Bilinear Interpolation on a Virtual Hexagonal Structure

Xiangjian He, *Senior Member, IEEE,* Jianmin Li and Namho Hur

***Abstract*-- Spiral Architecture (SA) is a relatively new and powerful approach to machine vision system. The geometrical arrangement of pixels on SA can be described as a collection of hexagonal pixels. However, all the existing hardware for capturing image and for displaying image are produced based on rectangular architecture. Therefore, it becomes important to find a proper software approach to mimic SA so that images represented on the traditional square structure can be smoothly converted from or to the images on SA. For accurate image processing, it is critical to best maintain the image resolution during the image conversion. In this paper, we present an algorithm for bilinear interpolation of pixel values on a simulated SA. Our experimental results show that the bilinear interpolation improves the image representation accuracy while keeping the computation simple.**

***Index Terms*—Image interpolation, Spiral Architecture, image processing, hexagonal structure.**

I. INTRODUCTION

THE The advantages of using a hexagonal grid to represent digital images have been investigated for more than thirty years [1-5]. The importance of the hexagonal representation is that it possesses special computational features that are pertinent to the vision process [4]. Its computational power for intelligent vision has pushed forward the research in areas of image processing and computer vision. 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 [6-7].

In spite of its numerous advantages, a problem that limits the use of hexagonal image structure is the 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 approaches using rectangular pixels [1-2], pseudo hexagonal pixels [3], mimic hexagonal pixels [4] and virtual hexagonal pixels [5,8]. The use of these techniques provides a practical tool for image processing on a hexagonal structure and makes it possible to carry out research based on a hexagonal structure using existing computer vision and graphics systems.

The new simulation scheme as presented in [8] was developed to virtually mimic a special hexagonal structure, called *Spiral Architecture* (SA) [4]. In this scheme, each of the original square pixels and simulated hexagonal pixels is regarded as a collection of smaller components, called *sub-pixels*. The light intensities of all sub-pixels constituting a square pixel (or hexagonal) are assigned the same value as that of the square pixel (or hexagonal) pixel in the square (or hexagonal) structure. This simple assignment method does not give accurate enough intensity interpolation of sub-pixels, and hence results in some resolution loss when images are converted between the square structure and the hexagonal structure.

In this paper, we present a new assignment scheme using the simple bilinear interpolation algorithm [9] which can be easily and fast implemented. We will use experimental results to show the improvement in terms of PSNR and other indexes after conversion between images represented in the two different image structures.

The rest of this paper is organized as follows. In Section 2, we briefly review SA and its simulation as shown in [8]. In Section 3, the bilinear interpolation scheme is presented. The experimental results are demonstrated in Section 4. We conclude in Section 5.

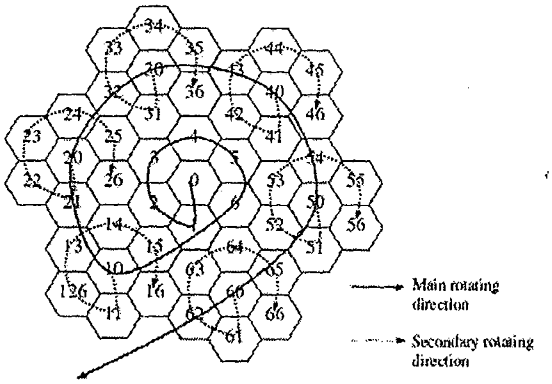

Fig. 1. Spiral Architecture with spiral addressing

II. SA AND ITS SIMULATION

A collection of 49 hexagonal pixels is shown in Figure 1 below. In order to properly address and store hexagonal images data, Sheridan [10] proposed a one-dimensional addressing scheme for SA, a hexagonal structure (Figure 1 [8]).

Xiangjian He is with the Computer Vision Research Group, University of Technology, Sydney, PO Box 123, Broadway, 2007, NSW, Australia. (Contact telephone: 61-2-95141814, e-mail: sean@ it.uts.edu.au).
Jianmin Li is with the School of Computer and Mathematics, Fuzhou University, Fujian, 320002, China.
Namho Hur is with the Digital Broadcasting Research Division, [3]Electronic and Telecommunications Research Institute, South Korea.

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 example, 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

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

If we use $L(a)$ to denote the relative location of the hexagonal pixel with spiral address a to the central pixel, then as shown in [8] the relative location of the pixel with a given spiral address

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

can be computed by

$$L(a_n a_{n-1} \cdots a_1) = \sum_{i=1}^{n} L(a_i \times 10^{i-1}). \tag{2}$$

A. Construction of a Virtual SA

Because there has been no hardware available for image display and capture on hexagonal structure, in this section, we review a software approach to the construction of virtual hexagonal structure [8]. To construct hexagonal pixels, in [8], each square pixel was first separated into 7×7 small pixels, called *sub-pixels*. To be simple, the light intensity for each of these sub-pixels was set to be the same as that of the pixel from which the sub-pixels were separated. Each virtual hexagonal pixel was formed by 56 sub-pixels as shown in Figure 2. Figure 2 shows a collection of seven hexagonal pixels constructed with spiral addresses from 0 to 6. The light intensity of each virtual hexagonal pixel was approximated as the average of the light intensities of the 56 sub-pixels that constitute the hexagonal pixel. For a hexagonal pixel at the image boundary, it may not find all its 56 sub-pixels. In this case, the light intensity of this incomplete hexagonal pixel can be computed as the average of the intensities of the all its sub-pixels found.

It is not difficult to locate each virtual hexagonal pixel when the size of an image is known. 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. Then the centre of the virtual hexagonal structure can 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 sub-pixels. Thus, the first (or the central) hexagonal pixel with address 0 consists of 56 sub-pixels has its 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, all sub-pixels for all hexagonal pixels can be assigned using (2). Therefore, the pixel (or intensity) values of all hexagonal pixels can be computed and an image represented on a square structure is then converted to an image on the virtual hexagonal structure.

									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									

Fig. 2. A cluster of seven hexagonal pixels

B. Conversion from Virtual Hexagonal Structure to Square Structure

Converting images from the virtual structure to the square structure can be simply performed in the same way as from square to hexagonal.

All of the 56 sub-pixels constituting each individual hexagonal pixel are assigned the same intensity value as of the hexagonal pixel. After this step, all sub-pixels in the virtual square structure will have been re-assigned intensity values, which may be different from the original. The intensity value of each square pixel can then be computed as the average of the intensities of the 7×7 sub-pixels that form the square pixel as shown in [11].

III. Bilinear Interpolation for Image Conversion

The approach introduced in the previous section for image conversion between the square structure and the virtual hexagonal structure is simple. However, the reconstructed image may not be accurate enough or may not be close enough to the original image because of potential loss of image resolution. In order to improve the conversion accuracy, in this section, we adopt the bilinear interpolation method that was originally proposed for image interpolation on the square structure. The detailed approach is presented as follows.

A. Conversion from Square Structure to Hexagonal Structure

As shown in Section 2, each square pixel is separated into 7×7 sub-pixels and each hexagonal pixel is formed using 56 sub-pixels as shown in Figure 2.

For every hexagonal pixel, we can compute the coordinates of its centre at the two dimensional coordinate system using Equation (2). Let us denote the arbitrarily given hexagonal pixel by X. Then in the original image space, there exist four square pixels denoted by A, B, C and D, as shown in Figure 3, lying on two consecutive rows and columns such that, point X falls onto the rectangle with vertices at A, B, C and D.

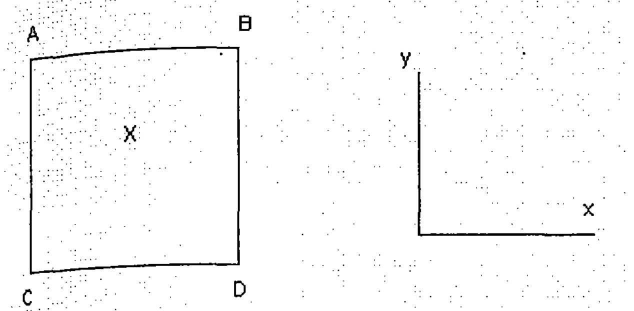

Fig. 3. A hexagonal pixel X located on a rectangle formed from square pixels A, B, C and D

Let us denote the coordinates of A, B, C and X by (A_x, A_y), (B_x, B_y), (C_x, C_y) and (X_x, X_y) respectively. Let

$$\alpha = \frac{|A_x - X_x|}{|A_x - B_x|}, \beta = \frac{|A_y - X_y|}{|A_y - C_y|}.$$

Then, it is easy to derive that

$$X = (1-\alpha)(1-\beta)A + \alpha(1-\beta)B + (1-\alpha)\beta C + \alpha\beta D.$$

Let f be the image brightness function that maps a pixel (either square pixel or hexagonal pixel) to its light intensity value. Then the intensity value assigned to X using the bilinear interpolation method as shown in [9] is computed as

$$f(X) = (1-\alpha)(1-\beta)\cdot f(A) + \alpha(1-\beta)\cdot f(B) + (1-\alpha)\beta\cdot f(C) + \alpha\beta\cdot f(D). \quad (3)$$

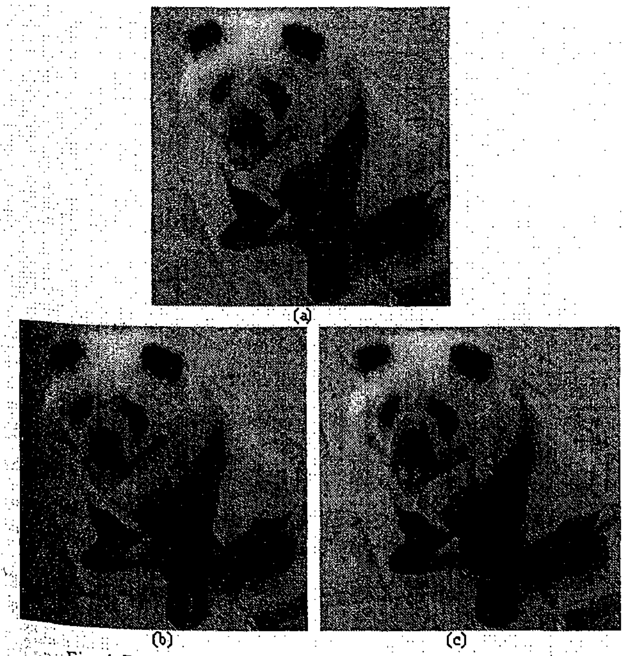

Fig. 4. Results from data 1 using two interpolation methods

B. Conversion from Hexagonal Structure to Square Structure

When the intensities for all virtual hexagonal pixels are obtained using Equation (3), mapping images from the hexagonal structure to the original square structure is performed in the same way as shown in Subsection B in Section 2. All square pixels are re-assigned new intensity values. Images represented on the virtual hexagonal structure can then be displayed on the traditional square structure with the modified pixel values.

IV. Experimental Results

To assess the two methods objectively, we use three figures of merit, which are PSNR (Peak Signal-to-Noise Ratio), RMSE (Root Mean Square Error) and MAXE (Maximum Error). The formula for computation of PSNR, RMSE and MAXE are given by

$$PSNR = 10\log_{10}\frac{255^2 \times M \times N}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[f(i,j)-g(i,j)]^2},$$

$$RMSE = \sqrt{\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[f(i,j)-g(i,j)]^2}{M \times N}},$$

$$MAXE = \max_{i=0}^{M}\max_{j=0}^{N}|f(i,j)-g(i,j)|,$$

where $M \times N$ is the image size, $f(i, j)$ represents the original intensity value of the pixel at location (i, j), and $g(i, j)$ presents the re-assigned intensity value of the pixel at location (i, j) after an interpolation algorithm. The bigger the PSNR is, the closer the match between the original and the modified images is. Similarly, the smaller the RMSE or MAXE is, the better match between the two images.

In order to compare the interpolation methods shown in Sections 2 and 3, two black and white images with 256 grey levels are employed.

The first dada is from a dog image. The original image is shown in Figure 4(a). Figure 4(b) shows the reconstructed image after using the interpolation method shown in Section 2 for image representation on the virtual hexagonal structure. Figure 4(c) shows the reconstructed image after using the bilinear interpolation shown in Section 3. We can see that the result provided by the bilinear algorithm has better visual effect. This can be more clearly found when the images are zoomed 2 or 3 times.

For the first data, the index values corresponding to PSNR, RMSE and MAXE are shown in Table 1. From Table 1, one will find that the PSNR provided by the bilinear method is about 10% higher than the old interpolation method. Meanwhile, the RMSE and MAXE provided by the bilinear method are smaller. Hence, the bilinear method gives more accurate result with less resolution loss when transferring images between the two structures.

Table 1. Comparison of two interpolation methods for data 1

Method \ Index	PSNR	RMSE	MAXE
Simple Interpolation	36.600	3.402	93
Bilinear Interpolation	40.195	2.249	41

The second data is from a lady image. The original image is shown in Figure 5(a). Figures 5(b) and 5(c) are the results obtained using the old interpolation and the bilinear interpolation methods respectively. The index values are listed in Table 2. As can be seen from Figure 5 and Table 2, the conclusions similar to the first data can be made.

Fig. 5. Results from data 2 using two interpolation methods

Table 2. Comparison of two interpolation methods for data 2

Method \ Index	PSNR	RMSE	MAXE
Simple Interpolation	33.038	5.661	104
Bilinear Interpolation	37.620	3.341	78

The superiority of using the bilinear method over the old method is very obvious especially from the values of MAXE obtained.

V. CONCLUSIONS

In this paper, we have presented a bilinear interpolation method used to obtain light intensities of pixels on a virtual hexagonal structure. The experimental results show that the bilinear method outperforms the interpolation method shown in [8]. When converting between images on square structure and hexagonal structure, the bilinear method gives better match and results in less loss of image resolution.

It is worth to note that, in the bi-linear method, the computation of intensity of each hexagonal does not require to take into account the intensities of all its 56 sub-pixels. This has greatly saved the processing time for interpolation compare with the method shown in Section 2.

The bi-cubic interpolation method as shown in [9] may also be used for image conversion and is expected to give more accurate image matching between the two image structures. However, bi-cubic interpolation may also increase the conversion time. A hybrid method combining a bi-linear method with a bi-cubic interpolation method, which can convert images fast and also provide accurate image matching, will be our future goal to achieve.

In this paper, the bi-linear method is used in one way in the process converting image from the square structure to the hexagonal structure only. As another future work, it is expected that the accuracy can be further improved if the bi-linear interpolation method is also used when converting image from the hexagonal structure to the square structure.

REFERENCES

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

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

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

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

[5] Q. Wu,, X. He, and T. Hintz, *Virtual Spiral Architecture*, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, Vol.1, 2004, pp. 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, Vol.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 (ICICT05), 2005, pp.52-64.

[8] Xiangjian He, Tom Hintz, Qiang Wu, Huaqing Wang and Wenjing Jia, *A New Simulation of Spiral Architecture*, International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV06), 2006, pp.570-575.

[9] Tian Yan, Liu Bin and Li Tao, *A Local Image Interpolation Method Based On Gradient Analysis*, International Conference on Neural Networks and Brain (ICNN&B05), Vol.2, 2005, pp.1202 – 1205.

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

[11] 11. Xiangjian He, Huaqing Wang, Namho Hur, Wenjing Jia, Qiang Wu, Jinwoong Kim, and Tom Hintz, *Uniformly Partitioning Images on Virtual Hexagonal Structure*, 9th International Conference on Control, Automation, Robotics and Vision (IEEE ICARCV06), 2006, to appear.