

# Image Transformation on Hexagonal Structure Based on Conversion between 1D and 2D Coordinates

Yuhuang Ye<sup>1</sup>, Xiangjian He<sup>2</sup>, Jianmin Li<sup>2,3</sup>, Wenjing Jia<sup>2</sup> and Qiang Wu<sup>2</sup>

<sup>1</sup>College of Physics and Information Engineering  
Fuzhou University  
China  
yeyuhuang@fzu.edu.cn

<sup>2</sup>Centre for Innovation in IT Services and Applications (iNEXT)  
University of Technology, Sydney  
Australia  
{sean, jianmin, wejia, wuq}@it.uts.edu.au

<sup>3</sup>College of Mathematics and Computer Science  
Fuzhou University  
China

**Abstract.** Spiral Architecture, a hexagonal image structure is a novel and powerful approach to machine vision system. The pixels on Spiral architecture are geometrically arranged using a 1D (Spiral) addressing scheme in an ascending order along a spiral-like curve. Spiral addition and Spiral multiplication are defined based on the Spiral addresses on Spiral Architecture. These two fundamental operations result in fast and easy translation, rotation and separation on images, and hence play very important roles for image processing on Spiral Architecture. Moreover, 2D coordinates according to rows and columns defined on Spiral Structure provide a good mapping to the ordinary 2D coordinates defined on the common square image structure. Therefore, how to convert the 1D Spiral addresses from and to the 2D coordinates on Spiral Architecture has become very important to apply the theory developed on a hexagonal image structure for image processing (e.g., rotation). In this paper, we perform a fast way to correctly locate any hexagonal pixel when its Spiral address is known, and compute the Spiral address of any hexagonal pixel when its location is known. As an illustration of the use of conversions, we demonstrate the accurate image translation and rotation using experimental results.

**Keywords:** hexagonal structure, Spiral Architecture, image transformation.

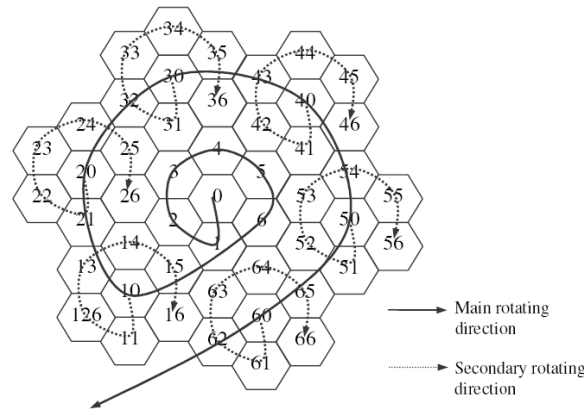
## 1 Introduction

Fast (or real-time) image transformation including image translation and rotation forms core operations in many applications such as medical image processing, computer vision, computer graphics and pattern recognition [1]. Conventional image

transformation involves the mapping of pixels in the domain which is the original image coordinate system to pixels in the range which is the transformed image coordinate system [2] based on the traditional square image structure. Rotation of an image usually performs trigonometric operations on each pixel location and is not reversible. CORDIC, developed by Volder [3] is a popular hardware efficient algorithm that can be used to compute the trigonometric operations. Although a CORDIC engine requires low hardware resources, it is iterative in nature, which can lead to intensive computations. In this paper, Spiral Architecture provides a more flexible and reversible image translation and rotation based on a hexagonal image structure.

The advantages of using a hexagonal structure to represent digital images have been investigated for more than thirty years. The arrangement of hexagonal pixels on a hexagonal structure is similar to the distribution of cones on human retina [4]. In order to properly address and store hexagonal image data, a 1D addressing scheme for a hexagonal structure, called Spiral Architecture, as shown in Figure 1 [5]. Based on the addressing scheme, two algebraic operations called Spiral addition and Spiral multiplication have been defined and used to easily and fast perform image translation and rotation with scaling respectively.

We would like to draw your attention to the fact that it is not possible to modify a paper in any way, once it has been published. This applies to both the printed book and the online version of the publication. Every detail, including the order of the names of the authors, should be checked before the paper is sent to the Volume Editors.



**Fig. 1.** Spiral Architecture with spiral addressing [5].

In order to arrange hexagonal pixels also in rows and columns as seen in traditional square structure and to easily locate pixel location, in [5], He et al. defined the rows and columns on the hexagonal image structure.

The contributions of this paper is to perform the conversion between the 1D Spiral addresses and the 2D coordinates according to rows and columns on Spiral Architecture in order to take the advantages of both 1D and 2D coordinate systems.

The rest of this paper is organized as follows. In Section 2, we briefly review the relationship between the 1D and 2D addressing schemes on Spiral Architecture. In Section 3, the algorithm to convert from a 1D address to corresponding 2D row and column is presented. In Section 4, we perform the conversion from 2D coordinates to 1D Spiral addresses. Experimental results together with sample translations and rotations are demonstrated in Section 5. We conclude in Section 6.

## 2 Spiral Architecture

In this section, we review the methods to locate hexagonal pixels and the definition of rows and columns on Spiral Architecture.

### 2.1 Iterative locating hexagonal pixels

For the whole image represented on Spiral Architecture, 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. Let  $L(a)$  denote the relative location of the hexagonal pixel with Spiral address  $a$  to the central pixel with Spiral address 0. From Figure 1, as shown in [6], 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

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

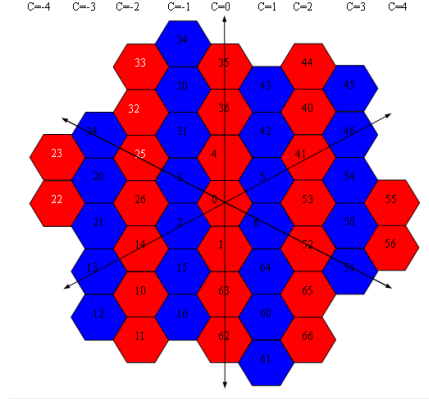
such that

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

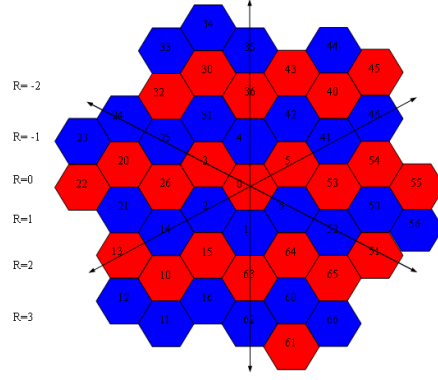
### 2.2 Definition of rows and columns

Following the definition shown in [5], let  $R$  and  $C$  represent the number of rows and number of columns needed to move from the central hexagonal pixel to a given hexagonal pixel taking into account the moving direction corresponding to the signs of  $R$  and  $C$ . Here, pixels on the same column are on the same vertical line. For example, as shown in Figure 2, pixels with addresses 43, 42, 5, 6, 64, 60 and 61 are on the same column with  $C = 1$ . The row with  $R = 0$  consists of the pixels on the horizontal line passing the central pixel and on the columns with even  $C$  values, and the pixels on the horizontal line passing the pixel with address 3 and on the columns with odd  $C$  values. Other rows are formed in the same way. For example, pixels with

addresses 21, 14, 2, 1, 6, 52, 50 and 56 are on the same row with  $R = 1$ . Figure 3 show rows in a hexagonal structure consisting of 49 hexagons.



**Fig.2.** Columns on a hexagonal structure [5].



**Fig.3.** Rows on a hexagonal structure [5].

### 3 Conversion from 1D to 2D Coordinates

From Figure 1, it is easy to see that

$$L(a \times 10^i) = 3L(a \times 10^{i-1}) - 2L((a-1) \times 10^{i-1}) \quad (2)$$

$$L(10^i) = 3L(10^{i-1}) - 2L(6 \times 10^{i-1}) \quad (3)$$

for  $i = 1, 2, \dots, a = 2, 3, \dots, 6$ .

For a given Spiral address  $a$ , let

$$L(a) = [a_x, a_y] \quad (4)$$

and  $R$  and  $C$  values at  $L(a)$  be  $R_a$  and  $C_a$  respectively. Suppose the distance between any two neighbouring hexagonal pixels is 1, then according to Figure 1, we have that

$$a_y = A \times \frac{1}{\sqrt{2}} \quad \text{and} \quad C_a = A = \sqrt{2}a_y, \quad (5)$$

where  $A$  is an integer (either positive or negative).

We also have that

$$a_x = \begin{cases} B + \frac{1}{\sqrt{2}} & \text{when } A \text{ is odd,} \\ B & \text{when } A \text{ is even;} \end{cases} \quad (6)$$

and

$$R_a = B = \max\{\text{int } c \mid c \leq a_x\}, \quad (7)$$

where B is an integer (either positive or negative).

From Equations (5) and (7) above, we can conclude that for any given hexagonal pixel with Spiral address  $a$ , its row and column can be easily computed and determined by Equations (5) and (7).

#### 4 Conversion from 2D to 1D Coordinates

For any given hexagonal pixel  $a$ , if its row and column i.e.,  $R_a$  and  $C_a$  are known, then its location relative to the central hexagonal pixel with Spiral address 0 can be computed and determined using Equations (5) and (6) above. Assume that the maximum Spiral address used for the image representation on the Spiral Architecture is not bigger than  $7^N - 1$ .

Then we determine the Spiral address of the given hexagonal pixel  $a$  in the following steps.

1. Applying Equations (1) – (3), determine the locations of hexagonal pixels with addresses  $b \times 10^N$ , for  $b=1, 2, \dots, 6$ .
2. Select the integer

$$a_N \in \{0,1,2,\dots,6\}$$

such that

$$\begin{aligned} |(a_x, a_y) - L(a_N \times 10^N)| = \\ \min\{|(a_x, a_y) - L(b \times 10^N)| \mid b = 0,1,2,\dots,6\}. \end{aligned}$$

3. Similar to Step 1, if  $N-1 \geq 0$ , compute

$$L(a_N b \times 10^{N-1})$$

for  $b=1, 2, \dots, 6$ .

4. Select the integer

$$a_{N-1} \in \{0,1,2,\dots,6\}$$

such that

$$\begin{aligned} |(a_x, a_y) - L(a_N a_{N-1} \times 10^{N-1})| = \\ \min\{|(a_x, a_y) - L(a_N b \times 10^{N-1})| \mid b = 0,1,2,\dots,6\}. \end{aligned}$$

5. Similar to Steps 3 and 4 above, perform an iterative process by reducing the  $N$  by 1 at the beginning of each iteration and we will eventually find all

$$a_i \in \{0,1,\dots,6\}, i = 0,1,\dots,N$$

such that the Spiral address of the hexagonal pixel  $a$  is

$$a_N a_{N-1} \cdots a_0.$$

## 5 Experimental Results

To illustrate the application of the above derived conversions, we implement two transformations on a newly designed virtual Spiral Architecture [6] using C++ programming language and test them 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\*384 is shown in Figure 4. An example of image translation on the virtual hexagonal structure is shown in Figure 6. In Figure 5, the “building” is translated to a new location while the central hexagonal pixel is moved to the location at the Spiral address of 33506. It can be easily computed that transformation shifts the image by how many rows and columns on the hexagonal structure using the algorithms developed in Section 3. 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 including the time cost for conversion of coordinates.

The advantages of this newly constructed virtual hexagonal structure can also been demonstrated by image rotations. Figure 6 shows the rotation of the image “building” by 60 degrees in clockwise. The corresponding Spiral address which is 2 for this rotation is computed fast using the algorithms presented in Section 4.



**Fig.4.** A sample image, “building”.



**Fig.5.** The “building” image is translated to the destination at spiral address 33506.



**Fig.6.** An example of image rotation on the virtual hexagonal structure.

The total time to complete the rotation computation for an image with size of  $384 \times 384$  is 1.1 seconds, and an image with size of  $256 \times 256$  is 0.5 seconds including

the time for the conversion of coordinates. Compared with the method introduced in [7] that takes minutes to complete a rotation, a great improvement has been achieved using this virtual structure.

## 5 Experimental Results

In this paper, we have developed algorithms to convert between the 1D and 2D coordinates on Spiral Architecture. Based on a virtual hexagonal structure, algorithms for image translation and rotation have been presented using the coordinate conversion results. The conversion is accurate and the speed is fast.

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. It is very different from any other approaches that our algorithms for image rotation can rotate image with any angle without loss of any image information and the rotation process is reversible.

## References

1. Suchitra, S., Lam, S.K., Srikanthan, T.: High-throughput Rotation Using Sign-prediction Based Redundant CORDIC Algorithm. In: 2004 IEEE International Conference on Image Processing, pp. 2833–2836, IEEE Press, New York (2004)
2. Srinivasan, S.: Reversible Image Rotations with Modulo Transforms. In: 2005 IEEE International Conference on Image Processing, pp. 137–140, IEEE Press, New York (2005)
3. Volder, J.E.: The CORDIC Trigonometric Computing Techniques. JRE Trans. Electron. Computing, Vol. EC-8, pp. 330–334 (1959)
4. Staunton, R.: The design of Hexagonal Sampling Structures for Image Digitization and Their Use with Local Operators. Image and Vision Computing, vol. 7(3), pp. 162–166 (1989)
5. He, X., et al.: Uniformly Partitioning Images on Virtual Hexagonal Structure. In: 9th International Conference on Control, Automation, Robotics and Vision, pp. 891–896 (2006)
6. He, X., et al.: Basic Transformations on Virtual Hexagonal Structure. In: 3rd International Conference on Computer Graphics, Imaging and Visualization, pp. 243–248, (2006)
7. Wu, Q., et al.: Virtual Spiral Architecture. In: International Conference on Parallel and Distributed Processing Techniques and Applications, vol. 1, pp. 399–405 (2004)