A New Simulation of Spiral Architecture

Xiangjian He, Tom Hintz, Qiang Wu, Huaqing Wang and Wenjing Jia
Department of Computer Systems
Faculty of Information Technology
University of Technology, Sydney
{sean,hintz,wuq,huwang,wejia}@it.uts.edu.au

Abstract - *Spiral Architecture is a relatively new and powerful approach to machine vision system. The geometrical arrangement of pixels on Spiral architecture can be described in terms of a hexagonal grid. However, all the existing hardware for capturing image and for displaying image are produced based on rectangular architecture. It has become a serious problem affecting the advanced research on Spiral Architecture. In this paper, a new approach to mimicking Spiral Architecture is presented. This mimic Spiral Architecture almost retains image resolution and does not introduce distortion. Furthermore, images can be smoothly and easily transferred between the traditional square structure and this new hexagonal structure. In this paper, we also perform a fast way to locate hexagonal pixels. Another contribution in this paper is a novel construction of hexagonal pixels that are four times as big as the virtual hexagonal pixels. This construction of larger hexagonal pixels does not change the axes of symmetry, and does not create any spaces or overlaps between hexagons.*

Keywords: Hexagonal structure, Spiral Architecture.

1 Introduction

The advantages of using a hexagonal grid to represent digit images have been investigated for more than thirty years. The importance of the hexagonal representation is that it possesses special computational features that are pertinent to the vision process. Its computational power for intelligent vision pushes forward the image processing field. Dozens of reports describing the advantages of using such a grid type have been found in the literature. The hexagonal image structure has features of higher degree of circular symmetry, uniform connectivity, greater angular resolution, and a reduced need of storage and computation in image processing operations.

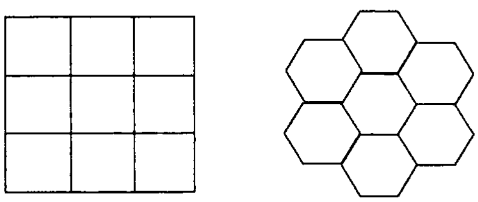
In spite of its numerous advantages, hexagonal grid has so far not yet been widely used in computer vision and graphics field. The main problem that limits the use of hexagonal image structure is believed due to lack of hardware for capturing and displaying hexagonal-based images. In the past years, there have been various attempts to simulate a hexagonal grid on a regular rectangular grid device. The simulation schemes include those using rectangular pixels [1, 2], pseudo hexagonal pixels [3], mimic hexagonal pixels [4] and virtual hexagonal pixels [5]. Although none of these simulation schemes can represent the hexagonal structure without depressing the advantages that a real hexagonal structure possesses, the use of these techniques provides us a practical tool for image processing on a hexagonal structure and makes us possible to carry out research based on a hexagonal structure using existing computer vision and graphics systems.

The arrangement of a hexagonal grid is different from a rectangular grid as seen in Figure 1. On a hexagonal image structure, each pixel has only six neighboring pixels which have the same distance to the central pixel.

In this paper, we will construct a hexagonal structure that is converted from the traditional square structure easily and quickly. Based on the structure, we will present a easy and fast way to compute the location of each hexagonal pixel.

We will also develop a method that is used to increase the size of the constructed hexagonal pixels to four times bigger. Increasing the pixel size does not change the three axes of symmetry in vertical and two diagonal directions respectively. Furthermore, enlargement of the pixel size does not create any spaces (or gaps) or overlaps between hexagons. In the previous research on fractal compression on hexagonal structure [6], we could only increase the size of a hexagonal region to be seven times bigger in order to keep shape of region similar. This has led to imperfect compression results. This paper provides a means that can easily increase the size of a region to be four times bigger while keeping the shape of the region exactly the same. Based on this, it is expected that the results of fractal image compression on hexagonal structure can be improved. How image compression can be improved is beyond this paper.

The rest of this paper is organized as follows. In Section 2, we review the related works on simulation of hexagonal structure. We briefly review a special hexagonal structure, called Spiral Architecture in Section 3. In Section 4, a simulation of hexagonal structure is performed, followed by the methods to determine the pixel locations and for increasing the size of hexagonal pixels. We conclude in Section 5.



(a) Rectangular *(b) Hexagonal*

Figure 1. Vision unit in two different image architectures

2 Related Work

How to simulate hexagonally sampled images on common square display equipments has become a serious problem that affects the advanced research on hexagonal architecture in the field of computer vision and graphics.

There have been several ways to simulate a hexagonal grid on a regular rectangular grid. Some of the related works are summarized as follows. More complete review on simulation of hexagonal structure can be found in [7].

2.1 Mimic hexagonal pixels using square pixels

Wuthrich et al. [3] proposed a pseudo hexagonal pixel (see Fig. 2) in order to evaluate the visual effect of hexagonal pixel and square pixel. A hexagonal pixel, called a hyperpel, is simulated using a set of many square pixels and the simulated square grid had to be adapted in order to make its density comparable with the hexagonal grid. This results in a great loss of image resolution and an inexact simulation of the square grid.

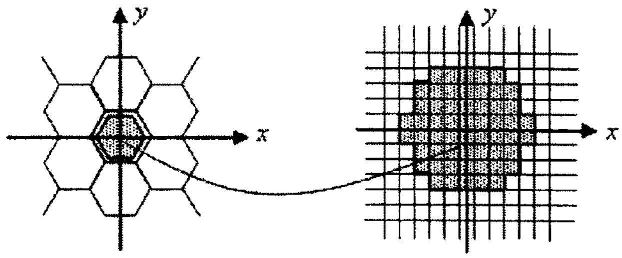


Figure 2. Simulated hyper pixel (right)

He [4] proposed a mimic hexagonal structure, called *mimic Spiral Architecture*, where one hexagonal pixel consists of four traditional square pixels and its grey level value is the average of the involved four pixels (see Figure 3). This mimic scheme preserves the important property of hexagonal architecture that each pixel has exactly six surrounding neighbours. However, because the grey-level value of the mimic hexagonal pixel is taken from the average of the four corresponding square pixels, this mimic scheme introduces loss of resolution. In addition, we know that according to hexagonal structure theory the distance between each of the six surrounding pixels and the central pixel is the same. However, this property is lost in the mimic Spiral Architecture.

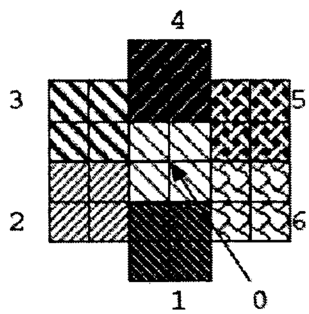


Figure 3. A cluster of 7 mimic hexagons

2.2 Virtual hexagonal structure

Wu et al. [5] constructed a virtual hexagonal structure which is an important milestone for the theoretical research and the practical application exploration of this architecture. Using virtual Spiral Architecture, images on rectangular structure can be smoothly converted to Spiral Architecture. The

Virtual Spiral Architecture exists only during the procedure of image processing. It builds up virtual hexagonal pixels in the memory space of a computer. Then, processing algorithms is implemented based on the virtual spiral space. Finally, the resulted data can be mapped back to rectangular architecture for display (see Figure 4). Unlike the previously proposed mimicking methods, this mimicking operation almost does not introduce distortion or reduce image resolution, which is the most remarkable advantage over other mimicking methods, while keeping the isotropic property of the hexagonal architecture. But one of the disadvantages of using this approach is that the computation cost is high when converting between the square based images and hexagon based images because of the complex computation in determining the locations (or the areas) of hexagonal pixels.

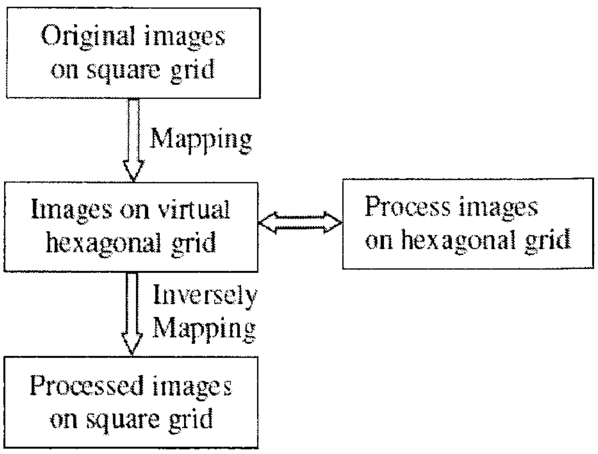


Figure 4. Image processing on virtual Spiral Architecture

3 Spiral Architecture

Obviously, no matter which kind of simulation scheme is applied, the hexagonal pixels cannot be labeled in row and column order as in the traditional rectangular structure. In order to properly address and store hexagonal images data, Sheridan [8] proposed a one-dimensional addressing scheme for a hexagonal structure, together with the definitions of two operations, Spiral Addition and Spiral Multiplication. This hexagonal structure is called the Spiral Architecture (SA) (see Figure 5). The Spiral Architecture is inspired from anatomical consideration of the primate's vision system.

In the Spiral Architecture, Spiral Addition and Spiral Multiplication correspond to image translation and rotation respectively. The result of an addition or a multiplication is a spiral address [8], and it can be computed based on the spiral addition between the seven addresses from 0 to 6 [9].

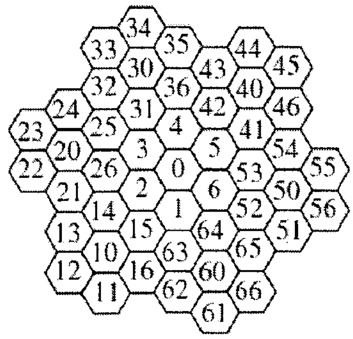


Figure 5. Spiral Architecture and Spiral addressing

For the whole image, following the spiral rotation direction, as shown in Figure 6, one can find out the location of any hexagonal pixel with a given spiral address starting from the central pixel of address 0. From Figure 6, it is easy to see that finding neighbouring pixels plays a very important role to locate a pixel and hence is critical in the process of the two operations defined on the SA. The location of the pixel with a given spiral address

$$a_n a_{n-1} \cdots a_1,\ (a_i = 0,1,2,\cdots,6 \text{ for } i = 1,2,\cdots,n.)$$

can be found from the locations of

$$a_i \times 10^{i-1} \text{ for } i = 1,2,\cdots,n.$$

For example, to find the location of the pixel with spiral address 243, we need only know the locations of the pixels with spiral addresses 200, 40 and 3.

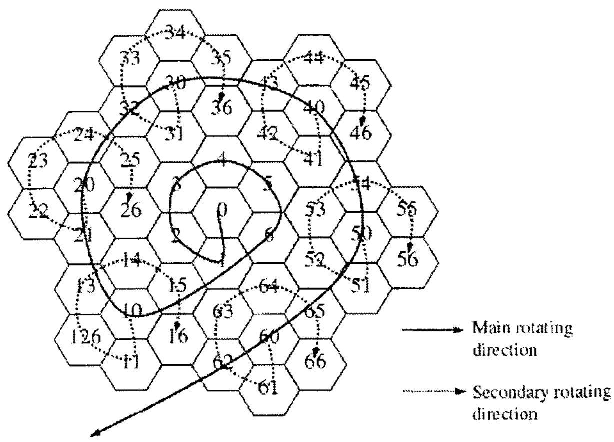


Figure 6. Spiral rotating direction

4 Simulation of Spiral Architecture

After the review of various mimicking methods for simulation of hexagonal structure in Section 2, a new mimic Spiral Architecture that modifies the existing Virtual Spiral Architecture [5] is presented in this section.

4.1 Construction of hexagonal pixels

The idea as shown in [3] for the construction of hyperpels is adopted in this section to construct hexagonal pixels. We first separate each square pixel into 7×7 smaller pixels called sub-pixels. The light intensity for each of these sub-pixels is the same as that of the pixel from which the sub-pixels are separated. Each virtual hexagonal pixel is formed by 56 sub-pixels arranged as shown in Figure 7. The light intensity of each constructed hexagonal pixel can be easily computed as the average of the intensities of the 56 sub-pixels forming the hexagonal pixel. Note that the size of each constructed pixel is

$$\frac{56-49}{56} = 12.5\%$$

bigger than each square pixel. Hence, the number of hexagonal pixels is 12.5% less than the number of square pixels to cover the same image. From the observation result obtained in [10], it is claimed that 13.4% fewer sampling points (or pixels) are required with a hexagonal structure to maintain equal amount of image information (or the same image resolution) with the traditional square structure. Because 12.5% is less than 13.4%, the image represented using the hexagonal pixels constructed in the way above will not lose image resolution.

		X	X	X	X	X		
	X	X	X	X	X	X	X	
	X	X	X	X	X	X	X	
X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X
	X	X	X	X	X	X	X	
	X	X	X	X	X	X	X	
		X	X	X	X	X		

Figure 7. The structure of a single hexagonal pixel

Figure 8 shows a collection of seven hexagonal pixels constructed with spiral addresses from 0 to 6. From Figure 8, it is easy to see that the hexagonal pixels constructed in this way tile the whole plane without and spaces and overlaps. From Figure 8, it can be easily computed that the distance from pixel 0 to pixel 1 or pixel 4 is 8. The distance from pixel 0 to pixel 2, pixel 3, pixel 5 or pixel 6 is

$$\sqrt{7^2+4^2} = 8.06.$$

which is close to 8. Hence, the feature of equal distance is almost retained and hence this construction hardly introduces image distortion.

									4	4	4	4	4									
								4	4	4	4	4	4	4								
								4	4	4	4	4	4	4								
							4	4	4	4	4	4	4	4	4							
		3	3	3	3	3	4	4	4	4	4	4	4	4	4	5	5	5	5	5		
	3	3	3	3	3	3	3	4	4	4	4	4	4	4	5	5	5	5	5	5	5	
	3	3	3	3	3	3	3	4	4	4	4	4	4	4	5	5	5	5	5	5	5	
3	3	3	3	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	3	3	0	0	0	0	0	5	5	5	5	5	5	5	5	5
	3	3	3	3	3	3	3	0	0	0	0	0	0	0	5	5	5	5	5	5	5	
	3	3	3	3	3	3	3	0	0	0	0	0	0	0	5	5	5	5	5	5	5	
		3	3	3	3	3	0	0	0	0	0	0	0	0	0	5	5	5	5	5		
		2	2	2	2	2	0	0	0	0	0	0	0	0	0	6	6	6	6	6		
	2	2	2	2	2	2	2	0	0	0	0	0	0	0	6	6	6	6	6	6	6	
	2	2	2	2	2	2	2	0	0	0	0	0	0	0	6	6	6	6	6	6	6	
2	2	2	2	2	2	2	2	2	0	0	0	0	0	6	6	6	6	6	6	6	6	6
2	2	2	2	2	2	2	2	2	1	1	1	1	1	6	6	6	6	6	6	6	6	6
	2	2	2	2	2	2	2	1	1	1	1	1	1	1	6	6	6	6	6	6	6	
	2	2	2	2	2	2	2	1	1	1	1	1	1	1	6	6	6	6	6	6	6	
		2	2	2	2	2	1	1	1	1	1	1	1	1	1	6	6	6	6	6		
							1	1	1	1	1	1	1	1	1							
								1	1	1	1	1	1	1								
								1	1	1	1	1	1	1								
									1	1	1	1	1									

Figure 8. A cluster of seven hexagonal pixels

4.2 Locating hexagonal pixels

To locate a pixel, we only need to derive the way to locate the pixel with spiral address in the form of

$$a \times 10^i, i = 1,2,\cdots, a = 1,2,\cdots,6.$$

Let us use vector [0, 0] to denote the location of the hexagonal with spiral address 0, and vector [j, k] (j, k are integers) to denote the location of a pixel that is obtained by moving from [0, 0] towards right (or left if j is negative) for $|j|$ sub-pixels and down (or up if k is negative) for $|k|$ sub-pixels. If we also use $L(a)$ to denote the location of the hexagonal pixel with spiral address a, then we have $L(0)$ = [0, 0]. From Figure 8, it is easy to see that

$$L(1)=[0,\ \ 8], L(2)=[-7,\ \ 4], L(3)=[-7,\ \ -4],$$
$$L(4)=[0,\ \ -8], L(5)=[7,\ \ -4], L(6)=[7,\ \ 4].$$

Note that location of hexagonal pixel with address 10 is obtained by moving from pixel 1 in the direction from pixel 6 towards pixel 1 for two pixels distance [8] (see Figure 5). Similarly, we can determine the locations of pixels 20, 30, 40, 50 and 60. From Figure 8, the locations of these pixels can be computed as follows.

$$L(10)=L(1)+2L(2), L(20)=L(2)+2L(3),$$
$$L(30)=L(3)+2L(4), L(40)=L(4)+2L(5),$$
$$L(50)=L(5)+2L(6), L(60)=L(6)+2L(1).$$

Following this track, it is easy to derive that

$$L(a\times 10^{i})=L(a\times 10^{i-1})+2L((a+1)\times 10^{i-1})$$
$$L(6\times 10^{i})=L(6\times 10^{i-1})+2L(10^{i-1})$$

for $i=1,2,\cdots, a=1,2,\cdots,5$.

The location of the pixel with a given spiral address

$$a_n a_{n-1}\cdots a_1,\ (a_i=0,1,2,\cdots,6 \text{ for } i=1,2,\cdots,n.)$$

can be computed by

$$L(a_n a_{n-1}\cdots a_1)=\sum_{i=1}^{n} L(a_i\times 10^{i-1}).$$

For example,

$$\begin{aligned}
&L(243)\\
&=L(200)+L(40)+L(3)\\
&=L(20)+2L(30)+[L(4)+2L(5)]+L(3)\\
&=L(2)+2L(3)+2[L(3)+2L(4)]\\
&+L(4)+2L(5)+L(3)\\
&=L(2)+5L(3)+5L(4)+2L(5)\\
&=[-7,\ \ 4]+5[-7,\ \ -4]+5[0,\ \ -8]+2[7,\ \ -4]\\
&=[-28,\ \ -64].
\end{aligned}$$

Therefore, our new simulation of Spiral Architecture avoids the complex computation of pixel regions as shown in our previous paper [5] and we no longer need to create a large table (saved in the PC memory) to record the locations of hexagonal pixels. This greatly increase the speed of image processing based on SA.

4.3 Enlarge hexagonal pixels

To form hexagonal pixels which are four times as big as the hexagonal pixels constructed above, we take 24 sub-pixels from each of neighbouring pixels of a reference pixel. As illustrated in Figure 9, we take 28 sub-pixels (marked red) from pixels labeled 1 up to 6. These pixels together with the 56 sub-pixels labeled 0 (of the reference hexagonal pixel 0) form a larger pixel consisting of $4\times 56=224$ sub-pixels.

									4	4	4	4	4									
								4	4	4	4	4	4	4								
								4	4	4	4	4	4	4								
							4	4	4	4	4	4	4	4	4							
		3	3	3	3	3	4	4	4	4	4	4	4	4	4	5	5	5	5	5		
	3	3	3	3	3	3	3	4	4	4	4	4	4	4	5	5	5	5	5	5	5	
	3	3	3	3	3	3	3	4	4	4	4	4	4	4	5	5	5	5	5	5	5	
3	3	3	3	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	3	3	0	0	0	0	0	5	5	5	5	5	5	5	5	5
	3	3	3	3	3	3	3	0	0	0	0	0	0	0	5	5	5	5	5	5	5	
	3	3	3	3	3	3	3	0	0	0	0	0	0	0	5	5	5	5	5	5	5	
		3	3	3	3	3	0	0	0	0	0	0	0	0	0	5	5	5	5	5		
		2	2	2	2	2	0	0	0	0	0	0	0	0	0	6	6	6	6	6		
	2	2	2	2	2	2	2	0	0	0	0	0	0	0	6	6	6	6	6	6	6	
	2	2	2	2	2	2	2	0	0	0	0	0	0	0	6	6	6	6	6	6	6	
2	2	2	2	2	2	2	2	2	0	0	0	0	0	6	6	6	6	6	6	6	6	6
2	2	2	2	2	2	2	2	2	1	1	1	1	1	6	6	6	6	6	6	6	6	6
	2	2	2	2	2	2	2	1	1	1	1	1	1	1	6	6	6	6	6	6	6	
	2	2	2	2	2	2	2	1	1	1	1	1	1	1	6	6	6	6	6	6	6	
		2	2	2	2	2	1	1	1	1	1	1	1	1	1	6	6	6	6	6		
							1	1	1	1	1	1	1	1	1							
								1	1	1	1	1	1	1								
								1	1	1	1	1	1	1								
									1	1	1	1	1									

Figure 9. A larger size hexagonal pixel obtained from 6 pixels of smaller size

The distance between the neighbouring pixels of this larger size in the vertical direction is 16. The distance between the neighbouring pixels of this larger size in the two diagonal directions is

$$\sqrt{14^2+8^2}=16.12$$

Furthermore, each larger pixel has six neighbouring pixels of the same size in the vertical and two diagonal directions respectively like the structure consisting of smaller hexagonal pixels.

5 Conclusions and Discussion

In this paper, we have developed a novel method to construct or mimic the Spiral Architecture. This constructed hexagonal structure does not change the image resolution and introduce image distortion. It retains the advantages of the real hexagonal system such as higher degree of symmetry, uniformly connected and closed-packed form. This structure together with the light intensities cannot be displayed and it exists only in the computer memory during the procedure of image processing. Image processing based on a hexagonal structure can be implemented using this structure. As there are simple non-overlapping mappings between the sub-pixels and the square pixels, and the mappings between the sub-pixels and the hexagonal pixels, the results of image processing on the hexagonal structure can be easily mapped back to the square structure for to display.

Unlike the Virtual Spiral Architecture shown in [5], the construction of this new mimic structure does not require complex computation for determining the regions of hexagonal pixels, and does not request build a large table stored in the computer memory to record the pixel locations. The location of each pixel can be easily and fast determined and computed using mathematical formulae. The computation speed for image processing based on this newly developed structure can hence be greatly improved.

In order to show its potential application to the area of image compression and other areas, we have also presented a method to construct hexagonal pixels that are four times bigger than the original hexagonal pixels. The structure consisting of the larger pixels does not change the orientation of the hexagonal structure, i.e., the three symmetry axes of the hexagonal structure keep unchanged. The way to enlarge each pixel provides a means to sketch regions of lager size with the same shape. For example, for fractal image compression [6], we can now easily construct domain blocks that are four times bigger than range blocks, and the domain and range blocks look exactly the same except their sizes.

In order to show the applications of our work, our next step is to refine our existing algorithms based on the Spiral Architecture for edge detection, image compression and others. A great improvement both in accuracy and speed is expected to achieve.

6 References

[1] B.K.P. Horn, *Robot Vision*. 1986: MIT Press, Cambridge, MA & McGraw-Hill, New York, NY.

[2] R. Staunton, *The design of hexagonal sampling structures for image digitization and their use with local operators*. Image and Vision Computing, 1989. **7**(3): p. 162-166.

[3] C.A. Wuthrich and P. Stucki, *An algorithmic comparison between square- and hexagonal-based grids*. CVGIP: Graphical Models and Image Processing, 1991. **53**(4): p. 324-339.

[4] X. He,, *2-D Object Recognition with Spiral Architecture*. 1999, PhD Thesis, University of Technology, Sydney.

[5] Q. Wu,, X. He, and T. Hintz, *Virtual Spiral Architecture*. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, 2004. **1**: p. 399-405.

[6] Huaqing Wang, Meiqing Wang, Tom Hintz, Xiangjian He and Qiang Wu, *Fractal image compression on a pseudo Spiral Architecture*, Australian Computer Science Communications, 27(2005), pp.201-207.

[7] Xiangjian He and Wenjing Jia, *Hexagonal structure for intelligent vision*, Proceedings of International Conference on Information and Communication Technologies (IEEE), Karachi, September 2005, pp.52-64.

[8] P. Sheridan, T. Hintz, and D. Alexander, "Pseudo-invariant Image Transformations on a Hexagonal Lattice," *Image and Vision Computing*, vol. 18, pp. 907-917, 2000.

[9] Qiang Wu, *Advanced Image Processing Research on Spiral Architecture*, 2003, PhD Thesis, University of Technology, Sydney.

[10] R.M. Mersereau, *The processing of Hexagonally Sampled Two-Dimensional Signals*. Proceedings of the IEEE, 1979. **67**: p. 930-949.