Bi-cubic Interpolation for Image Conversion from Virtual Hexagonal to Square Structure

Jianmin Li[1] and Xiangjian He[2]

[1]School of Computer and Mathematics
Fuzhou University
China
[2]Faculty of Information Technology
University of Technology, Sydney
Australia

Abstract – *Hexagonal image structure represents an image as a collection of hexagonal pixels rather than square pixels in the traditional image structure. However, all the existing hardware for capturing image and for displaying image are produced based on square pixel image structure. Therefore, it becomes important to find a proper software approach to mimic the hexagonal structure so that images represented on the traditional square structure can be smoothly converted from or to the images on the hexagonal structure. 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 bi-cubic interpolation of pixel values on a hexagonal structure when convert from the hexagonal structure to the square structure. We will compare with the results obtained through bi-linear interpolation for the conversion. Our experimental results show that the bi-cubic interpolation outperforms the bi-linear interpolation for most of testing images at the cost of slower and more complex computation.*

Keywords: Bi-linear interpolation, bi-cubic interpolation, image processing, hexagonal structure.

1 Introduction

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 through a bi-linear interpolation computation as shown in [9].

In [9], when converting an image represented in the square structure to an image in the hexagonal structure, a tri-linear interpolation algorithm was defined and applied to re-compute the intensity value of each sub-pixel.

In this paper, we present a new scheme for assignment of the intensity value of each sub-pixel. We define a bi-cubic interpolation in the hexagonal structure and apply it to re-compute the intensity of each sub-pixel. 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 bi-cubic interpolation scheme in the hexagonal structure is presented. The experimental results are demonstrated in Section 4. We conclude in Section 5.

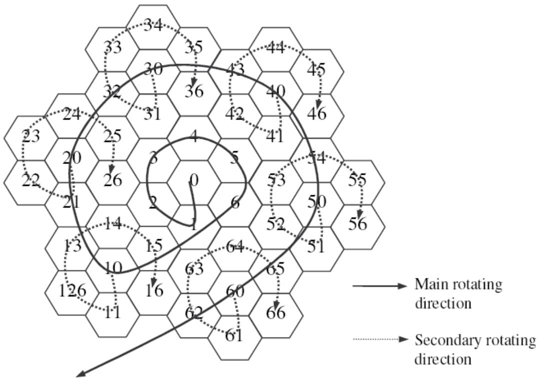

Figure 1. Spiral Architecture with spiral addressing.

2 SA and Its Simulation

A collection of 49 hexagonal pixels together with one-dimensional addressing scheme as shown in [10] is displayed in Figure 1.

To construct hexagonal pixels, in [8], each square pixel was first separated into 7×7 small pixels, called sub-pixels. We assume that the centre of each square pixel is located at the middle sub-pixel of its total 7×7 sub-pixels. 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 collection of virtual pixels covering an image constitutes a virtual hexagonal structure.

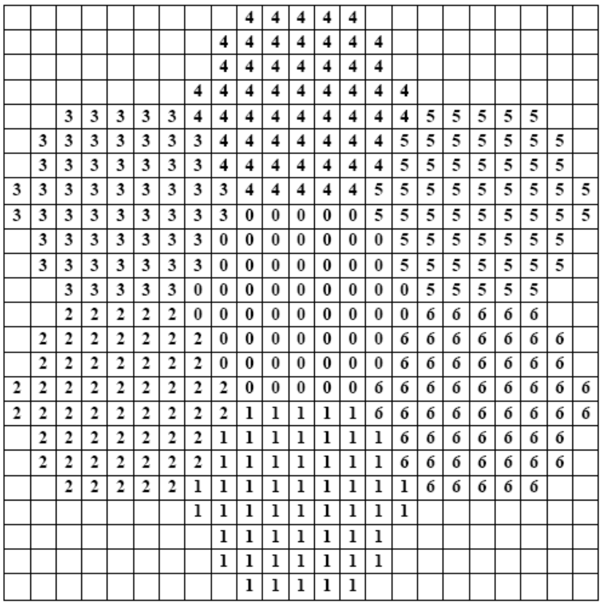

Figure 2. A cluster of seven hexagonal pixels.

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 other hexagonal pixels can be assigned easily as shown in [8].

3 Bi-cubic Interpolation

The approach introduced in [9] used a bi-linear interpolation for conversion of images in square structure to the images in hexagonal structure. This approach is relative simple and fast. On the other hand, the reconstructed image after conversions between image structures may not be accurate enough or may not be close enough to the original image. In order to improve the conversion accuracy, in this section, we adapt the bi-cubic interpolation method, which was originally proposed for image interpolation on the square structure, for image interpolation on hexagonal pixel image structure. The detailed approach is presented as follows.

3.1 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. We then apply the bi-cubic interpolation algorithm defined on square structure to assign the intensity values of all sub-pixels. As defined in [9], the intensity value of each hexagonal pixel is also obtained and it is defined as the intensity value of the sub-pixel (called reference sub-pixel) that is the middle sub-pixel at the 4th row as shown in Figure 3, where the reference sub-pixels are marked 'red'.

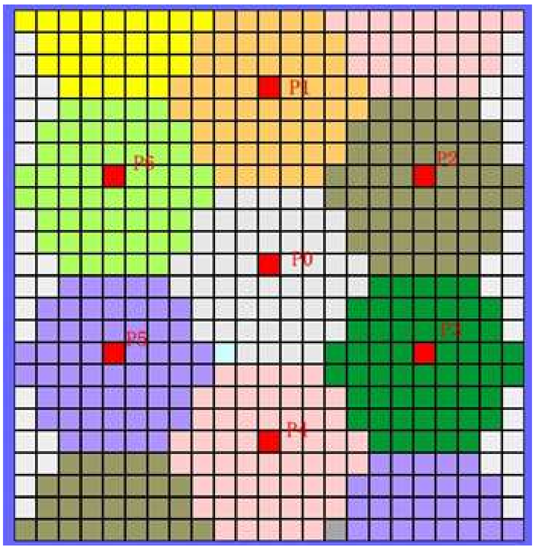

Figure 3. Reference sub-pixels of virtual hexagonal pixels.

3.2 Conversion from Hexagonal Structure to Square Structure

In the hexagonal structure, rows and columns were defined in [11]. Figures 4 and 5 below show the columns and rows defined with 49 hexagonal pixels each.

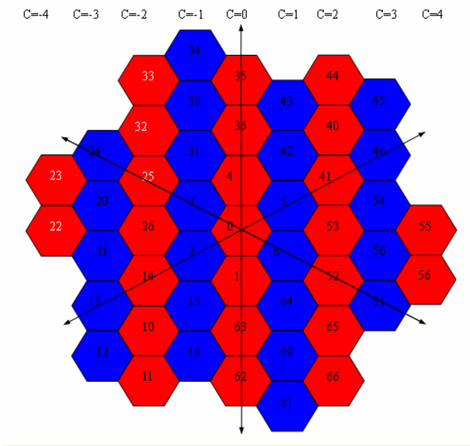

Figure 4. Columns on a hexagonal structure.

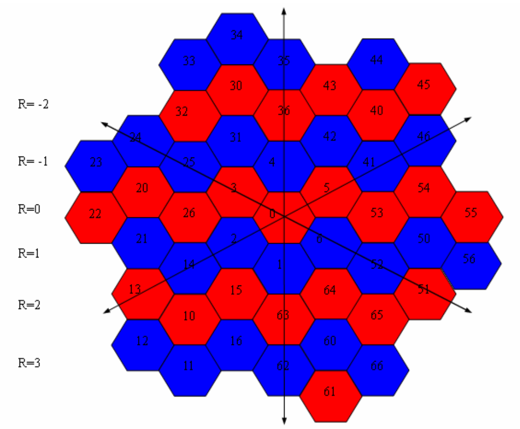

Figure 5. Rows on a hexagonal structure.

As shown in Figure 3, each given sub-pixel is located on a parallelogram form by four reference sub-pixels of four virtual hexagonal pixels (e.g., P_1, P_2, P_3 and P_4 in Figure 3). These four hexagonal pixels are further bounded by 12 other hexagonal pixels (such as P_4, P_5 and P_6 in Figure 3). We use the intensities of these 16 hexagonal pixels to compute the bi-cubic interpolation result for each sub-pixel located on the parallelogram as the bi-cubic interpolation in square structure that takes 16 square pixels into account to determine the intensity value of an interpolated pixel. Thereafter, all sub-pixels have been re-assigned intensity values. Like the approach shown in [9], the intensity value of each square pixel can then be computed as the average of the light intensities of 7×7 that were separated from the square pixel as shown in Section 2.

4 Experimental Results

To compare the cubic approach presented in Section 3 with the linear approach shown in [9], we use two commonly used images (Lena and Mary) for image processing and three merits, which are PSNR (Peak Signal-to-Noise Ratio) and RMSE (Root Mean Square Error). The formula for computation of PSNR and RMSE are given by

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

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

where M×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, the closer the match between the original and the modified images is. Similarly, the smaller the RMSE, the better match between the two images is.

The original Lena and Mary images are shown in Figures 6(a) and 6(b) respectively.

(a) (b)
Figure 6. Original Lena (left) and Mary (right) images.

Figure 7(a) shows the reconstructed Lena image after using tri-linear interpolation method shown in [9] for image representation on the hexagonal structure. Figure 7(b) shows the reconstructed Lena image after using the bi-cubic interpolation shown in Section 3. We can see that the result provided by the bi-cubic algorithm has better visual effect.

(a) (b)
Figure 7. Results using tri-linear interpolation (left) and bi-cubic interpolation (right) on Lena image.

For the Lena image, the index values corresponding to PSNR and RMSE are shown in Table 1. From Table 1, one will find that the PSNR provided by the bi-cubic method is higher than the tri-linear interpolation method. Meanwhile, the RMSE provided by the bilinear method is smaller. Hence, the bi-cubic method gives more accurate result with less resolution loss when transferring images between the two structures.

Table 1. Comparison of two interpolation methods for Lena image.

Method \ Index	RMSE	PSNR
Tri-linear Interpolation	9.1901	28.8644
Bi-cubic Interpolation	8.5226	29.5193

When testing on the Mary image, Figures 8(b) and 8(c) are the results obtained using the tri-linear interpolation and the bi-cubic interpolation methods respectively. The index values are listed in Table 2.

(a) (b)
Figure 8. Results using tri-linear interpolation (left) and bi-cubic interpolation (right) on Mary image.

As can be seen from Figure 8 and Table 2, the conclusions similar to the ones for the Lena image can be made.

Table 2. Comparison of two interpolation methods for Mary image.

Method \ Index	RMSE	PSNR
Tri-linear Interpolation	5.1875	33.7974
Bi-cubic Interpolation	5.1511	33.8587

5 Conclusions and Discussion

In this paper, we have presented a bi-cubic interpolation method used to obtain light intensities of sub-pixels on a virtual hexagonal structure. The experimental results show that the cubic interpolation method outperforms the linear interpolation method shown in [9]. When converting between images on square structure and hexagonal structure, the cubic method gives better match and results in less loss of image resolution than the linear method.

It is worth to note that, in the bi-cubic method, the computation of intensity of each hexagonal does not have to take into account the intensities of all its 56 sub-pixels. This will greatly saved the processing time.

The bi-cubic interpolation method gives more accurate image matching between the two image structures. However, bi-cubic interpolation may also increase the conversion time. A hybrid method combining a linear method with a 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-cubic method is used in both ways in the process converting images from and to the square structure. As another future work, it is expected that the accuracy can be further improved if a tri-cubic interpolation method is proposed when converting image from the hexagonal structure to the square structure. For the idea of how to define the tri-cubic interpolation, one can refer to the tri-linear interpolation defined in [9]. The tri-cubic will take into account 12 hexagonal pixels as the interpolation reference pixels compared with 16 square pixels used for bi-cubic interpolation defined in this paper and in the square image structure.

References

[1] B.K.P. Horn, *Robot Vision*. MIT Press, 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, Vol.7, No.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] Xiangjian He, Jianmin Li and Tom Hintz, *Comparison of Image Conversions between Square Structure and Hexagonal Structure*, in book ‘Advanced Concepts for Intelligent Vision Systems’, LNCS, Springer, vol. 4678, 2007, pp. 262-273.

[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] Xiangjian He, Huaqing Wang, Wenjing Jia, Qiang Wu, Namho Hur, Jinwoong Kim and Tom Hintz, *Uniform Image Partitioning for Fractal Compression on Virtual Hexagonal Structure*, International Journal of Information and Systems Science, Institute for Scientific Computing and Information, vol. 3, no. 3, 2007, pp. 492-509.