How Pseudo Model May Help Evaluate Image Compression on Spiral Architecture

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

Abstract

Although Spiral Architecture (SA) has many advantages in image processing and machine vision, there is no available image capture device yet to support this structure. Hence, in order to implement our theoretical results, it is necessary to construct the SA from the existing image structure, on which the traditional image representation is based. Therefore, in this paper we provide two methods, Mimic model and Pseudo model. We can construct the architecture from the rectangular pixels and thus they share the Spiral addressing mechanism. This paper then presents how Pseudo model may help to evaluate the performance of image compression on Spiral Architecture. Based on the two unique properties of Spiral Architecture, i.e. locality of light intensity and uniformity of image partitioning, possible approaches to image compression may be discovered. Our future work would focus on partitioning the image into blocks and implement compression on the transform domain.

Keywords: spiral architecture, image compression, pseudo model

1. Introduction

The research work to be presented in this paper is based on a novel data structure, Spiral Architecture [1], which is inspired from anatomical considerations of the primate's vision system [2]. On Spiral Architecture, an image is a collection of hexagonal picture elements [3] as shown in Fig. 1. In the case of the human eye, these elements (hexagons) would represent the relative position of the rods and cones on the retina. Each pixel on Spiral Architecture is identified by a designated positive number, called *Spiral Address*. The numbered hexagons form the cluster of size 7^n. The hexagons tile the plane in a recursive modular manner along the spiral direction as shown in Figure1. Any hexagonal pixel has only six neighboring pixels which have the same distance to the centre hexagon of the seven-hexagon unit of vision.

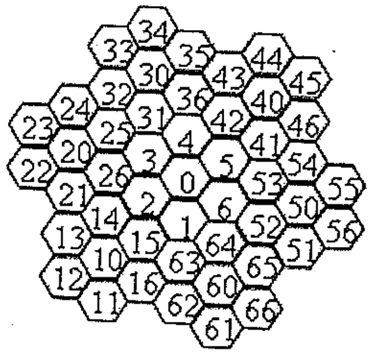


Figure 1. A collection of 49 hexagonal pixels

In our preliminary research on image compression based on Spiral Architecture [4], we focused on the properties of the hexagonal pixel address labeling scheme. The property of interest was the physical proximity of the hexagonal pixels with neighboring addresses. Rectangular systems may, for instance, have vertical physically adjacent pixels but the address distance is the length of a scan line. It was demonstrated that in the Spiral Architecture, unlike the rectangular system, neighboring pixels (for example, the pixels with addresses 0, 1, 2, 3, 4, 5 and 6) have similar intensities.

The research reported in our most recently written paper [5] uses the properties of uniform image partitioning, which is a novel image operation developed recently based on Spiral Architecture. On Spiral Architecture, an image can be partitioned into a few sub-images each of which is a scaled down near copy of the original image. Namely, each sub-image holds all the representative intensity information contained in the original. Using such properties, in our work, the points in the original image first are re-allocated into a few groups (sub-images), rather than being unwound in the spiral address order to be another one-dimensional data set as shown in [4]. The similar pixel intensity is found between the corresponding points in the different sub-images. Then, it is possible to choose one sub-image as a reference image and work out the intensity difference between the reference image and other sub-images such that the original image will be stored by recording only the reference sub-image and the intensity difference information thus giving opportunities for better image compression.

The organization of this paper is as follows. Spiral Architecture representation is reviewed in Section 2 with two different representation models: Mimic model and Pseudo model. We present the experimental results using Pseudo model in Section 3 and then conclude in Section 4.

2. Spiral Architecture Representation

Although Spiral Architecture has many advantages in image processing and machine vision, an image capture device is as yet unavailable to support this structure. Thus to implement our theoretical results, it is necessary to construct the SA from the existing image structure on which the traditional image representation is based. Therefore, in this section, we provide two methods, Mimic model and Pseudo model, to construct the architecture from the rectangular pixels and they share the addressing mechanism of the Spiral Architecture.

2.1 Mimic Model

2.1.1 Representation of hexagonal pixels [6]

We are using a set of four rectangular pixels which are adjacent to each other to mimic a hexagonal pixel. The basic cluster of seven hexagons with Spiral addresses 0~6 are represented in Figure 2.

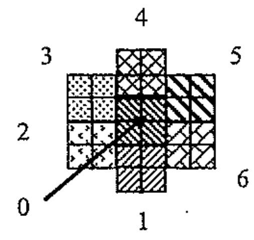


Figure 2. Distribution of 7 pixels constructed from rectangular pixels

It is obvious that our mimic is consistent with the important property of hexagon distribution that each such pixel has exactly six surrounding pixels, and the image represented in the traditional way can be represented as a collection of the new constructed pixels. We also call these mimic pixels 'hexagons' for convenience. This mimic suffers loss of resolution because we are using four rectangular pixels to represent one hexagon. When we implement this scheme to mimic the SA on square-pixel architecture, the light intensity of each hexagon can either simply be the mean of light intensities for the group of four square pixels representing that hexagon, or just the light intensity for one of them, i.e. for the top left square pixel in the group.

2.1.2 Finding Mimic pixels

In Figure 2, the central four pixels are identified as the hexagon with Spiral address zero and we correspondingly set the Cartesian coordinates of this hexagon as (0, 0). Suppose that each square pixel has the unit edge length. Then, the hexagon with Spiral addresses 1, 2, 3, 4, 5 and 6 have corresponding Cartesian coordinates (0, -2), (-2, -1), (-2, 1), (0, 2), (2, 1) and (2, -1) respectively.

The rules to find the Cartesian coordinates for given mimic hexagon address have been stated by He in [6]: if we want to denote the Cartesian coordinates of the mimic hexagon corresponding to the Spiral address a by $Pos(a)$ and assume that j is a positive integer, then the rules say

$$Pos(10_j) = 3Pos(10_{j-1}) - 2Pos(6 * 10_{j-1}), \qquad (2.1)$$

$$Pos(i*10_j) = 3Pos(i*10_{j-1}) - 2Pos((i-1)*10_{j-1}) \quad (for\ i = 2, 3..., 6) \qquad (2.2)$$

More generally, assume that

$$a = k_0 + k_1 10 + k_2 10^2 + \ldots + k_n 10^n, \tag{2.3}$$

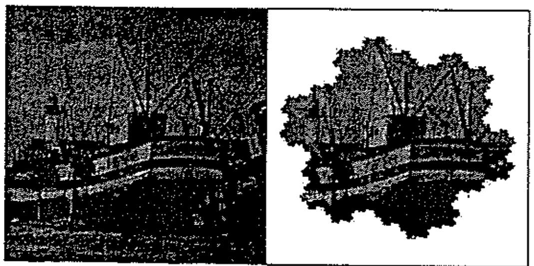
where n is a positive integer such that 7^n is the number of hexagonal pixels in the image, and k_j ($j = 0, 1, 2, \ldots, n$) are integers between 0 and 6. Then,

$$Pos(a) = \sum_{j=0}^{n} Pos(k_j 10^j). \tag{2.4}$$

Hence, from the above procedures, all hexagonal pixels can be constructed with spiral addresses, and their Cartesian coordinates can be computed from the coordinates of the hexagons with the spiral addresses from 0 to 6.

2.1.3 Image representation

As an example, our source image (see Figure 3a) uses 256 grey levels and represented by 384*384 rectangular pixels. Create the mimic hexagonal pixels and assign Spiral addresses to each mimic hexagon as described previously. And then compute the light intensity value at each hexagonal pixel as the mean of the light intensities of the corresponding four rectangular pixels, which together represent the mimic hexagon. As a result, we get a new image in Figure3b and we call it the SA sample image of Figure 3a.



a. in traditional structure b. in Mimic Model

Figure 3. Boat

Comparing the two images, the SA sample image is slightly 'blurred' as it uses four square pixels to represent one mimic hexagon pixel and thus only has one-fourth the resolution of the source image. Moreover, in the sample image, we have 7^5 (16807) mimic hexagons, resulting in 67228 (i.e. 16807*4) traditional square pixels.

2.2 Pseudo Model [7]

The Pseudo model is newly invented by Sheridan [7]. Should hereby we appreciate his kind help to share the information with us before it is published.

2.2.1 Representation of hexagonal pixels

In the Pseudo model, we are using only one rectangular pixel to represent a hexagonal pixel. The basic cluster of seven pixels with Spiral addresses 0~6 are represented in the following figure (see Figure4).

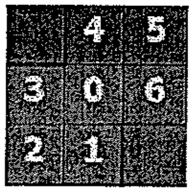


Figure 4. Distribution of 7 pixels constructed from rectangular pixels

Following the same labeling scheme, we could extend the previous cluster to get the first 49 Pseudo hexagons with Spiral addresses 0 to 66. (See Figure5)

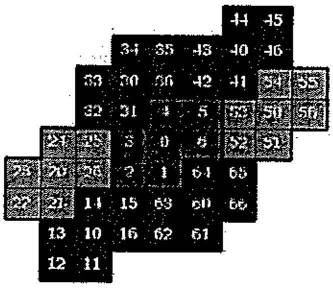


Figure 5. The first 49 Pseudo pixels with labeled spiral addresses

If we repeat the same scheme we are able to cover any image on traditional structure and label each of the pixels with a unique Spiral address. In order to be consistent with the important property of hexagonal distribution that each such pixel has exactly six surrounding pixels, we only consider six of the eight neighbors for the centre pixels. The election of these six neighbors is as shown in the previous figure so that we can keep the properties of Spiral architecture to the greatest extent.

2.2.2 Finding Pseudo pixels

Again the central pixel is labeled with Spiral address 0 and we correspondingly set the Cartesian coordinates of this pixel as (0, 0). Suppose that each square pixel has the length of 1 as edges. Then, the hexagon with Spiral addresses 1, 2, 3, 4, 5 and 6 have corresponding Cartesian coordinates (0, -1), (-1, -1), (-1, 0), (0,1), (1, 1) and (1, 0) respectively.

The rules to find the Cartesian coordinates for given Pseudo pixel with a labeled Spiral address are exactly the same as on the Mimic model we mentioned previously. Follow the formulas 3.1~3.4, given any Spiral address we can easily find the physical location, i.e. the Cartesian coordinates, of the pixel on the image.

2.2.3 Mimic vs. Pseudo

Comparing these two different representation methods, i.e. Mimic and Pseudo models, we may find Pseudo model has greater advantages especially when we do research on image compression.

First of all, Pseudo model establishes a one-to-one relationship between a square pixel on traditional image architecture and a hexagonal pixel on Spiral architecture. Obviously, there is no loss in resolution when converting the original image to Spiral representation. As mentioned earlier, in the Mimic model we use four square pixels to imitate one hexagon so that the image resolution has been reduced to one fourth of the original, which makes it harder to compare the compression performance on Spiral and square architecture. Before the invention of the Pseudo model, we have to evaluate the performances by also reducing the resolution of the original image on square architecture to one fourth. Refer to [5] for details.

Secondly, the Pseudo model keeps most major characteristics of Spiral Architecture as illustrated in Figure 6.

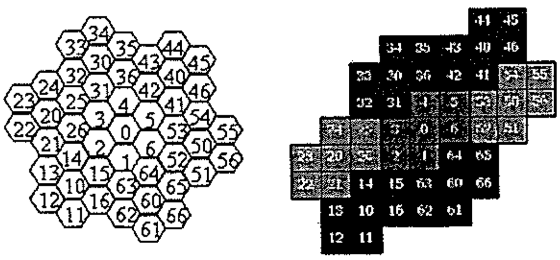


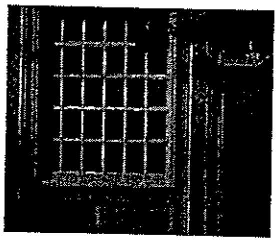
Figure 6. Spiral Architecture and Pseudo Model

First let's focus on the centre hexagon with Spiral address 0, 10, 20, 30 40, 50 and 60 in the Spiral Architecture respectively. Clearly, the first property in spatial domain of Spiral Architecture is that each of these central hexagons has exactly 6 neighbors with continuous Spiral address from the centre so that if we separate the image into blocks containing 7-based numbers of hexagons, i.e. 7, 49 and 343, the hexagons in each block will have continuous Spiral addresses and be physically next to each other. Our newly invented Pseudo model inherits this property perfectly. In addition, if we look closer into the Spiral Architecture, we will find out the hexagon with address 10 is located by extending doubled diameters of each hexagon from hexagon with address 1 through the direction defined by hexagons with address 6 and 1. In a similar manner the hexagon with address 20 is defined by hexagons with addresses 1 and 2. So do the other centre hexagons 30, 40, 50 and 60. Our newly invented Pseudo model inherits all the previous properties so that we could directly implement Spiral operations: Addition and Multiplication defined in [7]. Therefore, based on the perfect heritage from Spiral Architecture introduced by Pseudo model, we can utilize the two compression advantages offered by Spiral Architecture, i.e. locality of pixel density [5] and uniformity of image partitioning [6], and compare the compression performance over the Pseudo model.

3. Experiment Results

3.1 Locality of Pixel Density

We choose four images as Figure 7a~7d and scan each of the images pixel by pixel. All rows of the images are linked together to be a one-dimensional data set. After that we present them in Spiral Architecture based on the Pseudo model. On Pseudo model, all the points in one image are unwound in spiral addressing order to be another one-dimensional data set. We then statistically analysed the two corresponding data sets of both original image and image on Pseudo model by the variance of grey level change within a certain number of pixels. These certain numbers are set to be 7-based integers, i.e. 7, 49, 343 and 2401, in order to make best use of the property of Spiral Architecture. Generally, as the number of pixels considered in the run length (pixels of certain number) increases, the variance of the pixel intensity increases correspondingly. However, the changes in the Spiral Architecture are always less than for the traditional structure when the run length is under the same range. (See Figure 8a~8d)



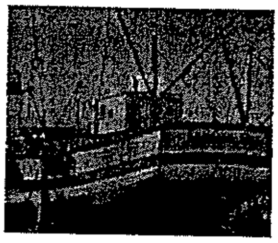


Figure7a.BLD **Figure 7b. Boat**

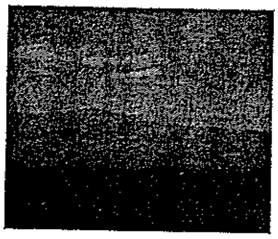


Figure7c. House **Figure7d. Viewnorth**

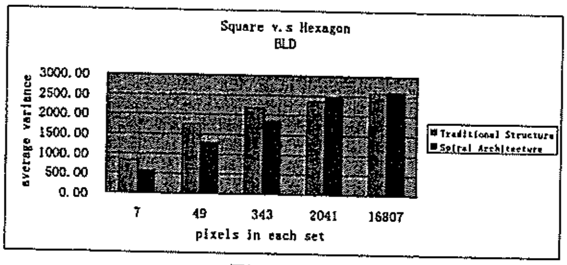


Figure8a.

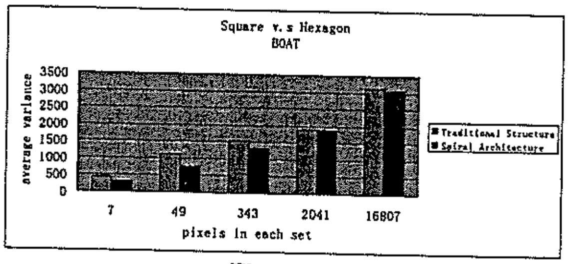


Figure8b.

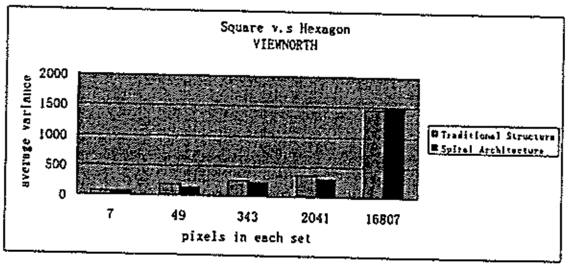


Figure8c.

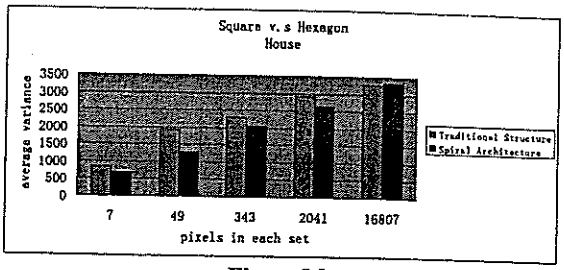
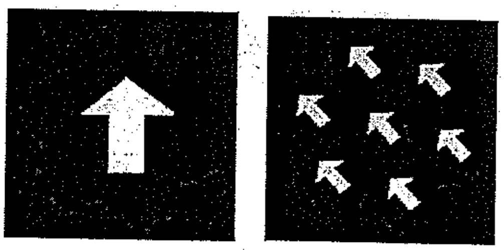


Figure8d.

3.2 Uniformity of Image Partitioning

In this paper, we utilize Spiral Multiplication [6] to implement the uniform image partitioning. First of all, a common Spiral address is selected to multiply the Spiral address of each pixel on an image under the Spiral Multiplication rules. Then the original image will be partitioned into a few sub-images (see Figure9). The number of sub-images is determined by the common Spiral address we have chosen. In this particular case, we have chose 7 as the common Spiral address so that we could have 7 sub-images as the result in Figure 9.

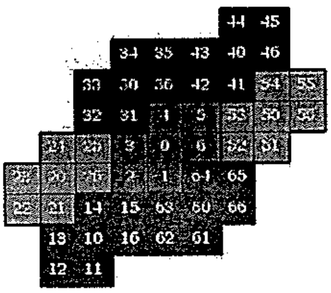


a. Original image b. Result of image partitioning

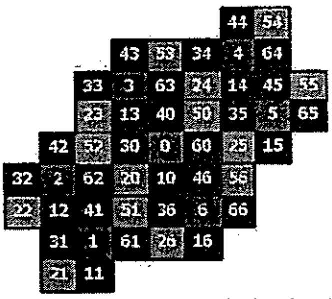
Figure 9. Uniform Image partition on Spiral Architecture

Taking a closer look at these sub-images, it is discovered that each of them is a scaled-down similar copy of the original image. Namely, those sub-images together hold all the representative intensity information contained in the original. Utilizing such properties, rather than being unwound to be a one-dimensional data set based on the Spiral addresses as mentioned formerly, the original image first would be partitioned into a few sub-images and then we would work out the difference of pixel intensity between the corresponding pixels in each of the sub-images. Afterwards, one of sub-images could be chosen as a reference so that the original image will be stored by recording only the reference sub-image and the intensity difference information. Therefore, it may give opportunities for better image compression.

If we implement the Spiral multiplication by seven and present the result on Pseudo model of only 49 pixels, the result is as follows. See Figure10a and Figure 10b.



a. Original first 49 pixels



b. After Spiral multiplication by 7

Figure 10. Spiral multiplication on Pseudo Model

One sub-image has seven pixels and follows the same color pattern. For instance, comparing the first sub-image containing pixels with Spiral addresses 0, 10, 20, 30, 40, 50 and 60 and the second one labeled with Spiral addresses 1, 11, 21, 31, 41,51 and 61, these two sub-images share similar information of light intensity and are able to be coded efficiently by collecting reference and difference information.

4. Conclusion

To sum up, this paper presents how the Pseudo model may help to evaluate the performance of image compression on Spiral Architecture. Based on the two unique properties of Spiral Architecture, i.e. locality of light intensity and uniformity of image partitioning, possible approaches to image compression may be further invented. Our future work would focus on partitioning the image into blocks and implement compression on the transform domain.

5. References

[1] P. Sheridan, T. Hintz. and W. Moore, *Spiral Architecture in Machine Vision*, Proceedings of Australian Occam and Transputer Conference, Amsterdam, 1991.

[2] E. Schwartz, *Computational Anatomy and Functional Architecture of Striate Cortex: A Spatial Mapping Approach to Perceptual Coding*, Vision Research 20, pp.645-669, 1980.

[3] P. Sheridan, T. Hintz and D. Alexander, *Pseudo-invariant image transformations on a hexagonal lattice*, Image and Vision Computing, 18(2000), pp.907-917.

[4] T. Hintz and Q. Wu, *Image Compression on Spiral Architecture*, Proceedings of The International Conference on Imaging Science, Systems and Technology, Las Vegas, Nevada, USA, pp. 201-204, 2003.

[5] Q. Wu, X. He and T. Hintz, *Preliminary Image Compression Research Using Uniform Image Partitioning on Spiral Architecture*, ICITA04, Harbin, China, 2004

[6] X. He, *2D-object Recognition with Spiral Architecture,* PhD thesis, University of Technology, Sydney, 1999

[7] P. Sheridan, *Spiral Counting on a Rectangular Lattice*, PDPTA 2004, Las Vegas, Nevada, 2004.