Neural Network Based Image Edge Detection within Spiral Architecture

Xiangjian He, Tom Hintz and Qiang Wu
Department of Computer Systems
Faculty of Information Technology
University of Technology, Sydney
PO Box 123, Broadway 2007
NSW, Australia

Abstract *Edge detection of 3-dimensional physical objects in a 2-dimensional image is one of the main research areas of computer vision. Edge point is defined as a hexagonal pixel where an abrupt change in grey level takes place. In this paper, we present a new approach to edge detection based on a neural network. Our task is divided into two parts:*

1. *Mapping typical grey levels in primitive small image blocks (groups of adjacent 7 hexagonal pixels) within the Spiral Artchitecture.*
2. *Combining this locally derived information (including presence, orientation and strength of edge) in a consistent way.*

This new edge detection scheme, because of its parallel structure, is fast and can be easily implemented.

Keywords: Edge Detection, Computer Vision, Neural Networks, Image Processing

1 Introduction

Edge detection plays a key role in computer vision, image processing and related areas. It is a process which detects the significant features that appear as large delta values in light intensities. At an early stage of computation in a large scale computer vision application, edge points are detected from the original image. Edge points contain useful structural information about objects' boundaries in a compact form. It requires a relatively small amount of memory space for storage. If needed, a replica image can be reconstructed from its edge map. Thus edge detection serves to simplify the analysis of images by dramatically reducing the amount of data to be processed. Edge information is useful in many image understanding problems such as segmentation, registration, feature and line extraction, and stereo matching [2].

During the last three decades, many algorithms have been developed for edge detection (e.g. [6] and [13]), among which the most important ones are the Marr-Hildreth [10] method based on detecting zero crossings at the output of Laplacian-Gaussian operators of different widths, Haralick's [3] facet model based method that uses the zero crossings of a second directional derivative of Gaussian edge operator, and Canny's [1] computational approach to edge detection by formulating the task as a numerical optimization problem. There are also other algorithms for refining step edges derived from Gaussian operators [14].

There are also some other edge detection schemes in which neural networks are either implicitly or explicitly involved. Moura [11] and Lepage [8] have developed, independently, a so called competitive-cooperative network to refine edge patterns obtained from simple gradient operators, e.g. Sobel's or Prewitt operators. The networks shown in their paper consist of several neurons corresponding to pixels in a small neighborhood of image. The neurons have a competitive/cooprative effect on each other. In Moura's method, gradient magnitude and direction for image intensity (derived from Sobel's operators) are given to the network and based on that the network enforces edge patterns normal to gradient and weakens edge patterns along gradient direction. In Lepage's method, however, the competitive/cooperative network idea is used in a multi-scale scheme. The network consists of several layers corresponding to different resolution levels, and at each layer edges are detected using lateral connections and Sobel's operator. Edges at coarser scales will then reinforce corresponding edge patterns at finer scales through competitive/cooperative connections, so those edges in finer scales which are not reinforced will tend to disappear.

Furthermore, Etemad et al. [2] presented an alternative approach to edge detection based on neural networks. Their method has features such as:

1. High speed and simplicity. Since it only relies on the non-linear mapping capability of a simple neural network that directly provides binary edge patterns as well as edge strangth.
2. Edge detection independent of edge strength.
3. Incorporating contex in edge detection without going to lower resolutions.
4. Robustness to noise.
5. Extendability to corner and line detection as well as multi-resolution methods.

In this paper, we present a new algorithm for edge detection based on neural network by adapting the detection scheme shown in [2]. Our image algebra is established on the Spiral Architecture and the edge definition is based on the Gaussian operator [9].

The Spiral Architecture described by Sheridan [12] and futher elaborated by He and Hintz [5] is a relatively new data structure for computer vision. The image is represented by a collection of hexagons of the same size (in contrast with the traditional rectangular representation). The importance of the hexagonal representation is that it possesses special computational features that are pertinent to the vision process.

As image data is typically massive in mature, it is always desirable to devise methods that involve parallel processing of data or methods that can easily be implemented using parallel algorithms. On the other hand, due to massive amounts of image data, except for a small number of applications where image sizes are small, it is not feasible to process the whole image concurrently. An important characteristic of typical images is that there is a large amount of local spatial correlation among pixels. Thus, in this paper, we establish an efficient local processing scheme that has many parallel computational structures. We also show a method to combine information extracted from the local processing to arrive at consistent global results.

The context of this paper is arranged as follows. We briefly introduce the Spiral Architecture in Section 2. In Section 3, a new definition of edge point in the Spiral Architecture is developed. This is followed by an edge detection algorithm for a block of seven hexagonal pixels based on a neural network in Section 4, and the global edge detection in Section 5. We conclude in Section 6.

2 The Spiral Architecture

Traditionally, an image is considered as a collection of rectangular pixels of the same size. Since the late 1990s, edge detection within a relatively new data structure, called the Spiral Architecture has been considered by He et al. in their papers [4] and [7]. This significantly extends and simultaneously makes practical the Spiral image structure. In the Spiral Architecture, an image is represented as a collection of hexagonal picture elements. As an example, a collection of seven pixels is displayed in Figure 1. The distribution of cones on the retina

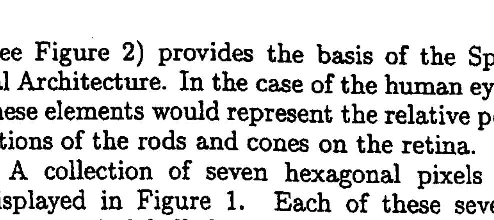

Figure 1: A cluster of 7 hexagons.

(see Figure 2) provides the basis of the Spiral Architecture. In the case of the human eye, these elements would represent the relative positions of the rods and cones on the retina.

A collection of seven hexagonal pixels is displayed in Figure 1. Each of these seven hexagons is labelled consecutively with numbers 0, 1, 2, 3, 4, 5 and 6. These numbers

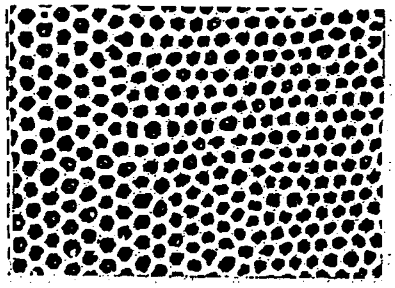

Figure 2: Distribution of cones on the retina.

are refered to as *Spiral Addresses* of individual pixels [12].

3 Edge Definition within Spiral Architecture

In the following, the image brightness function will be parameterized. A large change in image brightness over a short spatial distance indicates the presence of an edge.

Let $L : \Re^2 \to \Re$ be a brightness function of an image which maps the coordinates of a pixel, (x, y) to a value in light intensities. Lindeberg defined edges from the continuous grey-level image function $L : \Re^2 \to \Re$ as the set of points for which the gradient magnitude assumes a maximum in the gradient direction [9]. This can be further described as follows.

Let $\bar{v}$ be the gradient of $L(x,y)$ at (x,y), and $L_x(x,y)$ and $L_y(x,y)$ be the derivatives of $L(x,y)$ with respect to x and y. Denote $L_x(x,y)$ and $L_y(x,y)$ L_x and L_y respectively. Then $\bar{v}$ is parallel to (L_x, L_y). Furthermore, the derivative of $L(x,y)$ in gradient direction at (x,y) is $\sqrt{L_x^2 + L_y^2}$. We denote this derivative by $L_{\bar{v}}$ i.e.,

$$L_{\bar{v}} = \sqrt{L_x^2 + L_y^2}. \tag{1}$$

Hence, by Lindeberg's definition, (x,y) is an edge point (or edge pixel) if and only if $L_{\bar{v}}$ assumes a maximum at (x,y).

Lindeberg's work assumed a continuous space. In this section, we give a discrete approach within the Spiral Architecture.

3.1 Approach to gradient of L

Given discrete data in the Spiral Architecture, if we assume that the distance between centres of two neighbouring hexagonal pixels is 1 and the Cartesian coordinates of the Hexagon with Spiral Address 0 is (x,y) (Figure 1), then the hexagons with the spiral addresses 1, 2, 3, 4, 5 and 6 have Cartesian coordinates $(x, y-1)$, $(x - \frac{\sqrt{3}}{2}, y - \frac{1}{2})$, $(x - \frac{\sqrt{3}}{2}, y + \frac{1}{2})$, $(x, y+1)$, $(x + \frac{\sqrt{3}}{2}, y + \frac{1}{2})$ and $(x + \frac{\sqrt{3}}{2}, y - \frac{1}{2})$ respectively. Let us denote the gradient of L at (x,y) by $G(x,y)$. We implement $G(x,y)$ by

$$\begin{aligned}
& G(x,y) \\
= \; & [L(x, y-1) - L(x,y)](0,-1) \\
+ \; & [L(x - \frac{\sqrt{3}}{2}, y - \frac{1}{2}) - L(x,y)](-\frac{\sqrt{3}}{2}, -\frac{1}{2}) \\
+ \; & [L(x - \frac{\sqrt{3}}{2}, y + \frac{1}{2}) - L(x,y)](-\frac{\sqrt{3}}{2}, \frac{1}{2}) \\
+ \; & [L(x, y+1) - L(x,y)](0,1) \\
+ \; & [L(x + \frac{\sqrt{3}}{2}, y + \frac{1}{2}) - L(x,y)](\frac{\sqrt{3}}{2}, \frac{1}{2}) \\
+ \; & [L(x + \frac{\sqrt{3}}{2}, y - \frac{1}{2}) - L(x,y)](\frac{\sqrt{3}}{2}, -\frac{1}{2})
\end{aligned} \tag{2}$$

Let us use the cluster of seven hexagonal pixels as displayed in Figure 3 as an example of an image block to calculate the gradient of L at the central pixel. In Figure 3, the grey values

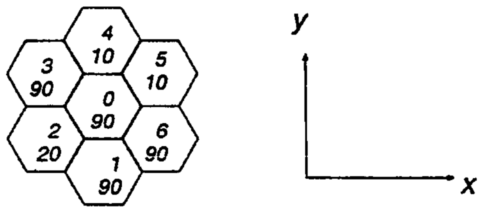

Figure 3: Gradient example in an image block.

of individual pixels are indicated by the numbers under the corresponding Spiral Addresses. For example, the grey value at Spiral address 1 is 90. By Equation 2, the gradient at spiral address 0 is

$$(90 - 90)(0, -1) + (20 - 90)(-\frac{\sqrt{3}}{2}, -\frac{1}{2})$$

$$
\begin{aligned}
&+ \quad (90-90)(-\frac{\sqrt{3}}{2},\frac{1}{2}) + (10-90)(0,1) \\
&+ \quad (10-90)(\frac{\sqrt{3}}{2},\frac{1}{2}) + (10-90)(\frac{\sqrt{3}}{2},-\frac{1}{2}) \\
&= \quad (-5\sqrt{3},-85) \qquad (3)
\end{aligned}
$$

3.2 Decide edges at central pixels

Given any Spiral address i, denote the values of $G(x,y)$ at i by G_i, and the corresponding derivative at gradient direction by L_i. Let P_i^+ and P_i^- be grey values of the neighbouring pixels of i in the positive gradient direction and in the negative gradient direction respectively. Then we define

$$L_i = \frac{|P_i^+ - P_i^-|}{2}. \qquad (4)$$

For example, in Figure 3, the grey levels of pixel 1 and pixel 4 are 90 and 10 respectively, and pixel 1 and pixel 4 are the neighbouring pixels of pixel 0 in the positive gradient direction and in the negative gradient direction respectively. Hence,

$$L_0 = \frac{|90-10|}{2} = 40. \qquad (5)$$

In the following, we propose a procedure to determine whether the central pixel of a block of seven pixels as displayed in Figure 1 is an edge point. Without loss of generality, we assume that the seven pixels for the block have Spiral addresses as displayed in Figure 1. Denote the angle between G_0 and $(1,0)$ A_0.

- If $0 \le A_0 < \frac{\pi}{6}$ or $\pi \le A_0 < 1\frac{\pi}{6}$, then it is obvious that pixels 2 and 5 among the neighbouring pixels of 0 contribute the most to the change of brightness (or grey value) at 0 in the gradient direction. Hence,
  - if ($L_0 \ge L_2$ and $L_0 > L_5$) or ($L_0 > L_2$ and $L_0 \ge L_5$),
    we record 0 as an edge pixel. This is because that L_0 in these cases is a local maximum along the gradient direction at 0.
- Similarly if $\frac{\pi}{6} \le A_0 < \frac{2\pi}{3}$ or $1\frac{\pi}{6} \le A_0 < 1\frac{2\pi}{3}$, then
  - if ($L_0 \ge L_1$ and $L_0 > L_4$) or ($L_0 > L_1$ and $L_0 \ge L_4$)
    record 0 as an edge pixel.
- And if $\frac{2\pi}{3} \le A_0 < \pi$ or $1\frac{2\pi}{3} \le A_0 < 2\pi$, then
  - if ($L_0 \ge L_3$ and $L_0 > L_6$) or ($L_0 > L_3$ and $L_0 \ge L_6$)
    record 0 as an edge pixel.

The above procedure implies that if 0 is an edge pixel, then L_0 is a local maximum in the gradient direction.

4 Edge Detection in Primitive Blocks

The basic idea for the edge detection in this paper is the following: if for each primitive image block (a block of seven pixels) one can correctly distinguish the edge (if there is an edge) and can proberly combine the edge information (e.g., existence, orientation and strength) extracted from adjacent blocks, then a set of consistent boundaries for the whole image can be obtained.

So the problem can be broken into two parts:

1. Given any grey level pattern in a small block of an image, find the corresponding 'most likely' edge pattern. We discuss this problem in this section.
2. Combine the information derived from neighboring blocks in a consistent way. This problem will be discussed in the next section.

The problem to find the 'most likely' edge pattern is basically to find a nonlinear mapping from typical grey level patterns at the input to their 'most likely' edges at the output. For each pattern presentation (patterns like those in the first and second columns of Figure 4) at the input we require the traget output to be the most likely edge pattern (e.g., edge patterns like those in the third column in Figure 4). What we mean by 'most likely' here is simply what humans may claim about the presense and orientation of the edge in the given input.

Considering the complexity of the mapping one could expect that just two layers are not

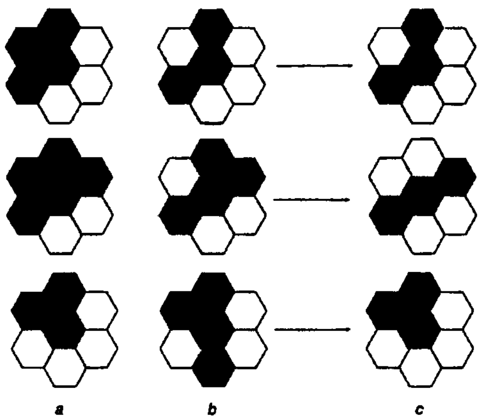

Figure 4: a. Perfet input patterns; b. Impect input patterns; c. Most likely edge patterns.

enough. Hence, in this paper, we construct a neural network for edge detection consisting of three (i.e., input, hidden and output) layers as shown in Figure 5 The hidden layer contains

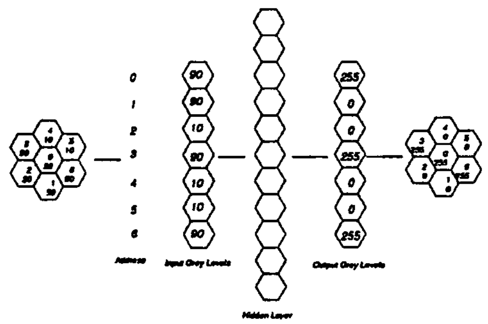

Figure 5: Three layer neural network.

information about the the derivative of $L(x, y)$ at each pixel of primary block in the gradient direction. Hence, the hidden layer require information of grey-levels of all pixels in the primary block and of the pixels adjacent to the primary block (see, for example, Figure 6a).

4.1 Segmentation of primary blocks

In the following, we decribe the procedures for determining edge pixels or non-edge pixels in any primary block based on the neural network esblished above. See Figure 6 for the illustration.

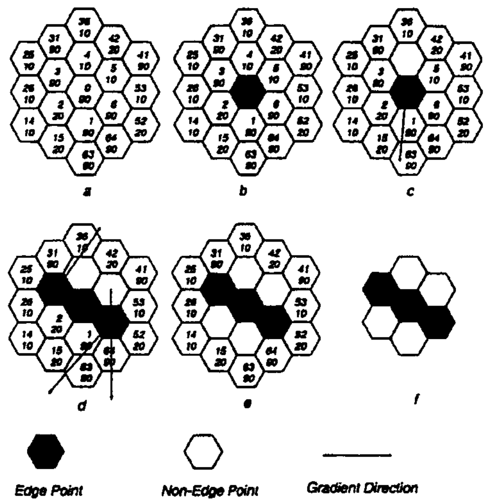

Figure 6: Segmentation steps. a. Grey-levels on extended block; b. Step 1; c. Step 2; d. Step 3; e. Step 4; f. Final results.

Step 1. We apply the edge definition described in the previous section to the central pixel of the primary block. If the central pixel is an edge point, then go to Step 2. Otherwise, we claim that none of the pixels in the block is an edge point. For example, in Figure 6, we have that

$$L_0 = 40, \ L_4 = 40, \ L_1 = 0. \tag{6}$$

Hence, L_0 is a local maximum in the gradient direction of pixel 0. Thus, pixel 0 is recorded as an edge point.
Step 2. In the gradient direction (both positive and negative), compare the differences of the grey-levels between the central pixel and the two pixels in the gradient direction. The pixel of the two with greater difference is recorded as a non-edge point. For example, in Figure 6, the grep level difference between pixels 0 and 1 is 0, and between pixels 0 and 4 is 80. Hence, pixel 4 is recorded as a non-edge point.
Step 3. This step is to determine the edge at any other pixel in the primary block, which is not yet classified as an edge point or non-edge point, and which is next to a non-edge point. Suppose a is such a pixel. We further assume

that the vector from this pixel to the central pixel is V_1 and the vector from the same pixel to the non-edge pixel is V_2. Then we do the following.

1. Calculate the gradient of L at a, denoted by $G(a)$, as we did at the central pixel.
2. Calculate the absolute values of $G(a) \cdot V_1$ and $G(a) \cdot V_2$. If $|G(a) \cdot V_1| \geq |G(a) \cdot V_2|$, we record a as a non-edge point. Otherwise, we record a as an edge point.

In Figure 6, pixels 3, 5 and 6 are such pixels, of which pixel 3 and 6 are classified as edge points and pixel 5 is recorded as a non-edge point.
Step 4. Record all other undetermined pixels as non-edge pixels. For example, pixels 1 and 2 are the pixels which are classified as non-edge points in this step.

The final edge results of the primary block are shown in Figure 6f.

5 Combining Edge Information of Primary Blocks

In any dege detection scheme in addition to edge presense and orientation one has to give a relative measure of edge strength which can be used in discarding weak as well as parasitic edges at the output. After the required mapping is achieved we need to combine the edge information derived from adjactive blocks to get boundaries that are as consistent and smooth as possible and to decrease the effect of noise and spurious edges.

There are several ways of achieving this. For example, one simple way is to use the following observation.

As we sweep across the image each pixel appears in several window blocks. In our example of seven pixel windows each pixel at a appears in its own 'principal block', which has that pixel a at its centre, as well as six surrounding blocks. For example, in Figure 7 all seven blocks contain pixel 0. Of course, pixels on the image boundaries are contained in fewer numbers of windows. Here, we concentrate on interior pixels.

If a pixel is along an edge and if our proposed network performs its job correctly it will assign a 255 (corresponding to an 'edge point') to that pixel several times. See, for example,

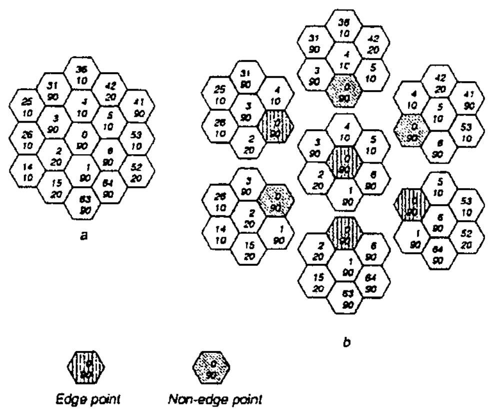

Figure 7: a. Grey-levels on extended block; b. Seven separated blocks.

Figure 7). In this figure, pixel 0 is on the edge and therefore assigned to be on the edge by four windows consistently. But on real images where noise is present, it makes sense to use a majority of votes based on the accumulated vote given by all such windows compared to a threshold. Also it is reasonable that in this majority voting policy we give more weight to the principal block because logically it is the window that is most informative about its central pixel. Suppose we want to claim that a is an edge point if the principal block and one other block, or at least three out of six blocks except for the principal block assign 255 (grey level) to this pixel. We claim that a is a non-edge point otherwise. In order to implement this rule, we add up the votes given by all seven windows that contain pixel a, with two votes for the principal block, and then compare the accumulated number to three and claim an edge point only if the number is greater than or equal to three. Of course, we need to consider blocks that are on the border separately.

In order to increase the accuracy of edge detection, edge strength could also be considered in our decision process either in the previous section or in this section. These are classified as 'hard decision approach' and 'soft decision approach'. For the hard decision approach, one can ignore all weak edge patterns with strengths less than a threshold and just consider strong edges in the previous section, i.e., each window votes for a pixel to be an edge point only if our local edge detection in that window gives a '255' at the corresponding output neuron and the edge strength output is above some threshold. For the soft decision approach, one can multiply the vote of each window for each pixel by the edge strength derived at that window and add up all the combined votes and strengths for a pixel and compare the result to a final threshold. An edge is declared if this combined result is more than a preset threshold.

6 Conclusion

In this paper, we have done the following:

1. We implemented the gradient and the derivative of the grey level function locally. The gradient and the derivative were used to re-define edge points.

2. A neural network was set up for edge detection on a 'primary block'.

3. A global edge detection scheme was proposed by combining the edge information on all primary blocks.

References

[1] J. F. Canny, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986), 679–698.

[2] Kamran Etemad and Rama Chellappa, *A neural network based edge detector*, Proc. IEEE International Conference on Neural Networks (San Francisco, California), 1993, pp. 132–137.

[3] R.M. Haralick, *Digital step edges from zero crossings second directional derivatives*, IEEE Transactions on Pattern Analysis and Machine Intelligence (1984), 58–68.

[4] Xiangjian He and Tom Hintz, *Multi-scale edge detection based on Spiral Architecture*, 1998 International Computer Symposium (Taiwan), 1998.

[5] ______, *Object contour extraction in spiral space*, Australian Computer Science Communications **21** (1999), 182–192.

[6] ______, *Refining occluded object recognition within the spiral architecture*, 2000 International Conference on Image Science, Systems and Applications (Las Vegas), 2000, pp. 701–708.

[7] Xiangjian He, Tom Hintz, and Ury Szewcow, *Replicated shared object model for edge detection with Spiral Architecture*, Lecture Notes in Computer Science **1388** (1998), 252–260.

[8] R. Lepage and D. Poussart, *Multiresolution edge detection*, Proc. Interbational Join Conference on Neural Networks (Baltimore), 1992, pp. 438–443.

[9] T. Lindeberg, *Scale-space theory in computer vision*, Kluwer Academic Publishers, London, 1994.

[10] D. Marr and E. Hidreth, *Theory of edge detection*, Proc. Royal Soc., 1980, pp. 187–217.

[11] L. Moura and F. Martin, *Edge detection through cooperation and competition*, Proc. International Join Conference on Neural Networks (Singapore), 1991, pp. 2588–2593.

[12] Phil Sheridan, *Sprial architecture for machine vision*, Ph.D. thesis, University of Technology, Sydney, 1996.

[13] H. Tian, T. Srikanthan, and K. Asari, *A new approach for object boundary extraction based on Heuristic search*, Proc. International Conference on Image Science, Systems, and Technology (Las Vegas), 2000, pp. 693–699.

[14] Qiang Wu, Tom Hintz, and Xiangjian He, *Image edge detection in a mimic spiral architecture*, Proc. of 9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (Czech Republic), 2001, pp. 320–327.