

© 2006 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Basic Transformations on Virtual Hexagonal Structure

Xiangjian He<sup>1,2</sup>, Wenjing Jia<sup>1</sup>, Qiang Wu<sup>1</sup>, Namho Hur<sup>2</sup>, Tom Hintz<sup>1</sup>, Huaqing Wang<sup>1</sup>, and Jinwoong Kim<sup>2</sup>

<sup>1</sup>Department of Computer Systems  
Faculty of Information Technology  
University of Technology, Sydney  
{sean,wejia,wuq,hintz,huwang}@it.uts.edu.au

<sup>2</sup>Broadcasting System Research Group  
Digital Broadcasting Research Division  
Electronics Telecommunications Research Institute  
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Korea  
{namho,jwkim}@etri.re.kr

## Abstract

*Hexagonal structure is different from the traditional square structure for image representation. The geometrical arrangement of pixels on hexagonal structure can be described in terms of a hexagonal grid. Hexagonal structure provides an easy way for image translation and rotation transformations. However, all the existing hardware for capturing image and for displaying image are produced based on square architecture. It has become a serious problem affecting the advanced research based on hexagonal structure. In this paper, we introduce a new virtual hexagonal structure. Based on this virtual structure, a more flexible and powerful image translation and rotation are performed. The virtual hexagonal structure retains image resolution during the process of image transformations, and does not introduce distortion. Furthermore, images can be smoothly and easily transferred between the traditional square structure and the hexagonal structure.*

*Keywords-- Hexagonal structure, image translation, image rotation, 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 hexagonal grids 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 [1-6]. Wuthrich et al. [3] proposed a pseudo hexagonal pixel 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. 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 four pixels involved. This mimic scheme preserves the important property of hexagonal architecture that each pixel has exactly six surrounding neighbours. However, this mimic scheme also 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.

Wu et al. [5] constructed a virtual hexagonal structure which is an important milestone for the theoretical research and the practical application using this architecture. Images on rectangular structure can be

smoothly converted to the virtual hexagonal structure, and vice versa. The virtual hexagonal pixels do not physically exist but are recorded in the memory space of a computer only during image process. Unlike the previously proposed mimicking methods, this mimicking operation almost does not introduce distortion or reduce image resolution, 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.

In order to increase the computation speed for image processing based on hexagonal structures and to reduce the computation complexity, He et al. [7] have proposed a new virtual hexagonal structure. In this new structure, the locations of virtual hexagons can be easily located through simple computations. It avoids the necessity of building a large table to record the information of pixel locations. Like the virtual structure shown in [5], this new virtual structure hardly changes the image resolution and almost does not introduce image distortion either.

In this paper, we will perform in detail the computations to easily and quickly convert between the square pixels and hexagonal pixels on the virtual structure proposed in [7]. We will also present basic image transformations including translation and rotation on this new virtual structure. These transformations are critical for image processing on hexagonal structure.

The rest of this paper is organized as follows. In Section 2, we briefly review a hexagonal image structure with a numbering scheme. In Section 3, the algorithms to convert between the traditional square structure and a virtual hexagonal structure are preformed. Two basic image transformations on the virtual structure are presented in Section 4. Experimental results are demonstrated in Section 5. We conclude in Section 6.

## 2. Spiral Architecture

Apparently, 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, called Spiral Architecture, as shown in Figure 1. The Spiral Architecture is inspired from anatomical consideration of the primate's vision system.

For the whole image, following the spiral-like curve, as shown in Figure 1, 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 1, it is easy to see that the location of the pixel with a given spiral address

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

can be found from the locations of

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

For example, to find the location of the pixel with spiral address 43, we need only know the locations of the pixels with spiral addresses 40 and 3.

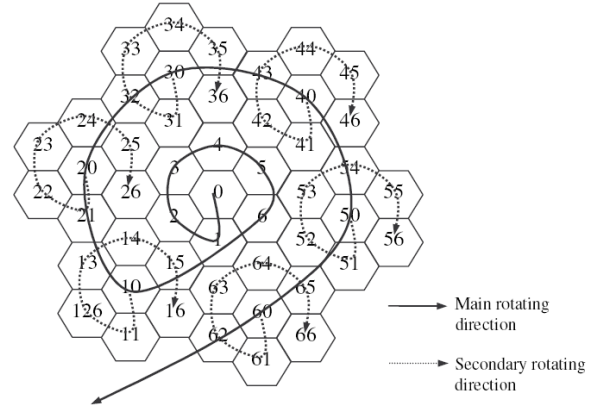


Figure 1. Spiral Architecture with spiral addressing

## 3. Virtual hexagonal structure

In this section, we introduce the construction of this new virtual hexagonal structure and perform the algorithms that convert between the pixels on the traditional square structure and the virtual hexagonal structure.

### 3.1. Construction of hexagonal pixels

To construct hexagonal pixels, each square pixel is first separated into  $7 \times 7$  smaller pixels, called sub-pixels. To be simple, the light intensity for each of these sub-pixels is set to be 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 2. To be simple, the light intensity of each constructed hexagonal pixel is computed as the average of the intensities of the 56 sub-pixels forming the hexagonal pixel.

		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 2. The structure of a single hexagonal pixel



### 3.3. Conversion between image structures

Let us assume that original images are represented on a square structure arranged as  $2M$  rows and  $2N$  columns, where  $M$  and  $N$  are two positive integers.  $M=0$  corresponds to the first row and  $N=0$  corresponds to the first column. Let the centre of the virtual hexagonal structure be located at the middle of rows  $M$  and  $M+1$ , and at column  $N$ .

Note that there are  $14M$  rows and  $14N$  columns in the (virtual square) structure consisting of virtual sub-pixels obtained from the original square pixels.

Let us construct the first hexagonal pixel using the 56 sub-pixels with centre located in the middle of rows  $7M$  and  $7M+1$  and the column  $7N$  of the virtual square structure. After the 56 sub-pixels for the first hexagonal pixel are allocated, the corresponding sub-pixels for its six neighbouring hexagonal pixels can be easily allocated from (1). Then all sub-pixels for all hexagonal pixels can be assigned from (4). For our translation and rotation algorithms in this paper, the assignment of sub-pixels to corresponding hexagonal pixels is not required. We do not need to compute the intensities for the virtual hexagonal pixels either.

## 4. Image Transformations

In this section, we perform two basic transformations on the virtual hexagonal structure.

### 4.1. Translation

We perform image translation in any direction based on movement of hexagonal pixels. Let us assume that we wish to translate an image such that the central hexagonal pixel  $O$  is moved to the hexagonal pixel with spiral address  $A$  represented in the form of (3). Then, from Equation (4), we can compute the numbers of rows and columns in the virtual structure to move, and the direction of the movement in order to move the central pixel from  $O$  to  $A$ . If, for example, the computed numbers of rows and columns for the movement are  $R$  and  $C$ , the translation transformation can be performed by moving all sub-pixels  $|R|$  rows downwards (upwards if  $R$  is negative) and  $|C|$  columns rightwards (leftwards if  $C$  is negative).

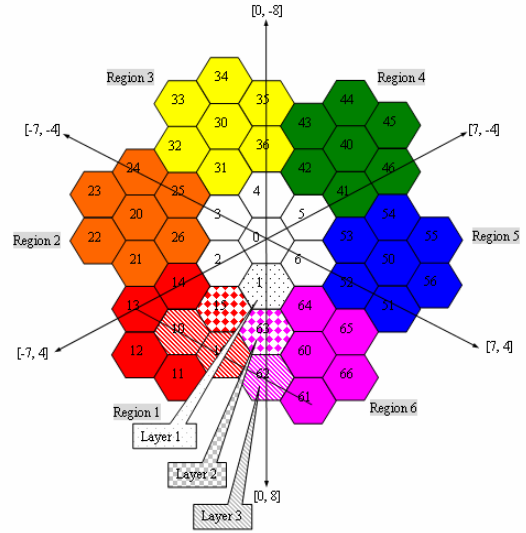
### 4.2. Rotation

In this paper, we consider the rotation with an angle of  $60^\circ$  only. The rotations with angles  $120^\circ$ ,  $180^\circ$ ,  $240^\circ$  and  $300^\circ$  can be achieved in the same way.

As shown in Figure 4, let us draw 6 lines from the central hexagonal pixel along the directions to hexagonal pixels with spiral addresses 1, 2, 3, 4, 5 and 6 respectively. In our virtual hexagonal structure, according to the results shown in (1), the 6 directions can be described by the 6 vectors in (1). Again, as shown in Figure 1, hexagonal pixels in any region, formed by two adjacent lines of the 6 lines obtained above, are arranged

as multiple layers from the central pixel onwards. The first layer of each region is formed by a single pixel that is closest to the central pixel (but not the central pixel). The second layer consists of two pixels next to the first layer. Every upper layer has one more pixel included. All pixels at the same layer are on the same line.

For example, the region formed by  $L(1)$  and  $L(2)$  has pixel with address 1 at its first layer, and pixels with addresses 63 and 15 at its second layer and so forth.



**Figure 4. Regions, layers and positions of hexagonal pixels**

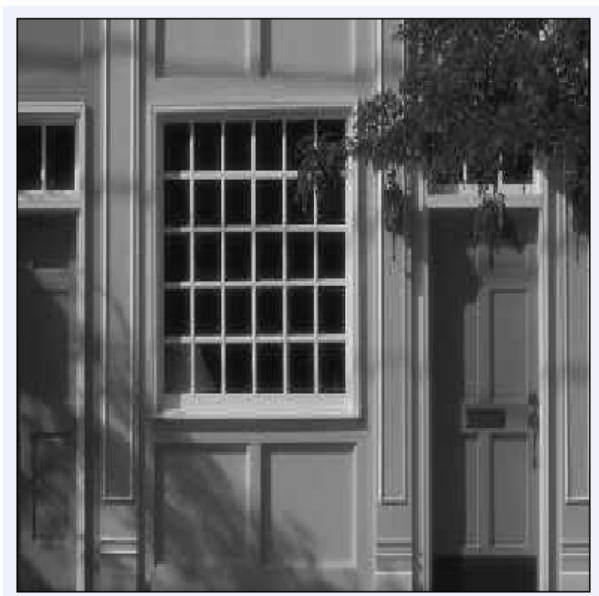
Now, in order to perform image rotation, for any sub-pixel location, we need to know the region it belongs to. We need also know which layer it is on, its position at the layer, and its relative location to the center of the hexagonal pixel that it lies on. After the region, layer, position and relative location of any given sub-pixel are computed. The algorithm for rotation can be followed. The idea is to replace any sub-pixel in a region  $r$  at layer  $s$  located at position  $t$  with relative location  $l$  by the sub-pixel in the previous region  $r-1$  at the same layer  $s$  located at the same position  $t$  with the same relative location  $l$ .

## 5. Experimental Results

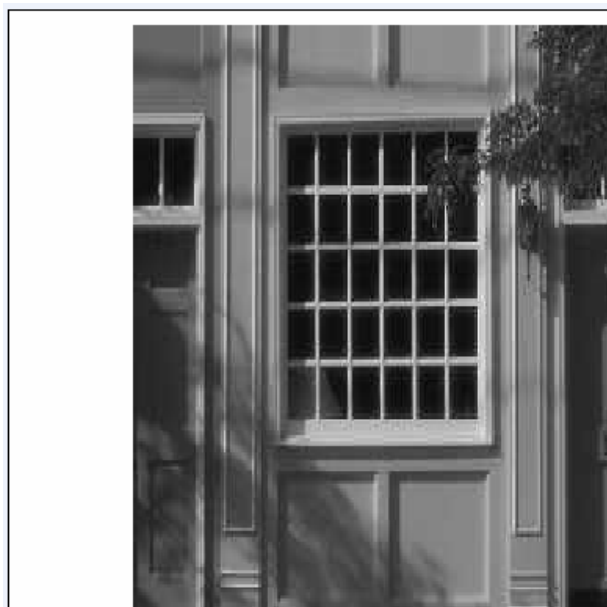
The above two transformations on the newly designed virtual hexagonal structure are implemented using C++ programming language and tested on a computer with Intel Pentium IV 2.8GHz CPU and 480MB of RAM. Experimental results of the two image transformations on grey-level images are presented here.

For any given destination spiral address, the image can be translated fast to the target address. A sample image, called "building" with size of  $384 \times 384$  is shown in Figure 5. An example of image translation on the virtual hexagonal structure is shown in Figure 6. In Figure 6, the "building" is translated to a new location while the central hexagonal pixel is moved to the

location at the spiral address of 45632. It is easily computed that transformation shifts the image by 70 sub-pixels downwards and 560 sub-pixels rightwards, which is equivalent to 9 square pixels downwards and 80 square pixels rightwards. As seen from Figure 6, the translated image does not change any resolution. Furthermore, the operation can be completed very fast. The total time to complete the translation of an image with size of 384\*384 is 0.29 seconds.



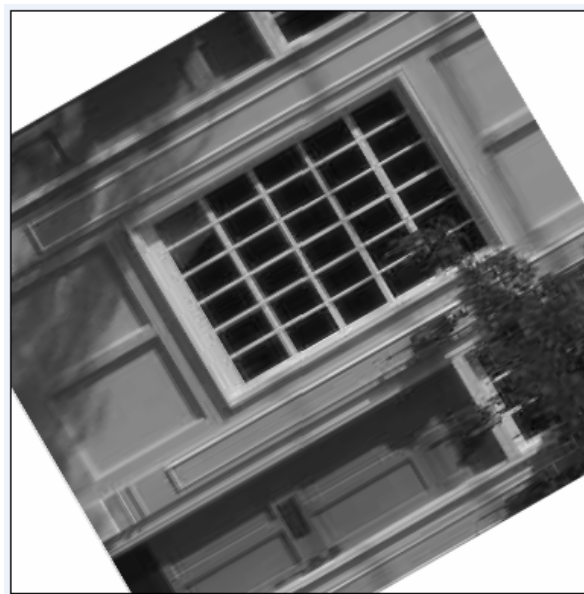
**Figure 5. A sample image, “building”**



**Figure 6. The “building” image is translated to the destination at spiral address 45632**

From Figure 6, we can see that if a sub-pixel is moved out the original image area, its intensity information is not retained, and if a sub-pixel cannot be obtained by moving any sub-pixel in the given direction and for the given distance, its intensity is replaced by 255 (assuming we are working on a 256 grey level image).

The advantages of this newly constructed virtual hexagonal structure can also be demonstrated by image rotations. Figure 7 shows the rotation of the image “building” by 60 degrees in clockwise.



**Figure 7. An example of image rotation on the virtual hexagonal structure, where the original image “building” in Figure 5 is rotated by 60 degrees in clockwise direction**

Again, there is almost no loss of image resolution in the rotated image, and no distortion introduced. Also, the processing speed is much faster than previous mimicking scheme [5]. The total time to complete the rotation computation for an image with size of 384\*384 is 1.1 seconds, and an image with size of 256\*256 is 0.5 seconds. Compared with the method introduced in [5] that takes minutes to complete a rotation, a great improvement has been achieved using this virtual structure. When more powerful computers are used and the algorithms are optimized, the computation speed can be further improved.

## 6. Conclusions and Discussion

In this paper, we have developed algorithms to convert between the traditional square structure and a newly developed virtual hexagonal structure. Based on the virtual hexagonal structure, algorithms for image translation and rotation have been presented. As we do not compute the light intensities for virtual hexagons, image resolution is maintained during the

transformations, and we save the processing time and memory storage.

In our implementation, we adopt the ideas of two operations defined on Spiral Architecture, namely spiral addition and spiral multiplication, and use them for translation and rotation. However, we do not perform these two operations to avoid a large amount of time requested for the complex computations on the virtual structure. This is very different from any of previous approaches, and has significantly improved the performance in terms of speed and complexity.

The two transformations performed in this paper provide a more flexible tool for image translation and rotation. Using our virtual structure, we can not only translate images from pixels to pixels as in square structure, but also translate images from sub-pixels to sub-pixels showing potential for more accurate image matching. Furthermore, in addition to the ability to perform image rotation with  $90^\circ$  as in square structure, we have shown efficient ways to rotate images by multiple  $60^\circ$ s.

Images can be smoothly and easily transferred between the traditional square structure and the hexagonal structure without any change in image shape. The reason to separate a square pixel into  $7 \times 7$  sub-pixel grid is because the total number of sub-pixels that is 49 (the next higher one is  $8 \times 8 = 64$ ) for each square pixel separated in this way is closest to the total number of sub-pixels that is 56 for each hexagonal pixel. Furthermore, this separation retains the image resolution as discussed in Section 3.

This virtual 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. 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 display.

Unlike the Virtual Spiral Architecture shown in [5], the virtual structure does not require complex computation for determining the regions of hexagonal pixels, and does not request to 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.

One other important note to make is that the shape of image for transformations does not have to be hexagon-like as shown in Figure 1. We can perform transformations directly based on the rectangle-like images as seen in our experimental results.

To complete the basic transformations on the virtual hexagonal structure, image scaling will be performed in another paper.

We list other future work related to this paper as follows.

In order not to lose any image information after a transformation, an algorithm is to be developed to

properly assign the pixels that are moved out the image areas to the pixels who intensities were replaced by 'white' in this paper. The idea commonly used for spiral addition and multiplications can be adopted. This may request a new way of addressing scheme.

On Spiral Architecture, we have performed rotation with any angle without scaling. This work can be extended to the virtual hexagonal structure discussed in this paper.

## References

- [1] [1] B.K.P. Horn, *Robot Vision*. 1986: MIT Press, Cambridge, MA & McGraw-Hill, New York, NY.
- [2] [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] [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] [4] X. He,, *2-D Object Recognition with Spiral Architecture*. 1999, PhD Thesis, University of Technology, Sydney.
- [5] [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] [6] 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.
- [7] [7] X. He, T. Hintz, Q. Wu, H. Wang and W. Jia, *A new simulation of Spiral Architecture*, Proc. of International Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, June 2006, to appear.
- [8] [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] [9] R.M. Mersereau, *The processing of Hexagonally Sampled Two-Dimensional Signals*. Proceedings of the IEEE, 1979. 67: p. 930-949.