Edge Detection on Spiral Architecture: An Overview

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

Abstract

Gradient-based edge detection is a straightforward method to identify the edge points in the original grey-level image. It is intuitive that in the human vision system the edge points always appear where the grey-level value is greatly changed. Spiral Architecture is a relatively new image data structure that is inspired from anatomical considerations of the primate's vision. In Spiral Architecture, each image is represented as a collection of hexagonal pixels. Edge detection on Spiral Architecture has features of fast computation and accurate localization. In this paper, we briefly review the edge detection methods on Spiral Architecture including the edge focusing technique, bilateral filter, and triple-diagonal gradient. Parallel algorithms for edge detection will be discussed. We will also list problems for future work.

Keywords: Spiral Architecture, edge detection, triple-diagonal gradient, bilateral filter

1. Introduction

Computer vision involves compositions of picture elements (pixels) into edges, edges into object contours and object contours into scenes. The determination of edges depends on detection of edge points (pixels) of a 3-D physical object in a 2-D image. This first step in the process is critical to the functioning of machine vision. As the success of subsequent steps are sensitive to the quality of results at this step, the performance of higher level processes such as extraction of object contours and object recognition relies heavily on the complete and correct determination of edges [1]. Edges contain major image information and need only a small amount of memory storage space compared to the original image. Hence, edge detection simplifies images and thus facilitates image analysis and interpretation [2].

Edge detection is based on the relationship a pixel has with its neighbours. It extracts and localizes points (pixels) around which a large change in image brightness has occurred. A pixel is unsuitable to be recorded as an edge if the brightness around a pixel is similar (or close). Otherwise, the pixel may represent an edge.

There have been many edge detection algorithms proposed in the past. Since 1996, there have been many papers on edge detection based on Spiral Architecture. Spiral Architecture described by Sheridan [3] is a 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. In [4], edge detection using edge focusing technique was proposed. This method starts from the edge detection of a blurred image with a large scale of Gaussian filter. The scale is gradually decreased for finer images. The final edge map is recorded when it will no longer change the edge map with any smaller scale. The second edge detection method proposed by Zhou et al [5] applied a bilateral filter rather than a Gaussian filter to remove image noise. The bilateral filter combines a domain filter like Gaussian filter with a range filter. A range filter gives more weights to those neighbouring pixels with light intensity that is more similar to the reference pixel value. This method has been proved being more efficient for suppressing image noise for edge detection. Another method for edge detection on Spiral Architecture, as shown in [6], was based on triple-diagonal gradient. The gradient of grey-level function was defined as a combination of three vectors in three diagonal directions of hexagonal image structure. Note that an edge point is a pixel at which the gradient magnitude assumes a local maximum. Hence, this method is a more accurate detection mechanism where the gradient is implemented in a more accurate way in the discrete image space.

One important issue in edge detection is to improve the computation efficiency because detection is very computationally intensive. The computation is conducted pixel by pixel, and several dozens of arithmetic operations are performed for each pixel. In order to increase the computation speed, He et al [7] proposed a parallel detection algorithm based on Spiral Multiplication. Spiral Multiplication is a powerful operation defined on Spiral Architecture, which uniformly separates an image into smaller and similar sub-images.

In this paper, we will review the above-mentioned edge detection methods on Spiral Architecture. We will also list the problems for future research work.

2. Spiral Architecture

In Spiral Architecture, an image is represented as a collection of hexagonal pixels. Each pixel has only six neighbouring pixels with the same distance to it. Each pixel is identified by a number of base 7 called a spiral address. The numbered (or addressed) hexagons form the cluster of size 7^n, where n is a positive integer. These hexagons starting from address 0 towards address 7^n tile the plane in a recursive modular manner along a spiral-like curve. As an example, a cluster with size of 7^2 and the corresponding spiral addresses are shown in 1.

Spiral Architecture possesses some geometric and algebraic properties, which are very useful and can be interpreted in terms of a mathematical object, Euclidean ring (Refer to [3] for details). Two algebraic operations have been defined on Spiral Architecture based on spiral addresses. They are Spiral Addition and Spiral Multiplication. These two operations correspond to two transformations on Spiral Architecture, which are translation and rotation with a scaling.

In order to make research results based on Spiral Architecture practically workable with the existing image capture devices, a mimic Spiral Architecture [5] has been widely used.

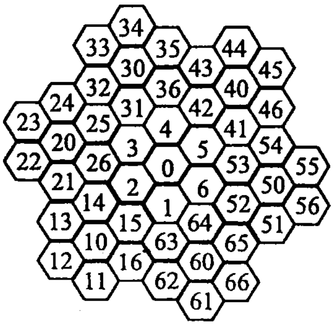


Figure 1: Cluster of size 49 including spiral addresses

In the mimic system, one hexagonal pixel is formed by four square pixels in the traditional system and its grey level value is the average of those four pixels values. Figure 2 shows seven mimic hexagonal pixels.

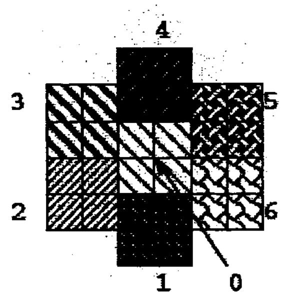


Figure 2. A cluster of 7 mimic hexagons.

3. Edge focusing edge detection

Gaussian Multi-scale Theory is one of the best understood multi-resolution techniques available to the computer vision community [8]. Edge detection using Gaussian Multi-scale Theory ensures not only good performance of detection but also accurate localization of edge points.

Let $f:\Re^2 \rightarrow \Re$ be a brightness function of an image which maps the coordinates of a pixel, (x,y) to a value in light intensities. Let $g:\Re^2 \times (0,+\infty) \rightarrow \Re$ be the Gaussian kernel

$$g(x,y;t)=\frac{1}{2\pi t}e^{-\frac{(x^2+y^2)}{2t}} \quad . \tag{1}$$

Then the Gaussian scale-representation of the original image at scale t is

$$L(x,y;t)=g(x,y;t)*f(x,y)$$
$$=\iint_{\sigma} f(u,v)\frac{1}{2\pi t}e^{-\frac{(x-u)^2+(y-v)^2}{2t}}dudv \tag{2}$$

when $t\in(0,\ \infty)$, and $L(x,y;0)=f(x,y)$.

Scale-space representation is used to suppress and remove unnecessary and disturbing details so that later stage processing tasks can be simplified.

Edge point from a continuous grey-level image represented as $L(x, y; t)$ for given t is defined as a pixel at which the gradient magnitude of $L(x, y; t)$ assumes a local maximum in the gradient direction.

The edge focusing technique starts with a blurred image with a large Gaussian scale (or coarse resolution). An edge map consisting edge points is obtained through this strong blurring. The next step is to gradually focus these edge points by continuously decreasing the scale (or resolution). This method compares the edge maps at different levels of resolution (or with different scales) to decide whether to repeat the detection process with a smaller scale or terminate the process. The edge focusing algorithm essentially consists of the following four steps:

1. **Blurring.** The Gaussian convolution defined in Equation (2) is used to blur the original image in order to identify the essential structure in an image and eliminate noise. The blurred image has lower resolution than the original image.
2. **Registering.** Edge points are located by searching for pixels at which the gradient magnitude of light intensity function reaches a local maximum.
3. **Matching.** Edge focusing is an iterative process using two edge maps of different levels of blurring resolution. These are the map in the previous iteration and the map on the current iteration. The edge map obtained by a stronger blurring scale will have less noise. However, the edge map obtained by using weaker blurring will have more precise edge locations because it is affected less by blurring Gaussian convolution. The matching process compares the two edge maps to obtain a new edge map. The edge points are determined by the edge map with larger scale while the locations of the edge points are on the edge map with smaller scale.
4. **Repeating or Halting.** This focusing process is repeated by going back to Step 1 using a smaller blurring scale until the Gaussian convolution does not have significant 'blurring effects', i.e., there is very little difference between two concessive edge maps.

4. Edge detection using bilateral filter

Recall that a bilateral filter is defined by the combination of a domain filter and a range filter. Let a_0 be a reference pixel and a_1,a_2,a_3,a_4,a_5,a_6 be the six neighbouring pixels of a_0. Then, the range filter is defined by

$$r(a)=e^{-\frac{[f(a_i)-f(a_0)]^2}{2\sigma_r^2}} \quad , i=0,1,2,..6$$

where σ_r is the standard deviation of intensity value distribution. The domain filter defined by

$$g(a)=e^{-\frac{[d(a_i,a_0)]^2}{2\sigma_d^2}} \quad ,i=0,1,2,..6$$

where $d(a_i,a_0)$ is the Euclidean distance between a_i and a_0, and σ_d is the standard deviation in spatial distribution.

Through the domain filter, details can be gradually smoothed. Through the range filter, blurring effects across the edges can be reduced and edges can be preserved better than using domain filtering alone.

Application of the new bilateral smoothing filter produces, for each pixel in the image, a weighted average such that each pixel contributes more significantly to the resulting grey value of the pixel than all its neighbouring pixels. Meanwhile, the pixels with more similar intensity values or closer to the reference pixel contribute more than those with more different values or further away.

Smoothing using the bilateral filter clearly reduces the blurring effect introduced by smoothing to images than Gaussian domain smoothing alone. It implies that edge information would be better retained with less averaging across the kernel under bilateral smoothing.

This can be seen in the experimental results shown in Section 6 below.

5. Using triple-diagonal gradient for edge detection

In this method, three gradient components on three diagonal directions instead of one approximated gradient direction are computed at each pixel.

Let $L_{\bar{v}}$ be the gradient of the brightness function L at a given reference point and $|L_{\bar{v}}^{i}|$ ($i \in \{1,2,3\}$) be the three gradient components in the three diagonal directions respectively for a given reference point as shown in Figure 3. 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 then easy to see that

$$L_{\bar{v}} = |L_{\bar{v}}^{1}|(0,1) + |L_{\bar{v}}^{2}|(\sqrt{3}/2, 1/2) + |L_{\bar{v}}^{3}|(\sqrt{3}/2, -1/2),$$

where the three vectors corresponding to the three diagonal directions.

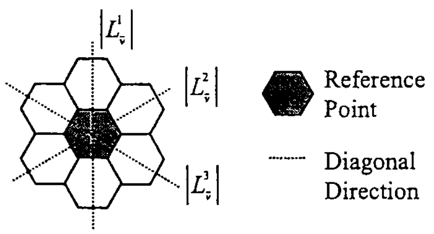


Figure 3: Triple-diagonal gradient components on Spiral Architecture

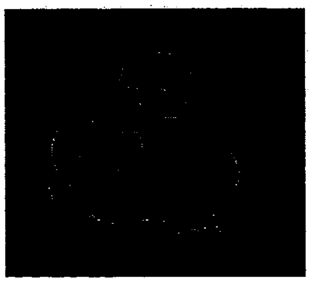
Figure 4. A toy duck represented on mimic Spiral Architecture

Each gradient component is dependent on the grey values at two neighbouring points of the reference in the corresponding diagonal direction. It can be computed using the following equation,

$$|L_{\bar{v}}^{i}| = |L_{i,1} - L_{i,2}|/2,$$

where $L_{i,1}$ and $L_{i,2}$ $i \in \{1,2,3\}$ are the grey values of two neighbouring points of the reference point in the i-th diagonal direction. As can be seen in Equation 2, the difference of two grey values at two neighbouring pixels of a reference point is used to approximate the magnitude of the first-order directional derivative in a diagonal direction at the reference point.

(a)



(b)

(c)

Figure 5. (a) edge map using edge focusing technique, (b) edge map using bilateral filter, (c) edge map using triple-diagonal gradient

6. Experimental results

In this section, we show the experimental results using the three different edge detection methods described above. A toy duck represented on a mimic Spiral Architecture with 256 grey levels is used as the original image for edge detection (see Figure 4). The edge detection results area displayed in Figure 5.

Comparison between Figure 5(a) and Figure 5(b) clearly demonstrates better edge-preserving quality of the bilateral smoothing than Gaussian smoothing. Contour of the mouth is again more complete in the final edge map when bilateral smoothing was applied than it is in the edge map obtained after Gaussian smoothing only.

Comparison between Figure 5(b) and Figure 5(c) shows that the edge noise has been better filtered out using bilateral smoothing than using triple-diagonal gradient. On the other hand, the use of triple-diagonal gradient gives us the sharpest and clearest edge map among the three methods.

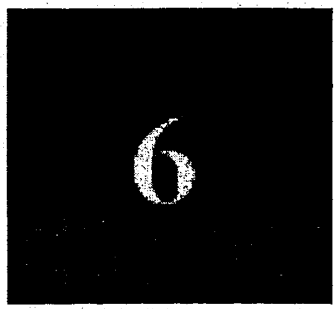
7. Parallel algorithms for edge detection

On Spiral Architecture, image can be easily separated into seven parts by a Spiral multiplication. Each part is a near copy of the original image rotating in some degree (see Figure 6).

Each copy results from a unique sampling of the input image. Each sample is mutually exclusive and the collection of all such samples represents a partitioning of the input image. As the scaling in effect represents the viewing of the image at a lower resolution, each copy has less information. However, as none of the individual light intensities have been altered in any way, the scaled image still holds all of the information contained in the original. So for Gaussian processing as mentioned in previous section, it is still a global Gaussian processing for each slave node but has a smaller convolution operation range. This means that the computational complexity has been both reduced and nicely partitioned without giving away any information for distributed processing.

Hu et al [9] proposed a parallel algorithm based on Spiral Multiplication. We outline the algorithm as follows.

1. Data read from image file is stored into a one-dimensional array of the data structure representing the mimic spiral architecture.
2. The image is then partitioned. The master node forks child processes to handle the communication between the master node and slave nodes. Initial edge map is preset with the gray level equal to 255 (i.e. white) for all the hexagons.



(a)

(b)

Figure 6. (a) Sample image – numeral 6, (b) Separation of the sampled numeral

3. Each slave node receives its share of the work units and starts edge detection process.
   - 3.1 The slave node receives a command from the child process manager. If the value of the scale (as part of the command) is 0 then go to step 3.5; otherwise go to the next step.
   - 3.2 Use Gaussian convolution to blur the image.
   - 3.3 The edge map of the blurred image is compared with the previous one. The new edge map consists of the points in the set of edge points of the sample image such that each of these points has at least one edge point of the blurred image as an adjacent neighbour. Here the difference between two edge maps is measured.
   - 3.4 A new sample image is then constructed as follows. The light intensities are set to 0 at

the points that are more than one pixel away from the set of the new edge points obtained from step 3.2. Go back to step 3.1

3.5 When scale equals to 0 the slave node sends back the image data and finish the current process.

4. The child process manager decides if the difference between current edge map and previous edge map is sufficiently small. If yes, it goes to next step; If not, go back to step 3.1.

   4.1 The child process manager sends a command to ask the slave nodes to send back the edge map.

5. The master node combines the edge maps of subregions and forms the final edge map of the original image.

Figure 7 shows the edge detection result of the sample image shown in Figure 6(a).

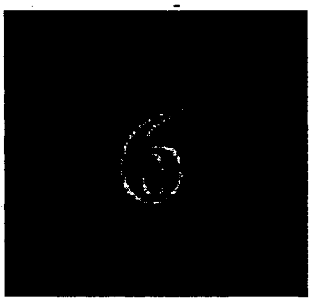


Figure 7. Result of a parallel edge detection algorithm

8. Future work

The edge detection work done up to now is based on the mimic Spiral architecture, on which each hexagonal pixel is formed by 2×2 square pixels. The grey level of each mimic hexagonal pixel is computed as the average of the grey levels of the four square pixels forming the hexagonal pixel. Hence, it blurs the image and reduces the image resolution. This mimic structure does not retain the feature of equal distance from a hexagonal pixel to any of its six neighbouring pixels. This results in an image distortion after image rotation and separation as seen in Figure 5. The structure also introduces errors when the gradient of light intensity function is implemented. This kind of errors occur no matter which detection method is applied.

In order to overcome the above problems, there are two possible solutions: hardware approach and a software approach. In the hardware approach, an image capture device and an image display device must be designed and implemented strictly based on the hexagonal structure. In the software approach, a better mimic technique is needed. The new mimic method should not reduce the image resolution and should retain the feature of equal distance. The hardware approach is now being proposed. As a software approach, Wu et al [10] has designed a new mimic structure, called a Virtual Spiral Architecture. In this approach, a square pixel is divided into many virtual pixels as needed. A hexagonal pixel is constructed by the virtual pixels fallen into its area.

For a more accurate and fast edge detection, we propose a parallel algorithm on the Virtual Spiral Architecture. We outline the algorithm as follows.

1. Master node transfers image from square structure to Virtual Spiral Architecture. Image resolution is retained.
2. Master node separates the image represented on Virtual Spiral Architecture into sub-images using a Spiral Multiplication. No image distortion can be seen after image separation. Each sub-image is then assigned to a slave node.
3. Each slave node compute triple-diagonal gradient at each hexagonal pixel consisting of some virtual pixels. The angel between any two diagonal directions is exactly 60 degrees as seen in the real Spiral Architecture. It is believed that this implementation of gradient is the most accurate.
4. Bilateral filter is constructed on each node for each sub-image and used to blur the image and suppress noise in the image. This should be working better than using Gaussian filter only as proven.
5. Edge focusing technique is used to find the edge map of each sub-image in a slave node.
6. Master node collects the edges of sub-images from slave nodes.
7. Master node applies revised edge thinning and edge linking techniques as shown in [11] to obtain a complete and single square pixel wide edge map.

9. Conclusions

In this paper, we have reviewed the edge detection methods on Spiral Architecture. The reviewed techniques include edge focusing, bilateral filter and triple-diagonal gradient. We have listed a few problems that influence the edge detection results. The problems are from the mimic of Spiral Architecture. These include loss of image resolution, image distortion after rotation and separation and inaccuracy of the gradient of light intensity function. To refine the edge detection on Spiral Architecture, as our future work, we have proposed a parallel algorithm based on our most recent research work called Virtual Spiral Architecture. The parallel algorithm is supposed to be fast and accurate.

10. References

[1] Nalini K. Ratha, Tolga Acar, Muhittin Gokmen and Anil K. Jain, *A distributed edge detection and surface reconstruction algorithm*, Proc. Computer Architectures for Machine Perception (Como, Italy), 1995, pp.149-154.

[2] G. Economou, S. Fotopoulos, and M. Vemis, *A novel edge detector based on nonlinear local operations*, Proc. IEEE International Symposium on Circuits and Systems (London), 1994, pp.293 - 296.

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

[4] Xiangjian He and Tom Hintz, *Refining Edge Detection within Spiral Architecture*, in Australian Computer Science Communications, **Vol. 22**, No.1 (2000), J.Edwards (ed.) pp.113-119, IEEE Computer Society Press, Washington.

[5] Jun Zhou, Qimei Hu and Xiangjian He, *Detection with Bilateral Filtering in Spiral Space*, Proc. of The Second International Conference on Information Technology and Applications, Harbin, China, January, 2004. http://charybdis.mit.csu.edu.au/icita/2004/

[6] Qiang Wu, Xiangjian He and Tom Hintz, *A Triple-Diagonal Gradient-Based Edge Detection*, 6th IASTED International Conference on Computers, Graphics, and Imaging (CGIM'2003), Honolulu, USA, August 2003 CGIM'03, pp.244-249.

[7] Xiangjian He, Qiang Wu, Dan Liu and Li-Hong Zheng, *A Distributed and Parallel Edge Detection Scheme within Spiral Architecture*, Proceedings of 3rd International Conference on Visualization, Imaging and Image Processing, Calgary, ISSN1482-7921, Vol. 1, ACTA Press, pp.371-375, 2003.

[8] Jon Sporring, Mads Nielsen, Luck Florack, and Peter Johansen, *Gaussian scale-space theory*, Kluwer Academic Publishers, 1997.

[9] Qimei Hu, Xiangjian He and Qiang Wu, *Concurrent Edge Detection with Spiral Architecture on Linux*, Proc. of International Conference on Information Technology (IEEE), Las Vegas, April 2003, pp. 524-528.

[10] Qiang Wu, Xiangjian He and Tom Hintz, *Virtual Spiral Architecture*, Proc. International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, 2004, to appear.

[11] Qiang Wu, Xiangjian He and Tom Hintz, *Edge map Improvement on Spiral Architecture*, International Conference on Image Science, Systems and Technology (CISST'2003), Las Vegas, June, 2003, pp. 179-185.