform", Pattern Recog., vol. 28, 1995, pp. 331-341.
- [12] Franz Aurenhammer, "Voronoi diagrams - A Survey of a fundamental geometric data structure," ACM Computing Surveys, vol. 23, 1991, pp. 345-405.
- [13] Jonathan W. Brandt and V. Ralph Algazi, "Continuous skeleton computation by Voronoi diagram," CVGIP : Image Understanding, vol. 55, 1991, pp. 329-338.
- [14] Kenneth E. Hoff III, Tim Culver, John Keyser, Ming Lin and Dinesh Manocha "Fast computation of generalized Voronoi diagrams using graphic hardware," in Proc. 26th annual Conf. Computer graphics and interactive technique, 1999, pp. 277-286.
- [15] Xiaopeng SUN, J. PAN, and Xiaopeng WEI, "3D Mesh Skeleton Extraction Using Prominent Segmentation", ComSIS Vol. 7, No. 1, Special Issue, February 2010.

- [16] Ma, W., Wu, F., Ouhyoung M., "Skeleton Extraction of 3D Objects with Radial Basis Functions", IEEE Proceedings of Shape Modeling International. Seoul, Korea, pp. 207-215, 2003
- [17] Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, Tong-Yee Lee, "Skeleton Extraction by Mesh Contraction", ACM Transactions on Graphics (SIGGRAPH 2008 issue), Vol. 27, No. 3, 44:1-44:10. (2008)
- [18] Han-Bing Yan, Shi-Min Hu, Ralph R Martin, and Yong-Liang Yang, "Shape Deformation using a Skeleton to Drive Simplex Transformations", IEEE Transaction on Visualization and Computer Graphics, vol. 14, No. 3, 693-706. (2008)
- [19] Dey T.K., Sun J., "Defining and Computing Curve-Skeletons with Medial Geodesic Function", In Proceedings of Eurographics Symp. Geometry Processing, Cagliari, Sardinia, Italy, 143-152. (2006)
- [20] Telea, A., Vilanova A., "A robust level-set algorithm for centerline extraction", in Eurographics/IEEE Symposium on Visualization, Grenoble, France, 185-194. (2003)
- [21] Zhang, X., Liu, J., Jaeger Z. Li. M., "Volume decomposition and hierarchical Skeletonization", In VRCAI '08: Proceedings of The 7th ACM International Conference on Virtual-Reality Continuum and Its Applications in Industry. Singapore; December. (2008).
- [22] C.W. Niblack, P.B. Gibbons, and D.W. Capson, "Generating skeletons and centerlines from the distance transform", CVGIP: Graphical Models and Image Processing, vol. 54, no. 5, pp. 420-437 (1992)
- [23] T. Saito and J. Toriwaki, "Reverse distance transformation and skeletons based upon the Euclidean metric for n-dimensional digital binary pictures", IEICE Trans. INF & SYST, Japan, vol. E77-D, no. 9, pp. 1005-1016 (1994)
- [24] Y. Kawase, S. Yokoi, J. Toriwaki, and T. Fukumura, "A thinning algorithm for three-dimensional binary images", Trans. on IEICE, Japan, vol. J68-D, no. 4, pp. 481-488 (1985)
- [25] La, L., Lee, S.W., Suen, C.Y., "Thinning methodologies- a comprehensive survey", IEEE Trans. PAMI, Vol. 14, pp. 869-885, 1992.
- [26] Mohd sherfuddin khan et.al "Volumetric Features Extraction using 2.5D and 3D Algorithms" international journal of communication networks and distributed systems inderscience publishers issue No.3-4 vol 18 2017 pp:248-263
- [27] Mohd sherfuddin khan et.al " skeletonization of 3D images using 2.5D and 3D Algorithms" 1st International Conference on Next Generation Computing Technologies (NGCT), 2015, At Deharadun, INDIA sept 4-5, pp 971-975
- [28] M. Golay, "Hexagonal parallel pattern transformation", IEEE Transactions on computers, Vol. 18, No. 8, pp. 733-740, 1969. R.M. Mersereau, "The process of hexagonally sampled Two - dimensional signals", Proceedings of the IEEE, Vol 67, pp. 930-949, 1979.
- [29] B. Kamgar-Parsi and W.A. Sander, III, "Quantization error in spatial sampling: comparison between square and hexagonal pixels", Computer Vision and Pattern Recognition, Proceedings CVPR '89' pp. 604 - 611, 1989.
- [30] B. Kamgar-Parsi, "Quantization error in hexagonal sensory configurations", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 6, pp. 665-671, 1992.

- [31] J.P.Mylopoulos and T.Pavlidis, "On the topological properties of quantized spaces, I. the notion of dimension", *Journal of the ACM (JACM)*, Vol.18, No.2, pp. 239-246, 1971.
- [32] J.P.Mylopoulos and T.Pavlidis, "On the topological properties of quantized spaces, II. connectivity and order of connectivity", *Journal of the ACM (JACM)*, Vol 18, No 2, pp.247- 254,1971.
- [33] R.Vitulli, et al., "Aliasing effects mitigation by optimized sampling grids and impact on image acquisition chains", *Geoscience and Remote Sensing Symposium, IGARSS '02*, pp. 979-981, 2002.
- [34] Eric Anterrieu, Philippe Waldteufel and André Lannes, "Apodization Functions for 2-D Hexagonally Sampled Synthetic Aperture Imaging Radiometers", *IEEE Transactions On Geoscience And Remote Sensing*, Vol. 40, No. 12, pp. 2531-2542, December 2002.
- [35] Lee Middleton and Jayanthi Sivaswamy , "Hexagonal Image Processing – A Practical approach", Springer-Verlag London Limited, 2005.
- [36] Senthil Periaswamy, "Detection of Micro calcifications in Mammograms Using Hexagonal Wavelets", Thesis work, University of South Carolina, 1996.
- [37] L. Condat, D. Van De Ville, and T. Blu, "Hexagonal versus orthogonal lattices: A new comparison using approximation theory," in *Proceedings of IEEE, ICIP*, Vol. 3, Sept. 2005.
- [38] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, Vol. 16, No. 6, pp. 22–38, Nov. 1999.
- [39] D. Van De Ville, W. Philips, and I. Lemahieu, "Least-squares spline resampling to a hexagonal lattice," *Signal Processing: Image Communication*, Vol. 17, No. 5, pp. 393–408, May 2002.
- [40] L. Condat and D. Van De Ville , "Quasi-interpolating spline models for hexagonally sampled data," *IEEE Transactions on Image Processing*, Vol. 16, No. 5, May 2007, pp.1195-1206.
- [41] P. Th´evenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Transactions on Medical Imaging* , Vol. 19, No.7, pp. 739–758, July 2000.
- [42] D. E. Dudgeon and M. E. Mersereau, *Multidimensional Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [43] D. Van De Ville, T. Blu, M. Unser, W. Philips and I. Lemahieu, "Hex-spline: A novel family for hexagonal lattices," *IEEE Transactions on Image Processing*, Vol. 13, No. 6, June 2004, pp.758–772.
- [44] Laurent Condat and Dimitri Van De Ville, "Three-directional box-splines: Characterization and efficient evaluation," *IEEE Signal Processing Letters*, Vol. 13, No. 7, July 2006 , pp. 417–420.
- [45] C. de Boor, K. Hollig, and S. Riemenschneider, "Box Splines": Springer-Verlag, 1993, Applied Mathematical Sciences, Vol. 98.
- [46] H. Prautzsch and W. Boehm, "Box splines," in *Handbook of Computer Aided Geometric Design*. Berlin: Springer, 2001.
- [47] Lee, T.S, "Image Representation Using 2D Gabor Wavelets, *IEEE Transactions on Pattern Analysis and Machine Intelligence*," Vol. 18, Issue 10, 1996, pp. 951-979.

- [48] Xu, Y., Zhang, X ,“ Gabor Filter bank and its Application in the Fingerprint Texture Analysis,” PDCAT 2005, pp. 829-831.
- [49] Huang, L., Shimizu, A., Kobatake, H , “Classification-Based Face Detection Using Gabor Filter Features,” *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 397-402.
- [50] Jain, A.K., Prabhakar, S., Hong, L., Pankanti, S,“Filter bank-Based Fingerprint Matching,” *IEEE Transactions on Image Processing*, Vol. 9, Issue 5, 2000, pp. 846-859.
- [51] Jones, J.P., Palmer, L.A, “An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex, “ *Journal of Neurophysiology*, Vol. 58, Issue 6, 1987, pp. 1233-1258.
- [52] Veni. S, K.A. Narayanankutty, Satyapal Reddy.L, Vidya.P,“ Wavelet based Edge detection and Multiresolution analysis on Hexagonal Sampled Grid Using Interpolation,” proceedings ICSIP - 2009 , August 12-14 2009 , Excel publications, ISBN:978-93-80043-26-5,pp.186-200.
- [53] Veni.S, K.A.Narayanankutty , Vidya.P ,“ Performance analysis of Edge Detection methods on hexagonal sampling Grid,” *International Journal of Electronics Engineering and Research (IJEER)*, Research India publications Vol.1, No. 4 (2009) ,pp. 313–328.