Proceedings of the 2004 International Conference on Information and Communication Technologies (ICT 2004), November 18-19, 2004
Edited by Dobri Atanassov Batovski and Sergey Alexandrovich Fedoseev, published by Assumption University, Thailand.

Edge Analysis on Rectangular and Hexagonal Architectures

Wenjing Jia, Xiangjian He, and Qiang Wu
Department of Computer Systems
Faculty of Information Technology
University of Technology, Sydney, Australia
E-mail: {wejia, sean, wuq}@it.uts.edu.au

Abstract

This paper examines the gradient-based edge detection algorithm on hexagonal grids and compares it with that on conventional rectangular grids. Since there is no mature hardware for hexagonal-based image capture and display, our research work is based on the newly proposed mimic scheme, called virtual Spiral Architecture, which is demonstrated to make the research work based on hexagonal image processing practically workable on current hardware. From our experiments, it is found that a better edge map can be obtained using this architecture than using the square architecture.

Keywords: *hexagonal-image processing, virtual Spiral Architecture, gradient, edge map.*

1. Introduction

When a scene is observed by a human, the human visual system first segments the scene. Edge detection is an important approach for image segmentation in computer vision systems. This approach measures the rate of change and decides the existence of an edge at each point.

Besides the classical edge detectors, there are many researches proposed in the literature to improve the performance of the edge detection algorithms. So far, all of these edge detection algorithms are based on the traditional image architecture, i.e. square represented images, where the basic image unit is square grid (see Figure 1(a)). An alternative way to represent digital images is based on hexagonally sampled lattices, where the basic unit is hexagonal grid (see Figure 1(b)). For convenience, in this paper, we denote the square represented images as square images and hexagonally represented images as hexagonal images.

Sampling on a hexagonal lattice is a promising solution which has received some attention and been proved to have better efficiency and less aliasing [1]. The importance of the hexagonal representation is that it possesses special computational features that are pertinent to the vision process. Hence, one objective of our research is to investigate the use of hexagonal images for image processing in order to improve the performance of our algorithms. Specifically, we wish to investigate the merit of using the Spiral Architecture, which is a relatively newly proposed hexagonal representation scheme of digital images, to improve the performance of the edge detection.

This paper contains 7 main sections. In section 2, a brief review on hexagonal-based image processing is given. Then we introduce the recently proposed the Spiral Architecture and the virtual Spiral Architecture, on which our research is based, in section 3 and 4 respectively. The gradient-based edge map is illustrated in section 5. We give our experimental results and conclusion in section 6 and 7.

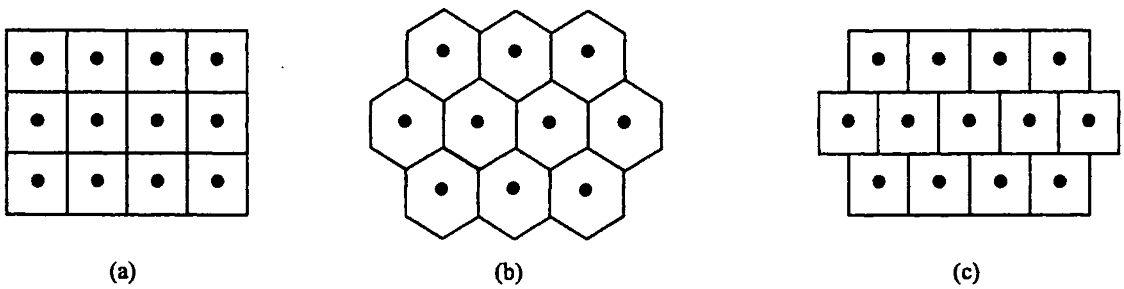


Figure 1. Two-dimensional sampling points grids, (a) rectangular grids, (b) hexagonal grids, and (c) hexagons tiled by rectangles.

2. Review on Hexagonal Image Processing

Hexagonal image processing is not a new idea. Over the past 40 years many researches have examined various aspects of this field. In [2], a literature review on the field of hexagonal image processing was given.

Theoretical studies of hexagonal sampling can be traced to begin with Peterson [3], who concluded that the most efficient sampling schemes were not based on square lattices. Mersereau [1] showed that for circularly band limited signals, 13.4% fewer sampling points are required with the hexagonal grid to maintain equal high frequency image information with the rectangular grid. Myopoulos etc. [4, 5] have shown that connectivity in hexagonal images is more easily defined as it is six-way to either of the nearest neighbours for both the object and the background image components. Whereas on a rectangular grid object, connectivity can be defined as four-way to any of the four nearest neighbours, or eight-way if connectivity to diagonal neighbours is permitted. Background connectivity must be eight-way if object connectivity is four-way or four-way if object connectivity is eight-way. The simpler hexagonal connectivity definition results in simpler skeleton evaluation. Serra [6] has developed many of morphological operators that were currently used for image processing. He prefers the hexagonal grid to the rectangular because of the connectivity definition and the higher symmetry, which lead to simpler processing algorithms. Deutsch [7] investigated the thinning algorithms for use with rectangular, hexagonal, and triangular arrays using the same approach to the development of each algorithm, and the algorithm operating in conjunction with the hexagonal array was found to be the most computationally efficient, produced a skeleton with fewer points than the rectangular, and easily chain coded. Staunton [8] presented an analysis, using mathematical morphology, of the thinning operation and the formation of skeletons from hexagonally sampled images and compared the algorithm experimentally to a similar parallel algorithm designed for a conventional rectangular sampling grid. The hexagonal skeleton exhibited more accurate corner representation, noise immunity, and a processing time of 55% of that required to process the rectangular scheme skeleton. Middleton [2] investigated the edge detection using classical edge detectors on hexagonal images and compared the result with the edge detection on square images. The results showed that 1) the computational requirement for processing a hexagonal image is less than that for a square image, and 2) a better qualitative performance which is due to the compact and circular nature of the hexagonal lattice.

Regarding how to represent hexagonal data and how to store hexagonal image data, different schemes have been proposed. As currently there is no hexagonal-based hardware to capture and display images, most research work on hexagonal image processing are based on Rosenfeld suggested 'brick wall' [9] (see Figure 1(c)).

Our research is based on the recently proposed Spiral Architecture [10], which is a powerful approach to a general purpose image processing. In the following several sub-section, we illustrate briefly the idea of Spiral Architecture and its related operations.

3. Spiral Architecture (SA)

Spiral Architecture (SA) proposed by Sheridan [10] is inspired from anatomical consideration of the primate's vision system. It represents an image as a collection of hexagonal pixels, in contrast with the conventional rectangular pixels.

3.1. Spiral Addressing

It is obvious that the hexagonal pixels cannot be labelled in column-row order as in rectangular architecture. Wüthrich etc. [11] represented the hexagonal coordinates using a skewed coordinate system. Sheridan [10] presented a one dimensional indexing scheme, called *Spiral Addressing*, to address each hexagon on the image, which grows from the centre in powers of seven with a pattern of spiral to label each hexagon by a unique number in base seven (see Figure 2). This unique addressing scheme combined with two later proposed mathematic operations, *spiral addition* and *spiral multiplication*, is called *Spiral Architecture (SA)* [10, 12]. The two mathematical operations are directly related to image translation and image rotation respectively.

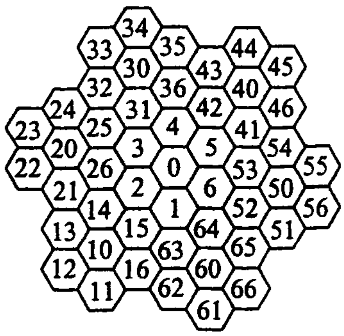


Figure 2. Spiral addressing

ISBN: 974-615-191-6

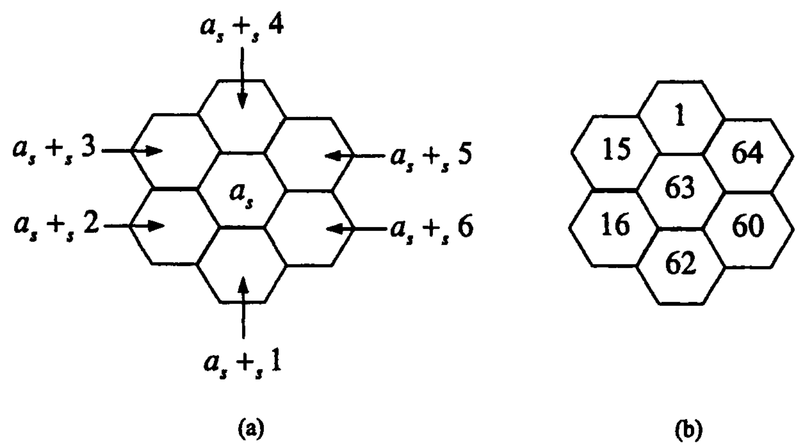


Figure 3. Neighbourhood relationship within Spiral Architecture.
(a) Neighbourhood relationship. (b) An example of neighbourhood.

3.2. Neighbourhood Operation on SA

Neighbourhood operations are often used in image processing. Finding the neighbours of certain pixel in a hexagonal image makes use of the spiral addition operation, which details are given in [10]. In a seven-pixel cluster, the neighbourhood relationship can be determined by spiral addition shown as follows.

Let the spiral address of the central pixel, as shown in Figure 3(a), be denoted by a_s. Then the spiral address of its neighbour pixels can be described by spiral addition, denoted by $+_s$, with a certain number of displacements, as shown in Figure 3 (a). An example of this operation is given in Figure 3 (b).

The SA has some distinguishing features compared to the square image processing. First, a one dimensional index addressing scheme leads to an efficient storage, while the placement of the origin at the centre of the image simplifies geometric transformations of a given image. Finally, the hexagonally sampled image allows non-traditional neighbourhoods with consistent boundary connectivity, which is useful for many computer vision applications.

4. Virtual Spiral Architecture

In spite of the many advantages of Spiral Architecture, the hexagonal-based SA image processing has not been used widely in this area. The main reason is that there is no mature hardware device that is currently available to sample and display images on hexagonal grids. So how to describe the hexagonal image on existing image display devices has once became a serious problem that affects the advanced research on Spiral Architecture.

In order to make the research results based on Spiral Architecture practically workable with the existing image capture devices, He [12] proposed a *mimic Spiral Architecture*, where one hexagonal pixel consists of four traditional square pixels and its gray-level value is the average of those of the corresponding four square pixels (see Figure 4). This mimic scheme preserves the important property of hexagonal architecture that each pixel has exactly six surrounding neighbours. However, because the gray-level value of the mimic hexagonal pixel is taken from the average of the four corresponding square pixels, this mimic scheme introduces loss of resolution, where the resolution of the mimicked image is reduced by four times. In addition, we know that according to hexagonal architecture 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.

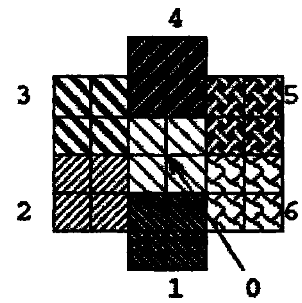


Figure 4. A cluster of 7 mimic hexagons.

Later, Wu etc. [13] constructed a novel mimic scheme called *virtual* Spiral Architecture which is another important milestone for the theoretical research and the practical application exploration of Spiral Architecture. Using virtual Spiral Architecture, images on rectangular architecture (or called square grids as indicated in the Figure 5) can be smoothly converted to virtual Spiral Architecture. Such virtual Spiral Architecture only exists during the procedure of image processing. It builds up a virtual hexagonal grid system on memory space of computer. Then, processing can be implemented on such virtual space. Finally, resulted data can be mapped back to rectangular architecture for display (see Figure 5).

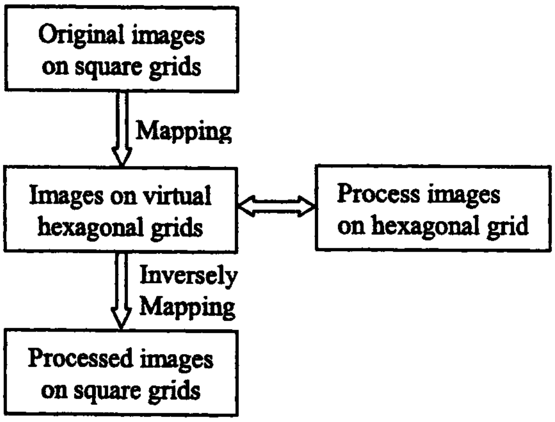


Figure 5. Flowchart of image processing on virtual Spiral Architecture

In order to make virtual Spiral Architecture keep the resolution of the original image, it uses the same area size for each hexagonal grid as square grid to represent a given image on hexagonal grids.

Let N denote the number of square grids which are connected to a particular hexagonal grid and let s_i represent the size of overlapped area between square grid i, one of connected square grid, and the hexagonal grid (see Figure 6). We define the size of grid is 1 unit area, then, the percentage of overlapped area in a referenced hexagonal grid is,

$$p_i = s_i/1 \times 100\% = s_i. \tag{1}$$

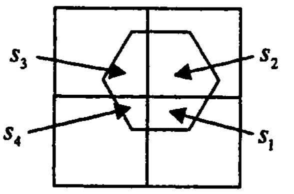


Figure 6. The relationship between virtual hexagonal grids and the connected square grids.

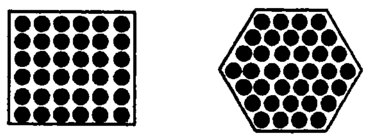
Besides the size of grid, the gray value of a hexagonal gird needs to be worked out as well. After investigation the different contribution of each connected square grid's gray value to the referenced hexagonal grid (see Figure 6), the gray value of mimic hexagonal grid is calculated as the weighted average of the gray values of the connected square grids as,

$$g_{Hex} = \sum_{i=1}^{N} (g_{Squ_i} \cdot p_i) \tag{2}$$

where g_{Hex} denotes the gray value of hexagonal grid, and g_{Squ_i} denotes the gray value of square grid in the area of s_i.

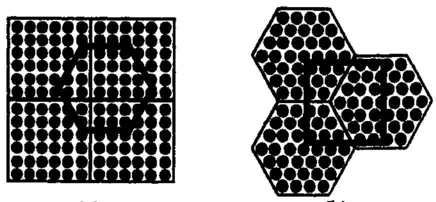
However, for digital images, it is hard to calculate the size of a pixel, so approximation methods must be used. In virtual Spiral Architecture and rectangular architecture, each grid is considered as a set which is composed of many small points (see Figure 7). Then, alternatively, the size of grid and the size of overlapped area between hexagonal grid and square grid (see Figure 8) can be approximated by calculating the number of small points. The accuracy of approximation will be improved when the number of small points in a grid is increased. But this will correspondingly increase computational complexity. In our research, each square and hexagonal grid is assumed to be composed of 100 small points which are uniformly distributed in the grid.

This mimicking scheme assumes that the light intensity of each small point within a pixel is uniformly distributed. In order to calculate the contribution of each of the several related square pixels to the mimicked hexagonal pixel, this scheme assumes that the contribution is decided by only the size of overlapped area. This unavoidably will introduce certain loss of resolution of image information which results in blur effects. In order to avoid the effects of approximation on our results, we applied the similar algorithm on the virtual SA processed square images to make the results comparable.



(a) (b)

Figure 7. One grid of, (a) rectangular, (b) hexagon, that is composed of many small points.



(a) (b)

Figure 8. The size of overlapped area that can be calculated by the number of small points in (a) rectangular, (b) hexagon.

5. Gradient-Based Edge Map

To study the effect of using the virtual Spiral Architecture for image processing, we investigate the gradient-based edge detection algorithms within virtual Spiral Architecture.

In this paper, in order to avoid the effect introduced during thresholding operation, we directly compared the gradient images that are acquired on square architecture and Spiral Architecture respectively.

In the conventional square architecture, the gradient of an input image $f(x,y)$ at location (x,y) is defined as the two-dimensional column vector:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix} \tag{3}$$

The magnitude of this vector, denoted by $mag(\nabla\vec{f})$, is:

$$\nabla f = mag(\nabla \mathbf{f}) = \sqrt{G_x^2 + G_y^2} \tag{4}$$

Computation of the gradient of an image is based on obtaining the partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ at every pixel location. Let the 3×3 area shown in Figure 9(a) represent the gray levels in a neighbourhood of an image. One of the simplest ways to implement a first-order partial derivative at the central point z_5 is to use the following approach:

$$\begin{cases} G_x = [(z_7+z_8+z_9)-(z_1+z_2+z_3)]/6 \\ G_y = [(z_3+z_6+z_9)-(z_1+z_4+z_7)]/6 \end{cases} \tag{5}$$

In this formulation, the difference between the third and first columns of the 3×3 image region is used to approximate the derivative in the x-direction, and the difference between the third and first rows approximates the derivative in the y-direction. The mask shown in Figure 10(a) and (b) called the Prewitt operators, can be used to implement these two equations respectively.

Implementation of the gradient operator in hexagonal images involves computing equivalent masks for the hexagonal case, as shown in Figure 9(b). In this paper, we take use of a triple-diagonal gradient-based edge detection method [14] on Spiral Architecture. In this method, three gradient components on three diagonal directions instead of one approximated gradient direction are computed at each point (pixel).

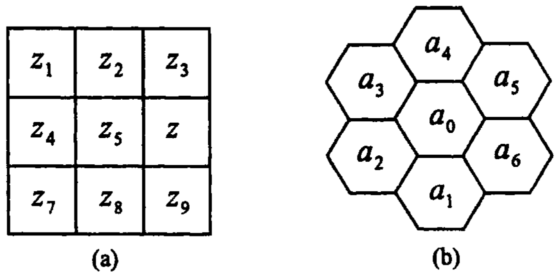


Figure 9. Masks in (a) rectangular and (b) hexagonal architectures

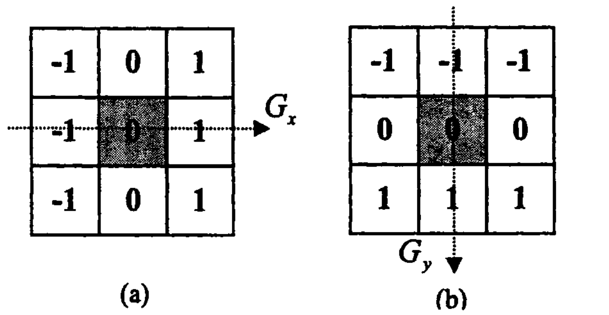


Figure 10. Prewitt masks used to compute the gradient at the central shadowed point in (a) x-direction, and (b) y direction on square architecture

Let $\vec{G}$ be the gradient of the brightness function f at a given reference point, $\vec{G}_1$, $\vec{G}_2$, and $\vec{G}_3$ be the three gradient components in the three directions respectively for a given reference point as shown in Figure 11(a)-(c). We call the three gradient components *triple-diagonal gradient components*. In the real Spiral Architecture, the distance between the reference point and any of its neighbouring point is same. Without loss of generality, we assume that the distance is 1. It is easy to see that:

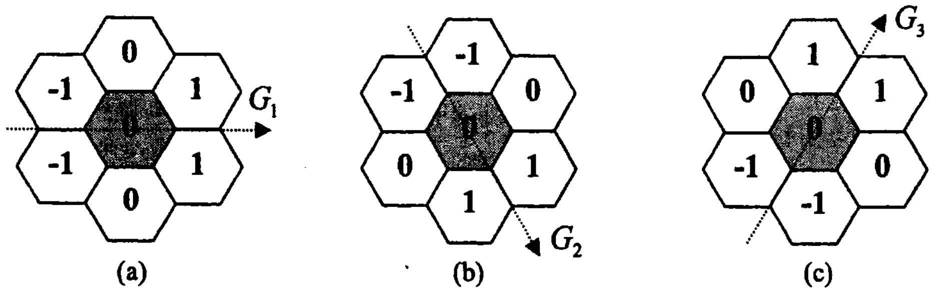


Figure 11. Prewitt masks used to compute the gradient at the central shadowed point in 3 directions on hexagonal architecture.

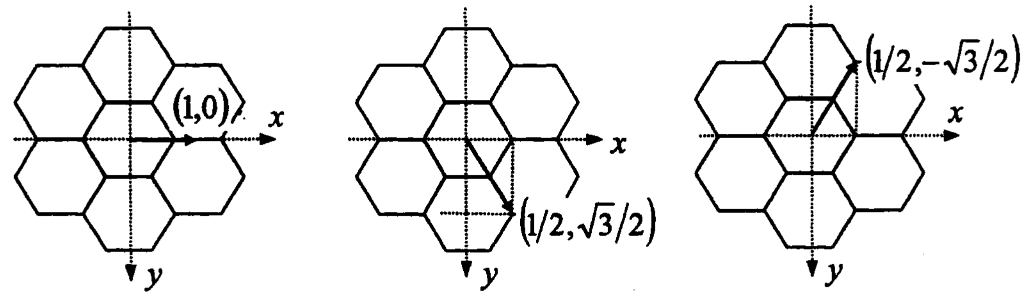


Figure 12. Three unit vectors indicated as the bold angled lines.

$$\vec{G} = G_1 \cdot (1,0) + G_2 \cdot (\frac{1}{2}, \frac{\sqrt{3}}{2}) + G_3 \cdot (\frac{1}{2}, -\frac{\sqrt{3}}{2}) \quad (6)$$

where the three vectors corresponding to the three diagonal directions. The three unit vectors are indicated in Figure 12.

Then the magnitude of this vector, denoted by $mag(\nabla\vec{f})$ can be got as:

$$mag(\nabla\vec{f}) = \sqrt{\left(\frac{2G_1 + G_2 + G_3}{2}\right)^2 + \left(\frac{\sqrt{3}(G_2 - G_3)}{2}\right)^2} \quad (7)$$

Similarly, we use the masking method to compute G_1, G_2, and G_3, as shown in Figure 12(c)-(e), where,

$$\begin{cases} G_1 = [(a_5 + a_6) - (a_2 + a_3)]/(2\sqrt{3}) \\ G_2 = [(a_1 + a_6) - (a_3 + a_4)]/(2\sqrt{3}) \\ G_3 = [(a_4 + a_5) - (a_1 + a_2)]/(2\sqrt{3}) \end{cases} \quad (8)$$

6. Experimental Results

Images represented on virtual Spiral Architecture are shown in Figure 13(a) and (d).

Edge detection is performed on two test images, 'lena' and 'duck' images. The results for the edge detector are shown in Figure 13(b) to (f), where images (b)(e) are gradient maps implemented on square architecture, images (c)(f) are gradient maps implemented on virtual Spiral Architecture. In our experiments, all of the gradient values are firstly normalized by the maximum gradient value and then mapped into the range of 0~255 for display purpose.

From the results, we can see that, even though the overall intensity of the gradient maps implemented on virtual SA is relatively lower than that implemented on square architecture, the gradient maps on virtual Spiral Architecture keep enough edge information. The lower intensity is mainly due to the blur effect when we map the output gradient map from its virtual Spiral Architecture to square architecture.

Moreover, if we focus on the edge maps of the circular edges, e.g., the edges of the eyes of the duck (see Figure 13 (f)), we can find that a thinner edge map is acquired on hexagonal-based processing. This is mainly due to the good performance of the triple-diagonal edge detectors on hexagonal images that are able to highlight the strongest response not only on horizontal and vertical directions, but also on the other two diagonal directions. However, in order to obtain the similar performance on square images, we have to use another two additional masks in diagonal and inverse diagonal directions. This will unavoidably introduce more computation expense.

7. Conclusion

In this paper, we investigate the gradient-based edge detection on rectangular architecture and hexagonal architecture. Since currently there is no hardware to sample and display hexagonal images, we take use of the newly proposed virtual Spiral Architecture for our experiments. From our results, we can find that the gradient maps acquired on Spiral Architecture provide a more clear circular edge than that acquired on square architecture.

8. References

[1] R. M. Mersereau, "The processing of hexagonally sampled two-dimensional signals," Proceedings of the IEEE, vol. 67, pp. 930-949, 1979.

[2] L. Middleton and J. Sivaswamy, "Edge detection in a hexagonal-image processing framework," Image and Vision Computing, vol. 19, pp. 1071-1081, 2001.

[3] D. P. Petersen and D. Middleton, "Sampling and Reconstruction of Wave-Number-Limited Functions in N-Dimensional Euclidean Spaces," Information and Control, vol. 5, pp. 279-323, 1962.

[4] J. P. Mylopoulos and T. Pavlidis, "On the topological properties of quantized spaces, I." Journal of the ACM, vol.18(2), pp.239-246, 1971.

[5] J. P. Mylopoulos and T. Pavlidis, "On the topological properties of quantized spaces, II. connectivity and order of connectivity," Journal of the ACM, vol. 18(2), pp. 247-254, 1971.

[6] J. Serra, "Introduction to Mathematical Morphology," Computer Vision, Graphics, and Image Processing, vol. 35, pp. 283-305, 1986.

[7] E. S. Deutsch, "Thinning algorithms on rectangular, hexagonal, and triangular arrays," Communications of the ACM, vol. 15, pp. 827-837, 1972.

[8] R. C. Staunton, "An analysis of hexagonal thinning algorithms and skeletal shape representation," Pattern Recognition, vol. 29, pp. 1131-1146, 1996.

[9] A. Rosenfeld and J. Pfaltz, "Distance functions on digital pictures," Pattern Recognition, vol. 1, pp. 33-61, 1968.

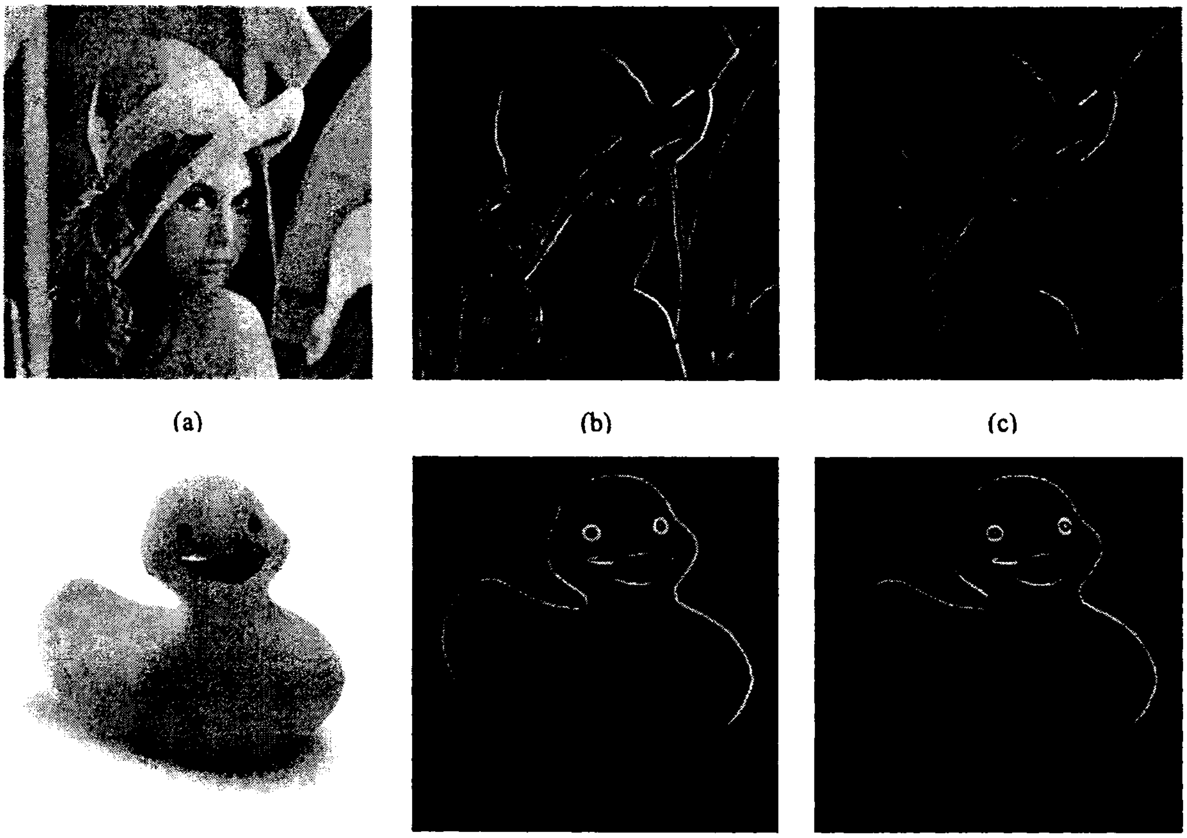
[10] P. Sheridan, "Spiral Architecture for machine vision," PhD Thesis, University of Technology, Sydney, 1996.

[11] C.A. Wüthrich and P. Stucki, "An Algorithmic Comparison Between Square- and Hexagonal-Based Grids," Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing, vol. 53, pp. 324-339, 1991.

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

[13] 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, pp. 399-405, 2004.

[14] Q. Wu, X. He, and T. Hintz, "A Triple-Diagonal Gradient-Based Edge Detection," Proceedings of the 6th IASTED International Conference on Computers, Graphics, and Imaging (CGIM2003), pp. 244-249, 2003.



(a) (b) (c)

(d) (e) (f)

Figure 13. Gradient-based edge maps of the image 'lena' and 'duck'.
(a)(d) image represented on virtual SA, (b)(e) gradient maps implemented on square architecture, and (c)(f) gradient maps implemented on virtual SA.